International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 4, Issue 3, May-June 2015 ISSN 2278-6856

An Improved Image Encryption Using Pixel XOR Transpose Technique Over Images

Peethala Kusuma Sree[1], **Madugula Padmaja**[2]

[1]PG Scholar, GITAM University, Visakhapatnam, India.

[2] Assistant Professor , CSE Department, GITAM University, Visakhapatnam, India.

ABSTRACT

Generally, Compression-Then-Encryption (CTE) paradigm meets the requirements in many secure scompression and encryption needs to be reversed in some other situations. Practically, image encryption has to be conducted prior to image compression. For designing a pair of image encryption and compression algorithms such that compressing the encrypted images can still be efficiently performed, there is some problem. Most of the existing Encryption-Then-Compression (ETC) solutions induce significant penalty on the compression efficiency. High level of security is being provided in the proposed image encryption scheme operated in the prediction error domain. It is also observed that in terms of compression efficiency, the proposed compression approach applied to encrypted images is only slightly worse than the state-of-the-art lossless/lossy image coders, which take original, unencrypted images as inputs. In this paper, we design a highly efficient image ETC system, where both lossless and lossy compressions are considered. We also demonstrate that an arithmetic coding-based approach can be exploited to efficiently compress the encrypted images.

Keywords: Image Encryption, Pixel XOR, Pixel Transpose, Image Compression

1. INTRODUCTION

Ever since childhood, my interest was on cryptography. My favourite cipher from the Pen-and-Paper Era of Cryptography was the one-time pad because it was the only one that was proven to be theoretically unbreakable by information-theory guru Claude Shannon. Now that we're in the Cyber Era, the sophistication of ciphers for text and data streams has moved far beyond anything Lord Playfair and Vigenère could dream of to encompass things like quantum cryptography and public-key systems. Because mainstream cryptography has moved far beyond the pig-pen code and the scytale, my interest have been increasingly drawn to the odd corners of the cryptographic world. One of my fringe cryptographic interests was in finding a way to encrypt images graphically in a way that could be done with any reasonably good graphics software by anybody who isn't a professional cryptographer.

The tried-and-true method of adding encryption to a picture is through steganography, which is the art of creating hidden images. In the digital world, this is done by methods like least-significant bits in bitmap images or flashing subliminal messages in a video stream. Steganography is very useful for putting digital watermarks in an image; however, it isn't very secure from a cryptanalytical perspective. Once an eavesdropper suspects that there's a hidden message in an image, it's usually broken very easily.

If security of the image is paramount, then the usual method is to take the image file and encrypt that like any other data file. This has several drawbacks: first, if you're not well-educated in cryptography and computer programming, you have to run out and buy somebody else's encryption software. Then there's the very likely possibility that the software company or some government agency has a 'backdoor' method of reading files encrypted by the software. Finally, any cryptographic system that isn't based on a random, one-time key is theoretically breakable.

A graphics-based system for encrypting images has several advantages. For starters, you're not dependent upon a specific piece of encryption software, but can use just about any reasonably sophisticated graphics program. You directly control the making of cryptographic keys and the encryption process rather than have it happen 'under the hood'. This gives you reasonable assurance that no software company or government agency has a backdoor way of hacking into your encrypted images. Since my method is based on the graphics equivalent of a one-time key, it has the advantage of being unbreakable--at least in theory. And, finally, making your own encrypted images with graphics software is gosh-darned fun for those of us who are terminally geeky.

So, how does one go about encrypting an image anyways? My intuitive guess was that a key image the exact size of the plain image would be added as a layer in Photoshop, then some sort of filter would add the two images together into a final encrypted image. It seemed obvious that the key image would have to be made out of noise (resembling 'snow' on old analog TVs) as that seemed like the closest graphical analog to a one-time key. Also experimented with the different filters in my graphics program and finally decided that the Difference filter was the most useful because it was fully reversible. The test encryption is done by adding noise to a normal image to get the following:

As you can see, the results are less than perfect and it's quite easy to make out the original image in the noise. Being still not 100% sure why it didn't work as planned, this failure spurred me on to find a better solution. First read up on Moni Naor and Adi Shamir's clever method of visual cryptography using plastic transparencies. Then modified this method to work with my graphics software and later expanded it to work with RGB images instead of just black and white. Then removed a superfluous step or two--and pretty much ended up where it started, but with a slight twist. That slight twist was to reduce the number of colors in the key and plain images to a specific eight-color palette before adding the two images together. This was the theoretical breakthrough that made my method practically.

Nowadays, information security is becoming more important in data storage and transmission. Images are widely used in several processes. Therefore, the protection of image data from unauthorized access is important. Image encryption plays a significant role in the field of information hiding. Image hiding or encrypting methods and algorithms range from simple spatial domain methods to more complicated and reliable frequency domain ones. From the study of research paper and other my conclusion was that in there are no clarifications which type of images they are using to perform image encryption and decryption procedure. And also analyzed that there is no clarification about the configuration of machine and platform where all the experiment are calculating. Another thing which was measured by me was that the proposed transformation table have very complex structure and not easy to understand which is the cause of poor efficiency. From further study it is observed that Images are different from text. Although we may use the raditional cryptosystems to encrypt images directly, it is not a good idea for two reasons. One is that the image size is almost always much greater than that of text. Therefore, the traditional cryptosystems need much time to directly encrypt the image data. The other problem is that the decrypted text must be equal to the original text. However, this requirement is not necessary for image data. Due to the Characteristic of human perception, a decrypted image containing small distortion is usually acceptable.

2. OVERVIEW

2.1 PIXEL XOR TRANSPOSE TECHNIQUE

In this module the image should be encrypt and decrypt by using pixel XOR transpose technique. In this the sender will encrypt the image before sending it. The sender will take the data hide image and encrypt it by performing XOR operation and then followed by transpose operation. The output image from these pixels is the encrypted image. The receiver will do the reverse operation for decryption.

2.2 ADVANCED ENCRYPTION STANDARD (AES)

In this module the AES (Advanced Encryption Standard) Algorithm is used for encrypting the message by the sender. This data is converted into the binary format by the sender. The hidden binary data is retrieved by the receiver. The decryption process is done by the receiver using the same AES Algorithm.

2.3 ARITHMETIC CODING

After image encryption the sender will compress the sent image. The compression of image can be done by using arithmetic coding technique. The compressed image will be sent to the receiver. The receiver will retrieve the compressed image and decompress that image using arithmetic coding technique. By performing this technique we can improve efficiency of given system.

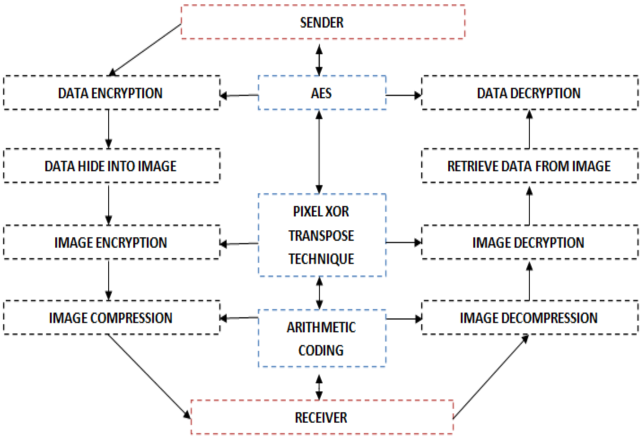


Fig.1 SYSTEM DESIGN

3. EXISTING SYSTEM

In previous days the image is transferred through network in the form of plain format. Transferring image through network in the form of plain format will cause security problem. Security can be provided by transferring the image in the form of unknown format. By converting it into the unknown format encrypting the image is done by using any technique.

3.1 DISADVANTAGES

I) The transfer of data and image through network will cause a problem of security.

II) The problem of size by transferring the data and the image through network is also found. Also if the size increases then the efficiency of the network will decrease.

3.2 PROBLEM IDENTIFICATION

The identification of problem is to reduce the security of given data and image. Another issue is to increase efficiency of the existing system. So these are the problems to identify problem of existing system.

3.3 PROBLEM DEFINITION

Nowadays, information security is becoming more important in data storage and transmission. Images are widely used in several processes. Therefore, the protection of image data from unauthorized access is important. Image encryption plays a significant role in the field of information hiding. Image hiding or encrypting methods and algorithms range from simple spatial domain methods to more complicated and reliable frequency domain ones. From the study of research paper and other it is concluded that in there are no clarifications that which type of images they are using to perform image encryption and decryption procedure. It is also analyzed that there is no clarification about the configuration of machine and platform where all the experiments are calculated. Another thing which is measured is that proposed transformation table have very complex structure and not easy to understand which is the cause of poor efficiency. From further study it is observed that Images are different from text. Although we may use the raditional cryptosystems to encrypt images directly, it is not a good idea for two reasons. One is that the image size is almost always much greater than that of text. Therefore, the traditional cryptosystems need much time to directly encrypt the image data. The other problem is that the decrypted text must be equal to the original text.

4. PROPOSED SYSTEM

Now-a-days steganography places an important role for provide more security in the network. Because the steganography technique will give more security and those are efficient one. In this paper it is proposed that concepts of data hiding, image encryption and compression of image. So using those techniques the efficiency and security of given system can be improved. The following are the steps involved in the proposed system.

4.1 ENCRYPTION KEY GENERATION

The Encryption Key Generation is the basic step which is used for the encoding and decoding of the data. The following are the steps used for generating Encryption Key:

1. Sender and receiver agree to use a prime numbers *p* and *g*.

2. Sender chooses a secret integer ***a***, then sends receiver the calculated public key *A =(g pow (a))* mod *p*.

3. Receive chooses a secret integer ***b***, then sends sender the calculated public key *B* = (*g pow* (*b*)) mod *p*.

4. Sender computes ***e*** = (*B pow (a)*) mod *p*.

5. Receiver computes *e* = (*A pow* (*b*)) mod *p*.

6. Now the sender and the receiver contain same encryption key.

4.2 ENCRYPTION AND DECRYPTION OF MESSAGE

Before sending the message the sender will encrypt the message using Steganography technique. In this project the AES algorithm is used for message encryption and decryption purpose. After the generation of public key encryption of the message using this key is done. After encryption the original message is converted into a cipher text. This cipher text is converted back to the original text at the receiver side.

4.3 DATA HIDE INTO IMAGE

In this module the sender will hide data into sent image by using LSB technique. Before storing data into image, encryption is done by using AES algorithm. After encrypting the data the sender will get cipher text and convert into binary format. After converting into binary format the sent image is retrieved and the pixels of the image is displayed. In each pixel the LSB bit is replaced with the data. This process will continue complete data is hidden. After hiding the data into pixel the data hide image is obtained.

4.4 PIXEL XOR TRANSPOSE TECHNIQUE

In this module the image should be encrypted and decrypted by using pixel XOR transpose technique. In this the sender will encrypt the image before sending it. The sender will take the data hide image and encrypt by using following procedure.

4.4.1 ENCRYPTION PROCESS

The encryption process is used to convert the image into the unknown format. This can be done using the following steps:

(i) The whole image is selected and named as I.

(ii) Then store all the pixel values of I in a two dimensional array named as P like every image have rows or column wise pixels.

(iii) Now, row wise XOR all the bits of pixel from top to bottom like firstly XOR first and second row and then store first row as XOR Result and second rows as it is. Then XOR second and third rows and store according to previous operation and then apply to all the rows.

(iv) A square grid of required size constructed by taking binary data from the XOR data.

(v) Now grid transposition applied by reading data diagonally and writing it down on row basis from left to right.

(vi) A new grid is generated after transposition.

(vii) The new grid is converted into ASCII sequence and written into Image file.

4.4.2 DECRYPTION PROCESS

The encrypted image is now decrypted by the receiver to obtain the original image. After decompression of image, the image can be converted into pixel and perform the decryption process.

(i) square grid of required size is constructed by taking binary data from the Image file.

(ii) Now grid transposition is applied by reading data row wise and writing it down diagonally from left to right.

(iii) A new grid is generated after transposition.

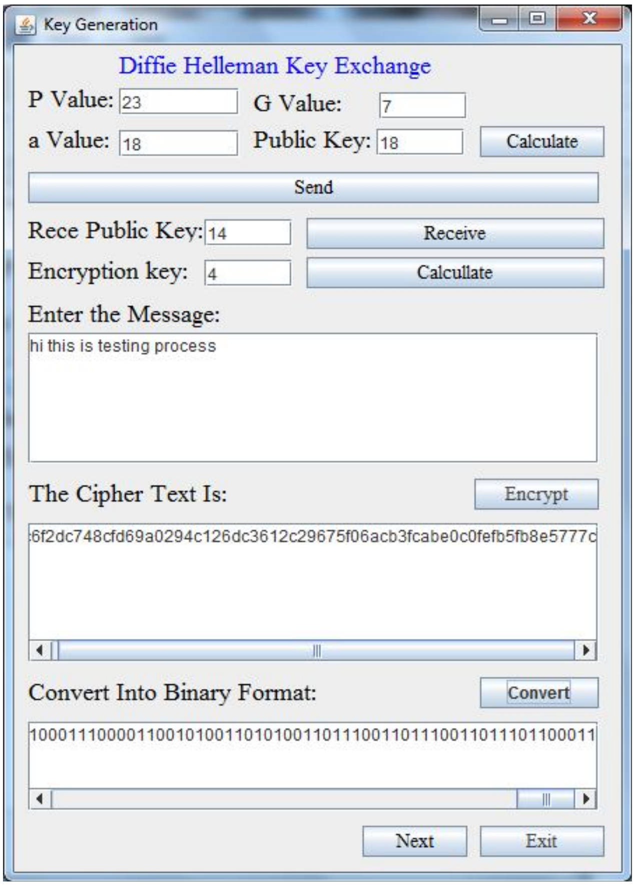
(iv) Now, row wise XOR of all the bits of pixel from top to bottom like firstly XOR first and second row and then store first row as XOR Result and second rows as it is. Then XOR second and third rows and store according to previous operation and then apply to all the row.

(v) Then store all the pixel values of I in a two dimensional array named as P and obtain the original image.

4.5 COMPRESSION AND DECOMPRESSION IMAGE

After completion image encryption the sender will compress the sent image. The compression of image can be done by using arithmetic coding technique. The completion of image compression the sender will send to receiver. The receiver will retrieve the compressed image and decompress that image using arithmetic coding technique. By performing this technique we can improve efficiency of given system.

5. EXPERIMENTAL RESULT



Diffie Helleman Key Exchange

P Value: 23 G Value: 7

a Value: 18 Public Key: 18

Rece Public Key: 14

Encryption key: 4

Enter the Message:

hi this is testing process

The Cipher Text Is:

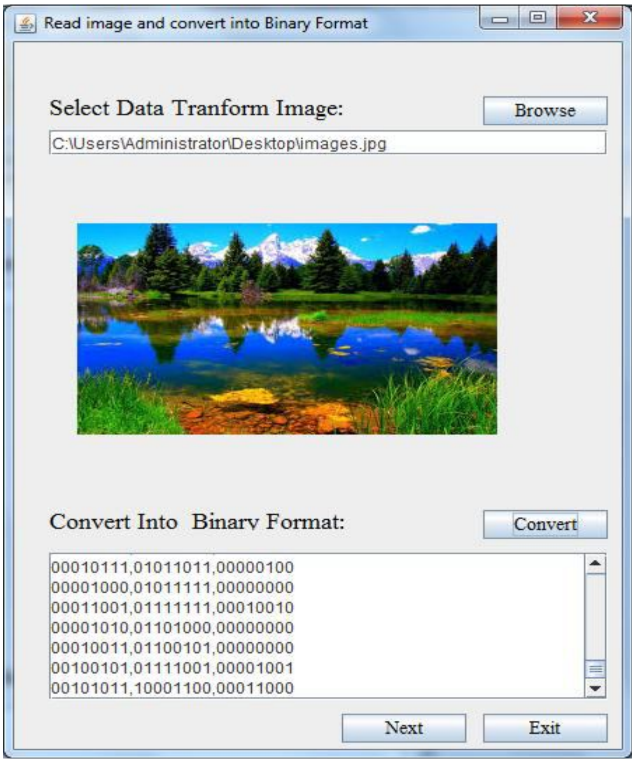
6f2dc748cfd69a0294c126dc3612c29675f06acb3fcabe0c0fefb5fb8e5777c

Convert Into Binary Format:

1000111000011001010011010100110111001101110011011101100011

Fig 2. DEFFIE HELLMAN KEY EXCHANGE

Fig. 2 is the page after sender login and here the public key is generated and sent to the receiver. The receiver's public key is received. Using the receiver's public key the encryption key is generated. The message is encrypted and converted to binary format.



Select Data Tranform Image:

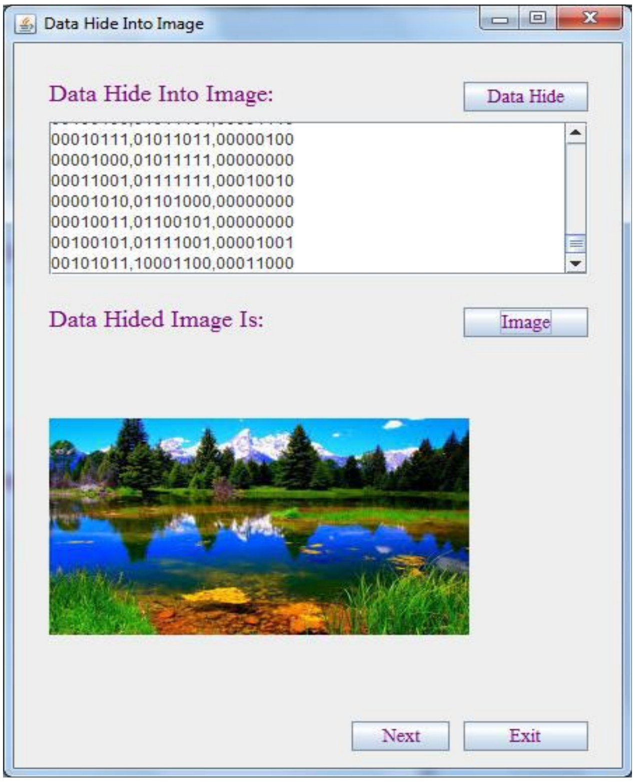
C:\Users\Administrator\Desktop\images.jpg

Convert Into Binary Format:

```
00010111,01011011,00000100
00001000,01011111,00000000
00011001,01111111,00010010
00001010,01101000,00000000
00010011,01100101,00000000
00100101,01111001,00001001
00101011,10001100,00011000
```

Fig. 3 READ IMAGE

In Fig. 3 image is browsed and the R,G,B Values are obtained.



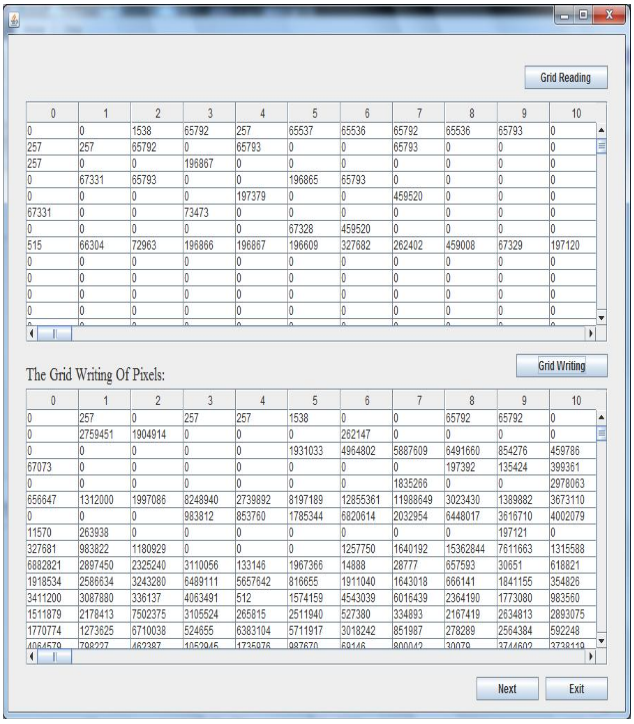
Data Hide Into Image:

```
00010111,01011011,00000100
00001000,01011111,00000000
00011001,01111111,00010010
00001010,01101000,00000000
00010011,01100101,00000000
00100101,01111001,00001001
00101011,10001100,00011000
```

Data Hided Image Is:

Fig 4. DATA HIDE

In Fig. 4 R,G,B Values, the binary values of the message are saved.



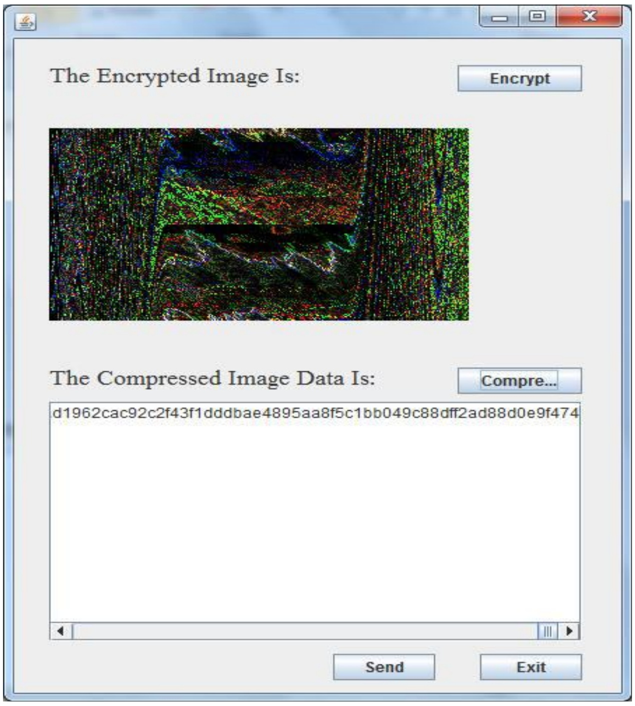
0	1	2	3	4	5	6	7	8	9	10
0	0	1538	65792	257	65537	65536	65792	65536	65793	0
257	257	65792	0	65793	0	0	65793	0	0	0
257	0	0	196867	0	0	0	0	0	0	0
0	67331	65793	0	0	196865	65793	0	0	0	0
0	0	0	0	197379	0	0	459520	0	0	0
67331	0	0	73473	0	0	0	0	0	0	0
0	0	0	0	0	67328	459520	0	0	0	0
515	66304	72963	196866	196867	196609	327682	262402	459008	67329	197120
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

The Grid Writing Of Pixels:

0	1	2	3	4	5	6	7	8	9	10
0	257	0	257	257	1538	0	0	65792	65792	0
0	2759451	1904914	0	0	0	262147	0	0	0	0
0	0	0	0	0	1931033	4964802	5887609	6491660	854276	459786
67073	0	0	0	0	0	0	0	197392	135424	399361
0	0	0	0	0	0	0	1835266	0	0	2978063
656647	1312000	1997086	8248940	2739892	8197189	12855361	11988649	3023430	1389882	3673110
0	0	0	983812	853760	1785344	6820614	2032954	6448017	3616710	4002079
11570	263938	0	0	0	0	0	0	0	197121	0
327681	983822	1180929	0	0	0	1257750	1640192	15362844	7611663	1315588
6882821	2897450	2325240	3110056	133146	1967366	14888	28777	657593	30651	618821
1918534	2586634	3243280	6489111	5657642	816655	1911040	1643018	666141	1841155	354826
3411200	3087880	336137	4063491	512	1574159	4543039	6016439	2364190	1773080	983560
1511879	2178413	7502375	3105524	265815	2511940	527380	334893	2167419	2634813	2893075
1770774	1273625	6710038	524655	6383104	5711917	3018242	851987	278289	2564384	592248
4064579	798227	462387	1052945	1735976	987670	69146	800042	30079	3744602	3738119

Fig. 5 PIXEL XOR TRANSPOSE

In Fig. 5 the PIXEL XOR is done and also the TRANSPOSE is done diagonally. This is the image encryption.



d1962cac92c2f43f1dddbae4895aa8f5c1bb049c88dff2ad88d0e9f474

Fig. 6 IMAGE COMPRESSION

In Fig. 6 the image compression is done using Arithmetic Coding. This compressed data is sent to the receiver.

The receiver will receive the sender's public key and using that the encryption key is generated. This key will be same as the sender's encrypted key. Now the received compressed data is decompressed and decrypted by doing all the reverse process.

6. CONCLUSION

Within the proposed framework, the image encryption will be achieved. Highly efficient compression of the encrypted data has then been realized by a context-adaptive arithmetic coding approach. More notably, the coding efficiency of our proposed compression method on encrypted images is very close to that of the state-of-the-art lossless/lossy image codecs, which receive original, unencrypted images as inputs. The image obtained at the receiver end will be secure compared to previous methods like sending through plain format.

REFERENCES

[1] J. Zhou, X. Liu, and O. C. Au, "On the design of an efficient encryptionthencompression system," in Proc. ICASSP, 2013, pp. 2872–2876.

[2] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," IEEE Trans. Inf. Forensics Security, vol. 4, no. 1, pp. 86–97, Mar. 2009.

[3] T. Bianchi, A. Piva, and M. Barni, "Encrypted domain DCT based on homomorphic cryptosystems," EURASIP J. Inf. Security, 2009, Article ID 716357.

[4] T. Bianchi, A. Piva, and M. Barni, "Composite signal representation for fast and storage-efficient processing of encrypted signals," IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 180–187, Mar. 2010.

[5] M. Barni, P. Failla, R. Lazzeretti, A.-R. Sadeghi, and T. Schneider, "Privacy-preserving ECG classification with branching programs and neural networks," IEEE Trans. Inf. Forensics Security, vol. 6, no. 2, pp. 452–468, Jun. 2011.

[6] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," IEEE Trans. Inf. Forensics Security, vol. 7, no. 3, pp. 1053–1066, Jun. 2012.

[7] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[8] D. Schonberg, S. C. Draper, and K. Ramchandran, "On blind compression of encrypted correlated data approaching the source entropy rate," in Proc. 43rd Annu. Allerton Conf., 2005, pp. 1–3.

[9] D. Schonberg, S. C. Draper, and K. Ramchandran, "On compression of encrypted images," in Proc. IEEE Int. Conf. Image Process., Oct. 2006, pp. 269–272.

[10] Menda.VasudhaRani and JayanthiRaoMadina, "An Improved Pixel Transpose Technique over Images"

in International Journal of Engineering Trends and Technology, Nov2014.

AUTHORS

Peethala Kusuma Sree is a student of M.Tech (CSE) in GITAM University, Visakhapatnam, Andhra Pradesh, India. I received my B.Tech (ECE) from Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, Andhra Pradesh, India. My area of interest are Information Security, Image Processing, Network Security, Java.

M.Padmaja, received her M.Tech degree in Software Engineering from JNTU, Anantapur during the year 2006. Presently she is working as Assistant professor in the Department of CSE, Gitam Institute of Technology, GITAM University, Visakhapatnam, Andhra Pradesh, India. Her research area of interest are Software Engineering, Mining, Computer Networks and Image Processing.