International Journal of Advanced Science and Technology
Vol.129 (2019), pp.1-10
http://dx.doi.org/10.33832/ijast.2019.129.01

DATA SECURITY USING 512 BITS SYMMETRIC KEY BASED CRYPTOGRAPHY IN CLOUD COMPUTING SYSTEM

Bijoy Kumar Mandal[1], Debnath Bhattacharyya[2] and Xiao-Zhi Gao[3]

[1]*NSHM Knowledge Campus, Durgapur, India*
[2]*Computer Science and Engineering Department,*
Vignan's Institute of Information Technology,
Visakhapatnam-530049, India
[3]*School of Computing, University of Eastern Finland,*
FI-70211 Kuopio, Finland
writetobijoy@gmail.com, debnathb@gmail.com, xiao.z.gao@gmail.com

***Abstract*— Cloud Computing is a very new and advance technology that is growing very fast today. It is also belongs to distributed architecture that have centralized data space (Server) to store the data. In this paper, we mainly focus on data security. There are so many security levels like browser security, information security, virtual machine security, data storage security. Data access or information transmission issues and attacks are more related with data security that is provided to the users when they are handling the data for different issue.There are lots of security issues for cloud computing to handling the data including more technologies of networks, databases system or storage, resource handling schedule, operating systems, controlling, load balancing, and virtualization. The network system including cloud concept must have secure technology for mapping from virtual machines to the physical machines. There is proposed an algorithm for Data security including data encryption technology and used for data sharing or data transmission.**

***Keyword*— CSS, CSP, Cryptography, Symmetric Key, KGA**

1. INTRODUCTION

Cloud computing system is also known as distributed computing system and it is based on Internet that is used for sharing and processing data to the system on demand. Cloud Service System (CSS) is a new concept that is used on-demand network access to a shared pool of configurable computing data over enable network on demand for storage and application [1, 2]. The system is enabling to share the raw data for achieving the target over a grid network system. CSS allows to allfor avoiding the huge costing that is used to upfront infrastructure and focus only on projects. CSS also claim that cloud computing involves to enterprises or system for gettingown applications and runningfaster with less maintenance that enables IT to adjust the data resourcesrapidly. CSS has become a highly demandable service provider due to the more benefits of high computing speed and less cost of services.However, field for CSS is fast growing in terms of size and productivity. It also rises to potentially risk regarding the security concerns with other issue.

Received: May 26, 2019
Reviewed: July 18, 2019
Accepted: August 2, 2019

ISSN: 2207-6360 IJAST

The Solutions for security in Cloud Computing System (CCS) including policy, legislation and choices of users for transmission of data must be strong. Users can encrypt data that is transmitted or stored within the CCS to secure the data from unauthorized access [3].Information Security is the foremost requirement to ensuring capable security [4] on any cloud computing based platform, be it for communication or any other cloud-based applications. The transmition of information is the primary objective towards achieving a secure aspect of a profoundly progressive technological advancement. This can be achieved by implementing various cryptographic methods and techniques [5], either in a single method form or multiple or complex variations of it.The strong systemof cryptographyshould able to save from different types of attack, which includedifferential cryptanalysis, linear cryptanalysis andkey cryptanalysis [6, 7].

2. PREVIOUS WORK

There are multiple encryption algorithms which have been used primarily for cloud computing to increase Information Security. These algorithms include both Public Key Cryptography and Secret-Key Cryptography [8]. Over time, drawbacks of each of the methods and techniques used have come to light, thereby, calling for the designing of better and more robust algorithms.

A. Existing Security Algorithms

For more security in communication through the network, method of encryption is the fundamental tool for securing and protecting the data. The data are converted into scrambled form by using "the key" using encryption algorithm and only user have the key to decrypt the data [9]. There are two types of encryption method. In one method, we use only one key for encryption and decryption the data, is called as Symmetric key encryption and in other method, we use two keys as private and public keys, is known as asymmetric key encryption. The public key is applied mainly for encryption and private key is taken mainly for decryption [10, 11].

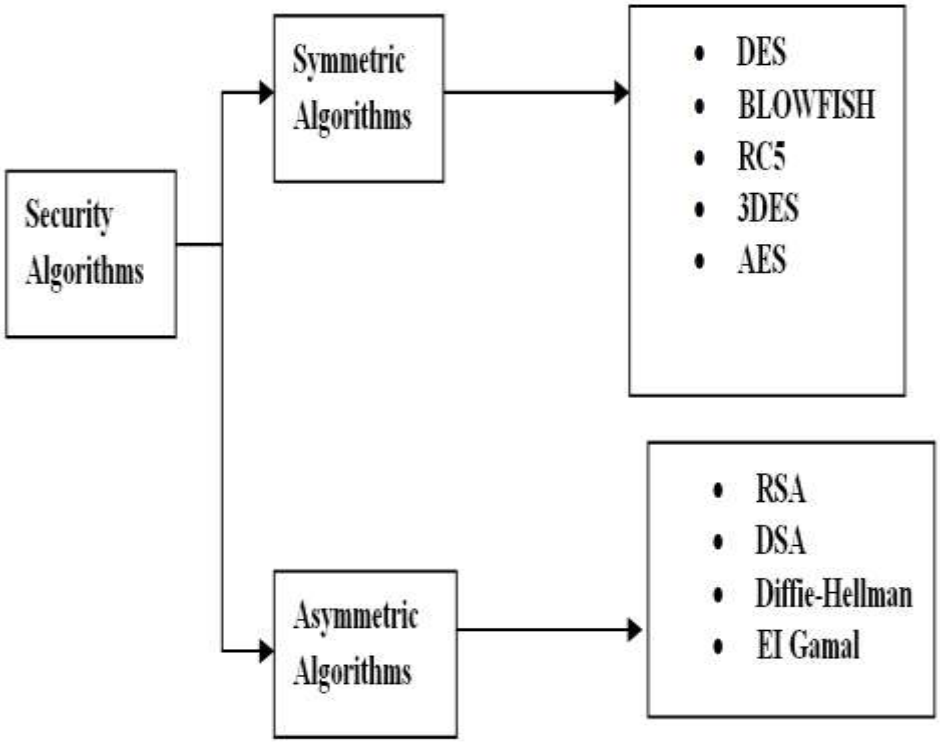

Fig. 1 The symmetric & asymmetric algorithms

B. Different Issues in Cloud Computing

There are more issues that are started from the user loses control of own data, because it is kept into a computer that is related to other as shown in Table I.

Table I. Different Issues in Cloud Computing

Sl. No	**Type of Issues in mobile cloud computing**	**Methods under different types of Problem**	**Description about issue**	**Problems arise in particular type**
1	Operational issue	Offloading method	Offloading job from mobile device to the cloud.	1. For distance from the cloud to mobile device. 2. For system that is related with heterogeneity policy
2	End user issue		This explain the issues related with last user forparticipating, and cost.	1. Incentives to collaborate 2. Usability and presentation problems
3	Service and application level issue		The problem is mainly related to throughput of system and QOS.	1. Availability 2. Fault tolerance
4	Privacy, security and trust	1. Simple cloud security 2. Mobile security in cloud 3. Privacy	It deals regarding issues of data storage orcomputation using cloud for mobile device	1. Low bandwidth 2. Availability 3. Heterogeneity 4. Low capacity
5	Data management		Information may be stored, shared andaccessed with external user or device	1. Data access 2. Data portability 3. Interoperability

C. Disadvantage of AES 256

The new process of attacks is combined for boomerang and the rectangle attack. This uses the weaknesses of few nonlinear transformations in the key schedule algorithm of ciphers and it can break some reduced-round versions of AES [11, 12].

Rijndael inherits many properties from Square algorithm [13]. So, the Square attack is also valid for Rijndael which can break round-reduced variants of Rijndael up to 6 or 7 rounds (*i.e.*, AES-128 and AES-192) faster than an exhaustive key search [14] proposed some optimizations that reduce the work factor of the attack

3. MATHEMATICAL EXPRESSION

The input data would be communicated as data blocks of 128 bytes and it will be combined with the inputs of polynomial:

$$s(x)=\sum_{i=0}^{7}\sum_{j=0}^{Nc-1}(s_{ij}x^{ij})$$

The first two steps of AES (i.e. First and Second steps) are done by the proposed algorithm in same way. For the third step of MAES, we propose the following polynomial:

$$a(x) = x^7+ 2x^6+ 3x^5+ 4x^4+ 5x^3+ 6x^2+ 7x + 8$$

The encryption polynomial has a higher order in comparison to the classic AES algorithm, considering the increased number of rows, from 4 to 8:

$$a(x) = \sum_{i=0}^{7} a_i x^i$$

For column multiplication step, it may be used the encryption matrix *A*, given below:

$$A = \begin{vmatrix} 13 & 7 & 5 & 4 & 4 & 3 & 2 & 1 \\ 2 & 13 & 7 & 5 & 4 & 4 & 3 & 2 \\ 7 & 2 & 13 & 7 & 5 & 4 & 4 & 3 \\ 5 & 7 & 2 & 2 & 7 & 5 & 4 & 4 \\ 6 & 2 & 2 & 6 & 9 & 4 & 7 & 5 \\ 3 & 1 & 7 & 6 & 2 & 10 & 6 & 6 \\ 5 & 4 & 4 & 3 & 2 & 1 & 8 & 7 \\ 10 & 6 & 7 & 4 & 3 & 2 & 1 & 8 \end{vmatrix}$$

It is 8*8 matrix based on the invertible polynomial $a(x)$and it isgiven for the polynomial expression for $a(x)$. Its determinant is computed on GF (256) which is basically Galois Field for 256 bit and equal to 242. So it is an invertible matrix. The multiplication of the two polynomials on GF (256) is made the modulo-$p(x)$, where $p(x)$ can be x^8+1 or another 8-degree primitive polynomial and it is obtained based on the primitive element of GF (256) equal to 3 and its minimal polynomial:

$$p(x)= M_3x = x^8+ x^4+ x^3+ x^2+ 1$$

The encoded polynomial is computed on GF (256), as:

$$c(x) = \{a(x).s(x)\} \text{ mod- } p(x) = \sum_{i=0}^{7} c_i x^i$$

Its coefficients are derived and it is written using the invertible encryption matrix *A*of 8*8 elements and two column-vectors of 8 coefficients each, one is a column of the state-matrix *S* and the other is the output encoded column-vector:

$$\bar{c}= A.\ \bar{s}$$

The encryption key is written as a key-matrix *K*, having the same dimensions as the state-matrix (*S*), and then the sum of the matrices is computed on GF (256), according to the following relation:

$$E = K \text{ XOR } C$$

The decryption algorithm makes the inverse operation:

$$\bar{s} = A^{-1}.\bar{c}$$

An algorithm for matrix inversion, on GF (256), is applied to compute the decoding matrix A^{-1}:

$$A^{-1} = \begin{vmatrix} 4 & 9 & 177 & 145 & 195 & 84 & 19 & 151 \\ 107 & 44 & 216 & 46 & 123 & 60 & 62 & 78 \\ 47 & 195 & 27 & 54 & 206 & 183 & 173 & 104 \\ 182 & 202 & 228 & 72 & 96 & 10 & 121 & 86 \\ 129 & 95 & 137 & 7 & 11 & 19 & 35 & 174 \\ 51 & 169 & 162 & 180 & 34 & 15 & 76 & 68 \\ 143 & 183 & 49 & 204 & 206 & 50 & 237 & 51 \\ 32 & 180 & 197 & 64 & 87 & 34 & 52 & 108 \end{vmatrix}$$

4. PROPOSED ALGORITHM

To increase the robustness of the encryption algorithm, we use longer key expansion and larger data matrix. To keep the processing time at low values, we have to maintain the complexity of the AES algorithm upon which the proposed algorithm is partially based upon. We generate a random key to be used on the algorithm and instead of a fixed 4*4 data matrix; we use a data matrix of 8 rows and 8 columns. The algorithm works on a block cipher code.

First of all, we need to consider more regarding details ofthe generation of the Round Key dynamically. Since there is much differences one for each round in the process. Finally, it is used for generation of a set of twenty-five 512-bit keys by Key Generation Algorithm (KGA). This key will be combined with the information during encryption of information or data. Although there are twenty-four rounds, twenty-five keys are needed because one extra key is need to be added with the initial state of array before starting the rounds for encryption.

A. KGA ( )

1. Start

2. Generate 512-bit secret key using PRNG

3. Store the encrypted key in a memory location

4. Call step 3 as and when required

5. Stop

B. Encryption Algorithm

1. Start

2. Call the file to be encrypted

3. Break the data into 128-bit cipher blocks containing plaintext

4. Cipher blocks are taken at a time

5. Encryption round starts (24 Rounds)

6. Generate 8*8 State Matrix to store the cipher data

7. Permute the State Matrix by shifting each row as per the row number

7.1. For row=0, no shift

7.2. For row=1, right-shift 1 element

7.3. Continue until row 8, right-shift 8 elements

8. Use 8-degree polynomial imposed generated by Rjindael

9. Take each column vector

10. Perform multiplicative operation with S-Box matrix created

11. Continue this for the entire matrix

12. Calling the KGA( )

13. XOR Key column vector with step 11 column vectors

14. Step 5 till Round 24

15. Step 4

16. Store the Encrypted Data

17. Stop

C. Decryption Algorithm

1. Start

2. Call the encrypted file to be decrypted

3. Break the data into 128-bit cipher blocks containing plaintext

4. Cipher blocks are taken at a time

5. Decryption round starts (24 Rounds)

6. Generate 8*8 State Matrix with cipher data

7. Call the KGA( )

8. Reverse XOR Key with Column Vector of State Matrix

9. Continue for entire state matrix

10. Take the column vectors generated in Step 9

12. Multiplicative function with inverse of invertible S-Box matrix created

13. Continue for end of state matrix columns

14. Use 8-degree polynomial generated by Rjindael

15. Permute the State Matrix to get the decrypted date

15.1. For row=0, no shift

15.2. For row=1, left-shift 1 element

15.3. Continue till row=8, left-shift 8 elements

16. Step 5 till 24 Rounds

17. Step 4

18. Store the decrypted data

19. Stop

5. SIMULATION

We use a GUI developed on Java Swing to display the front end operation of the algorithm. The form is designed so that it contains the following components:

1. Sender Side – containing JPanels for Enter Text and Cipher Text: The Panels are used to implement the data to be encrypted as well as the Cipher Text generated in Figure 1.

2. Encrypt Button: The button or JButton in Swing JAVA is used to trigger the encryption process. It encrypts the data concerned in Figure 1.

3. Client Side – containing JPanels for Enter Keyword and Decipher Text: The Panels are used to verify the Key included as well as the deciphered data if the key is the same as that of the generated key in Figure 1.

4. Decrypt Button: The JButton in Swing JAVA is used to trigger the decryption process so as to implement the decryption of the data in Figure 1.

5. Monitor: It is used to display the state matrix containing the input data as well as the data that is the intermediate data during the operations in Figure 1.

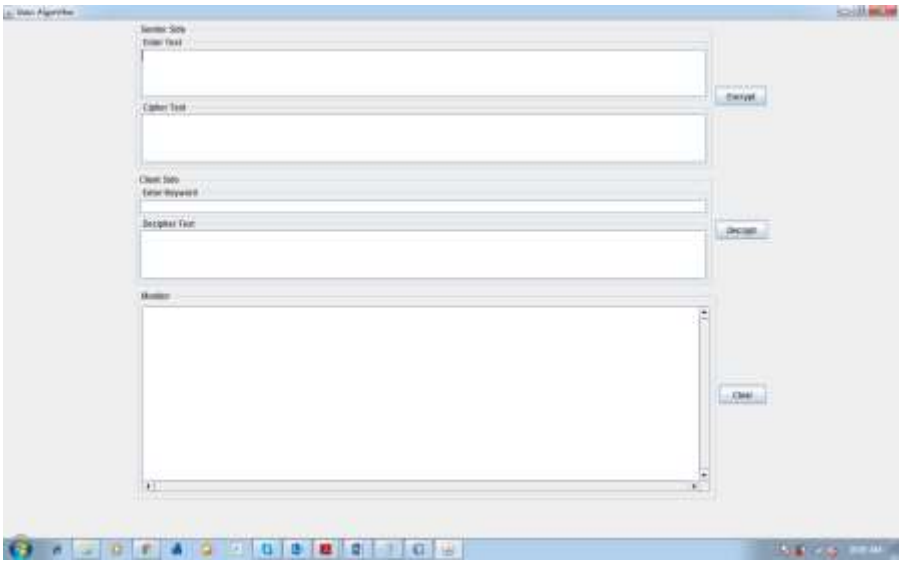

Fig. 1 The General Layout of Encryption & Decryption Process

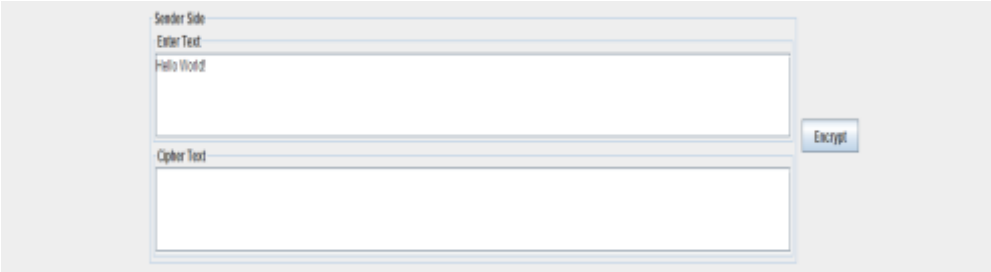

Fig. 2 The text to be encrypted in Enter Text panel

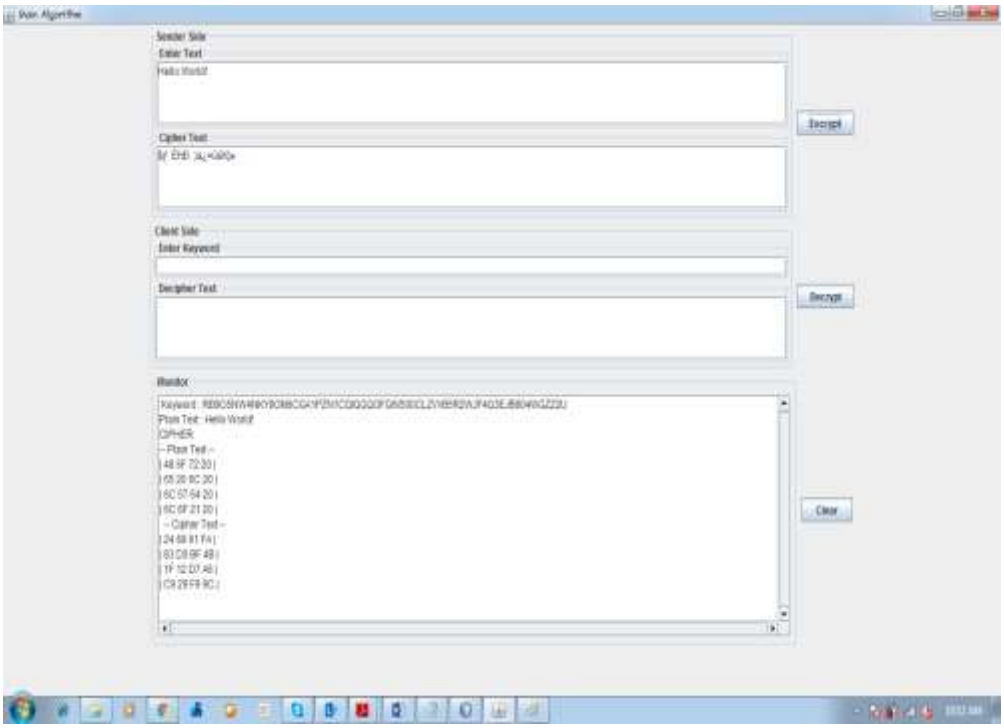

Fig. 3 Press the Encrypt button to start the encryption and the Cipher Text is displayed along with the key at the Monitor with the Plain Text State Matrix.

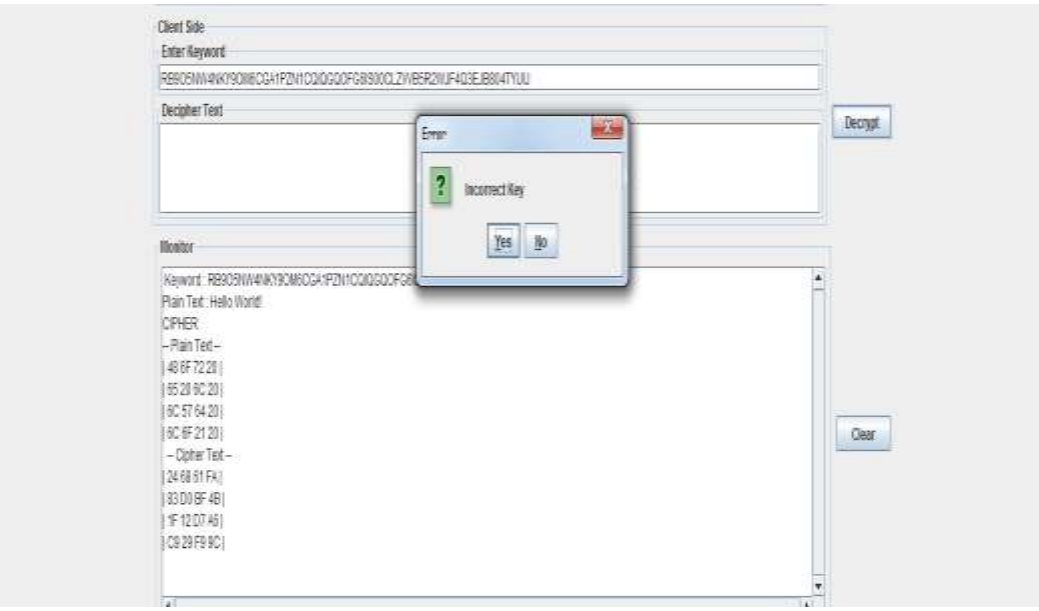

Fig. 4 When the client fails to provide the symmetric key, a pop-up message comes up requesting for the correct key.

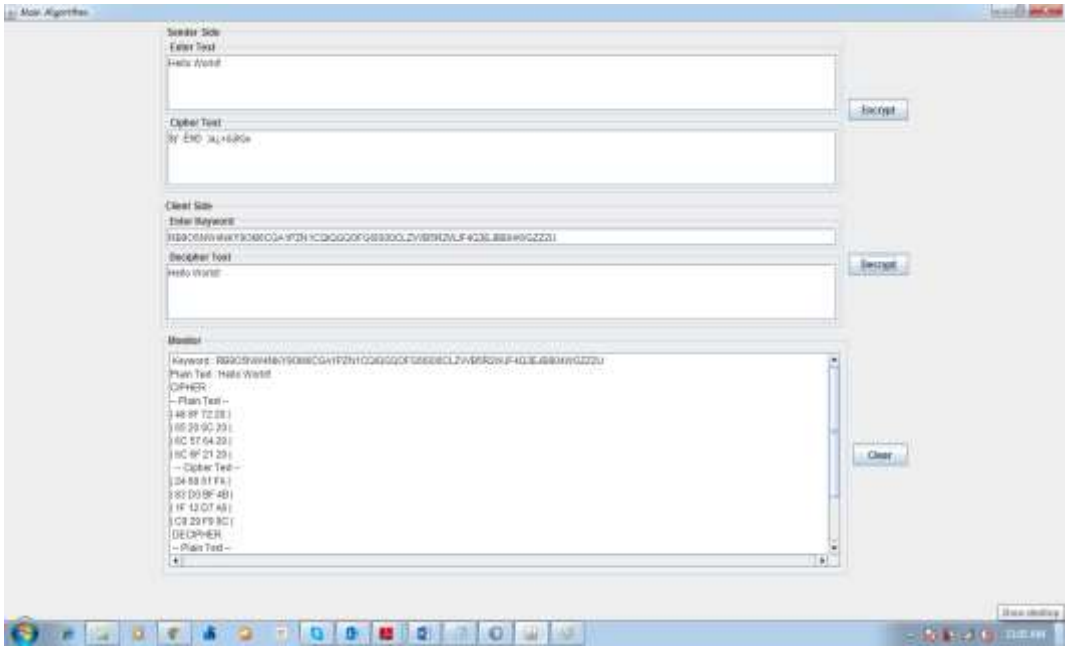

Fig. 5 On providing the correct key, we get the deciphered text in the Decipher Text Panel along with State Matrix displayed in the Monitor

6. CONCLUSION

The proposed algorithm is quit complex and much secure to compare with the AES algorithm or any other consequent algorithms in use today and the encoding time is mostly the same as the industry standard algorithm used *i.e.*, AES.The proposed algorithm although it ensures increased information security for any data that is used or transmitted on a cloud computing based platform poses certain limitations at present which can be made a priority so as to reinforce its standing as an industry standard level algorithm.

REFERENCES

[1] B. Schneier, “Applied cryptography”, second edition, NY: John Wiley & Sons, Inc., (1996).
[2] J. G. Proakis, D. G. Manolakis, “Introduction to Digital Signal Processing”, MacMillan Publishing Company, (1988).
[3] L. Scripcariu, S. Ciornei, “Improving the Encryption Algorithms Using Multidimensional Data Structures”, Proceedings of the Third European Conference on the Use of Modern Information and Communication Technologies, ECUMICT 2008, Gent (Belgium), March (2008): 375-384.
[4] http://phys.org/news/2011-08-world-toughest-encryption-scheme-vulnerable.html.
[5] www.tutorialpoints.com.
[6] B. D. Hahn, D. T. Valentine, “Essential MATLAB for Engineers and Scientists, 4e”, Academic Press, (2010).
[7] L. Scripcariu, A. Alistar, M. D. Frunza, “JAVA Implemented Encryption Algorithm”, Proceedings of the 8th Int. Conference "Development and Application Systems", Suceava, DAS 2006, (2006) May: 424-429.
[8] U.S. Department of Commerce/NIST, ―Data Encryption Standard,‖ FIPS PUB 46-3, (1999) October: 1-26.
[9] NIST, ―Advanced Encryption Standard,‖ FIPS PUB 197, (2001) November: 1-51.
[10] J. Daemen and V. Rijmen, “The Design of Rijndael: AES - The Advanced Encryption Standard”, Springer-Verlag, Berlin Heidelberg, (2002).
[11] H. Gilbert and M. Minier, “A collision attack on seven rounds of Rijndael”, Proceedings of the 3rd AES Candidate Conference, (2000) April: 230-241.
[12] B. Schneier, “The GOST Encryption Algorithm”, Dr. Dobb’s Journal, v.20, n. 1, (1995) January: 123-124.
[13] D. J. Wheeler and R. Needham, “TEA, A Tiny Encryption Algorithm”, Technical Report 355, “Two Cryptographic Notes,” Computer Laboratory, University of Cambridge, (1994) December: 1-3.
[14] B. Yee, “Using Secure Coprocessor”, Ph. D. Dissertation, School of Computer Science, Carnegie University, (1994) May.

