Encryption and Decryption using 3D Matrices

Priya. R
Assistant professor,
Department of AMCS, PSG college of Technology,
Peelamedu, Coimbatore

Janani. R
Pursuing B.Sc Computer Technology,
Department of AMCS, PSG college of Technology,
Peelamedu, Coimbatore

Lejna. M
Pursuing B.Sc Computer Technology,
Department of AMCS, PSG college of Technology,
Peelamedu, Coimbatore

Mangai Siva Surya. J
Pursuing Bsc Computer Technology,
Department of AMCS, PSG college of Technology,
Peelamedu, Coimbatore

Nandhini. M
Pursuing Bsc Computer Technology,
Department of AMCS, PSG college of Technology,
Peelamedu, Coimbatore

Rajalakshmi. V
Pursuing Bsc Computer Technology,
Department of AMCS, PSG college of Technology,
Peelamedu, Coimbatore

ABSTRACT
With rapid evolution of science and technology, people rely on networks a lot; the aspect of security receives a lot of attention. With a notion to provide security, many algorithms have been proposed in the world. This paper proposes a new method for encrypting and decrypting the data by using Shared-Secret Key, Session Key and Intermediate Key. Based on the length of plain text variable number of Session Key and Intermediate Key are generated.

General Terms
Security algorithms, Network Security

Keywords
Encryption, Decryption, DRDP

1. INTRODUCTION

1.1 Cryptography
Cryptography is a technique which allows encrypting the data in such a way that the decryption can be performed without the aid of sender. The messages to be encrypted, known as the plain text, are transformed by a function that is parameterized by a key. The output of the encryption process, known as the cipher text, is then transmitted. The art of breaking ciphers is known as cryptanalysis, and the art of devising them is cryptography. Both cryptanalysis and cryptography are collectively known as cryptology. [5]

The encryption of the plain text P using key K gives the cipher text C, Similarly the decryption of C gives the plaintext again. It then follows that C = EK (P) and P = DK (C). The entire operation can be represented as follows, P = DK (EK (P)). [5]

Cryptography can be classified as Symmetric key cryptography and Asymmetric key cryptography. In symmetric key cryptography the key used for encryption is same as the key used for decryption whereas in asymmetric key cryptography different keys are used for encryption and decryption. [5]

1.2 The Double-Reflecting Data Perturbation
Method

- The Double-Reflecting Data Perturbation Method denoted by DRDP is a perturbation technique used to preserve privacy in data mining.
- The distortion operation performed(op) to the confidential attribute is given by,

$$opj = \rho Ci + (\rho Ci - Ii) = |2\,\rho Ci - Ii|$$

where Ci ($1 \le i \le n$) is a confidential attribute and Ii ($1 \le i \le n$) is an instance of Ci. ρCi is defined by the following formula

$$\rho Ci = (maxCi + minCi)/2$$

where maxCi and minCi are the maximum value and minimum value of attribute Ci respectively.[1]

For example let the text be, “hello”. The ASCII values are 104,

101, 108, 108, 111.Now the DRDP is applied as follows

MAXIMUM VALUE: 111 MINIMUM VALUES: 101

ρCi = (111+101)/2 = 106

The data after applying DRDP becomes,

|104-(2*106)|=108 |101-(2*106)|=111

|108-(2*106)|=104 |108-(2*106)|=104

|111-(2*106)|=101

The final data set is 108, 111, 104, 104 and 101

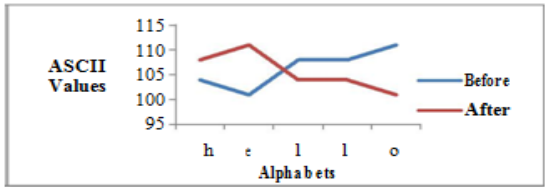

Figure 1 chart depicting the data before and after DRDP Figure 1 shows the values before and after applying DRDP

Thus the original values are reverberated in x axis and y axis.

1.3 Message Digest

- Message Digest deploys an hash function
- A hash function is a one-way function
- This digest is used to provide integrity (means that the message sent by the sender, is not altered in the course of its travel).
- It is done by, appending a digest on sender side which is calculated from the message.
- The receiver on receiving the message, extracts the digest, calculates the hash of the received message, if both the digest matches then there is no alteration in the message sent. Else the message is discarded.
- The message Digest is calculated in this algorithm also to serve the purpose of integrity and authentication.
- A slight modification is made in existing SHA256, where the word generation process includes the Shared Secret Key also in the process of generation.
- This enables only the sender and receiver to compute the correct digest, which provides authentication.

1.4 Format of cipher text

The figure 2 shows the format of cipher text

Figure 2 Format of cipher text

VL – Variable Length

1.5 Proposed system

The proposed system uses 3 keys namely,

- Shared Secret Key
- Session Key
- Intermediate Key

All keys are of 16 bytes and they may consist of Letters (both upper case and lower case), Numbers (0-9) and Symbols.

Shared Secret Key (SSK)
The Shared Secret Key is agreed upon by both Sender and Receiver.

Intermediate Key (IK)
Intermediate Key can be obtained as SSK Ex-OR SK

Session Key (SK)
The Session Key is randomly generated each time using random generator.

1.6 Variable Number of Keys

Depending upon the length of the text, variable numbers of keys are generated. Let the length of plain text be denoted as „L". If

'L' is not divisible by 16, increment it to the next digit divisible by 16. Two cases are possible since „L"/16 can be an even number or odd number.

- If „L"/16 is an even number, L/32 number of Session Keys and L/32 number of Intermediate Keys are obtained.
- If „L"/16 is an odd number, (L/32)+1 number of Session Keys and L/32 number of Intermediate Keys are obtained.

2. ENCRYPTION MODULE

- Generation of Variable Number of Session Keys
- Generation of Variable Number of Intermediate Keys
- Operations on plain text
- Creation of message digest

2.1 Generation of Variable Number of Session Keys

STEP I: Find the length of plain text as say, "l". If 'l' is not divisible by 16, increment it to the next digit divisible by 16.

STEP II: Arrange the 16 characters of Session Key in a form of 4*4 matrix.

STEP III: Repeat the below steps for i=2 to l/32 if l/16 is even or from i=2 to l/32 + 1 if l/16 is odd.

STEP IV: EX-OR previous matrix and index „i" and save in current matrix.

STEP V: Shift each of the column once

STEP VI: Perform DRDP for each row of the matrix

STEP VII: Increment i by one and go to step III.

2.2 Generation of Variable Number of Intermediate Keys

STEP I: Find the length of plain text as say, "l". If 'l' is not divisible by 16, increment it to the next digit divisible by 16

STEP II: Arrange the 16 characters of Shared Secret Key in a form of 4*4 matrix.

STEP III: EX-OR the previous matrix with the first matrix of generated Session Key.

STEP IV: Repeat the steps IV to VI of generation process of Session Key, with Intermediate Key in place of Session Key for i=2 to l/32 .

2.3 Operations on plain text

STEP I: Input the plain text

STEP II: Write the ASCII equivalent of each character in the plain text.

STEP III: Count the number of characters in the plain text say „l". If „l" is not divisible by 16, find l%16, say „r". Append „r" number of spaces.

STEP IV: Perform DRDP

STEP V: Group the plain text into groups of 16 characters, in the form of a matrix.

STEP VI: Transpose each matrix for i=1 to l/16

STEP VII: Perform DRDP on each matrix for every single row for i=1 to l/16.

The Encryption process consists of following sub modules.

STEP VIII: For each matrix, except the first matrix EX-OR the matrix with the previous matrix.

STEP IX: EXOR plain text matrices with the Intermediate Key matrices for even numbered plain text matrices and for odd numbered plain text matrices EXOR with Session Key for i=2 to l/16

STEP XI: Find the character equivalent of each element in each of the matrix and arrange them in a single row.

STEP XII: Append the number of padded bytes „r" to the end of it.

2.4 Creation of message digest

The digest is created using SHA-256 with a modification that Shared Secret Key is used to calculate the words for consecutive rounds. This enables only the sender and receiver to calculate the correct digest. Now the digest is appended along with the message.

Message schedule of existing SHA 256

Wj=Mji for j=0, 1, 2,.., 15 and

for(j=16 to 63)

{

Wj=σ1(Wj-2)+ Wj-7+ σ0(Wj-15)+ Wj-16

}

The modification to be done is

for(j=16 to 63)

{

Wj=σ1(Wj-2)+ Wj-7+ σ0(Wj-15)+ Wj-16+ sharedkey

}

Where Shared Secret Key is generated each time according to the procedure of Session Key, with Shared Secret Key in place of Session Key.

3. DECRYPTION MODULE

The Decryption process consists of following sub modules

- Extracting the digest, Intermediate Key, length
- Generation of Variable Number of Session Keys
- Generation of Variable Number of Intermediate Keys
- Operations on cipher text

3.1 Extracting the digest, Intermediate Key, length

STEP I: Padded bits would be of 1 byte. So totally the last (32+1+16=49) 49 bytes are extracted.

STEP II: Leaving the 32 bytes of digest, create the digest using rest of the bytes.

STEP III: Compare the digest with the digest received; if both the digest match then proceed to the following steps else discard the message.

Generation of Variable Number of Intermediate Keys

STEP I: Find the length of cipher text as say, "l"

STEP II: Arrange the 16 characters of the extracted Intermediate Key in a form of 4*4 matrix.

STEP III: Repeat steps IV to VI of generation of Session Key during encryption, with Intermediate Key in place of Session Key.

3.2 Generation of Variable Number of Session Keys

STEP I: Find the length of cipher text as say, "l"

STEP II: Arrange the 16 characters of Shared Secret Key in a form of 4*4 matrix.

STEP III: EX-OR the previous matrix with the first matrix of extracted Intermediate Key.

STEP IV: Repeat the steps IV to VI of generation of Session Key during encryption for i=2 to l/32 if l/16 is even or for i=2 to l/32 +1 if l/16 is odd.

3.3 Operations on cipher text

STEP I: Extract the cipher text

STEP II: Write the ASCII equivalent of each character in the cipher text.

STEP III: Group the cipher text into groups of 16 characters, in the form of a matrix.

STEP IV: From i=1 to l/2 EX-OR cipher text matrices with the Session Key matrices for odd numbered cipher text matrices and for even numbered cipher text matrices EX-OR with Intermediate Key

STEP V: For each matrix, except the first matrix EXOR the matrix with the previous matrix.

STEP VI: Perform DRDP, for i=1 to l/16.

STEP VII: Transpose each matrix

STEP VIII: Arrange all the elements of all the matrices in a single row

STEP IX: Perform DRDP

STEP X: Write the characters corresponding to the ASCII STEP I: Since SHA 256 was used to create the digest would be values of 256 bits, 32bytes. Intermediate Key would be of 16 bytes

STEP XI: Extract the number of characters needed leaving „r" padded bits.

4. COMPARISON OF PROPOSED SYSTEM AND EXISTING SYSTEM

The features below are discussed with using the Shared Secret Key as 'my project title' and Session Key as 'network security'

4.1 Diffusion

The change of a single character in plain text changes multiple characters in cipher text. This is achieved in this system by performing EX-OR with the previous matrix. But the existing system does not have this feature. Let plain text be"cryptography is devising of algorithms and cryptanalysis breaks" , the corresponding cipher is"ke2"-0/b"1b-koa<lt:x-47y(&(9ip6hlg¨D|uHÃ±WP¨µwM¥/0b~xc:m+&,8%3!!"

Figure 3 and Figure 4 show the plot of plain text ASCII VALUES VERSUS cipher text ASCII VALUES

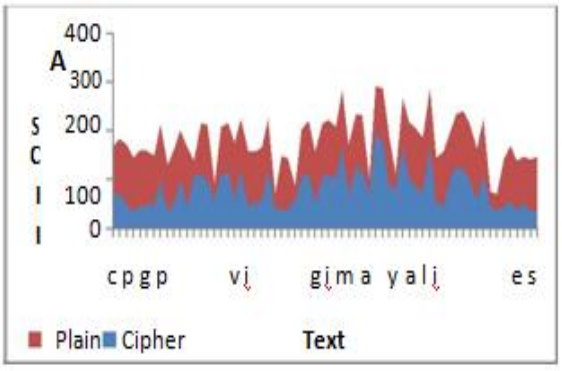

Figure 3 ASCII of Plain text and cipher text before change

The chart changes when the plain text "cryptography is devising of algorithms and cryptanalysis breaks" is changed to „Cryptography is devising of algorithms and cryptanalysis breaks", cipher text is "Ke2"4[F©"1bkoa<LT¾X±‚»Y¬ª¬½IPºHLG$d\Uh?-wp$1Wm!³´B^XC¾M¯ª°¼©•¥¥"

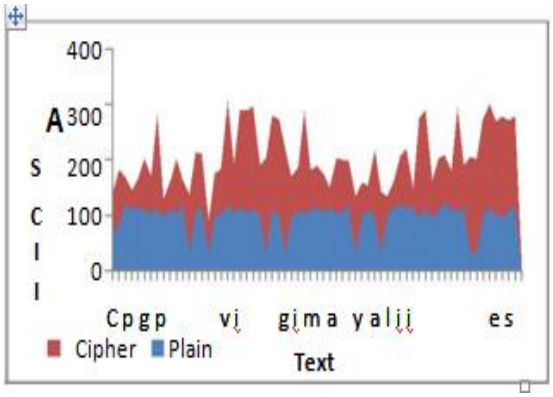

Figure 4 ASCII of Plain text and cipher text after change

4.2 Confusion

The change of one letter in the key introduces changes in many letters of cipher text. It is present even in existing system but it is still more existent in proposed system because of shift column operation on keys. Figure 5 and Figure 6 show the mapping of ASCII values of cipher text for the plain text "cryptography is devising of algorithms and cryptanalysis breaks". The single letter change in the key introduces considerable change in cipher text.

Shared Secret Key: AY project title

Ciphertext:GE2"-0/b"1b-koa<lt:x-47y(&(9ip6hlg¨D|uHÃ±WP¨iSI©/0b~xc:m+&,8%3!!

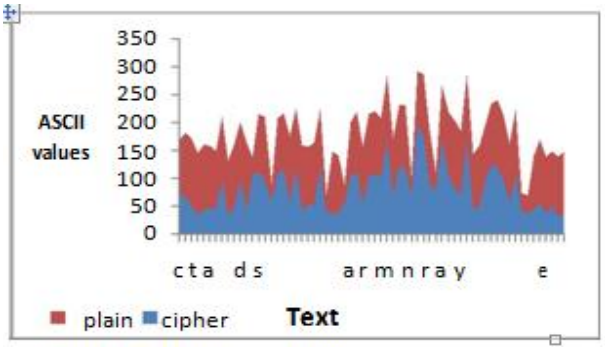

Figure 5 ASCII of Plain text and cipher text before change Shared Secret Key: my project title

Cipher text: ke2"-0/b"1b-koa<lt: 47y(&(9ip6hlg¨D|uHÃ±WP¨µwM¥/0b~xc:m+&,8%3!!

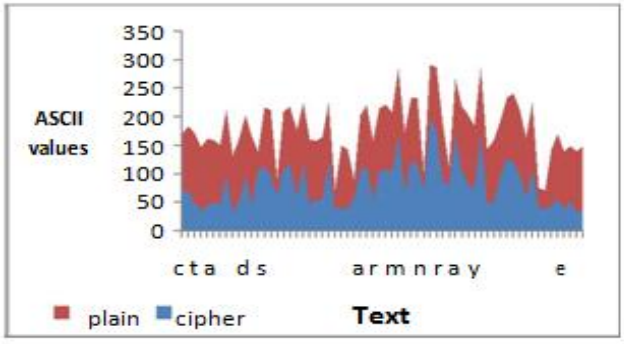

Figure 6 ASCII of Plain text and cipher text after change

4.3 Dot is not treated as delimiter

The proposed system does not treat dot as a delimiter, so the frequency of one sentence affects another. But it is not present in the existing system. Figure 7 and Figure 8 represent the DRDP calculation in the proposed and existing system.

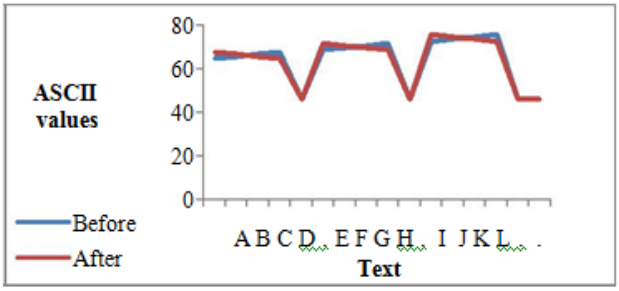

Figure 7 DRDP calculation in existing system

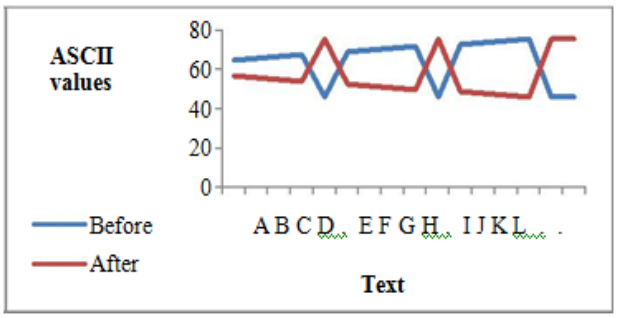

Figure 8 DRDP calculation in proposed system

4.4 Unique keys for each step

The proposed system aims to find unique keys for each step by employing EX-OR and shift column mechanism. But the existing system does not have that feature.

4.5 Message digest

The proposed system employs message digest which not only handle integrity but also authentication, since only the sender and receiver know the Shared Secret Key. But the existing system only provides integrity.

5. CONCLUSION

This paper has aimed to satisfy most of the aspect of information security in a vital way. The creation of digest for the cipher text makes out the decryption of messages to be carried out on receiver side only when the message is unaltered in transit and the Shared Secret Key provided by receiver must be same as the Shared Secret Key used by the sender, which only enables the creation of correct digest. The usage of EX-OR operation allows higher diffusion rate as the change in plain text is reflected in multiple places in cipher text. The system employs stream cipher which enables further confusion. Also the cipher text generation is dependent only on the plain text and no other external factors. The cipher text obtained for the message varies from time to time, since the Session Key is generated randomly each time.

6. REFERENCES

[1] Balajee Maram, K Lakshmana Rao, Y Ramesh Kumar, "Encryption and Decryption Algorithm using 2-D Matrices", International Journal of Advanced Research in Computer Science and Software Engineering Research Paper,(IJARCSSE), Vol.3, Issue 4, April-2013

[2] Balajee Maram and Narasimham Chella, "Double-reflecting Data Perturbation Method for Information Security", Oriental Journal of Computer Science & Technology, Vol. 5, No. (2): Pp. 283-288, December 2012

[3] Behrouz .A. Forouzan, "Cryptography and Network Security", first edition, Tata McGraw-Hill Companies, 2007

[4] Viji Amutha Mary, Dr. T. Jebarajan, "A Novel Data Perturbation Technique with higher Security", IJCET, No.3(2): Pp.126-132, 2012

[5] William Stallings, "Cryptography and Network Security- principles and practice", fifth edition, Pearson, 2011