I J C T A, 9(41) 2016, pp. 1151-1159 ISSN: 0974-5572
© International Science Press

DATA ENCRYPTION WITH RANDOM KEY GENERATION IN CLOUD COMPUTING

Radhika Dhomane[1], Malay Kumar[1], Manu Vardhan[1] and Sanjeev Jain[2]

Abstract: Cloud computing is a buzz word in business and academia. It offers on-demand easy access of computing services over the Internet in pay-as-per use model. Besides the tremendous benefits maintaining the privacy and security of data is a challenging task. In this paper, we pro- posed a novel key generation technique, which is further utilized in the formulation of the encryption scheme. The proposed encryption scheme is compared with the existing state-of-the-art encryption schemes such as DES, AES and Blowfish on the parameters like key-strength, encryption time and avalanche effect. The result shows that the encryption time of the proposed scheme is significantly reduced as compared to the existing methods for the same size data blocks. There is improvement of (64% in plain-text and 57% with key) observed in the avalanche effect. We have also conducted a Brute-Force analysis of the proposed method, which shows significant improvement over the existing methods. The implementation result and the analytical analysis demonstrate the superiority of the proposed scheme.

Key Words: *Cloud, Key Generation, Brute Force, Avalanche Effect.*

I. INTRODUCTION

In this era of technology, cloud computing has come forth as a booming rescue to most of the computational problems in both the industry and academia and eventually popularized a new business computing paradigm on pay as per use model for permissive ubiquitous, on-demand access to a shared pool of various configurable computing re- sources (e.g., computer networks, servers, storage, applications, and services), which can be swiftly provisioned and released with least possible management effort [1]. It depreciates the investment burden for infrastructure, software, hardware or for any kind of resource in an organization. Many users approx 95% of the big and small organization have 93% of data in digital form and prefer to pool their data and application in the cloud as it provides global access, large storage and ease of use [2]. However due to this liberty in accessing the shared data by large users the confidentiality of data is affected. The most important concern becomes the security of data which in one way can be attained by some encryption algorithm. This procedure stores data in morphed form thus guarding the confidentiality and maintains the authenticity [3-8]. Cryptography plays an important role in various scenarios like the military applications, e-banking, medical databases and the various personal information like e-mails [3].

The main module of a cryptographic system is the key generation module along with the encryption and decryption module. Key generation module does the most crucial task of generating the key as it defines the strength of the encrypting algorithm [9, 10]. Depending on the key used for encryption and the decryption purpose we have two categories of cryptography as symmetric cryptography and asymmetric cryptography. If a single key is used for both the encryption and the decryption purpose it is a symmetric encryption scheme. This scheme has a major advantage of the higher speed and a better performance. Now when two different keys are

[1]*Department of Information Technology, National Institute of Technology, Raipur, Chhattisgarh, India*
Emails: radhikadhomane@gmail.com, {mkumar.phd2014.cs, vardhanmanu}@gmail.com
[2]*Department of Electronics & Communication Engineering, GEC Bikaner, Rajasthan, India*
Email: snjece@gmail.com

used for encryption and decryption it is an asymmetric encryption. Symmetric cryptography is even classified on the basis of cipher it generates as the block cipher and the stream cipher. The block cipher operates on the data in groups or blocks, stream ciphers only operate on a single bit data [11,12]. Cryptology essentially combines cryptography and cryptanalysis. Cryptanalysis attempts to deduce the specific plain text or the specific key. It is used to measure the vulnerability of any algorithm under some circumstances [13,14].

In the proposed method parameters like the encryption time, avalanche effect and strength of the key is measured. The encryption time is compared depending on the average number of times the algorithm is executed and the values obtained. The avalanche effect shows the variance in bits of cipher text depending on the changes made in the key bits and the plaintext bits. For measuring the key strength, we have used the Brute Force attack on some of the existing symmetric block cipher techniques and the proposed scheme. The proposed scheme is found to be more efficient than the existing ones.

The remainder section of the paper consists of as follows. Section II is related works having description of existing DES,AES and the Blowfish algorithms and about the key generation. In next section, the key generation scheme is proposed followed by encryption and decryption schemes in the next two consecutive subsections. Section III has security analysis of the key and section IV shows the experimental outcomes followed by conclusion in section V.

II. RELATED WORK

Designed in the year 1973 by IBM, DES was the first encryption scheme to be approved by NIST (National Institute of Standards and Technology) in 1976 describing the most well-organized method for encryption. This became the most widely used standardized scheme across the world [15]. As described by Davis R. it takes a fixed length string of plaintext bits and then converts it into a ciphertext of the same length through a series of operations. It has a block of 64 bits and uses a key of 56 bits to get the transformations done. DES can work in CBC, ECB, CFB and OFB modes. There are 16 Feistel rounds of identical nature involving the f function and the s boxes. The 64-bit plaintext block is divided into 32 bits each and is passed through the various rounds. However, the drawback of this algorithm is it is susceptible to the Brute Force attack with the key of 56 bits that can be cracked in 22 hours [16-18].

A replacement for DES was needed, due to small key size and vulnerability to security breach. AES became the solution with the variable key size of 128,192,256 bits and works on bytes rather than bits. It has an iterative substitution and permutation mechanism rather the Feistel networks. For 128 bits, it processes as a 4 column 4 bytes' data block for every 16 rounds. However, the encryption and decryption process are not just the reversal of keys but the order changing of the used blocks in the encryption scheme. The AES scheme was designed by Vincent Rijmen and Joan Daemen in Belgium [19-21].

Blowfish algorithm designed by Bruce Schneier in 1993[22] uses a variable length key of 32 bits to 448 bits and a block size of 64 bits. It has two parts of key expansion and data encryption. There are 18 P arrays and 4 S-boxes initialized to hexadecimal digits of π. Overall there are 16 rounds of the process considering initial 32 bits of the plain text and the corresponding bits of the key and an f function. The process is repeated on the next bits to get the cipher. One plus point of blowfish algorithm is the simplicity of the involved operations as XOR and addition giving a faster speed up for encryption [23-27].

There are various random key generation mechanisms like the BLUM SHUB key generator [28] and the Nothing up my sleeves key generators [29] for the random binary bit strings. Hung-Yu Chien [30] and Sean-Philip Oregano [31] presented time coupled hierarchical key assignment scheme for significantly improving the computational performance at the cost of implementation.

Cryptanalysis is breaking the encrypting scheme or the key. Eli Biham and Adi Shamir introduced attack against DES on the S-boxes structure. by using linear approximations the action of the block cipher can be judged making it prone to linear cryptanalytic attack as given by Mitsuru Matsui [32].

III. PROPOSED METHOD

3.1 Key Generation-

In this proposed encryption algorithm, a key generator γ(L') will generate two 128 bit keys which are random in nature. Symmetric key is taken because it is much faster and efficient as compared to asymmetric encryption technique, hence leads to overall improvement in time complexity of the proposed encryption algorithm. The key generator will generate two keys as:

$$\gamma(L') \xrightarrow{generates} k_i \quad \forall\, i(1 to 2)\ and\ \ k_i \in (2^{L'-1}\ to\ 2^{L'}), \text{ where L'=128} \tag{1}$$

Let m bit plain text is divided into 128 bits block such that

$$\phi(M) = \text{mod}(m,128) \tag{2}$$

$$\phi(M) = \begin{Bmatrix} 0 \ \rightarrow\ no\ padding\ required \\ otherwise \rightarrow padding\ required \end{Bmatrix}$$

Therefore,

$$M \rightarrow \{m_1, m_2, m_3, \ldots\ldots m_{n-1}, m_n\}$$

$$K_1 \rightarrow \{k_{1,1}, k_{1,2}, k_{1,3}, k_{1,4} \ldots\ldots k_{1,n}\} \qquad K_2 \rightarrow \{k_{2,1}, k_{2,2}, k_{2,3}, k_{2,4} \ldots\ldots k_{2,n}\}$$

The resultant of these two key will be determined by EXORing K_1 and K_2 which will be used for selecting the key.

$$K' = K_1 \oplus K_2$$

$$K' = \{k_{1,1} \oplus k_{2,1}, k_{1,2} \oplus k_{2,2}, k_{1,3} \oplus k_{2,3}, \ldots\ldots k_{1,n} \oplus k_{2,n}\} \tag{3}$$

$$K' = \{k_1' + k_2' + k_3' + k_4' \ldots\ldots + k_n'\}$$

where,

$$k_1' = k_{1,1} \oplus k_{2,1} = \overline{k_{1,1}} \cdot k_{1,2} + k_{1,1} \cdot \overline{k_{1,2}} \tag{4}$$

Now, if MSB bit of k'_1 and m_1 are equal then subkey of k_1 will be selected of that sub plain text i.e. $k_{1,1}$ for the encryption process.

Key Generation Algorithm

```
BEGIN
STEPS:
1. Read Plaintext
2. Call φ(p)having p bits
3. Initialize random keys K1 and K2
4. Compute K as K1⊕K2
5. If MSB(K)=MSB(mi)
       Select K1
    Else
       Select K2
 END
```

3.2 Proposed Encryption Technique-

In this Proposed Encryption Technique 'm' bit plain text is divided into 128-bit block which is further divided into 16-bit blocks by the splitter. From key generator, two random key of 128 bits are generated which are divided into 16-bitblocks. Initially first 16 bits of both the keys are EXORed and then MSB bit of resultant key is compared with MSB bit of plain text block, if found same then first key will be selected or the second key is selected in the selection module. The cipher generated from first EXORing operation is used to select key for next block as it is EXORed with a resultant key which is formed by EXORing the two random keys and then the comparison is made for selecting key. The selected key is EXORed with plain text and the process goes on till whole 'm' bit plain is encrypted. For each plaintext block of 128 bits, the key generator will generate different keys. Finally, in the end an ORing operation is done between the generated ciphers and the 0-bit stream to get the complete ciphertext.

Suppose the key selected for encryption be

$$K_s \rightarrow \{k_{1,1}, k_{1,2}, k_{2,3}, \ldots\ldots\ldots k_{1,n-1}, k_{2,n}\} \tag{5}$$

Now EXORing the plain text with selected key, we get

$$C_i \rightarrow m_i \oplus k_{i,j} \tag{6}$$

$$C = \{ k_{1,1} \oplus m_1, k_{1,2} \oplus m_2, k_{1,3} \oplus m_3, \ldots\ldots k_{2,n} \oplus m_n \}$$

$$c_i \rightarrow m_i * \overline{k_{1,j}} + \overline{m_i} * k_{i,j}$$

From the property of the complement, it can be concluded that $k_{1,1} = 1 - \overline{k_{1,1}}$ & $m_i = 1 - \overline{m_i}$, therefore above equation can be written as:

$$c_i \rightarrow m_i * (1 - k_{i,j}) + (1 - m_i) * k_{i,j}$$

$$c_i \rightarrow m_i + k_{i,j} - 2 * m_i * k_{i,j}$$

So for 'n' sub plain text it can be expressed as:

$$c_1 \rightarrow m_1 + k_{1,1} - 2 * m_1 * k_{1,1}, c_2 \rightarrow m_2 + k_{1,2} - 2 * m_1 * k_{1,2},$$

$$c_3 \rightarrow m_3 + k_{2,3} - 2 * m_1 * k_{2,3} \text{ and so on } c_n \rightarrow m_n + k_{2,n} - 2 * m_1 * k_{2,n}$$

Overall cipher text is resultant of all cipher text and is generalized form it can be represented as

$$C \rightarrow M + K_{i,j} - 2 * M * K_{i,j} \quad (7)$$

Where $C \rightarrow c_1 + c_2 + c_3 \ldots\ldots\ldots + c_n$ and $M \rightarrow m_1 + m_2 + m_3 \ldots\ldots\ldots + m_n$

Therefore,

$$C = \{c_1, c_2, c_3 \ldots\ldots\ldots c_n\}$$

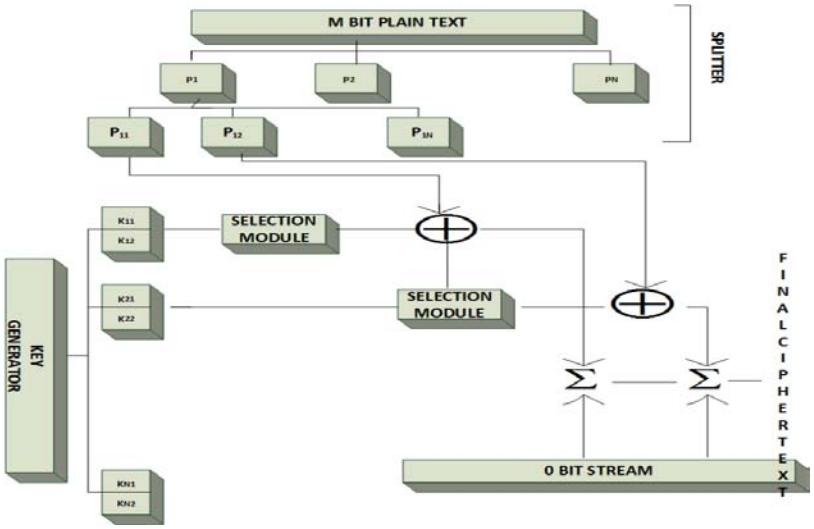

Figure 1. Block Diagram of Proposed Encryption Technique

3.3 Decryption Technique-

For decryption, the two randomly generated keys and cipher text are EXORed and the plaintext can be recovered. An additional 8 bits array of selected key denoted by 0 or 1 is added to any one of the random keys and these 8 bits are used to select the 16-bit block from 128 bit of two random keys. The message will be recovered by:

$$m_i = c_i \oplus k_{i,j}$$

3.4 Security Analysis-

The security of the key is measured by the Brute Force attack which is an exhaustive search or generates and test technique that consists of semantic listing all feasible solutions and looking for the most appropriate for the problem's statement.

In the case of cryptography, a brute-force attack involves checking all possible keys until the correct key is found [33]. The key length used in the encryption determines the practical feasibility of performing a brute-force attack, with longer keys exponentially more difficult to crack than shorter ones.

Table- 1 Brute Force Analysis of the schemes

Encryption Technique	Key Size	Key Combination	Decryption rate at 1 Key/µsec	Decryption rate at 10^6 Key/µsec
DES	56	7.2×10^{16}	2.0×10^7 hrs	2.0 x 10 hrs
AES	128	3.4×10^{38}	8.8×10^{28} hrs	8.8×10^{22}hrs
Blowfish	64	1.8×10^{19}	5.0×10^9 hrs	5.0×10^3 hrs
Proposed Encryption Technique	128(each)	1.1×10^{77}	3.2×10^{67}hrs	3.2×10^{61} hrs

IV. Experimental Results

The experimental analysis of the proposed algorithm is based on mathematical and theoretical discussion presented in the previous sections. The algorithm is implemented on MATLAB simulator 2016a.The algorithm has been tested on the system of configuration CPU Intel® Core ™ I5 (CPUs)~1.8 GHZ 8 GB Ram. The section below shows the results obtained.

4.1 Encryption Time-

On comparing the proposed scheme with the DES,AES and Blowfish we conclude the following in terms of the time needed for the process of encryption. Following graphs shows the comparison.

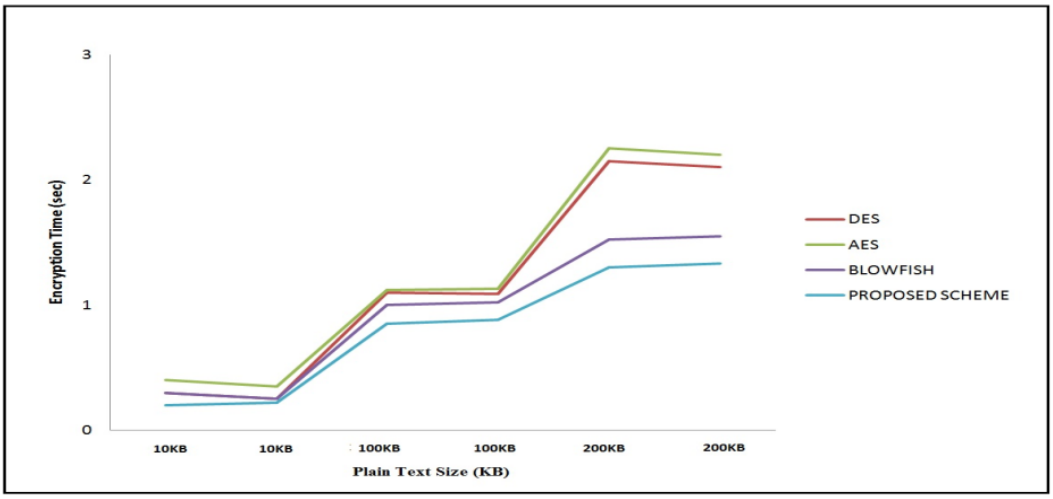

Figure 2. Comparison of various Encryption Schemes

4.2 Avalanche Effect-

The avalanche effect has a condition that slight change(a single bit) in input changes the output that is cipher significantly(half of the bits).The change in input can be either in the key used or the

plain text itself. It is one of the most desirable property for an algorithm to keep it safe from cryptanalysis.

In proposed encryption technique to measure the avalanche effect 128-bit block of plaintext is taken corresponding to its 128 bit key and then 1 bit is flipped and the number of bits that are changed in the cipher text is counted.

The avalanche effect is calculated by:

$$\text{Avalanche effect } (A_v) = \frac{\text{Number of bits changed in cipher text}}{\text{Total number of bits in ciphertext}}$$

Table- 2 Avalanche effect for the scheme

Encryption Technique	Change in 1 bit of key (Keeping plain text constant)	Avalanche Effect(A_v)	Change in 1 bit of plain text (Keeping key constant)	Avalanche Effect(A_v)
DES	**30**	**0.53**	**34**	**0.53**
Blowfish	**37**	**0.29**	**23**	**0.18**
AES	**64**	**0.50**	**71**	**0.55**
Proposed	**73**	**0.57**	**83**	**0.64**

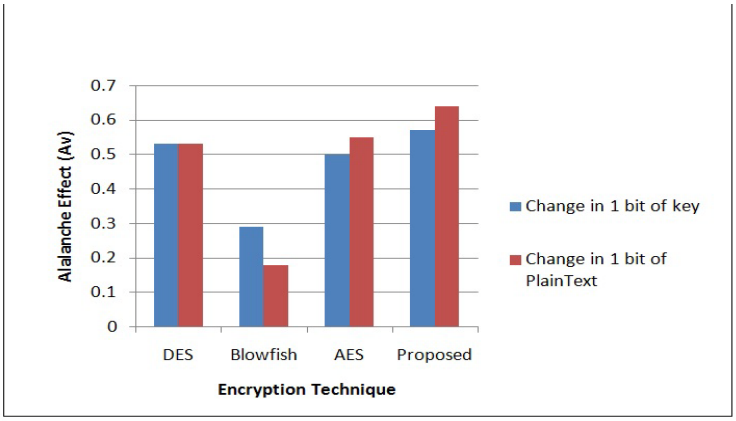

Figure 3. Avalanche effect on various Encryption Schemes

V. CONCLUSION

Thus, in the today's world, the spotlight from mobility has changed the way we use technology. Accessing data from everywhere, outsourcing or delegation of the data is preferred by the businesses as well as individuals. In order to safeguard this data, a blossoming science of technology emerged as Cryptography. Various existing algorithms like DES, AES and Blowfish are studied and compared with the proposed scheme. The performance of the proposed scheme is compared with the parameters like time for encryption for the various size of the blocks, the avalanche effect on the plain text which is 64 percent and with the key is 57 percent. The key

strength is compared on the basis of cryptanalysis attack as brute force .On all of these mentioned parameters the given scheme is found to be better than the existing.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Draft)," NIST Special Publication, Vol. 800, No. 145, 2011, p. 7.

[2] N. Sengupta and J. Holmes, "Designing of cryptography based security system for cloud computing," Proc. - 2013 Int. Conf. Cloud Ubiquitous Comput. Emerg. Technol. CUBE 2013, pp. 52–57, 2013

[3] A. N. Jaber and M. F. Bin Zolkipli, "Use of cryptography in cloud computing," Proc. - 2013 IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2013, pp. 179–184, 2013.

[4] I. Sriram and A. Khajeh-Hosseini, "Research Agenda in Cloud Technologies," 1st ACM Symp. Cloud Comput. SOCC, vol. cs.DC, pp. 1 11, 2010.

[5] T. E. Issue, "Vulnerabilities," 2012.

[6] D. Nurmi et al., "The eucalyptus open-source cloud-computing system," 2009 9th IEEE/ACM Int. Symp. Clust. Comput. Grid, CCGRID 2009, pp. 124–131, 2009.

[7] S. Ullah and Z. Xuefeng, "TCLOUD: A Trusted Storage Architecture for Cloud Computing," Int. J. Adv. Sci. Technol., vol. 63, pp. 65–72, 2014.

[8] Joe Wilson " Developing a Framework to Improve Critical Infrastructure Cybersecurity ," no. March, 2013.

[9] M. Agrawal and P. Mishra, "A comparative survey on symmetric key encryption techniques," Intern. J. Comput. Sci. Eng., vol. 4, no. 5, pp. 877–882, 2012.

[10] M. G. V. Kumar, "A Survey on Current Key Issues and Status in Cryptography," pp. 0–5, 2016.

[11] G. Singh, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security," Int. J. Comput. Appl., vol. 67, no. 19, pp. 975–8887, 2013.

[12] M. Khari, "Secure Data Transference Architecture for Cloud Computing using Cryptography Algorithms," pp. 2141–2146, 2016.

[13] Dr. Sharada, "Cryptography and Network Security," 2014.

[14] C. Tan and Q. Ji, "An Approach to Identifying Cryptographic Algorithm from Ciphertext," pp. 19–23, 2016.

[15] B. Schneier, "Schneier on Security," wwschneier.com, p. 336, 2008.

[16] S. Singh, "International Journal of Advanced Research in Computer Science and Software Engineering Enhancing the Security of DES Algorithm Using Transposition Cryptography Techniques," nternational J. Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 6, pp. 464–471, 2013.

[17] S. Key, "Chapter 2 The Data Encryption Standard (DES)," pp. 10–21, 1970.

[18] H. Feistel, "Data Encryption Standard (DES)," Fips Pub 46-3, vol. 3, 1999.

[19] S. Mewada, P. Sharma, and S. S. Gautam, "Exploration of efficient symmetric AES algorithm," 2016 Symp. Colossal Data Anal. Networking, CDAN 2016, pp. 4–8, 2016.

[20] A. Kak, "Lecture 8 AES : The Advanced Encryption Standard Lecture Notes on ' Computer and Network Security ' by Avi Kak (kak@purdue.edu) Goals To review the overall structure of AES and to focus particularly on the," no. 3, 2015.

[21] J. Daemen and V. Rijmen, The Rijndael Block Cipher: AES Proposal. 2003.

[22] B. Schneier, "Schneier on Security," Wwwschneiercom, p. 336, 2008.

[23] O. Access, "Comparison of Blowfish and Cast-128 Algorithms Using Encryption Quality , Key Sensitivity and Correlation Coefficient Analysis," no. 7, pp. 161–166, 2014.

[24] A. Alabaichi, F. Ahmad, and R. Mahmod, "Security analysis of blowfish algorithm," 2013 2nd Int. Conf. Informatics Appl. ICIA 2013, pp. 12–18, 2013.

[25] P. C. Mandal, "Superiority of Blowfish Algorithm," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 2, no. 9, pp. 196–201, 2012.

[26] R. K. Meyers and A. H. Desoky, "An Implementation of the Blowfish Cryptosystem ofattWindtowsreak toolwforse ing fimles enwh tich," pp. 346–351, 2008.

[27] A. Mousa, "Data encryption performance based on Blowfish," 47th Int. Symp. ELMAR, 2005., no. June, pp. 131–134, 2005.

[28] L.Blum , M.Blum and M.Shub, "A SIMPIE UNPREDCTABLE PSEUDO-RANDOM NUMBER GENERATOR*," Soc. Ind. Appl. Math., vol. 15, no. 2, p. 20.

[29] Bruce Schneier, "Schneier on Security," A Million Random Digits. 2006.

[30] N. Kumar, A. Mathuria, and M. Das, "An Efficient Time-Bound Hierarchical Key Assignment Scheme," pp. 1–15.

[31] Sean-Philip Oriyano, J. M. Tanna, M. P. Sanghani, M. Ayushi, and R. J. Anderson, "A Symmetric Key Cryptographic Algorithm," Int. J. Comput. Appl., vol. 1, no. 15, pp. 73–114, 2010.

[32] J. Jia, J. Liu, and H. Zhang, “Cryptanalysis of Cryptosystems Based on General Linear Group,” no. June, pp. 217–224, 2016.
[33] Cryptanalysis and Design of Symmetric Cryptographic Algorithms, no. March. 2011.