Multimedia Security

Chirag Tahilyani[#1],Niketa Chellani [#1], Ankur Bohra [#1], Varun Darak [#1]
Mani Roja [#2], Anushree Gupta [#2]

[#1] Student, Electronics and Telecommunication Dept, TSEC, Mumbai University
[#2]Associate Prof, TSEC, Mumbai University

Abstract

Multimedia encryption is widely used to ensure security in open networks such as the internet. Each type of data has its own features; therefore, different techniques should be used to protect confidential data from unauthorized access. Most of the available encryption algorithms are used for text data. However, due to large data size and real time constrains, algorithms that are good for textual data may not be suitable for multimedia data like images, audio and video. This paper aims at implementing and analysing secure encryption algorithms (scrambling, key based pixel manipulation, random mapping, and encryption using random sequences) for multimedia content files. It compares the various parameters such as time taken, entropy, mean square error and also its histograms for these algorithms.

Keywords

Scrambling, Random sequence, pixel manipulation, Random mapping.

1. Introduction

Multimedia data encryption attempts to prevent unauthorized disclosure of confidential multimedia information in transit or storage.

The most straightforward technique for multimedia encryption is to treat the multimedia signal as a traditional digital data stream, such as text, and select an application appropriate classical encryption scheme and key management scheme to encrypt the entire data stream. Upon reception, the entire cipher text data stream would be decrypted and playback can be performed at the client device. The key factors to consider when choosing an application appropriate encryption scheme include:

- It should provide suitable security for the target application
- It should be cost effective for the specific application and the end user device capability.

There are three areas which come under Multimedia Encryption:

a. Image
b. Audio
c. Video

2. Understanding and implementing encryption and decryption of audio and image

Image can be considered to be a block of pixels arranged in a rectangular (or square) manner. The values of these pixels are between 0-255[1]. Audio, on the other hand, consist of a matrix of intensity values and has a dimension m*1 where m is the no of rows. Since the dimensions of the image and audio matrices are different it would be difficult to implement the same algorithms on both the matrices. Hence, the audio matrices are manipulated in such a way that it looks like an image matrix which makes it possible for us to apply image encryption algorithms on audio.

Converting Sound to Image:-

- The number of samples in the audio matrix is calculated.
- A perfect square closest to that number is found out which is greater than that number
- The square root of the perfect square is then calculated (assume m).

- A square matrix of all zeroes of size m*m is generated which looks like an image matrix.
- The values of audio intensities are then filled in the image matrix in a sequential manner column-wise.

Thus an audio matrix is converted to an image matrix.

2.1. Implementing different algorithms for Audio and Image

2.1.1. Block Scrambling

The basic principle of this technique is to divide the image into a number of blocks and scramble them with the help of a certain sequence. The decryption process involves rearranging the blocks with the help of the exactly reverse sequence to retrieve the original image [2].

Algorithm:-

- Divide the matrix into blocks of n*n; where n can be any value. The complexity of the algorithm depends on the value of 'n'.
- Higher the value of 'n', more will be the complexity of the algorithm and hence more secure will be the encryption.
- Scramble the rows of the matrix. This sequence acts as a key for encryption procedure.
- Scramble the columns of the matrix with the help of the same sequence. This gives us our encrypted matrix.
- The decryption involves the exact reverse procedure of the encryption.

2.1.2. Encryption using Random Sequence Generation

The principle of this algorithm is to generate an encrypted matrix by ex-oring each pixel value in the original image/sound with random values [3]. Before ex-oring, the pixel values and the random values have to be first converted to binary. For decryption, the encrypted matrix is again ex-ored with the same random values to obtain the original image back.

Algorithm:-

- Assume an m*n matrix consisting of pixel values of an image/sound. Each pixel of this matrix is converted into an 8 bit binary number (2^N bits binary representation can be used for each pixel generally).
- Another matrix consisting of m*n random values is generated. The generation is done using this function in Matlab :

  randint (m, n, max, key)

  Where, the matrix is of size m*n; max is the greatest value of integer that can be present in the values of the random sequence and key is the encryption key is used for encryption and decryption. The random sequence generated depends on the value of 'key' [4].
- Each pixel value of the original image/sound is ex-ored with the corresponding value of the random matrix generated using the encryption key to obtain the encrypted matrix.
- To obtain the original pixel values back from the encrypted values, we ex-or the encrypted values again with the same key. This gives back the original image/sound.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \oplus \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \oplus \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

Figure 1. Illustration of encryption using Random sequence generation

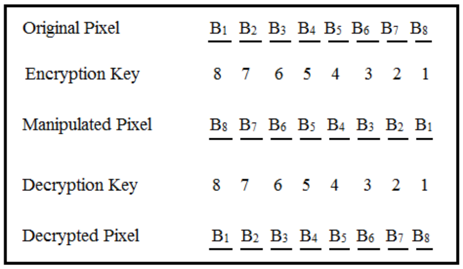
2.1.3. Key based Pixel Manipulation

The basic principle of this algorithm is to encrypt an image/sound by making it difficult to understand by scrambling the bits positions of each of its pixel according to the encryption key.

Algorithm:-

- Convert each value of the matrix into an 8 bit binary number.
- Enter an 8 digit encryption key having values from 1-8 in any random order.

- Change the bit position of each pixel in accordance to the key.
- Apply the same key again to the encrypted pixels to retrieve the original pixel values back. Applying the same key on the encrypted image/sound re-arranges the bit positions of the pixels to give back the original pixel value.



Original Pixel	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
Encryption Key	8	7	6	5	4	3	2	1
Manipulated Pixel	B_8	B_7	B_6	B_5	B_4	B_3	B_2	B_1
Decryption Key	8	7	6	5	4	3	2	1
Decrypted Pixel	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8

Figure 2. Illustration of encryption using Key based Pixel Manipulation

2.1.4. Random mapping

The proposed method adopts the classic framework of the permutation-substitution properties in cryptography and thus ensures both confusion and diffusion properties for a secure cipher. The proposed method is able to encrypt an intelligible image into a random-like one from the statistical point of view and the human visual system point of view [5]. The three basic steps involved in encryption of an image using random mapping are permutation, substitution and transposition. The decryption involves just the reverse.

Algorithm for encryption:-

- Permute an m*n matrix of an image. This permutation depends on the random map selected. This map should have good random properties (good examples of such maps include baker map, cat map etc.). The result of this step is an image with its pixels scrambled.
- As in the previous step, though the pixels have been scrambled, the statistical relation between the pixels in the original image and the resultant image remains the same. For this purpose substitution is performed and so the pixel values are substituted with the values from another matrix.
- Finally the elements of the resultant matrix are transposed to obtain the encrypted image[6].

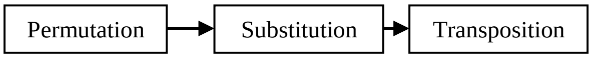


Figure 3. Steps involved encryption using random mapping

- For the purpose of decryption, the transpose of the encrypted matrix is taken.
- The original values of the original image are re-substituted by using now using the same substituting matrix in a reverse manner.
- Now, using the same map used while encryption the reverse permutation process has to be applied to obtain the original image.

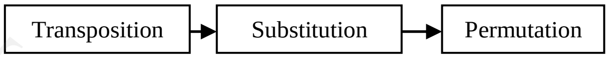


Figure 4. Steps involved decryption using random mapping

3. Understanding and implementing encryption and decryption of video

Video is combination of many images scanned at a particular rate and the associated synchronized sound. Thus its encryption would include the separation of both the image and sound from the video and its encryption [7]. Decryption would involve the actual decryption of the image and sound and then their combination appropriately to obtain the original video.

Encryption:-

- Separate the audio and visual component from a video file. Further, extract image frames from the visual component, forming a series of images.
- Now, apply Random Noise Addition technique to encrypt the audio component and Random Sequences generation algorithm (as discussed earlier) on each of the extracted image frame. This results in an encrypted audio file and a series of encrypted images [6].

- Random Noise Addition is a technique in which a noise matrix (random matrix) is added to the original audio matrix to make it unrecognizable.
- Permute (change the order of frames) the encrypted image frames to obtain the encrypted visual component [2].
- Combine the encrypted audio and visual components. The result is an encrypted video.

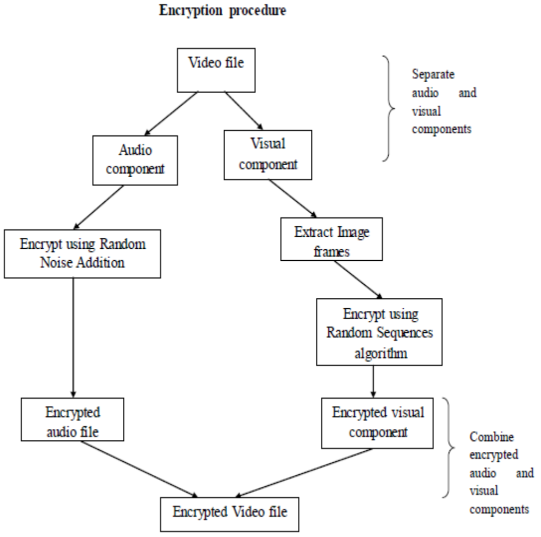

Encryption procedure
Video file
Separate audio and visual components
Audio component
Visual component
Encrypt using Random Noise Addition
Extract Image frames
Encrypt using Random Sequences algorithm
Encrypted audio file
Encrypted visual component
Combine encrypted audio and visual components
Encrypted Video file

Figure 5. Flow Chart of video encryption

Decryption:-

- Separate the encrypted audio and visual components from the encrypted video. Extract the encrypted image frames from the visual component.
- Decrypt the encrypted audio file and each of the encrypted image frames using the encryption algorithms in the reverse manner.
- Arrange the decrypted image frames into their original sequence and combine them to form the decrypted visual component.
- Combine the decrypted audio and visual components to obtain the decrypted video.

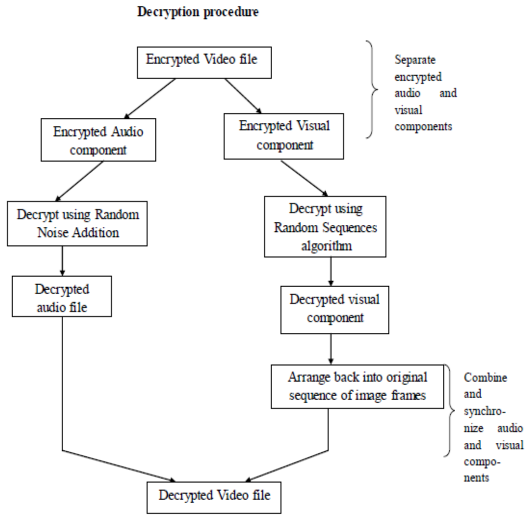

Decryption procedure
Encrypted Video file
Separate encrypted audio and visual components
Encrypted Audio component
Encrypted Visual component
Decrypt using Random Noise Addition
Decrypt using Random Sequences algorithm
Decrypted audio file
Decrypted visual component
Arrange back into original sequence of image frames
Combine and synchro-nize audio and visual compo-nents
Decrypted Video file

Figure 6. Flow Chart of video decryption

4. Analysis

4.1 Parameters used for comparison

4.1.1. Time taken

The time taken for an encryption algorithm is the total implementation time for the encryption plus decryption of the entire multimedia file.

Time taken cannot be considered as a proper parameter of comparison as the time taken for a program to execute will depend on the processing speed of the computer and the version of Matlab. However here we have calculated time taken to show the relative idea of the time taken to complete the encryption and the decryption process.

4.1.2. Entropy

Entropy is the randomness/irregularity in the system i.e. image, audio or video file in this paper [8]. A comparison of the entropies of the original, encrypted and decrypted image gives a measure of the effectiveness and security of the algorithm.

4.1.3 Mean square error

The mean square error is a measure of the difference between the original and decrypted image [8]. Less the MSE more is the accuracy of the encryption and decryption method. MSE can be mathematically represented as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y-\tilde{y})^2 \quad \textbf{(1)}$$

Where,

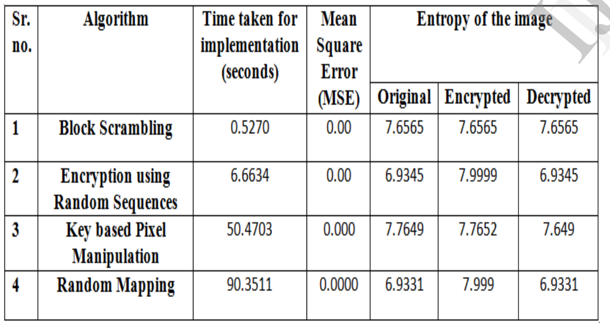
y = the original value
$\tilde{y}$ = the expected value (here, the decrypted value)
n = the total number of values

4.1.4. Histogram

Histogram is the plot of number of pixels/intensity versus the different value of pixels/intensities present in the sample

4.2. Analysis of the Encryption and Decryption algorithms implemented on Image

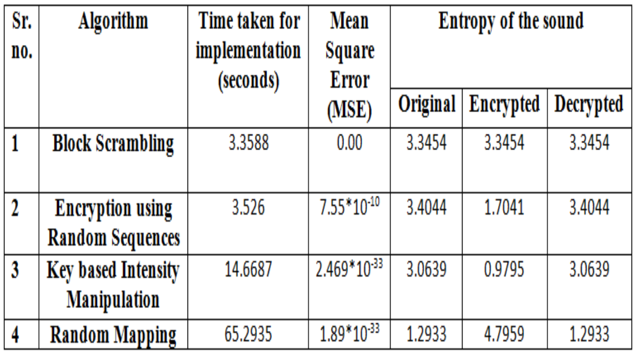
Table 1. Comparison of encryption and decryption algorithms for image



Sr. no.	Algorithm	Time taken for implementation (seconds)	Mean Square Error (MSE)	Entropy of the image		
				Original	Encrypted	Decrypted
1	Block Scrambling	0.5270	0.00	7.6565	7.6565	7.6565
2	Encryption using Random Sequences	6.6634	0.00	6.9345	7.9999	6.9345
3	Key based Pixel Manipulation	50.4703	0.000	7.7649	7.7652	7.649
4	Random Mapping	90.3511	0.0000	6.9331	7.999	6.9331

4.3. Analysis of the Encryption and Decryption algorithms implemented on Sound

Table 2. Comparison of encryption and decryption algorithms for sound



Sr. no.	Algorithm	Time taken for implementation (seconds)	Mean Square Error (MSE)	Entropy of the sound		
				Original	Encrypted	Decrypted
1	Block Scrambling	3.3588	0.00	3.3454	3.3454	3.3454
2	Encryption using Random Sequences	3.526	$7.55*10^{-10}$	3.4044	1.7041	3.4044
3	Key based Intensity Manipulation	14.6687	$2.469*10^{-33}$	3.0639	0.9795	3.0639
4	Random Mapping	65.2935	$1.89*10^{-33}$	1.2933	4.7959	1.2933

5. Results

To give an idea of how encryption is done image and sound files are taken and the encryption algorithms are applied on them. The original content , the encrypted content and the decrypted content has been shown along with few other representations like a histogram for and image and an audio plot and an histogram for an audio.

5.1 Image Encryption and Decryption

5.1.1. Block Scrambling algorithm:-

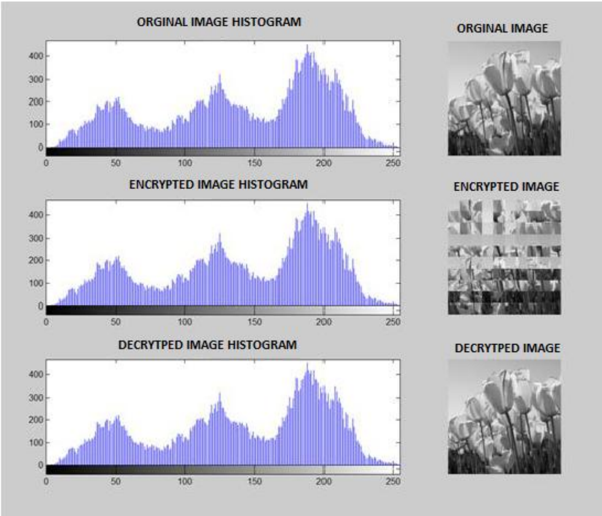


Figure 7. Output of Image Encryption and Decryption using Block Scrambling Algorithm

5.1.2. Random Sequence generation algorithm

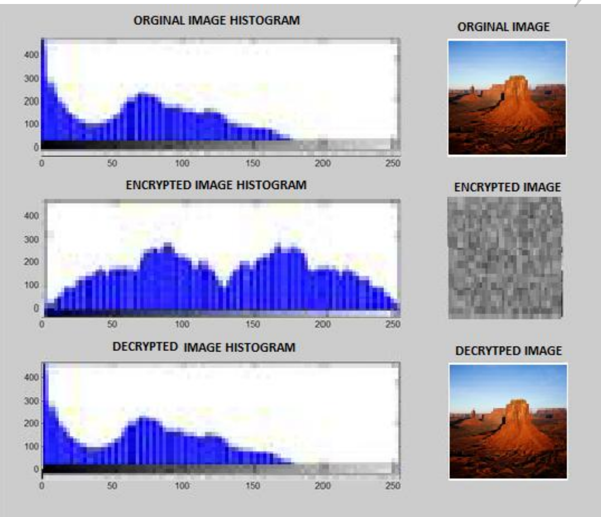


Figure 8. Output of Image Encryption and Decryption using Random Sequence Generation Algorithm

5.1.3. Key based pixel manipulation algorithm

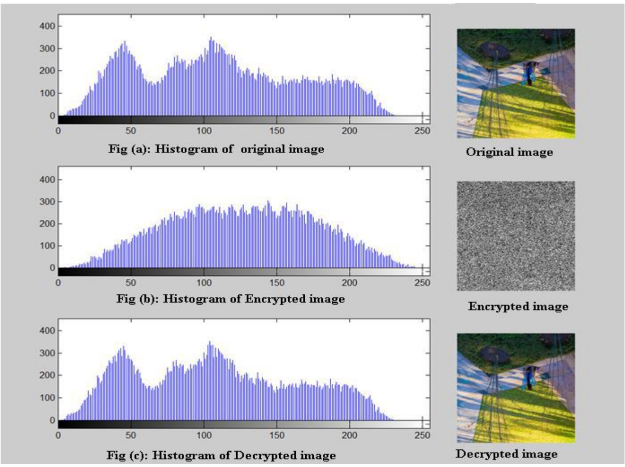


Fig (a): Histogram of original image

Fig (b): Histogram of Encrypted image

Fig (c): Histogram of Decrypted image

Figure 9. Output of Audio Encryption and Decryption using Key based Pixel Manipulation Algorithm

5.1.4. Random mapping algorithm

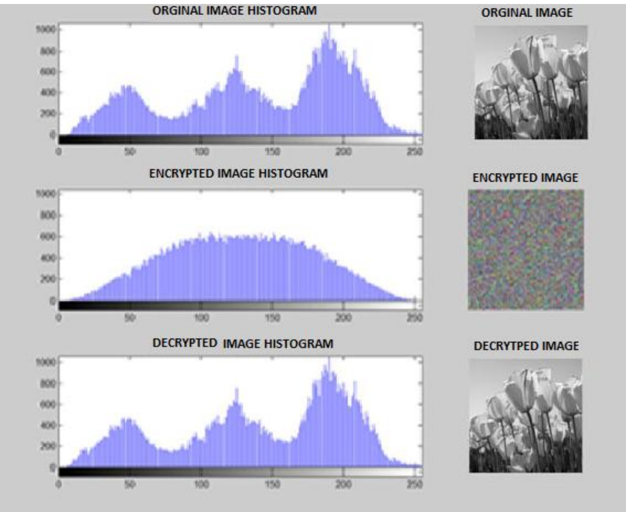


Figure 10. Output of Image Encryption and Decryption using Random Mapping Algorithm

5.2. Audio Encryption and Decryption

5.2.1. Block Scrambling algorithm

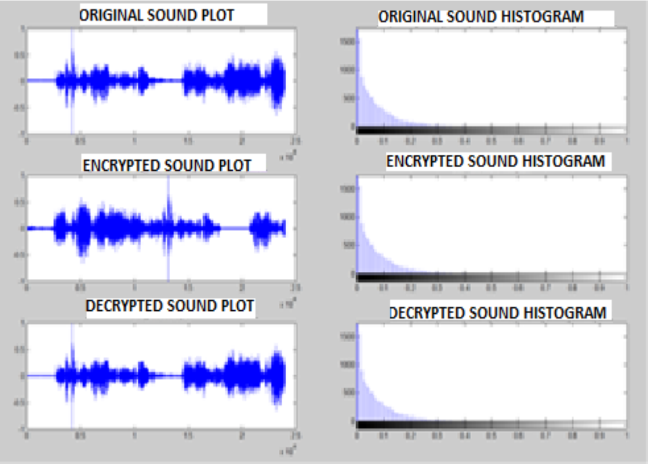


Figure 11. Output of Audio Encryption and Decryption using Block Scrambling Algorithm

5.2.2. Random Sequence generation algorithm

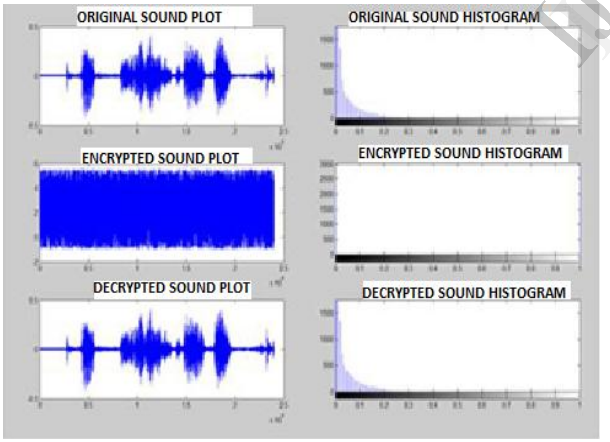


Figure 12. Output of Audio Encryption and Decryption using Random Sequence Generation Algorithm

5.2.3. Key based pixel manipulation algorithm

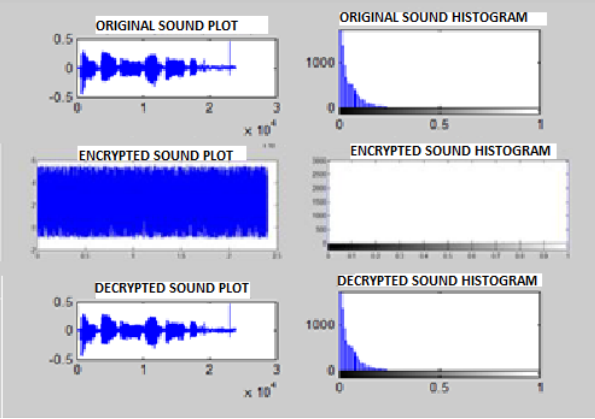


Figure 13. Output of Audio Encryption and Decryption using Key based Pixel Manipulation Algorithm

5.2.4. Random mapping algorithm

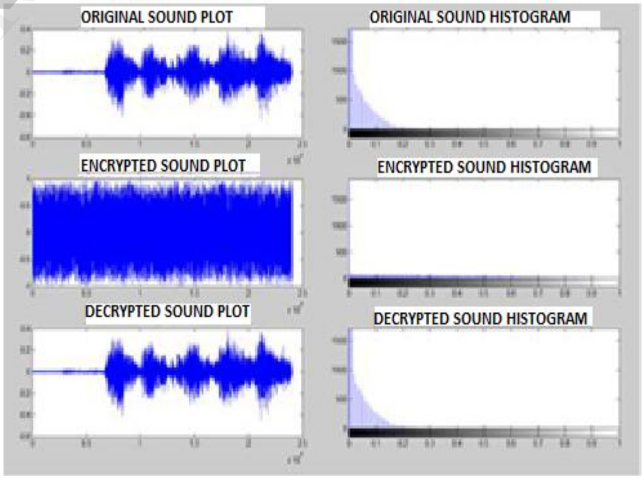


Figure 14. Output of Audio Encryption and Decryption using Random Mapping Algorithm

5.3 Video Encryption and Decryption

Since a video may contains frames and also the associated sound, only the encryption of the first extracted frame is shown below to give an idea about how video encryption is done.

In the below video a person counts down from 4 to 0 where the person counting 4 has been captured in the first frame as shown below:-

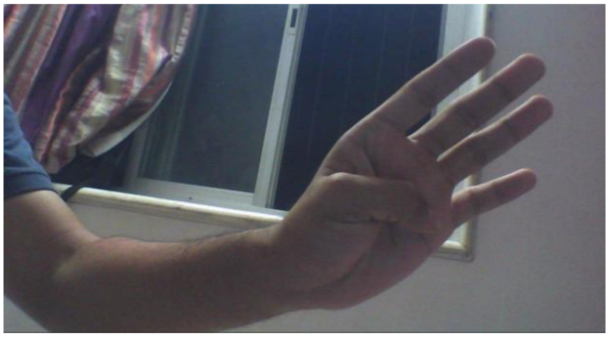


Figure 15. Original first frame of video

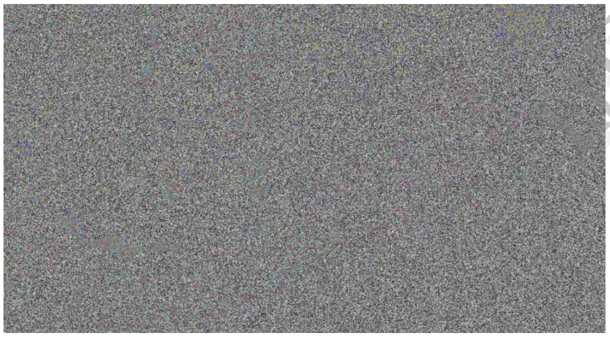


Figure 16. Encryption of first frame

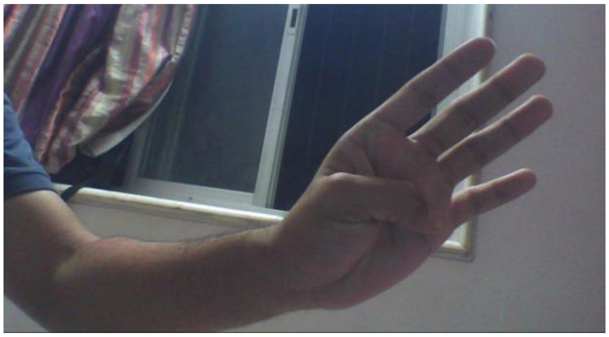


Figure 17. Decrypted first frame of video

6.Conclusions

This paper includes the study of existing encryption/decryption algorithms and has extended their principles to encrypt/decrypt image and audio files. Four different algorithms have been implemented and analyzed based on MSE, time taken and entropy of original, encrypted and decrypted content. Encryption of video also has been studied on a very basic level. These algorithms implemented for image/audio/video still have further scope of improvement in terms of security, error and time taken.

7.References

[1] Ian T. Young, Jan J. Gerbrands and Lucas J. van Vliet, “Fundamentals of Image Processing”, Printed in *The Netherlands* at the Delft University of Technology, Version 2.2,pages4-5

[2] Ratinder Kaur and V. K. Banga, “Image Security using Encryption based Algorithm*”* in *International Conference on Trends in Electrical, Electronics and Power Engineering Jul 15-16* (ICTEEP’2012), *Singapore,* pages 1-2

[3]Mani Roja,Sudhir Sawarkar “MySQL and MATLAB Interfacing for Biometric Template Protection with Encryption” in *International Journal of Computer Technology and Applications, Vol 3 (1),114-118, JAN-FEB 2012,page 3*

[4] Mani Roja, Sudhir Sawarkar “Biometric Database Protection using Public Key Cryptography” in *International Journal of Computer Science and Network Security, VOL.13 No.5, May 2013,page 5*

[5] Ji Won Yoona, Hyoungshick Kim, “An image encryption scheme with a pseudorandom permutation based on chaotic maps”, *Elsevier, 2010,* pages 2-5

[6] R.Gnanajeyaraman, K.Prasadh, Dr.Ramar, “Audio encryption using higher dimensional
Chaotic map”, *International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009,* pages 104-106.

[7] Z. Chen, Z. Xiong, and L. Tang. A novel scrambling scheme for digital video encryption. *In Proc. of Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, 2006, pages 997–1006.

[8] Deniz Erdogmus and Jose C. Principe, “Comparison of entropy and mean square error criteria in adaptive system training using higher order statistics”, Computational Neuro-Engineering Laboratory Department of Electrical and Computer Engineering University of Florida, Gainesville, FL 32611.