IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 87-91
www.iosrjournals.org

Dynamic S-Box In Aes Algorithm in Cloud Environment

Mr.Yash Shah[a],Mr.Anand Khandare[b]
[a]*PG Student, Thakur College of Engineering and Technology,Mumbai-400 101,India*
[c]*Assistant Professor,, Thakur College of Engineering and Technology,Mumbai-400 101,India*

Abstract: *Cloud computing today is the one of the most booming area of research and development in present scenario. The proliferation of computer usage and their interconnections through network have increased the need of protection of information stored against viruses, hackers, eavesdroppers, and electronic data deception. Although these threats require a variety of countermeasure, encryption process is a primary method of protecting valuable electronic information. The encryption process also needs to be dynamic in order to face new technique and more advance methods used by cryptanalysis. Substitution box (S-box) is keystone of modern symmetric cryptosystem. They bring nonlinearity to cryptosystem and strengthen their cryptographic security.*
In this project RC4 algorithm which is well known stream cipher is used to generate S-box for advance encryption standard (AES). The generated S-boxes are more dynamic and key dependant which will increase the complexity and also make the differential and linear cryptanalysis (DC&LC) more difficult. Various randomness tests would be applied to the customized AES (AES-RC4) algorithm and the results will be shown that the new design pass all tests which proven its security
Keywords: *Type your keywords here, separated by semicolons;*

I. Introduction

As the Internet and other forms of electronic communication become more prevalent, electronic security is becoming increasingly important. The art or science, encompassing the principles and methods of transforming an intelligible message into unintelligible and then retransforming that message back to its original form, to keep messages secure is Cryptography. Cryptography is used to protect e-mail, messages, credit card information, and corporate data. Within the context of any application-to-application communication, there are some specific security requirements, including:

- **Authentication:** The process of proving one's identity. Authentication is a process in which the credentials provided are compared to those on file in a database of authorized users' information on a local operating system or within an authentication server.
- **Privacy/confidentiality:** Ensuring that no one can read the message except the intended receiver.
- **Integrity:** Assuring the receiver that the received message has not been altered in any way from the original.
- **Non-repudiation:** A mechanism to prove that the sender really sent this message.

Cryptography is science and art of creation secrete code and cryptanalysis is art of breaking those codes. AES is the best and strongest cryptography algorithm, because of three areas: Security, Cost, and Implementation [1].

Two types of cryptographic systems have been developed for these purposes that are listed below.

- **Symmetric cryptography:** It uses the same cryptographic keys for both encryption of plaintext and decryption of cipher text. Figure 3. If the same keys are used for encryption or decryption, we call it symmetric cipher, i.e.,
  - Ek (M) = C
  - Dk (C) = M
  - Those functions have the property that, Dk(Ek(M))=M

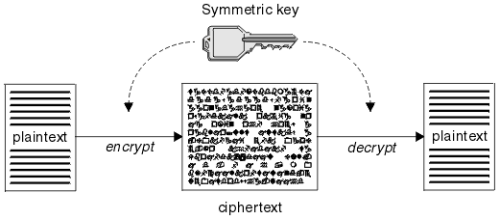

Figure 1. Symmetric Key Cryptography

- **Asymmetric cryptography:** Asymmetric cryptography or public-key cryptography is cryptography in which a pair of keys (k1, k2) is used to encrypt and decrypt a message so that it arrives securely. In case of asymmetric cipher we have a key pair (k1 ,k2), k1 being public & k2 private, then
  - Ek1 (M) = C
  - Dk2 (C) = M
  - Those functions have the property that,
  - Dk2 (Ek1(M))=M

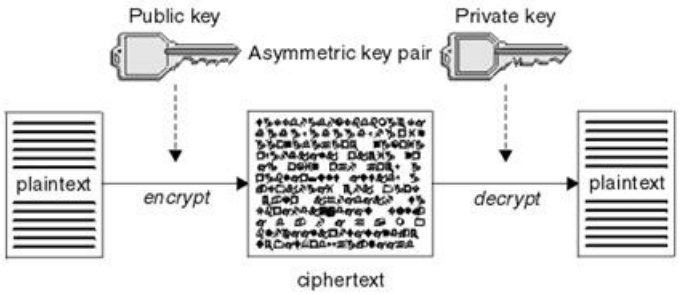

Figure 2. Asymmetric Key Cryptography

II. Motivation

Cloud computing today is the one of the most booming area of research and development in present scenario. Security is one of the prime considerations in cloud computing.
There are mainly 3 bodies involved in cloud architecture [2]:

- **End user**

End user is responsible for uploading (for future access) as well as downloading (whenever needed) the data in whatsoever format he/she desires.

- **Service provider**

A variety of services are needed by the end user such as user interface, download and upload section for files (data),modification privileges ,access control, security policies etc. All these services are provided by the service provider. The service provider mainly deals with the software aspect of the cloud.

- **Infrastructure provider**

Infrastructure provider is the body in the architecture where the files uploaded by end user are actually stored. IP provides with the actual hardware needed to store and retrieve data.

This project is focussing on the security aspect that is provided by the Service provider. It is the duty of the service provider to provide security and authentication policies for the end user. SP has to provide security for the data or the files that are uploaded by the end users so that there is no misuse of propriety data.
In order to achieve this security as an additional layer we propose to implement AES algorithm.
AES algorithm is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits.

III. Traditional Algorithm

The Advanced Encryption Standard (AES), also known as Rijndael (its original name), is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.AES is a subset of the Rijndael cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes.[3]
For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits.

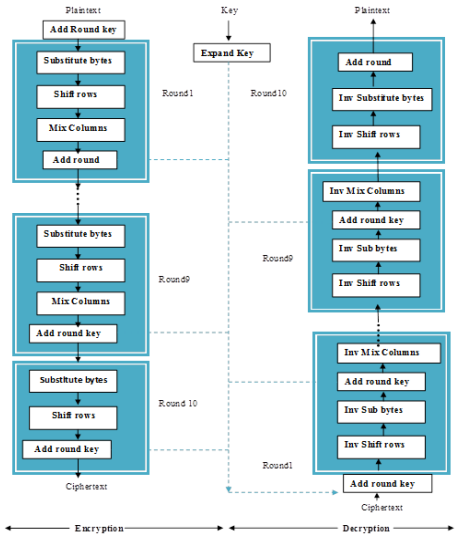

Figure 2. Flow chart of AES algorithm

IV. Scope

As discussed earlier there are 3 main bodies or actors in a cloud system viz. the end user, the service provider (SP) and the infrastructure provider (IP).Whenever SP selects an IP, there undergoes a risk assessment. Security of data is one the primary criteria among many others in that assessment. Many IPs claim with a non-disclosure agreement but SP cannot fully trust an IP on just basis of that agreement. So, in order to take an extra precaution SP should not directly upload the files received from the user as it is on the server of IP.

An encryption mechanism before uploading the data onto the cloud storage provided by the IP is must. This will definitely decrease the risk of security threat to a great extent. Moreover the alteration proposed in AES algorithm in this project will further increase the security of the entire system. The entire modification in the traditional AES algorithm is described in segment 4.2.

V. Proposed system

The traditional algorithm works on the four steps mentioned in section 1.4.

As seen in subsection 1.5.1, the lookup matrix for substitution is static i.e. it is fixed for each and every file. The substitution look table is called substitution box or S-box. This creates a huge vulnerability at the Service providers end. This is because if the attacker gets hold of the substitution box (S-box), he/she can easily exploit the vulnerability and access the data file. So in order to deal with this problem, this project gives a solution by creating Dynamic S-Box. This S-box will be constructed based on the encryption key provided by the end user.

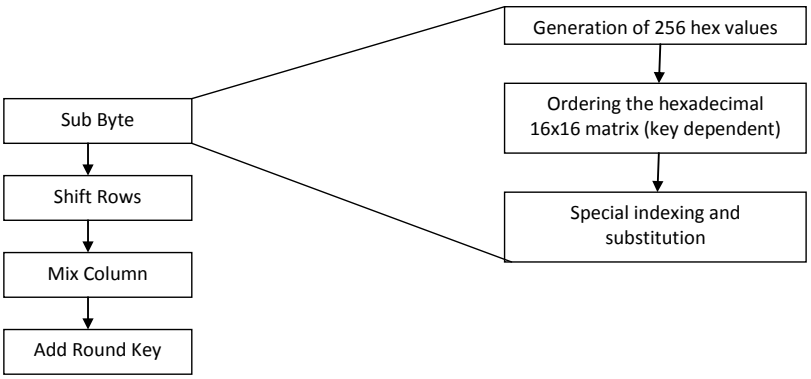

Figure 3. Proposed System

VI. Expected Outcome

6.1. Encryption Side

Generation of AES Key Dependent S-Boxes helps us to create a new S-box every time with even a minor change in the key. Table 1 shows the Dynamic S-box that would be generated after we use the proposed algorithm.

Table 1. Dynamic S-box

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

The ByteSub Transformation layer uses S-box to perform the byte substitute operation. AES defines a 16x16 matrix of byte values, called an S-box as given in Table (1) that contains a permutation of all possible 256 8-bit values.

Each individual byte of state is mapped into a new byte in the following way: The leftmost 4 bits are used as a row value and the rightmost 4 bits are used as a column value. These row and column values serve as indexes into the S-box to select a unique 8-bit output value.

For Example: The value {8A} signifies that {8} is the row number and {A} is the column number which results in the value that is to be substituted as "F1" when referred from Table 1.

6.2. Decryption Side

Here it is necessary to generate an inverse S box at the last step of decryption so as to get the original content of the state. The inverse S-box is shown in table 2.If the value of the encrypted state is {F1}, the first 4 bits i.e. {F} states the row index and the last four bits i.e. {1} states the column index. So, {F1} give us the value as {8A} from the inverse S-box depicted in table 3.

Table 2. Inverse S-box

6.3 Deliverables:

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

- **Avalanche Analysis:** The avalanche effect property is very important for encryption algorithm. This property can be seen when changing one bit in plaintext and then watching the change in the outcome of at least half of the bits in the cipher text. One purpose for the avalanche effect is that by changing only one bit there is large change then it is harder to perform an analysis of cipher text, when trying to come up with an attack.
- **Bit independence criteria (BIC) :**A second property which would seem desirable for any cryptographic transformation is that, for given set of avalanche vectors generated by the complementing of single plaintext bit, all the avalanche variable should be pair wise independent. In order to measure the degree of independence between a pair of avalanche variable, we calculate their correlation coefficient, if its zero it mean that the variable are independent, if its 1 that mean stronger positive correlation and -1 is stronger negative correlation.
- **Randomness Analysis**: In this test we use CrypTool to test randomness of AES-RC4 S-box and comparing it with AES S-box using same inputs for both S-boxes . There are multiple outcomes of randomness analysis viz. frequency, poker, long run and serial test.

Table 3. Expected results

Test	**Tradition AES Algorithm**	**AES with Dynamic S box on cloud**
Avalanche Analysis	0.4688	0.6
Bit independence criteria	-0.1530	-0.5
Frequency test	0.06	0.3
Poker test	5	8
Long run	22	20
Serial	4	7

The values of the analysis mentioned above for the traditional algorithm and the values we hope to achieve are given in table.

References

[1]. L. Scripcariu, ”A Study of Methods Used To Improve Encryption Algorithms Robustness”, IEEE 978-1-4673-7488-0/15/©2015.

[2]. Karim Djemame ,“A Risk Assessment Framework for Cloud Computing”,IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. X, NO. Y, SEPTEMBER 2013.

[3]. Xiang Li, “An Improved AES Encryption Algorithm”, Second International Conference on Emerging Applications of Information Technology, 2011.

[4]. Evgeny Pyshkin, “A Provisioning Service for Automatic Command Line Applications Deployment in Computing Clouds“,IEEE International Conference on High Performance Computing and Communications, 2014.

[5]. B.Thiyagarajan, ”Data Integrity and Security in Cloud Environment Using AES Algorithm”,ICICES 2014.

[6]. Takanori Machida, “Modifications to AES Algorithm for Complex Encryption”, IEEE transactions 2015.

[7]. C. Giraud, “An Implementation of DES and AES,Secure against Some Attacks”, c Springer-Verlag Berlin Heidelberg.CHES, LNCS 2162, pp. 309–318, 2001.

[8]. R. Buyya, C. ShinYeo, J. Broberg, I. Brandic, “Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Generation Comput. Syst., vol. 25, pp.599–616, 2009.

[9]. Dongyoung Kooa , Junbeom Hurb,HyunsooYoona ,“Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage”, CEE on Recent Advanced Technologies and Theories for Grid and Cloud Computing ,vol.39, pp. 34–46,2013.

[10]. Miss. Rehana Begum, Mr. R.Naveen Kumar, Mr. Vorem Kishore,”Data Confidentiality Scalability and Accountability (DCSA) in Cloud Computing “, Volume 2, Issue 11, November 2012.

[11]. R. Kandukuri, V, R. Paturi and A. Rakshit, “Cloud security issues,” in Proceedings of the 2009 IEEE International Conference on Services Computing, pp. 517-520, September 2009.