



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Securing Data in IOT Using Cryptography and Steganography Techniques

Dr.Dahlia sam,S.hariadharsh,
S.yugenthiran

Department of Information Technology
DR.MGR Educational and research
Institute,Chennai,Tamilnadu,India

Abstract—Although cryptography and steganography could be used to provide data security, each of them has a problem. Cryptography problem is that, the cipher text looks meaningless, so the attacker will interrupt the transmission or make more careful checks on the data from the sender to the receiver. Steganography problem is that once the presence of hidden information is revealed or even suspected, the message is become known. According to the work in this paper, a merged technique for data security has been proposed using Cryptography and Steganography techniques to improve the security of the information. Firstly, the Advanced Encryption Standard (AES) algorithm has been modified and used to encrypt the secret message. Secondly, the encrypted message has been hidden using method in [1]. Therefore, two levels of security have been provided using the proposed hybrid technique. In addition, the proposed technique provides high embedding capacity and high quality stego images.

Keywords—Image Steganography; Pixel Value Difference (PVD); Encryption; Decryption; Advance encryption standard (AES)

I. INTRODUCTION

Information transmission through internet may include sensitive personal data which may be intercepted. Also, there are many applications on the internet and many web sites require the users to fill forms that include sensitive personal information such as telephone numbers, addresses, and credit card information. So, the users may need private and secure communications for many reasons such as protect their confidential information from hackers during it passed over an open channel, so the confidentiality and data integrity are required to protect against unauthorized access and use. Cryptography and steganography are the common methods to secure communications.

Cryptography is the science of using mathematics to encrypt and decrypt data to keep messages secured by transforming intelligible data form (plaintext) into unintelligible form (ciphertext). The term cryptography has come from the Greek word “kryptós” standing for “hidden”

and “gráphin” standing for “writing”. Thus, the proper meaning of cryptography is “hidden writing”. Any cryptosystem consists of plaintext, encryption algorithm, decryption algorithm, Cipher text, and Key. Plaintext is message or data which are in their normal, readable (not encrypted) form. Encryption is the process of converting plaintext to cipher text by using key. Cipher text results from encryption by applying the encryption key on the plaintext. Decryption is the process of retrieving the plaintext back from the cipher text. The Key is used info to control the cryptosystem (cipher system), and it is known by the sender and receiver only. While cryptography is very powerful for securing data; the cryptanalysts could success to break the ciphers by analyzing the contents of cipher text to get back the plaintext.

Cryptographic systems are generally classified into three independent dimensions

A. Type of Operation on Plaintext

There are two types of operations that are occurred on plaintext to transform plaintext to cipher text. According to the first operation, each element in plaintext (i.e., bit, letter, group of bits or letters) is substituted for one another in the ciphertext. In this type of operation, a one-to-one mapping between the elements such as Caesar cipher [5]. The principle of the second type of operation is that each character in plaintext is transposed with one another based on a mapping dictated by the key. In this type, the plaintext characters stay the same but they are just moved into different positions such as Rail Fence cipher. Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions.

B. The Number of Used Keys

If the sender and the receiver use one key to encrypt and decrypt the plaintext, the system is referred to as symmetric, single key, secret key or conventional encryption. Symmetric encryption is fairly straightforward and very fast. If the sender and receiver use different keys, public key and private key, to encrypt and decrypt the plaintext respectively, the system is referred to as asymmetric, two – key, or public key encryption.

C. The Way in which The Plain Text is processed

Block cipher operates on fixed-length groups of bits, called blocks, and produces an output block for each input block. A stream cipher operates on each plaintext element continuously, and produces one element at a time, as it goes along.

On the other hand, Steganography is considered the art and science of hiding information in other information. The word Steganography is derived from the Greek words “steganos” meaning “impenetrable” and, “grafia” meaning “writing” defining it as “impenetrable writing”. There are two common techniques for image embedding in steganography; spatial domain and transform domain.

According to spatial domain embedding, the messages are embedded directly into the Least Significant Bits (LSBs). The least significant bits (LSB) insertion method is considered the most common and simplest Steganography method. According to transform domain embedding, the messages are embedded by modifying frequency coefficients of the cover image such as the Fourier transform, discrete cosine transform, or the wavelet transform.

Image steganography system is comprised two algorithms, one for embedding and one for extraction. The embedding process hides a secret message within a cover media (cover image), and the result of embedding process is stego image. The main issue is that the secret message will not be unnoticed if a third party tries to intercept the cover media (cover image). The extraction process is simply because it is the inverse of the embedding process, where the secret message is revealed at the end.

To evaluate the quality of image, stego image and cover image are compared. This requires a measure of stego-image quality, commonly used measures are Mean Square Error (MSE), and Peak Signal-to-Noise Ratio (PSNR). Mean Square Error (MSE) is used to quantify the difference between the initial (cover) and the distorted or noisy (Stego) image [8, 9].

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N (S_{xy} - C_{xy})^2 \quad \dots\dots\dots (1)$$

Where X and Y are the image coordinates, M and N are the number of rows and columns in the input images, respectively. Sxy is the generated stego-image and Cxy is the cover image.

Peak Signal-to-Noise Ratio (PSNR) is used to measure image distortion due to embedding and it is measured in decibels (dB).

$$PSNR = 10 \log_{10} (C_{max}^2 / MSE) \quad \dots\dots\dots (2)$$

Where, Cmax holds the maximum value in the image that is 255 and MSE is the mean square error which is determined by equation (1).

The steganography approaches can be divided into three types

- 1) *Pure Steganography*; it is a technique simply uses the steganography approach only without combining other methods. It is working on hiding information within cover carrier.
- 2) *Secret Key Steganography*; it uses the combination of the secret key cryptography technique and the steganography approach. The idea of this type is to encrypt the secret message by secret key technique and then hide the encrypted data within cover carrier.

3) *Public Key Steganography*; it is the combination of the public key cryptography approach and the steganography approach. The idea of this type is to encrypt the secret data using the public key approach and then hide the encrypted data within cover carrier.

The Difference between Cryptography and Steganography

- Cryptography prevents unauthorized party from discovering the content of communication but Steganography prevents discovery of the existence of communication (i.e., Cryptography makes data gibberish and known the message passing while Steganography tends to conceal presence of hidden data and unknown the message passing).
- Cryptography alters the structure of secret message while Steganography does not alter the structure of secret message.
- Cryptography is more common technology than Steganography technology.
- The most algorithms of Cryptography are well known, but the algorithms of Steganography are still being developed by certain formats.
- In Cryptography, the strong algorithm depends on the key size, the more key size; the more expensive computing power is required to decrypt ciphertext. In Steganography, once the hidden message is detected, the message is become known.
- Cryptography can provide all security objectives by implementing the public and private key(s) with hash functions or authentication codes or digital signatures. Steganography cannot provide most of security objectives (Integrity, authenticity, non-repudiation) by itself without using the cryptographic techniques. However it provides confidentiality by itself because mostly, the concerning person knows that the message is hidden in what kind of medium.

In this paper, the secret Key steganography approach is used to improve security by using modified AES and method in [1] which includes PVD_MPK and MSLDIP-MPK methods to encrypt and hide the message in cover image. Therefore, if an attacker doubts about the stego image and tries to detect the message from the stego image, he would still require the key to decrypt the encrypted message.

The rest of this paper is organized as follows; related work will be discussed in section 2 and the proposed method will be presented in section 3. Then experimental results of the proposed method will be given in section 4. Finally, section 5 concludes the paper and future work.

II. RELATED WORK

There are many methods have been used to provide data security whether by using encryption, steganography or combination between them. Advance encryption standard (AES) method, it is also known as Rijndael, is a symmetric-key block cipher. Unlike DES method, AES method is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, and the number of rounds depend on the key size because it allows the secret key to be expanded to produce sub key for each round. In AES method, the input and output sequences have the same length. According to AES method, substitution

byte, shift rows, mixing column and key adding steps are implemented in every encryption round to encrypt the message, but the Mixing Column step doesn't included in the last round. In the decryption, the four steps are implemented in the reverse way. Also, the inverse of mixing column step doesn't include in the last round of the decryption. The pseudo code of AES is as follows :

1. InitialRound(State, RoundKey)


```
{
  AddRoundKey (State, RoundKey);
}
```
2. Rounds (State, RoundKey) {


```
  SubBytes (State);
  ShiftRows(State);
  MixColumn(State);
  AddRoundKey(State, RoundKey);
}
```
3. FinalRound(State, RoundKey)


```
{
  SubBytes (State);
  ShiftRows(State);
  AddRoundKey(State, RoundKey);
}
```

Where SubBytes is a non-linear substitution step where each byte is replaced with another according to a lookup table after each byte is interpreted as two hexadecimal digits. ShiftRows is a transposition step where it performs a circular rotates on each row of 0, 1, 2 & 3 places for respective rows. MixColumns is a mixing operation which operates on the columns of the state. It transforms each column of the state to a new column. AddRoundKey is precedes one column at a time. AddRoundKey adds a round key word with each state column matrix using bitwise XOR operation [14]. The advantages of using AES algorithm are; it is more secure, support larger key sizes than DES, faster in both hardware and software, reasonable cost, and its main characteristics flexibility and simplicity [15].

In a method has been proposed based on spatial domain instead of using LSB1 (First Least Significant Bit) of the cover image for embedding the message bits, LSB-3 (Third Least Significant Bit) has been used to hold the message bits and LSB-1, LSB-2 may also be modified. According to this method, it is found that the LSB-1 method has more PSNR (Peak Signal to Noise Ratio) values which mean that the image's quality using LSB-1 method is better than that LSB-3 method. Also, the capacity is still the same. So, the basic LSB1 method has better results than the proposed LSB-3 method.

In [1], a steganography algorithm has been proposed by combining PVD-MPK and MSLDIP-MPK methods. The idea of this proposed algorithm is that using the digits of MPK encoding instead of bits of secret message in hidden step to increase the capacity and quality because MPK encoding represents each character by two digits instead of 8 bits. As an example, the letter "a" will be represented as 21, this mean that the letter "a" can be typed by pressing the key no.# (2) in the keypad only one time. Firstly, the cover image is divided into non-overlapping blocks, where each block consists of two consecutive pixels and each byte in the message is converted to Mobile Phone Keypad (MPK) format, then the difference value d_i for each block of two consecutive pixels P_i, P_{i+1} is calculated. If the difference value is larger than 19, this means that the

two pixels are located in high level (edge area) and MSLDIPMPK method is used to hid data to increase capacity. Therefore, each last digit in two pixel of block is replaced by the digit value of the secret message. On the contrary, if the difference value is smaller than 19, this means that the two pixels are located in low level (smooth area) and PVD_MPK method is used to hide data to increase image quality. Then, find the optimum range R_i for the difference d_i where $R_i \in [l_i, u_i]$, and l_i and u_i are the lower and upper bound of each range of the range table and $l_i \leq d_i \leq u_i$ because this method uses the ranges (0-4), (5-9), (10-14), (15-19), (20-24), (25-29) etc. to calculate the new difference and make it nearly as possible to the original difference. The new difference is calculated by equation (3):

$$d_i' = l_i + b/2 \dots \dots \dots (3)$$

Where d_i' is the new difference, l_i is the lower bound of the range, and b is the digit that needs to be hidden. Replace the old difference with the new difference by using Equation (4):

$$(p_i', p_{i+1}') = \begin{cases} (p_i + \lfloor \frac{m}{2} \rfloor, p_{i+1} - \lfloor \frac{m}{2} \rfloor), & \text{if } p_i \geq p_{i+1} \text{ and } d_i' > d_i; \\ (p_i - \lfloor \frac{m}{2} \rfloor, p_{i+1} + \lfloor \frac{m}{2} \rfloor), & \text{if } p_i < p_{i+1} \text{ and } d_i' > d_i; \\ (p_i - \lfloor \frac{m}{2} \rfloor, p_{i+1} + \lfloor \frac{m}{2} \rfloor), & \text{if } p_i \geq p_{i+1} \text{ and } d_i' \leq d_i; \\ (p_i + \lfloor \frac{m}{2} \rfloor, p_{i+1} - \lfloor \frac{m}{2} \rfloor), & \text{if } p_i < p_{i+1} \text{ and } d_i' \leq d_i. \end{cases} \dots \dots (4)$$

This equation adjusts the value of p_i and p_{i+1} to modify the difference. Where, m is the difference between the original and the new differences (i.e., $m = |d_i - d_i'|$), p_i and p_{i+1} are the first and second pixel in a block before embedding, and p_i' and p_{i+1}' are the first and second pixel in a block after embedding. Then, Compute the remainder value by:

$$\text{Rem} = b \bmod 2 \dots \dots \dots (5)$$

If Rem equal 1, make p_i odd, else make p_i even. The experimental results showed that the PSNR and Maximum Hidden Capacity (MHC) of this technique are high comparing to the existed techniques.

In [17], a proposed technique has been introduced. The results of the proposed technique prove that it is more secure than other techniques against statistical attacks which are commonly used in steganalysis. This is because it ranked images in a user's library based on their suitability as cover objects for some data. The process is repeated for each image in a user's image library. Each bit of the encrypted data is compared to the least significant bit of the pixel bytes in an image. The comparisons are made starting from the first byte until the last byte in the image that permits all data to be hidden in that image. By matching data to an image, there is less chance of an attacker being able to use steganalysis to recover the data. The image is then given a rank based on the percentage of least significant bits that match the encrypted data bits. Before hiding the data in an image, the data is first encrypted using the RSA public key algorithm, and then the encrypted data is hidden by using LSB.

In [18], a new method has been proposed for hiding any encrypted secret message inside a cover file. To embed a secret message file in the cover file, they have used two distinct methods; (1) they encrypt the secret message file using simple bit shifting and XOR operation in the secret message file. (2) The encrypted secret message is embedded in the cover file in

alternate byte instead of changing the LSB of the cover file bytes. They change LSB and LSB+3 bits of the cover file bytes. Their method could be most appropriate for hiding any file in any standard or nonstandard cover file such as word, excel, .ppt, .exe, image, audio, video files. Also, this method may be used for sending some secret key to someone over mail as the intruder may not be able to unhide and decrypt the secret message.

III. THE PROPOSED METHOD

The main objective of the proposed method is to introduce more secure communication by merging cryptography and steganography techniques to make it more difficult for a steganalyst to retrieve the plaintext of a secret message from a stego-object. The proposed method is divided into two parts. In the first part, the AES algorithm will be modified to be suitable for steganography method and it is called AES_MPK algorithm. In the second part, the AES_MPK algorithm will be merged with steganography algorithm to hide the encrypted data in image where a message being sent is concealed [1]. Therefore, two levels of security have been applied.

A. The Modified AES (AES_MPK) Algorithm

According to the modified AES_MPK algorithm, four types of transformations are used like AES; substitution (SubBytes), permutation (ShiftRows), MixColumns, and keyadding, to provide security [13]. Because of the AES is based on the Rijndael cipher, it performs four types of transformation based on the operations in finite field GF (2⁸). Several operations are defined at byte level, and used with bytes representing as elements in the finite field or Galois field GF (2⁸). Then, it represents the input and the output in form of hexadecimal digits (i.e. two hexadecimal digits for each byte) [13]. Therefore, AES algorithm is modified to make the input and the output in the form of MPK digits because the PVD_MPK and MSLDIP-MPK methods use the MPK digits for hiding the data. This modified AES algorithm called AES_MPK algorithm.

The basic idea of AES_MPK algorithm is to convert each block (i.e. 16 byte) of data to MPK digits (i.e. two digits for each byte). Then each block is divided into two states; the first state represents the first digits of 16 byte and second state represents the second digits of 16 byte. Two states are filtered by replacing each digit 8 or 9 with 7 to make state suitable for operations of GF (2³) that are applied at each stage of encryption but save them in their location in each state. Finally, apply operations of GF (2³) in MPK digits in the four types of transformation expect 8 and 9 digits which will be replaced with 7 and keep in their location. This is because the elements of GF (2³) are 0, 1, 2, 3, 4, 5, 6, and 7. The four types of transformation are:

1) *Substitution*: The first transformation is SubBytes. To substitute a byte, this requires two steps; first apply multiplicative inverse in GF (2³) on each element in two states. Second, apply an affine transformation over GF (2³) on each element in the two states by representing each element as 3 bits (b₀, b₁, b₂). AES_MPK depicts this transformation in matrix form as follows:

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Where (b₀, b₁, b₂) are bits of element before affine transformation and (c₀, c₁, c₂) are bits of element after affine transformation.

The SubBytes stage provides confusion (i.e., it obscures the relationship between the plaintext and the ciphertext) [5]. In decryption, InvSubBytes is performed by applying affine transformation first, and then multiplicative inverse in GF (2³) is applied. The affine transformation in decryption will be as follows:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

2) *Permutation*: Another transformation found in a round is called ShiftRows, which permutes the bytes. It performs a circular rotate on each row of 0, 1, 2 & 3 places for respective rows. This shift moves an individual byte from one column to another, and ensures that the 4 bytes of one column are spread out to four different columns. The ShiftRows stage provides diffusion of column values between columns. The rows shift to left as follows:

- Row0: No shift
- Row1: 1-byte shift
- Row2: 2-byte shift
- Row3: 3-byte shift

In decryption, InvShiftRows is performed where the circular will be shifted in the opposite direction for each row.

3) *MixColumns*: This stage is used to mix bytes by using matrix multiplication. It operates at the column level where each column of the state is transformed to a new column by multiplying it in constant matrix. This stage changes the bits inside a byte, based on the bits inside the neighboring bytes.

We need to mix bytes to provide diffusion at the bit level.

In decryption, InvMixColumns is performed by the inverse of constant matrix in multiplication.

$$\begin{array}{cccc} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ C = 3 & 1 & 1 & 2 \\ 5 & 0 & 6 & 2 \\ 2 & 5 & 0 & 6 \\ 6 & 2 & 5 & 0 \\ 0 & 6 & 2 & 5 \end{array} \longleftrightarrow \text{Inverse} \quad C^{-1} =$$

Where C is constant matrix and C⁻¹ is inverse constant matrix in GF (2³)

4) *AddRoundKey*: the AddRoundKey stage is a simple bitwise XOR of the current block (two states) with two portions of the expanded key. Note; this is the only step which makes use of the key. This step makes the cipher as a series of XOR with expanded keys then scramble/permute block repeated. This provides efficient and highly secure.

Note that: AES has defined three versions, with 10, 12, and 14 rounds. Each version uses a different cipher key size. We work in version with 10 rounds and key size is 16 byte.

The pseudo code of the modified AES_MPK algorithm is as follows:

AES_MPK Algorithm

Input: Secret Message SM, Cipher Key K.

Output: Cipher Message CM.

Steps:

1. Make key expansion of K that produces two lists of all sub keys.

2. Partition SM to blocks ($B_1, B_2, B_3 \dots B_n$) each block consists of 16 byte.
3. **for** each B_i block **do**
4. Convert each byte to MPK digits (two digits for each byte).
5. Divide B_i to two state arrays (4×4).
6. Filter two states.
7. Make pre round AddRoundKey which is a simple bitwise XOR of the current two states with two sub keys
8. **repeat**
9. Apply the four transformations (SubBytes, ShiftRows, MixColumns, and AddRoundKey) in two states.
10. **until** nine round.
11. At final round implements SubBytes, ShiftRows, and AddRoundKey but MixColumns is deleted.
12. Return the digits 9 and 8 in their place in each state.
13. Mix two states to be one block.
14. Convert block to characters by using MPK decoding (i.e. two digits represent character). The result represents cipher block
15. **end**
16. Concatenate the currently cipher block with the previous cipher blocks to collect CM.

The designed ciphers and inverse ciphers of ASE_MPK algorithm are as follows:

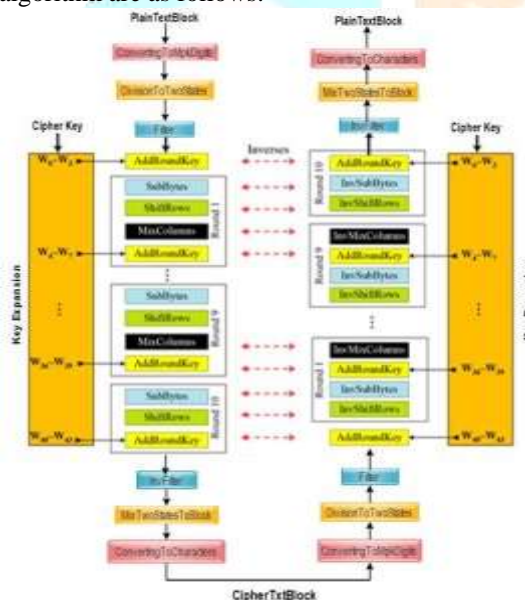


Fig. 1. Encryption and Decryption design

Key Expansion:

As shown in Fig. 1, a key expansion is used in encryption and decryption to create sub key for each round. ASE_MPK algorithm likes the original AES algorithm where it uses a key expansion process. If the number of rounds is N_r , the keyexpansion routine creates $N_r + 1$ 128-bit round keys from one single 128-bit (16 byte) cipher key [13]. In addition, cipher key is converted to MPK digits (32 digits), then divided it to two blocks each block is 16 digits (i.e. the first block represents the first digits in 16 byte and the second block represents the second digits in 16 byte), and then make expansion on two block as follows:

Fig. 2. Key Expansion

Fig. 2 shows each group of 4 bytes in the key being assigned to the first 4 words, then the calculation of the next 4 words based on the values of the previous 4 words, which is repeated several times to create all subkeys. Where $T_i = \text{RotateWord}(\text{SubWord}(w_{i-1}))$, RotateWord performs a onebyte left circular rotation on the 4-byte word, and SubWord performs a byte substitution for each byte of the word as SubBytes function that is used in encryption rounds.

Example:

Suppose the currently block is: "as programs grow" that is 16 byte and key is "test secret key!" 1) Make key expansion of key. 2) Convert block to MPK digits so becomes: "21 74 00 71 73 63 41 73 21 61 74 00 41 73 63 91" 3) Divide block to two states:

State 1 is	7	7	0	9	State 2 is	1	3	0	1
------------	---	---	---	---	------------	---	---	---	---

4) Apply filtering on the two states by replacing digits 9 and 8 with 7. In this case, the edition on only state1 because state 2 doesn't contain 9 or 8 digits.

5) Make pre round addition on the two states with two blocks of key.

6) **repeat**

7) Apply SubBytes, ShiftRows, MixColumns, and AddRoundKey in two states.

8) **until** round nine.

9) Make SubBytes, ShiftRows, AddRoundKey of last round on the produced two states that are produced from round nine.

10) Make InvFilter to return 9 or 8 digits to its location, the result is:

State1	1	3	3	1	6	4	1	6	State 2 is
4	0	7	0	2	4	7	6		
2	0	5	0	is	2	6	5	97	
3	2	3	4						
6	1	7	2						

11) Mix the two states to be one block

14 62 23 26 30 40 42 61 37 15 73 57 10 60 64 97 02

12) Covert block to characters by converting each two digits to character :

The cipher text of this block is: "-ncB?\hmF.rL)|6X0".

13) Concatenate the currently cipher block with the previous cipher blocks.

B. Merging Encryption and Steganography Algorithms

The pseudo code of the proposed merged algorithm is as follows:

CRPTO_STEGNO Algorithm

Input: Secret Message M, Cipher Key K, Cover Image C.

Output: Stego Image S. **Steps:**

1. M has been encrypted by using the AES_MPK that takes M and K then produces cipher text.
2. The cipher text has been hidden in C by using the method in [1] that is combining PVD-MPK method with MSLDIP-MPK method and then produces S.

The block diagram of the proposed merged method is shown in Fig. 3.



Fig. 3. Block Diagram of Proposed System

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Experimental Environment:

The used PC with windows 8 and equipped with a Genuine Intel(R) Core(TM) i5-4210U CPU 1.70 GHz 240 GHz with 8 GB RAM memory. MATLAB R2011b and Matlab code are used to implement the algorithm.

B. Benchmarks:

Several experiments with size 512 * 512 and 256 * 256 standard gray-scale images (Cameraman, Lena, Peppers, Lake, Airplane, and Baboon) were employed to embed a text encrypted message.

The message is firstly encrypted by AES-MPK algorithm, and then it is hidden by PVD-MSLDIP-MPK algorithm to be sent. At the receiver, the hidden message is extracted and then decrypted.

C. Evaluation Parameters:

The performance of the proposed merged algorithm is evaluated by measuring imperceptibility (Stego-image quality), and payload (hiding capacity). Imperceptibility (Stego-image quality) measures how much difference (distortion) was caused by data hiding in the original cover, where the higher the stegoimage quality, the more invisible the hidden message. The stego-image quality could be judged by using Peak Signal to Noise Ratio (PSNR) which determined by equation (2). If PSNR of gray scale image is larger than 36 dB then the human visual system (HVS) cannot distinguish between the cover image and the stego image [20].

Payload (Hiding Capacity) indicates how much data can be hidden within a cover image without making obvious distortion in the cover image quality. It is important to know that it has no meaning that an algorithm hides large amount of data and produces large distortion in image quality. So, it can say that a

steganographic technique is an addition if it proves the increasing in payload while maintaining an acceptable visual quality of stego-image or improve the stego-image quality, at the same, hiding capacity level or if it can improve both [10].

The results that are obtained from these experiments are recorded and summarized in the table 1, table 2, table 3, and figure 4.

TABLE I.

COMPARISON THE ALGORITHM IN [21] WITH THE PROPOSED ALGORITHM WITH DIV=19

Cover Image 256*256	Hiding Capacity (bytes)	PSNR of Method in [21]	Hiding Capacity (bytes)	PSNR of Proposed Method
Baboon	18.616	33.80	18.624	41.7875
Lena	13.003	43.56	13.008	44.9864
Pepper	16.394	36.91	16.400	43.6845

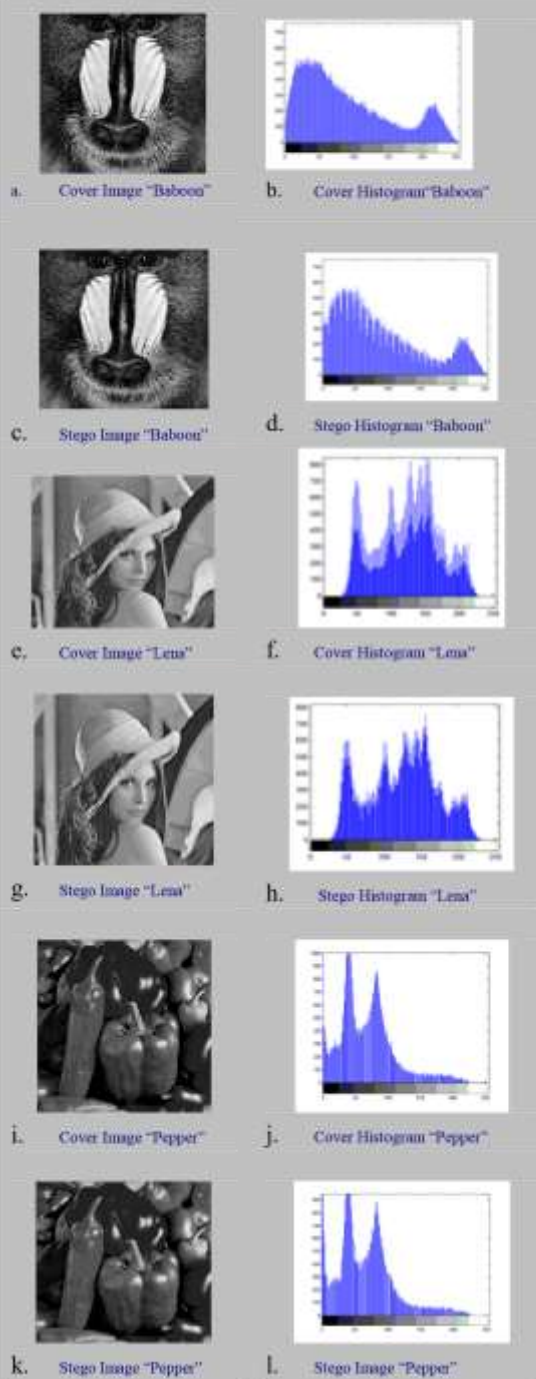


Fig. 4. Three cover images and output stego-images used in system simulation with their corresponding histogram

In table I, a comparison between the proposed merged method and method in [21] has been made by hiding (18.616, 13.003, and 16.394) secret bytes in 256 x256 cover images (Baboon, Lena, and Peppers) respectively. The results indicate that, the proposed method has higher PSNR values than method in [21], and also the PSNR values are much greater than 36 dB. This proves the suitability of the proposed method.

In figure 4, a comparison between the resulting stego images and their histograms with cover images and their histograms has been made. We can see that there is no significant change in stego histograms and visual quality of the resulting stego-image of the three images. Also, the change in histograms is influenced by the properties of image (i.e. the smooth area and edge area), so the larger number of edge areas in the original image, the more change in histogram of stegoimage such as Baboon and Lena contrast to Pepper image. This is because the method that is used in hiding in smooth areas is MPK_PVD method.

TABLE II. COMPARISON OF PAPER [22] WITH THE PROPOSED METHOD WITH DIV=19

Cover Image 512x512	Hiding capacity (bytes)	PSNR of method in [22]	Hiding capacity (bytes)	PSNR of Proposed method
Baboon	57.043	39.2	63.408	42.8113
Boat	52.490	41.0	62.576	44.5444
Lake	52.662	41.5	62.672	44.0388
Lena	50.894	43.4	62.208	44.9678
Peppers	50.815	42.5	62.000	44.8326

In table II, a comparison between the proposed method and method in [22] has been made by hiding (63.408, 62.576, 62.672, 62.208, and 62.000) secret bytes in 512 x 512 cover images (Baboon, Boat, Lake, Lena, and Peppers) respectively. The results indicate that, the proposed method has higher PSNR values than method in [22], although more hiding capacity is used than that used by method in [22]. This proves the improved the stego-image quality and the hiding capacity, also the suitability of the proposed method.

TABLE III. LISTING OF COVER IMAGES AND SECRET MESSAGES WITH THE TIME REQUIRED TO ENCRYPT, HIDE, EXTRACT, AND DECRYPT OF SECRET MESSAGE

Cover Image		Secret Message (Bytes)	Time (Seconds)			
Name	Size		Encrypt	Hide	Extract	Decrypt
Lena	256*256	144	4.28	0.01	0.02	4.67
Pepper	256*256	416	12.31	0.03	0.04	13.72
Baboon	256*256	992	30.01	0.17	0.10	32.34
Camera man	512*512	1.744	50.43	0.09	0.19	55.96
Boat	512*512	2.720	81.42	0.15	0.29	89.01
Airplane	512*512	3440	101.1	0.17	0.49	106.3

V. CONCLUSIONS AND FUTURE WORK

In this paper, a new secure communication model has been presented that combines cryptography and steganography techniques to provide two layer of security, so the steganalyst can't reach to plaintext without knowing the secret key to decrypt the ciphertext. Firstly the secret data has been encrypted by using the AES_MPK then the encrypted data has been hidden in gray image by using PVD-MPK and MSLDIPMPK methods. Due to this combination, the secret data can transmit over open channel because the cipher text does not look meaningless but its presence is concealed by using steganography for hiding cipher text in the images. Experimental results showed that our proposed model can be used to hide much more information than that other existed methods and the visual quality of the stego

image is also improved, in addition to it is effective for secret data communication.

In the future work, we are looking forward to try applying the proposed method on audio and video. Also, we are looking forward to enhance the proposed method to make the capacity higher than it while keeping the same PSNR or higher.

REFERENCES

- [1] M. E. Saleh, A. A. Aly, and F. A. Omara, "Enhancing Pixel Value Difference (PVD) Image Steganography by Using Mobile Phone Keypad (MPK) Coding," International Journal of Computer Science and Security (IJCSS), Volume (9), Issue (2), pp. 397 - 397, 2015
- [2] F. A. P. Petitcolas et al, "Information Hiding-A Survey," Proceedings of the IEEE, special issue on protection of multimedia content, Vol. 87, Issue. 7 PP. 1062-1078, July 1999.
- [3] K. R. Babu et al, "A Survey on Cryptography and Steganography Methods for Information Security," International Journal of Computer Applications (0975 - 8887), Vol. 12, No.2, PP. 13-17, November 2010
- [4] R. Oppliger, "SSL and TLS: Theory and Practice," ARTECH HOUSE, 2014.
- [5] B. Schneier, "Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C (cloth)," pp. 1-1027, January 1996. [6] K. Nitin K and N. Ashish V, "Comparison of Various Images Steganography Techniques," International Journal of Computer Science and Management Research, Vol 2, Issue 1, PP. 1213 - 1217, January 2013.
- [7] S. Sharda and S. Budhiraja, "Image Steganography: A Review," International Journal of Emerging Technology and Advanced Engineering (IJETA), Vol.4, Issue 1, PP. 707-710, January 2013.
- [8] J. Raphael, and V. Sundaram, "Cryptography and Steganography - A Survey," International Journal, ISSN: 2229 -6093, Vol 2 (3), PP. 626- 630, 2011.
- [9] A. J. Altaay et al, "An Introduction to Image Steganography Techniques," International Conference on Advanced Computer Science Applications and Technologies, PP. 122 - 126, 2012.
- [10] L. M. Marvel et al "Spread Spectrum Image Steganography," IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 8, NO. 8, PP. 1075 - 1083 AUGUST 1999
- [11] B. Zaidan, A. A. Zaidan, A.K. Al-Frajat and H.A. Jalab, "On the Differences between Hiding Information and Cryptography Techniques: An Overview", Journal of Applied Sciences 10(15), PP.1650-1655, 2010.
- [12] A. Shoukat et al "A Survey about the Latest Trends and Research Issues of Cryptographic Elements," IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, PP. 140-149, May 2011.
- [13] J. Daemen, and V. Rijmen "AES Proposal: Rijndael," version 2, PP. 145, 1999, Available in <http://csrc.nist.gov/archive/aes/rijndael/Rijndaelammended.pdf#page=1>.
- [14] R. H. Sakr, F. Omara, and O. Nomir, "A COMPARATIVE STUDY OF SECURITY ALGORITHMS FOR CLOUD COMPUTING," International Journal of Intelligent Computing and Information Science, Vol.13, PP. 73-84, OCTOBER 2013.
- [15] S. Murphy, "The Advanced Encryption Standard (AES)," information Security Technical Report, Vol. 4, No. 4, PP.12-17, 1999.
- [16] I. A. Sada, "Hiding Data Using LSB-3", J.basrah researches (sciences), vol. 33. No. 4, pp. 81-88, Dec. 2007.
- [17] M. Juneja, and P. S. Sandhu, "Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption," International Conference on Advances in Recent Technologies in Communication and Computing, PP. 302-305, 2009.
- [18] Dutta et al, "New Data Hiding Algorithm in MATLAB using Encrypted secret message," International Conference on Communication Systems and Network Technologies, PP. 262-267, 2011.
- [19] W. Stallings, "Cryptography and Network Security Principles and Practices, Fourth Edition," Prentice Hall, PP.1-663 November 16, 2005.
- [20] R.O. El Safy, H. H. Zayed, and A. El Dessouki, "An adaptive steganography technique based on integer wavelet transform," ICNM International Conference on Networking and Media Convergence, PP. 111-117, (2009).
- [21] H.B. Kekre, P. Halamkar, and K. Dhamejani, "Capacity Increase for Information Hiding Using Maximum Edged Pixel Value

Differencing,” Springer-Verlag Berlin Heidelberg , PP. 190-194, 2011.

- [22] J.-C. Joo, H.-Y. Lee, and H.-K. Lee, “Improved Steganographic Method Preserving Pixel- Value Differencing Histogram with Modulus Function”, EURASIP Journal on Advances in Signal Processing, PP.1–13, 2010

