**Bulletin of Electrical Engineering and Informatics**
**ISSN: 2302-9285**
Vol. 6, No. 3, September 2017, pp. 219~227, DOI: 10.11591/eei.v6i3.627 

# Chaotic Rivest-Shamir-Adlerman Algorithm with Data Encryption Standard Scheduling

**Edwin R. Arboleda*[1], Joel L. Balaba[2], John Carlo L. Espineli[3]**
Department of Computer and Electronics Engineering, Cavite State University, Cavite, Philippines
*Corresponding author, e-mail: edwin.r.arboleda@cvsu.edu.ph[1], jbalaba26@gmail.com[2], johncarloespineli@gmail.com[3]

***Abstract***

*Cryptography, which involves the use of a cipher, describes a process of encrypting information so that its meaning is hidden and thus, secured from those who do not know how to decrypt the information. Cryptography algorithms come with the various types including the symmetric key algorithms and asymmetric key algorithms. In this paper, the authors applied the most commonly used algorithm, which is the RSA algorithm together with the Chaos system and the basic security device employed in the worldwide organizations which is the Data Encryption Standard (DES) with the objective to make a hybrid data encryption. The advantage of a chaos system which is its unpredictability through the use of multiple keys and the secrecy of the RSA which is based on integer factorization's difficulty is combined for a more secure and reliable cryptography. The key generation was made more secure by applying the DES schedule to change the keys for encryption. The main strength of the proposed system is the chaotic variable key generator that chages the value of encrypted message whenever a different number of key is used. Using the provided examples the strength of security of the proposed system was tested and demonstrated.*

***Keywords****: cryptography, encryption algorithm, chaos, RSA algorithm, Data Encryption Standard (DES)*

## 1. Introduction

The most important concern in the current digital technology is the privacy and security of information. It is also one of the principal challenges of resource sharing on data communication network. Data security becomes critical especially when resources are being shared between computers.

Chaos system has been gaining popularity in the world of cryptography for its sensitivity to initial condition and its capability to generate apparently unpredictable behavior which make the system more complex and secure [1]. Chaotic systems have properties that created huge impact, such as the sensitive dependence on initial conditions and system parameters, pseudorandom property, no periodicity and topological transitivity, etc. Most properties meet some requirements such as diffusion and mixing in the sense of cryptography making chaotic cryptosystems to have more useful and practical applications [2–5]. Chaos functions aids in the development of mathematical models for nonlinear systems. The deterministic property of the chaos maps can make the system able to get the same set of values given there is exactly the same mapping function and initial value. But when there is a slight difference from the used initial value, it will cause a huge difference in the cipher-text produced using chaos. This makes the system to be secure from brute force attacks and it is difficult to make long term predictions on chaotic systems. Due to the properties of chaos system, recent researches of encryption algorithms have been increasingly based on chaotic systems [6–9].

RSA algorithm with a specific message block size allows a message sender to generate a public keys to encrypt the message and the receiver is sent a generated private key using a secured database. An incorrect private key will still decrypt the encrypted message but in a form different from the original message [10]. RSA is named after its inventors Ron Rivest, Adi Shamir and Leonard Adleman at MIT who first proposed a description of the algorithm publically in 1977. It is a form of asymmetric cryptography. A user of RSA creates and then publishes a public key based on the two large prime numbers, along with an auxiliary value. The prime

*Received July 22, 2017; Revised August 17, 2017; Accepted August 30, 2017*

numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime numbers can feasibly decode the message [11]. RSA have many flaws in its design therefore not preferred for the commercial use. When the small values of the two large prime numbers, p & q are selected for the designing of key the encryption process becomes too weak and one can be able to decrypt the data by using random probability theory and side channel attacks. On the other hand if large p & q lengths are selected then it consumes more time and the performance gets degraded in comparison with DES. Further, the algorithm also requires of similar lengths for p & q, practically this is very tough conditions to satisfy. Padding techniques are required in such cases increases the system's overheads by taking more processing time [12]. RSA cryptography is a widely used encryption algorithm and is used in many different applications. [13–15]. Over the years a number of researches have enhanced the RSA or combined it with other algorithms to improve its security features [16–19].

The Data Encryption Standard (DES) has been the most extensively used encryption algorithm in recent times. The strength of DES is that decryption process can only be done by knowing the customized keys used in the encryption process. It uses the subkey rotation table by shifting number of bits indicated by the subkey index. It is the most widely used private key block cipher. It was first adopted in the year 1977 by the National Bureau of Standards as Federal Information Processing Standard 46 (FIPS PUB 46) [20-21].

In this paper, the authors proposed a hybrid cryptographic technique using RSA, Chaos System and DES scheduling concepts to provide more secure and reliable encryption algorithm. The shifting schedule of the DES gives an added complexity to the combined Chaos-based and RSA cryptosystem. The advantage of a Chaos system which is its unpredictability through the use of multiple keys and the secrecy of the RSA which is based on integer factorization's difficulty were combined for a more secure and reliable cryptography.

## 2. Research Method

### 2.1. RSA, Chaos, and DES Scheduling Cryptosystems

a. RSA Cryptosystem

Rievst-Shamir-Adlerman (RSA) Algorithm, created and developed by Ron Rivest, Adi Shamir and Leonard Adlerman at MIT in 1978, is the most widely used public key cryptosystem. It may be used to provide both secrecy and digital signatures and its security is based on the difficulty of the integer factorization. The RSA algorithm involves three steps: key generation, encryption and decryption [22].

- Key Generation

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated through the following ways:

1) To generate the two keys, choose two random large prime numbers (p) and (q). For maximum security, choose p and q of equal length. Compute the product n=p*q
2) Then randomly choose the encryption key e such that e and (p−1) (q−1) are relatively prime.
3) Finally, use the extended Euclidean algorithm to compute the decryption key (d) such that d=e-1 mod ((p-1)*(q-1)). Note that d and n are also relatively prime.
4) The numbers e and n are the public key; the number d is the private key. The two prime numbers p and q are no longer needed. They should be discarded, but never revealed.

- Encryption

At first the receiver transmits her public key (n, e) to sender and keeps the private key secret. If sender wishes to send message M to receiver, sender changes the message M in to integer m, such that $0 \leq m<n$. Then sender computes the cipher text c corresponding to

$$c=m^e \bmod(n) \quad (1)$$

- •Decryption:

Receiver can recover m from c by using her private key exponent d via computing

$$m=c^d \bmod(n) \quad (2)$$

### 2.2. Secure and Fast Chaos Cryptosystem

The chaos algorithm of this paper is based on the Secure and Fast Chaos [23]. The following are the steps involved in the key generation, encryption and decryption of the Secure and Fast Chaos:

Pi=Plain text
Ci=Cipher text

- **Key generation**

a) Choose the values of parameter (M, A, Xn).
b) Use the following equation to generate the pseudo random numbers.

For n=1 to j

$$X_{n+1=}\{A * X_n(X_n-1)\} \bmod (256) \quad (3)$$

where; A=any integer (1, 2, 3,.....)
Xn=initial value of chaotic function which is 2, 3, 4,….., n.
j=Number of keys
$X_{n+1}$=keys K1, K2, K3,.....Kj (after applying gray code on Xn+1 )

c) Apply gray code on these random numbers which are generated from X n+1 to build up the keys K1, K2 , K3 ,.....Kj .
d) Keys are represented in 8 bit binary form.

- **Encryption**

a) Each character is shown in ASCII character, Pi=ASCII(character i).
b) ASCII character Pi is converted into 8 bit binary form.
c) Using the equation Ek m (Pi ) C i for all i>0, and m=1 to j for encryption, Where Ekm (Pi) is bit wise XORing on plaintext Pi with single key Km.
d) Find the 1`s complement of ciphertext.

- **Decryption**

a) Find the 1`s complement of receiving ciphertext (Ci).
b) Using the equation Pi Dkm (Ci). Where m=1 to j for decryption, where Dkm (Ci) is bit wise XORing on cipher text Ci with single key Km.
c) Plain text Pi is converted into ASCII (Pi) with respect to its decimal value.
d) Then Character i=ASCII (Pi).

### 2.3. DES Schedule for Keyshift

DES uses the subkey rotation table to split and shift number of bits as indicated by the subkey index [20] shown in Table 1.

Table 1. DES Rotation Table

| **Round number** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| ***Rotate by*** | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| **Round number** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| ***Rotate by*** | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

## 3. Results and Analysis

### 3.1. Proposed Hybrid Algorithm

The proposed method is the combination of DES Schedule, Chaos and RSA cryptosystem. The shifting schedule of the DES gives an added complexity to the combined chaos-based and RSA cryptosystem. The advantage of a chaos system which is its unpredictability through the use of multiple keys and the secrecy of the RSA which is based on integer factorization's difficulty is combined for a more secure and reliable cryptography. Figure 1 shows the block diagram of the proposed algorithm.

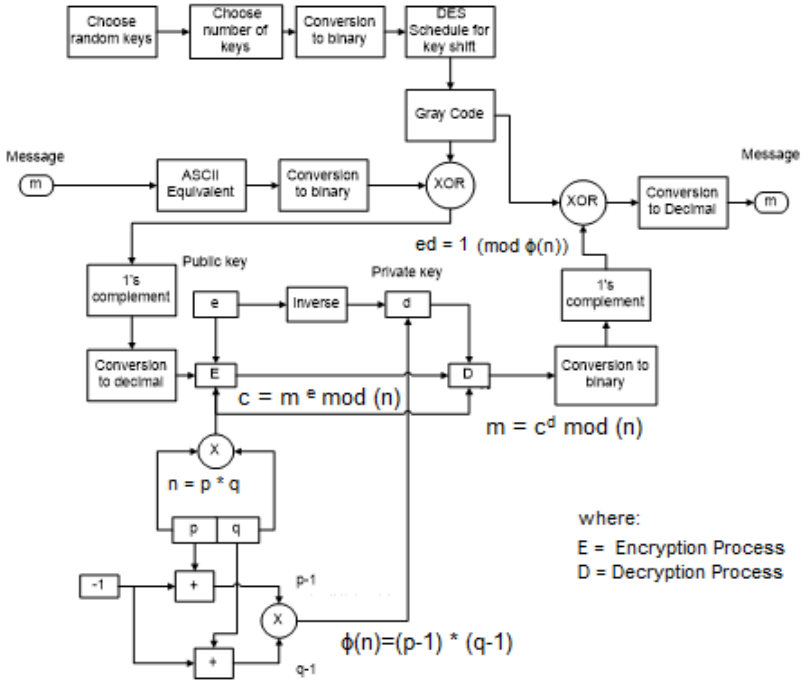


Figure 1. Block diagram of the proposed algorithm

Figure 1. is explained in the following processes:

- Key Generation

*Steps:*

1. Choose random number keys
2. Choose number of keys
3. Convert to binary
4. Use DES scheduling for key shift
5. Get the Gray code (this will be used to encrypt the message before RSA Encryption)
   Using the RSA Key Generation:
6. Choose p (prime number greater than any number in the message; p>m)
7. Choose q (any prime number)
8. Get n=p * q
9. Get ϕ(n)=(p-1)*(q-1)
10. Choose value for e=(any prime number with no GCD with ϕ(n))
11. Get the value of d using Euclidean Algorithm

- Encryption

*Steps:*

Using Chaos encryption:

1. Get the decimal equivalent of ASCII
2. Convert decimal to binary
3. XOR message with corresponding g1 to g6
4. Get the 1's complement
5. Convert to decimal (this will be used for RSA encryption)
6. Get the cipher text by RSA Encryption using the formula:

$$c=m^e \bmod(n)$$

- Decryption

RSA Decryption:

1. Get the value of m using the formula:

$m=c^d \bmod(n)$

2. Get the original messages using Chaos decryption:
3. Convert to decimal Get the decimal equivalent of ASCII
4. Get the 1's complement
5. XOR with corresponding g1 to g6
6. Convert binary to decimal
7. Get the ASCII equivalent

### 3.2. Example of the Proposed Hybrid Algorithm

Table 2, 3 and 4 shows the key generation, encryption and decryption of the proposed algorithm using the message below.

ASCII Equivalence: H=72, E=69, L=76, L=76, O=79

Table 2. Key Generation

| STEP | PROCESS | RESULT |
|---|---|---|
| 1 | Choose random keys | k1=12, k2=14, k3=16, k4=18, k5=20 |
| 2 | Choose number of keys | 5 keys |
| 3 | Conversion to binary | 12=0000 1100<br>14=0000 1110<br>16=0001 0000<br>18=0001 0010<br>20=0001 0100 |
| 4 | DES Schedule for key shift | 0001 1000<br>0001 1100<br>0100 0000<br>0100 1000<br>0101 0000 |
| 5 | Gray code | 0001 0100<br>0001 0010<br>0110 0000<br>0110 1100<br>0111 1000 |
| 6 | Choose p (prime number greater than any number in the message | p=127 |
| 7 | Choose q (any prime number) | q=7 |
| 8 | Get n=p*q | n=127 *7=889 |
| 9 | Get $\phi(n)=(p-1)*(q-1)$ | $\phi(n)=126*6=756$ |
| 10 | Choose value for e=(any prime number with no GCD with $\phi(n)$) | e=11 |
| 11 | Get the value of d using Euclidean Algorithm | Step 1: Euclidean Algorithm<br>$e*d=1 \pmod{\phi(n)}$<br><br>756=68(11)+8<br>11=1(8)+3<br>8=2(3)+2<br>3=1(2)+1<br><br>Step 2: Extended Euclidean Algorithm (Back Substitution)<br>1=3–1(2)<br>1=3–1[8–2(3)]<br>1=3–1(8)+2(3)<br>1=3(3)–1(8)<br>1=3[11-1(8)]–1(8)<br>1=3(11)–3 (8)–1(8)<br>1=3(11)–4 (8)<br>1=3[11]–4(756-68(11))<br>1=3(11)–4(756)+272 (11)<br>1=275(11)–4(756)<br>Use d=275 since it is already a positive number<br>d=275 |

Note: The complete instruction on how to get d can be found in [24]

Table 3. Encryption

| STEP | PROCESS | RESULT | |
|---|---|---|---|
| | | Message | ASCII |
| 1 | Convert message into ASCII equivalent | H | 72 |
| | | E | 69 |
| | | L | 76 |
| | | L | 76 |
| | | O | 79 |
| 2 | Convert ASCII equivalent into Binary | 01001000 | |
| | | 01000101 | |
| | | 01001100 | |
| | | 01001100 | |
| | | 01001111 | |
| 3 | Message in binary ⊕ Gray code ($g_n$) | 01011100 | |
| | | 01010111 | |
| | | 00101100 | |
| | | 00100000 | |
| | | 00110111 | |
| 4 | Get the 1's complement | 10100011 | |
| | | 10101000 | |
| | | 11010011 | |
| | | 11011111 | |
| | | 11001000 | |
| 5 | Convert to decimal (this will be used for RSA encryption) | 163 | |
| | | 168 | |
| | | 211 | |
| | | 223 | |
| | | 200 | |
| 6 | Get the cipher text by RSA Encryption using the formula: $c=m^e \bmod (n)$ | 648 | |
| | | 875 | |
| | | 15 | |
| | | 118 | |
| | | 856 | |

Table 4. Decryption

| STEP | PROCESS | RESULT |
|---|---|---|
| 1 | Get the value of m using the formula: $m=c^d \bmod (n)$ | 163 |
| | | 168 |
| | | 211 |
| | | 223 |
| | | 200 |
| 2 | Convert to decimal Get the decimal equivalent of ASCII | 10100011 |
| | | 10101000 |
| | | 11010011 |
| | | 11011111 |
| | | 11001000 |
| 3 | Get the 1's complement | 01011100 |
| | | 01010111 |
| | | 00101100 |
| | | 00100000 |
| | | 00110111 |
| 4 | XOR with corresponding g1 to g6 | 01001000 |
| | | 01000101 |
| | | 01001100 |
| | | 01001100 |
| | | 01001111 |
| 5 | Convert binary to decimal | ASCII |
| | | 72 |
| | | 69 |
| | | 76 |
| | | 76 |
| | | 79 |
| 6 | Get the ASCII equivalent | Message ASCII |
| | | H 72 |
| | | E 69 |
| | | L 76 |
| | | L 76 |
| | | O 79 |

### 3.3 Analysis of the Proposed Method

#### 3.3.1. Sensitivity to Number of Keys

It is claimed by the proposed method that if number of keys are changed, then the cipher texts completely changed from one another for same plain text.

Table 5. Sensitivity of Number of Keys of the Proposed System

| Message | No. of keys | Public keys e=11 n=889 | Private keys n=889 | Ciphertext |
|---|---|---|---|---|
| HELLO | 1 | k=12<br>g1=0001 0100 | d=275 | 648,419,839, 839,607 |
| HELLO | 2 | k=12,14<br>g1=0001 0100<br>g2=0001 0010 | d=275 | 648,875,232, 821,181 |
| HELLO | 3 | k=12,14,16<br>g1=0001 0100<br>g2=0001 0010<br>g3=0001 0000 | d=275 | 648,875,15, 232,71 |
| HELLO | 4 | k=12,14,16,18<br>g1=0001 0100<br>g2=0001 0010<br>g3=0001 0000<br>g4=0001 0010 | d=275 | 648, 875, 15, 118,122 |
| HELLO | 5 | k=12,14,16,18,20<br>g1=0001 0100<br>g2=0001 0010<br>g3=0001 0000<br>g4=0001 0010<br>g5=0001 0100 | d=275 | 648,875,15, 118, 856 |

Table 5 shows the effect of using different keys in the generation of cipher text. It can be observed that a change in the number of keys would also result to a change in the generated cipher text. In the message “HELLO”, when only one key is used, the letter “L” has the same cipher text. But when using two or more keys, the message “L” generates different cipher text which makes it more advantageous and difficult to be deciphered by the attackers because it will be a mystery on how many keys did the sender used.

#### 3.3.2. Security Analysis

Changing one bit in the plaintext or one bit in the key changes the bits in the cipher text [25]. Using two different keys also produces different cipher text given the same data information [26]. This makes the system harder to decipher when trying to come up with an attack. This important characteristic of an encryption algorithm is called the avalanche effect [27].

Multiple keys k and one private key d were used in table 4 to encrypt the message. Another implementation of the proposed method is use not only one random number k but multiple k’s for encryption.

Table 6. The Avalanche Effect of the Proposed System using One K and Multiple K

| Message | Parameters | Random number | Cipher text |
|---|---|---|---|
| blowzy night-frumps vex’d jack q | p=127<br>q=7<br>n=889<br>e=11<br>d=275 | k1=12 | 737,32,487,27,815,419,<br>11,12,744,883,389,648<br>633,809,569,101,624,232,<br>191,11,35,40,182,557,131<br>11,525,705,829,76,11,518 |
| blowzy night-frumps vex’d jack q | p=127<br>q=7<br>n=889<br>e=11<br>d=275 | k1=12<br>k2=14<br>k3=16<br>k4=18<br>k5=20 | 737,340,557,212,253, 419<br>272,299,27,693,389,870<br>40,299,819,9,382,389,693<br>839,35,246,84,829,839,11<br>809,110,210,150,11,858 |

In Table 6, the proposed system was implemented in two different ways. The same message was encrypted using same prime number, public keys and private key. The only difference is in the number of k and values of k used. It can be observed that the encrypted essage greatly differs from one another. In Tables 5 and 6, the avalanche effect was demonstrated by the proposed method on the number and values of random numbers used.

## 4. Conclusion

The proposed system was able to combine the secure and fast chaos cryptosystem, DES scheduling and the RSA cryptosystem. It is a combination of the multiple keys of chaos system and the strength of the RSA algorithm by having multiple keys as in the concept of the chaos cryptosystem in able to have different cipher text for encryption. The key generation was made more secure by applying the DES schedule to change the keys for encryption. The proposed method provided for the RSA cryptosystem's way of manipulating random numbers used to create an avalanche effect in the produced cipher text. The chaotic properties of the proposed system lie in the number of random numbers used. The system successfully proved the important characteristics of the proposed hybrid system in the encryption of data in a more secure way.

## References

[1] Agrawal B, Agrawal H. Implementation of AES and RSA Using Chaos System. *Int J Sci Eng Res [Internet].* 2013; 4(5): 1413–7. Available from: http://www.ijser.org
[2] Fadhel S, Shafry M, Farook O. Chaos Image Encryption Methods : A Survey Study. *Bulletin of Electrical Engineering and Informatics (BEEI).* 2017; 6(1): 99–104.
[3] Xiao G, Liu H, Guo Y, Zhou Y. Research on Chaotic Firefly Algorithm and the Application in Optimal Reactive Power Dispatch. *TELKOMNIKA (Telecommunication Computing Electron and Control)*. 2017; 15(1): 93-100.
[4] Zhao C, Wang G. Application of Chaotic Particle Swarm Optimization in Wavelet Neural Network. *TELKOMNIKA (Telecommunication Computing Electron and Control]).* 2014; 12(4): 997-1004. Available from: http://journal.uad.ac.id/index.php/TELKOMNIKA/article/view/533
[5] Gholipour, Yousof ; Ramezani, Amin; Mola M. Illustrate the Butterfly Effect on the Chaos Rikitake system. *Bulletin of Electrical Engineering and Informatics (BEEI)*. 2014; 3(4):273–6.
[6] Suryadi M, Nurpeti E, Widya D. Performance of Chaos-Based Encryption Algorithm for Digital Image. *TELKOMNIKA (Telecommunication Computing Electron and Control)*. 2014; 12(3): 675–82.
[7] Kumar RR, Sampath A, Indumathi P. Secure Optical Communication Using Chaos. *Indian J Sci Technol*. 2011; 4(7): 773–8.
[8] Thamizhchelvy K, Geetha G. Data Hiding Technique with Fractal Image Generation Method using Chaos Theory and Watermarking. *Indian J Sci Technol*. 2014; 7(September):1271–8.
[9] Qian Y, Liu LL. The Analysis and Application on a Fractional-Order Chaotic System. *TELKOMNIKA (Telecommunication Computing Electron and Control)*. 2014; 12(1): 23-32. Available from: http://www.journal.uad.ac.id/index.php/TELKOMNIKA/article/view/8
[10] Goshwe NY. Data Encryption and Decryption Using RSA Algorithm in a Network Environment. *IJCSNS Int J Comput Sci Netw Secur*. 2013; 13(7): 9–13.
[11] Shankar M. Hybrid Cryptographic Technique Using RSA. *Int J Netw Secur Its Appl*. 2014; 6(6): 39–48.
[12] Singh G. A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. *Int J Comput Appl*. 2013; 67(19): 975–8887. Available from: http://research.ijcaonline.org/volume67/number19/pxc3887224.pdf
[13] Chadha A. Dual-Layer Video Encryption using RSA Algorithm Dual-Layer Video Encryption using RSA Algorithm. *Int J Adv Res Ideas Innov Technol.* 2015; 116(August): 33–40.
[14] Karakra A, Alsadeh A. *A-RSA: Augmented RSA.* Proc 2016 SAI Comput Conf SAI 2016. 2016; (September): 1016–23.
[15] Tan X, Li Y. *Parallel Analysis of an Improved RSA Algorithm*. Proc-2012 Int Conf Comput Sci Electron Eng ICCSEE 2012. 2012; 1: 318–20.
[16] Espalmado JMB, Arboleda ER. DARE Algorithm : A New Security Protocol by Integration of Different Cryptographic Techniques. *Int J Elecrical Comput Eng*. 2017; 7(2): 1032–41.
[17] Ahmed JM, Ali ZM. *The Enhancement of Computation Technique by Combining RSA and El-Gamal Cryptosystems.* Proc 2011 Int Conf Electr Eng Informatics, ICEEI 2011. 2011; (July).
[18] Al-Hamami AH, Aldariseh IA. *Enhanced method for RSA Cryptosystem Algorithm*. Proc-2012 Int Conf Adv Comput Sci Appl Technol ACSAT 2012. 2013; 402–8.

[19] Sharma S, Sharma P, Dhakar RS. *RSA Algorithm Using Modified Subset Sum Cryptosystem*. 2011 2nd Int Conf Comput Commun Technol ICCCT-2011. 2011; 457–61.
[20] G. Sai Venkat PSRT. VLSI Implementation of DES and Triple DES Algorithm with Cipher Block Chaining Approach*. Int J Sci Eng Technol Res*. 2014; 3(45): 9207–10.
[21] Rivest RL, Shamir A, Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun ACM [Internet]. 1978; 21(2): 120–6. Available from: http://portal.acm.org/citation.cfm?doid=359340.359342
[22] A. Khare A, B. Shukla P, C. Silakari S. *Secure and Fast Chaos based Encryption System using Digital Logic Circuit.* Int J Comput Netw Inf Secur [Internet]. 2014; 6(6): 25–33. Available from: http://www.mecs-press.org/ijcnis/ijcnis-v6-n6/v6n6-4.html
[23] Kuchhal S. Security Design of TDES Using Low Power VLSI Implementation. *Paripex-Indian J Res*. 2013; 97–100.
[24] Rhee MY. Internet Security: Cryptographic Principles, Algorithms, and Protocols. Chichester, West Sussex, England ; Hoboken, NJ: J. Wiley; 2003. 172-175 p.
[25] Mandal A, Tiwari M. Analysis of Avalanche Effect in Plaintext of DES using Binary Codes. I*nt J Emerg Trends Technol Comput Sci.* 2012; 1(3): 166–77. Available from: http://www.ijettcs.org/Volume1Issue3/IJETTCS-2012-10-25-097.pdf
[26] Enriquez M, Garcia DW, Arboleda E. Enhanced Hybrid Algorithm of Secure and Fast Chaos-based, AES, RSA and ElGamal Cryptosystems. *Indian J Sci Technol*. 2017; 10(July).
[27] Kumar A, Tiwari N. Effective Implementation and Avalanche Effect of AES. *Int J Secur Priv Trust Manag (IJSPTM)*. 2012; 1(3): 31–5.