ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 5, Issue 1, January 2017

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Comparative Study and Analysis of Cryptographic Algorithms AES and RSA

Harshala B. Pethe[1]
Department of Electronics and Comp.Sc.
RTMNU, Nagpur – India

Dr. Subhash R. Pande[2]
SSESA's Science College,
Nagpur – India

Abstract: ***The use of internet is growing day-by-day. Therefore information security is the major issue today. To keep our data safe over the internet use of cryptography is important. Cryptography makes the data like text, image, audio and video unreadable during transmission and the main goal is to keep data secure from unauthorized access. Cryptography plays a vital role in the network security to achieve confidentiality, authentication, integrity and non-repudiation of information. This paper gives the comparative study and analysis of cryptographic algorithms Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) algorithm.***

Keywords: AES, RSA, Symmetric key cryptography, encryption, decryption, cryptographic algorithms.

I. INTRODUCTION

Cryptography is basically the process of hiding information. Computer passwords and transferring data from one place to another are done with cryptography. It is the science of using mathematics to encrypt and decrypt data. It enables to store sensitive information or transmit it across insecure networks so that it cannot be read by anyone except the intended recipient. It is the art and science of protecting information from unintended individuals by converting it into a non-recognizable form during transmission [1]. Data cryptography is the scrambling of the content of data, such as text, image, audio, video and so forth to make the data unreadable, invisible or unintelligible during transmission or storage called Encryption. The main goal of cryptography is keeping data secure form unauthorized attackers. The reverse of data encryption is data Decryption. Original data that to be transmitted or stored is called plaintext, this data can be readable and understandable either by a person or by a computer[2]. Whereas the data, which is unreadable, neither human nor machine is called cipher text. A system that provides encryption and decryption is called cryptosystem. Cryptosystem uses an encryption algorithms which determines how simple or complex the encryption process will be, the necessary software component, and the key which works with the algorithm to encrypt and decrypt the data. The security level of an encryption algorithm is measured by the size of its key space [3]. The range of possible values of keys is called key space. The larger size of the key space, more efforts will be required by attacker to break the key, and security level of such algorithm will be higher. In encryption, the key is piece of information which specifies the particular transformation of plaintext to cipher text, or vice versa during decryption. If the key space is larger, more possible keys can be constructed. For Example if the key sizes are of 128, 192 or 256 bit, then the key size of 256 would provide a 2^{256} key space. The strength of the encryption algorithm relies on the secrecy of the key, length of the key, the initialization vector, and how they all work together.

Cryptography algorithms are either symmetric algorithms, which use symmetric keys (also called secret keys), or asymmetric algorithms, which use asymmetric keys (also called public and private keys).

Symmetric Algorithms

Symmetric algorithms, sometimes called conventional algorithms, are algorithms where the encryption key can

be calculated from the decryption key and vice versa. These algorithms, also called secret-key algorithms, single key algorithms, or one-key algorithms, require that the sender and receiver agree on a key before they can communicate securely. The security of a symmetric algorithm rests in the key

Encryption and decryption with a symmetric algorithm are denoted by:

$E_K(M) = C$

$D_K(C) = M$

Symmetric algorithms can be divided into two categories. Some operate on the plaintext a single bit (or sometimes byte) at a time; these are called stream algorithms or stream ciphers. Others operate on the plaintext in groups of bits. The groups of bits are called blocks, and the algorithms are called block algorithms or block ciphers.

Public-Key Algorithms

Public-key algorithms are also called asymmetric algorithms and are designed so that the key used for encryption is different from the key used for decryption. Therefore the decryption key cannot be calculated from the encryption key. The algorithms are called "public-key" because the encryption key can be made public. The encryption key can be used by any person to encrypt a message, but only a specific person with the corresponding decryption key can decrypt the message. The encryption key is called public key, and the decryption key is often called the private key.

Encryption using public key K is denoted by:

$E_K(M) = C$

Even though the public key and private key are different, decryption with the corresponding private key is denoted by:

$D_K(C) = M$

In this paper we have done the comparative analysis of AES which is symmetric key algorithm and RSA, which is asymmetric key algorithm.

II. GOALS OF CRYPTOGRAPHY

A. Confidentiality

Confidentiality means protection against unauthorized disclosure of information. It may be applied to whole messages, parts of messages, and even existence of messages. Confidentiality provides the protection of transmitted data from passive attacks.

B. Authentication

The process of proving one's identity. This includes verifying the message's source. Authentication is of two types: (i) Peer entity authentication, and (ii) Data origin authentication.

C. Data integrity

The integrity is an assurance that the message has not been modified. This can be applied to a stream of messages, a single message, or selected fields within a message. It assures that messages are received as sent, with no duplication, insertion, modification, reordering, or replays.

D. Access control

It is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

E. Non repudiation

Sender or receiver cannot deny for a transmitted message. When a message is sent, the receiver can prove that the sender in fact sent the message [4][5].

III. BLOCK CIPHER MODES

For providing the flexibility, block cipher algorithms can operate in Electronic Code Book(ECB), Cipher Block Chaining(CBC), Cipher Feedback(CFB), Output Feedback(OFB) and Counter(CTR) modes.

A. Electronic Code Book (ECB)

This mode is used for processing a series of sequentially listed message blocks. Operation of this mode is as follows:

The user takes the first block of plaintext and encrypts it with the key to produce the first block of cipher text. Then the second block of plaintext is taken and follows the same process with same key and so on so forth. The ECB mode is deterministic, that is, if plaintext block P1, P2,…, PN are encrypted twice under the same key, the output cipher text blocks will be the same. In fact, for a given key technically we can create a codebook of cipher texts for all possible plaintext blocks. Encryption would then entail only looking up for required plaintext and select the corresponding cipher text.

Thus, the operation is analogous to the assignment of code words in a codebook, and hence gets an official name − Electronic Codebook mode of operation (ECB).

Analysis of ECB Mode

In reality, any application data usually have partial information which can be guessed. For example, the range of salary can be guessed. A cipher text from ECB can allow an attacker to guess the plaintext by trial-and-error if the plaintext message is within predictable. For example, if a cipher text from the ECB mode is known to encrypt a salary figure, then a small number of trials will allow an attacker to recover the figure. In general, we do not wish to use a deterministic cipher, and hence the ECB mode should not be used in most applications.

B. Cipher Block Chaining (CBC)

CBC mode of operation provides message dependence for generating cipher text and makes the system Non-deterministic.

Operation of CBC mode is as follows:

The operation of CBC mode is depicted in the following illustration. The steps are as follows:

- Load the n-bit Initialization Vector (IV) in the top register.
- XOR the n-bit plaintext block with data value in top register.
- Encrypt the result of XOR operation with underlying block cipher with key K.
- Feed cipher text block into top register and continue the operation till all plaintext blocks are processed.
- For decryption, IV data is XORed with first cipher text block decrypted. The first cipher text block is also fed into to register replacing IV for decrypting next cipher text block.

Analysis of CBC Mode

In CBC mode, the current plaintext block is added to the previous cipher text block, and then the result is encrypted with the key. Decryption is thus the reverse process, which involves decrypting the current cipher text and then adding the previous cipher text block to the result. Advantage of CBC over ECB is that changing IV results in different cipher text for identical

message. On the drawback side, the error in transmission gets propagated to few further block during decryption due to chaining effect.

It is worth mentioning that CBC mode forms the basis for a well-known data origin authentication mechanism.

C. Cipher Feedback (CFB)

In this mode, each cipher text block gets 'fed back' into the encryption process in order to encrypt the next plaintext block. Operation of CFB mode is as follows:

For example, in the present system, a message block has a size 's' bits where $1 < s < n$. The CFB mode requires an initialization vector (IV) as the initial random n-bit input block. The IV need not be secret. Steps of operation are:

- Load the IV in the top register.
- Encrypt the data value in top register with underlying block cipher with key K.
- Take only's' number of most significant bits (left bits) of output of encryption process and XOR them with 's' bit plaintext message block to generate cipher text block.
- Feed cipher text block into top register by shifting already present data to the left and continue the operation till all plaintext blocks are processed.
- Essentially, the previous ciphertext block is encrypted with the key, and then the result is XORed to the current plaintext block.
- Similar steps are followed for decryption. Pre-decided IV is initially loaded at the start of decryption.

Analysis of CFB Mode

CFB mode differs significantly from ECB mode, the ciphertext corresponding to a given plaintext block depends not just on that plaintext block and the key, but also on the previous ciphertext block. In other words, the ciphertext block is dependent of message. CFB has a very strange feature. In this mode, user decrypts the ciphertext using only the encryption process of the block cipher. The decryption algorithm of the underlying block cipher is never used. Apparently, CFB mode is converting a block cipher into a type of stream cipher.

The encryption algorithm is used as a key-stream generator to produce key-stream that is placed in the bottom register. This key stream is then XORed with the plaintext as in case of stream cipher.

By converting a block cipher into a stream cipher, CFB mode provides some of the advantageous properties of a stream cipher while retaining the advantageous properties of a block cipher.

D. Output Feedback (OFB)

It involves feeding the successive output blocks from the underlying block cipher back to it. These feedback blocks provide string of bits to feed the encryption algorithm which act as the key-stream generator as in case of CFB mode. The key stream generated is XOR-ed with the plaintext blocks. The OFB mode requires an IV as the initial random n-bit input block. The IV need not be secret[4].

Counter (CTR)

It can be considered as a counter-based version of CFB mode without the feedback. In this mode, both the sender and receiver need to access to a reliable counter, which computes a new shared value each time a cipher text block is exchanged. This shared counter is not necessarily a secret value, but challenge is that both sides must keep the counter synchronized. Both encryption and decryption in CTR mode are depicted in the following illustration. Steps in operation are −

- Load the initial counter value in the top register is the same for both the sender and the receiver. It plays the same role as the IV in CFB (and CBC) mode.
- Encrypt the contents of the counter with the key and place the result in the bottom register.
- Take the first plaintext block P1 and XOR this to the contents of the bottom register. The result of this is C1.

Send C1 to the receiver and update the counter. The counter update replaces the cipher text feedback in CFB mode.

- Continue in this manner until the last plaintext block has been encrypted.
- The decryption is the reverse process. The ciphertext block is XORed with the output of encrypted contents of counter value. After decryption of each ciphertext block counter is updated as in case of encryption.

Analysis of Counter Mode

It does not have message dependency and hence a ciphertext block does not depend on the previous plaintext blocks. Like CFB mode, CTR mode does not involve the decryption process of the block cipher. This is because the CTR mode is really using the block cipher to generate a key-stream, which is encrypted using the XOR function. In other words, CTR mode also converts a block cipher to a stream cipher. The serious disadvantage of CTR mode is that it requires a synchronous counter at sender and receiver. Loss of synchronization leads to incorrect recovery of plaintext. However, CTR mode has almost all advantages of CFB mode. In addition, it does not propagate error of transmission at all.

IV. Overview of AES and RSA

A. Advanced Encryption Standard

AES is a block oriented symmetric key encryption algorithm. It is developed in 2000 and considered to be more secure than Data Encryption Standard (DES) algorithm. AES is based on the design principle known as substitution permutation network.

It operates on a 128 bit data block at a time and uses 128, 192 or 256 bits key length and uses 10, 12 0r 14 rounds. A data block is partitioned into an array of bytes. Such bytes are interpreted as a finite field elements using polynomial representation. The input is divided into 16 bytes and then arranged into a 4x4 matrix column wise [6]. This matrix is known as the state matrix. The original 128-bit key is also divided in to 16 bytes as like 128 bit data and arranged in the form of 4x4 matrix. This matrix is called keyMatrix.

Both these matrices form the necessary inputs to the algorithm.

AES encryption includes,

1) An initial round(0)
2) Nine general rounds (1 to 9) and
3) A final round (10)

In round (0) the two matrices are simply XORed under AddRoundKey transformation. The output of Round0 is given as the input to Round 1. Each round composed of four distinct, uniform and invertible transformations: Subbytes, ShiftRows, MixColumn and AddRoundKey [7][8] as shown in the following figure.

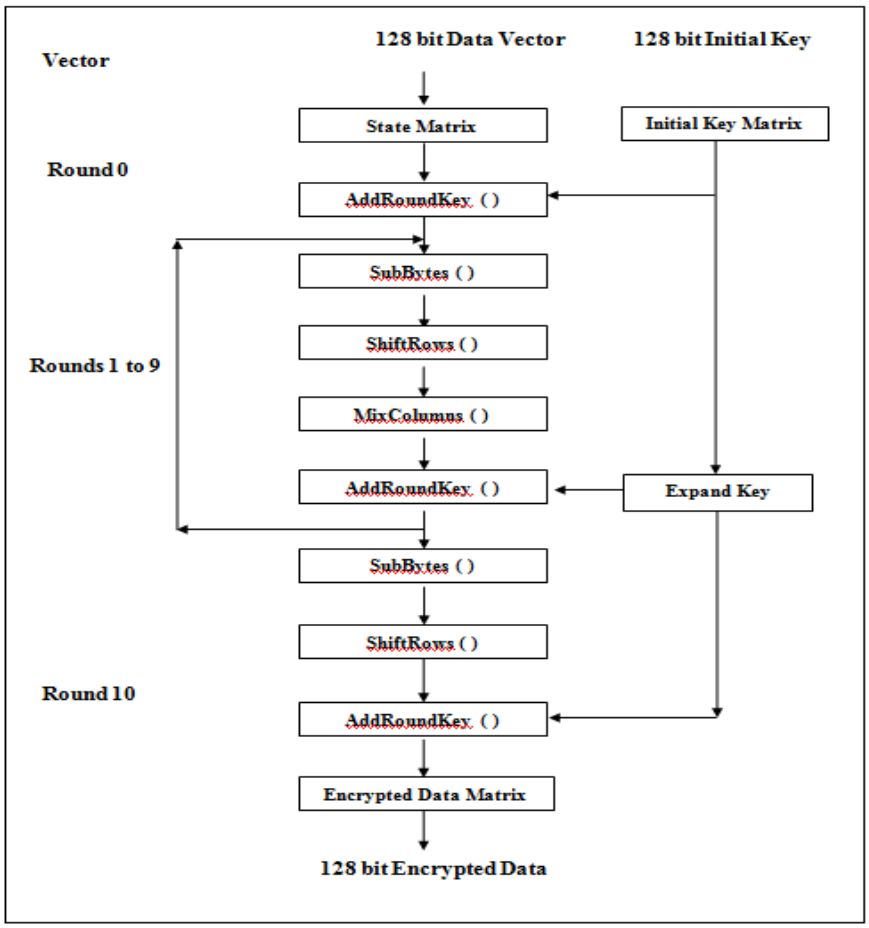

Figure 1 : Structure of AES encryption

A. RSA Algorithm (Rivest-Shamir-Adleman)

RSA is widely used in encrypted connection, digital certificates core algorithms. Public key algorithm invented in 1977 by Ron Rivest,Adi Shamir and Leonard Adelman(RSA).It is the main operation of RSA to compute modular exponentiation. Since RSA is based on arithmetic modulo large numbers, it can be slow in constraining environments [9].Especially, when RSA decrypts the cipher text and generates the signatures, more computation capacity and time will be required. Reducing modules in modular exponentiation is a technique to speed up the RSA decryption. The security of RSA comes from integer to find. Generation of random prime numbers gives the algorithm extra strength and efficiency.

Following steps are followed in RSA to generate the public and private keys [10]:

Step 1: Choose large prime numbers p and q such that p not equal to q.

Step 2: Compute n=p*q

Step 3: Compute φ (pq) = (p-1)*(q-1)

Step 4: Choose the public key e such that gcd (φ (n), e) =1; 1<e< φ (n)

Step 5: Select the private key d such that d*e mod φ (n) =1

In RSA algorithm encryption and decryption are performed as follows:

Encryption:

Calculate cipher text C from plaintext message M such that

C=M ^e mod n

Decryption:

M=C^d mod n=M^ed mod n

Parameters considered	AES	RSA
Key Length	128,192 or 256 bits	1024 bits
Block Size	128,192 or 256 bits	128 bits
Cipher Text	Symmetric block cipher	Asymmetric block cipher
Developed	2000	1978
Better	In terms of cost and security	In terms of speed and security
Possible Keys	2^{128}, 2^{192} and 2^{256}	2^{128}

Table1: Theoretical comparison of AES

The same key is used for both encryption and decryption in AES algorithm[11].

In RSA algorithm Public key is used for encryption and private key is used for decryption. The values of chosen prime numbers p and q controls time in key generation and makes it more secured.[12]

V. EXPERIMENTAL RESULTS

The AES algorithm is implemented using MATLAB2013a. The time required for encryption and decryption is shown in the following table with key "abcdefgh." The image files are taken from NEOCR dataset.

Image File	Encryption Time	Decryption Time
img_1	13.6437	15.3152
img_2	15.0955	17.1888
img_3	15.9024	17.3949
img_4	14.5828	19.9549
img_5	16.2648	16.5337
img_6	13.6232	16.6028
img_7	14.4369	18.7118
img_8	15.0703	17.333
img_9	14.6732	18.7691
img_10	15.7618	16.7662
img_11	13.2121	15.8784
img_12	16.147	16.4052
img_13	14.4895	16.3243
img_14	17.5585	19.4406
img_15	16.9337	19.4408

Table2: Encryption and decryption time using AES algorithm

RSA algorithm is implemented using MATLAB2013a. The time required for encryption and decryption is shown in the following table with p=17 and q=19. The image files are taken from NEOCR dataset.

Image file	Encryption Time	Decryption Time
img_1	5.49546	45.8044
img_2	5.51208	46.326
img_3	5.05037	46.5991
img_4	4.32142	55.5687
img_5	5.40518	59.1869

img_6	5.97177	68.4956
img_7	6.82683	71.3607
img_8	9.93299	111.206
img_9	8.10602	85.3519
img_10	9.15301	82.4232
img_11	12.2978	150.649
img_12	13.5354	151.906
img_13	9.67519	117.821
img_14	17.1598	142.407
img_15	13.5254	142.674

Table 3: Encryption and decryption time using RSA algorithm

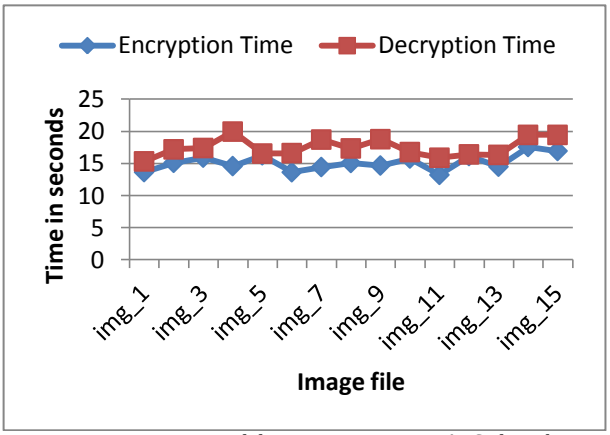

Figure 2: Encryption and decryption time using AES algorithm

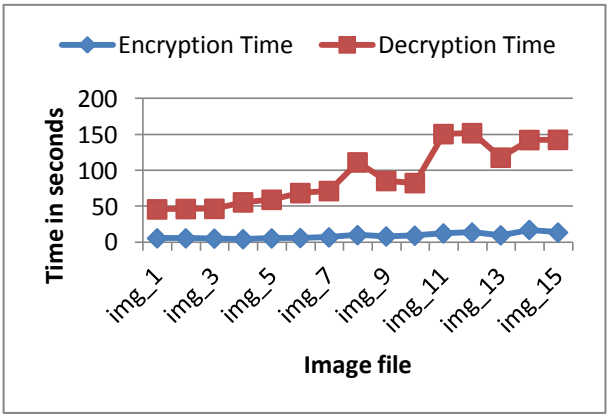

Figure 3: Encryption and decryption time using RSA algorithm

VI. Conclusion

In this paper we have implemented the Advanced Encryption Standard (AES) and RSA algorithms using MATLAB R2013a for different image files of increasing sizes, keeping key constant and it is observed that the time required for encryption and decryption increases as the file size increases. Also the time required for encryption is less than the time required for decryption in both algorithms but the time required for encryption in RSA algorithm is less as compared to the time required for encryption in AES algorithm. AES is found to be better in terms of cost, security and implementation.RSA algorithm is better in terms of speed and security.

References

1. DiaasalamaAbdElminaam, HatemMohamadAbdual Kader, Mohly Mohamed Hadhoud, ―Evalution the Performance of Symmetric Encryption Algorithms‖, international journal of network security vol.10,No.3,pp,216-222,May 2010.
2. Shraddha Soni, Himani Agrawal, Dr. (Mrs.) Monisha Sharma Analysis and Comparison between AES and DES cryptographic algorithm International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 6, December 2012

3. Schneier, B. Applied Cryptography
4. Ritu Tripathi, Sanjay Agrawal Comparative Study of Symmetric and Asymmetric Cryptography Techniques International Journal of Advance Foundation and Research in Computer (IJAFRC) Volume 1, Issue 6, June 2014. ISSN 2348 – 4853.
5. Vikrant M. Adki, Prof. Shubhanand S. Hatkar A Survey on Cryptography Techniques International Journal of Advanced Research in Computer Science and Software Engineering Volume 6, Issue 6, June 2016 ISSN: 2277 128X.
6. T. Saravanan, V. Srinivasan, R, Udayakumar "MATLAB-Simulink Implementation of AES Algorithm for Image Transfer" Middle-East Journal of Scientific Research 18(12) 1709-1712, 2013.
7. Bawna Bhat, Abdul Wahid Ali, Apurva Gupta DES and AES Performance Evaluation International Conference on Computing, Communication and Automation (ICCCA2015).
8. T. Saravanan, 1 1V. Srinivasan and 2R.Udayakumar MATLAB-Simulink Implementation of AES Algorithm for Image Transfer Middle-East Journal of Scientific Research 18 (12): 1709-1712, 2013 ISSN 1990-9233
9. AnnapoornaShetty, Shravya Shetty K, Krithika K A Review on Asymmetric Cryptography – RSA and ElGamal Algorithm International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 5, October 2014ISSN(Online): 2320-9801 ISSN (Print): 2320-9798
10. Aman Kumar, Dr. Sudesh Jakhar, Mr. Sunil Makkar Comparative Analysis between DES and RSA Algorithm's International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 7, July 2012 ISSN: 2277 128X.
11. Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani New Comparative Study Between DES, 3DES and AES within Nine Factors JOURNAL OF COMPUTING, VOLUME 2, ISSUE 3, MARCH 2010, ISSN 2151-9617
12. Ali E. Taki El_Deen, El-Sayed A. El-Badawy, Sameh N. Gobran Digital Image Encryption Based on RSA Algorithm IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 9, Issue 1, Ver. IV (Jan. 2014), PP 69-73.

