Effective Data Hiding for Image Using Code-Word Separable Reversible Data Hiding Model

[1]Naveenkumar A,[2]Rahunathan L
[1]Final Year PG Scholar, [2] Assistant Professor
[1]Master of Computer Applications,[2]Master of Computer Applications
[1]Kongu Engineering College, Perundurai, Tamil Nadu,India,[2]Kongu Engineering College, Perundurai, TamilNadu,India.

ABSTRACT

In recent years, signal processing in the encrypted domain has attracted considerable research interest. As an effective and popular means for privacy protection, encryption converts the ordinary signal into unintelligible data, so that the traditional signal processing usually takes place before encryption or after decryption. However, in some scenarios that a content owner does not trust the processing service provider, the ability to manipulate the encrypted data when keeping the plain content unrevealed is desired. For instance, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource. This project proposes a novel scheme for separable reversible data hiding in encrypted images. In the first phase, a content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider may replace the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data.With an encrypted image containing additional data, if a receiver has the data-hiding key, he can extract the additional data though he does not know the image content. If the receiver has the encryption key, he can decrypt the received data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.

1 INTRODUCTION

As an effective and popular means for privacy protection, encryption converts the ordinary signal into unintelligible data, so that the traditional signal processing usually takes place before encryption or after decryption. However, in some scenarios that a content owner does not trust the processing service provider, the ability to manipulate the encrypted data when keeping the plain content unrevealed is desired. For instance, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource.

The source is first compressed to its entropy rate using a standard source code. Then, the compressed source is encrypted using one of the many widely available encryption technologies. At the receiver, decryption is performed first, followed by decompression. Compression of encrypted data has attracted considerable research interest .The traditional way of securely and efficiently transmitting redundant data is to first compress the data to reduce the redundancy, and then to encrypt the compressed data to mask its meaning. At the receiver side, the decryption and decompression operations are orderly performed to recover the original data. However, in some application scenarios, a sender needs to transmit some data to a receiver and hopes to keep the information confidential to a network operator in provides the channel resource for the transmission.

That means the sender should encrypt the original data and the network provider may tend to compress the encrypted data without any knowledge of the cryptographic key and the original data. At receiver side, a decoder integrating decompression and decryption functions will be used to reconstruct the original data.

A. Types Of Encryption

The following list encryption methods are:

a) Hashing Encryption
b) Symmetric Encryption
c) Asymmetric Encryption

a) Hashing Encryption

The first encryption method, called hashing, creates a unique, fixed-length signature for a message or data set. Hashes are created with hash function, and people commonly use them to compare sets of data. Since a hash is unique to a specific message, even minor changes to that message result in a dramatically different hash, there by alerting a user to potential tampering.

b) Symmetric Encryption

Symmetric cryptography, also called private-key cryptography, is one of the oldest and most secure encryption methods. The term "private key" comes from the fact that the key used to encrypt and decrypt data must remain secure because anyone with access to it can read the coded messages. A sender encodes a message into cipher text using a key, and the receiver uses the same key to decode.

c) Asymmetric Encryption

Asymmetric or public key cryptography is potentially more secure than symmetric methods of encryption. This type of cryptography uses two keys, a "private" key and a "public key" to perform encryption and decryption. The use of two keys overcomes a major weakness in symmetric key cryptography, since a single key does not need to be securely managed among multiple users. In asymmetric cryptography, a public key is freely

available to everyone and used to encrypt messages before sending them. A different, private key remains with the receiver of cipher text messages, who uses it to decrypt them. Algorithms that use public key encryption methods include RSA and Differ-Hellman.

To encode a message and decode an encrypted message, one needs the proper encryption key or keys. The encryption key is the table or formula that defines which character in the data translates to which encoded character. Here, encryption keys fall into two categories: public and private key encryption.

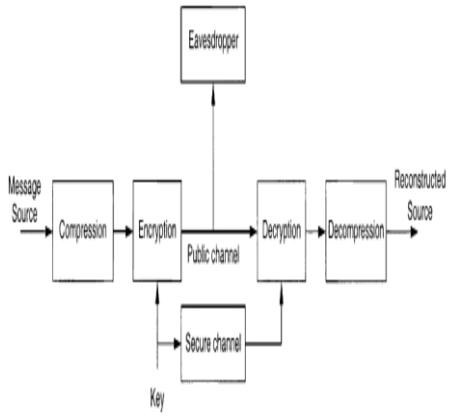

Figure 1.1.Encryption and Decryption

B. Data Hiding Applications

- Covert communication using images (secret message is hidden in a carrier image)
- Ownership of digital images, authentication, copyright
- Data integrity, fraud detection, self-correcting images
- Adding captions to images, additional information, such as subtitles, to video, embedding subtitles or audio tracks to video (video-in-video)
- Intelligent browsers, automatic copyright information, viewing a movie in a given rated version.
- Copy control (secondary protection for DVD)

C. Reversible Data Hiding

The reversible data hiding in encrypted image is investigated . Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain. But, in some applications, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. And it is also hopeful that the original content should be recovered without any error after image decryption and message extraction at receiver side.

A content owner encrypts the original image using an encryption key, and a data-hider can embed additional data into the encrypted image using a data-hiding key though he does not know the original content. With an encrypted image containing additional data, a receiver may first decrypt it according to the encryption key, and then extract the embedded data and recover the original image according to the data-hiding key. In the scheme, the data extraction is not separable from the content decryption.

In other words, the additional data must be extracted from the decrypted image, so that the principal content of original image is revealed before data extraction, and, if someone has the data-hiding key but not the encryption key, he cannot extract any information from the encrypted image containing additional data.

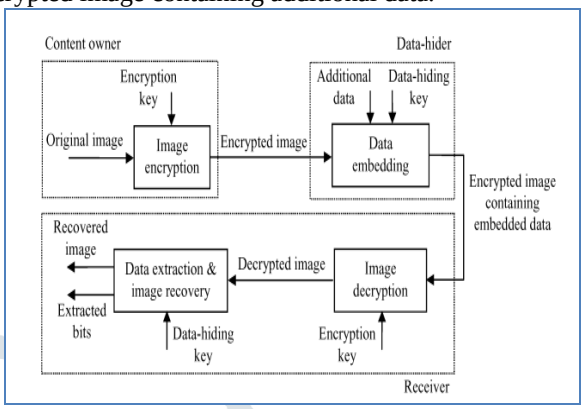

Figure 1.2.Reversible Data Hiding

II. RELATED WORKS

Tiziano Bianchi and Alessandro Piva [1] describe a Signal processing tools working directly on encrypted data could provide an efficient solution to application scenarios where sensitive signals must be protected from an untrusted processing device. In this paper, they considered the data expansion required to pass from the plaintext to the encrypted representation of signals, due to the use of cryptosystems operating on very large algebraic structures. A general composite signal representation allowing packing together a number of signal samples, and processing them as a unique sample is proposed. The proposed representation permits to speed up linear operations on encrypted signals via parallel processing and to reduce the size of the encrypted signal. A case study - 1D linear filtering - shows the merits of the proposed representation and provides some insights regarding the signal processing algorithms more suited to work on the composite representation.

The possibility of processing encrypted signals directly in the encrypted domain (hereafter referred to as s.p.e.d., standing for signal processing in the encrypted domain) is receiving an increasing attention as a way to satisfy the security requirements stemming from applications wherein valuable or sensible signals have to be processed by a non-trusted party. The list of applications that would benefit from the availability of s.p.e.d. tools is virtually endless, including: access to a database containing encrypted data or signals, database access by means of encrypted queries, remote processing of private data, like medical recordings or biometric signals, by non-trusted .

Nasir Memon and Ping Wah Wong describe [2] digital watermarks have recently been proposed for the purposes of copy protection and copy deterrence for multimedia content. In copy deterrence, a content owner (seller) inserts a unique watermark into a copy of the content before it is sold to a buyer. If the buyer sells

unauthorized copies of the watermarked content, then these copies can be traced to the unlawful reseller (original buyer) using a watermark detection algorithm. One problem with such an approach is that the original buyer whose watermark has been found on unauthorized copies can claim that the unauthorized copy was created or caused (for example, by a security breach) by the original seller. In this paper they proposed an interactive buyer-seller protocol for invisible watermarking in which the seller does not get to know the exact watermarked copy that the buyer receives. Hence the seller cannot create copies of the original content containing the buyer's watermark. In cases where the seller finds an unauthorized copy, the seller can identify the buyer from a watermark in the unauthorized copy, and furthermore the seller can prove this fact to a third party using a dispute resolution protocol. This prevents the buyer from claiming that an unauthorized copy may have originated from the seller.

Mina Deng and Tiziano Bianchi [3] describe buyer-seller watermarking protocols integrate watermarking techniques with cryptography, for copyright protection, piracy tracing, and privacy protection. In this paper, they proposed an efficient buyer-seller watermarking protocol based on homomorphism public-key cryptosystem and composite signal representation in the encrypted domain.

A recently proposed composite signal representation allows us to reduce both the computational overhead and the large communication bandwidth which are due to the use of homomorphic public-key encryption schemes. Both complexity analysis and simulation results confirm the efficiency of the proposed solution, suggesting that this technique can be successfully used in practical, applications.

Deepa Kundur and Kannan Karthi [4] provides a tutorial and survey of digital fingerprinting and video scrambling algorithms based on partial encryption. Necessary design tradeoffs for algorithm development are highlighted for multicast communication environments. They also proposed a novel architecture for joint fingerprinting and decryption that holds promise for a better compromise between practicality and security for emerging digital rights management applications.

This paper investigates multimedia security algorithms that enable digital rights management (DRM) in resource constrained communication applications. Their focus is on the video-on-demand (VoD) business model, in which subscribers to a content-providing service request and receive video information at scheduled intervals. They considered situations in which on the order of hundreds or even thousands of users may wish near-simultaneous access to the same video content. Thus, for superior scalability the network service provider must transmit the content by making use of a multicast distribution model.They focused on the problems of video fingerprinting and encryption. Fingerprinting, which was first introduced by Wagner in 1983, is the process of embedding a distinct set of marks into a given host signal to produce a set of fingerprinted signals that each “appear” identical for use, but have a slightly different bit representation from one another.

These differences can ideally be exploited in order to keep track of a particular copy of the fingerprinted signal. The marks, also called the fingerprint payload, are usually embedded through the process of robust digital watermarking. In digital watermarking, subtle changes are imposed on the host signal such that the perceptual content of the host remains the same, but the resulting composite watermarked signal can be passed through a detection algorithm that reliably extracts the embedded payload.

In contrast, video encryption has the goal of obscuring the perceptual quality of the host signal such that access to the content is denied. In comparison to traditional cryptographic algorithms, those for video may often be “lightweight” in order to accommodate computational complexity restrictions; the term “video scrambling” is often used to refer to such processes. The main objective of fingerprinting and encryption in a DRM context is to protect video content from a set of attacks applied by one or more attackers. They defined an attacker as any individual who attempts to use a given piece of content beyond the terms, if any, negotiated with the content provider. Common attacks on video data include illegal access and tampering.

Nasir Memon and Ping Wah Wong [5] describe a digital watermarks have recently been proposed for the purposes of copy protection and copy deterrence for multimedia content. In copy deterrence, a content owner (seller) inserts a unique watermark into a copy of the content before it is sold to a buyer. If the buyer sells unauthorized copies of the watermarked content, then these copies can b e traced to the unlawful reseller (original buyer) using a watermark detection algorithm. One problem with such an approach is that the original buyer whose watermark has been found on unauthorized copies can claim that the unauthorized copy was created or caused (for example, by a security breach) by the original seller.

In this paper we propose an interactive buyer-seller protocol for invisible watermarking in which the seller does not get to know the exact watermarked copy that the buyer receives. Hence the seller cannot create copies of the original content containing the buyer's watermark. In cases where the seller sends an unauthorized copy, the seller can identify the buyer from a watermark in the unauthorized copy, and furthermore the seller can prove this fact to a third party using a dispute resolution proto col. This prevents the buyer from claiming that an unauthorized copy may have originated from the seller.

The original image is encrypted using an encryption key and the additional data are embedded into the encrypted image using a data-hiding key. With an encrypted image containing additional data, if the receiver has only the data-hiding key, he can extract the additional data though he does not know the image content. If he has only the encryption key, he can de-crypt the received data to obtain an image similar to the original one, but cannot extract the embedded additional data. If the receiver has both the data-hiding key and the encryption key, he can extract the additional data and recover the original image without any error when the amount of additional data is not too large.

The proposed scheme is made up of image encryption, data embed-ding and data-extraction/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider compresses the least significant bits (LSB) of the encrypted image using a data-

hiding key to create a sparse space to accommodate the additional data. At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted image containing additional data according to the data-hiding key. Since the data embedding only affects the LSB, a decryption with the encryption key can result in an image similar to the original version. When using both of the encryption and data-hiding keys, the embedded additional data can be successfully extracted and the original image can be perfectly recovered by exploiting the spatial correlation in natural image

III. METHODOLOGY

Now a day the data security and integrity are the two challenging areas for research. There are numerous research is progressing on the field like internet security, steganography, cryptography Images used in military, medical science are the media in which we found certain distortion sometime which is un-acceptable. Hence for data hiding we have a technique using which we can extract data correctly and after that original cover content can be perfectly recovered.

This technique is also known as reversible data hiding or it is also named as lossless, distortion free, or invertible data hiding technique. The author presented an exclusive reversible (lossless) data hiding technique which supports the exact recovery of the original cover medium with the extraction of the embedded information. And the process of this recovery with lossless data is nothing but the reversible data hiding.

Generally the well-known LSB (least significant bit) method is used as the data embedding method. Reversible data hiding is a technique that is mainly used for the authentication of data like images, videos, electronic documents etc. Mainly the reversible data hiding is applicable for in IPR (Intellectual Property Rights) protection, authentication, and conditional access. In some application scenarios it is essential to provide security, authentication and privacy while communication or transferring data. To hide the data or to provide the data security we need some new approach in communication.

A. IMAGE FILE SELECTION

In this module, the image file selection is carried out open file dialog control and the path is displayed in text box and the image is displayed in picture box control. Then the image data is saved into 'Images' table. During saving, the image data, width and height, image type (grayscale or RGB) are the information saved.

B. PSEUDO RANDOM BIT INPUT

In this module, the pseudo-random bits are keyed in which is used during image encryption. The details are saved into 'RandomBits' table.

C. IMAGE ENCYPTION

In this module, the image file is selected, image bits and pseudo-random bits are applied with X-or operation and the result bit sequence is replaced for image bits. So the image is encrypted.

D. TEXT DATA INPUT

In this module, the data to be hiding is keyed in and saved into 'TextData' table. This is the text being embed into the encrypted image.

F. DATA EMBEDDING

In this module, the record in which the data to be hiding is selected (last record of the 'TextData' table). The data is perturbed such that random number of characters is added inside the characters in the given text. The text data is encrypted using TripleDES encryption and the bit sequences are taken for hiding. Then the encrypted image (last record of the 'Images' table) is taken and the bit locations (where the text data bits are to be replaced) are selected. The text data is converted into bits and replaced in the image. This is the output image being sent to the receiver.

G. DATA EXTRACTION

In this module, the encrypted image sent by the sender is received and the reverse operations are carried out to fetch the text data.

H. IMAGE DECRYPTION

In this module, the encrypted image (after the text data is fetched) is applied with reverse operations to get the original source image.

There are two kinds of reversible data hiding techniques (according to key distribution) separable reversible data hiding technique and non-reversible data hiding technique (also called as reversible data hiding). Non-separable technique reversible data hiding scheme, a content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image and then a data-hider embeds additional data into the encrypted image using a data-hiding key. Having an encrypted image containing additional data a receiver firstly decrypt it using the encryption key and can further extract the embedded data.

Thus in non-separable technique it is compulsory to have both the keys i.e. encryption key and the data-hiding key for retrieving data. But in separable technique it is not compulsory to have both the keys for retrieving data here if the receiver has a data hiding key only then he can extract the embedded or hidden data from the encrypted image containing additional data. Here they are separating two activities i.e. Cover image decryption and pay load data extraction. This paper is stating one of the types of reversible data hiding method i.e. separable reversible data hiding method which is consists of three main procedures

- Image encryption
- Data embedding
- Data extraction/image recovery.

As shown in **Figure 3.1** in separable scheme the content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then using a data-hiding key the data-hider compresses the least significant bits (LSB) of the encrypted image to create some space to accommodate the additional data. At the receiver side the data embedded can be easily retrieved from the encrypted image containing additional data according to

the data-hiding key. As the data embedding only affects the LSB a decryption with the encryption key may result in an image similar to the original version.

Here at the receiver side there exists three cases as shown in figure 2.With an encrypted image containing additional data which is hidden case one is when the receiver has only the data- hiding key, he is able to extract the additional data even if he does not know the image content. Case two is if he has only the encryption key, user decrypt the received data i.e. encrypted image to obtain an image similar to the original cover media, but cannot extract the embedded additional data. Case three is if the receiver has both the keys i.e. data-hiding key and the encryption key, he can extract the additional data and recovers the original image without any error. The proposed scheme is describing the method in which the concept of separable reversible data hiding is executed using RGB-LSB method

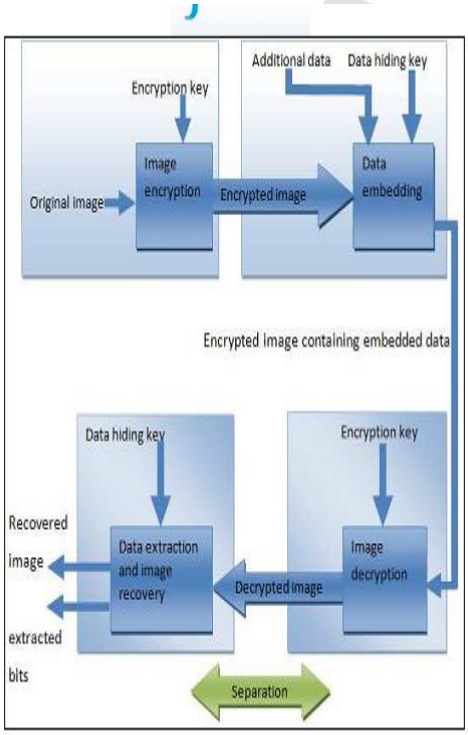

Figure 3.1 Separable reversible data hiding in encrypted image

This part of the literature expose on implementing separable reversible data hiding schemes in encrypted image based on RGB-LSB method. The existing approach of using separable reversible data hiding involve simple LSB method in which the least significant bits of the pixel position value(representing one pixel as one byte or eight bits) is used for creating space.

The space creation is for making room for external/additional confidential information which is to be hide/embedded. The author xinpeng zang uses this simple LSB based compression .But this technique could not hide enough data the scheme is having limitation that at the receiver side receiver can extract the additional data and recover the original content without any error when the amount of additional data is not too large. Thus the scheme is not suitable if anyone wants to embed the more data. So to overcome this problem we need a new method or new technique which can handle enough data to embed.

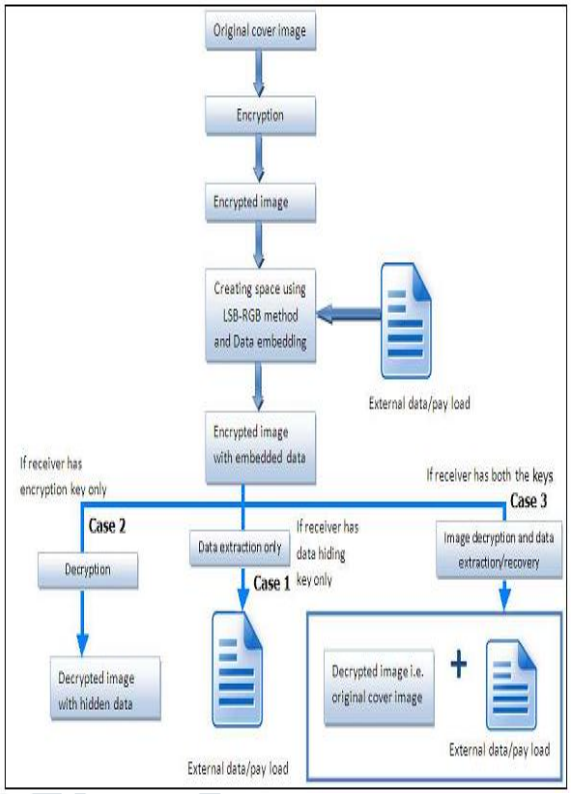

Figure 3.2 Separable reversible data hiding with three cases.

In general, in LSB methods, hidden information is stored into a specific position of LSB of image. In our paper, hidden information is stored into different position of LSB of image i.e. LSB of red color value, LSB of green color value and blue color value. Different positions of LSB of image means the LSB of RGB.The true color image is having five parameters two parameters represents the pixel position and three parameter represents red, green and blue color values. Actually these three colors are represented as three matrices red, green and blue as shown in figure.3. As a result, it is difficult to extract the hidden information knowing the retrieval methods. The proposed method results in LSB based image steganography using secret key which provides good security issue than general LSB based image steganography methods.

IV. EXPERIMENTAL RESULT

The following **Table 4.1** describes experimental result for proposed system algorithms. The table input text size for DES and 3DES algorithms for encryption execution time details are shown

Table1 4.1: Execution Time (Milliseconds) of Encryption of different Text data size (DES & 3DES)

Input Text Size (Bytes)	DES	3DES
75	21	57
96	32	55
112	54	81
286	97	173

359	188	198
600	198	202
951	391	327
5345	1399	1149
Throughput (MB/sec)	**3.01**	**2.8**

The following **Fig 4.1** describes experimental result for proposed system algorithms. The table input text size for DES and 3DES algorithms for encryption execution time details are shown

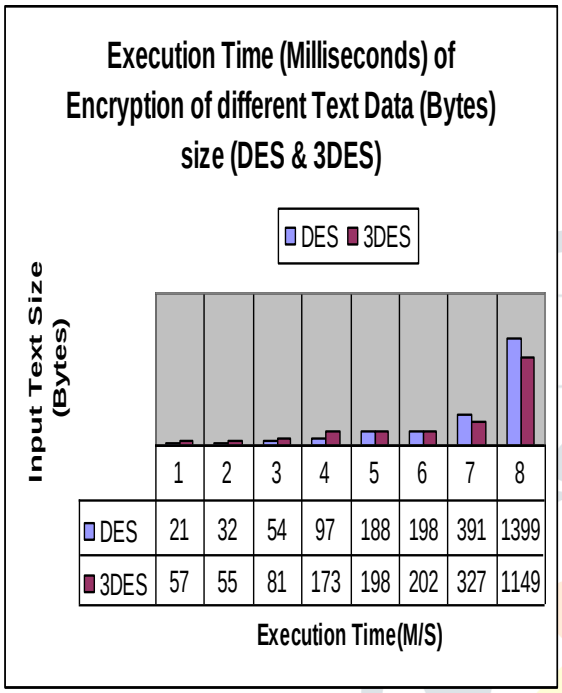

Fig 4.1: Execution Time (Milliseconds) of Encryption of Different Text Data Size (DES and 3DES)

The following **Table 6.2** describes experimental result for proposed system algorithms. The table input text size for DES and 3DES algorithms for decryption execution time details are shown

Table2: Execution Time (Milliseconds) of Decryption of Different Text data size (DES & 3DES)

Input Text Size (Bytes)	**DES**	**3DES**
75	25	62
96	34	59
112	56	84
286	102	176
359	192	203
600	205	210
951	402	333
5345	1401	1152
Throughput (MB/sec)	**3.03**	**2.84**

The following **Fig 6.2** describes experimental result for proposed system algorithms. The table input text size for DES and 3DES algorithms for decryption execution time details are shown

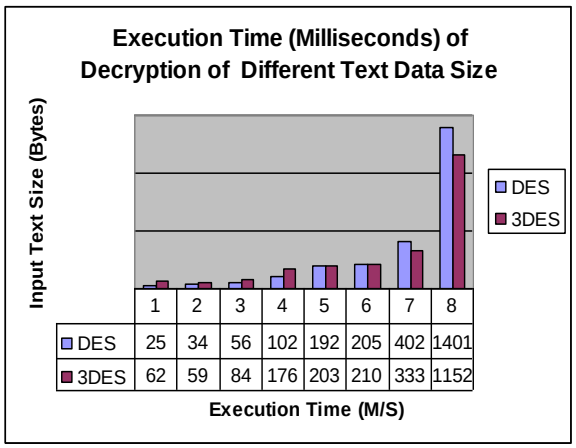

Fig 6.2: Execution Time (Milliseconds) of Decryption of Different Text Data Size (DES and 3DES)

- Any one or both the source image and the embedded data can be retrieved from the output image.
- The proposed system is suitable for RGB images also.
- Before embedding the data, the data is perturbed so that the security is more.
- 4 Since 3DES algorithm is used, even small image can be used as carrier image to embed more data.
- The size of the image data is not increased during the embedding process.
- More amounts of noise are added in all part of the image so that no part of the original image is visualized.
- The receiver having the data-hiding key can successfully extract the embedded data, even he is not interested in any information about the original image content.
- Original image content can be recovered without any loss of appearance.
- A comprehensive combination of image encryption and data hiding compatible can be achieved in future if compression type is lossy.

REFERENCES

[1] 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.

[2] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of en-crypted grayscale images," IEEE Trans. Image Process. , vol. 19, no. 4, pp. 1097–1102, Apr. 2010.

[3] X. Zhang, "Lossy compression and iterative reconstruction for en-crypted image," IEEE Trans. Inform. Forensics Security , vol. 6, no. 1, pp. 53–58, Feb. 2011.

[4] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the dis-crete Fourier transform in the

encrypted domain," IEEE Trans. Inform. Forensics Security , vol. 4, no. 1, pp. 86–97, Feb. 2009.
[5] T. Bianchi, A. Piva, and M. Barni, "Composite signal representation forfast and storage-efficient processing of encrypted signals," IEEE Trans.Inform. Forensics Security , vol. 5, no. 1, pp. 180–187, Feb. 2010.
[6] W. J. Lu, A. Varna, and M. Wu, "Secure video processing: Problems andchallenges," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Prague, Czech Republic, May 2011, pp. 5856–5859.
[7] B. Zhao, W. D. Kou, and H. Li, "Effective watermarking scheme in the encrypted domain for buyer-seller watermarking protocol," Inf. Sci., vol. 180, no. 23, pp. 4672–4684, 2010.
[8] P. J. Zheng and J. W. Huang, "Walsh-Hadamard transform in the homo- morphic encrypted domain and its application in image watermarking," in Proc. 14th Inf. Hiding Conf., Berkeley, CA, USA, 2012, pp. 1–15.
[9] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images," Proc. SPIE, vol. 6819, pp. 68191E-1–68191E-9, Jan. 2008.
[10] X. P. Zhang, "Reversible data hiding in encrypted image," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.
[11] W. Hong, T. S. Chen, and H. Y. Wu, "An improved reversible data hiding in encrypted images using side match," IEEE Signal Process. Lett., vol. 19, no. 4, pp. 199–202, Apr. 2012.
[12] X. P. Zhang, "Separable reversible data hiding in encrypted image," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.
[13] K. D. Ma, W. M. Zhang, X. F. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, pp. 553–562, Mar. 2013.
[14] A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, "Robust watermarking of compressed and encrypted JPEG2000 images," IEEE Trans. Multimedia, vol. 14, no. 3, pp. 703–716, Jun. 2012.
[15] S. G. Lian, Z. X. Liu, and Z. Ren, "Commutative encryption and watermarking in video compression," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 6, pp. 774–778, Jun. 2007.
[16] S. W. Park and S. U. Shin, "Combined scheme of encryption and watermarking in H.264/scalable video coding (SVC)," New Directions Intell. Interact. Multimedia, vol. 142, no. 1, pp. 351–361, 2008.
[17] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.
[18] S. G. Lian, Z. X. Liu, Z. Ren, and H. L. Wang, "Secure advanced video coding based on selective encryption algorithms," IEEE Trans. Consumer Electron., vol. 52, no. 2, pp. 621–629, May 2006.
[19] Z. Shahid, M. Chaumont, and W. Puech, "Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames," IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 5, pp. 565–576, May 2011.
[20] M. N. Asghar and M. Ghanbari, "An efficient security system for CABAC bin-strings of H.264/SVC," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 3, pp. 425–437, Mar. 2013.
[21] T. Stutz and A. Uhl, "A survey of H.264 AVC/SVC encryp- tion," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 3, pp. 325–339, Mar. 2012.
[22] Advanced Video Coding for Generic Audiovisual Services, ITU, Geneva, Switzerland, Mar. 200**5**