An Approach for Enhancing Security of Cloud Data using Cryptography and Steganography with E-LSB Encoding Technique

Mohammad Obaidur Rahman[†], Muhammad Kamal Hossen[†*], Md. Golam Morsad[†], Animesh Chandra Roy[†], and Md. Shahnur Azad Chowdhury[††]

[†]Department of Computer Science and Engineering, CUET, Chittagong-4349, Bangladesh
[††]Department of Business Administration, IIUC, Chittagong-4318, Bangladesh
[*]Corresponding Author

Summary
Cloud data storage (Storage as a Service) is an important service of cloud computing referred to as Infrastructure as a Service (IaaS). The cloud storage provides data storage facilities as well as sharing across multiple users. Day by day it is gaining popularity because of enormous benefits. But emerging data security and privacy issues have become a subject of primo to the users as well as the service providers. That's why we proposed a technique for enhancing the security of cloud data using cryptography, steganography, and hash function. For cryptography, we use Blowfish algorithm and for steganography, a new efficient embedded algorithm using Embedded Least Significant Bits (E-LSB), and for integrity checking, we use SHA-256 Hashing Algorithm. This system is implemented in Eclipse using Java. We first encrypt the data and then hide it in an image to fulfill our purpose. As Blowfish [1] is an existing encryption algorithm which is secure enough, so, we just check out the steganography method's security. After hiding the data in a cover image, data detection and data destruction attacks are applied to evaluate the security of this steganography system. Detection attacks, such as visual attack, RS attack can't detect any data. In case of destruction attacks, such as jpeg compression, format conversion, salt& pepper, rotation, we got average NC value. So this steganography method is quite sensitive to destruction attack but it is secure in data detection attacks, which is the main purpose of steganography. During quality measurement, we are getting better PSNR value such as after hiding 1KB data in a cover image of size 512x512 pixels, we get PSNR value on average around 63 dB which is better than the previously existing methods.

Keywords:
Cloud data storage, Steganography, Blowfish, Embedded Least Significant Bits (E-LSB), SHA-256

1. Introduction

Cloud computing is an internet-based computing. It dynamically delivers everything as a service over the internet based on user demands, such as network, operating system, storage, hardware, software, and resources, etc. These services are classified into three types, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Cloud computing is deployed as three models, such as Public, Private, and Hybrid clouds.

Nowadays we are moving towards cloud computing. Cloud storage services enable users to outsource their data to cloud servers and access those data remotely over the Internet. These services give users an efficient and flexible way to manage their data without deploying and maintaining local storage devices and services. Specifically, users can process their data on their PCs, outsource the processed data to cloud servers, and use the data on other devices (e.g., mobile phones).

Along with various advantages, the cloud storage has gained great attention from both industry and academics since 2007. However, it also brings new challenges in creating a secure and reliable data storage and access facility over insecure storage and unreliable service providers. One of the challenges needs to be addressed before the novel storage model is applied widely is to keep the integrity of data stored in the cloud.

The cloud environment is vulnerable to different attacks which are giving a negative mark to impede the trust in adopting cloud computing. External and internal attacks have been identified as a threat to cloud infrastructures [1].

In this paper, we proposed a new scheme for secure and private data storing and data sharing technique considering data integrity. This will encrypt the data and then hide it in a cover image in a manner so that no one can detect those data from that stego-image. And for integrity check, there is no need for the third party for verification. The user thyself can do that easily.

2. Literature Review

In [2], the authors proposed a combined approach of steganography with LSB encoding technique and DES algorithm. Where they encrypted the data by DES

Manuscript received September 5, 2018
Manuscript revised September 20, 2018

encryption algorithm and then embedded the decrypted data by the LSB method. As LSB is not secure enough, we can say that this system doesn't provide better security. An advanced technique to share and protect cloud data using multilayer steganography and cryptography is used in [3]. Where data is encrypted by the AES encryption algorithm, then embedded the encrypted data in a cover image using Hash-LSB algorithm.

The authors of [4] implemented an improved LSB based image steganography technique for RGB color images that provide better PSNR value than previously used LSB methods. In [5], the authors provided a cryptographic public verification of data integrity for the cloud storage system, where data integrity is checked by the third-party auditor. But the corrupted third-party auditor may not provide proper verification.

In [6], the authors suggested an efficient data security method to control data in the cloud storage system using cryptographic techniques. Here data is encrypted and then store in the cloud. The authors of [7, 8] proposed a steganography technique where they used diffusion-based image compression techniques and where stego-image quality depends on the number of significant points.

A system enhancing data storage security in cloud computing through steganography is implemented in [9]. For enhancing, they hide the data in an image using steganography algorithm, then store that stego-image in the cloud. In [10], the authors proposed a multiple layer text security using variable block sized cryptography and image steganography where for embedding, authors used modified LSB together with raster scan technique.

The authors gave an opinion on that data security in cloud storage can be gained through cryptography [11]. Where data is encrypted by applying any encryption algorithm, then store in the cloud server. It is secure but for more sensitive data, it is not perfect. In [12], the authors put forward an algorithm for concealing message in encrypted images using a predetermined watermark embedding before the process of encryption. Here the encryption/decryption has a unique key and watermark processing has a different key, thus decryption of message is independent of extracting the image.

3. The Proposed System Architecture

The architecture of the proposed system is depicted in Fig. 1 and its working steps are described in the following section.

The steps involved in this scheme are explained in below:

1. Encryption: The secret message will be encrypted using the Blowfish encryption algorithm.
2. Embedding: In this step, we will hide the encrypted data in a cover image using E-LSB embedding algorithm which would create a stego-image as an output.
3. Hashing: Here the hash value of the stego-image will be calculated using the SHA-256 hashing algorithm for checking data integrity in the cloud storage later.
4. Retrieving: To extract the data from the stego-image, we need to apply the retrieving algorithm.
5. Decryption: For getting the secret message, the extracted data will be decrypted by Blowfish decryption algorithm.

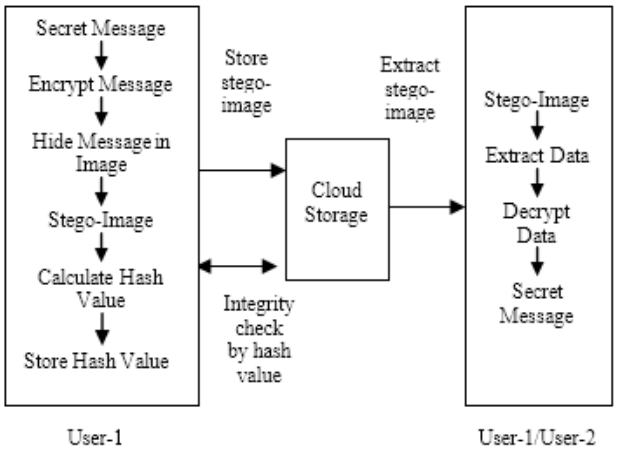


Fig. 1 The architecture of the proposed system.

In the proposed scheme, there are basically three options and those are described in below:

1. Secure data storage in the cloud: This involves two parts, i.e.,
   a. Data storing: For secure data storing, we need to use the encryption and the embedding algorithm.
   b. Data extraction: For secure data extraction, we need to apply the retrieving and the decryption algorithm.
2. Data integrity check: For checking the integrity of the data, the hash value of the stego-image needs to be calculated prior to store it in the cloud storage so that later we can check the integrity easily.
3. Secure data sharing in the cloud: For sharing the data securely, it is needed to provide adequate information to the receiver so that by using the retrieving and the decryption algorithm the receiver can extract the data from the cloud storage.

3.1 Embedded Least Significant Bits (E-LSB) Steganography Technique

In Embedded Least Significant Bits (E-LSB) process, all pixels of RGB images are broken into frames of 8-bits for Red (R), Green (G), and Blue (B) plane, respectively. To apply this technique, we first encrypt the message using Blowfish encryption method. Then the least significant 3, 3, and 2-bits of R, G and B frame respectively of a pixel of the image will get embedded with the encrypted data. We

take a character of the encrypted data and convert it into 8-bits binary data (ASCII value), then hide 3 least significant bits of the binary data in the Rplane, next 3 significant bits in the Gplane, and 2 most significant bits in the B plane respectively.
For calculating the value of every LSB for the reconstructed image pixel, we use equation (1).

$$\mathbf{V} = \mathbf{P} \oplus \mathbf{M} \quad (1)$$

Here, V = nth bit of pixel plane, P = kth bit of pixel plane, and M = mth bit of message.
For extracting, we just need to know the value of P and using equation (2), we can get the encrypted secret message again,

$$M = V \oplus P \quad (2)$$

Suppose we have to hide the encrypted binary message "10101101" in a pixel whose R, G, and B plane's value are "01101010", "11101011", and "10001001" respectively. For this example, we use P = 3rd bit (considering left to right) of each pixel plane for embedding. This process has shown in Table 1 and Table 2.

Table 1: Binary 8-bits frames of a pixel of the cover image (before embedding)

01101010	11101011	10001001

The encrypted message bits are: 10101101

Table 2: Binary 8-bits frame of a pixel of the stego-image (after embedding)

01101010	11101010	10001010

In the existing methods such as LSB, H-LSB, Modified LSB, anyone can easily extract the hidden data. But in E-LSB, we are not replacing LSB bits in actual data, so it's not possible to extract the hidden data from the stego-image without knowing the value of P. That's why it provides better security than the existing methods. And for the extraction process, we just need the stego-image only.

3.2 Blowfish Encryption with E-LSB Encoding

Below is the flowchart in Fig. 2 which show "how the proposed system will work" to save the information into the cloud storage. The user needs to pick two files, one is his/her information file and another one is the cover image where he/she wants to hide the information. The information texts are encrypted using the Blowfish algorithm. Then the encrypted information is embedded into the cover image using the above mentioned E-LSB technique.
The system will generate the reconstructed image. It also calculates the hash value of the stego-image and saves it for checking the integrity of the stored data later. Finally, the generated image gets stored in the cloud. Now, the data owner can share the same stego- image with other users also. For getting the information by the receiver, he/she needs to login into the same system and get original text by extracting it.

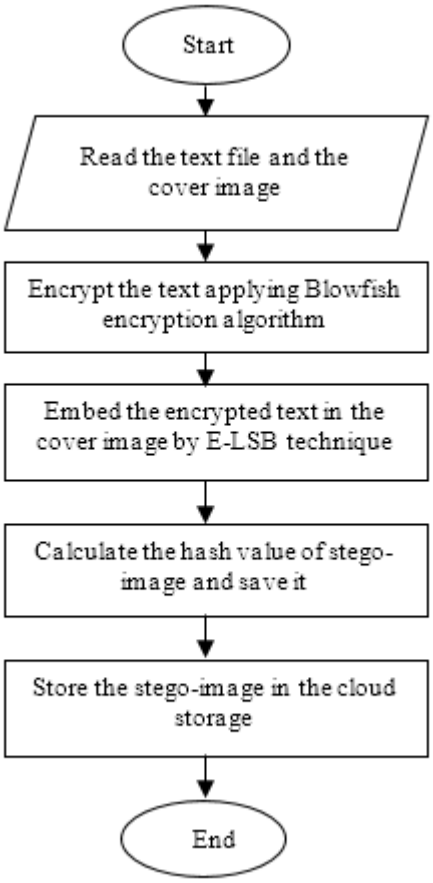


Fig. 2 The flowchart of Blowfish encryption with E-LSB encoding.

The methodology of hiding the encrypted message in the cover image is also stated in Algorithm 1.

Algorithm 1: The Blowfish encryption and embedding with E-LSB technique

1. Input the secret message and the cover image.
2. Encrypt the secret message by Blowfish encryption algorithm using the secret key.
3. Get the R, G, and B values of each pixel of the RGB format of the cover image.

4. Embed the encrypted message into R, G, and B planes of the cover image using E-LSB technique with equation (1) and the stego-key.
5. Obtain the stego-image by combining the three embedded planes.
6. Calculate the hash value of the stego-image and save it.
7. Store the stego-image in the cloud storage.

3.3 Blowfish Decryption with E-LSB Decoding

The working principle of the proposed system for getting back the secret data, the user/receiver needs to follow the steps as shown in the flowchart of Fig. 3. First, the user/receiver needs to download the stego-image. The embedded message is then extracted from the downloaded image following the reverse E-LSB technique stated in section 3.1. Finally, the Blowfish decryption method is applied with the secret key to get back the original secret data.
The process of extraction of the embedded message from the stored stego-image is also mentioned in Algorithm 2.

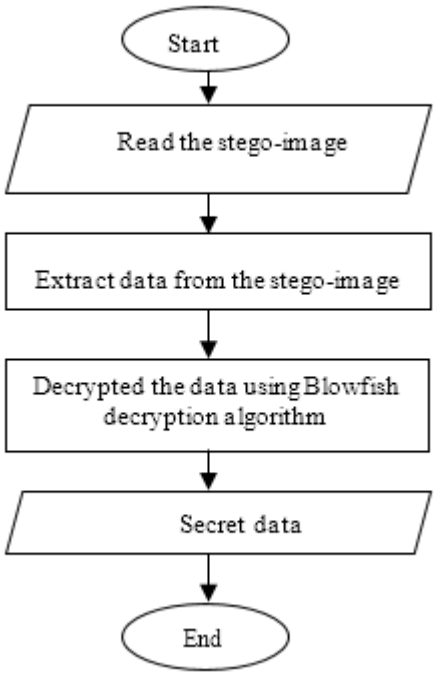


Fig. 3 The flowchart of E-LSB decoding with Blowfish decryption.

Algorithm 2: The Retrieval with reverse E-LSB and Blowfish decryption technique

1. Download the stego-image from the cloud storage.
2. Find the value of 3 (in case of R, and G) or 2 (in case of B) LSB bits of each RGB pixel of the downloaded stego-image.
3. Extract the embedded bits using equation (2) with the stego-key.
4. Combine the extracted bits to form the hidden data.
5. Apply Blowfish decryption over that hidden data using the secret key.
6. If the keys are correct, then the user/receiver will get the actual message back.

4. Experimental Result Analysis

As stated earlier that we are using steganography with encryption for providing cloud data security & privacy. To justify this, the proposed system is evaluated in the following sections. For evaluating a steganography system, Peak Signal to Noise Ratio (PSNR) value is considered as a parameter. First, we need to calculate Mean Square Error (MSE) value for calculating PSNR value. Equation (3) and equation (4) are used to calculate the MSE and PSNR value respectively.

$$\mathrm{MSE} = \frac{1}{\mathrm{L*H}} \sum_{\mathrm{L,H}} [\mathrm{O(L,H)} - \mathrm{R(L,H)}]^2 \qquad (3)$$

Here, O(L, H) represents the cover image, R(L, H) represents the stego-image, and L*H denotes the size of the cover image.

$$\mathrm{PSNR} = 10 \log_{10} \frac{255*255}{\mathrm{MSE}} \qquad (4)$$

For evaluating the performance of the developed system, we have used several 24-bits RGB images as cover images and hide 1KB message in each of the cover images. The simulation has been done using equation (3) and equation (4) in Eclipse with Java language. The result of the analysis is shown in Table 3 for six sample cover images with their respective MSE and PSNR values.
A comparative study of different cryptographic algorithms for data security in cloud computing is done in [14]. In Table 4, the comparison of the proposed method with the two existing methods [2, 3] on the basis of MSE and PSNR values for the six sample cover images is shown and found the better result for our system.

4.1 Attacks Analysis

For steganalysis, we use Chi-square attack and RS attack which are considered as statistical attacks, and Enhanced LSB attack which is considered as a visual attack. The

experimental results of different attacks on the proposed method are shown in Table 5.

By statistical attack using Chi-square attack, no embedded data has been detected from the stego-image. But in the visual attack using strong Enhance LSB attack, the attacker can detect the portion where data is hidden but not accurate size.

The system will secure the hidden data even if by any chance the attacker can detect it. If the attacker can detect, first, he/she needs to know the starting position of the stego-image where data has been embedded. For this, he needs to try 2n times, where n denotes data length. Even if he/she can get it, but he/she never gets the real embedded data.

Suppose we hide the encrypted data “11010010” in image pixel where R=“11011110”, G= “00110101”, and B= “10001001” and we select 2nd bit for embedding.

Table 3: The result of analysis with MSE and PSNR values

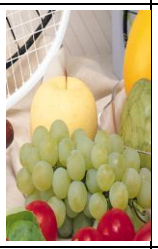
Image Details (512x512)	**Cover Image**	**Stego-image**	**MSE**	**PSNR (dB)**
Airplane .bmp			0.0301	63.33
Lena .bmp			0.0245	64.23
Pepper .bmp			0.0327	62.98
Baboon .png			0.0302	63.33
Fruits .png			0.0304	63.31
Cat .png			0.0298	63.38

Table 4: The comparison of the proposed method with the existing methods

Cover Image	Existing Method [2]		Existing Method [3]		The Proposed Method	
	MSE	PSNR (dB)	MSE	PSNR (dB)	MSE	PSNR (dB)
Airplane. bmp	0.2367	54.39	0.1111	57.67	0.0301	63.33
Lenna.bmp	0.1169	57.45	0.1078	57.80	0.0245	64.23
Pepper.bmp	0.5253	50.93	0.1112	57.65	0.0327	62.98
Baboon.png	2.9285	43.46	0.1135	57.58	0.0302	63.33
Fruits.png	0.5149	51.01	0.1102	57.71	0.0304	63.31
Cat.png	1.7500	45.68	0.1107	57.69	0.0298	63.38

After embedding the data, the value of that RGB pixel will be R= "1101101", G= "00110101", and B= "10001000".Now the attacker will get the data "10110100" but the actual embedded data is "11010010". So we can consider this system as a secure system.

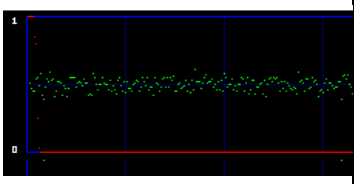
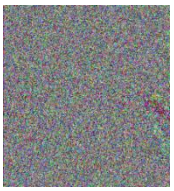
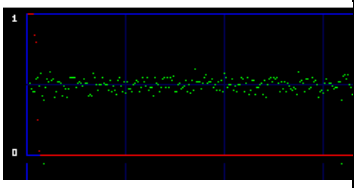
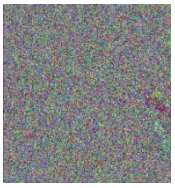
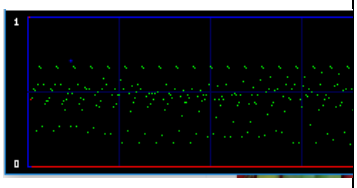
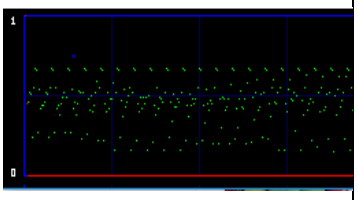
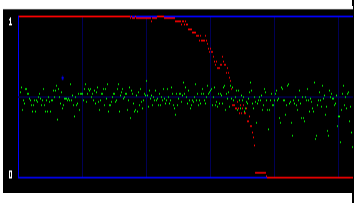
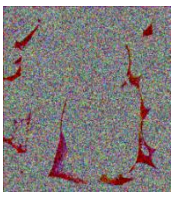
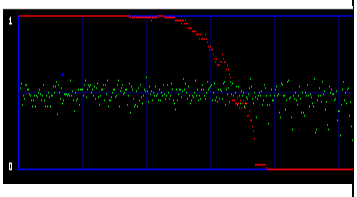
The main drawback of spatial-based steganography is that it is vulnerable to small manipulation in the stego-image. **If we convert the image of GIF/BMP format, which reconstructs the original message exactly (lossless compression) to a JPEG format, which does not (lossy compression), then the reverse could destroy the information hidden in the Least Significant Bits (LSBs) [13]. So, such kind of unintentional attacks is possible.**

In Table 6, we show the normalized correlation (NC) value analysis of different attacks on different images of size512x512 pixels embedded with 1KB data. Considering the NC value the developed method is fragile.

Table 5: Some unintentional attacks on the different images

Attack Name	NC Value			
	Pepper .bmp	Baboon .bmp	Airplane .png	Tiger.jpg
JPEG(30)	0.710163	0.745411	0.722208	0.748499
JPEG(90)	0.740435	0.743432	0.747495	0.758170
Salt & Pepper	0.997140	0.996959	0.998328	0.998429
Rotation 180°	0.739988	0.753884	0.750067	0.72862
Format Conversion	0.738158 0	0.746755	0.750705	0.753044

Table 6: The visual and statistical attacks analysis considering the text size of 1 KB

Sl. No.	Image	Attack Name		
		Chi-Square (Statistical Attack)	Enhanced LSB (Visual Attack)	RS (Statistical Attack)
1	 Original Image	 Initial status	 Initial status	Initial status
	 Stego-image	 Data detected = 0 KB	 No data detected	RS detected message length: 428 bytes (41.79%)
2	 Original Image	 Initial Status	 Initial Status	Initial status
	 Stego-image	 Data detected = 0 KB	 No data detected	RS detected message length: 59 bytes (5.76%)
3	 Original Image	 Initial Status	 Initial Status	Initial status
	 Stego-image	 Data detected = 0 KB	 No data detected	RS detected message length: 14 bytes (1.36%)

Conclusion

The proposed method uses the combination of cryptography and image steganography. For encrypting the secret message, the Blowfish encryption algorithm is used and for steganography, E-LSB based steganography is used. Both of them provide better security of the data. In the proposed method, we got better PSNR values in comparison with other existing methods which mean our system is better in terms of security. The developed method can hide approximately 16 KB message in a cover image of size 128×128 pixels, 256 KB message in a cover image of size 512×512 pixels, and 468 KB message in a cover image of size 800×600 pixels. Here we also used the SHA-256 hashing algorithm which calculates the hash value of the stego-image by which we can check the integrity of the data when it is stored in the cloud storage. In the future, video can be taken as cover media for increasing the data hiding capacity. Different ratio of data like 3:3:2 can be used for hiding and randomness can be increased between techniques by switching between the techniques randomly. The methods to reduce pixel variation can also be applied.

References

[1] S. Nawaz, M. Adib, M. Nawaz, and R. Kamran, "Identifying and analyzing security threats to virtualized cloud computing infrastructures", Proc. of IEEE International Conference on Cloud Computing Technologies, Applications and Managements, pp.151-155, 2012.

[2] B. Karthikeyan, A. Deepak, K. S. Subalakshmi, A. Raj, and V. Vaithiyanathan, "A combined approach of steganography with LSB encoding technique and DES algorithm", Proc. of 3rd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics, 2017.

[3] A. Ranjan, and M. Bhonsle, "Advanced technics to shared & protect cloud data using multilayer steganography and cryptography", Proc. of IEEE International Conference on Automatic Control and Dynamic Optimization Techniques, 2016.

[4] A. Singh, and H. Singh, "An improved LSB based image steganography technique for RGB color images", Proc. of IEEE International Conference on Electrical, Computer and Communication Technologies, pp.1-4, 2015.

[5] Y. Zhang, C. Xu, X. Liang, and H. Li, "Cryptographic public verification of data integrity for cloud storage system", IEEE Cloud Computing, vol.3, no.5, pp.44-52, 2016.

[6] G. L. Prakash, M. Prateak, and I. Singh, "Efficient data security method to control data in cloud storage system using cryptographic techniques", Proc. of IEEE International Conference on Recent Advances and Innovations in Engineering, pp.1-6, 2014.

[7] M. Mainberger, C. Schmaltz, M. Berg, J. Weickert, and M. Backes, "Diffusion-based image compression in steganography", Proc. of LNCS, vol.7432(II), pp.219-228, 2012.

[8] M. Gomathymeenakshi, S. Sruti, B. Karthikeyan, and M. Nayana, "An efficient arithmetic coding data compression with steganography", Proc. of IEEE International Conference on Emerging Trends in Computing and Nanotechnology, pp.342-345, 2013.

[9] M. K. Sarkar, and T. Chatterjee, "Enhancing data storage security in cloud computing through steganography", Proc. of ACEEE Int. Conf. on Network Security, vol.5, no.1, 2014.

[10] S. Chauhan, Jyotsna, J. Kumar, and A. Doegar, "Multiple layer text security using variable block size cryptography and image steganography", Proc. of 3rd IEEE International Conference on Computational Intelligence and Communication Technology, 2017.

[11] S. Kamara, and K. Lauter, "Cryptographic cloud storage", Proc. of Financial Cryptography and Data Security Workshops, pp.136-149, 2010.

[12] D. Bouslimi, G. Coatrieux, M. Cozic, and C. Roux, "Data hiding in encrypted images based on predefined watermark embedding before encryption process", Proc. of MEDECOM, vol.47, pp.263-270, 2016.

[13] E. Walia, P. Jainb, and Navdeep, "An analysis of LSB & DCT based steganography", Global Journal of Computer Science and Technology, vol.10, no.1(1), pp.4-8, 2010.

[14] P. Semwal, and M. K. Sharma, "Comparative study of different cryptographic algorithms for data security in cloud computing", Proc. of IEEE 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA) (Fall), pp. 1-7, 2017.

Mohammad Obaidur Rahman received the B. Sc. Engineering Degree in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh in 1998. He has currently received M. Engineering degree from the Department of CSE, Chittagong University of Engineering and Technology (CUET). From September 2001 to onwards, he has been serving as a faculty member in the Department of CSE, CUET, Bangladesh. He is currently working toward the Ph.D. degree as a part-time basis in the Department of CSE, CUET, Bangladesh. His major research interests include cognitive radio networks, game theory, neural networks, Internet security, and cryptography, etc.

MuhMuhammad Kamal Hossen has received his B. Sc. and M. Sc. in Computer Science & Engineering (CSE) degrees from the department of Computer Science & Engineering of Chittagong University of Engineering & Technology (CUET), Bangladesh in 2005 and 2015, respectively. He is now pursuing his Ph. D. degree in CSE from the same university. Since 2006, he has been serving as a faculty member in the Department of

CSE, CUET. His research interests include digital image processing, cryptography, steganography, pattern recognition, and data mining, etc.

Md. Golam Morsad received his B. Sc. Engg. degree from CUET, Bangladesh in 2017 in Computer Science and Engineering (CSE). He is now serving as a project engineer in a prominent mobile operator named Robi Axiata Limited, Dhaka, Bangladesh. His research interests include computer security, cryptography, and steganography, etc.

Animesh Chandra Roy received B. Sc. in Computer Science & Engineering degree from Department of Computer Science and Engineering of CUET), Bangladesh in 2014. He is now serving as a faculty member in the Department of CSE, CUET. His research interests include digital image processing, steganography, and pattern recognition, etc.

Md. Shahnur Azad Chowdhury received B. Sc. in Computer Science and Engineering from International Islamic University Chittagong (IIUC), Bangladesh in 2003 and M. Sc. in 2012 from Daffodil Int'l University, Dhaka. He has been serving the IIUC for the last fourteen years as a lecturer, assistant professor, and associate professor, respectively. His areas of research are natural language processing, data mining, IoT, Internet security, and cryptography, etc.