Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Development of a Robust Steganographic Scheme using a tweaked AES algorithm

Prithvija. M*, Pavithra. S, Kavin Kumar. N.R
Department of CSE
Amrita School of Engineering, Amrita Vishwa Vidyapeetham
Amrita Nagar (P.O), Ettimadai, Coimbatore-Tamil Nadu 641 112, India

Abstract— Spatial domain steganographic algorithms take advantage of the large amount of redundant data that is created in the way that digital images are stored in the spatial domain. However recent research shows that all steganographic system in spatial domain is vulnerable to attacks if we embed more than 10 percent. An analyst can even estimate the length of hidden bits from the histogram. Here we experimented with a steganographic scheme in which we used a compression algorithm and a secure block cipher to the compressed data before embedding. We believe that this will help to protect the stego in the event of successful attack on steganographic system. We used a variable range pixel value differencing to survive histogram attack to some extent. In this paper we report the result of our experiments.

Keywords— Spatial domain, steganography, Block cipher, S-Box, Pixel Value Differencing, Modified PVD

I. INTRODUCTION

The prevalence of the Internet as a mass communication means and the proliferation of digital multimedia circulated via the web has brought the ancient art of steganography into the digital era. Recent years have seen increased interests even in commercial software for using digital media data, such as images, audio and video files. Steganography refers to the science of invisible communication [3]. Unlike cryptography, where the goal is to secure communication from an eavesdropper, steganographic techniques strive to hide the very presence of the message itself from an observer. Encryption could normally be used to encipher the secret message so that the encrypted information can only be deciphered with a secret key. However, due to the appearance of encrypted information and legislation, encryption does not comply with the requirements of inconspicuousness and legality. Since steganography also offers confidentiality of information, image steganography is thus implemented as an alternative.

Alice wishing to send a secret message *m* to Bob. In order to do so, she embeds *m* into a cover object *c* , to obtain the stego object *s* . The stego object *s* is then sent through the public channel. In a pure steganography framework, the technique for embedding the message is unknown to Eve and shared as a secret between Alice and Bob. However, it is generally not considered as good practice to rely on the secrecy of the algorithm itself as suggested by Kerckhoff's principle.

There are two kinds of image steganographic techniques: spatial-domain and frequency-domain based methods. Spatial domain based methods embed messages in the intensity of pixels of images directly. For frequency domain based methods images are first transformed into the frequency domain and then messages are embedded in the transform coefficients. Here we used spatial domain based methods. Spatial domain formats can be divided into raster image formats and palette based image formats. In a raster image format, an image is represented in a row-by-row grid of pixels with one or more bytes used to store one pixel depending on the bit depth. We are working with this type of images. Spatial domain steganography uses images in the spatial domain format for hiding information. It encompass bit-wise methods that apply bit insertion and noise manipulation to embed information Data embedding is done by directly replacing data of the image pixel values with secret information. These algorithms take advantage of the large amount of redundant data that is created in the way that digital images are stored in the spatial domain.

Recent research show that all steganographic system in spatial domain is vulnerable to attacks if we embed more than 10% data. So we added a robust cryptographic module in the system. We used a tweaked version of Rijndael algorithm in CBC mode for encryption. Before encrypting the secret message, we applied a compression algorithm. We used a modified pixel value differencing method for embedding the encrypted data. We showed that the scheme work perfectly. In the following sections, we explain compression module, encryption module and embedding module.

II. COMPRESSION MODULE

Since some content dependent patterns in the original message may reveal the existence of the messages, and embedding more bits of messages will introduce more degradation, the compression module is introduced first to deal with these problems. When working with larger images of greater bit depth, the images tend to become too large to transmit over a standard network connection. In order to display an image in a reasonable amount of time and use a

reasonable amount of space to store the image, techniques must be incorporated to reduce the image's file size. These techniques make use of mathematical formulas to analyse and condense image data, resulting in smaller file sizes. This process is called compression. Two types of image compression methods exist: lossy and lossless.

We used discrete wavelet transform techniques for compression, which is a lossy compression method. Lossy compression creates smaller files by discarding excess image data from the original image. It removes details that are too small for the human eye to differentiate, resulting in close approximations of the original image, although not an exact duplicate.

III. ENCRYPTION MODULE

To raise the security level, a suitable encryption algorithm (modified AES-128) is then applied on the compressed messages to conceal the meaning of messages. This encryption module conceals the content-dependent property, and the compression module is performed prior to the encryption module for the benefit of entropy coding. In this case, Stego keys are the following: The value of β and the table, while the 128 bit key is for the block cipher.

In the situation where the sender and the receiver does not share both the stego key and the secret key, the sender can use the public key or the receiver to encrypt these keys, and then embeds these encrypted keys in fixed positions of the stego-image. When the receiver receives the stego-image, these keys arc extracted and then decrypted using the private key of the receiver.

We used a modified AES algorithm for image encryption. We used Cipher Block Chaining (CBC) mode, since it has several advantages over other modes. This uses an *Initialization Vector* (*IV*) the size of one block. The *IV* is Exclusive-ORed with the first message block before encryption to give the first ciphertext block. Each subsequent message block is Exclusive-ORed with the *previous* ciphertext block. The process is reversed on decryption. At each stage, one ciphertext block is transmitted. It must also be arranged that the same secret key and the same initialization vector *IV* are at both ends of the transmission, although the IV could be included in an initial transmission. The CBC mode is essentially a stream cipher that handles one block's worth of bits at a time. The state or memory in the system is the previous ciphertext. Each ciphertext block depends on the current plaintext and on all plaintext that came before: If a single bit of the initialization vector of the first plaintext block is changed, then all ciphertext blocks will be randomly altered. CBC is secure against the various attacks. This includes all the ways of fiddling with and searching for encrypted blocks.

Since Rijndael algorithm was chosen by NIST as an advanced encryption standard (AES) in 2000, much attention has been attracted to it and many methods have been proposed to attack it. But there has never been successful attack on the full AES up to now. As the only nonlinear operation of AES, S-box plays a crucial role against various attacks.

S-boxes are the most important and the only nonlinear component of a block cipher since diffusion and confusion properties which are related with the security of cryptographic algorithms are added to a block cipher by S-boxes. So, bijective S-boxes play an important role in the design of symmetric ciphers. To date, the techniques for the construction of S-boxes have included pseudorandom generation, finite field inversion, power mappings and heuristic techniques. From these techniques, the use of finite field operation in the construction of an S-box yields linear approximation and difference distribution tables in which the entries are close to uniform. Therefore, this provides security against differential and linear attacks. Some of the desirable cryptographic properties are completeness, avalanche, strict avalanche, bit independence [1]. An $n \times n$ S-box, $S(x) : GF(2^n) \rightarrow GF(2^n)$, maps an n-bit input to an n-bit output and can be viewed as consisting of n Boolean functions. This type of S-boxes, one of which is the AES S-box and maps an 8-bit input to an 8-bitoutput, has been used in most ciphers in the literature. The *S-box* is a simple table lookup with integer values defined by a 16×16 matrix which contains all possible 256 bytes. Each individual byte is mapped into a new byte in the following way: the left most 4-bits are used as a row value and the right most 4-bits are used as column value. This row and column values serve as indices (0 to 15) into the *S-boxes* to select a unique 8-bit output value. It is to be noted that the entries are given in decimal format. These S-boxes also facilitate the cipher not go into spurious (*all zero*) state after key initialization.

We have used an improved S-box in the AES in our scheme for more robustness. We used the technique of power mapping for constructing this S-Box. While constructing this S-Box, we also have taken into the account of the suggestion given in [2]. The improved S-Box has following cryptographic properties: the affine transformation period is increased from 4 to the most 16, the iterative period is increased from less than 88 to the most 256, and the distance to SAC is reduced from 432 to 372. Moreover, the number of terms in the improved AES S-box algebraic expression is increased from 9 to 255, and its inverse S-box keeps almost the same as AES inverse S-box.

A. The S-Box construction

Aiming at very simple algebraic expression of AES S-box, Liu [5] proposed an improved S-box by exchanging order of taking multiplicative inverse and applying affine transformation. The algebraic expression of the S-box involves 255 terms. However, the algebraic expression of the inverse S-box involves only 9 terms. That is, the algebraic expression of the inverse S-box is too simple. However in [2] the construction processes of both S-box and inverse S-box have an affine transformation before taking multiplicative inverse as a result as the result, the number of terms in the improved AES inverse S-box algebraic expression is up to 253.

The improved AES S-box is constructed by the following three steps:

Construction of S-Box for encryption

Select a 8 by 8 matrix M with the rows are m[0] to m[7], where
m[0]=1,1,0,1,1,0,1,0
m[1]=0,1,1,0,1,1,0,1
m[2]=1,0,1,1,0,1,1,0
m[3]=0,1,0,1,1,0,1,1
m[4]=1,0,1,0,1,1,0,1
m[5]=1,1,0,1,0,1,1,0
m[6]=0,1,1,0,1,0,1,1
m[7]=1,0,1,1,0,1,0,1
and
a column vector b = (1, 0, 1, 1, 1, 0, 1, 0) is chosen
Choose the field polynomial g(x) = x^8 + x^4 + x^3 + x + 1
Then find x'= M x + b
S box (x) = m(x'^-1) + b
We get S box as follows (all elements are in decimals)

TABLE I: *S-Box* FOR ENCRYPTION

39	240	59	135	169	115	223	61	24	5	42	90	156	72	136	127
174	249	2	10	27	140	126	63	246	76	46	9	193	41	134	52
130	51	89	168	25	175	143	47	100	102	230	45	161	254	49	68
38	0	210	94	13	88	233	82	48	86	172	239	139	50	116	152
144	3	7	22	79	44	54	108	202	95	131	142	216	133	118	65
20	220	93	190	17	241	55	111	138	228	84	198	11	199	224	74
122	253	107	171	146	77	35	114	101	159	200	141	192	57	81	178
170	209	112	186	234	58	187	194	197	103	145	218	229	67	124	215
214	183	165	160	16	87	235	231	113	26	23	43	36	80	182	173
18	56	104	73	243	99	123	137	98	119	201	237	157	247	117	221
83	97	128	180	255	62	15	211	31	155	252	184	66	125	150	28
105	245	232	179	164	149	154	148	248	212	250	21	188	34	177	225
185	14	205	217	213	204	207	12	158	29	176	208	64	8	19	69
92	153	60	195	238	33	106	151	162	71	203	78	181	236	166	196
222	32	91	120	191	30	147	75	53	37	121	129	242	219	189	167
4	163	110	244	85	40	96	226	109	1	70	6	132	227	206	251

Construction of S-Box for decryption

A= Inverse of M= $[a[0],\ldots\ldots a[7]]^T$ where,

a[0] = [0 1 1 1 0 0 0 0]
a[1] = [0 0 1 1 1 0 0 0]
a[2] = [0 0 0 1 1 1 0 0]
a[3] = [0 0 0 0 1 1 1 0]
a[4]= [0 0 0 0 0 1 1 1]
a[5]= [1 0 0 0 0 0 1 1]
a[6]= [1 1 0 0 0 0 0 1]
a[7]= [1 1 1 0 0 0 0 0]

Let *y* = sbox(x) then
y= m(x'^-1) +b where x'= mx + b

x'= A(y + b)^-1

x=A(x' + b)

S1= inverse(S) is given below

TABLE II: *INVERSE S-BOX* FOR DECRYPTION

49	249	18	65	240	9	251	66	205	27	19	92	199	52	193	166
132	84	144	206	80	187	67	138	8	36	137	20	175	201	229	168
225	213	189	102	140	233	48	0	245	29	10	139	69	43	26	39
56	46	61	33	31	232	70	86	145	109	117	2	210	7	165	23
204	79	172	125	47	207	250	217	13	147	95	231	25	101	219	68
141	110	55	160	90	244	57	133	53	34	11	226	208	82	51	73
246	161	152	149	40	104	41	121	146	176	214	98	71	248	242	87
114	136	103	5	62	158	78	153	227	234	96	150	126	173	22	15
162	235	32	74	252	77	30	3	14	151	88	60	21	107	75	38
64	122	100	230	183	181	174	215	63	209	182	169	12	156	200	105
131	44	216	241	180	130	222	239	35	4	112	99	58	143	16	37
202	190	111	179	163	220	142	129	171	192	115	118	188	238	83	228
108	28	119	211	223	120	91	93	106	154	72	218	197	194	254	198
203	113	50	167	185	196	128	127	76	195	123	237	81	159	224	6
94	191	247	253	89	124	42	135	178	54	116	134	221	155	212	59
1	85	236	148	243	177	24	157	184	17	186	255	170	97	45	164

The new S-Boxes constructed, satisfy all desirable cryptographic properties, showing that they are robust against any attacks. In this paper, Bora Aslan et al.[1] classified 8 × 8 S-boxes based on power mappings according to the DDT and LAT distributions. According to them, the original AES S-box is classified under the class[d], where d = 127. This means that we can construct similar S boxes with same properties if we use any of the following powers: 127, 254 (which is same as -1), 253, 251 247, 239, 223 and 191 in this case also.

IV. EMBEDDING MODULE

All steganographic algorithms have to comply with a few basic requirements. The most important requirement is that it has to be imperceptible. To meet the requirement of imperceptibility and maximize the embedding capacity, three key concepts are used in the embedding module. First, the embedding capacity of each pixel must adapt to local image characteristics, such as contrast and luminance. Secondly, the new grey scale of each embedded pixel should be as close to the original one as possible. Finally, the stego image should not have any artifact.

We tested the robustness of the embedding algorithm by standard steganalytic methods like histogram techniques and Chi-Square techniques and the result is given below. Histogram analysis is used to visualize the changes made to the image histogram due to embedding. Image histogram is a graphical representation of the distribution of colors or grayscales in an image. It has been applied to detect embedding by methods. Although in general visual artifacts are not noticeable by human eyes in the stego-image, changes in the histogram can be easily observed.

We have used a modified version of Pixel Value Differencing (PVD) method for efficient embedding.

Initially we implemented pixel value differencing method as given in the paper [6]. The pixel-value differencing embeds a large amount of secret bits into a still image with high imperceptibility as it makes use of the characteristics of human vision sensitivity. However, a loophole exists in the PVD method. It is susceptible to histogram attack.

In order to make the PVD steganography immune to the histogram analysis, measures have to be taken to eliminate the abnormal steps introduced by data embedding [7]. For this purpose, we pseudorandomly select a parameter $\beta \in [0, 1]$ generated from an embedding key, for each block of two consecutive pixels, and calculate

$$l_k' = l_k + \lfloor \beta . w_k \rfloor$$

$$u_k' = l_{k+1} + \lfloor \beta . w_{k+1} \rfloor - 1$$

$$= u_k + \lfloor \beta . w_{k+1} \rfloor = l_{k+1}' - 1$$

where k is a range index. Thus, instead of the fixed ranges as used in the original PVD method, the new ranges are defined by the varied l_k' and l_{k+1}'. Here we used this method for various β values and concluded that as β approaches the value 1, we get desired results.

V. EXPERIMENTAL RESULTS

We have tested the embedding module on five gray scale images. We tested the quality of stego image with PSNR values and also we subjected the embedding method against histogram attacks and compared against PVD method. PSNR is a standard measurement used in steganography in order to test the quality of the stego images. The higher the value of PSNR, the more quality the stego image will have which is evident from Table IV. Though we got satisfactory results, we are not claiming that our system is secure against all attacks. Recent research shows that all steganographic system in spatial domain is vulnerable to attacks when we embed more than 10%. So currently, much research is going on embedding in transform domain. That is the reason for adding a robust cryptographic module in our system.

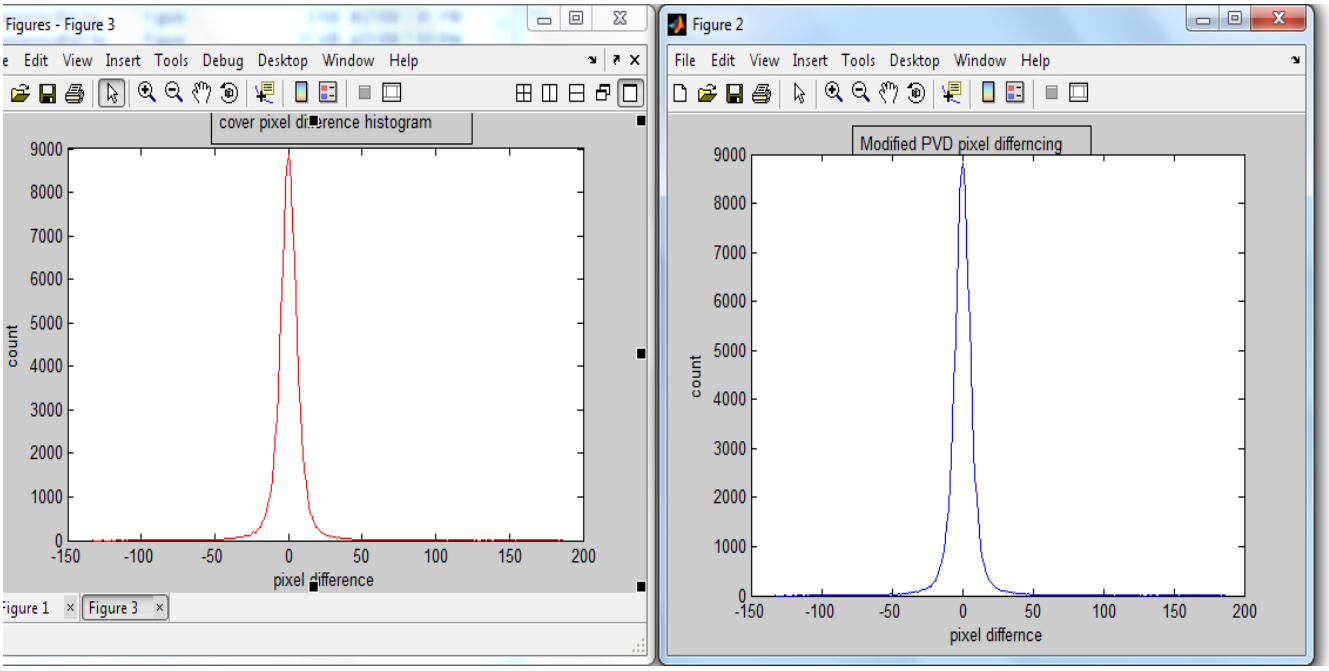
We also analysed robustness of the S-Box constructed against the standard measures [1] and obtained satisfactory results which is given in Table III below. Inverse S-Box also satisfies similar properties

TABLE III: *PROPERTIES OF SUBSTITUTION BOXES*

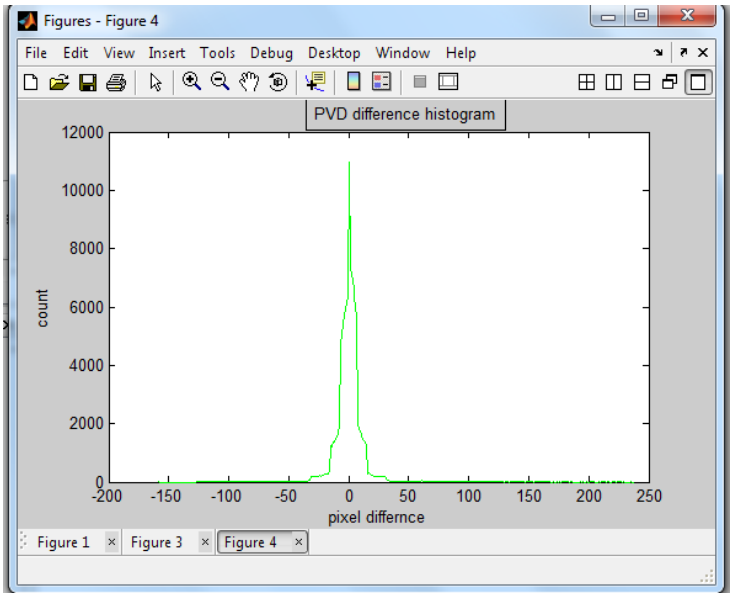
Property	**S box**	**Inv S box**
Nonlinearity	112	112
Differential uniformity	4	4
Balancedness	Yes	Yes
Robustness	0.984375	0.984375
Completeness	Complete	Complete

TABLE IV: *RESULTS OF EMBEDDING MODULES-PSNR VALUES, (RANGE: 8,8,16,32,64,128]*

Cover Image	Payload (No. of bits embedded)	PSNR (dB) (PVD)	PSNR (dB) (Modified PVD with β= 0.1)	PSNR (dB) (Modified PVD with β= 0.5)	PSNR (dB) (Modified PVD with β= 0.8)
Lenagray.BMP (512×512)	28×10^4	42.9862	43.9553	45.2225	45.4724
	28×10^2	65.2371	66.6038	67.2674	67.4590
Fruit.BMP (512×480)	28×10^4	43.7428	45.1092	46.1566	46.4810
	28×10^2	64.8451	66.4285	67.2575	67.2807
Pepper.BMP (512×512)	28×10^3	42.5201	52.2545	55.7485	56.1491
	28×10^2	64.6003	65.2320	66.4315	66.4616
Monarch.BMP (768×512)	28×10^3	56.6642	59.0830	59.2631	59.2982
	28×10^2	65.5001	69.2015	69.5165	69.6289
Baboon.BMP (500×480)	28×10^3	47.2936	48.3264	49.6191	49.9681
	28×10^2	57.3130	58.6168	60.0752	61.1447



a) Cover pixel difference histogram b) modified PVD pixel difference histogram



c) PVD pixel difference histogram

FIGURE 1

PIXEL DIFFERENCE HISTOGRAM FOR PEPPER.BMP, PEPPERMODPVD.BMP AND PEPPERPVD.BMP

Figure 1 illustrates that histogram attack is difficult in modified PVD compare to PVD.

VI. CONCLUSIONS

In this paper we report the result obtained from our secure steganographic system. The notable contribution is the integration of compression, encryption and embedding algorithms. We could successfully use a tweaked version of Rijndael algorithm to protect the secret in the event of successful attack on steganographic system. We used a variable range pixel value differencing to survive histogram attack to some extent.

REFERENCES

[1] Aslan, Bora, M. Tolga Sakalli, and Ercan Bulus. "Classifying 8-bit to 8-bit S-boxes based on power mappings from the point of DDT and LAT distributions." *Arithmetic of Finite Fields*. Springer Berlin Heidelberg, 2008. 123-133.

[2] Cui, Jie, et al. "An improved AES S-Box and its performance analysis." *International Journal of Innovative Computing, Information and Control* 7.5 (2011).

[3] Fridrich, Jessica. " Steganography in digital media: principles, algorithms, and applications". Cambridge University Press, 2010.

[4] Wagner, Neal R. "The Laws of Cryptography with Java Code." *Available online at Neal Wagner's home page* (2003).

[5] J. M. Liu, B. D. Wei and X. M. Wang, One AES S-box to increase complexity and its cryptanalysis, Journal of Systems Engineering and Electronics, vol.18, no.2, pp.427-433, 2007.

[6] Wu, Da-Chun, and Wen-Hsiang Tsai. "A steganographic method for images by pixel-value differencing." *Pattern Recognition Letters* 24.9 (2003): 1613-1626.

[7] Zhang, Xinpeng, and Shuozhong Wang. "Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security." *Pattern Recognition Letters* 25.3 (2004): 331-339.

