RANDOM LSB STEGANOGRAPHIC AUTHENTICATION USING EIGEN FACE RECOGNISATION TECHNIQUE FOR MOBILE SYSTEM

VENKATESH ASAMPELLI, PATEL RAVI, BAJIRAO SHINDE, TUSHAR RAUT

Abstract*— The aim of this work is to provide 2 levels in terms of security for transaction in banking apps. In order to make more improved security we are making use of "Random Least Significant Bit Steganography" technique in a way that is never used before & to provide user level authentication, face recognition is implemented. Recently, with the awareness of businessmen and consumers and the development of mobile technologies, the potential use of mobile devices in financial applications such as banking and stock trading has seen a rapid increase. We also have proposed the use of LSB-steganography as a means to improve the communication channel for any intrusion by the hackers as a mobile App in an android Operating System.

***Index Terms*— Authenticity, Biometrics, Face Recognition, Legitimacy, LSB Steganography, Mobile Banking, Security, Steganography, Transactions.**

I. INTRODUCTION

In last few decades large technology development raises in various new needs of technology. Financial sector has also no exception due to rapid development. People are approaching all over the world to fulfill their dreams. Any sector needs to understand change of requirement of customer. In order to satisfy financial need for customer banks are taking help of new technology such as internet. Only issue remain is of security.

In this Paper a method for increasing security of the information requested by users with the use of Steganography. In this, instead of direct sending of the information, it is encrypted and hide into a picture using random bit Steganography technique.

Then the picture i.e. an image file is sent to the server. After receiving the image on server, the sample http download socket program downloads the image, decrypts it and decodes to receive the message[1] at the receiver side.

***Venkatesh Asampelli**, B.E.Student, Information Technology, JSPM's ICOER Pune, Maharashtra, India,* (+91)-8983252180

***Patel Ravi Vasudevbhai**, B.E.Student, Information Technology, JSPM's ICOER Pune, Maharashtra, India, (+91)-8806144547*

***Bajirao Shinde**, B.E.Student, Information Technology, JSPM's ICOER Pune, Maharashtra, India,* (+91)-8149805871

***Tushar Raut**, B.E.Student, Information Technology, JSPM's ICOER Pune, Maharashtra, India,* (+91)-8793334540

The encrypted message is then processed on the server to verify user credentials such as user name and password. Once the user passes credential test, camera is switched ON the client side and image is captured. This image is then compared with the server face database images, on successful match – user is taken to the menu screen.

A. Two-level Security

Task of enhancing security include construction of formula for both data encryption and also for hiding pattern. In addition to this, 2- level security using face recognition can be provided. Server should not process any fake request hence concept of custom "Session id" and "Request id" is introduced for every time& auto generated. Implementation of such a security constraints in banking sector not only help to serve customer in better way but also make customer confident and satisfy.

Instead of making use of some previous techniques of Steganography, such as LSB method which are not too difficult to detect, it uses two mathematical formula. To make secure use of Steganography, appropriate image format should be selected such as loss less image formats.

II. LITERATURE SURVEY

A. Related Survey

Steganography technique is no more secrete and also by making use various advanced tools, secrete messages in images can also be detected. Some concepts used in steganography makes detection somewhat easy such as sequential use of pixels to hide data, using same bit from 'n' bit pixel to hide data e.g. LSB, availability of original image which is used to compare with modified image etc.

Those loop holes can be avoided by various ways. Making use of mathematical formula is one of them. Here we make use on two mathematical formula[2]. First formula will automatically generate series of pixel number in which data will hide. Now, Second formula will generate the bit number for 'n' bit pixel in which one bit of data will hide. Using these formulae data will be randomly distributed in image instead of sequential and data hiding bit is also changed each & every time. Using custom images created by the owner (in this case bank) will help to avoid the comparison of the image.

1) To provide an interactive service.
2) To help user with basic mobile handsets to use Internet Banking application.

To make an application for end users who do not have an idea of computer/internet.

ISSN: 2278 – 7798

III. EXISTING SYSTEM

The previous schemes were vulnerable to insider attack and did not preserve anonymity of a user, long and random password for a user to remember, no provision for revocation of lost or stolen smart card and no support for session key agreement during authentication process. To overcome the identified problems we proposed[1] an enhanced biometrics based steganography approach which improves all the identified weaknesses and is more secure and robust for real-life use.

The proposed scheme can withstand the forged authenticating attacks besides providing better communication with the system as the information traveling across the insecure channel is always hidden. The system is very secure as mutual authentication[1] takes place between the communicating parties for processing of the supplied information. Moreover, our scheme is robust, practical and more efficient than other schemes[1].

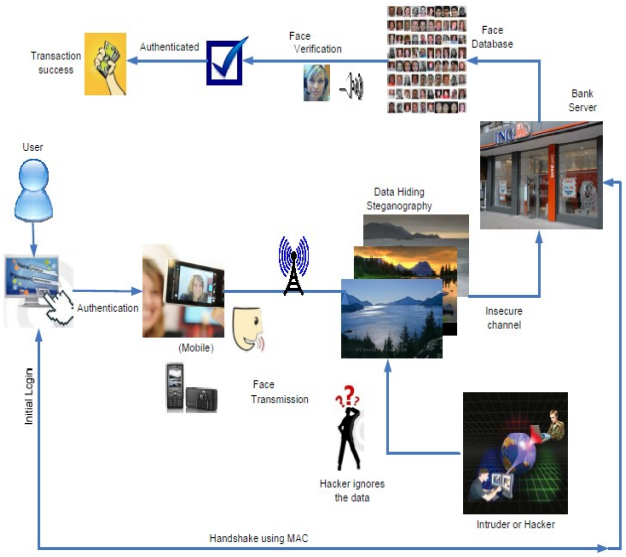

Fig -1: Existing System.

IV. PROBLEM STATEMENT

First of all, steganography program uses a form of basic LSB alteration procedure. This form of steganography is believed to be weak and easily breakable, but selected to use it for two main reasons:

1) It is not too difficult so that numerous experiments related to it can be done.

2) To strengthen this weak security technique by adding more randomness.

The method itself does still encode data by flipping LSBs, but by adding randomness to images and how the data is actually encoded, we can make the cover images and stego-images seem more similar. There are several types of "randomness" that we can employ in our application[5].

First of all, the exact pixels altered to encode each letter are semi-randomly filled. When embedding a message within an image, the very first step in this randomness is to count the number of letters in the message and divide the total number of rows of pixels in the image by that number[5]. For example, if the image is 2000 pixels tall and we want to embed a message 150 letters in length, divide 2000 by 150 to get 10 rows per letter. This is how the number of rows of pixels that could be associated with each letter.

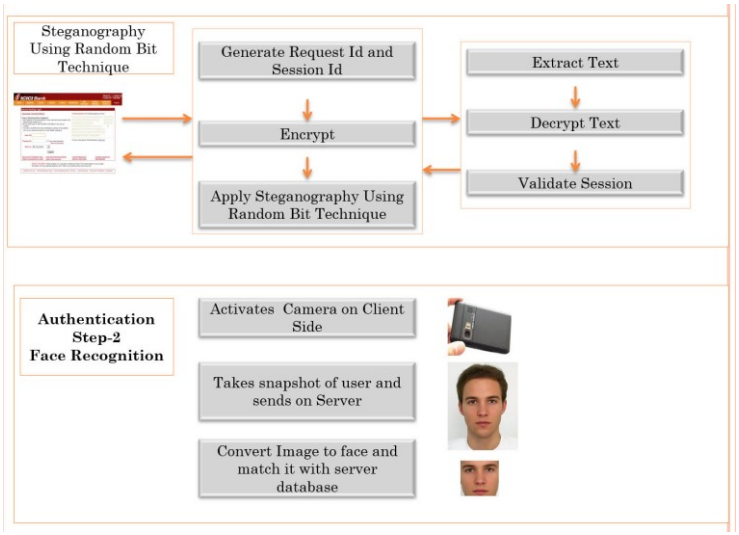

Fig-2: Random Bit Steganography & Face Recognize System Architecture.

V. STEGANOGRAPHY ALGORITHM

Though steganography's most obvious goal is to hide data, there are many other related goals used to judge a method's steganographic strength. These include capacity (how much data can be hidden), invisibility (inability for human to detect a distortion in the stego-object), UN-detectability (inability for a computer to use statistics or other computational methods to differentiate between covers and stego-objects), robustness(message's ability to persist despite compression or other common modification), tamper resistance (message's ability to persist despite active measures to destroy it), and signal to noise ratio (how much data is encoded versus how much unrelated data is encoded). There are three main components, which work in resisting to one another are capacity, UN-detectability, and robustness. Increasing one of these causes while the others to decrease; thus no steganographic method can be perfectly undetectable and robust and have maximum capacity[5].

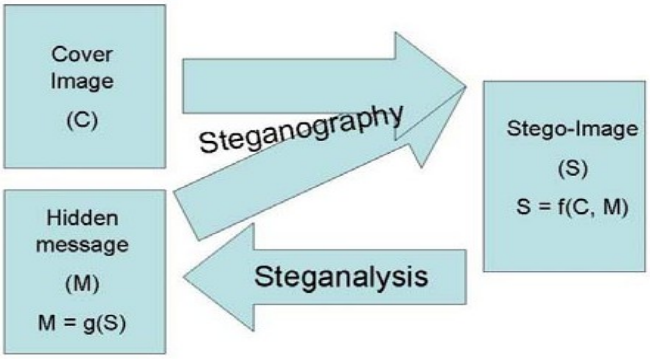

Fig -3: Steganalysis Procedure.

ISSN: 2278 – 7798

A. Request/Session Management and Encryption using private/public key:

Table -1: Request & Session Management

Symbol	Description
C	Client
S	Server
AB->M	A sends M message to B
K_A, K_A^{-1}	Public Key of A, Private Key of A
H(M)	Compute hash value of M using one-way hash function
$\{M\}_{KA}$	Encrypt M using public key of A
$\{M\}_{KA-1}$	Encrypt M using private key of A
$Sig_A(M)$	A signs digital signature on M: $Sig_A(M)=\{H(M)\}_{KA-1}$
sessionID	Session id
T_{i1}, T_{i2}	Time-Stamp used by I message's request, response

Firstly, the client registers a key pair and its related information with X-KRSS service, then selects the session's encryption algorithm, and encrypts the request message. After that process, it signs digital signature on the message, and then encrypts the session key using the server's public key. Finally the client attaches a time-stamp to the message and sends the message to the server after secure processing.
Secondly, After receiving the client's message, at the server-side decrypts the session key using its private key, then decrypts the request message using the session key. According to the key's information of the client, the server receive the client's public key with X-KISS service, and examines the signature's validity. Then the server encrypts the response message using the session key plus signs digital signature on it, and attaches auto generated time-stamp. Lastly, it sends the message to the client.
To the third way, After receiving the response message, the client decrypts it, then examines the signature's validity. Then the client encrypts the request message using the session key plus signs digital signature on it and attaches auto generated time-stamp. And so finally the message is sent to the server.
Finally, the server examines the validity of the Session ID and signature. If valid, encrypts the response message using the session key with signs digital signature on it and attaches other auto generated time-stamp. Hence, it sends the message to the client.

Communications in the proposed Model
Secure communications in Web service secure model:
_C| |S: {Request1}k ,SigC(Request1) ,{k}KS ,T11
_S| |C: SessionID,{Response1}k,SigS(Response1),T12
_C| |S: SessionID,{Request2}k, SigC(Request2), T21
_S| |C:SessionID,{Response2}k, SigS(Response2),T22

B. Algorithm

First of all, steganography program uses a form of basic LSB alteration procedure. This form of steganography is believed to be weak and easily breakable, but selected to use it for two main reasons:

1) It is not too difficult so that numerous experiments related to it can be done.

2) To strengthen this weak security technique by adding more randomness.

The method itself does still encode data by flipping LSBs, but by adding randomness to images and how the data is actually encoded, we can make the cover images and stego-images seem more similar. There are several types of "randomness" that we can employ in our application.
First of all, the exact pixels altered to encode each letter are semi-randomly filled. When embedding a message within an image, the very first step in this randomness is to count the number of letters in the message and divide the total number of rows of pixels in the image by that number. For example, if the image is 2000 pixels tall and we want to embed a message 150 letters in length, divide 2000 by 150 to get 10 rows per letter. This is how the number of rows of pixels that could be associated with each letter[5].
Next, take the number of pixels the image is wide and divide by the number of letters in the alphabet (26). So if the image is 624 pixels wide, divide by 26 to get 24 columns per letter of the alphabet.
By grouping these sets of (10 in the example) rows and (24 in the example) columns together, we get a grid of sets of pixels within the image. Each of these sub grids corresponds to an encoding of a different letter of the message. The groups of row correspond to the index of the letter in the message (Eg1: The very first letter, second letter, etc.) and the groups of columns[3] correspond to the format of actual letter encoded. So, for example, to encode the third letter[3] as the letter "b" look at the third set of rows (rows 20-23 in our example) and the second set of columns (columns 26-51 in our example). Now to perform actually encode that the third letter in the message is "b", take a random bit from this sub grid (from rows 20-20 and columns 26-51 in our example) and flip it. See table-2 below for a visual representation of how sub grids map to different letters in the message for part of the image[5].

Table -2: Image Grouping by Rows & Columns Grid

IMAGE GRID	Cols 0-25	Cols 26-51	Cols 52-77	Cols 78-103	Cols 104-129
Row 0-9	Letter 1=a	Letter 1=b	Letter 1=c	Letter 1=d	Letter 1=e
Row 10-19	Letter 2=a	Letter 2=b	Letter 2=c	Letter 2=d	Letter 2=e
Row 20-29	Letter 3=a	Letter 3=b	Letter 3=c	Letter 3=d	Letter 3=e
Row 30-39	Letter 4=a	Letter 4=b	Letter 4=c	Letter 4=d	Letter 4=e

ISSN: 2278 – 7798

Row 40-49	Letter 5=a	Letter 5=b	Letter 5=c	Letter 5=d	Letter 5=e
Row 50-59	Letter 6=a	Letter 6=b	Letter 6=c	Letter 6=d	Letter 6=e

1) Now someone decoding the message by comparing the altered image against the original can easily determine the message even with this randomness since the letters appear in the message in the same order we find altered bits going down the rows and the decoder knows which groups of columns correspond to each letter. However, by adding the feature of randomness, it may be more highly secure task for someone who lacks the original image to detect something is wrong with the bit patterns of the modified image. Additionally, by separating the letters out across all the rows of the image, we reduce the chances of steganalysis detecting something wrong with a small portion of the image. This technique does have a rather low data capacity (at most the number of rows of the image), but this also increases its strength against[3] stegano-analysis since there are fewer alterations to be detected. In addition to the randomness built directly embedding data into my program offers options for adding random noise to the image before embedding data into it. The noise frequency is added by selecting random pixels and flipping their LSB before the secret data is even embedded (and then using this altered cover as the new cover). The method here is to create a cover with a bit pattern that is random in such a way that it is then difficult to tell covers apart from stego-objects, that is, covers that might seem themselves[3] like stegano-objects and thus confuse stegano-analysis tactics.
2) Taking this added randomness even one step further & implemented the ability for my program to generate its own random images to be used for covers. A fully random image presents no patterns in its least significant bits[3], and so it is (nearly) impossible to reliably detect alterations in bit patterns caused by my steganography method since there are no patterns to begin with. The problem faced with this is that the complete lack of a pattern in the cover image may itself arouse suspicion of an outside intruder, as would the fact that communicators are sending images to one another that are composed of entirely random pixels. As a counter to this, we modified our random image generator to generate images with pixels that are similar to those surrounding them. And so the result is a fairly random image that still displays visual patterns and thus could potentially pass as a form of modern technique, which is less suspicious than a fully random image[5].
3) T The goal of all of this is to automatically generate both covers and stegano-objects that are statistically very similar. Some specific results are presented in the associated paper ("Unseen"). We found overall that it is very possible to generate cover/stegano pairs that are almost identical statistically, thus making it nearly impossible for stegano-analysis to identify a single image as either cover or stegano. Not every random image will have same kind of this property, but due to the nature of randomness, maximum possible large number are expected to.
4) On further work on this application includes refining the random image generator, exploring different ways to add randomness to images and to the embedding of messages within images, and altering images in other (perhaps larger) ways, such as flipping or shifting pixels.

VI. Face Recognition Using Eigen Faces:

This picture represents the set of images used to create our Eigen space for face recognition. The overall problem is to be able to accurately recognize a person's identity and take some action based on the outcome of the recognition process. To recognize a person's identity is important mainly for security reason, but it could also be used in future to obtain quick access to medical report, criminal case or any type of records. Solving such kind of problem is important because it could allow personnel to take preventive action, provide better service- in the case of a Hospitality[6] or allow a person access to a secure area[6].

Fig -4: Eigen Vector Procedure[7].

A. Eigen Face Algorithm-Construction Procedure:

This section gives step-by-step instructions along with photos and formulas on how to recognize faces and implemented into Matlab. All the necessary files to complete this would be provided[7].

Steps:-

1) The first step is to obtain a set S with M face images. In our example M = 25 as shown at the beginning of the tutorial. Each image is transformed into a vector of size N and placed into the set.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \ldots\ldots\ldots, \Gamma_M\}$$

2) After you have obtained your set, you will obtain the mean image Ψ[7].

ISSN: 2278 – 7798

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$$

3) Then you will find the difference Φ between the input image and the mean image[7].

$$\Phi_i = \Gamma_i - \Psi$$

4) Next we seek a set of M orthonormal vectors, un, which best describes the distribution[8][9] of the data. The k^{th} vector, uk, is chosen such that[8]

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}\left(u_k^T\Phi_n\right)^2$$

$$u_l^T u_k = \delta_{lk} = \begin{cases}1 & \text{if } l = k\\ 0 & \text{otherwise}\end{cases}$$

Note: uk and λk are the eigenvectors and eigenvalues of the covariance matrix C[10].

5) We obtain the covariance matrix C in the following manner

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T$$

$$= AA^T$$

$$A = \left\{\Phi_1, \Phi_2, \Phi_3, \ldots\ldots, \Phi_n\right\}$$

6) A^T

$$L_{mn} = \Phi_m^T\Phi_n$$

7) Once we have found the eigenvectors, vl, ul

$$u_l = \sum_{k=1}^{M} v_{lk}\Phi_k \qquad l = 1, \ldots\ldots, M$$

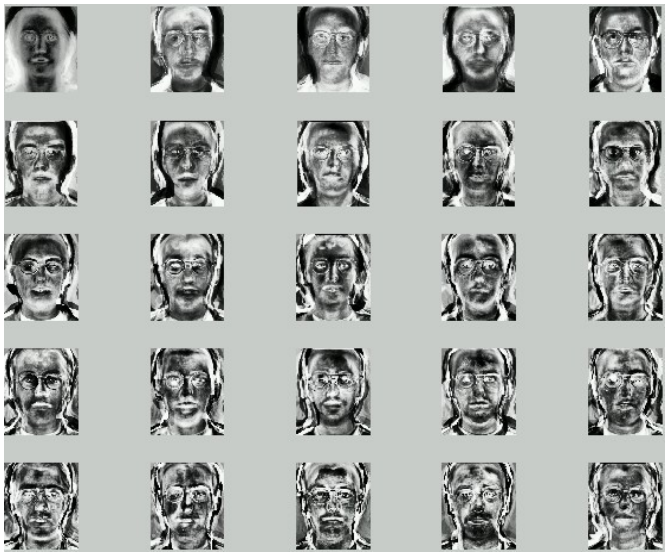

Figure 5: These are the Eigen faces of our set of original images

B. Eigen Face Algorithm-Recognition Procedure

1) A new face is transformed into its eigenface components. Firstly we will be comparing our input image with our mean image and multiply their difference with each eigenvector of the L matrix. And so each value would represent a weight and would be saved on a vector Ω.

$$\omega_k = u_k^T\left(\Gamma - \Psi\right)$$

$$\Omega^T = \left[\omega_1, \omega_2, \ldots\ldots, \omega_M\right]$$

2) We now determine which face class provides the best description for the input image. This is done by minimizing the Euclidean distance[7].

$$\varepsilon_k = \left\|\Omega - \Omega_k\right\|^2$$

3) The input face is consider to belong to a class if εk is bellow an established threshold θε. Then the face image is considered to be a known face[11]. If the difference is above the given threshold, but bellow a second threshold, the image can be determined as a unknown face[7]. If the input image is above these two thresholds, the image is determined NOT to be a face.

If the image is found to be an unknown face, you could decide whether or not you want to add the image to your training set for future recognitions. You would have to repeat steps 1 through 7 to incorporate this new face image[7].

VII. CONCLUSIONS

As The Existing system OTP(One time password) is not secure enough to provide secure authentication to e-banking in mobile system so, Application we proposed more robust to user authentication for e-banking by the means of random bit LSB-Stegnography technique and face recognition mechanism. This application provides Two-level Authentication for user which will make the transaction with

ISSN: 2278 – 7798

bank highly secure. At first stagestegnography does encryption of login id & password and hides behind image by an technique of an Random Bit LSB & then at second stage the user face will be matched with database face to authenticate valid end user.

ACKNOWLEDGMENT

We take this opportunity to thank our project guide, and Head of the Department Prof.S.R.Todmal & Prof. D. P. Gadekar And our special project guide Prof. Sonal Fatangare for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the Department of Information Technology of JSPM Imperial College of engineering, Wagholi, Pune for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

REFERENCES

[1] M-Banking Using Steganography and Cued Click Points,July-13-2013.
[2] Steganographic Authentications in conjunction with Face and Voice Recognition for Mobile Systems. Feb, 2011.
[3] Unseen: An Overview of Steganography and Presentation of Associated Java Application C-Hide, April-20-2009.
[4] M. Mattila, H. Karjaluoto, and T. Pento, 2002. Internet banking adoption factors in Finland.
[5] Random_Bit_Steganography Unseen: An Overview of Steganography and Presentation of Associated Java Application C-Hide. April 20,2009.
[6] Analysis of Face Recognition in MATLAB, june25-2005.
[7] www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm,October-28-2003.
[8] Face Recognition System Based on PCAand Feedforward Neural Networks,January-1-2005.
[9] A Comprehensive Methodological Analysis of Eigen Face Recognition and Reconstruction,Feb-20-2012.
[10] Face Recognition in Wireless Communication Using MOCHA Architecture on Jelastic Cloud, ISSN: 2231-4946,Nov-15-2013.
[11] Stand alone face recognition system using principle component analysis ISSN(ONLINE): 2279-0055, Dec 15,2013.
[12] Steganography in digital media by Jessica Fridrich.
[13] M. Zviran and W.J. Haga, 1990. Cognitive passwords: the key to easy access control. Computers and Security 723–736.

Mr. VENKATESH RAJAM. ASAMPELLI, has pursuing his B.E Degree in Information Technology from University of Pune 2014. His Research interest in Image Processing & Artificial Intelligence. Java & Android Programmer.

Mr. PATEL RAVI VASUDEVBHAI, has pursuing his B.E Degree in Information Technology from University of Pune 2014. His Research interest in Image Web Processing & Artificial Intelligence, Ruby on Rails & iOS Advance. Expert in UX Engg.

Mr. BAJIRAO SHARADRAO. SHINDE, has pursuing his B.E Degree in Information Technology from University of Pune 2014. His Research interest in Image Processing & Artificial Intelligence. Won first prize in Project Presentation Competition. Java & Android Programmer. Email

Mr. TUSHAR ARJUN RAUT, has pursuing his B.E Degree in Information Technology from University of Pune 2014. His Research interest in Image Processing & Artificial Intelligence. Java & Android Programmer.

ISSN: 2278 – 7798