International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

Encrypted Email System

Ankit Shukla, Aniket Mohite, Ashish Singh Rawat
Computer Department, Vishwatmak Om Gurudev College of Engineering, Maharashtra, India

ABSTRACT

Encryption is of prime importance when confidential data is transmitted over the network. Varied encryption algorithms like AES, DES, RC4 and others are available for the same. The most widely accepted algorithm is AES algorithm. We have developed an application which allows the user to encrypt the messages before it is transmitted over the network. We have used the Advanced Encryption Standards algorithm(512 bits) for encryption and decryption of the data. This application provides a secure, fast, and strong encryption of the data. There is a huge amount of confusion and diffusion of the data during encryption which makes it very difficult for an attacker to interpret the encryption pattern and the plain text form of the encrypted data.The main objective of this paper is to provide stronger security for communication network over the Internet by enhancing the overall strength of the AES algorithm. Rijndael's algorithm was been selected as the Advanced Encryption Standard. The AES algorithm provides much more security without any limitations. The various uses of this application in real life and its functionality are explained in this paper.

Keywords: Web page application, AES algorithm, Cryptography, Decryption, Encryption.

I. INTRODUCTION

Network security is becoming much more important as people spend much more time connected in a network. To protect the value and ongoing usability of assets, the integrity and continuity of operations it involves all activities that institutions, enterprises and organizations undertake. Security attacks include modification of messages or files, denial of service, traffic analysis and unauthorized reading of a message of file.The application developed for end to end secure transmissionof the EMAIL. The algorithm used is Advanced Encryption Standards algorithm. The later part of the paper explains the working of EMAIL, the AES algorithm and the working of our developed application.

1.1 Literature Survey

Electronic mail (email) is a method of exchanging messages between people using electronics. Email first entered substantial use in the 1960s and by the mid-1970s had taken the form now recognized as email. Email operates across computer networks, which today is primarily the Internet. Some early email systems required the author and the recipient to both be online at the same time, in common with instant messaging. Today's email systems are based on a store-and-forward model. Email servers accept, forward, deliver, and store messages. Neither the users nor their computers are required to be online simultaneously they need to connect only briefly, typically to a mail server or a webmail interface, for as long as it takes to send or receive messages.

Encryption has long been used by militaries and governments to facilitate secret communication. Encryption is now commonly used in protecting information within many kinds of civilian systems. For example, the Computer Security Institute reported that in 2007, 71% of companies surveyed utilized encryption for some of their data in transit, and 53% utilized encryption for some of their data in

storage [3] Encryption can be used to protect data "at rest", such as files on computers and storage devices (e.g. USB flash drives). In recent years there have been numerous reports of confidential data such as customers' personal records being exposed through loss or theft of laptops or backup drives. Encrypting such files at rest helps protect them should physical security measures fail. [2] Digital rights management systems which prevent unauthorized use or reproduction of copyrighted material and protect software against reverse engineering (see also copy protection) are another somewhat different example of using encryption on data at rest.

Email encryption can rely on public-key cryptography, in which users can each publish a public key that others can use to encrypt messages to them, while keeping secret a private key they can use to decrypt such messages or to digitally encrypt and sign messages they send.

Various algorithms for encryption and decryption are in place. Out of the entire group of algorithm AES is the most preferred one.

AES require very low RAM space and its very fast. On Pentium Pro processors AES encryption requires only 18 clock cycles/byte equivalent to throughput of about 11Mib/s for 200MHz processor. This was the main reason why we decided to use AES algorithm for encryption and decryption. There are few EMAIL applications on internet which encrypts the MAIL using AES algorithm. We have programmed our application meticulously considering various factors which might benefit the user. It provides functionality like conversation view, Inbox, Draft ,compose all the functionality which a standard EMAIL application should provide. The main advantage is that it is very simple app, easy to understand and very easy to operator.

II. ELECTRONIC MAIL (EMAIL)

The email system is the network of computers handling electronic mail (email) on the Internet. This system includes user machines running programs that compose, send, retrieve, and view messages, and agent machines that are part of the mail handling system. Like other complex systems, the email system is best explained by looking separately at different perspectives, applying the principle of separation of concerns. There are two coequal ways of looking at email systems - the administrative perspective (who does what), and the process perspective (how it flows).

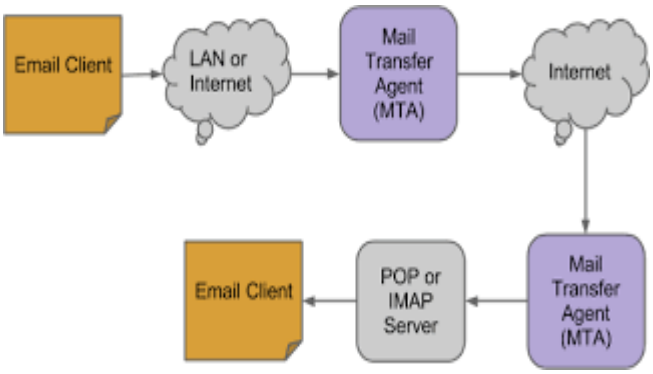

Figure 1. Transmission of SMS

2.1. EMAILPROTOCOL

2.1.1. SMTP

SMTP or Simple Mail Transfer Protocol is a protocol for sending email messages between servers. It is the most common protocol for sending email between two servers of the Internet. These emails can then be retrieved from servers either (Post Office Protocol) POP or (Internet Message Access Protocol) IMAP. SMTP servers are also sometimes referred as outgoing mail servers.

2.1.2. POP

POP or Post Office Protocol is a protocol used to retrieve email from a mail server. Almost all email applications use the POP protocol. There are two versions of POP, namely, POP2 (requires SMTP to send messages) and POP3 (can be used with or without SMTP).

2.1.3. IMAP

IMAP or Internet Message Access Protocol is a protocol for retrieving email messages from the mail server. The latest version, IMAP4, is similar to POP3 but supports some additional features. For example, with IMAP4, one can search through email messages for keywords while the messages are still on mail server. User can then download chosen messages to the machine.

2.1. 4. MIME

MIME or Multipurpose Internet Mail Extensions is a protocol used for formatting non-ASCII messages to be sent the Internet. Many of the current email services support MIME, enabling them to send and receive graphics, audio, and video files through the email system. In addition, MIME supports messages in character sets other than ASCII.

III. 3. ADVANCE ENCRYPTION STANDARDS

Algoritham

The Advanced Encryption Standard comprises three block ciphers, AES-128, AES-192, AES-256 AND AES-512. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits. The block-size has a maximum of 512 bits, but the key-size has no theoretical maximum. The cipher uses number of encryption rounds which converts plain text to cipher text. The output of each round is the input to the next round. The output of the final round is the encrypted plain text known as cipher text. The input given by the user is entered in a matrix known as State Matrix.

3.1. Aes-512 Architecture

The top level architecture of the AES-512 bits is shown in Figure 1. The plaintext and the key size are 512-bits each (organized in bytes). The AES-512 algorithm processes the data in 10 rounds. The key and the input data are loaded when the Loadkey control signal is one and zero, respectively. The Encrypt signal starts the encryption process, while reset resets everything to zero. The resulting cipher text is also 512-bits. More details about each of the transformations used in the AES-512 are described in the coming subsections. Where the key expansion procedure is explained later since each round needs its own key generated according to this procedure.

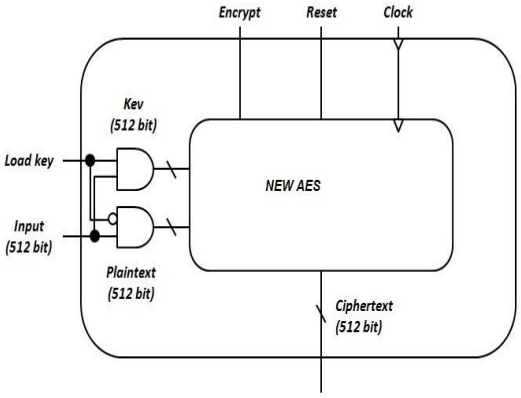

Figure 1. Top level of architecture AES-512

3.2. Implementation of Encryption

AES operates on a 16×32 array of bytes, termed the state. The input key for encryption is 512 bits. To represent the 512 values 9 bits are required. So each entry in S-box of AES 512 is 9 bits long. The cipher is specified in terms of repetitions of processing steps that are applied to make up rounds of keyed transformations between the input plain- text and the final output of cipher-text. The encryption procedure of AES 512 has been illustrated in figure 2.Each round in AES 512 encryption includes four different round transformations namely Substitute Bytes, Shift Rows, Mix Columns and Add Round Key. The last round of AES 512 encryption alone does not include the Mix Columns transformation.

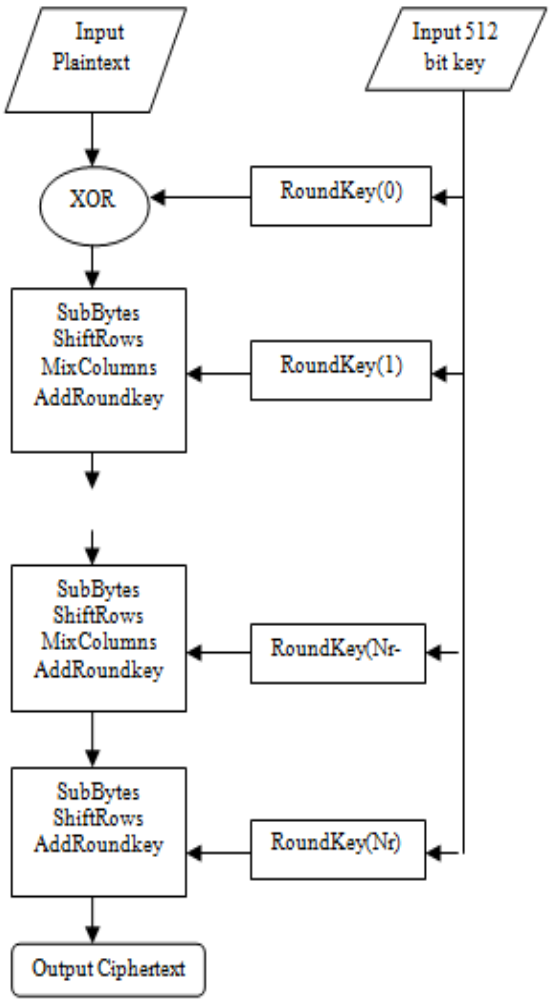

Figure 2. Encryption procedure of AES 512

3.3. Implementation of Decryption

A set of reverse rounds are applied to transform cipher- text back into the original plain-text using the same encryption key. The four reverse transformations used are Add Round Key, Inverse Mix Columns, Inverse Shift Rows and Inverse Substitute Bytes. The inverse S-box contains 512 values in its 16x32 array of bytes. Each round in decryption of AES 512 includes all the four reverse transformations except in the first round. The Inverse Mix Column transformation is violated in the first round of decryption since it does not occur in the last round of encryption. The decryption procedure of AES 512 is illustrated in figure 3.

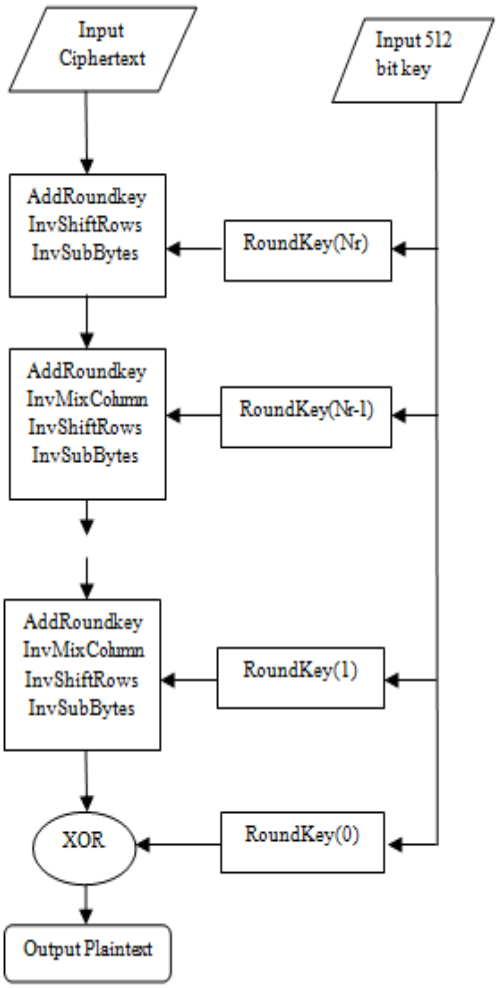

Figure 3. Decryption procedure of AES 512

3.4. Stages in Encryption and Decryption

3.4.1 Byte Substitution

The 512-bits input plaintexts are organized in array of 64- bytes and are substituted by values obtained from Substitution boxes. This is done (as in the original AES) to achieve more security according to diffusion-confusion Shannon's principles for cryptographic algorithms design. To overcome the overhead of the huge data size used (512-bits), the Substitution boxes are implemented as lookup tables, and accessed in parallel as shown in Figure 4.

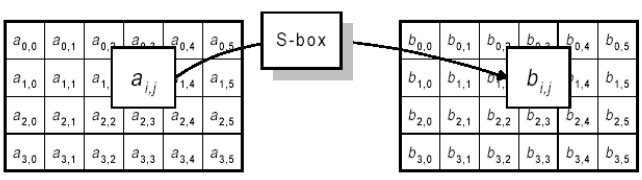

Figure 4. Byte Substitution

3.4.2 Shift Row

After the original 512-bit data is substituted with values from the S-boxes, the rows of the resulting matrix are shifted in a process called Shift Row transformation. What happened in this part is that the bytes in each row in the input data matrix will be rotated left. The number of left rotations is not the same in each row, and it can be determined by the row number. For example, row number zero is not shifted; the first row is shifted by one byte, and so on.

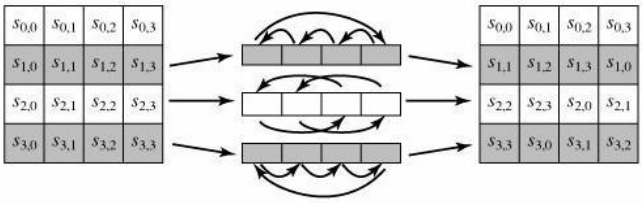

Figure 5. Shift Row

3.4.3 Mix Column

Now, and after the rows of the input data are rotated left by different offsets, an operation must be applied to the columns of the data matrix. The Mix Column transformation multiplies the columns of the data matrix by a pre-defined matrix. The AES-512 and original AES process the data in bytes basis. Each byte is considered as polynomials over GF (2^8) with 8 terms. To explain how the Mix Column works, we have to explain the concept of polynomials over GF (2^n) in general and for GF (2^8) as example when n=8.

The conversion might be dictated by the accompanying grid increase on state demonstrated in figure 7.3.3. Every component of the item framework is the entirety of results of components of one line and one segment. For this situation the unique augmentations & multiplication are achieved in GF (2^8).

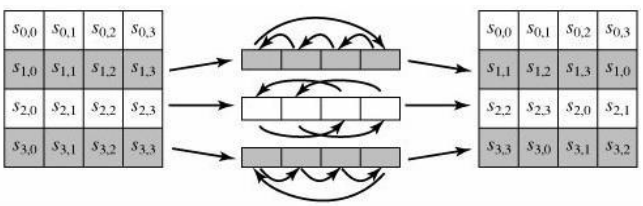

Figure 6. Mix Column

3.4.4 Add Round Key

In this process, the 128 bits of state are bitwise XORed through the 128 bits of the round key. The procedure is seen as a column wise process between the word of a state column and one WORD of the round key. This conversion is as basic as would be prudent which benefits in effectiveness yet it additionally influences all of state. To make the relationship between the key and the cipher text more complicated and to satisfy the confusion principle, the Add Round Key operation is performed. This addition step takes the resulting data matrix from the previous step and performs on it a bitwise XOR operation with the sub key of that specific round (addition operation in GF (2n)). We must mention that the round key is 512 bits that is arranged in a square matrix of eight columns where each column has 8 bytes.

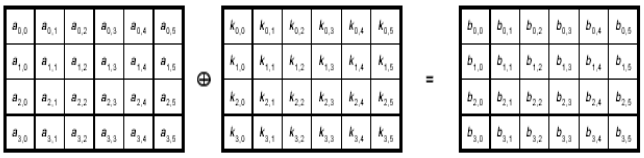

Figure 7. Add Round Key

3.4.5 Key Expansion and Rounds

The 512-bit input key of the new AES-512 algorithm is used to generate ten sub-keys for each of the ten AES rounds. The round ±keys expansion process involves arranging the original 512-bits input key into eight words of eight bytes each. After that, the round keys expansion is performed according to the following equations:

W (I) =W (i-8) XORW (I-1) I is not a multiple of 8

W (I) =W (i-8) XORT (W (I-1)) I is a multiple of 8

Where the T (I) transformation is defined as:

T (I) =Byte Sub (ShiftLeft(W(I)))XORRoundConst

The round constant is defined by the following equation: RoundConst = 00000010(i-8)/8 I is the round number.

The round structure of the AES-512 algorithm

(shown in Figure 8) uses the transformation defined in the previous section. First, byte substitution is performed on 512 bits data, followed by row rotation according to the row number, where left rotations are performed in this step. Then, the columns are multiplied by the new defined matrix column by column in the Mix Column transformation (except in the final round). The last operation will be the bitwise XORing with the round key expanded using the key expansion process. The output at of the final round will be the 512-bit encrypted message.

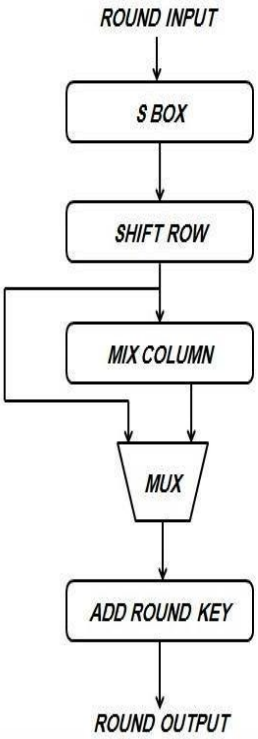

Figure 8. Single Round of AES-512 Algorithm.

Table 1. Number of Rounds

VERSIO	NUMBER OF
AES-192	
AES-256	
AES-512	

IV. GOALS OF THIS APPLICATION

The main goals of our application are:

1. Developing a secure Email application.
2. Maintaining encrypted information of message recipients.
3. Decrypting of message as per users requirement.
4. Protection against misuse of message information.
5. High confidentiality and improved security.

V. CONCLUSION

As a conclusion requirement for security and communication were we met. By encrypted email system , we can easily encrypt our sending mail and receiver easily receive the mail without disturbing any third party. We can easily know about mail corrupted or not during the transmission. Most importantly, the messages containing delicate information are stored securely and remain undisclosed even when the device is accessed by an adversary. This application guarantees end-to-end transmission of the data.

VI. REFERENCES

[1]. U.S. Department of Commerce/NIST, -Data Encryption Standard,FIPS PUB 46-3, pp. 1-26, October 1999.

[2]. NIST, Advanced Encryption Standard, FIPS PUB 197, pp. 1-51, November 2001.

[3]. J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Springer-Verlag, Berlin Heidelberg, 2002.

[4]. H. Gilbert and M. Minier, A collision attack on seven rounds of Rijndael, Proceedings of the 3rd AES Candidate Conference, pp.230-241, April 2000.

[5]. N. Ferguson, J. Kelsey, S. Lucks, et al. - Improved cryptanalysis of Rijndael, Lecture Notes in Computer 1-4244-1035-5/07/.2007 IEEE. 221 Science,vol. 1978, pp.213-230, Berlin: Springer-Verlag, 2001.

[6]. S. Lucks, -Attacking seven rounds of Rijndael under 192-bit and 256-bit keys, Proceedings of

the 3rd AES Candidate Conference, pp. 215-229, April 2000.

[7]. J. Daemen and V. Rijmen, -The Block Cipher Rijndael, Lecture Notes in Computer Science, vol.1820, pp.277- 284, Berlin: Springer-Verlag, 2000.

[8]. J. Daemen, and V. Rijmen, -The Wide Trail Design Strategy, Lecture Notes in Computer Science, vol. 2260, pp.222 - 238, Berlin: Springer-Verlag, 2001.