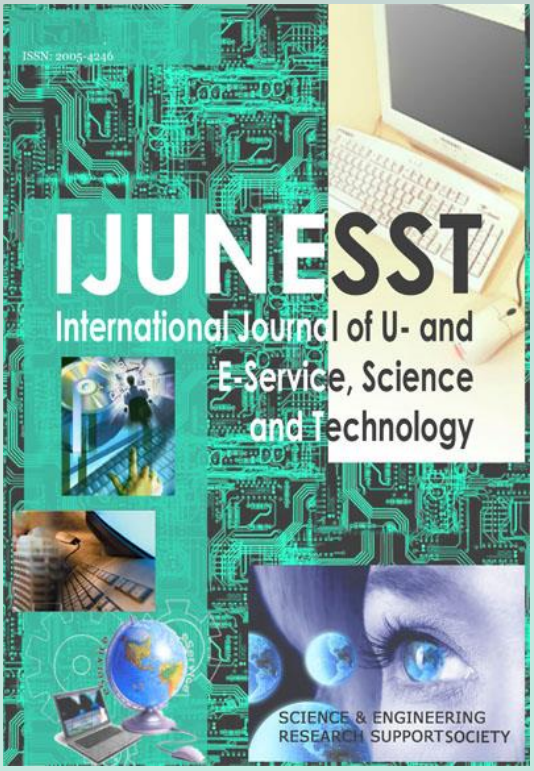
Securing the Smart Card Authentications Process by Embedment Random Number of Data Bits into Each Pixel

A. H. M. Kamal



ISSN 2005-4246
Volume 10, Number 7, 2017

International Journal of u- and e- Service, Science and Technology
Vol. 10, No. 7 (2017), pp.43-54
http://dx.doi.org/10.14257/ijunesst.2017.10.7.05

SERSC
Science & Engineering Research Support soCiety

International Journal of u- and e- Service, Science and Technology
Vol.10, No.7 (2017), pp.43-54
http://dx.doi.org/10.14257/ijunesst.2017.10.7.05

Securing the Smart Card Authentications Process by Embedment Random Number of Data Bits into Each Pixel

A. H. M. Kamal*

Dept. of Computer Science and Engineering
Jatiya Kabi Kazi Nazrul Islam University,
kamal@jkkniu.edu.bd,

Abstract

Smartcards do not tolerate any security threats because these are usually used to manage the financial transactions and for the official or personal security purposes. Authenticating the users in the smartcard based applications through data embedment is a promising research area. Because of the simplicity of used computing devices, i.e. processors, less memory, of the terminals, least significant bit (LSB) replacement schemes are used to hide the secret information in a photograph inside into the card. Nevertheless, there are statistical tests like Chi-Square test, comparison of histograms of pixel values of original and modified images, regular and singularity test of the pixel values, which can guess the possibility of modifications in the LSBs of image data. In this research, a new smartcard authentication scheme is proposed by embedding random number of bits into each pixel. The number of replaced bits in the i^{th} pixel may not be the same of $(i+1)^{th}$ or other pixels. Security of the scheme is further enhanced by relating the starting position of the data embedment process with the user password as well as allowing that starting position in the image to be dissimilar for different user. The scheme is tested experimentally and it is found that it operates successfully.

Keywords*: LSB, steganography, smartcard, authentication, embedding data, terminal, chip*

1. Introduction

Data communication through public communication channel is always challenging. Even if the communication is within two devices which are connected together by a one meter wire is also somewhere intricate in ensuring privacy and authenticity. Unwanted threats can make hazards to the data communication system and the data security can be broken at any time. Therefore, to prevent the message from being hacked a good number of strategies are adopted in the communication world. The most commonly used strategies are encryption, watermarking and steganography. The encryption is a mechanism to convert a secret data to other values using a key by applying a one way functions, Hibner *et al.* in 2011 [1]. The encryption process destroys the meaningful information in the data; however, the existence of message within the encrypted values becomes sensible in many instances. From periodical tries by repeatedly guessing a portion of the message or the key, the intruder may retrieve the original message. Any high compution performing computer, being an intruder, can easily break the encryption algorithms if the key length is smaller. Another good mechanism of providing the data integrity is watermarking where the data is concealed into a media, *e.g.* image, in a way that one cannot remove it from the media, Chen, and Tsai in 2011 [2]. The objective of the watermarking system is

*

The article is an outcome of a research based project which is funded by University Grant Commission of Bangladesh.

ISSN: 2005-4246 IJUNESST

to ensure the data integrity. In many instances, though the embedded data is not editable, the watermarked data becomes visible. Hence, it does not provide the actual security and privacy of the secret data. On the other hand, if the data is hid inside into a media, this will overlook the intension of the intruders. This media will carry the message to the destination. This message hiding method is known as steganography, in Liao, Xin *et al.* 2011 [3].

The process of steganography hides the secret message bits into a media, known as cover media. The media with the concealed information is phrased as stego media. Numerous medias are used for steganographic purpose. Very widely used medias are text (Yee, Lip *et al.* in 2012 [5]), audio (Sagar and Amberker in 2013 [6]), video (Jorge, *et al.* in 2012 [7]) and image (Lee and Chen in 2010 [8], Hong *et al.* in 2012 [9]). As a media, the image is the most popular one in the field of steganography Lin C-C in 2011 [4], because image exhibits a set of particular characteristics, *e.g.*, more redundant information inside into them, frequent transmission of images by the users and applications over the Internet, its flexible size for communicating over a low bandwidth network. The following Figure 1 depicts the total embedding process. The steganographic process takes an image as an input, also known as cover image, and conceal message bits into the pixel values of that image by the embedding rules. This modified image, contained hidden information, is called stego image. The quantity of bits that are implanted into an image is called payloads. The implantation rate is measured by embedded bits per pixel (bpp) which is called embedding capacity. Due to the data embedment, the cover values will be modified. This amount of modification is termed as image distortion.

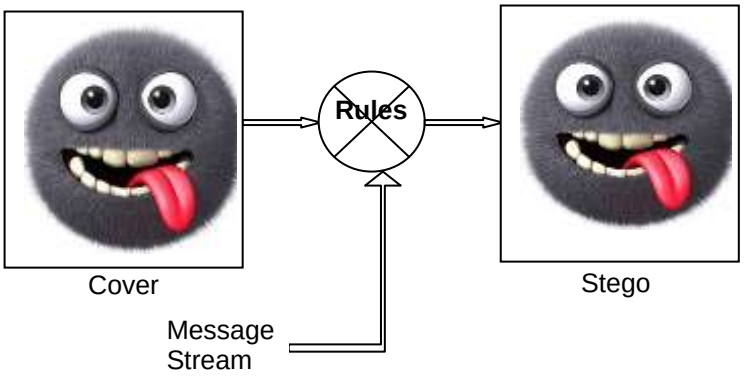


Figure 1. Image Steganography

Uses of image steganography is expanding from message communication over Internet to medical or forensic imaging applications, Ulutas, Mustafa *et al.* in 2011 [10], smart card authentication process, Brindha, S., and Ila Vennila in 2011 [11], Kamal, A. H. M., and M. Mahfuzul Islam in 2014 [12]. There are many methods which are used in the field of image steganography. The earliest methods used least significant bit (LSB) replacement, Fridrich *et al.* (2002) [13]. The LSB replacement methods suffer from the detection of embedded information by many statistical test, *e.g.*, Chi-square test, Liu *et al.* in 2011 [14]. Addition and subtraction based schemes are also observed in the literature, Tian *et al.* in 2003, [15]. These are detectable by other steganalysis method like vertical and horizontal histogram difference, Zhao *et al.* in 2009, [16]. The transformed domains are used as an embedding space to hide data. In this case, the transformation is performed first. Applied common transformation methods are Fourier transform (Wang, Xiang-yang, *et al.* in 2013 [17]), discrete cosine transform (Sagar and Amberker [6]), wavelets transform (Kamstra *et al.* (2005) [18]). A reverse transformation is performed just after

the data embedment task. Prediction errors based embedding schemes (Wien Hong *et al.* in 2010 [19], Kamal and Islam in 2015 [24] and Kamal and Islam in 2016 [25-27]) present higher embedding capacity and less image distortions. In this embedding process, a prediction is performed first. Data bits are embedded into the computed prediction errors. The embedding capacity depends on the prediction accuracy.

In all the above stated embedding processes, the cover image is modified for the data implantation. Due to this modification, the stego image differs from the cover image. The quality of the stego image is measured by the amount of the distortions. The visual quality of stego image is computed by the peak signal to noise ratio (PSNR) or the mean square error (MSE). In authenticating smart card, the visual quality of the stego image is not a concern issue at all [12] because the image is stored in the chip of the smartcard and it is never communicated over the public network. Nevertheless, these images are communicated through an wire of about one meter from the chip to the terminal. The intruders will try to hack the embedded secrets during this short communication or through other unauthorized card. Hence, the steganographic process, applied for the smartcard authentication, demands for stronger security of the implanted data rather than image quality. The LSB replacement based embedding schemes are famous for smartcard based applications [11], [12] for the simplicity of the algorithms as well as for the lack of computing devices with high performance. Brindha *et al.*[11] employed LSB replacement technique to embed secret information in the photograph of the card owner. There are lots of steganalysis schemes in the literatures to detect the possibility of LSB replacement based implanted data. Again, PANG *et al.* in 2012 [20] applied the image steganographic process for ensuring the privacy of the electronic voting. A similar application was also presented in the literature by Rura *et al.* in 2011 [21]. These applications differ a bit with the concept of smartcard applications. Kamal and Islam in 2014 [12] employed a set of functions to select the pixels from the different places in the image for applying the LSB replacement process. This scheme uses a function from a predefined function's list and the initial values of the parameters from another list. These two values are used to register and further authenticate a user. The scheme also suffers from a security threats as these two tables can be hacked. Besides, both the schemes, [11], [12] will be failed to show enough resistance against LSB replacement steganalysis as these schemes replace a single LSB only. Indeed, the uses of traditional LSB replacement process in hiding data cannot ensure the security of the implanted bits.

The authors in this article present a steganographic based smartcard authentication process taking the matters of ensuring both the improved security and the simplicity of the algorithm into consideration so that the algorithm is implementable in the commonly used chips and terminals. Three important objectives of this research work is to increase the security of the LSB replacement based method, enhance the embedding capacity and implement a smartcard authentication process. The scheme replaces random number of LSBs in each pixel rather than a single LSB to improve the security of the implanted data the embedding capacity. This process divides the image into two parts. A pseudo random number generator is used to generate a random number m within the range of [1, k]. The scheme replaces m number of LSBs of the i-th pixel of lower image part by m bits of secret information. The k LSBs of i-th pixel of the upper part is modified by the value of m (by converting m into k-bits binary). At the receiver end, the decoder extracts the value of m from the k LSBs of the i-th pixel of upper part and then extract m bits of information from the i-th pixel of lower part. The maximum number of replaceable bits by the random function, *i.e.*, the value of k, and the starting position of data embedment are made a part of the user password. Thus, in this research, the security of the smart card authentication is improved. Finally, the stego image is installed in the chip memory of the card to use it in the terminals for the authentication purpose.

This article is organized into several sections. It follows more four sections. Section two illustrates the related works and the next of it is to explain the proposed algorithm.

Experimental results are presented in the section four while section five concludes the article.

2. Related Work

Smart card based applications demand for stronger data security rather than higher payloads and visual quality of stego image. At smart card very few information like password, user ID, biometric parameters, are embedded into a photograph. These are very secret information. The ability of retrieving the secret information by any unauthorized person may cause the card owner to loss pecuniary or personal dignity. Hence, securing the authentication process in the smartcard based applications is demanded. Authenticating smartcard using image steganography is a newfangled idea. To the best of our knowledge, such an authentication scheme is first time proposed by the authors (Kamal, A H M and Islam M M [12]) of this article. Two most important parts of the objectives of this proposed research work is to increase the security of LSB replacement schemes in smartcard based applications and boosting up its embedding capacity. The steganographic based smartcard authentication field contains a very limited number of articles in the literatures. Therefore, as a related work, only the scheme of Brindha and Vennila. [11] and Kamal and Islam [12] are presented.

Brindha and Vennila used a pseudo random number generator (PRNG) to select the pixels for bit implantation from scattered positions in the image. An LSB of these pixels are replaced by the message bits. The number of embedded bit in each selected pixel is one. Many statistical analysis based techniques can detect the perturbed image. A periodical try of rearranging the LSBs of perturbed pixels can lead a steganalyzer to the success of breaking the security. Besides, while using inverse PRNG, if it is sensed by someone, the sensor will be able to reveal the exact secret information from the stego image. For that reason, Kamal and Islam has proposed multi-function based LSB replacement scheme where each of the functions is used to select the pixel position in the image grid. Examples of such functions are rectangle, triangle, polygon, circle, oval, parabola, line, sinc, *etc.* As a sample, four of those are depicted in the following Figure 2.

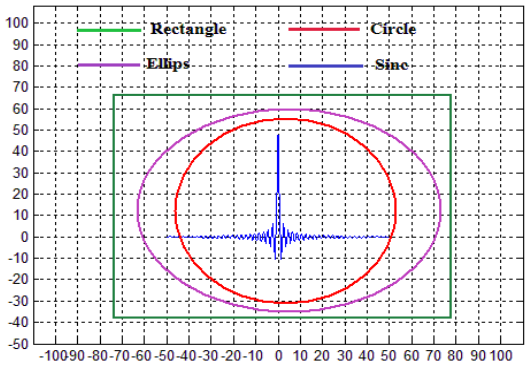


Figure 2. Some Sample Functions

These functions, *f1, f2, ..., fn,* are permuted and the permuted values are stored into a table name 'application sequence of functions (ASF)'. Each row presents a unique applying sequence of functions. That is, if row one is selected then functions will be applied in the embedding algorithm in the sequence *f1* => *f2*=> … =>*fn*, where '=>' stands for 'next execution' and that sequence will never be the same for the selection of other rows. Again, the parameters used by all the functions are arranged in a row first. Then, *n* numbers of rows are generated assigning different values to each of the parameters. These are stored into a table named 'parameters with different values (PwDV)'. Thus, each row will provide a unique pattern of assigned values to the

parameters. Each user is assigned two rows, *RN1* and *RN2*, randomly selecting a one from each of the tables, *i.e.*, *RN1*ϵ ASF and *RN2*ϵ PwDV. These row numbers are then utilized in the formation of the password. The password contains three parts – (user_ID) || *RN1* || *RN2* where || stands for string concatenations. The stego image and these two tables are then installed into the chip of smartcard. During the verification phase, the terminal collects the password from the user and separates *RN1* and *RN2* from this password. The terminal, then, select *RN1* indexed row from ASF and *RN2* index row from PwDV table. The information stated in these two rows are applied in extracting embedded information from the image. Nowadays statistical analysis can detect the tempered pixels. Thus, by collecting the LSBs of the tempered pixels, the security can be broken.

3. Proposed Work

To enhance the security, in this article m-Random Bits (m-RB) LSB replacement algorithm is proposed where the number of embedded bits, say *m*, into each individual pixel is defined by a random number generator. That random number *m* is less than *k*. The *k* will be made a part of the user password.

The proposed scheme undergoes a set of activities. The owner of the smartcard is registered first with the provider of the card. The provider embeds the personal secret information and the authentication data into an image. This is called embedding phase. Relating the embedded information, a unique password is generated by the card provider. That password along with the card is supplied to a client. This task is known as registration phases. Again, when the card holder pushes the card into a terminal, the terminal extracts the embedded information from the image. This process is called as secret extraction phase. The extracted secrets like biometrics are compared with the user provided information. That process is termed as authentication phase of the scheme. If the test is passed, the user is allowed to operate further in the terminals. These four phases are described at the following.

3.1. m-Random Bits LSB Replacement Embedding

Firstly, the image is partitioned into two parts. The second portion starts from *R***S*-th pixels of image where both *R* and *S* are integer number and each of these are consisted of three digits. To make *R* and *S* unpredictable, these are made longer. To make these variables private, these are pertained to the password by engendering the password as R||S||T where || stands for concatenation and *T* is the last part of the password; other than the first six digits. To strengthen the password, this is further concatenated to Q, *i.e.*, new password is Q||R||S||T, where Q is known to the terminal as well as to the vendors. Nevertheless, for the convenience of explanation of the scheme, the password is consider as R||S||T in the following. Secondly, a random number *m* is generated within the range [1 *k*]. The *k* is a value which is less than 9. The value of *m* is further converted to binary *b* of length *k*. If the number of bits in *b* is less than *k*, enough '0's are pushed to the left of *b* to make it *k*-bits length. Finally, that *b* is embedded to the first part of the image, starting from R^{th} pixel, by LSB replacement algorithm. Thereafter, an *m*-bits of message chunk is embedded to the second part of the image by the same LSB replacement algorithm, starting from $R*S^{th}$ pixel. The process is repeated and embeddings are done to the subsequence pixels in both the image parts. The process is outline in the following pseudo code.

i. Set pointer *Ptr1* to R^{th} pixel and *Ptr2* to $R*S^{th}$ pixel of cover image;
ii. Scan pixel at *Ptr1*;
iii. *m*=rand(*k*);
iv. Convert *m* to binary digits *b* of length *k*. Replace *k*-bits LSBs of *Ptr1* scanned pixel by the bits in *b*.

v. Scan another pixel by *Ptr2* and replace *m*-bits LSBs of the pixel by *m*-bits message bits.
vi. Advance both pointers *Ptr1* and *Ptr2* by 1 to point to the next pixel.
vii. If still message bits are in hand to be embedded go to step ii

3.2. Registration Process

In the registration process a password is generated first after accepting a demand from a new client. It is already said that password is generated by R||S||T. Among the three major parts of the password, *T* is selected such a way so that T=k||L where k ε {1, 2, … , 8}. That is, the first symbol of *T* is a digit which is *k*. Thus, at the utmost 8 LSBs are allowed to be replaced. Since, *k* is the first digit of *T* and seventh digit of the user password, the value of *k* will be sensed by the terminal easily from the user provided password at all login attempts. This password is generated at registration phase by the vendor like Citi Bank, World Bank, hospital, clinic, government or non-government office, etc after accepting a demand for that from a client. The value of *R*, *S* and *k* are variable. Therefore, these three can be variants for different users. The *L* is the only distinct one that can differentiate the card owners uniquely. The length of *L* can also vary based on the number of users of the respective vendors. The user name, user ID, biometric parameters like minutiae for finger print and other secret data are embedded to the photograph of the card owner according to the value of *R*, *S* and *k* and by the rules stated in the Section 3.1. The photograph is then saved to the chip memory of the card. The card and the password are then delivered to the card owner.

3.3. Extraction Process

To authenticate a card holder secret information those were embedded into the card during the registration process are extracted. After inserting the card into terminal, user provides the password to be verified. Terminal separates *R*, *S* and *k* from the first seven symbols of the password. This, then, collects the stego image from the card. Two pointers *Ptr1* and *Ptr2* are set to the R^{th} pixel, and $R*S^{th}$ pixel of the image. The first pointer *Ptr1* provides the information of how many bits are embedded into a pixel pointed by *Ptr2*. The algorithm works as follows.

i. Set pointer *Ptr1* to R^{th} pixel and *Ptr2* to $R*S^{th}$ pixel of stego image ;
ii. Scan pixel at *Ptr1*. Separate *k*-bits of LSBs from it and convert it to a decimal value, *m*.
iii. Scan pixel at *Ptr2*. Extract *m*-bits LSBs from *Ptr2* scanned pixel.
iv. Go to step ii until the full secrets are not extracted.

3.4. Authentication Process

After accepting the card, terminal requests for the password and direct the user to provide it. It separates *R*, *S*, *k* and *L* from the received password. The terminal again collects the stego image from the chip of the card. It then extracts the user id, biometrics from the stego image. Thereafter compare these ID with *L* value and extracted biometrics with user provided biometrics. If these are matched, it allows the user to perform further operations like money transactions or to receive various services like medical service, person identification, *etc*. The authentication process is highlighted in the following Figure 3.

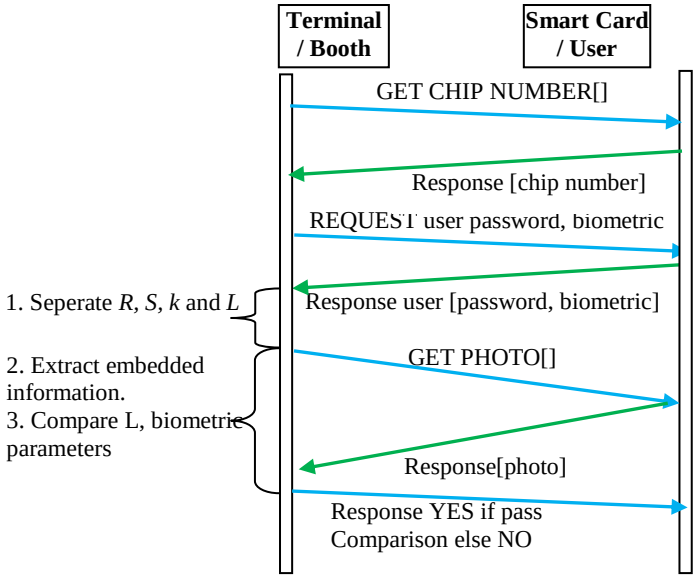


Figure 3. Card Verification

4. Result Analysis

To setup the experiment, about 50 photographs and 200 images were collected first by our camera and from various research sites, online newspapers and Wikimedia. The proposed embedding, extraction and authentication phases are tested on these images. It is experimentally observed that the success rate of these three modules of the proposed scheme is 100%. As the scheme operates successfully, the only affair to be checked here is the issue of stego security, *i.e*. retrieval possibility of the stego contents by unwanted persons. Some of the methodologies employing statistical analysis are observed in the literatures to detect where the LSB replacement is applied or not. However, none of the schemes can detect the numbers of bits that are replaced even if the length of replaced bits of all the pixels is a fixed value. Contrarily, the proposed scheme replaces random number of bits in each of the pixels. Hence, the number of bits that are replaced in i^{th} pixel may vary at $(i+1)^{th}$ and in the other pixels. Thus, the scheme has made the mission of extracting secret data by unwanted person harder. Nevertheless, two schemes of detecting stego contents are analyzed here to observe the resistance against statistical attacks. These are Chi-Square test [14] and horizontal-vertical difference histogram analysis [16].

4.1. Resistance against Chi-Square Test

The proposed scheme performs well to overcome the Chi Square test, Liu et al, 2011 [14]. Chi-Square test is used to check any perturbations to the LSBs of the pixels of an image. The Chi-Square test for $m \times n$ image was done by the following relation.

$$X^2 = \sum_{i=0}^{255} \frac{O_i - E_i}{E_i} \quad (1)$$

The O_i and E_i in equation (1) are the frequencies of i^{th} bin of histogram of cover and stego image respectively. In the equation these are considered as expected and observed values respectively. Therefore, the degree of freedom (*DF*) is $(0 - 255) + 1$, *i.e*. 256, or close to it. As DF>30, according to the rules of Chi-Square statistics, normal distribution is measured by the equation (2) to find the Chi-Square statistics.

$$\sqrt{X^2} - \sqrt{2DF - 1} \quad (2)$$

MATLAB tool *chi2inv* is used to measure the critical value $X_{\propto}^2$ setting the probability at $\propto = 0.005$. The experimental result is shown in the Figure 4. The results are found embedding a one bit at each attempted pixel. The figure demonstrates that it passes the null hypothesis in all the images as the Chi values are smaller than critical values.

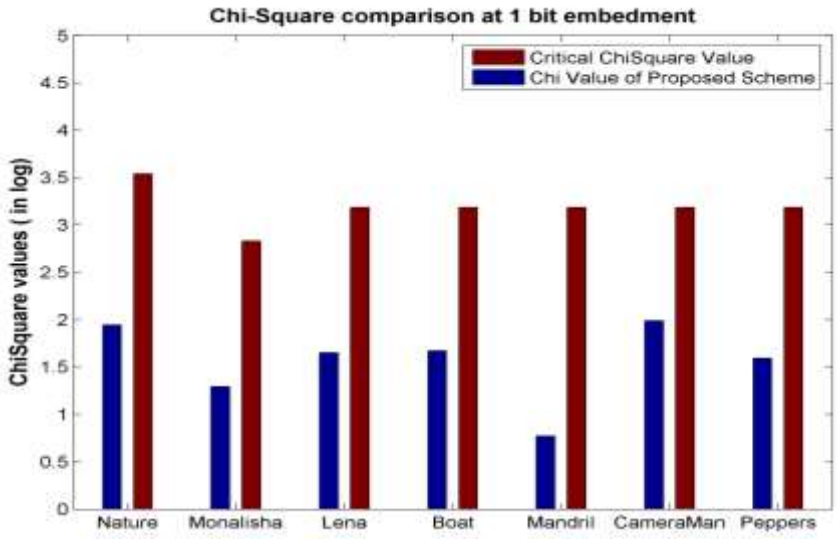


Figure 4. Result of Chi Square Statistics

Another test is performed setting *k* value to 4. There 1 to 4 LSBs are allowed to be replaced at the embedding time. The results of 50 images are plotted in the Figure 5. About in 84% images it passes the null hypothesis. Thus, it delineates enough resistance against attacks by Chi-Square statistics.

4.2. Testing by the Histogram of Vertical and Horizontal Differences

Another different test employing vertical difference histogram (VDH) and horizontal difference histogram (HDH), Zhao *et al.*, 2009, [16], Hong *et al.* 2012[22], is performed to check any changes in the stego image regarding to original image. The differences of adjacent pixels along vertical direction and again at horizontal direction are measured. These vertical and horizontal differences are applied to form VDH and HDH respectively. In the natural images it can be expected that both the VDH and the HDH will be, almost, similar and will remain very close to each other. It will differ in a tempered image. The scenario can be observed in Figure 6(a) and Figure (b) for *k*=1 and *k*=4 respectively.

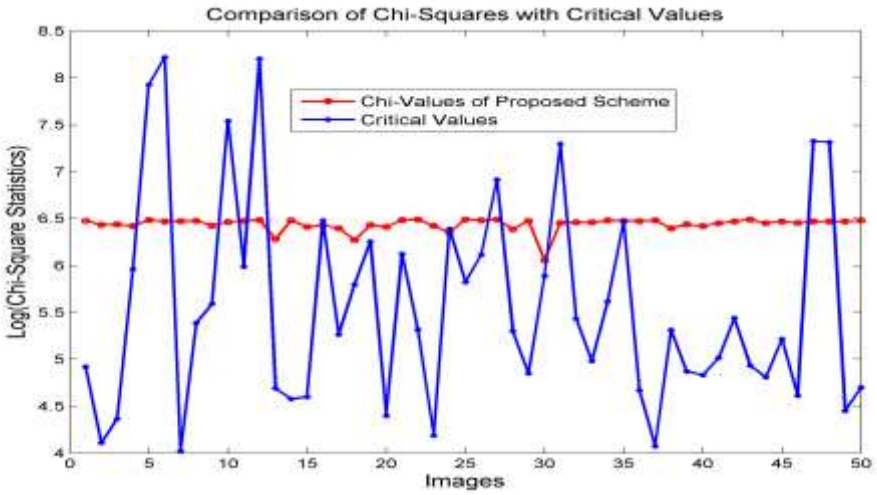
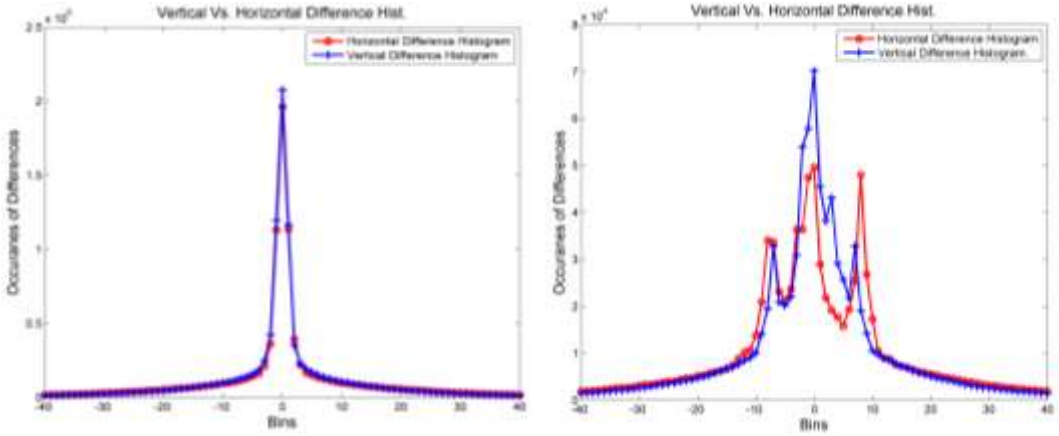


Figure 5. Chi-Statistics for k=4



(a) Comparison for Single Bit Embedment (b) Comparison for 4-bit Embedment

Figure 6. Comparison of Vertical Difference Histogram and Horizontal Difference Histogram

The deviations among the two histograms can also be verified by a numerical value too. Because the difference of all the i^{th} bins of those two histograms equates very smaller value for all the regular images and larger value for the tempered images. That numerical value can be measured by the following equation (3).

$$D = (\sum_{i=-T}^{T} (\breve{H}_h(i) - \breve{H}_v(i)))^{1/2} \quad (3)$$

Where $\breve{H}_h$ and $\breve{H}_v$ are HDH and VDH respectively and T is a threshold. Thus, the equation (3) works with only the bins within the range [-T T]. Most of the frequencies stay to a certain number of bins which are close to '0'. Therefore, application of such a threshold T will effectively minimize the computational complexity not affecting the desired results. In our experiment, 150 images were tested setting T=40. The highest value of D was found 126 and 419 and lowest to 10 and 80 for k value of 1 and 4 respectively. Though 419 is a large value, however, extraction of secret message is still challenging as the random bits are embedded to each pixel. Thus, the proposed scheme is able to challenge all the steganalyzer by resisting them from breaking the security.

5. Conclusion

The application of image steganography in smart card will improve the security of authentication phase. This article presents a very innovative scheme to embed *m* random bits in each pixel. Doing so, it has increased the robustness of the scheme as well as the embedding capacity. The functional validity is tested on 250 different images. In all the images it operates successfully. It also passes the hypothetical tests on different methodologies for all k<4. Besides, no scheme is found in the literature that can detect the length of replaced bits in each pixel. Moreover, the proposed scheme embeds random number of bits in each pixel. Therefore, the presented scheme will enhance the security of embedded data as well as the security of smartcard authentications.

In our future work we hope to authenticate users of a smartcard through wireless communication, Kamal, A. H. M. in 2013 [23] by the uses of image steganography. In the wireless it is a bit challenging than in the terminal because the stego image can be sensed and analyzed by a malicious or intruders steganalyzer during the communication of stego image for authentication purpose.

Acknowledgments

The author A H M Kamal (AHMK) is funded by the University Grand Commission of Bangladesh through their small project. Therefore, the authors like to acknowledge the University Grand Commission of Bangladesh.

Author Contributions

The first author, AHMK, is a PhD student of the department of Computer Science and Engineering of the Bangladesh University of Engineering and Technology. He is working under the supervision of second author, Mohammad Mahfuzul Islam (MMI). Hence, the whole work is supervised and guided by MMI. Mr. MMI has been consulted on all the way to the progress of the research work by the author AHMK. Mr. AHMK has completed the experiments and made the draft of the manuscript. Mr. MMI has revised the manuscript and given the final approval to submit it to that journal.

Conflicts of Interest

The authors do not have any economical interest from that article. The first author is a PhD student and working under the supervision of the second author. To meet the requirement for achieving the PhD degree, the first author has to publish his research works on ranked journals which are published by well recognized publishers. Therefore, the authors have chosen this journal to publish that work. Both the authors are aware of that submission.

References

[1] K. A. Hibner, N. Koblitz, and A. Menezes, “Elliptic curve cryptography: The serpentine course of a paradigm shift”, Journal of Number theory, vol. 131, no. 5, **(2011)**, pp. 781-814.
[2] C. C. Chang and Y. H. Tsai, “Adaptive reversible image watermarking scheme”, Journal of Systems and Software, vol. 84, no. 3, **(2011)**, pp. 428-434.
[3] L. Xin, Q. Y. Wen and J. Zhang, “A steganographic method for digital images with four-pixel differencing and modified LSB substitution”, Journal of Visual Communication and Image Representation, vol. 22, no. 1, **(2011)**, pp. 1-8.
[4] L. C. Chiuan, “An information hiding scheme with minimal image distortion”, Computer Standards & Interfaces, vol. 33, no. 5, **(2011)**, pp. 477-484.
[5] Y. Lip, K. S. Wong and K. O. Chee, “UniSpaCh: A text-based data hiding method using Unicode space characters”, Journal of Systems and Software, vol. 85, no. 5, **(2012)**, pp. 1075-1082.
[6] G. Sagar, and B. B. Amberker, “DCT based reversible data embedding for MPEG-4 video using HVS characteristics”, Journal of Information Security and Applications, vol. 18, no. 4, **(2013)**, pp. 157-166.

[7] B. Jorge, "A framework for avoiding steganography usage over HTTP", Journal of Network and Computer Applications, vol. 35, no. 1, **(2012)**, pp. 491-501.
[8] L. C. Feng, and H. L. Chen, "A novel data hiding scheme based on modulus function", Journal of Systems and Software, vol. 83, no. 5, **(2010)**, pp. 832-843.
[9] H. Wien, T. S. Chen, and C. W. Luo, "Data embedding using pixel value differencing and diamond encoding with multiple-base notational system", Journal of Systems and Software, vol. 85, no. 5, **(2012)**, pp. 1166-1175
[10] U. Mustafa, G. Ulutas and V. V. Nabiyev. "Medical image security and EPR hiding using Shamir's secret sharing scheme", Journal of Systems and Software, vol. 84, no. 3, **(2011)**, pp. 341-353.
[11] S. Brindha, and I. Vennila, "Hiding Fingerprint in Face using Scattered LSB Embedding Steganographic Technique for Smart card based Authentication System", International Journal of Computer Applications, vol. 26, no. 10, **(2011)**, pp. 51-55.
[12] A. H. M. Kamal and M. M. Islam, "Facilitating and securing offline e-medicine service through image steganography", Healthcare Technology Letters, vol. 1, no. 2, **(2014)**, pp. 74-79.
[13] F. Jessica, M. Goljan and R. Du, "Lossless data embedding for all image formats", Electronic Imaging 2002. International Society for Optics and Photonics, **(2002)**.
[14] L. C. Chiuan, "An information hiding scheme with minimal image distortion", Computer Standards & Interfaces, vol. 33, no. 5, **(2011)**, pp. 477-484.
[15] T. Jun, "Reversible data embedding using a difference expansion", Circuits and Systems for Video Technology, IEEE Transactions on, vol. 13, no. 8, **(2003)**, pp. 890-896.
[16] H. Zhao, H. Wang, and M. K. Khan, "Statistical analysis of several reversible data hiding algorithms," in Proc. Multimedia Tools and Applications, DOI: 10.1007/s11042-009-0380-y, **(2009)**.
[17] W. X. Yang, "A robust blind color image watermarking in quaternion Fourier transform domain", Journal of Systems and Software, vol. 86, no. 2, **(2013)**, pp. 255-277.
[18] K. Lute, and H. J. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting", Image Processing, IEEE Transactions on, vol. 14, no. 12, **(2005)**, pp. 2082-2090.
[19] H. Wien and T. S. Chen, "A local variance-controlled reversible data hiding method using prediction and histogram-shifting", Journal of Systems and Software, vol. 83, no. 12, **(2010)**, pp. 2653-2663.
[20] P. Lei, "Full privacy preserving electronic voting scheme", The Journal of China Universities of Posts and Telecommunications, vol. 19, no. 4, **(2012)**, pp. 86-93.
[21] R. Lauretha, B. Issac, and M. K. Haldar, "Secure electronic voting system based on image steganography", Open Systems (ICOS), 2011 IEEE Conference on. IEEE, **(2011)**.
[22] H. Wien, T. S. Chen and C. W. Luo, "Data embedding using pixel value differencing and diamond encoding with multiple-base notational system", Journal of Systems and Software, vol. 85, no. 5, **(2012)**, pp. 1166-1175.
[23] A. H. M. Kamal, "Steganography: Securing Message in wireless network", International Journal of Computers & Technology, vol. 4, no. 3, **(2013)**, pp. 797-801.
[24] A. H. M. Kamal and M. M. Islam, "Capacity Improvement of Reversible Data Hiding Scheme through Better Prediction and Double Cycle Embedding Process", Accepted for Proc. IEEE Int Conference on Advance Networks and Telecommunication Systems, **(2015)**.
[25] A. H. M. Kamal and M. M. Islam, "Boosting up the data hiding rate multi cycle embedment process, J. Vis. Commun", Image R., in press, DOI: 10.1016/j.jvcir.2016.07.023, **(2016)**.
[26] A. H. M. Kamal and M. M. Islam, "Enhancing the Embedding Payload by Handling the Affair of Association and Mapping of Block Pixels through Prediction Errors Histogram", Proc. of Int Conference on Networks Systems and Security, **(2016)**.
[27] A. H. M. Kamal and M. M. Islam, "Enhancing the Robustness of Visual Degradation Based HAM Reversible Data Hiding", Journal of Computer Science, **(2015)**.

