Volume 3, Issue 3, March 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Spread Spectrum Image Steganography with Advanced Encryption Key Implementation

B.Padmasri
Assistant Professor
PRIST University
Trichy Campus, India.

M.Amutha surabi
Assistant Professor
PRIST University
Trichy Campus, India.

***Abstract*-Security has become an increasingly important feature with the growth of electronic communication. This paper, describes a design of effective security for data communication by implementing, SSISAE (spread spectrum image steganography with advanced encryption key). This system hides and recovers a message of substantial length within digital imagery while maintaining the original image size and dynamic range. The hidden message can be recovered using appropriate keys without any knowledge of the original image. Image restoration, error-control coding, and techniques similar to spread spectrum are described, and the performance of the system is illustrated.**

***Index Terms*—Information Hiding, Steganography, Spread Spectrum, Advanced Encryption Key.**

1. Introduction

The information hiding techniques have become an important research area in recent years, ever since researchers realized that developing techniques to solve unauthorized copying, tampering, and distribution of multimedia data via Internet is urgent. Cryptography was created as a technique for securing the secrecy of communication and many different methods were developed to encrypt and decrypt data in order to keep the message secret. But sometimes it is not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. This concept is implemented through the technique called **steganography**.

Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. Steganography employs an innocent-looking media called a host image to imperceptibly carry secret data to an intended recipient. The image embedded with the secret data looks like a normal image. Unintended recipients of this image are unaware of the existence of the hidden data. In steganography secret message or the data that the sender wishes to transmit confidentially can be text, images, audio, video, or any other data type that can be represented by a stream of bits. The cover or host image is the medium in which the message is embedded and serves to hide the presence of the message. The message embedding technique is strongly dependent on the structure of the cover, and in this paper cover and secret messages are restricted to be digital images. The cover-image with the secret data embedded is called the Stego-Image. The Stego-Image should resemble the cover image under casual inspection and analysis. In addition to providing invisibility of hidden message, for higher security requirements the message data can be encrypted before embedding them in the cover-image to provide further protection. In spread spectrum communications, the signal energy inserted into any one frequency is too undersized to create a visible artifact and the secret image is scattered over a wide range of frequencies, that it becomes robust against many common signal distortions. Because of its good correlation properties, noise like characteristics, easier to generate and resistance to interference, Pseudo noise sequences are used for Steganography.

The different requirements that need to be considered while designing a steganographic system are:

Invisibility – To provide invisibility is the first and foremost requirement of a steganographic algorithm, since the strength of steganography lies in its ability to be unnoticed by the human eye. The moment that one can see that an image has been tampered with, the algorithm is compromised.

Payload capacity – Unlike watermarking, which needs to embed only a small amount of copyright information, steganography aims at hidden communication and therefore requires sufficient embedding capacity.

Robustness against statistical attacks - Statistical steganalysis is the practice of detecting hidden information through applying statistical tests on image data. To pass without being detected, a steganographic algorithm must not leave a mark in the image as be statistically significant.

Robustness against image manipulation – Image manipulation, such as cropping, rotation, scaling etc. can be performed on the image before it reaches its destination. These manipulations may destroy the hidden message. It is preferable for steganographic algorithms to be robust against either malicious or unintentional changes to the image.

Independent of file format - The most powerful steganographic algorithms should possess the ability to embed information in any type of file. This also solves the problem of not always being able to find a suitable image at the right moment, in the right format to use as a cover image.

2. Related Work

Generally, steganographic methods proposed in the past few years can be categorized into two types. The methods of the first type employ the spatial domain of a host image to hide secret data. In other words, secret data are directly embedded into the pixels of the host image. Steganographic methods of the second type employ the transformed domain of a host image to hide secret data. Transformation functions like the discrete cosine transform (DCT) or discrete wavelet transform (DWT) are first exploited to transform the pixel values in the spatial domain to coefficients in the frequency domain. Then the secret data are embedded in the coefficients.

LSB (Least significant Bit) Steganography -Here spatial features of image are used. This is a simplest steganographic technique that embeds the bits of secret message directly into the least significant bit (LSB) plane of the cover image. In a gray-level image, every pixel consists of 8 bits. The basic concept of LSB substitution is to embed the confidential data at the rightmost bits (bits with the smallest weighting) so that the embedding procedure does not affect the original pixel value greatly [1]. This method is easy and straightforward but this has low ability to bear some signal processing or noises and secret data can be easily stolen by extracting whole LSB plane.

Gray Level Modification (GLM) Steganography -It maps data (not embed or hide it) by modifying the gray level values of the image pixels. GLM Steganography uses the concept of odd and even numbers to map data within an image. It is a one-to-one mapping between the binary data and the selected pixels in an image. From a given image a set of pixels are selected based on a mathematical function. The gray level values of those pixels are examined and compared with the bit stream that is to be mapped in the image [2].

PVD Method for Gray-Level Image -The pixel-value differencing (PVD) method [3] segments the cover image into non overlapping blocks containing two connecting pixels and modifies the pixel difference in each block (pair) for data embedding. A larger difference in the original pixel values allows a greater modification.

3. Spread Spectrum

Spread spectrum communication describes the process of spreading the bandwidth of a narrowband signal across a wide band of frequencies. This can be accomplished by modulating the narrowband waveform with a wideband waveform, such as white noise. After spreading, the energy of the narrowband signal in any one frequency band is low and therefore difficult to detect. SSIS works by storing a message as Gaussian noise in an image. At low noise power levels, the image degradation is undetectable by the human eye, while at higher levels the noise appears as speckles or "snow."

The process consists of the following major steps, as illustrated in figure 1:

1. Create encoded message by adding redundancy via error-correcting code.
2. Add padding to make the encoded message the same size as the image.
3. Interleave the encoded message.
4. Generate a pseudorandom noise sequence, n.
5. Use encoded message, m using advanced encryption standard (AES) to modulate the sequence, generating noise, s.
6. Combine the noise with the original image, f.

Recover the hidden message. A filter is used to extract the noise from the stegoimage, resulting in an approximation of the original image. The better this filter works the fewer errors in the extracted message.

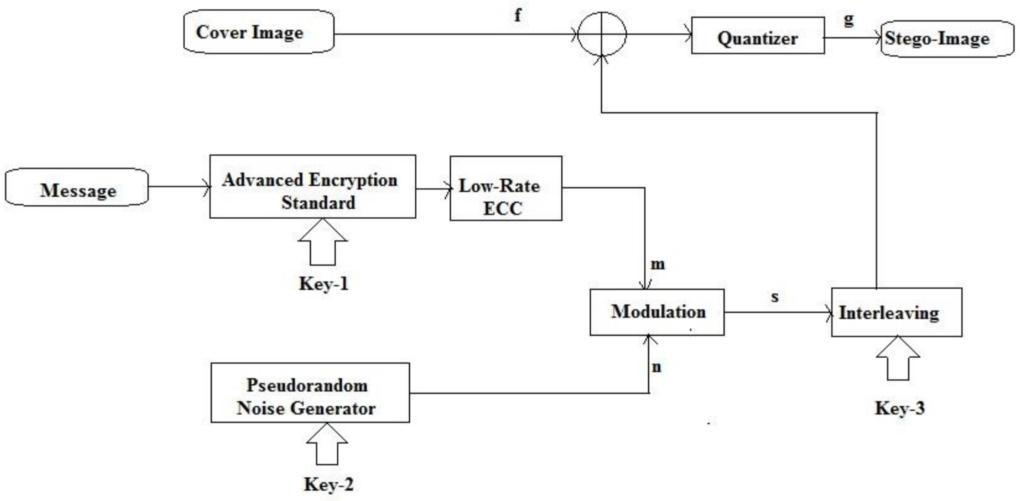


Figure.1. SSISAE Encoder.

The reverse process, of extracting and restoring the original message, is of course very similar and as illustrated in figure 2:

1. Filter the stegoimage, g, to get an approximation of the original image, $\overline{f}$.
2. Subtract the approximation of the original image from the stegoimage to get an estimate of the noise, $\overline{s}$, added by the embedder.
3. Generate the same pseudorandom noise sequence, n.
4. Demodulate by comparing the extracted noise with the regenerated noise.
5. Deinterleave the estimate of the encoded message, $\overline{m}$, using advanced decryption standard (ADS) and remove the padding.
6. Use error-correcting decoder to repair the message as needed.

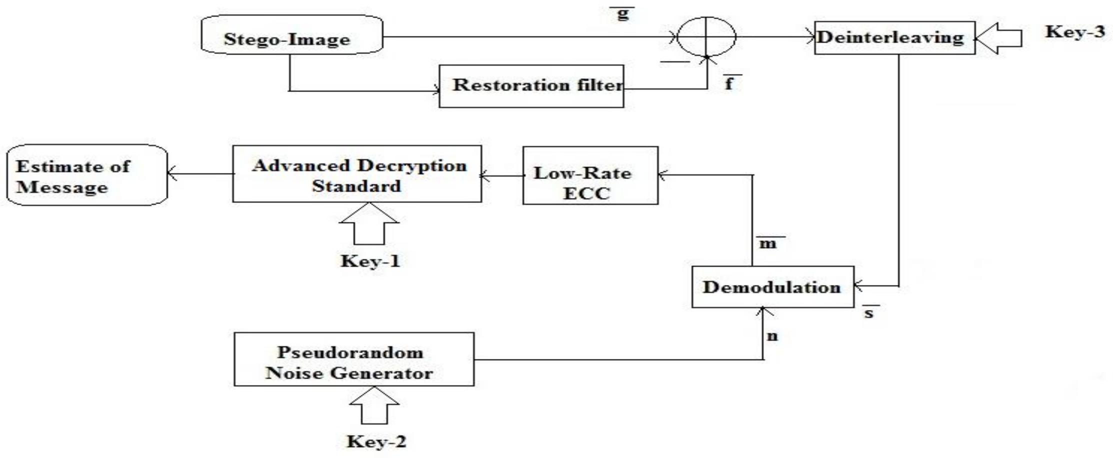


Figure.2. SSISAE Decoder.

4. The Aes Algorithm

The AES algorithm consists of ten rounds of encryption, as can be seen in Figure 3 First the 128-bit key is expanded into eleven so-called round keys, each of them 128 bits in size. Each round includes a transformation using the corresponding cipher key to ensure the security of the encryption.

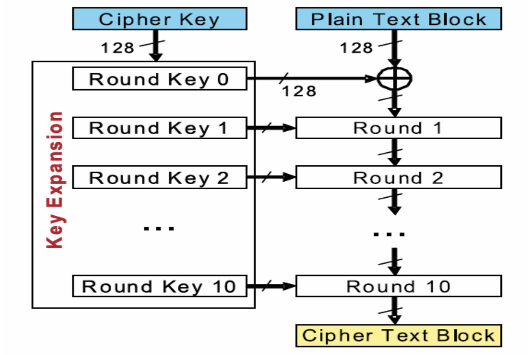


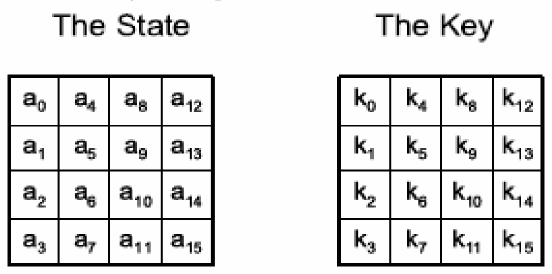
Figure3. AES Algorithm Structure

After an initial round, during which the first round key is XORed to the plain text (Addroundkey operation), nine equally structured rounds follow. Each round consists of the following operations:

- Substitute bytes
- Shift rows
- Mix columns
- Add round key

The tenth round is similar to rounds one to nine, but the Mix columns step is omitted.

A. Structure of Key and Input Data: Both the key and the input data (also referred to as the state) are structured in a 4x4 matrix of bytes. Figure 4 shows how the 128-bit key and input data are distributed into the byte matrices



The State

a_0	a_4	a_8	a_{12}
a_1	a_5	a_9	a_{13}
a_2	a_6	a_{10}	a_{14}
a_3	a_7	a_{11}	a_{15}

The Key

k_0	k_4	k_8	k_{12}
k_1	k_5	k_9	k_{13}
k_2	k_6	k_{10}	k_{14}
k_3	k_7	k_{11}	k_{15}

Figure 4. Structure of the Key and the State

B.Substitute Bytes (Subbytes Operation) : There are different ways of interpreting the Subbytes operation. In this application report, it is sufficient to consider the Subbytes step as a lookup in a table. With the help of this lookup table, the 16 bytes of the state (the input data) are substituted by the corresponding values found in the table (see Figure 5.)

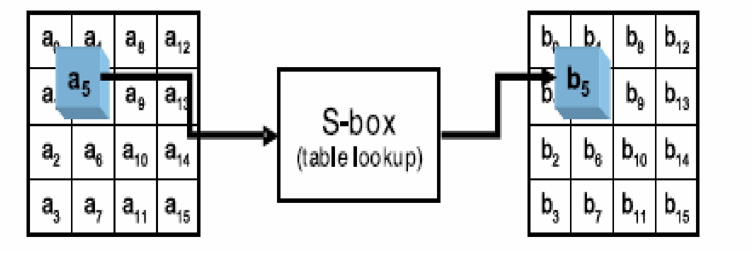


Figure 5.Sub bytes Operation

C. Shift Rows (Shiftrows Operation): As implied by its name, the Shiftrows operation processes different rows. A simple rotate with a different rotate width is performed. The second row of the 4x4 byte input data (the state) is shifted one byte position to the left in the matrix, the third row is shifted two byte positions to the left, and the fourth row is shifted three byte positions to the left. The first row is not changed.

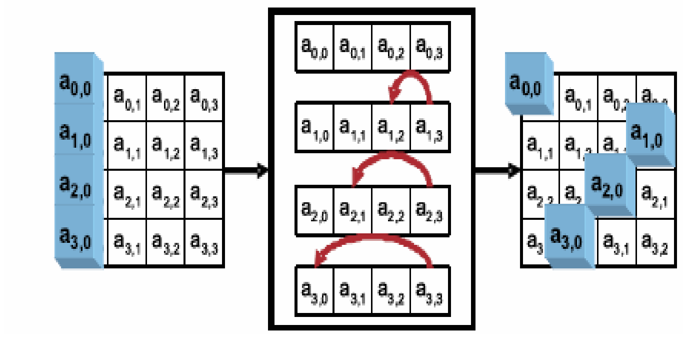


Figure 6. Shiftrows Operation

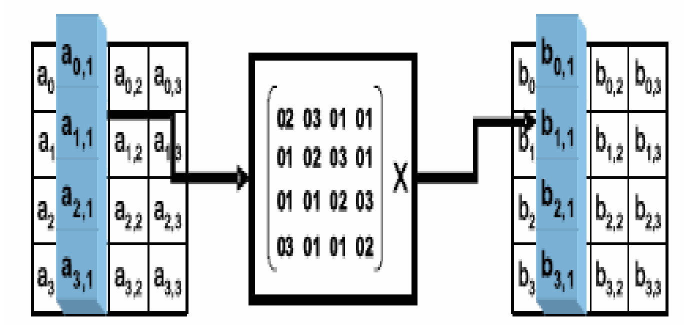


Figure 7. Mixcolumns Operation

D. Mix Columns (Mix columns Operation):

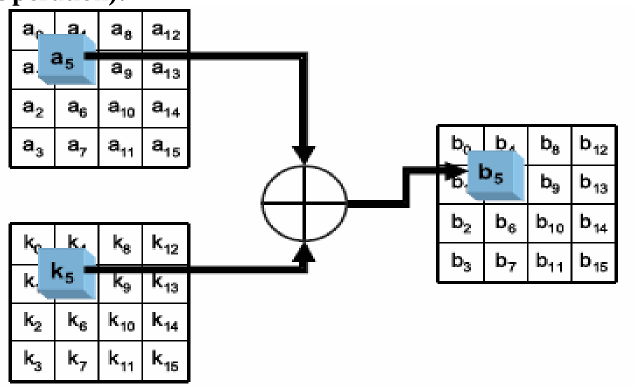


Figure 8.Addroundkey Operation

Opposed to the Shiftrows operation, which works on rows in the 4x4 state matrix, the Mix columns operation processes columns. Transformation in the Cipher that takes all of the columns of the State and mixes their data (independently of one another) to produce new columns shown in figure 7

E. Add Round Key (Addroundkey Operation): The Addroundkey operation is simple. The corresponding bytes of the input data and the expanded key are XORed (see Figure 8).As previously mentioned, Key expansion refers to the process in which the 128 bits of the original key are expanded into eleven 128-bit round keys.

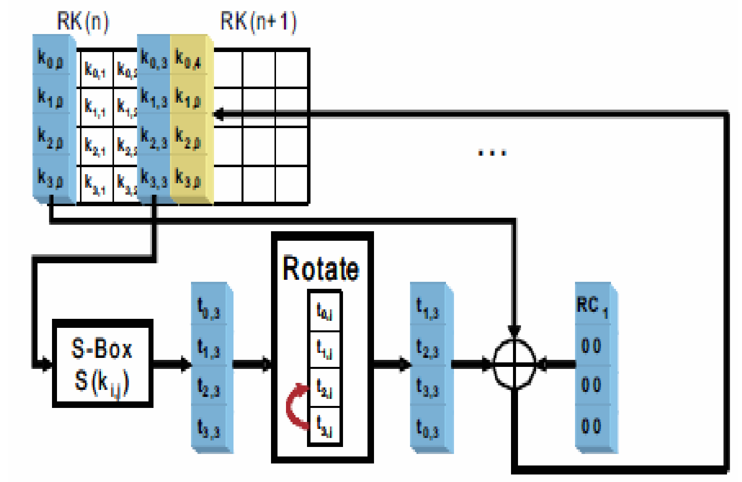


Figure 9. Expanding First Column of Next Round Key

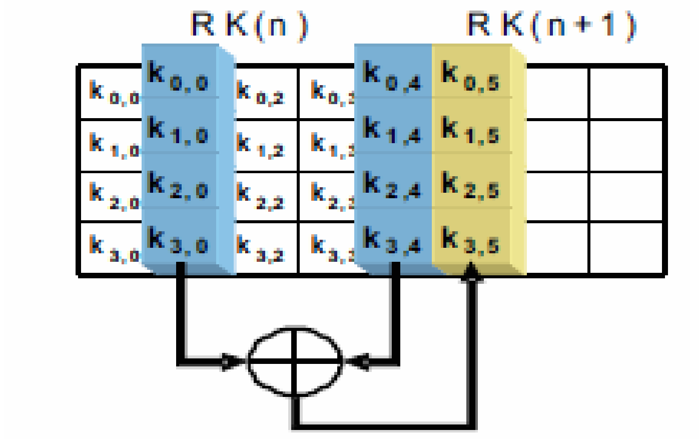


Figure 10.Expanding Other Columns of Next Round Key

1. To compute round key (n+1) from round key (n) these steps are performed: Compute the new first column of the next round key as shown in Figure 9:
First all the bytes of the old fourth column have to be substituted using the Subbytes operation. These four bytes are shifted vertically by one byte position and then XORed to the old first column. The result of these operations is the new first column.

2. Columns 2 to 4 of the new round key are calculated as shown:
- [new second column] = [new first column] XOR [old second column]
- [new third column] = [new second column] XOR [old third column]
- [new fourth column] = [new third column] XOR [old fourth column]

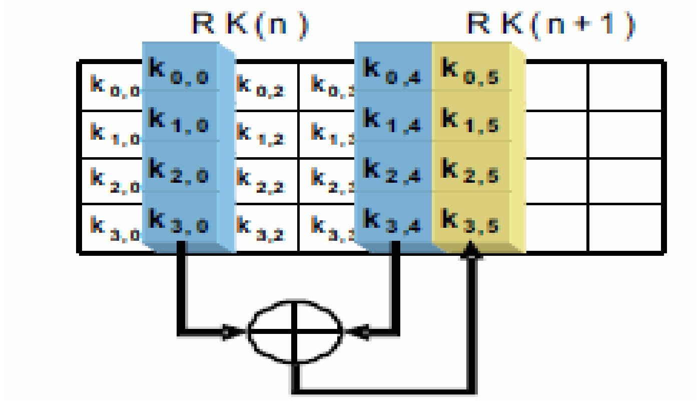


Figure 11. Expanding Other Columns of Next Round Key

F. AES Encryption Flow Chart

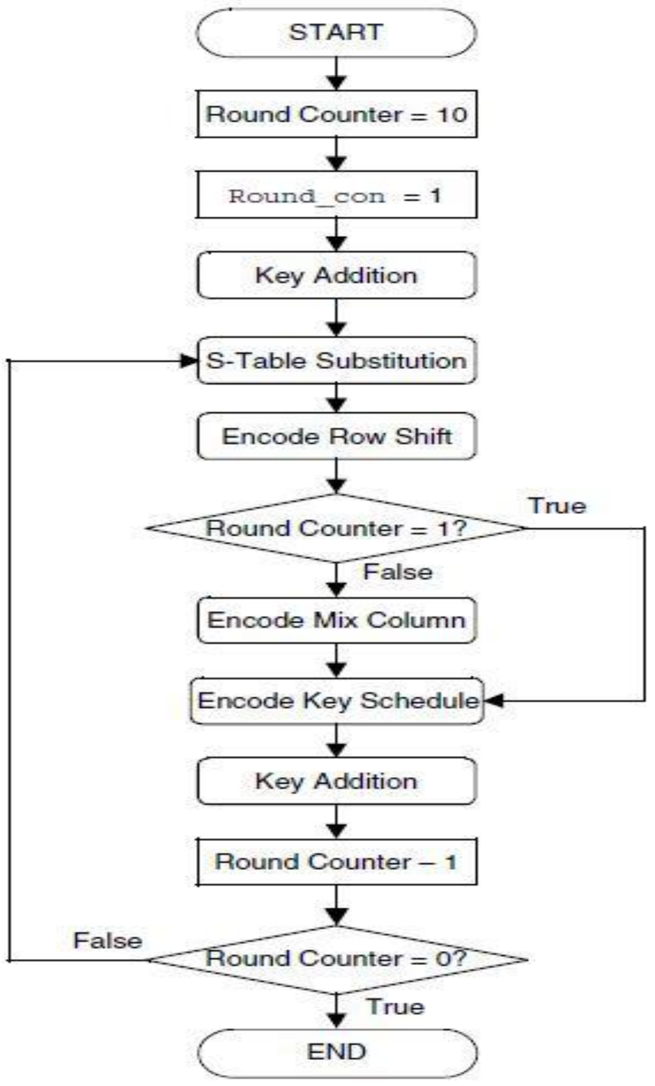


Figure 12.AES Encryption Flow Chart

In the flow chart of AES encryption algorithm round counter for 128 bit is 10. Then all other operation like key addition, Stable Substitution, Encode Row Shift, Encode Mix Column are performed.

5. Aes Key Expansion Pseudo Code

```
Parameters
Nb = 4
Nk = number of doublewords in the cipher key (4, 6, 8 for AES-128, AES-192, AES-256,
resp.)
Nr = number of rounds in the cipher (Nr=10, 12, 14 AES-128, AES-192, AES-256, resp.)
The KeyExpansion routine
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
word tmp
i = 0
while (i < Nk)
w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
i = i+1
end while
i = Nk
while (i < Nb * (Nr+1)]
tmp = w[i-1]
if (i mod Nk = 0)
```

```
tmp = SubWord(RotWord(tmp)) xor RCON[i/Nk]
else
if (Nk > 6 and i mod Nk = 4)
tmp = SubWord(tmp)
end if
w[i] = w[i-Nk] xor tmp
i = i + 1
end while
```

6. Encryption And Decryption Flows

AES encryption and decryption flows use the expanded key (recall that key expansion is independent of the processed data). The encryption/decryption procedure is a back-toback sequence of AES transformations, operating on a 128-bit State (data) and a round key.

The AES Encryption Flow

```
;; Data is a 128-bit block to be encrypted. The round keys are stored in
Round_Key_Encrypt
Tmp = Add Round Key (Data, Round_Key_Encrypt [0])
For round = 1-9 or 1-11 or 1-13:
Tmp = ShiftRows (Tmp)
Tmp = SubBytes (Tmp)
Tmp = MixColumns (Tmp)
Tmp = AddRoundKey (Tmp, Round_Key_Encrypt [round])
end loop
Tmp = ShiftRows (Tmp)
Tmp = SubBytes (Tmp)
Tmp = Add RoundKey (Tmp, Round_Key_Encrypt [10 or 12 or 14])
Result = Tmp
```

The AES Decryption Flow (Using the Equivalent Inverse Cipher)

```
;; Data is a 128-bit block to be decrypt. The round keys are stored in
Round_Key_Decrypt
Tmp = Add Round Key (Data, Round_Key_Decrypt [0])
For round = 1-9 or 1-11 or 1-13:
Tmp = InvShiftRows (Tmp)
Tmp = InvSubBytes (Tmp)
Tmp = InvMixColumns (Tmp)
Tmp = AddRoundKey (Tmp, Round_Key_Decrypt [round])
end loop
Tmp = InvShift Rows (Tmp)
Tmp = InvSubBytes (Tmp)
Tmp = AddRoundKey (Tmp, Round_Key_Decrypt [10 or 12 or 14])
Result = Tmp
```

7. Result And Discussion

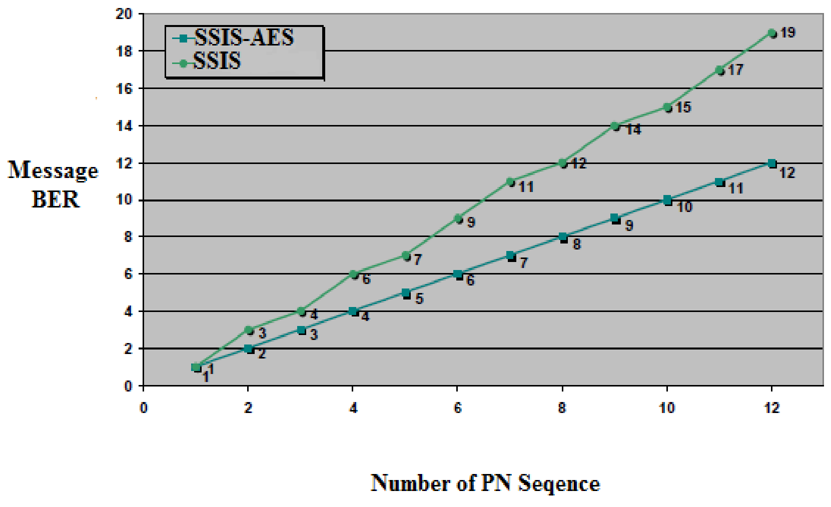


Figure 13.Comparison of the Message BER using SSIS-AES and SSIS

In this work we have implemented the 128 bit AES cryptography algorithm for wireless transmission. We have work on 433 MHz frequency. We have transmitted the data up to a maximum distance of 100m at 4.8 Kbps. If we increase the range then data transmission rate decreases.

It also improving the embedded signal estimation process in order to lower the signal estimation BER so that higher rate error-correcting codes may be employed increase the payload of this system. Additionally, more complex error-correction could be implemented and it may be advantageous to have the embedded signal be adaptive. Finally, the method presented here could be extended to color imagery and audio signals.

Acknowledgment

Support From The Director And Deputy Director Of Prist University-Trichy Campus Are Gratefully Acknowledged

References

[1] Fridrich, J., Goljan, M. and Du, R.., (2001). ―Reliable Detection of LSB Steganography in Grayscale and Color Images.‖ Proceedings of ACM, Special Session on Multimedia Security and Watermarking, Ottawa,Canada, October 5, 2001, pp. 27- 30.

[2] A Novel steganographic Method for Gray-Level Images AhmadT. Al-Taani and Abdullah M. AL-Issa, International Journal of Computer, Information, and Systems Science, and Engineering 3:1 2009

[3] D. C. Wu and W. H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognition Letters, 24(9-10),pp.1613–1626, 2003.

[4] X. Zhang and K. K. Parhi, “Implementation Approaches for the Advanced Encryption Standard

Al gorithm,” IEEE Circuits and Systems Magazine, vol.2, Issue.4, pp. 24-46, Fourth Quarter 2002.

[5] Kean, T. Duncan, A, “ DES key breaking, encryption and decryption in wireless Communication ”August 2008.

[6] Pekka Riikonen , “RSA algorithm.” Nov. 2002.

[7] MARVEL, L. M., BONCELET, CH. G., RETTER, CH. T. Spread spectrum image steganography. *IEEE Trans. On Image Processing*, vol. 8, no. 8, Aug. 1999, p.1075 – 1083.