International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-1 Issue-8 March-2013

DEDD Symmetric-Key Cryptosystem

Praloy Shankar De[1], Prasenjit Maiti[2]
School of Computer Engineering, KIIT University, Bhubaneswar, Odisha, India[1]
School of Information Technology, IIT, Kharagpur, West Bengal, India[2]

Abstract

Now a days the greatest challenge to the cyber world is security. We need more and more security for our unsecured systems because technology is not a constant factor. There are lots of catalysts, can break the security system. Day by day number of hackers is being increased. Existing old methods of security often fails to overcome it. But as an ancient quote says it's true that "Old is Gold", those methods help us to develop a new idea to sort out our problem in situ thus leads us to discover a new horizon. In this paper an attempt has been made to focus on an algorithm of cryptography that was made by using old methodologies. DEDD symmetric-key cryptosystem is the new approach to symmetric key algorithm. By this method we can doubly encrypt and doubly decrypt the message. It means the sender will generate the cipher text from the plain text twice. The receiver will also have to decrypt the ciphers for two times and then the communication between them will be completed. For generating the key, we will take the message length in first encryption and in second encryption we will apply shifting technique on a method integrate_expo which calculates the value of e^integration (m^k dm) from limit m=0 to limit m=m1(message length) where e=exponential, m=variable. Double de-cryption will be done by the reverse process of encryption techniques. There are lots of algorithms for symmetric key cryptography like DES, AES, BLOWFISH, and HMAC-MD5 etc.

Keywords

Double Encryption and Decryption, Symmetric-key cryptosystem, Cipher, Key generation.

1. Introduction

Cryptography, the art and science of preparing coded or protected communications is intended to be intelligible only to the person possessing a key. Using the key concept we know that there are two types of cryptography, Symmetric key and Asymmetric key. For symmetric case only one key will be used which is called secret key and for better security we will apply asymmetric key cryptography where two key will be used. One is called public key and another one is private key. To send a secured message to Bob, Alice first encrypts the message or plaintext in a cipher text using Bob's public key. For decryption Bob will use the private key. But can we use symmetric key concept for better security? So we are going for a symmetric key cryptography system with a new technique, 'DEDD symmetric-key cryptography'.

2. DEDD Symmetric-Key Cryptography

DEDD means Double Encryption and Double Decryption. In this cryptosystem, Alice encrypts the message twice with the public key, and Bob will decrypt that encrypted message twice. The procedure is as follows:
Bob generates a key and assigns it to Alice. Alice enciphers the message by applying "shift Cipher"[11,12] and encrypt the Message by the length of it and get cipher1(for 1st time). Second time encryption will be done by applying shifting technique on a method Integrate_Expo which calculates a value of e^integration (m^k dm) from Limit m=0 to Limit m=m1 (msg_length) i.e. value of equation(1) and this will be sent to Bob. Then Bob will decrypt this encoded message by applying shifting technique on the method differentiation_Expo which will calculate a value of equation(2).

Equations
The equations, used for second time encryption and first time decryption are as follows:

$$e^{\int_0^{m1} m^k \, dm} \qquad (1)$$

where e = exponential function, m=variable, k = key used in the system, m1 = message and cipher2 will be generated.

$$e^{\frac{d}{dm}\left(\frac{m^{k+2}}{(k+1)(k+2)}\right)} \quad (2)$$

Where d/dm=differentiation symbol. Here decipher1 will be generated. Finally by using shifting technique over the generated decipher1, original message will be extracted.

2.1 Proposed Methodology

In this section let us explain our proposed algorithm. Here a plain text is taken as an input. Encryption and decryption will be done twice on it. Some user defined functions will also be applied at the time of encryption and decryption. During second time of decryption the entered text will be obtained as an output.

Algorithm : DEDD

Input:
1. A set of different characters in a queue Q .
2. The input is a string S, it can be combination of all characters{A-Z, a-z},numbers{0-9}and special characters{:,!,@,#,etc}.

Output: The output is S', where this S' is extracted from twice encryption and twice decryption of S. Therefore S and S' both are same.

Pseudo code:

```
Algorithm DEDD
{
Step 1: The input string is provided by Alice.
Step2 : The key will be generated from the method keygen().
Step3: The string will be encrypted to cipher1 using the method Encrypt1().
Step4: The cipher1 will be given as input to the method Encrypt2() which encrypt the cipher1 to cipher2. This cipher2 will be sent to Bob.
Step5 : Bob uses the method Decrypt1() to decrypt cipher2 and extract cipher1.
Step6: Bob uses another method Decrypt2() to extract original message from cipher1.
}
Algorithm Keygen()
{
Step1 : It takes message length as input (mL).
Step2: Calculates the exponential of mL and returns the value.
}
Algorithm Encrypt1()
{
Step1: The input string is the input of the function.
Step2: If S=Q Then
Step3: S1[i] =Q[i+l], where i=index of S, l=length of input string.
Step4: The new S1[i] is the cipher1 and returned.
}
Algorithm Encrypt2(){
Step1: The new S1[i] and key are the inputs.
Step2: Variable IE=Integ_exp(0,key,mL).
Step3: If S1=Q Then S2[i] =Q [i+IE] ,
where i=index of S1.And this will be cipher2.
Step4: Return S2[i].
}
Algorithm Decrypt2(){
Step1: Here S2[i] is the input string.
Step2: Variable DE=Diff_exp(key, mL).
Step3: If S2=Q Then S3[i]=Q[i-DE].
Step4: Return S3[i].
}
Algorithm Decrypt1(){
Step1: Here S3[i] is the input.
Step2: If S3=Q Then S4[i]=Q[i-l],where i=index of S4, l=length of S3.
Step3: Return S4[i]. /*S4 is the original message*/
}

Algorithm Integ_exp(){
Step1: It takes mL and key as inputs and calculates the value of IE= $e^{\int_0^{m1} m^k \, dm}$
Step2: Returns IE.
}

Algorithm Diff_exp(){
Step1: It takes mL and key as inputs and calculates the value of DE= $e^{\frac{d}{dm}\left(\frac{m^{k+2}}{(k+1)(k+2)}\right)}$
Step2: Returns DE.
}
```

3. Implementation and Result

3.1 Environmental setting to implement the Proposed Methodology :

To implement the proposed algorithm we used a centrino machine with Windows Vista opearting system and the code has been written in turbo c++.

Message

P	R	A	L	O	Y	8	9	$

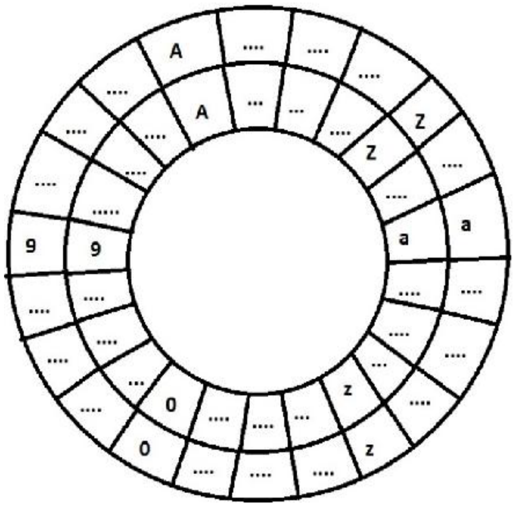


Figure 1: Representation of circular queue

1. The two concentric circles represent characters('A…….Z','a……z','0…..9', some special characters like (,),!,@,$,%etc.) (see the circular queue in [13]).
2. The first queue is used to get cipher1 and decipher1.Whereas, the second queue is used to get cipher2 and decipher2 by applying the methods.
3. After receiving the key Alice counts the length of the message and encrypts the message follows:

- ✓ Shift each character of the message according to the length of the message by shifting each character of the string found in the 1st queue up to the length .We get cipher1.
- ✓ Cipher1 is further encrypted to cipher2 by applying the function Prosat i.e. by shifting the queue to Prosat value in clock-wise direction. And we will get cipher2.

- ✓ Now this cipher2 is passed with the Prosat function through the channel.

4. On the Receiver's end decryption will be followed as:
- ✓ Now this decipher2 is changed into decipher1 with the help of the function Prosat by shifting the 2nd queue in anti clock-wise direction.
- ✓ Decipher1 is further decrypted to actual message by shifting each character of the decipher1 according to the length of the string found in the 1st queue up to the length.

To understand the algorithm we may take an example which is as follows:

If we think, we have to pass the message(msg) = "PRALOY89$", and we have a single circular queue(that will be used twice in encryption and also decryption). And supposes a key (anything you can take except it should not contain any letters like A..). We take the value of the key=5. Then by shift cipher we will get cipher1.

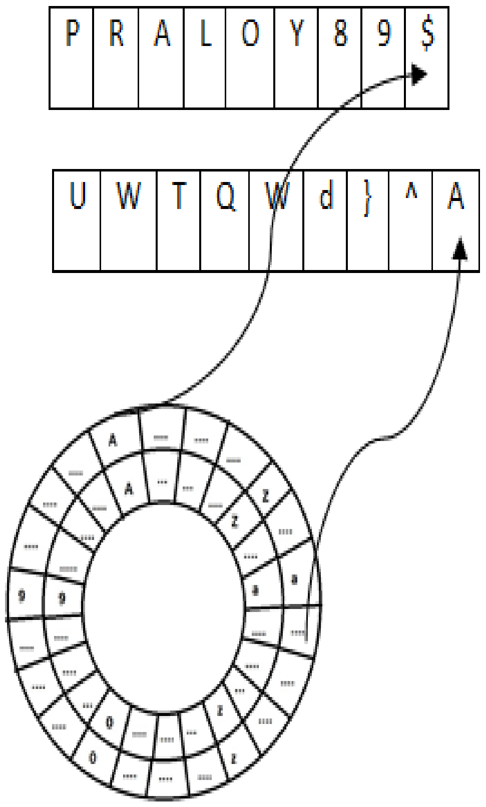


Figure 2: First encryption of plain text to Cipher1

Now, let key=63 and length of message(len)=9,the 1st queue shifts the position of 'P' to 63 more positions to get letter 'U',similarly 'R' changes to 'W' ,….'8' to 'd','9' to '}' and '$' to '^'.
This string is cipher1.

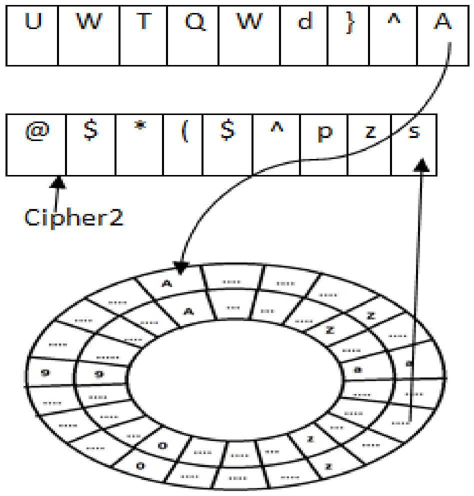


Figure 3: Cipher1 to Cipher2

When cipher2 is manipulated, a function Prosat is calculated as PROSAT=[(Key*Len)%queue size]. Here PROSAT can be calculated by Encrypt2 method and Decrypt2 method as defined in DEDD algorithm also. For an example we have taken this type of definition.
And 2nd Queue is shifted according to the value of that function in the clock-wise direction and cipher2 will be generated.

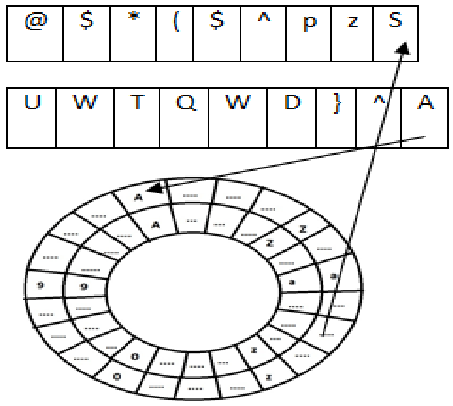


Figure 4: Cipher2 to Decipher1

Here also the above mentioned procedure is followed in the anti clock-wise direction and decipher1 is generated. Let the key is 5 then Prosat=[(5*9)%96](the total number of different characters are 96). So, Prosat=45. The circular arrow indicates the displacement of ‘s’ to 45 places in anti-clock-wise direction to get the letter “A’, and the procedure will give the string “UWQTQ}^A” as decipher1. In case of 2nd time decryption, the single queue will be shifted according to the length of the string. Therefore an actual message will be received.

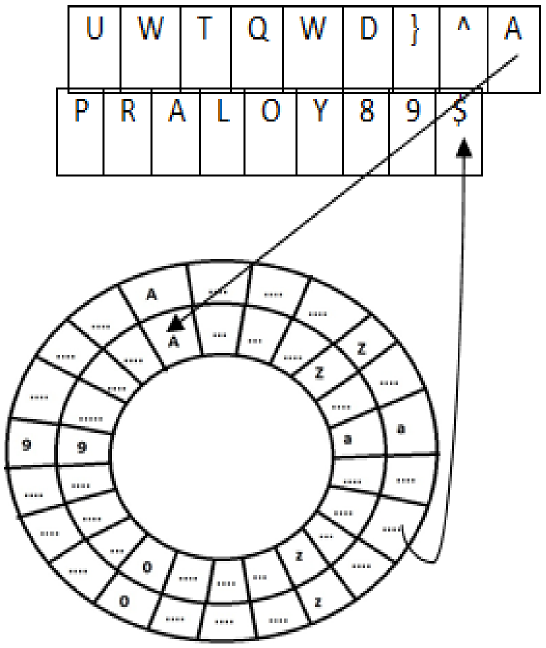
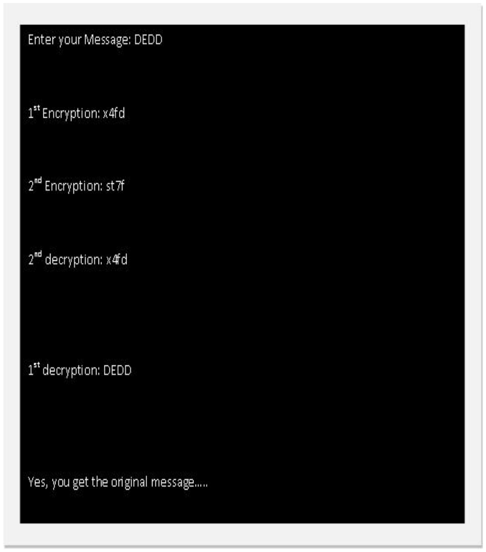


Figure 5: Decipher1 to actual message

3.2 Result of Proposed Methodology


```
Enter your Message: DEDD

1st Encryption: x4fd

2nd Encryption: st7f

2nd decryption: x4fd

1st decryption: DEDD

Yes, you get the original message.....
```

Figure 6: Output of DEDD

As a result we can see in fig.6. Alice is encrypting the message 'DEDD' for two times applying this methodology and Bob got the original message on its second time of decryption. In this paper we have shown our methodology with its work function but it has some limitations. The primary limitation is if the key size will be greater than 128 bit then it can generate erroneous value.

4. Conclusion and Future Work

This algorithm is meant for double time encryption and decryption. So, this is more secure than single encryption and decryption. As, we are using the single key with symmetric key cryptography, we can apply it for long type message. Cryptanalysis to this algorithm is little bit tough.
In future we can also develop this algorithm for Hybrid Cryptosystem by using more functions because we think that a methodology should be developed from secure to secured one.

Acknowledgment

We convey our special thanks to Mr. Satish Kumar Jaiswal for his help and suggestions about this technique. Satish Kumar Jaiswal is recently a TCS Employee.

References

[1] Behrouz A Frozen, "Data Communication & Networking", fourth edition, pp. 931-994, 2009.
[2] Bruce Schneier," Applied Cryptography" , second edition , Wiley-India, Aug 1, 2007 .
[3] A beginner’s guide to network security, CISCO Systems, found at http://www.cisco.com/warp/public/cc/so/neso/sqso/ beggu_pl.pdf , 2001.
[4] Eric Maiwald., " Networks Security A Beginner’s Guide", second edition , McGraw Hill Professional, May 29, 2003.
[5] Whitfield Diffie and Martin E . Hellman, "New Direction Cryptography", IEEE , Transactions On Information Theory, Vol. IT-22, No. 6, pp. 644-654, Nov.1976.
[6] Prashant Krishnamurthy, Joseph Kabara, Tanapat Anusas-amornkul, "Security in Wireless Residential Networks", Vol-48, pp. 157-166, Feb.2002.
[7] Ueli Maurer, "Information-Theoretic Cryptography (Extended Abstract)", CRYPTO '99, LNCS 1666, pp. 47–65, 1999. c Springer-Verlag Berlin Heidelberg 1999.
[8] Pratik A. Vanjara, "Analysis and Design of Cryptography Algorithms", International Journal of Computer Applications & Information Technology, Vol. I, Issue II, Sept.2012 (ISSN: 2278-7720).
[9] A. Nadeem, "A Performance Comparison of Data Encryption Algorithms", IEEE information and communication technologies, pp. 84-89, Aug.2005.
[10] A . Radhika, Parvathi Nair, M . Ramya, "A High Throughput Algorithm for Data Encryption", International Journal of Computer Applications (0975 – 8887), Vol. 13, No.5, Jan . 2011.
[11] http://www.math.cornell.edu/~mec/Summer2008/lundell/lecture1.html.
[12] http://www.codasaurus.com/Cipher-1.htm.
[13] Kruse Robert L., Robert Kruse, Cl Tondo , "Data Structures And Program Design In C", second edition, 4.3 circular queues in c , pp. 134.

Praloy Shankar De received the B.Tech Degree in Computer Science & Engineering from West Bengal University of Technology in 2011. He is currently a M.Tech Student in Computer Science & Engineering at KIIT University, Bhubaneswar, Odisha, India. His research interests include Network Security & Cryptography, Data Mining, Computational Intelligence.

Prasenjit Maiti is a M.Tech student in the Department of School of Information Technology, IIT Kharagpur, India. He received B.Tech degree in Computer Science & Engineering from West Bengal University Of Technology (2008). He is presently working as an Assistant Professor in the dept. of Computer Science & Engineering, Mallabhum Institute of Technology, Bishnupur, Bankura, West Bengal, India. His research interests include Cryptography, Hardware Security and VLSI Verification & Testing.