Secure and Efficient Method for Reversible Image Data Hiding over Encrypted Domain

Anisha Jose[1], Mary Mareena[2], Saritha K[3]

[1]*M.Tech Scholar, Dept. of CSE, NCERC, Kerala*

[2]*Assistant Professor, Dept. of CSE, NCERC, Kerala*

[3]*Assistant Professor, Dept. of CSE, NCERC, Kerala*

Abstract*— Reversible steganography, also called reversible data hiding in digital images has been studied extensively in recent years. Reversible Image Data Hiding (RIDH) is a category of data hiding technique that ensures perfect reconstruction of cover image upon the extraction of the embedded message. The property of reversibility means that the original image can be recovered completely after the embedded bits are extracted. The main focus is to apply reversible data hiding algorithms on encrypted images by wishing to remove the embedded data before the image decryption. This paper proposes a novel Reversible Image Data Hiding (RIDH) scheme over encrypted domain. A public key modulation mechanism is applied to achieve data embedding, in which access to the secret encryption key is not needed. A powerful two-class SVM classifier is designed at the decoder side to distinguish encrypted and non-encrypted image blocks, which allows to decode the embedded message and the original image signal jointly.

Keywords : Reversible image data hiding, public key modulation, SVM classifier.

I. INTRODUCTION

The amount of digital images has increased rapidly on the Internet. Image security becomes important for many applications especially in confidential transmission, video surveillance, military and medical applications. The protection of multi-media data can be done with encryption or data hiding algorithms. Data hiding is a technique to embed additional data into digital multimedia by altering the cover signals slightly[1]. When the data hiding is performed in a reversible manner, the original cover content can be restored perfectly after data extraction at receiver side. Reversible data hiding method embeds a piece of information into the host images and generate the marked one. After extracting the embedded data, the original image can be exactly recovered. Reversible data hiding can be used in many fields such as medical, military, and legal applications, which do not allow modifications in the digital representation of the cover image due to confidentiality issues.

The reversible data hiding methods can be classified into three types[2]: the difference expansion methods, histogram modification methods, and lossless compression based methods. In the difference expansion methods, the differences between two adjacent pixels are doubled which generates a new Least Significant Bit (LSB) plane for embedding the additional information. The histogram modification methods shift the histogram of cover data from its peak point towards its zero points, and utilize the cover data at the peak point of histogram to carry the additional data. The lossless compression based methods make use of statistical redundancy of the host media by performing lossless compression to spare space for accommodating the additional.

Most data hiding methods embed messages into the cover media to generate marked image by modifying only the LSB of the cover image. The embedding process will usually have low embedding capacity and also introduce permanent distortion of original image. That is, the original cover can never be reconstructed image from the marked cover. In the medical imagery, military imagery, and law forensics, these type of degradation in original cover is not allowed. Therefore a special kind of data hiding method is needed, which is Reversible Data Hiding (RDH) or lossless data hiding. The original cover can be restored reversibly after the extraction of embedded message.

Fig.1 shows the overall block diagram of reversible data hiding process. At the sender side, a secret data which is shared between the sender and receiver, is embedded into the host image. This image with secret data is now transmitted to the receiver side, where the data extraction is taken place and the host image is reconstructed.

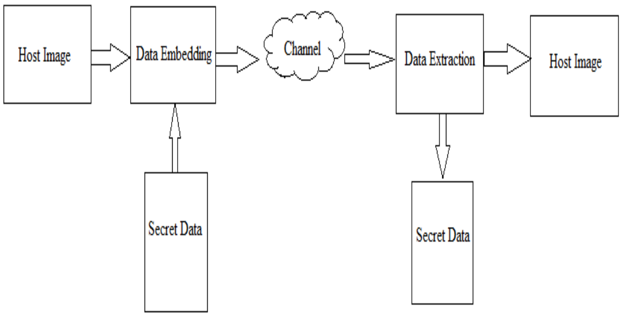

Fig. 1 Reversible Data Hiding Process

Most of works on reversible data hiding is applied on unencrypted domain. In some practical scenarios, a content owner encrypts the original images as unintelligible data for privacy protection. In the field of secure remote sensing and Cloud computing, the parties who process the image data are un-trusted[4]. All images will be encrypted before forwarding to an un-trusted third party for further processing so as to protect the privacy and security.

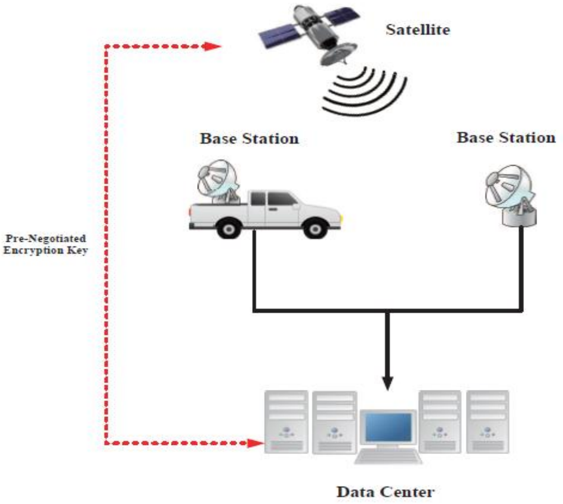

Fig. 2 Image data hiding in the scenario of secure remote sensing[3].

In secure remote sensing, the satellite images captured by on-board cameras, are encrypted and then sent to the base station, as illustrated in Fig 2. The base station embeds a confidential message, such as the base station ID, location information, time of arrival (TOA), local temperature, wind speed, etc., into the encrypted images. Now, the encrypted image which contains the additional message is transmitted to a data center over a public network for further investigation and storage. Any base station, for security reasons, has no privilege of accessing the secret encryption key pre-negotiated between the satellite and the data center. Therefore, it is clear that the message embedding operations have to be conducted entirely over the encrypted domain.

II. RELATED WORKS

Many of the traditional reversible data hiding approaches were based on unencrypted domain. Some recent attempts were made on embedding message bits into encrypted images. Mehmet Utku Celik *et. al.* [5] presents a lossless (reversible) data-embedding technique, which enables the recovery of original host signal after the extraction of embedded information. Here the work proposes a generalization of the well-known least significant bit (LSB) modification as the data-embedding method that introduces additional operating points on the capacity-distortion curve. It compresses a portions of the signal that are susceptible to embedding distortion and transmits these compressed descriptions as a part of the embedded payload. In this way, lossless recovery of the original is achieved. A prediction-based conditional entropy coder utilizes unaltered portions of the host signal as side-information which improves the compression efficiency. The lossless embedding step produces a watermarked signal in which the message data is embedded by taking the host signal and the message data as the input. The watermarked signal is used for data extraction and recovery process, to exactly extract the embedded data and to recover the original host signal. The LSB of each signal sample is replaced by a payload data bit. Two or more LSBs may be over written if additional capacity is required. During extraction, these bits are read in the same scanning order so that the payload data can be reconstructed. LSB modification is a simple and non robust embedding technique. It provides a high-embedding capacity and small bounded embedding distortion. However, the method is irreversible, i.e., it results in a permanent distortion of host signal when its lowest levels containing the residual signal are replaced with the watermark signal.

Work by Chuan Qin *et. al.* [2] proposes a prediction-based reversible steganographic scheme based on image inpainting. According to the distribution characteristics of the image content, reference pixels are chosen adaptively. Then, the image inpainting technique is introduced to generate a prediction image which has similar structural and geometric information as that of cover image. Finally, the histogram of the prediction error is shifted to embed the secret bits reversibly. The embedded secret bits can be extracted from the stego image correctly, since the same reference pixels can be exploited in the extraction procedure. Also, the cover image can be restored losslessly. A prediction process is conducted first to estimate the cover image pixels, and the prediction error, i.e., the difference between the cover image and the prediction result, is used to embed the secret data. The accuracy of the prediction result depends on choosing the reference pixels and how it is utilized for prediction. Here, the reference pixels are adaptively selected according to the distribution characteristics of the image. Fewer reference pixels are chosen in the smooth regions of the cover images, while more reference pixels are chosen in the complex regions. The PDE-based inpainting algorithm can effectively generate the prediction image that has the similar structural and geometric information as the cover image according to the chosen reference pixels.

In the paper proposed by William Puech *et. al.* [6], a reversible data hiding algorithm on encrypted images is applied which remove the embedded data before the image decryption. Reversible data hiding methods discussed so far are not applicable on encrypted images. In this paper, local standard deviation of the marked encrypted images is analyzed in order to remove the embedded data during the decryption step. If block encryption methods are applied to images, one can face three inconveniences. The first one is when there exist a homogeneous zones (regions with the same color), then all blocks in these zones are encrypted in same manner. The second problem is that block encryption methods are not robust to noise. The third problem is data integrity. However, the combination of encryption and data-hiding can solve these problems. The Advanced Encryption Standard (AES) algorithm consists of a set of processing steps repeated iterations called rounds which is dependent on the size of the key and the size of the data block. AES first perform an XORing of subkey with the block which is called the AddRoundKey step. Afterwards, the round operation is followed. Each regular round operation consist of four steps. In the SubBytes step, every byte of the block is replaced by a substitute in a substitution box (S-Box). The next is the ShiftRows step where the rows are shifted over different offsets in a circular manner. The next step is the MixColumns, where every column is multiplied with a matrix over the Gallois Field. Another AddRoundKey step is performed as the last step. In the proposed method, coding algorithm involves two steps which are the encryption and the data hiding step. For each block composed of n pixels, here apply the AES encryption algorithm by block. Same secret key is used for data encryption and data hiding. The decoding algorithm also involves two steps which are the message extraction and the decryption. The extraction of the message is just enough to read the bits of the pixels which are marked by using the secret key k and the same PRNG. But after the extraction, each marked cipher-text will remain marked. The problem is to decrypt the marked encrypted image. The removal of decryption is done by analyzing the local standard deviation during the decryption of the marked images.

The next paper proposed by Wien Hong *et. al.* [7], adopts a better scheme to measure the block smoothness. Then it uses the side-match scheme to decrease the error rate of extracted-bits. The evaluation of block smoothness favors a correct data extraction. The four borders of each block do not take part in the calculation of block smoothness. Therefore the correctness of data extraction may get decreased, especially when the block size is too small. Smoothness evaluation employs the summation of absolute of two neighboring pixels. Moreover, the message extraction and image recovery are performed starting from the noticeable change in smoothness to the least ones. It also adopts the side-match technique by concatenating the border of recovered blocks to the unrecovered blocks for the evaluation of block smoothness.

Xinpeng Zhang *et. al.* [8] proposes reversible data hiding scheme in encrypted images based on lossless compression of encrypted data. A stream cipher is used in the encryption phase to mask the original content. Then, a part of encrypted data in the cipher-text image is compressed using LDPC code, and then inserts the compressed data and the additional data into a part of encrypted data itself. Quality of decrypted image is satisfactory, since majority of the encrypted data are unchanged. A receiver who has the data-hiding key can extract the additional data and the compressed data successfully. The original image is encrypted by the content owner by using a stream cipher. Data hider may compress half of the 4th LSB of the encrypted image using LDPC code, even though he does not know the original content and the cryptographic key. Then insert the compressed data and the additional data into the encrypted image. The receiver can extract the additional data using the data-hiding key, also decrypt it using the cryptographic key to reconstruct the original version. If the receiver has both the data-hiding and cryptographic keys, he can further recover the original image without any errors.

III. PROPOSED SYSTEM

This work proposes a novel Reversible Image Data Hiding (RIDH) scheme[3] in encrypted domain. A public key modulation mechanism is performed to achieve data embedding, in which access of secret encryption key is not needed. A powerful two-class SVM classifier is used at the decoder side which is designed to distinguish encrypted and non-encrypted image blocks. This allows us to jointly decode the embedded message and as well as the original image signal. Compared with the state-of-the-arts, the proposed method provides higher embedding capacity. Also it ensures perfect reconstruction of original image and the embedded message.

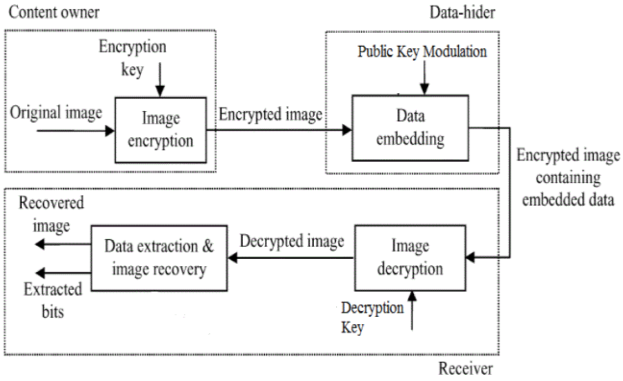

Fig. 3. RIDH scheme over encrypted domain

Fig. 3 gives the overall working of the proposed steganography scheme over encrypted domain. The sender side of the system is constituted by the content owner and the data-hider. The important operations at the sender site includes image encryption as well as data embedding. The secret message is received by only the intended receiver. The content owner first encrypts the original uncompressed host image, which acts as the cover image, using an encryption key. The key is randomly generated using a random number generator method. By the application of encryption key over the selected host image will eventually produce an encrypted image. The encrypted image is sent to the data-hider portion of the sender side. This is the site where the actual public key modulation mechanism takes place. The data-hider embeds the secret message into the encrypted image using the data-hiding key, which is generated as a result of the different combinations of public keys, irrespective of whether the data-hider knows the secret data or not. The encrypted image containing the embedded secret data is being sent to the intended receiver by the sender. With an encrypted image containing the additional secret message, the receiver first decrypts the image using the decryption key. The encryption/decryption keys will be known only by the sender and the intended receiver.

Further, in the data extraction and image recovery phase, the secret message can be extracted and the original image can be perfectly reconstructed from the decrypted version without any distortion.

A. Image Encryption

Here uses the conventional stream cipher applied in the standard format as the encryption algorithm. When the stream cipher is employed for this purpose, the encrypted image is generated as a function of original image and the key stream being generated by using a secret encryption key. The cipher text is generated by bitwise XORing of the plaintext and the key stream. When stream cipher is used, the encrypted image is generated by the function,

$$[[f\ ]] = Enc(f\ ,K) = f \oplus K \text{ ----- } (1)$$

where f and [[f]] denote the original and the encrypted images, respectively. Here, K is the key stream generated by using the secret encryption key.

B. Public Key Generation

The encrypted image [[f]] serves as the cover to hold the message to be hidden. First step is to divide encrypted cover image into a series of non-overlapping blocks of size M×N. Each block is designed in such a way to carry *n* bits of message. If the number of blocks in the image is B, the embedding capacity becomes *n* * B bits. To enable efficient embedding, here generates $S = 2^n$ binary public keys Q_0, Q_1, …… ,Q_{S-1}, each having length L = M × N × 8 bits. These public keys are selected prior to the message embedding, according to the criterion of maximizing the minimum Hamming distance[9] among all keys.

C. Data Encryption

Data to be hidden in the cover image is encrypted before the embedding to provide more security to the system. Data encryption is done using AES encryption method.

The Advanced Encryption Standard (AES) algorithm consists of a set of processing steps with repeated iterations called rounds which is dependent on the size of the key and the size of the data block. AES first perform an XORing of subkey with the block which is called the AddRoundKey step. Afterwards, the round operation is followed. Each regular round operation consist of four steps.

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

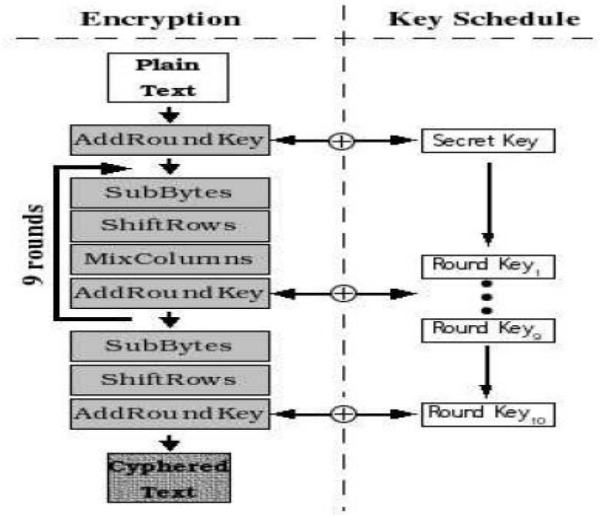

Fig. 4 AES encryption scheme

Fig.4 shows the AES encryption scheme. In the SubBytes step, each byte of the block is replaced by its substitute in a substitution box (S-Box). An S-box is a basic component of symmetric key algorithms which is used to obscure the relationship between the plaintext and the cipher text. In the ShiftRow step, each of the four rows of the matrix is shifted to the left. Any entries that ‘fall off’ are re-inserted on the right side of row. Shift is carried out as follows − First row is not shifted. Second row is shifted one byte position to the left. Third row is shifted two positions to the left. Fourth row is shifted three positions to the left. The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other. In MixColumn step, each column of four bytes is now transformed using a mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round. The last step of the round operation is another AddRoundKey. It is a simple XOR with the actual data and the subkey for the current round.

D. Data Hiding via Key Modulation

Embedding of encrypted data is done via public key modulation mechanism and data extraction is performed by exploiting the distinguishability of encrypted and non-encrypted image blocks. The pool of public keys will be generated randomly and the required set of keys are selected from this pool. Each bit of the message is bitwise XORed with the corresponding public key of the embedding block of the image, and thereby, the whole secret message is embedded into the encrypted image.

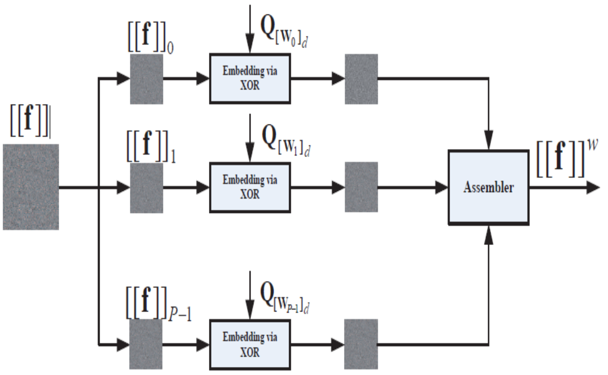

Fig. 5 Schematic diagram of data hiding over encrypted domain.

The schematic diagram of the proposed message embedding algorithm over encrypted domain is depicted in Fig. 5. The first step is to initialize the block index. Then *n* bits of message to be embedded is extracted, which is denoted by W_i. After that, find the public key $Q[W_i]_d$ associated with W_i, where the index $[W_i]_d$ is the decimal representation of W_i.Embed the *n* message bits by XOR ing the message bit with the public key. This process is repeated till all the message bits are embedded into the image.

It is clear that the message embedding is performed without the aid of a secret data-hiding key. Still a high level of embedding security for the data to be transmitted can be guaranteed, because the encryption key provides ample protection. In addition, the major computations involved in message embedding are rather small. They are just simple XOR operations, and all the block-by-block processing is readily made parallel, inorder to achieve high throughput.

E. Image Decryption

At the receiver side, the first step is the image decryption. The decoder has the decryption key K, and tries to recover both the embedded message and the original image simultaneously without any distortions. The original image is obtained by performing the following decryption function,

$$f = \mathrm{Dec}([[f]],K) = [[f]] \oplus K \text{ ----- } (2)$$

The result obtained is the cover image containing the hidden encrypted data.

F. Decoding Candidate Generation

S decoding candidates are created by XORing each block with all the S possible public keys Q_0, Q_1,……… ,Q_{S-1}.

$$\begin{aligned} \mathbf{f}_i^{(0)} &= \mathbf{f}_i^w \oplus \mathbf{Q}_0 = \mathbf{f}_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_0 \\ \mathbf{f}_i^{(1)} &= \mathbf{f}_i^w \oplus \mathbf{Q}_1 = \mathbf{f}_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_1 \\ &\vdots \\ \mathbf{f}_i^{(S-1)} &= \mathbf{f}_i^w \oplus \mathbf{Q}_{S-1} = \mathbf{f}_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_{S-1} \end{aligned}$$

The result so obtained will correspond to the encrypted version of the cover image with the equivalent key stream.

G. Data Extraction and Image Recovery

This module consists of the operations of joint data as well as image recovery, performed with the aid of a powerful classifier, namely, a two-class SVM classifier. Inorder to differentiate between the encrypted as well as the original image blocks, a feature vector is designed which integrates all the characteristics from multiple perspectives. When the comparison between the original image block is done with the encrypted one, the pixels present in the encrypted image will exhibit a much more uniform distribution pattern. The introduction of such a feature element helps in improving the classification performance as the data dispersiveness and denseness are clearly reflected in the process. Not only the feature vector component is used, but an additional component, namely, directional complexity is also included, to indicate the encoded local geometric information. After completely determining the feature vector, a two-class SVM classifier can be trained and the two classes include, 0-class and 1-class, where 0-class corresponds to un-encrypted, original image blocks and 1-class corresponds to encrypted image blocks.

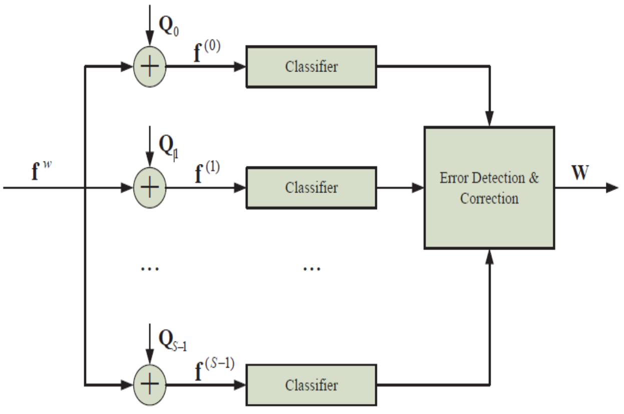

Fig. 6 Schematic diagram of data extraction and image recovery

Fig.6 represents the schematic diagram of joint data extraction and image recovery process. A two-class SVM classifier is used to identify the candidate corresponding to each block. SVM classifier classifies the encrypted and non-encrypted image patches. At the extraction phase, all the S possibilities are tried, and identify the one with generated structured image patches. Based on the index of the identified public key, the embedded message extraction is performed by converting the index i to its binary representation. Thus, the *n* bits of the message can be obtained. Finally, *n* bits from each block is combined to construct the hidden message.

H. Data Decryption

The data obtained in the previous step is the encrypted form of data, where encryption is done using AES algorithm. Now, this encrypted data has to be decrypted to obtain the original hidden message.

The process of decryption of an AES cipher text is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order.

- Inverse ShiftRows
- Inverse SubBytes
- Inverse AddRound Key
- Inverse MixColumns

Since sub-processes in each round are in reverse manner, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related.

The proposed RIDH scheme designed over encrypted domain provides higher embedding capacity, and is able to reconstruct the original image and the embedded message perfectly.

IV. EXPERIMENTAL RESULTS

The experimental evaluation of the embedding performance of the proposed encrypted-domain data hiding scheme is conducted. The host image which is used as the cover image is of dimension 512 x 512 with various characteristics, constituting natural images, synthetic images, and highly textured images. The chosen images are of gray scale texture. The test set is different from the training set which is used to derive the two-class SVM classifier.

The encryption of chosen images are done in a standardized manner. All the chosen images are encrypted using simple XOR operations. Here, the block size is taken as 8 x 8. The workspace section in MATLAB provides the details about data, image, dimensions, values before and after encryption as well as the respective values after decoding has taken place. Irrespective of the settings for block size, the message data can be extracted with 100% accuracy.

It is an advantage of the proposed system that the secret data being transmitted by the respective sender is perfectly extracted from the stego image without being dependent on the block size of the stego image. In this work, the number of data bits to be embedded within an image block is taken to be 4 bits, that is, each of the single image block can accommodate 4 bits of data message.

Hence, for a 512 x 512 image having block size of 8 x 8, the data embedding capacity is also high and it amounts to nearly, 4096 bits, if the number of bits to be embedded in a single block is taken as 4. As and when the number of bits to be embedded within a block is increased, the overall data embedding capacity also increases and can reach upto 50000 message bits per image. However, as the block size decreases, a small number of extraction errors appear.

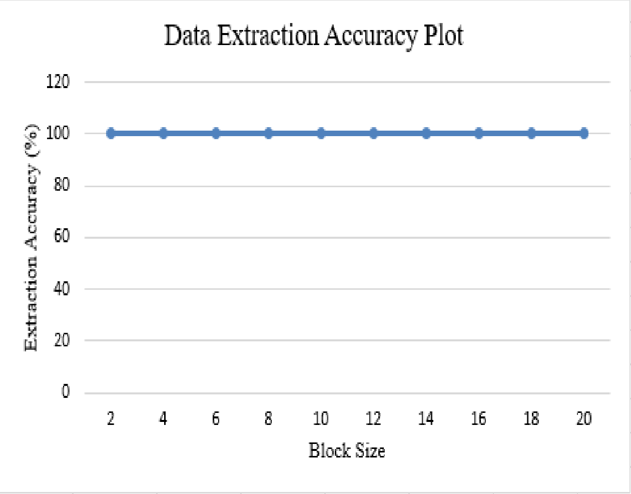

Fig.7 Data extraction accuracy plot

Fig.7 depicts the data extraction accuracy plot. X-axis shows the varying values for block size. Y-axis shows the percentage of data extraction accuracy. The effect brought by increasing the value of *n*, i.e., the number of embedding data bits, is also investigated. The case of embedding more number of data bits into a single image block was experimented.

As the value of *n* increases, it is observed that the number of public keys being generated also increased exponentially. This increases the complexity of data retrieval as the need of examining $S = 2^n$ decoding candidates is unavoidable here. The value of 2^n solely depends upon the value of *n*. Therefore, larger values for *n* results in extraction errors. Hence in the experiment conducted, the value of *n* was chosen as 4. It can be observed that the incorrectly decoded blocks are untypically homogeneous in its textual characteristics, which thereby explains the difficulty in discretion by an error correction mechanism.

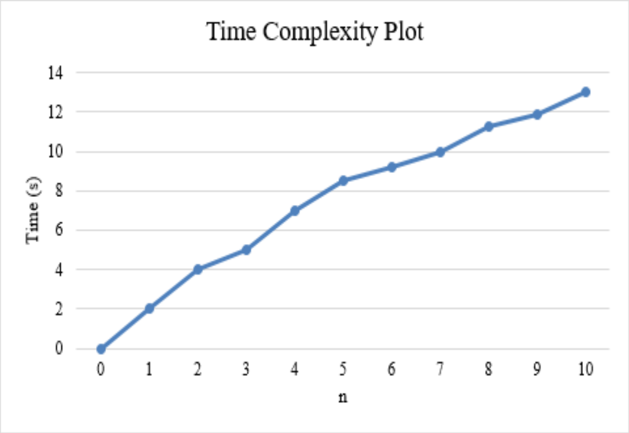

Fig.8 Time complexity plot

Fig.8 illustrates the plot of the time required to perform the joint decryption and the data extraction, with respect to different values of the data bits to be embedded within a single

image block, *n*. The complexity of computation mainly arises from applying SVM classifier to the 2^n decoding candidates. Since the SVM training is done offline, the associated complexity will not be counted while evaluating the joint data extraction and decryption operations. The measurement of time complexity was done over an unparalleled MATLAB implementation.

When the value of *n* is low, the time taken to extract the image and data is also low. But when the value of n grows, the time taken also increases, since the generation of public key solely depends on the value of *n*. This thereby increases the time required to examine the different public keys being generated. The modulations to be performed in that case will also be increased and hence causes a small time lag. However, the complexity of performing a joint decryption and data extraction may not be crucial in many real world applications, especially, in the case of secure remote sensing, where the recipient already has abundant computing resources.

V. SECURITY ANALYSIS

Taking into consideration the context of the attack, the attacker may have access to different amounts of information. Clearly, an attacker can access atleast the watermarked signal. In some cases, the embedded secret message may also become available to the hacker. Therefore, it is very important to analyse the security level of the encrypted domain RIDH scheme for possible different contexts. Three types of attacks have been considered.

- Watermarked Only Attack (WOA), in which the attacker has access only to the watermarked images.
- Known Message Attack (KMA), in which the attacker has ample access to different pairs of previously watermarked images and associated data. However, the recently transmitted message data are unknown to the attacker.
- Known Original Attack (KOA), in which the attacker has access to the previously watermarked as well as corresponding host image. However, the current host image is unknown to the hacker.

The purposes of KMA as well as KOA are to discover the secret data hiding key[10] so that they can hack the different pieces of content. But in the proposed RIDH scheme, no such secret data hiding key is used. Hence no such attacks will occur. Inorder to analyse over the WOA attack, the concept of message indistinguishability need to be considered. The concept of message indistinguishability implies that the attacker can do nothing more than just simple random guessing if the attacker only observes the cipher text. This property is a basic requirement for any secure encryption technique. The property of such message indistinguishability is guaranteed in the proposed system. Hence this justifies the security of the RIDH scheme against the WOA attack. Therefore, it can be concluded that the proposed reversible image data hiding scheme is secure not prone to any of the attacks.

VI. CONCLUSION

Security has become the need of the hour. Ensuring confidentiality of communication transactions has become a critical scenario in the modern cyber world. Many algorithm and methodologies are adopted thereof to provide security while conducting important transactions. In this project, a novel scheme for separable reversible data hiding in encrypted image is proposed, which consists of image encryption, data embedding and data-extraction/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key, which is known to both the sender as well as receiver. Although a data-hider is not aware of the original content of the data, he can compress the least significant bits of the encrypted image to create a sparse space to accommodate the additional data. With an encrypted image containing additional data, the receiver may extract the data using the secret key. The extraction of the secret data as well as the cover image is done jointly and the original content is recovered perfectly without any error by exploiting the property of the spatial correlation in natural image.

If the methodology used is a lossless compression technique, the secret data can be extracted and the original content can be recovered, without any confusion since the lossless compression does not alter the content of encrypted image that has the embedded data. However considering the lossy compression method, it is quite incompatible with encrypted images, if the images are generated by pixel permutation since the encryption is performed by bitwise XOR operation.

In the future, a comprehensive combination of image encryption algorithms and data hiding schemes compatible with lossy compression deserves further investigation.

REFERENCES

[1] M. U. Celik, G. Sharma, and A. M. Tekalp, “Lossless watermarking for Image authentication: a new framework and an implementation,” *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 1042-1049, 2006.

[2] C. Qin, C. -C. Chang, Y.-H. Huang, and L.-T. Liao, “An inpainting-Assisted reversible steganographic scheme using a histogram shifting mechanism, ”*IEEE Trans. Circuits Syst . Video Technol.*, vol. 23, no. 7, pp. 1109-1118, 2013.

[3] Jiantao Zhou, Weiwei Sun, Li Dong, Xianming Liu, Oscar C. Au Yuan Yan Tang,“Secure Reversible Image Data Hiding over Encrypted Domain via Key Modulation,” *IEEE Trans. On* Circuits and Systems for Video Technology, vol. 26, issue 3, pp.441-452,2015.

[4] Z. Erkin , T. Veugen, T. Toft, and R. Lagendijk , “Generating private Recommendations efficiently using homomorphic encryption and data packing,” *IEEE Trans. Inf. Forensics Security.*, vol. 7, no. 3, pp. 1053-1066, 2012.

[5] M. U. Celik, G. Sharma, A. Tekalp, and E. Saber,“Lossless generalized lsb data embedding,” *IEEE Trans. Image Process.*, vol. 14, no. 2, pp.253-266, 2005.

[6] W. Puech, M. Chaumont, and O. Strauss, “A reversible data hiding Method for encrypted images,” in *Proc. of SPIE 6819*, 2008, pp. 1-9.

[7] T. Hong, W. Chen and H. Wu, “An improved reversible data hiding in encrypted images using side match, ” *IEEE Signal Processing Lett.*, vol.19, no. 4, pp. 199-202, 2012.

[8] X. Zhang, Z. Qian, G. Feng, and Y. Ren,“Efficient reversible data hiding in encrypted images,” *J. Vis. Commun. Image R.*, vol. 25, no. 2, pp. 322-328, 2014.

[9] J. MacDonald, “Design methods for maximum minimum-distance error correcting codes,” *IBM J.*, pp. 43-57, 1960.

[10] F. Cayre, C. Fontaine, and T. Furon,“Watermarking security: theory and practice,” *IEEE Trans. Sig. Proc.*, vol. 53, no. 10, pp. 3976-3987, 2005.