INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 05, May 2023, pp : 1198-1205

e-ISSN : 2583-1062

Impact Factor : 5.725

MULTI SECRET SHARING: AN EFFICIENT DATA HIDING WITH ENCRYPTED SECRET SHARING FOR SECURE COMMUNICATION

Dr. Arockia Mary P[1], Albert Raja A[2], Anbarasan S[3], Arivazhagan R[4]

[1]Assistant Professor, Information Technology, V.S.B Engineering College, Karur, Tamilnadu, India.

[2,3,4]UG Scholar, Information Technology, V.S.B Engineering College, Karur, Tamilnadu, India.

ABSTRACT

Information/data hiding is a mechanism which ensures that the presence of the secret data remains undetected. Two types of data hiding techniques are most popular, they are cryptography and steganography. Where cryptography is science of writing secret code and steganography is art and science of hiding the secret code. In cryptography data is converted to unreadable form, so that unauthorized users cannot access the secret data. Steganography process hides message into cover file and forms a Stegno file. In image steganography there is a need of method which will increase the security, reduce the distortion in the stegno file and recovers the data without any loss. In the era of multimedia and internet there is need of reducing time for transmission. The main objective of this project is to establish a secured communication between the sender and the receiver by using emails and other communicating modes. The input text message was hidden within secret image. The secret image can be obtained by super imposing the random shares. Conventional n out of n visual cryptography scheme is used to convert a single image into n shares. In this work, an XOR based multi secret sharing is proposed to send images from the source to the destination in a secured way. A text is written and hidden within cover image. LSB method is used for this purpose. Now the image is splitted into shares. Each share is encrypted using XOR method. The proposed method is n out of n multi secret sharing scheme. Transmission of multiple secret images simultaneously is achieved through this proposed work. The secret image can be revealed only when all the n shares are received by the receiver and decrypted. At the receiver end, the hidden data is extracted from the image shares.

Index Terms—Secure Communication, Data Hiding, Least Significant Bit (LSB), Multi Secret Sharing, XOR Encryption, Key Verification, Data Extraction.

1. INTRODUCTION

1.1 OVERVIEW OF DATA HIDING

Steganography replaces unneeded or unused bits in regular computer files (Graphics, sound, text) with bits of different and invisible information. Hidden information can be any other regular computer file or encrypted data. Steganography differs from cryptography in a way that it masks the existence of the message where cryptography works to mask the content of the message.

Steganography sometimes used in conjunction with encryption. An encrypted file may still hide information using steganography, so even if the encrypted file is deciphered, the hidden information is not seen.

1.2 TYPES OF STEGANOGRAPHY

There are different ways to hide the message in another, well known are Least Significant bytes and Injection.

When a file or an image is created there are few bytes in the file or image which are not necessary or least important. These type of bytes can be replaced with a message without damaging or replacing the original message, by which the secrete message is hidden in the file or image. Another way is a message can be directly injected into a file or image. But in this way the size of the file would be increasing accordingly depending on the secrete message

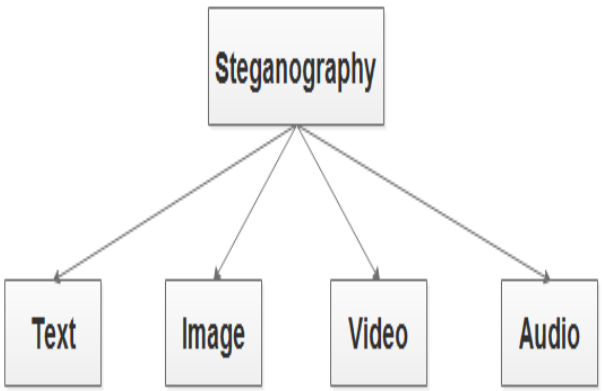

Fig 1.2: Types of Steganography

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 05, May 2023, pp : 1198-1205

1.3 FEW TYPES IN STEGANOGRAPHY IN IMAGES

- Least significant bit insertion
- Masking and filtering
- Redundant Pattern Encoding
- Encrypt and Scatter
- Algorithms and transformations

1. Least significant bit insertion

Least Significant Bit (LSB) insertion is most widely known algorithm for image steganography, it involves the modification of LSB layer of image. In this technique, the message is stored in the LSB of the pixels which could be considered as random noise. Thus, altering them does not have any obvious effect to the image.

2. Masking and filtering

Masking and filtering techniques work better with 24 bit and grey scale images. They hide info in a way similar to watermarks on actual paper and are sometimes used as digital watermarks. Masking the images changes the images. To ensure that changes cannot be detected make the changes in multiple small proportions. Compared to LSB masking is more robust and masked images passes cropping, compression and some image processing. Masking techniques embed information in significant areas so that the hidden message is more integral to the cover image than just hiding it in the "noise" level. This makes it more suitable than LSB with, for instance, lossy JPEG images.

3. Redundant Pattern Encoding

Redundant pattern encoding is to some extent similar to spread spectrum technique. In this technique, the message is scattered throughout the image based on algorithm. This technique makes the image ineffective for cropping and rotation. Multiple smaller images with redundancy increase the chance of recovering even when the stegno-image is manipulated.

4. Encrypt and Scatter

Encrypt and Scatter techniques hides the message as white noise and White Noise Storm is an example which uses employs spread spectrum and frequency hopping. Previous window size and data channel are used to generate a random number and within this random number, on the entire eight channels message is scattered throughout the message. Each channel rotates swaps and interlaces with every other channel. Single channel represents one bit and as a result there are many unaffected bits in each channel. In this technique it is very complex to draw out the actual message from stegano-image. This technique is more secure compared to LSB as it needs both algorithm and key to decode the bit message from stegano-image. Some users prefer this method for its security as it needs both algorithm and key despite the stegano image. This method like LSB lets image degradation in terms of image processing, and compression.

5. Algorithms and transformations

LSB modification technique for images does hold good if any kind of compression is done on the resultant stego-image e.g. JPEG, GIF. JPEG images use the discrete cosine transform to achieve compression. DCT is a lossy compression transform because the cosine values cannot be calculated exactly, and repeated calculations using limited precision numbers introduce rounding errors into the final result. Variances between original data values and restored data values depend on the method used to calculate DCT.

1.4 VISUAL SECRET SHARING APPROACH

A secret sharing (SS) scheme is a cryptosystem that encrypts a secret into multiple pieces called shares so that only qualified sets of shares can be employed to reconstruct the secret. Therefore the SS scheme is one of the most fundamental technologies to realize secure access control. A typical example of secret sharing schemes is a (k, n)-threshold secret sharing scheme. In (k, n)-threshold secret sharing schemes, a secret is encrypted into n shares in such a way that any k or more shares can be employed to reconstruct the secret, while no $k-1$ or less shares leak any information about the secret. In the ordinary secret sharing schemes, secrets and shares are both numerical data and their encryption and decryption is performed by computers. In contrast, there exist secret sharing schemes whose decryption do not require any numerical computations but can be performed by a human. A visual secret sharing (VSS) scheme is an example of such secret sharing schemes. In VSS schemes, secrets and shares are both visual data such as printed texts, hand written notes, pictures and so on. The schemes encrypt a visual secret into visual shares so that humans can recover the visual secret with their eyes by superposing a qualified set of visual shares printed on transparencies.

Visual cryptography is a secret sharing scheme where a secret image is encrypted into multiple shares which independently disclose no information about the original secret image. Due to vast increase in data

transmissions, there is a need of secure transmission. These secret image sharing techniques makes data more secure by converting them into shares. In initial days these techniques can be applied on only one image with multiple shares. But now it is extended to multiple images with multiple shares. There are so many algorithms existed but many of them are for single image with multiple shares. Some of them are multiple image with multiple shares but the number of shares are more than the number of image. The secret image is revealed only after superimposing a sufficient number of shares. In proposed work, vertical share splitting and XOR encryption techniques will be apply for secure multi secret sharing approach. The operation of separating the secret into n share is called encryption and the operation of recovery of secret by stacking of shares is called decryption. In visual cryptography, the sharing of multiple secrets is a novel and useful application.

2. RELATED WORK

Kamal, et al. proposed a method of multiplication by using only N ≥ k servers. This is implemented by sending two shares of the same input to each server. In a ''normal'' method, sending multiple shares to one server violates security because k shares can be leaked from k −1 servers. In our proposed distribution protocol, two encrypted shares for each secret input are sent to each server. In the typical Shamir's (k, n) method, if multiple shares are distributed to one server, k shares are leaked from k − 1 servers (this immediately violates the security of (k, n) threshold secret sharing). For example, if the secret input a is distributed using Shamir's (k, 2n) sharing (2n shares of input a are computed using a (k − 1) degree polynomial), and two shares are sent to each server, 2(k − 1) shares are leaked from k − 1 servers, and the adversary can reconstruct the secret input from these shares. Therefore, this approach violates the security requirement of (k, n) threshold secret sharing, where information from any k − 1 or fewer servers reveals nothing about the original secret input. In contrast, in our proposed method, multiplication is realized under the setting of n ≥ 2k − 1. The disadvantage of this approach is that n ≥ 2k shares are generated for each input. Hence, our protocol requires twice the amount of computational and communication costs for the distribution of the encrypted input. However, in our protocol, the shares of encrypted inputs are multiplied without the need to reconstruct any of the encrypted inputs as a scalar value. Therefore, no information about the input will be leaked even if one (e.g., a or b) or both inputs are equal to 0. This provides a significant advantage to our proposed method over the TUS method. No limitation on the input means that our method can be implemented for any type of application.

Kamal, et al. proposed a method that can realize the direct search function over sentences using the conjunctive search function without the construction of any index. Also propose a method that realizes the search function with multiple search queries using the disjunctive search function. Proposed model of secrecy computation is based on a client/ server model, where any number of clients (owner of the secret information) send shares of their inputs to multiple servers (n number of servers). However, clients who wish to search for any information (searcher) send shares of their trapdoors to n number of servers, which carry out the computation for the clients and return the results to them without learning anything. This model is widely used nowadays and is the business model. In proposed protocol, the searcher could produce valid trapdoor multiple times to perform searching. However, this can be changed according to the intended applications as shown in Section 6.1. In addition, our proposed methods assume a semi-honest adversary, where the adversary follows the protocol specification but may try to learn more than is allowed by the protocol, with at most k − 1 corrupted servers. In addition, also consider the following attacks: the adversary has information of the searcher in addition to information from k − 1 server and attempts to learn the registered document. Finally, we implemented our proposed methods using MATLAB, and demonstrated their effectiveness in realizing the direct search over encrypted documents.

Iwamura, et al. presented a secure computation with information-theoretic security against a semi-honest adversary is possible with $k \leq n < 2k - 1$. The TUS methods realize secure computation of secret sharing by using inputs that has been encrypted with random numbers. This is a combination of an encryption with a random number and computation using secret sharing. Proposed method is solved using the TUS 4 method, where the reconstruction of the multiplication result is only performed by the player that is allowed to know the result. It required when $n>k$. If the server or dealer distributes the random number using secret sharing to all n servers, even if $n-k$ servers are broken or lost, a substitute server can reconstruct the random number that was handled by the broken server and continue the computation. Thus, realizing the server loss resistance of secret sharing. However, it is important that the new server must handle the same random number as the server that it is substituting. This can be realized by implementing it in the algorithm (assuming a semi-honest adversary). Finally, it can be solved depending on the application considered. For example, when considering implementation in searchable encryption, because the owner of secret information will not be the adversary, it can be realized by requesting the owner to generate random numbers that satisfy Condition.

Huang, et al. presented a novel lightweight framework for privacy-preserving CNN feature extraction for mobile sensing based on edge computing. To get the most out of the benefits of CNN with limited physical resources on the

mobile sensors, design a series of secure interaction protocols and utilize two edge servers to collaboratively perform the CNN feature extraction. The proposed scheme allows us to significantly reduce the latency and the overhead of the end devices while preserving privacy. Several schemes have been proposed to outsource the tasks of CNN inference to the cloud servers. However, a deep CNN model typically has a substantially sophisticated structure that consists of many layers of non-linear feature extractors. The extra latency brought by the user-cloud and the inter-cloud interaction will become unacceptable since the users and the cloud servers are usually far away from each other. Moreover, to protect the privacy of the data, they usually utilize the computation-intensive cryptographic primitives like homomorphic encryption or garbled circuits. As described previously, this will incur lots of computation and storage overheads on the end devices and high communication cost between the end devices and the data center. Therefore, we propose a novel lightweight framework based on the edge computing and most of the data processing is moved away from the centralized cloud to the edge of the network (e.g., the edge gateways or the edge servers). To protect the privacy of the data and enable the CNN feature extraction over the encrypted data, we design a series of secure computation protocols based on the additive secret sharing techniques. However, different from previous secret sharing schemes that are designed for at least three parties and the multi-parties play almost the same role, we put the computation-intensive work on just two edge servers. This is due to the fact that the required storage and bandwidth resources for the storage and transmission of the shares have to be at least of the size of the data times the number of shares. Correspondingly, every additional party will increase the communication traffic and the risk of being attacked.

Liu, Yang, et al. implemented a lightweight privacy preserving Faster R-CNN framework (SecRCNN) for object detection in medical images. Faster R-CNN is one of the most outstanding deep learning models for object detection. Using SecRCNN, healthcare centers can efficiently complete privacy preserving computations of Faster R-CNN via the additive secret sharing technique and edge computing. To implement SecRCNN, here design a series of interactive protocols to perform the three stages of Faster R-CNN, namely feature map extraction, region proposal and regression and classification. To improve the efficiency of SecRCNN, we improve the existing secure computation sub-protocols involved in SecRCNN, including division, exponentiation and logarithm. The newly proposed sub-protocols can dramatically reduce the number of messages exchanged during the iterative approximation process based on the coordinate rotation digital computer algorithm. To reduce the communication overhead of the iterative approximation process, we redesigned the existing sub-protocols of common mathematical functions with the CORDIC algorithms. Then, we proposed a series of interactive protocols to implement the training and inference process of SecRCNN, which included feature extraction, region proposal, classification and bounding box regression. Based on SecRCNN, healthcare centers can collaborate to train a more accurate and more reliable model without concern of privacy disclosure.

3. EXISTING METHODOLOGIES

RHD-EI allows a server to embed additional message into an encrypted image uploaded by the content owner, and guarantees that the original content can be losslessly recovered after decryption on the recipient side.

The proposed secret sharing based model with multiple data-hiders is comprised of three phases: the image encryption phase, the data hiding phase, and the data extraction and image recovery phase. Different from previous models that only involve single data hider, the proposed model involves multiple data-hiders. In the proposed model, the original image is converted into multiple encrypted images of the same size as the original image, and the encrypted images are distributed to multiple different data-hiders for data hiding. Each data-hider can independently embed data into the encrypted image to obtain the corresponding marked encrypted image. On the receiver side, the original image is reconstructed from a certain number of marked encrypted images, as well as the embedded data. In joint methods, data extraction cannot be performed in the encrypted domain. Marked encrypted images need to be decrypted with decryption key kd before data extraction, which indicates that the data extraction is related to the content-owner. In this scenario, the receiver needs to obtain permission from content-owner when verifying the integrity of the marked encrypted image. In separable methods, the embedded data is directly extracted from the marked encrypted image without decryption key kd, which indicates that the data extraction is independent of the content-owner. In this scenario, the data hider can update embedded data with a data hiding key as needed.

4. MULTI SECRET SHARING APPROACH

The main objective of this project is to establish a secured communication between the sender and the receiver by using emails and other communicating modes. In this work, an XOR based multi secret sharing is proposed to send images from the source to the destination in a secured way. This method eliminates the fundamental security challenges of VC like external use of code book, random share patterns, expansion of pixels in shared and recovered images, lossy recovery of secret images and limitation on number of shares. The proposed method is n out of n multi secret sharing scheme. Transmission of multiple secret images simultaneously is achieved through this proposed work.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

Vol. 03, Issue 05, May 2023, pp : 1198-1205

e-ISSN : 2583-1062

Impact Factor : 5.725

Text message was created by sender and hidden within selected cover image file. The secret image can be revealed only when all the n shares are received by the receiver and decrypted. The text is typed and hidden in an image. This is done using Modified LSB method. Then the XOR based VC method is used to encrypt the image and send it to the receiver. The key which is used to encrypt the shares will be mailed to the receiver. The receiver will decrypt the shares using the same key that is used for encryption. After that, the hidden text will be extracted from the recovered image using the Modified LSB method.

5. ALGORITHM

Modified LSB Algorithm

In the embedding process of a secret message, a cover image is partitioned into non-overlapping blocks of nine consecutive pixels.

A difference value is calculated from these values of the nine pixels in each block.

All possible difference values are classified into a number of ranges.

The calculated difference value then replaced by a new value to embed the value of a sub-stream of the secret message.

The number of bits which can be embedded in a pixel pair is decided by the width of the range that the difference value.

The way of embedding the secret information within the cover file is called LSB insertion. In proposed technique, the binary representations of the secret data have been taken and the LSB of each byte is overwritten within the image. If 24-bit color images are used to perform LSB, then the amount of modification will be small.

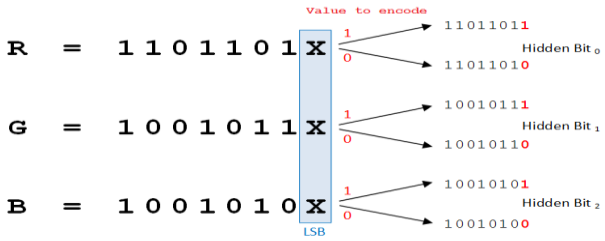

Fig 5.1 LSB Steganography

LSB Encoding- First the unique image and the compressed encrypted secret message are taken. Then the encrypted secret facts need to be transformed into binary format. Binary conversion is accomplished via taking the American Standard Code of Information Interchange (ASCII) values of the person and converting them into binary layout and producing move of bits. Similarly, in cover photo, bytes representing the pixels are taken in unmarried array and byte stream is generated. Message bits are taken sequentially after which are positioned in LSB little bit of image byte. Same process is followed till all the message bits are located in photograph bytes. Image generated is called 'Stego-Image'. It is prepared for transmission through the Internet.

Algorithm for hiding mystery facts in Cover image:

Step-1: Read the cover media image and secret information which is to be embedded in to the cover image.

Step-2: Compress the secret facts.

Step-3: Convert the compressed secrets into cipher textual content by means of using secret key shared by receiver and sender.

Step-4: Convert compressed encrypted textual content message into binary shape.

Step-5: Find LSBs value of each RGB pixels that present in cover image.

Step-6: Embed the bits of the secret data into bits of LSB of RGB pixels of the cover image.

Step-7: Continue the procedure till the secret information is absolutely hidden into cover document.

LSB Decoding- First, 'Stego-Image' is taken and single array of bytes are generated as it become carried out at the time of encoding. The general number of bits of encrypted secret information and the bytes representing the pixels of stego-image are taken. Counter is to begin with set to 1, which in turn offers the index range of the pixel byte where secret message bit is available in LSB. The procedure is continued till very last secret message bit is reached. After this, the bit circulation of the message shall be generated. Available bits are grouped to shape bytes such that each byte represents single ASCII character. Characters are stored in textual content record which represents the encrypted embedded message. After that the decryption and decompression are to be done.

Algorithm for unhiding secret data from Stego image:

Step-1: Read the stego image.

Step-2: Find LSBs value of each RGB pixel of the stego image.

Step-3: Find and retrieve the LSBs of every RGB pixel of the stego image.

Step-4: Continue the procedure till the message is absolutely extracted from stego image.

Step-5: Decompress the extracted secret facts.

Step-6: Using shared key, decrypt secret records to get original records.

Step-7: Reconstruct the secret statistics.

XOR Encryption Algorithm- Exclusive-OR encryption, while not a public-key system such as RSA, is almost unbreakable through brute force methods. It is susceptible to patterns, but this weakness can be avoided through first compressing the file (so as to remove patterns). Exclusive-or encryption requires that both encryptor and decryptor have access to the encryption key, but the encryption algorithm, while extremely simple, is nearly unbreakable. Exclusive-OR encryption works by using the boolean algebra function exclusive-OR (XOR). XOR is a binary operator (meaning that it takes two arguments - similar to the addition sign, for example). By its name, exclusive-OR, it is easy to infer (correctly, no less) that it will return true if one, and only one, of the two operators is true The idea behind exclusive-OR encryption is that it is impossible to reverse the operation without knowing the initial value of one of the two arguments. For example, if you XOR two variables of unknown values, you cannot tell from the output what the values of those variables are. For example, if you take the operation A XOR B, and it returns TRUE, you cannot know whether A is FALSE and B is TRUE, or whether B is FALSE and A is TRUE. Furthermore, even if it returns FALSE, you cannot be certain if both were TRUE or if both were FALSE. If, however, you know either A or B it is entirely reversible, unlike logical-AND and logical-OR. For exclusive-OR, if you perform the operation A XOR TRUE and it returns a value of TRUE you know A is FALSE, and if it returns FALSE, you know A is true. Exclusive-OR encryption works on the principle that if you have the encrypted string and the encryption key you can always decrypt correctly. If you don't have the key, it is impossible to decrypt it without making entirely random keys and attempting each one of them until the decryption program's output is something akin to readable text. The longer you make the encryption key, the more difficult it becomes to break it. The actual way exclusive-OR encryption is used is to take the key and encrypt a file by repeatedly applying the key to successive segments of the file and storing the output. The output will be the equivalent of an entirely random program, as the key is generated randomly. Once a second person has access to the key, that person is able to decrypt the files, but without it, decryption is almost impossible. For every bit added to the length of the key, you double the number of tries it will take to break the encryption through brute force. Exclusive-OR (XOR) encryption is an encryption method that is hard to break through with so called “brute force” methods (brute force = using random encryption keys in the hope you find the correct one.), but the encryption method is susceptible to pattern recognition. Patterns can be easily avoided by first compressing the file (compression already makes it unreadable, it removes patterns for you) before it is encrypted.

The XOR encryption method doesn’t make use of a public-key, such as RSA. Instead both the people that encrypt the file as well as the people that want to decrypt the file need to have the encryption key. The exclusive-OR encryption (as the name already tells you) makes use of the Boolean algebra function XOR. The XOR function is a binary operator, which means that it takes two arguments when you use it. If one of the two arguments is true and the other argument is false, then the XOR function will return true.

Share Encryption

Input: An Image

Output: Encrypted Shares

Select an image

Split into n shares (rows & columns)

Begin

Each share = a * b pixels (a – height, b – width)

For i=0 to a

For j=0 to b

Do

Calculate RGB value for each pixel

Convert to byte

Enter key to encrypt

Convert key to byte

Enc byte = (Byte value) XOR (key)

Replace RGB value by Enc byte value for each pixel

Repeat the same procedure for all shares

End

Decryption

Input: Enc Shares

Output: An image

Begin

Each Share = a*b pixels (a – height, b – width)

For i=0 to a

For j=0 to b

Do

Calculate RGB value for each pixel

Convert to byte

Enter key to decrypt

Convert to byte

Dec byte = (Byte value) XOR (key)

Convert Dec byte to RGB value for each pixel

Join all shares

End

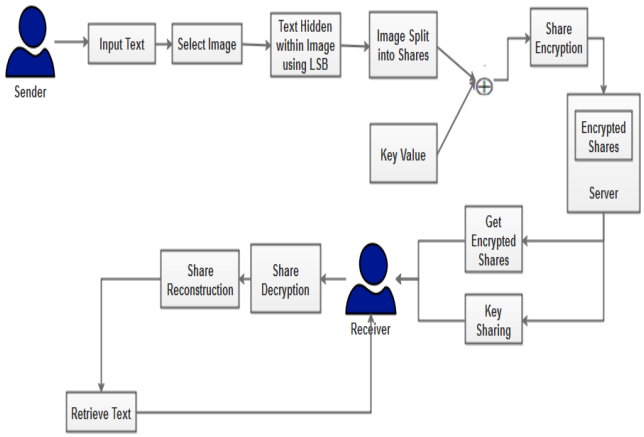

Fig 4.1 proposed work Architecture

6. IMPLEMENTATION

Enrolment and File Sharing - Enrolment is the process of registering in application to get access permission. Then sender could create text message for sharing to the receiver. The secret text message hiding is a process of embedding the secret text imperceptibly into the cover media by minimally modifying the elements of the cover media. In this module sender will generate the content for transmit to the receiver.

Image Upload and Hiding- This process is to select cover media for information hiding. Here images are used as a cover media for the secret message. Cover image is also select by the sender when create the secret message. Original message is hidden into the cover media (image) to improve the security of data sharing. The steganographed image that has to send should be uploaded. The image should be any one of the image supporting formats. A text is written and hidden inside a secret image. This is done by using modified LSB method. The cover image is called as a steganographed image.

Share Split and Encryption- The uploaded image will be divided into "n" number of shares according to the user requirements. "n" is the product of rows and columns. Here, in this project, the number of shares is 16 (4*4). Maximum number of shares is fixed to 8 * 8. Splited image shares will be encrypted separately using XOR method. A key is used to encrypt the shares. Exclusive-or encryption requires that both encryptor and decryptor have access to the encryption key, but the encryption algorithm, while extremely simple, is nearly unbreakable. That key will be mailed to the receiver. If JPEG image is used, the encrypted share will be in black and white color. It will look like a QR code.

Multi Share Sending- All the individual encrypted shares will be stored in a folder. By using this module, all the encrypted shares will be sent to the receiver in a single transmission. This single transmission enables receiver to receive all the shares at a time. This will helps to avoid the information or share missing and also it saves transmission and receiving time for both sender and receiver.

Share Decryption and Data Extraction- All the encrypted shares will be received by the receiver in a single transmission. Each received share will be decrypted individually using inverse XOR method. The key that is received through mail is used in this decryption process. Private key is used for both encryption and decryption process. The output of this module will be an individual share in the decrypted form. All the decrypted individual shares are the input for this module. These individual shares will be joined together to form the original (secret) image. The recovered image can be viewed as a complete single image. The dimensions of both the original image and the recovered image will be the same. The hidden text file will be recovered from the secret image. Receiver gets the secret message with cover text. LSB method is used to retrieve the hidden text Specific key is generated and shared to the receiver during the process of message sending. Receiver can decrypt the text using shared secret key. Then the original message is shown to the receiver.

7. CONCLUSION

The proposed method describes how a secret image is securely communicated from source to destination. The sender has to select the image that should be sent secretly to the receiver. The secret image is splited into “n” number of shares. Each share is encrypted using XOR operation. Then, all the encrypted shares are transmitted in a single transmission to the receiver. The receiver should use the decryption key to decrypt the shares. After decrypting, the individual shares will be joined together to form the recovered (original) image. The recovered image will be of the same size as the original image. The confirmation, outfitted with the thought behind the strategy guarantees that an enemy can't alter the last picture without messing with the previous, which makes its security analysis less difficult and more pragmatic. It will be the subject of future work to explore the validation in more detail.

8. REFERENCES

[1] Ochiai, Shogo, and Keiichi Iwamura. "New Approach to Dishonest-Majority Secure Multiparty Computation for Malicious Adversaries when n< 2k− 1." In 2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW), pp. 355-361. IEEE, 2020.

[2] Kamal, Ahmad Akmal Aminuddin Mohd, and Keiichi Iwamura. "(Server-Aided) Two-Party Multiplication of Encrypted Shares Using (k, n) Threshold Secret Sharing With N≥ k Servers." IEEE Access 9 (2021): 113117-113129.

[3] Kamal, Ahmad Akmal Aminuddin Mohd, and Keiichi Iwamura. "Searchable encryption using secret sharing scheme that realizes direct search of encrypted documents and disjunctive search of multiple keywords." Journal of Information Security and Applications 59 (2021): 102824.

[4] Iwamura, Keiichi, and Ahmad Akmal Aminuddin Mohd Kamal. "Secure Computation by Secret Sharing using Input Encrypted with Random Number." In SECRYPT, pp. 540-547. 2021.

[5] Huang, Kai, Ximeng Liu, Shaojing Fu, Deke Guo, and Ming Xu. "A lightweight privacy-preserving CNN feature extraction framework for mobile sensing." IEEE Transactions on Dependable and Secure Computing 18, no. 3 (2019): 1441-1455.

[6] Ragab, Hany, Alyssa Milburn, Kaveh Razavi, Herbert Bos, and Cristiano Giuffrida. "Crosstalk: Speculative data leaks across cores are real." In 2021 IEEE Symposium on Security and Privacy (SP), pp. 1852-1867. IEEE, 2021.

[7] Abdullah, Dakhaz Mustafa, Siddeeq Y. Ameen, Naaman Omar, Azar Abid Salih, Dindar Mikaeel Ahmed, Shakir Fattah Kak, Hajar Maseeh Yasin, Ibrahim Mahmood Ibrahim, Awder Mohammed Ahmed, and Zryan Najat Rashid. "Secure data transfer over internet using image steganography." Asian Journal of Research in Computer Science (2021): 33-52.

[8] Ibrahim, Dyala R., Je Sen Teh, and Rosni Abdullah. "An overview of visual cryptography techniques." Multimedia Tools and Applications 80 (2021): 31927-31952.

[9] Al-Shaarani, Faiza, and Adnan Gutub. "Securing matrix counting-based secret-sharing involving crypto steganography." Journal of King Saud University-Computer and Information Sciences 34, no. 9 (2022): 6909-6924.

[10] Yu, Han, Yong Li, Junhao Zhang, Dongyu Yang, Tianhao Ruan, Huaying Wang, and Yishi Shi. "Optical image encryption scheme with extended visual cryptography and non-mechanical ptychographic encoding." Journal of Optics 24, no. 3 (2022): 035702.