Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 10, October -2017

N-tier Encryption/Decryption for Data Security

[1]Prity Kumari, [2]Upendra Kumar, [3]Shyam Krishna Singh

[1]Magadh University, Bodh Gaya, India
[2]Birla Institute of Technology, Patna Campus, Patna, India,
[3]A. N. College, Patna, India

Abstract:- *Today data and information security is most challenging work in computer network system. In this paper we integrate two asymmetric algorithm such as N^{th} prime RSA-CRT with LSB Stegnography and ECC ,one symmetric algorithm AES, SHA-1 message digest function, and ECMQV for encrypt/decrypt data to extend the level of security. We uses Elliptic Curve Cryptography (ECC) for key generation, AES and N^{th} prime RSA-CRT with LSB Stegnography algorithm for encryption, ECDSA for authentication, SHA-1 for integrity, and ECMQV for key exchange. It provide high amount of confidentiality, data integrity and privacy. RSA algorithm is enhanced by combination of CRT with LSB Stegnography algorithm. Stegnography is the science that hiding the information in images involves communicating secret data in an appropriate multimedia carrier, e.g., image, audio, and video files. Due to this proposed system is much more efficient than existing system. This new security protocol has been designed for better security with integrity using a combination of both symmetric and asymmetric cryptographic techniques.*

Keywords: *Enhanced AES ; N^{th} prime RSA-CRT; LSB Stegnography; ECC ; SHA-1; ECDSA; ECMQV.*

1. Introduction

Cryptanalyst are expert in how to break the encryption techniques. We need to safe our programs and documents from cryptanalyst. Security of information means protecting data from unauthorized access in transmission network. There are many techniques to achieve the security of information from unauthorized access. There are two cryptographic techniques used for data encryption which are Symmetric and Asymmetric techniques.
In this paper we use symmetric cryptography like advanced AES, asymmetric cryptography like N^{th} prime RSA-CRT(Rivest-shamir-adlemen-Chinese Remainder Theorem) with LSB(Least significant bit) stegnography, ECC, AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. The AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits. Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES- 256 respectively. A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the 128 bit key. The input is a single 128 bit block both for decryption and encryption and is known as the **in** matrix. This block is copied into a **state** array which is modified at each stage of the algorithm and then copied to an output matrix . Both the plaintext and key are depicted as a 128 bit square matrix of bytes. This key is then expanded into an array of key schedule words (the **w** matrix). It must be noted that the ordering of bytes within the **in** matrix is by column. The same applies to the **w** matrix.
RSA cryptosystem [1] was designed in 1977 by Ron Rivest, Adi Shamir and Leonard Adlemen. It is very popular public key algorithm because of its simplicity. The security lies on computing the factors of a large composite integer. Currently factoring 1024 bits integer is assumed to be as complex as workload of 280 which is the current benchmark used in cryptography. The high computational cost and memory usage bring the algorithm in research area. Work is done on various parameters to improve the efficiency of the algorithm. A more sophisticated way to reduce the decryption time is to employ the Chinese Remainder Theorem (CRT) [19].

Steganography is the more conservative technology to hide any secret information within an image. The given data is embedded into an image to hide its data. Steganography is an art of sending hidden data or secret messages over a public channel so that a third party cannot detect the presence of the secret messages Steganography is an art of secret communication. The term hiding refers to the process of making the information imperceptible or keeping the existence of the information secret. The Steganography algorithm were primarily developed for digital images and video sequences. Interest in research in audio Steganography started. Any Steganography Technique has to satisfy two basic requirements. The first requirement is perceptual transparency that is cover object (object containing any additional data)

and stego object (object containing secret messages) must be perceptually indiscernible. The second constraint is high data rate of the embedded data. In this project we will hide data or secret message in an audio file using two algorithms.
One algorithm for providing security to the secret message by encrypting it.So the algorithm used for encryption is RSA algorithm.Other for embedding secret message in audio we will use multiple LSB algorithm[4].

SHA1 hashing algorithm is a cryptographic algorithm that can be used to provide data integrity and authentication. They are also typically used in password based systems to avoid the need to store plaintext passwords.if we are using SHA-1 for the storage of passwords, there are no password-recovery attacks as at December 2011 that make use of the collision attacks on SHA-1.
ECMQV has all the security attributes desired in a key agreement protocol, making it trusted and secure. ECMQV also has many desirable performance attributes, including that the dominant computational steps are not intensive, the protocol has low communication overhead, is role-symmetric, noninteractive, and does not use encryption or time-stamping.
While based on DH, ECMQV offers attributes that are not found with DH. This allows protocols that use ECMQV for key agreement to offer stronger authentication and ensure that malicious entities cannot masquerade as a third party to the entity whose key was compromised. When it is computed using Elliptic Curve Cryptography, ECMQV offers significant performance advantages over other key establishment schemes, which makes it ideal in the development of security protocols and systems that require efficient and authenticated key agreement.

2. State of Arts

Chauhan [1] present that hybrid cryptography is better approach to maintain confidentiality and privacy of information during communication. They also proposed a hybrid algorithm for strong encryption. They state that there are various algorithms are available for cryptography but all of them have certain drawbacks. Proposed algorithm is designed with combination of two symmetric algorithm techniques known as AES and DES. Proposed solution is implemented using 128 bit keys. Proposed solution is implemented using java technology. Here, they provide facility to select security algorithm as per requirement which may be AES, DES or hybrid algorithm. The complete work concludes that possibility of an algebraic attack on hybrid model is too poor and gives strong strength to encryption approach.
Several steganographic methods have been proposed in literature and most of which are performed in pixel domain. However major contribution is in the domain of Image Steganography. The existing methods are mainly based on LSB where LSBs of the cover file are directly changed with message bits. In [7] a robust image steganography technique based on LSB insertion and RSA encryption technique has been used. Masud et.al [8] has proposed a LSB
technique for RGB true color image by enhancing the existing LSB substitution techniques to improve the security level of hidden information. Other Examples of LSB schemes can be found in [9], [10]. Whereas EzStego developed by Machado [11] embed information into an image in the GIF format. It sorts the palette to ensure the difference between two adjacent colors is visually indistinguishable. Tseng and Pan [12] presented a data hiding scheme in 2-color images, it embeds the information in any bit where at least one of the adjacent bits is the same as the original unchanged bit. Kawaguchi et. al. [13] proposes bit plane complexity segmentation (BPCS) method to embed information into the noisy areas of the image. These techniques are not limited to the LSB. Existing steganographic software, such as Steganos, S-tools and Hide4PGP [14], are based on LSB.

Shankar [2] address that RSA is one of the most common algorithm for encryption and decryption. Subsequently, Round-robin scheduling is one of the most common useful algorithms for task scheduling and processing. Authors proposed a technique to integrate RSAA algorithm with Round robin scheduling algorithm to extend level of security. In this approach method uses RSA algorithm to generate cipher text based on priority. Receiver receive message and decrypt the message according to priority. Proposed method reduces probability of man-in-middle attack and timing attack.
Subasree[3] explore that computer network is an group of interconnected nodes. Various security threats attempt to compromise the network security and modify the content of packet. Confidentiality, authentication and integrity are the most crucial security principle used to maintain level of security. It requires the certain security algorithms to maintain security and maintain communication private. Proposed.
algorithm integrates Elliptic Curve Cryptography, Dual RSA algorithm and Message Digest MD5. This new security protocol has been designed for better security with integrity using a combination of both symmetric and asymmetric cryptographic techniques.
Tianfu [5] address that internet is one of the most unsafe communication medium due to huge connection and public network. Information protection is one the of essential requirement. At present various security algorithms are proposed to achieve security during communication. All of them have certain good point and certain bad point. To improve the strength of encryption algorithm they proposed a hybrid model. Proposed model is combination of AES and DES. Both algorithms are symmetric key technique and itself they are very much capable for encryption. Integration of AES and DES would give a strong level of security at encryption end. A significant improvement in results has been observed with proposed solution.

3. Hybrid Cryptosystem

Combines strengths of both methods symmetric and asymmetric-key cryptosystems. Asymmetric distributes symmetric key, also known as a session key. Symmetric provides bulk encryption. The example of a hybrid cryptosystem is SSL[23]

The shortcomings of a symmetric-key algorithm:

- Key-exchange becomes a problem.
- Trust problems among the intended parties occur.
- More damage is caused, when someone gets their hands on a symmetric key because they can decrypt everything encrypted with that key.
- As the number of participants using the secret key increases, the risk of damage and the consequences of this damage increases. [22]

The shortcomings of a asymmetric-key algorithm:

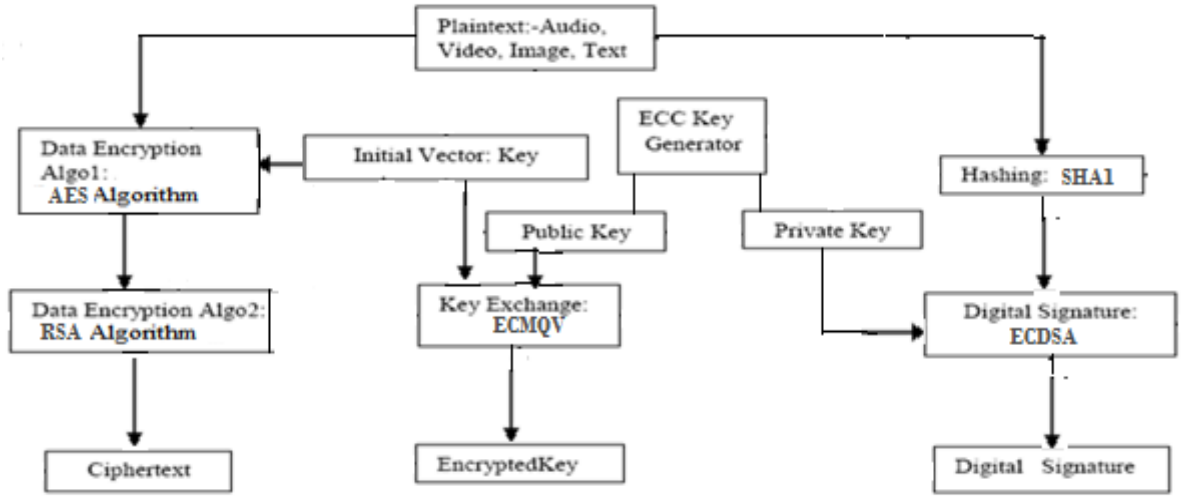
- Since asymmetric-keys must be many times longer than the secret-key in symmetric-key algorithm, asymmetric-keys are more computationally costly.
- They are susceptible to attacks in less than brute-force time.
- It is also vulnerable to man in the middle attack.
- Many public key systems use a third party to certify the reliability of public keys. [22]

The message-digest or one-way hashing functions were then proposed as an alternative to fulfill all the aspects of information security because of the following features:

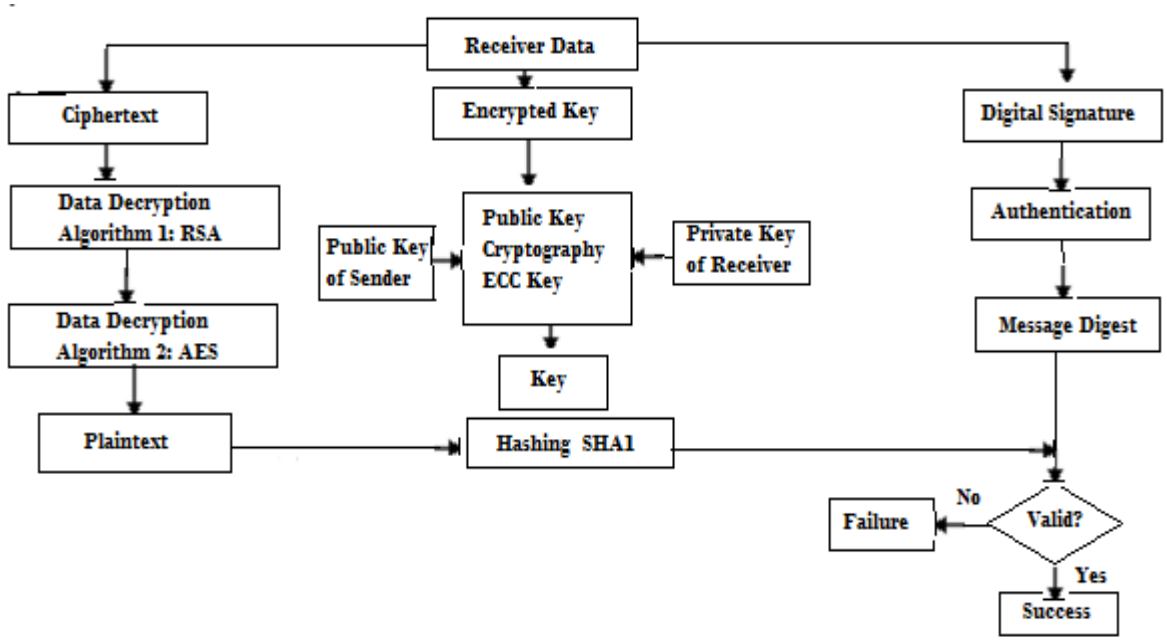
- It is computationally easy to calculate the hash of any given message.
- With the same hash, there can never be two messages associated with it.
- Message cannot be changed without any changes in the hash value.
- It is infeasible to generate the message with the given hash value. [23] The two cryptographic hashing algorithms widely known are:
  - MD5
  - Security Hash Algorithm (SHA).

4. Proposed Technique

Mix of improved AES and ECC encryption algorithm, can not only enhance the speed of data encryption and decryption, but also solute the problem of key distribution and authentication. It has high computing speed and anti-attack capability, which improved the security of data transmission process effectively [20].



1. Take plain text or any files such as video, image or text as input.
2. Generate private and public key by using ECC generator.
3. Choose a random generated key.
4. Apply AES algorithm on plaintext to encrypt it. AES use random generated key to encrypt message.
5. Again encrypt, encrypted message using N^{th} prime RSA-CRT with LSB Stegnography and give cipher text.
6. Apply ECMQV algorithm for key exchange using ECC public key and random generated key and give encrypted key.
7. Apply hash algorithm SHA1 on plaintext to generate fix size of message and then apply digital signature ECDSA using ECC private key to generate digital signature.
8. Now send cipher text, encrypted key and digital signature to destination.



Receiver's System Steps are as follows
The receiver system decrypts the received encrypted data by two decryption algorithms AES and RSA. Key to encrypt the data gets from the applying decryption of ECC algorithm (ECMQV). That is private key of sender and public key receiver. Result gives the plain text. Apply the message digest algorithm on this plaintext data. The message digest that we get from plain text at receiver and received message digest of size 128 bit are compare with each other to validate. If both the message digest are same the data get accepted else if the both are differ then the received data get discarded.
Receiver system receives three files

1) Encrypted Key, 2) Encrypted data, and 3) Digital signature.
   Digital signature authenticates the sender and gives the message digest.

1. Receive encrypted file along with key and perform cryptanalysis on it.
2. Then we will be having 3 blocks.
   a. Cipher text block
   b. AES key block
   c. Signature block
3. Apply Private Key of receiver on AES key block it will provide AES key.
4. Apply public key on Signature block for authentication which will generate Abstract result.
5. Apply AES key on cipher text block which will give plain text and then abstract result.
6. Compare both step 4 output and step 5 outputs.

a. If the comparison is found consistent then
i. Grant access
b. Otherwise
ii. Failure

Following algorithm will be used to develop hybrid security model.

1. Elliptic Curve Menezes-Qu-Vanstone Key Exchange Algorithm
2. RSA algorithm for Confidentiality
3. Private Key Encryption for Authentication
4. SHA-1 for Integrity
5. ECDSA to provide confidentiality over cipher text and message digest

5. Type of Cryptographic Algorithm are in used

5.1 Generate Private and Public Key of ECC key generator

Key pair generation

- Randomly select $d \in [1, n-1]$.
- Compute $Q=dP$, P, Q is a point on the curve

Public key is Q, private key is d

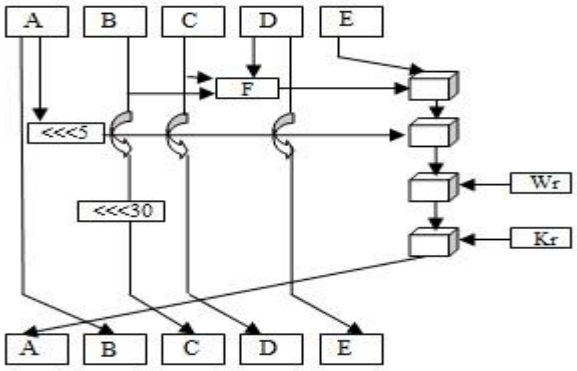
5.2 SHA1(Secure Hash Algorithm)

Input:- message of arbitrary length.

Output:- 160 bit hash code.

The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words).In case the message is not an integer multiple of 512-bit blocks, the message is padded so that its length is divisible by 512[5].

The padding works as follows:- Pad the message with a single 1 followed by 0's until the final block has 448 bits and append the size of the original message as an unsigned 64-bit integer.



One iteration within the SHA-1 compression function: A, B, C, D and E are 32-bit words of the state; F is a nonlinear function that varies; n denotes a left bit rotation by n places; n varies for each operation; Wt is the expanded message word of round t; Kt is the round constant of round t; denotes addition modulo 232.

The SHA algorithm operates on a 160-bit state, divided into five 32-bit words, denoted h0, h1, h2, h3, and h4. These are initialized to certain fixed constants.

h0 =0x67452301

h1= 0Xefcdab89

h2=0x98BADCFE

h3=0x10325476

h4=0XC3D2E1F0 [3]

The main algorithm then uses each 512-bit message block in turn to modify the state. The processing of a message block consists of 80 similar operations based on a non-linear function F, modular addition and left rotation.

A=h0, B=h1, C=h2, D=h3, E=h4

From iteration 16 to 79

w[i]= (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16]) leftrotate1

There are four possible functions F; a different one is used in each round:

F(B,C,D)=(B AND C) OR (NOT B AND D)

G(B,C,D)=B XOR C XOR D

H(B,C,D)=(B AND C) OR (B AND D) OR (C AND D) [5]

I(B,C,D)=B XOR C XOR D

SHA1 requires 80 processing constant words defined as:

K(t) = 0x5A827999 , (0 <= t <= 19)

K(t) = 0x6ED9EBA1, (20 <= t <= 39)

K(t) = 0x8F1BBCDC ,(40 <= t <= 59)

K(t) = 0xCA62C1D6, (60 <= t <= 79)

5.3 Digital Signature (ECDSA)

Alice	**Bob**
Private key d_A, Public key $Q_A=d_AP$. *Signature generation* 1.Select a random k from [1, n-1] 2.Compute $kP=(x_1,y_1)$ and $r=x_1 mod\ n$. if $r=0$ goto step 1 3.Compute $e=H(m)$, where H is a hash functon, m is the message. 4.Compute $s=k^{-1}(e+d_Ar)\ mod\ n$. If $s=0$ go to step 1. (r, s) is Alice's signature of message m	*Signature verification* 1.Verify that r, s are in the interval[1,n-1] 2.Compute $e=H(m)$, where H is a hash function, m is the message. 3.Compute $w=s^{-1}\ mod\ n$ 4.Compute $u_1=ew\ mod\ n$ and $u_2=rw\ mod\ n$. 5.Compute $X=u_1P+u_2Q_A=(x_1,y_1)$ 6.Compute $v=x_1\ mod\ n$ 7.Accept the signature if and only if $v=r$

Alice transfer m,[r,s] to Bob then Bob verify the signature. If signature verification is true then message transformation is success otherwise its failure.
e=H(m)=0xED076287532E86365E841E92BFC50D8C
Proof the correctness of ECDSA:

- Proof

If a signature (r,s) on a message m was authentic, then $s=k^{-1}(e+d_Ar)\ mod\ n$. It can be rewritten as:
$k \equiv s^{-1}(e+d_Ar) \equiv s^{-1}e+s^{-1}rd_A$
$\equiv we+wrd_A$
$\equiv u_1+u_2d_A \pmod n$
Thus $X=u_1P+u_2Q_A=(u_1+u_2d_A)P=kP$.
So v=r is required.

6. Apply Nth Prime RSA-CRT with LSB Stegnography with AES on plain text by using key to generate cipher text

6.1 AES

There are terms that are frequently used throughout this paper that need to be clarified.
Key:- Key with variable length (128,192, 256 bit)
• Represented with a matrix (array) of bytes with 4 rows and N_k columns, N_k=key length / 32
• key of 128 bits= 16 bytes N_k=4
• key of 192 bits= 24 bytes N_k=6
• key of 256 bits= 32 bytes N_k=8
Block:- AES is a block cipher. This means that the number of bytes that it encrypts is fixed. AES can currently encrypt blocks of 16 bytes at a time; no other block sizes are presently a part of the AES standard. If the bytes being encrypted are larger than the specified block then AES is executed concurrently. This also means that AES has to encrypt a minimum of 16 bytes. If the plain text is smaller than 16 bytes then it must be padded. Simply said the block is a reference to the bytes that are processed by the algorithm.
Block of length 128 bits=16 bytes
• Represented with a matrix (array) of bytes with 4 rows and N_b columns, N_b=block length / 32
• Block of 128 bits= 16 bytes N_b=4
State:- Defines the current condition (state) of the block. That is the block of bytes that are currently being worked on. The state starts off being equal to the block, however it changes as each round of the algorithms executes. Plainly said this is the block in progress.
XOR Refers to the bitwise operator Exclusive Or. XOR operates on the individual bits in a byte in the following way:
0 XOR 0 = 0
1 XOR 0 = 1
1 XOR 1 = 0
0 XOR 1 = 1
For example the Hex digits D4 XOR FF
11010100 XOR 11111111 = 00101011 (Hex 2B)
Another interesting property of the XOR operator is that it is reversible. So Hex 2B XOR FF = D4
AES Example:

- **Substitute bytes**

- if we had the plain text "Hello World!....".

H	o	r	.
e		l	.
l	W	d	.
l	o	!	.

AES-128 block example

- it would translate to the hexadecimal value of plain text by using an ASCII lookup table.
Hello World!.... –

0x48	0x6F	0x72	0x2E
0x65	0x20	0x6C	0x2E
0x6C	0x57	0x64	0x2E
0x6C	0x6F	0x21	0x2E

- The first step to a round is to do a byte by byte substitution with a lookup table called an S-box. An S-box is a one to one mapping for all byte values from 0 to 255. The S-box is used to change the original plain text in bytes to cipher text. The S-box is shown below in Figure. All values are represented in hexadecimal notation. This is how the S-box is commonly viewed.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

S-Box. From http://www.samiam.org/s-box.html

When using the S-box the first digit in hexadecimal represents the rows of the table or the values on the left side going down. The second digit represents the column of the S-box, which, in this example, is 48. Using the S-box we find row number 4 and column number 8 and find that the hexadecimal value in that cell is "52" and so on. i.e. the value after S-box replacement is

0x52	0xa8	0x40	0x31
0x4d	0xb7	0x50	0x31
0x50	0x5b	0x43	0x31
0x50	0xa8	0xfd	0x31

Block / State after S-box substitution

- **Rijndael Shifts rows**

  The rows are shifted x number of bytes to the left where x is the row number. This means row 0 will not be shifted, row 1 will be shifted 1 byte to the left, row 2 will be shifted 2 bytes to the left, and so on.

0x52	0xa8	0x40	0x31
0xb7	0x50	0x31	0x4d
0x43	0x31	0x50	0x5b
0x31	0x50	0xa8	0xfd

Block/State after shifting

- **Rijndael MixColumn**

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

0x52	0xa8	0x40	0x31
0xb7	0x50	0x31	0x4d
0x43	0x31	0x50	0x5b
0x31	0x50	0xa8	0xfd

- **Galois Field Multiplication**

The multiplication mentioned above is performed over a Galois Field .Which can be done quite easily with the use of the following two tables in (HEX). The only trick being that if the addition result is greater than FF we subtract FF from the addition result.
For example AF+B7= 166. Because 166 > FF, we perform: 166-FF which gives us 67.

E Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	03	05	0F	11	33	55	FF	1A	2E	72	96	A1	F8	13	35
1	5F	E1	38	48	D8	73	95	A4	F7	02	06	0A	1E	22	66	AA
2	E5	34	5C	E4	37	59	EB	26	6A	BE	D9	70	90	AB	E6	31
3	53	F5	04	0C	14	3C	44	CC	4F	D1	68	B8	D3	6E	B2	CD
4	4C	D4	67	A9	E0	3B	4D	D7	62	A6	F1	08	18	28	78	88
5	83	9E	B9	D0	6B	BD	DC	7F	81	98	B3	CE	49	DB	76	9A
6	B5	C4	57	F9	10	30	50	F0	0B	1D	27	69	BB	D6	61	A3
7	FE	19	2B	7D	87	92	AD	EC	2F	71	93	AE	E9	20	60	A0
8	FB	16	3A	4E	D2	6D	B7	C2	5D	E7	32	56	FA	15	3F	41
9	C3	5E	E2	3D	47	C9	40	C0	5B	ED	2C	74	9C	BF	DA	75
A	9F	BA	D5	64	AC	EF	2A	7E	82	9D	BC	DF	7A	8E	89	80
B	9B	B6	C1	58	E8	23	65	AF	EA	25	6F	B1	C8	43	C5	54
C	FC	1F	21	63	A5	F4	07	09	1B	2D	77	99	B0	CB	46	CA
D	45	CF	4A	DE	79	8B	86	91	A8	E3	3E	42	C6	51	F3	0E
E	12	36	5A	EE	29	7B	8D	8C	8F	8A	85	94	A7	F2	0D	17
F	39	4B	DD	7C	84	97	A2	FD	1C	24	6C	B4	C7	52	F6	01

L Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	19	01	32	02	1A	C6	4B	C7	1B	68	33	EE	DF	03	
1	64	04	E0	0E	34	8D	81	EF	4C	71	08	C8	F8	69	1C	C1
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	09	78
3	65	2F	8A	05	21	0F	E1	24	12	F0	82	45	35	93	DA	8E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	38
5	66	DD	FD	30	BF	06	8B	62	B3	25	E2	98	22	88	91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28	54	FA	85	3D	BA
7	2B	79	0A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	0B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	0C	F6	6F	17	C4	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	6C	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	BC	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2E	89	B4	7C	B8	26	77	99	E3	A5
F	67	4A	ED	DE	C5	31	FE	18	0D	63	8C	80	C0	F7	70	07

Mix Column Example During Encryption

Input = 52 b7 43 31

Output(0) = (52 * 2) XOR (b7*3) XOR (43*1) XOR (31*1)
= E(L(52) + L(02)) XOR E(L(b7) + L(03)) XOR 43 XOR 31
= E(FD + 19) XOR E(86 + 01) XOR 43 XOR 31
= E(116-FF=17) XOR E(87) XOR 43 XOR 31
= A4 XOR C2 XOR 43 XOR 31
= **14**

Output(1) = (52 * 1) XOR (b7*2) XOR (43*3) XOR (31*1)
= 52 XOR E(L(b7)+L(02)) XOR E(L(43)+L(03)) XOR 31
= 52 XOR E(86+19) XOR E(BD+01) XOR 31
= 52 XOR E(9F) XOR E(BE) XOR 31
= 52 XOR 75 XOR C5 XOR 31
= **D3**

Output(2) = (52 * 1) XOR (b7*1) XOR (43*2) XOR (31*3)
= 52 XOR b7 XOR E(L(43)+L(02)) XOR E(L(31)+L(03))
= 52 XOR b7 XOR E(BD+19) XOR E(2F+01)
= 52 XOR b7 XOR E(D6) XOR E(30)
= 52 XOR b7 XOR 86 XOR 53
= **30**

Output(3) = (52 * 3) XOR (b7*1) XOR (43*1) XOR (31*2)
= E(L(52)+L(3)) XOR b7 XOR 43 XOR E(L(31)+L(02))
= E(FD+01) XOR b7 XOR 43 XOR E(2F+19)
= E(FE) XOR b7 XOR 43 XOR E(48)
= F6 XOR b7 XOR 43 XOR 62

= **60**

Mix Column Inverse

During decryption the Mix Column the multiplication matrix is changed to:

0E 0B 0D 09
09 0E 0B 0D
0D 09 0E 0B
0B 0D 09 0E

Mix Column During Decryption

Input 14 D3 30 60
Output(0) = (14 * 0E) XOR (D3*0B) XOR (30*0D) XOR (60*09)
= E(L(14)+L(0E)) XOR E(L(D3)+L(0B)) XOR E(L(30)+L(0D)) XOR E(L(60)+L(09))
= E(34+DF) XOR E(3C+68) XOR E(65+EE) XOR E(7E+C7)
= E(113-FF) XOR E(A4) XOR E(153-FF) XOR E(145-FF)
= E(14) XOR E(A4) XOR E(54) XOR E(46)
= D8 XOR AC XOR 6B XOR 4D
= **52**
Output(1) = (14 * 09) XOR (D3*0E) XOR (30*0B) XOR (60*0D)
= E(L(14)+L(09)) XOR E(L(D3)+L(0E)) XOR E(L(30)+L(0B)) XOR E(L(60)+L(0D))
= E(34+C7) XOR E(3C+DF) XOR E(65+68) XOR E(7E+ EE)
= E(FB) XOR E(11B-FF) XOR E(CD) XOR E(16C-FF)
= E(FB) XOR E(1C) XOR E(CD) XOR E(6D)
= B4 XOR 1E XOR CB XOR D6
= **B7**
Output(2) = (14 * 0D) XOR (D3*09) XOR (30*0E) XOR (60*0B)
= E(L(14)+L(0D)) XOR E(L(D3)+L(09) XOR E(L(30)+L(0E)) XOR E(L(60)+(0B))
= E(34+EE) XOR E(3C+C7) XOR E(65+DF) XOR E(7E+68)
= E(122-FF) XOR E(103-FF) XOR E(144-FF) XOR E(E6)
= E(23) XOR E(04) XOR E(45) XOR E(E6)
= E4 XOR 11 XOR 3B XOR 8D
= **43**
Output(3) = (14 * 0B) XOR (D3*0D) XOR (30*09) XOR (60*0E)
= E(L(14)+L(0B)) XOR E(L(D3)+L(0D)) XOR E(L(30)+L(09)) XOR E(L(60)+L(0E))
= E(34+68) XOR E(3C+EE) XOR E(65+C7) XOR E(7E+DF)
= E(9C) XOR E(12A-FF) XOR E(12C-FF) XOR E(15D-FF)
= E(9C) XOR E(2B) XOR E(2D) XOR E(5E)
= 9C XOR 70 XOR AB XOR 76
= 31

- **Inverse Shift rows**

All the shifts are done backwards as well. So, instead of shifting left, we shift right. The inverse rows shifted is row 0 will not be shifted, row 1 will be shifted 1 byte to the right, row 2 will be shifted 2 bytes to the right and so on.

0x52	0xa8	0x40	0x31
0xb7	0x50	0x31	0x4d
0x43	0x31	0x50	0x5b
0x31	0x50	0xa8	0xfd

- **Inverse Substitute bytes**

Lastly, the S-box is applied using the inverse S-box table. The inverse S-box table can easily be generated by taking the S-box value at some row and column index and assigning the row and column to the inverse S-box value at the inverse S-box index defined by the S-box value. For example, the S-box has a value of 0x00 at index 5,2. This translates to the inverse S-box having the value 0x52 at index 0,0. The inverse S-box table is shown in below Figure.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
00	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
10	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
20	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
30	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
40	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
50	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
60	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06

70	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
80	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
90	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a0	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b0	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c0	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d0	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e0	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f0	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Inverse S-box. From http://www.samiam.org/s-box.html

Same process we use in inverse S-Box to replace the byte of shifted byte to S-Box byte. After replacement we get the matrix below:

0x48	0x6F	0x72	0x2E
0x65	0x20	0x6C	0x2E
0x6C	0x57	0x64	0x2E
0x6C	0x6F	0x21	0x2E

- it would translate to the character value of hexadecimal value by using an ASCII lookup table.
  Hello World!.... –

H	o	r	.
e		l	.
l	W	d	.
l	o	!	.

6.2 THE Nth Prime RSA-CRT with LSB Stegnography

Key Generation:

1) Let us assume three prime numbers p, q, r to find public key and private key .
2) Compute n = p*q*r and z = (p-1)(q-1)(r-1).
3) Choose a number, e< n, such that gcd(e,z)=1 .

- Basic RSA
  a) $S = M^D \bmod N$
  D, N: 1024-bit integers

- 2-Prime RSA with CRT ($N = pq$)
  a) $S1 = M^{D1} \bmod p$
  b) $S2 = M^{D2} \bmod q$
  D1, D2, p, q: 512-bit integers
  c) S = fn (S1, S2)

- Multi-prime RSA with CRT ($N = pqr$)
  a) $S1 = M^{D1} \bmod p$
  b) $S2 = M^{D2} \bmod q$
  D1, D2, D3, p, q, r: 342-bit integers
  c) $S3 = M^{D3} \bmod r$
  d) S = fn (S1, S2, S3)
  e) D1=E mod(p-1) ; D2=E mod (q-1); D3=E mod(r-1).

- Let p ,q and r be very be two very large primes of nearly the same size.
- Compute N = p*q*r and Phi=(p-1)*(q-1)*(r-1)
- Choose Public key E <n and Choose Private key D such that DE-1 should divisible by Phi value.
- $S=M^D \bmod (N)$;
  S=(S1C1qr+ S2C2pr+ S3C3pq)
- $S1=M^{D1} \bmod p$;
  $S2=M^{D2} \bmod q$;
  $S3=M^{D3} \bmod r$.
- D1=E mod(p-1);

D2=E mod(q-1);
D3=E mod(r-1).

- $C1=(rq)^{-1}$ mod p;
  $C2=(pr)^{-1}$ mod q;
  $C3=(pq)^{-1}$ mod r .
- S = S1*C1*q*r + S2*C2*p*r + S3*C3*p*q.

Encryption:

1) S = f mod(S,n), S is cipher text.

Decryption:

1) M=f mod(S,d,n); i.e $M=S^D$ mod (N);

Apply LSB:

1) Reading the Input Image .
2) Reading the Message/RSA Encoded data to be encoded
3) Apply Mask Processing to the all Pixel Values in the image
4) Check the weighted position of the message bit

Ex: If m = 74 then its equivalent binary value was 01001010. Then checking this trough program by making and operation with 2^6 i.e. 64, and 32,16,8,4,2,1

1 Pixel:

01 00 10 10 **(8 bit pixels)**
white red green blue
Insert 0011
00 00 11 11
white red green blue

5) Embedded the each value of message bit in every step of above operation by or operation of pixel value with 1.
6) Repeat the above step for all bits in cipher text and we will get embedded or encoded image .

To retrieve image:

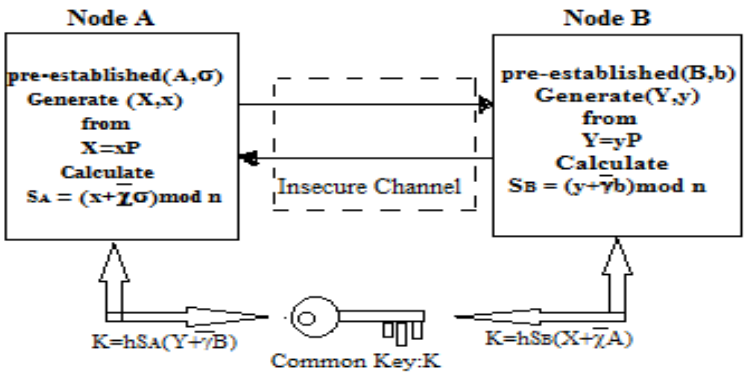
1) Observe the pixel values last bit by and OR operation with
2) Then update the message bit by or operation with its respective position i.e. 64,32,16,8,4,2,1
3) Then cipher text bit is retrieved at the receiver side

7 . Elliptic Curve Menezes-Qu-Vanstone key agreement protocol (ECMQV)

The ECMQV key agreement method is used to establish a shared secret between parties who already possess trusted copies of each other's public keys. Both parties still generate public and private keys and then exchange public keys. However, upon receipt of the other party's public key, each party calculates a quantity called an *implicit signature* using their own private key and the other party's public key. The shared secret is then generated from the implicit signature. The term *implicit signature* is used to indicate that the shared secrets will not agree if the other party's public key is not employed, thus giving implicit verification that the remote secret was generated by the remote party. An attempt at interception will fail as the shared secrets will not be the same shared secrets because the adversary's private key is not linked to either trusted public key.

Let's return to the example of Alice communicating with her Bank. If Alice has the Bank's public key and the Bank has Alice's public key, then the ECMQV key exchange may be used. Anyone intercepting the transmissions and substituting another remote key is unable to communicate because the resulting shared secrets differ. ECMQV protocol is shown in below Figure .

MQV Key Agreement



In this figure, the following nomenclature is used:
A, B- Long-term public keys
a, b- Long-term private (secret) keys

X, *Y*- Ephemeral public keys
x, *y* -Ephemeral private (secret) keys
h -The integer $h = \#E(F_q)/n$ where $\#E(F_q)$ is the order of the elliptic curve E and n is the order of the base point P.
$\overline{\chi}$, $\overline{\gamma}$- The integers derived from the *X*,*Y* public keys as follows:
i. Convert the x-coordinate of the public key to an integer (note this is not the private key, *x*);
ii. Calculate $x' \equiv x \bmod 2^{(\log_2 n)/2}$
iii. Calculate $\overline{\chi} = x' + 2^{(\log_2 n)/2}$
iv. Calculate $\overline{\gamma}$ similarly using the *Y* public key.
In the ECMQV scheme, Alice possesses a long-term key pair *(A, a)* with *A* being her public key and *a* being her private key. Similarly, the Bank possesses a long-term key pair *(B, b)* with *B* being his public key and *b* being his private key. Alice generates an ephemeral session key pair *(X, x)* by randomly generating *x* and calculating $X = xP$ where x is an integer and *P* is a point on an elliptic curve. The Bank generates its ephemeral key pair *(Y, y)* by randomly generating *y* and calculating $Y = yP$. *P* is a generating point on the elliptic curve.
Alice sends *X* to the Bank and the Bank sends *Y* to Alice. It is assumed that Alice already has the Bank's public key *B* and that the Bank already has Alice's public key *A*. The public keys have been received in some trusted manner.
ECMQV primarily owes its efficiency over the Station-to- Station (STS) protocol to the fact that it uses implicit signatures to ensure that the data contributed by the parties is authentic
and complete. STS must use a standard-explicit signature, such as ECDSA, which is computationally more expansive. In fact, the dominant calculations in ECMQV are only 1.5 scalar point multiples. As can be seen, the quantity SA acts as a signature on Alice's ephemeral public key, and only Alice can produce it. The implicit nature of the signature is because the Bank indirectly verifies it when deriving the shared secret since each party's shared secret will not agree if these signatures are invalid.

CONCLUSION

This paper presents an effective algorithm that combines techniques that can be used to successfully communicate secretively in a network. It uses RSA Algorithm, one of the most effective and commonly used cryptographic algorithms and adds LSB Stegnography to it to reduce attacks. The security RSA AES better than RSA-DES and our proposed algorithm is efficient than RSA AES during the application of data transmission. We observed that ECC having shorter key length than RSA for same level of security. The ECC is an emerging alternative for traditional Public-Key Cryptosystem like RSA, DSA and MQV. ECC concept using with MQV to solve the problem of key Exchange the algorithm is called ECMQV. ECC is also use for the digital signature is called as an ECDSA algorithm. This hybrid structure of enhanced AES and RSA provides more security. Better key exchange provided by ECMQV with authentication. Thus the proposed Hybrid Encryption Algorithm using Block cipher and symmetric key provides a more secure and convenient technique for secure data trans-mission for all kind application.

REFERENCES

[1] Jigar Chauhan , Neekhil Dedhia, Bhagyashri Kulkarni ," Enhancing Data Security by using Hybrid Cryptographic Algorithm." International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2, Issue 3, May 2013.
[2] Meenakshi Shankar and Akshaya.P, "Hybrid Cryptographic Technique Using RSA Algorithm and Scheduling Concepts." International Journal of Network Security & Its Applications (IJNSA) Vol.6, No.6, November 2014.
[3] S. Subasree and N. K. Sakthivel, "Design of A New Security Protocol Using Hybrid Cryptography Algorithms." IJRRAS 2 (2), February 2010.
[4] Teoh Suk Kuan Rosziati Ibrahim. "Steganography algorithm to hide secret message inside an image.Computer Technology and Application", 2:102-108, 2011.
[5] Wang Tianfu, K. Ramesh Babu, "Design of a Hybrid Cryptographic Algorithm.", International Journal of Computer Science & Communication Networks,Vol 2(2), 277-283.
[6] N. Koblitz., 1987 "Elliptic curve cryptosystems", *Mathematics of Computation"*, 48, pages 203–209.
[7] Fillatre. L, "Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption", IEEE Transactions on Signal Processing, Volume 60, Issue:2, pp. 556-569, Feb, 2012
[8] Masud K. S.M. Rahman, Hossain, M.L.," A new approach for LSB based image steganography using secret key", in Proceedings of 14th International Conference on Computer and Information Technology (ICCIT-2011), pp.-286-291, Dec. 2011.
[9] Hema Ajetrao, Dr. P.J.Kulkarni and Navanath Gaikwad, "A Novel Scheme of Data Hiding in Binary Images", in International Conference on Computational Intelligence and Multimedia Applications, Vol.4, pp. 70-77, Dec. 2007.
[10] Sachdeva S. and Kumar A, "Colour Image Steganography", Based on Modified Quantization Table, in Proceedings of Second International Conference on Advanced Computing & Communication Technologies (ACCT-2012), pp. 309-313, 2012.
[11] R. Machado, http://www.securityfocus.com/tools/586/scoreit, .EzStego., Nov. 1996. [last accessed on 16-04-2012]

[12] Y. C Tseng and H. K Pan," Data Hiding in 2-color Image in IEEE Transactions on computers", Vol. 51, No. 7, pp. 873-878, July 2002.
[13] E. Kawaguchi and R. O. "Eason, Principle and applications of BPCS-Steganography, in Proceedings of SPIE Int'l Symp. on Voice, Video, and Data Communications", pp. 464-473, 1998.
[14] Steganographic software, http://www.jjtc.com/Steganography/toolmatrix.html [last acessed on 16-04- 2012]
[15] B. Kaliski. "An unknown key-share attack on the MQV key agreement protocol.", Manuscript. Proceedings version appeared in RSA Conference 2000 Europe, Munich, April 2000. Work based on earlier communication to IEEE P1363a and ANSI X9F1 working groups, June 1998.
[16] Phillip Rogaway, Mihir Bellare, Dan Boneh, "Evaluation of Security Level of Cryptography: ECMQVS (from SEC 1)", Jan. 2001. Available at: http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1069_ks-ecmqv.pdf.
[17] "Another Look at HMQV Alfred Menezes", Nov. 2005.Available at: http://eprint. iacr.org/2005/205.pdf.
[18] X. Li, J. Chen, D. Qin, W. Wan**,** 2010**,** "Research and Realization based on hybrid encryption algorithm of improved AES and ECC", IEEE.
[19] J. J. Quisquater and C. Couvreur, "Fast decipherment algorithm for RSA public key cryptosystem," Electronic Letters, vol. 18, pp.905-907, 1982.
[20] X. Li, J. Chen, D. Qin, W. Wan**,** 2010**,** "Research and Realization based on hybrid encryption algorithm of improved AES and ECC", *IEEE*.
[21] Wikipedia, the free encyclopedia.
[22] Addam Schroll, ITNS and CERIAS CISSP Luncheon Series: Cryptography.
[23] http://science.opposingviews.com/advantages-disadvantages-symmetric-key-encryption-2609.html

