WAIMS© World Academy of Informatics and Management Sciences, Vol1 Issue 2, April-May 2012
ISSN (online):2278-1315
www.waims.co.in

ENHANCEMENT OF SECURITY ON E-COMMERCE BASED APPLICATION USING R-IDEA ALGORITHM WITH RANDOM KEY GENERATION APPROACH.

[1]Rahul Saxena, [2]Dr. C.S.Lamba

[1]Resarch Scholar, NIMS University Rajasthan Jaipur
[2] Research Supervisor, NIMS University, Jaipur

Abstract
A computer network is an interconnected group of computing nodes, using a well defined protocol to interact and share resource in predicate manner. Security over the network is widely required, recognized and accepted. Security plays a crucial and critical role in e-commerce and m-commerce channels. To address the security a number of algorithms have been proposed and many are being used in cryptography to secure our information. But security is a question till now. The purpose of this paper is to introduce and demonstrate a new algorithm for e-commerce based information security. The purposed algorithm is based on IDEA (International Data Encryption Algorithm), IDEA is a powerful algorithm, and no successful liner or algebraic weakness reported yet. Since 1996 till date there is no change in IDEA algorithm. The proposed paper has implement of randomized key generation, to enhance the security in the existing IDEA algorithm.

Keywords: - Randomized IDEA (R-IDEA), Diffie Hellman, International Data Encryption Algorithm (IDEA).

1. INTRODUCTION

With the popularity of the Internet, e-commerce has been developing very fast. On account of the success of ecommerce; international markets are now much more opportunity.
E-commerce business method is tremendous convenience and flexible. However, as is a web-based e-commerce, information and data security issues facing the attendant, such as internal theft and destruction, interception, unauthorized access, destruction of the integrity of the information and so many problems, so Construction of e-commerce security system becomes particularly important.

The purpose of this paper is to introduce and demonstrate a new algorithm for e-commerce information and data security. The purposed algorithm was developed based on the symmetric cryptography algorithm IDEA. In the new algorithm named R-Idea have random keys to enhance security. This paper organized as follow. Section 2 describes the Idea algorithm and its working and procedure. Section 3 discusses new developed algorithm and it's working. Section 4 describes implementation and results of developed algorithm. Finally, Section 5 concluded this paper.

2. ENHANCEMENT WITH OLD ONE

A. IDEA

IDEA (International Data Encryption Algorithm) is a cryptographic algorithm developed at ETH in Zurich, Switzerland. This algorithm uses a block cipher [1][2] with a 128-bit key to encrypt 64 bit data block, and is generally considered to be very secure. In this algorithm 52 sub-keys are generated from the 128-bit original key. Due to key strength [3] this algorithm is considered among the best publicly known algorithms. In the last several years no practical attacks on it have been published, even though a number of attempts to find some.

- Key size : 128 bits
- Plaintext Block size : 64 bits
- Rounds: 8.5

This algorithm provides high level security not based on keeping the algorithm a secret, but by keeping the key secret which makes it suitable for use in a wide range of applications worldwide.

B. ENCRYPTION

User input plaintext is divided into four [1][2] 16-bit sub-blocks: X1, X2, X3 and X4 (see fig. 1) and identical operations are performed on the four parts in 8

rounds. Sender's 128- bit key is break into eight 16-bit blocks, which become eight sub-keys. The first six sub-keys are used in round one, and the remaining two sub-keys are used in round two, similarly each round uses six 16-bit sub-keys for 8 rounds while the last half-round uses four, i.e. a total of 52 keys. First 6 keys are extracted directly from the main key. Further groups of keys are created by rotating the main key left by 25 bits.

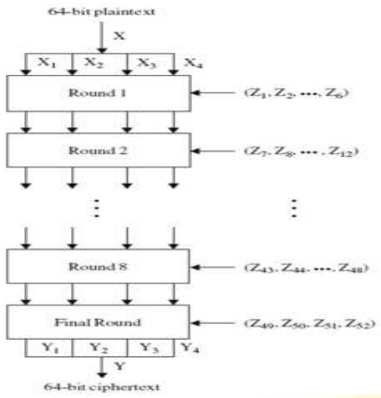


Fig. 1 dividing plain text

The mathematical operations involved in each of the rounds are:

- Bitwise Exclusive OR (denoted by).
- Addition modulo 216 (denoted by).
- Multiplication modulo 216+1 (denoted by).

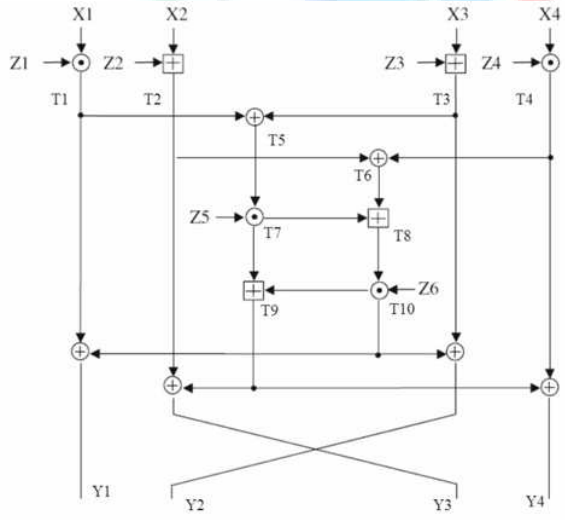


Fig. 2 First 8 rounds of IDEA

After the eight rounds output is cipher text: Y1, Y2, Y3 and Y4 (As illustrated in fig. 2):

C. Algorithm

Bruce Schneier [4] provided a fourteen-step algorithm of IDEA. Here are the fourteen steps of a complete round (multiply means multiplication modulo 216 + 1, and add means addition modulo 216):

1. Multiply X1 and the first subkey K1.
2. Add X2 and the second subkey K2.
3. Add X3 and the third subkey K3.
4. Multiply X4 and the fourth subkey K4.
5. Bitwise XOR the results of steps 1 and 3.
6. Bitwise XOR the results of steps 2 and 4.
7. Multiply the result of step 5 and the fifth subkey K5.
8. Add the results of steps 6 and 7.
9. Multiply the result of step 8 and the sixth subkey K6.
10. Add the results of steps 7 and 9.
11. Bitwise XOR the results of steps 1 and 9.
12. Bitwise XOR the results of steps 3 and 9.
13. Bitwise XOR the results of steps 2 and 10.
14. Bitwise XOR the results of steps 4 and 10.

For every round except the final transformation, a swap occurs, and the input to the next round is: result of step 11 k result of step 13 k result of step 12 k result of step 14, which becomes X1 k X2 k X3 k X4, the input for the next round. After round 8, a ninth "half round" final transformation occurs:

After the eighth round, there is a final output transformation:

(1) Multiply X1 and the first sub-key.

(2) Add X2 and the second sub-key.

(3) Add X3 and the third sub-key.

(4) Multiply X4 and the fourth sub-key.

Finally, the four sub-blocks are reattached to produce the cipher text.

D. Decryption

The computational process used for decryption of the cipher text is same as that used for encryption [1][2]. The only one difference is that each of the 52 16-bit key sub-blocks used for decryption is the inverse of the key sub-block used during encryption.

In addition, the key sub-blocks must be used in the reverse order during decryption in order to reverse the encryption process.

3 Related Work

IDEA is one of the world's most secure cryptographic algorithms but many researchers now consider it obsolete and feel a need to modify it. It has been emphasized that the modifications should be such that the algorithm remains efficient i.e. the time and space complexity should not increase too much, so increasing the rounds is not an intelligent approach as per Nick Hoffman [8]. Increasing the plaintext block size is also not feasible as per [9] because unlike 65,537 i.e.

2^16+1, 2^32+1 is not prime, so IDEA cannot be scaled up to a 128-bit block size.
Joe Daemen, Rene Govaerts and Joos Vandewalle [5], Alex Biryukov, Jorge Nakahara Jr, Bart Preneel [4] and Philip Hawkes [7] have highlighted the need to change the key schedule of IDEA and they have found a number of weak keys through different methods with numbers varying from 251 to 264.
Kelsey, Bruce Schneier and David Wagner [6] have suggested that the problem of weak keys and key attacks can be minimized in situations where random keys and a secure key distribution system are used.

4.. PROPOSED CHANGES

After an analysis of IDEA, we developed a new algorithm R-IDEA with the some viable and feasible changes in IDEA.

R-IDEA: - Here we are going to modify IDEA algorithm with the help random numbers. The total number of keys required in IDEA algorithm is 52, so it is ensured that random number is in the range 1 to 52 and system date-time may also be used to generate the random numbers as it will always be unique. It acts as a seed for the random function. In the original IDEA, as previously discussed, the key schedule is static but it becomes dynamic in the R-IDEA. Hence, the security increases.

In the RIDEA algorithm we are using Random number to shift key randomly. But sending Random numbers to receiver is again a problem. To synchronize both (sender and receiver) with random numbers we can use an algorithm that will generate random number by mathematical calculation on both sides. It may decrease time of encryption and decryption and dependency on algorithm.
To generate random numbers we are going to use Diffie Hellman [11][12] algorithm. This algorithm itself is quite simple, but to crack is not a piece of cake for any hacker. Here we are going to discuss the random number sharing algorithm for R-Idea called RDE. How we are using it in RDE

The Exchange:

1. Sender chooses a random number a and computes u ^{a} ga (mod p), and sends u to receiver.
2. Receiver chooses a random number b and computes v ^{a} gb (mod p), and sends v to Sender.
3. Reciver computes the key k ^{a} ub ^{a} (ga)b (mod p).
4. Sender computes the key k ^{a} va ^{a} (gb)a (mod p).

Now, Sender and Receiver have the same key, namely k = gab (mod p).

A. OPERATION:-

The R-IDEA encryption algorithm requires three inputs – Plain text, Key and a random number as against only two in the IDEA as discussed earlier. So even if any wiretapped able to temporal his hands on the cipher text and the key, he cannot obtain the plaintext.
So at the receiving end, the decryption process requires the following additional steps: The key has to be first reshuffled back to the original form using the random numbers and then fed into the decryption algorithm. The random number has to be applied on the key every time before decryption, as it is a different number always, to obtain the sub-keys in the correct order. This is illustrated in the operational diagram Fig.3.

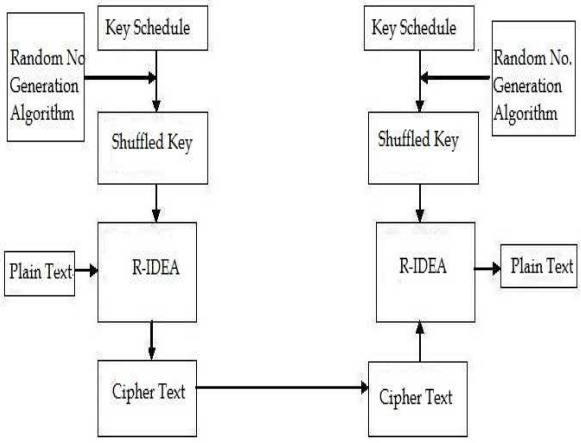


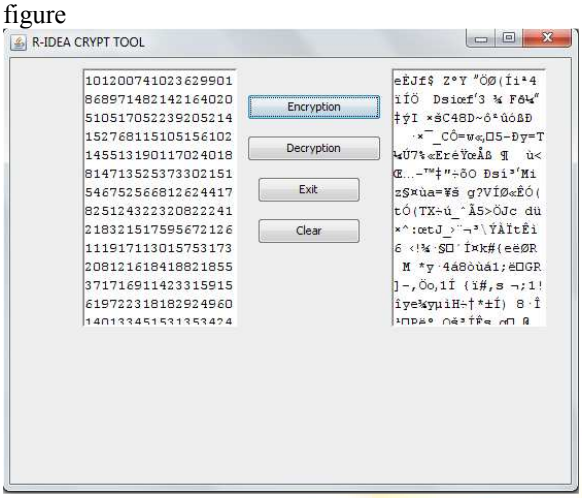
Fig.3 Operational diagram of R-IDEA

As it is seen in fig. 3, the original key is shuffled by the random number. The sub key whose serial number is equal to the random number, becomes the first sub-key of the schedule. Now this new randomization is fed into the R-IDEA along with the plaintext and the cipher text is generated. While decrypting the reverse procedure is followed. The same random number used in encryption is applied to the shuffle key schedule to get the original schedule. To send random numbers attach it with the message without encrypting.

5. EXPERIMENTAL RESULTS

The proposed R-IDEA algorithm was implemented on the following hardware specifications:
1. Intel Core 2 Duo processor.
2. Intel(R) Graphics Media Accelerator X4500MHD
3. 2GB SDRAM.
4. HARD DRIVE..., 320, S2, 2.5, 5.4, P11, SMSN-M7E
As discussed earlier, the implemented system based on the proposed algorithm consisted of these main modules i.e., random number generation (RDE). Key Shuffler,

data encryption and data decryption. All results related to these operations are presented in next section.
Figure .4 shows the graphic user interface (GUI) of the proposed algorithm. All functions was illustrated in this figure



All functions was illustrated in this figure like Encryption, Decryption, Clear and Exit.

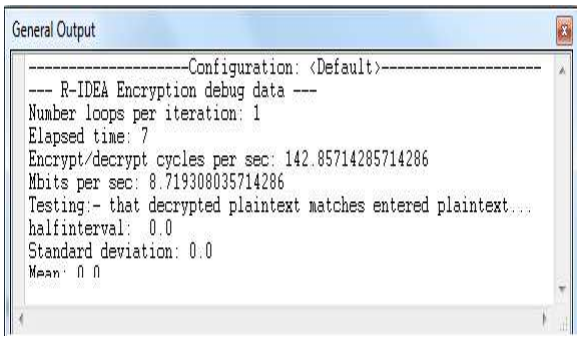
A. COMPARE THE PROPOSED ALGORITHM WITH RAS ALGORITHM

Comparison between the two algorithms, proposed algorithm with size 128 bit key and the IDEA algorithm with size 128 bit key has done. This comparison was conducted on five generated text, sized of 2000 bit, 4000 bit, 8000bit and 100000 bit for both algorithms. Table 1 shows the time taken in encryption and decryption.

Sr. No.	Text Size(bit)	R-Idea (ms)	Idea (ms)
1	2000	7.6293945	6.7884020
2	4000	7.6293955	6.9880102
3	8000	8.7193080	7.2395403
4	100000	18.608279	16.698013

Table 1.Time in Encryption and Decryption process

Figure 5 shows the output of R-Idea algorithm and speed test and standard deviation.



This algorithm has 0.0 standard deviation means entered data matches original one without any difference.

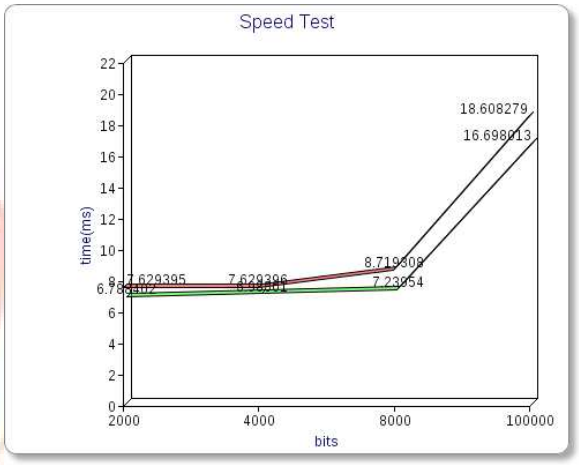


Fig.7 shows encryption time in both applications

From the previous figure, we found out that the encryption and decryption time consuming in the proposed algorithm is not too much larger than the IDEA algorithm, whereas the time increases according to the size of the text.

6 CONCLUSIONS

The problem of data transfer between sender and receiver are considered in this study. New algorithm was developed based on the Idea algorithm with the enhancement of key security for e-commerce based encryption. It is a powerful model to protect the e-commerce information. In this work we have tried to incorporate the goodness of IDEA and go even beyond it by introducing randomness and at the same time keeping the time and space constraints to a minimum. Initial experiments show that time taken for encryption and decryption depends on the length of the plaintext. An appreciable security enhancement is observed in the cipher texts obtained by the R-IDEA while making slight modification in the plaintext. The security can be further increased by using this algorithm with hybrid cryptosystem and by applying any other operation in place of shift key.

7. REFERENCES

1 . Sandipan basu international data encryption algorithm (idea) – a typical illustration , jgrcs,2011
2. Nick Hoffman, A Simplified Idea Algorithm
3. Suying Yang, Hongyan Piao, Li Zhang and Xiaobing Zheng: "An Improved IDEA Algorithm Based on USB Security Key", IEEE, 2007
4. Alex Biryukov, Jorge Nakahara Jr, Bart Preneel, Joos Vandewalle: "New Weak-Key Classes of IDEA", 4th International Conference, ICICS, 2002
5. Joe Daemen, Rene Govaerts and Joos Vandewalle: "Weak Keys for IDEA", Advances in Cryptology, 1993
6. Kelsey, Bruce Schneier, David Wagner, "Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES", 1997
7. Philip Hawkes, "Differential-Linear Weak Key Classes of IDEA", Springer-Verlag, 1998
8. Nick Hoffman, "A Simplified IDEA", Journal: Cryptologia, 2007
9. Wikipedia
10. L. Chang-Doo, C. Bong-Jun, P. Kyoo- Seok (2004), "Design and evaluation of a block encryption Algorithm using dynamic-key mechanism", Future
Generation Computer Systems 20, 327–338
11. David A. Carts A Review of the Diffie-Hellman Algorithm and its Use in Secure Internet Protocols, SANS
12. Jie Liu and Jianhua Li A Better Improvement on the Integrated Diffie-Hellman-DSA Key Agreement Protocol, IJNS, Vol.11, No.2, PP.114-117, Sept. 2010

AUTHORS INFORMATION

Dr. C. S Lamba has PhD degree in Computer science and currently working as a Head of department, RIET Jaipur Rajasthan India. He has more than 14 years of experience and holding diversified knowledge in the field of Information technology. He has published and Participated research paper / article in various conference/ seminar globally

Rahul Saxsena received MCA,M.Phil(CS) currently he is associated with the Jaipur National University at department of computer and systems sciences. She has also attended conference and seminar at repute