Training of a New Probobilistic Encryption Scheme Using a Optimal Matrix Key

[1]*Prof. A.V.N. Krishna,* [2]*Dr. A.Vinaya Babu*
[1]*Professor, Computer Science department, Indur Institute of Engineering & Technology, Siddipet, Andhra Pradesh, India.Cell No. 9849520995, Email: hari_avn@Rediffmail.com*
[2]*Director, School For Continues &Distance Education, J.N.T.U, Hyderabad.*

Abstract:
In this work the probabilistic encryption algorithm is presented. Essentially a cryptosystem that is probabilistic secured means that it may be possible to modify a cipher text into another cipher text containing the same plaintext. This provides the possibility of perfectly replayable RCCA secure encryption. By this, we mean that anybody can convert a cipher text y with plaintext m into a different cipher text y that is distributed identically to a fresh encryption of m. It proposes such a rerandomizable cryptosystem, which is secure against semi-generic adversaries. The limitation with this work is more data overhead, ie each plain text character is mapped to three cipher text characters. In this work the model is trained for different keys and sub_keys are evaluated. An optimal key is selected which gives more sub keys. This adds more characters to the alphabet, which reduces the data overhead of the cipher text at less computational complexity.

Keywords: *Training of Probabilistic Algorithm, Reduced Data Overhead, Plain text, Multiple Cipher texts, Examples & Results.*

1. INTRODUCTION

In public key encryption there is always a possibility of some information being leaked out. Because a crypto analyst can always encrypt random messages with a public key, he can get some information. Not a whole of information is to be gained here, but there are potential problems with allowing a crypto analyst to encrypt random messages with public key. Some information is leaked out every time to the crypto analyst, he encrypts a message.

With probabilistic encryption algorithms [5], a crypto analyst can no longer encrypt random plain texts looking for correct cipher text. Since multiple cipher texts will be developed for one plain text, even if he decrypts the message to plain text, he does not know how far he had guessed the message correctly. To illustrate, assume a crypto analyst has a certain cipher text ci. Even if he guesses message correctly, when he encrypts message the result will be completely different cj. He cannot compare ci and cj and so cannot know that he has guessed the message correctly. Under this scheme, different cipher texts will be formed for one plain text. Also the cipher text will always be larger than plain text. This develops the concept of multiple cipher texts for one plain text. This concept makes crypto analysis difficult to apply on plain text and cipher text pairs.

Historically, encryption schemes were the first central area of interest in cryptography. They deal with providing means to enable private communication over an insecure channel. A sender wishes to transmit information to a receiver over an insecure channel that is a channel which may be tapped by an adversary. Thus, the information to be communicated, which we call the plaintext, must be transformed (encrypted)to a cipher text, a form not legible by anybody other than the intended receiver. The latter must be given some way to decrypt the cipher text, i.e. retrieve the original message, while this must not be possible for an adversary. This is where keys come into play; the receiver is considered to have a key at his disposal, enabling him to recover the actual message, a fact that distinguishes him from any adversary. An encryption scheme consists of three algorithms: The encryption algorithm transforms plaintexts into cipher texts while the decryption algorithm converts cipher texts back into plaintexts. A third algorithm, called the key generator, creates pairs of keys: an encryption key, input to the encryption algorithm, and a related decryption

key needed to decrypt. The encryption key relates encryptions to the decryption key. The key generator is considered to be a probabilistic algorithm, which prevents an adversary from simply running the key generator to get the decryption key for an intercepted message. The following concept is crucial to probabilistic cryptography:

[Probabilistic Algorithm].
A probabilistic algorithm [7] is an algorithm with an additional command RANDOM that returns "0" or "1", each with probability 1/2. In the literature, these random choices are often referred to as coin flips.

2. A GENERALIZED ENCRYPTION SCHEME FOR DATA ENCRYPTION USING A RANDOMIZED MATRIX KEY [10]

The steps that are involved in the proposed algorithm.

1. The letters of the alphabet were given numerical values starting from 0
2. A random matrix used as a key. Let it be A.
3. Generate a "ternary vector" for 3^3 values i.e from 0 to 26
4. Subtract 1(one) from each of the ternary values. Let this be "B".
5. Multiply $A * B^{|}$;
6. All the positive values are consider as 1, and all the negative values are consider as -1 and zero as 0.
7. Add 1(one) to all the ternary values.
8. A sequence is generated.
9. Basins will be developed from this periodic sequence which contains similar values.
10. Thus each basin contains unequal number of values .
11. Eliminate basins with minimum values (say 1 value).
12. Each basin represents one character.
13. Convert the plain text to equivalent numerical value.
14. Represent each character of plain text by corresponding basin values.
15. The basins are considered for each character of plain text based on chosen base value.

3. ADVANTAGES

1. High speed. It will make users to code the text into cipher text with in a few seconds.
2. It is almost impossible to extract the original information.
3. Even if the algorithm is known, it is difficult to extract the information.
4. Versatile to users. Different users of internet can use different modified versions of the new algorithm. Since in this algorithm all positive values are consider as +1 and all negative values are consider as -1 and zero as 0,it is impossible to generate the matrix key even if plain text and cipher text are known .
5. As per the matrix the same character is substituted by different alpha numerical value which provides more security for the message.
6. By suitably combing basins, the number of characters of the alphabet can be increased.

4. EXAMPLE

n= 0: 26 ;
r = ternary vector 0:26
r=r-1;

A= key= 2 5 -6
3 1 3
4 -2 -3

r = Sign (A*r).

r=r+1;
r=r(3,1)+r(2,1)*3+r(1,1)*9;

OUTPUT
n=0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
r=2 0 0 18 0 3 18 18 6 20 2 6 20 13 6 20 24 6 20 8 8 23 26 8 26 26 24.

Identifying similar values of sequence and putting them in one basin.
Thus the basins that can be formed using this sequences are

b(0)=(0,1,2,4,10)
b(1)=(3,5,6,7,8,9,11,12,14,15,17,18,19,20,21,23)
b(2)=(16,22,24,25,26)

Now considering a base value of 3 i.e. representing a character with random values from 3 basins, the total characters that can be represented in this alphabet is 3^3 =27.
For example if K=11, it can be represented as 1 0 2 = b1 b0 b2.

1. Encryption
1. Plain text a s k
01 19 11
2 Base value of 3
b(0)b(0)b(1) b(2) b(0) b(1) b(1) b(0) b(2).

3. Random values from basins
1 2 8 24 4 9 12 2 26
Cipher text C_i
a b h x d I l b z
Random values from basins
2 2 16 26 1 14 16 4 24
Cipher text Cj
b b p z a n p d x

Decryption.

1. Cipher text C_i
a b h x d I l b z

C_j
b b p z a n p d x

2. Basins
b(0) b(0) b(1) b(2) b(0) b(1) b(1) b(0) b(2).

3. Equivalent numerical
01 19 11

4. Plain text
01 19 11
a s k

5. ANALYSIS OF THE ALGORITHM

"A Generalized Encryption Scheme for data transmission using a Randomized Key".

The encryption model involves variable length key, a plain text and an algorithm which applies the key on the plain text to generate cipher text. This allows for the data to be transmitted over the network in some form which cannot be read by any intruder.

Thus in the given problem, the model will generate multiple cipher texts for one plain text. But the limitation with this work is one plain text character is mapped to three cipher text characters. To get a optimal mapping of plain text to cipher text at equal security, the model will be trained for different keys and an analysis of the models will be done. The keys that are used for training on the model are 3*3 keys and 4*4 matrix keys. By having small variations in the keys, the models will be studied for their increase in performance and working. The models will also be trained for their increase in security by having slight variations in the key values. The models will also be trained for changes in data overhead for small variations in the key values.

5.1 Training of the Model

```
n=0 to n
n1=floor (n/k); r1=n-n1*k
n2=floor (n1/k); r2=n1-n2*k;
rk=n(k-1);
r= [rk ….r2 r1];
```

```
                  ___
        r=r'= |    rk
                   .
                   .    |
              |    r2    |
              |    r1    |
              |__     ___|
```

```
r=r-1
A=key(k*k)
r=sign (A*r);
r=r+1;
r=r(k,1)+r(k-1,1)*k+……+r(1,1)*(k**k)
                End;
```

For example

If k=2, n=0:3,

n1=0, r1=0, r2=0.

n1=1, r1=0, r2=1.

n1=2, r1=1, r2=0.

n1=3,r1=1,r2=1.

```
r=[0 0; 0 1; 1 0; 1 1];
r'=  0  0
     0  1
     1  0
     1  1
r=r-1
```

Key = A = [2*2].

r= sign(A*r) /*where the sign function converts all positive values to 1 , all negative values to -1 and zero to 0.*/

r=r+1. /*r=r-1 and r=r+1 are being done in the algorithm to get all positive integers in the sequence generated.*/

r=output sequence generated.

Once the sequence is generated, similar values of the sequence are stored in one basin.
For example if the sequence generated is 2 0 0 18 0 3 18 18 ,
For n=0, corresponding r is considered and compared with r of other n values. If there is a match, the n values of all the matches are stored in one basin. The procedure is repeated till all the basins are formed. The number of basins corresponds to the number of characters of the alphabet. To increase the number of characters of the alphabet, the basins will be used in combinations.

The model will be trained for different inputs to generate different sequences which will used to generate basins. Different traing cases will be considered by considering a 3*3 matrix and changing the values of 3*3 matrixes and identifying the differences in sequence generated and the number of basins formed. The model is also trained for a 4*4 matrix key .The models are trained for different n values say 27, 81 for a base value 3 and 64 and100 values for abase value of 4.

The Security and data overhead in the proposed models depends on the key chosen. It also varies on the size of the key. Some keys of size [3*3] and [4*4] are considered. The keys are also considered by slightly varying the values by small amount say 1. In different cases, the sequences generated and the basins formed from these sequences are varying at a larger level.

5.2 Example

Case 1: considering the key A of size(3*3)

A=key | 2 5 -6 |
| 3 1 3 |
| 4 -2 -3 |
|

Sequence generated from the proposed model

n=0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

r=2 0 0 18 0 3 18 18 6 20 2 6 20 13 6 20 24 6 20 8 8 23 26 8 26 26 24.

thus the basins that can be formed using this sequences are

b(0)=(0,1,2,4,10)
b(1)=(3,5,6,7,8,9,11,12,14,15,17,18,19,20,21,23)
b(2)=(13)
b(3)=(16,22,24,25,26)

Case 2: By increasing the values of key by 1 at each row

A = key = 3 5 -6
4 1 3
5 -2 -3

Sequence generated

r= 1 0 0 18 0 0 18 18 3 20 2 6 20 13 6 20 24 6 23 8 8 26 26 8 26 26 25.

Thus the basins formed are

b(0)= (0,1,2,4,5,10)
b(1)= (3,6,7,8,9,11,12,14,15,17,18,19,20,23)
b(2)=(13)
b(3)= (16,21,22,24,25,26);

Case 3:

By decreasing the key values by 1 at each row.

A = key = 1 5 -6
2 1 3
2 -2 -3

Sequence generated by the model.

r= 11,1,3,20,0,6,18,18,6,20,2,6,20,13,6,20,24,6,20,8,8,20,26,6,23,25,15.
Basins formed are
b(0) = (0,11,4)
b(1) = (1);
b(2)= (2,3,10);
b(3)= (5,6,7,8,9,11,12,14,15,17,18,19,20,21,22,23,24,26)
b(4)=(13)
b(5)=(25).

In the case 3, for the given key, 6 basins have been formed. By this key , the alphabet supports 6 characters. By properly choosing each alphabet value by two basins, the number of characters of the alphabet can be increased to 36 values. This reduces the data overhead by half of its value.

Case 4:

By increasing the key size to [4*4], we can increase the number of basins and also the values of each basin, which increases the strength of the algorithm.

A = key = 1 5 -6 1
2 1 3 2
3 -2 -3 3
4 2 4 4

Sequence generated for n=0 to 80 values represented to the base 3.
r=0,33,60,0,0,6,0,9,20,54,57,60,0,0,40,0,19,20,54,54,60,54,54,74,9,20,20,33,60,60,0,6,26,9,20,26,57,
60,61,0,40,80,19,20,23,54,60,71,54,74,80,20,20,47,60,60,71,6,26,26,20,26,26,60,61,80,40,80,80,20, 23,26,60,71,80,74,80,80,20,47,80.
The basins formed by considering similar values,
b(0)=(0,3,4,6,12,13,15,30,39,5,31,57,10,36)
b(1)= (9,54,7,24,33,18,19,21,22,45,48,1,27,16,42)
b(2)=(20,60,8,17,25,26,34,43,51,52,69,78,2,11,28,29,37,46,54,55,63,72,32,35,58,59,61,62,71,9,18, 19,21,22,45,48,38,64,47,56,73,7,24,33,16,42,53,79,1,27)
b(3)= (23,74,44,70,49,75)
b(4)= (40,14,66)
b(5)= (80,41,50,65,67,68,76,77).

Case 5:

A =key= 2 5 -6 1
3 1 3 2
4 -2 -3 3
5 2 4 4

Sequence generated n=0 to 80 values represented to the base 3.
r=0,6,33,0,0,6,0,0,20,54,54,60,0,0,0,0,9.20,54,54,57,54,54,65,0,20,20,33,60,60,0,6,26,9,20,26,57,60,
61,0,40,80,19,20,23,24,60,71,54,74,80,20,20,47,60,60,80,15,26,26,23,26,26,60,71,80,80,80,80,20,26, 26,26,16,80,80,74,80,80,47,74,80.
The basins formed from the generated sequence
b(0)=(0,3,4,6,7,12,13,14,15,24,30,39,1,5,31,57,45,20,36,8,17,25,26,34,43,44,51,52,69,32,35,58,59, 61,62,70,71,38,47,64,53,78)
b(1)= (9,54,16,33,10,18,19,21,27,48,72,2,42)
b(2)=(23,65,60,11,28,29,37,46,54, 55,63,9,10,18,19,21,22,48,16,33,42,72,2,27)
b(3)= (40)

Case 6:

A =key = 3 5 -6 1

4 1 3 2
5 -2 -3 3
6 2 4 4

Sequence generated for n=0 to 80 values being represented to the base 3.
r=0,3,6,0,0,3,0,0,10,54,54,60,0,0,0,0,0,20,54,54,54,54,54,55,0,10,20,33,60,60,0,6,26,9,20,26,57,60,
61,0,40,80,19,20,23,54,60,71,74,54,80,20,20,47,60,70,80,25,26,26,26,26,26,60,80,80,80,80,80,20,2
6,70, 80,80,77,80,80,74,77,80.
Basins formed from the generated sequence
b(0)=(0,3,4,6,7,12,13,14,15,16,22,28,37,79,80,1,5,2,29,39,48,54,62,63,64,65,66,67,71,72,74,75,78,
9, 10,18,19,20,43,47,45,46,76,31,8,23,40,17,24,32,41,49,50,68,51,42,38)
b(1)=(25,33,55,21,26,60,30,56,57,58,59,69,11,27,35,44,52,61,34,36)
b(2)= (70,53)
b(3)= (77,73)

Case 7:

A=key= 1 5 -6 1
2 1 3 2
3 -2 -3 1
4 2 4 4

Sequence generated for n=0-100 for a base value of 4.
r=0,72,136,136,0,0,8,42,0,16,34,42,32,34,34,34,128,132,136,137,0,0,85,170,0,33,34,38,33,34,34,34
,
128,128,136,154,128,128,162,170,16,34,34,98,34,34,34,34,128,128,170,128,145,162,162,161,162,
162,162,34,34,34,34,72,136,156,154,0,8,42,43,16,34,42,42,34,34,34,42,132,136,137,170,0,85,170,
33,34, 38,42,34,34,34,38,128,136,154,170,128.
Basins formed the generated sequence
b(0)= (0,4,5,8,20,24,21,68,84,6,69)
b(1)=(16,128,9,40,72,32,33,36,37,48,49,52,96,100,1,64,12,25,28,38)
b(2)=(34,136,10,13,14,15,26,29,30,31,41,42,44,45,46,47,60,61,62,63,73,76,77,78,89,92,93,94,2,3,1
8, 65,81,97,7,14,70,71,74,75,79,91)
b(3)=(38,162,27,90,95,54,55,57,58,59,)
b(4)=(85,22)
b(5)=(98,154,43,35,67)

Case 8:

A=key = 2 5 -6 1; 3 1 3 2; 4 -2 -3 1; 5 2 4 4;

Sequence generated for n=0:100 for a base value of 4.
r=0,8,72,136,0,0,8,26,0,0,34,42,16,34,34,128,128,136,136,0,0,0,106,0,16,34,34,32,34,34,34,128,12
8,
132,138,128,128,146,166,0,34,34,34,34,34,34,34,128,128,128,154,128,128,162,162,144,162,162,1
62,
34,34,34,34,72,136,136,154,0,8,42,42,16,34,42,92,34,34,34,42,132,136,137,170,0,85,170,170,33,3
4, 38,42,34,34,34,38,128,136,154,170,128.
Basins formed from generated sequence
b(0)=(0,45,8,9,20,21,22,24,40,68,84,1,6,69)
b(1)=(16,128,12,25,72,17,32,33,36,37,48,49,50,52,53,96,100,2,64,28,88)
b(2)=(26,34,7,10,13,14,15,27,29,30,31,41,42,43,44,45,46,47,60,61,62,63,73,76,77,78,89,92,93,94,1
1, 70,71,74,79,91,75)
b(3)=(38,146,90,95)
b(4)=(85)

Case 9:

A= key= [3 5 -6 1; 4 1 3 2;5 -2 -3 3;6 2 4 4]
Sequence generated for n=0:100 for a base value of 4.
r=0,4,8,72,0,0,4,9,0,0,17,38,0,33,34,34,128,128,136,136,0,0,26,0,0,34,34,16,34,34,34,128,128,128,137,128,128,129,162,17,34,34,33,34,34,34,128,128,128,134,128,128,146,162,64,162,162,162,34,34,
34,72,136,156,154,0,8,42,42,16,34,42,42,34,34,34,42,132,136,137,170,0,85,170,170,33,34,38,42,34, 34,34,38,128,136,154,170,128
Basins form the generated sequence
b(0)=(0,4,5,8,9,12,20,22,23,38,65,80,97,98,99,100,1,6,2,66,7,11,86,91)
b(1)=(16,128,26,69,15,30,31,32,34,35,46,47,48,50,51,92,96,21,14,24,25,27,28,29,40,41,43,44,45,5,56,60,70,73,74,75,85,88,89,90,81)
b(2)=(17,136,10,39,18,62,63,78,93)
b(3)=(33,137,13,42,84,67,68,71,72,76,87,3,61)
b(4)=(64,154,54,94)

From the above study, it has been observed that the generated model is trained for a [3*3] and [4*4} keys on a input ternary vector of 27 & 81 values with a base of 3, and on 100 values with a base of 4. The model is trained for a slight variations in the key values and observed the variations in the sequence generated, and the basins formed from these sequences. It has been observed that the more the values of the vector, it does not guarantee in the increase of basins. But increases in the values of basins are observed. It is also observed that for a certain variations in the key, the model is generating more basins. Thus for a random generated key, the number of basins is evaluated. By slightly reducing the key values the number of basins is increasing. A better metric can be provided to a random key which provides more number of basins. Since the number of basins form the characters of the alphabet, the more the number of basin, and the more the characters of the alphabet. A combination of basins will increase the number of characters of the alphabet. It is also observed that the more the number of basins, the less will be the data overhead. For example, if the basins generated is 6 and if 2 basins are considered for each character, then the number of characters of the alphabet will be 36. If the number of basins formed is only 3, then each character has to be represented by 3 basin values if the alphabet is supporting 27 characters. The analysis and results of the
Training model is illustrated in the table no.1

Table 1 : Relationship between Random Key considered with the Basins (Sub Keys) generated

Cases	Random Key Considered	Number of Basins formed	Metric that can be mapped between Key and Basins formed
1. considered to the base 3	[2 5 -6; 3 1 3; 4 -2 -3]	4	Low
2 .considered to the base 3	[3 5 -6; 4 1 3; 5 -2 -3]	4	Low
3 .considered to the base 3	[1 5 -6; 2 1 3; 2 -2 -3]	6	High
4 .considered to the base 3	[1 5 -6 1;2 1 3 2;3 -2 3 3;4 2 4 4]	6	High
5 .considered to the base 3	[2 5 -6 1;3 1 3 2;4 -2 -3 3;5 2 4 4]	4	Low
6 .considered to the base 3	[3 5 -6 1;4 1 3 2;5 -2 -3 3;6 2 4 4]	4	Low
7.considered to the base 4	[1 5 -6 1;2 1 3 2;3 -2 -3 1;4 2 4 4]	6	High
8.considered to the base 4	[2 5 -6 1;3 1 3 2;4 -2 -3 1;5 2 4 4]	5	Medium
9.considered to the base 4	[3 5 -6 1;4 1 3 2;5 -2 -3 3;6 2 4 4]	5	Medium

Table 2 : Level of Metric between Key and Basins (Sub Keys) generated.

S no.	No. of basins formed	Ratio of Cipher text to plain text	Number of characters of alphabet	Data overhead	Metric that can be mapped between Key and Basins formed
1	3	3	27	3	low
2	4	3	64	3	low
3	5	3	125	3	low
4	6	2	36	2	high

Table 3 : Possible Relationship between Plain text and Cipher Text Generated in Probabilistic encryption Algorithm (Model 4) (Depending on the random Key Chosen).

S no.	Ratio of Cipher text to Plain text	Number of basins formed	Possible characters in the alphabet	Number of values in each Basin	Possible Conversions of one plain text character to one cipher text character
1	3	3	27	6,15,5	30-450
2	3	4	64	6,1,13,6	6-78
3	2	6	36	3,1,3,1,1,17	3-51

Thus depending on the key considered and the number of sub keys(Basins) generated one plain text character can be replaced by different combinations of cipher texts. As per the above tables it is observed that if one plain text character is replaced by 3 basins, then the number of possible cipher texts for one plain text character is 30 to 450 combinations. For the different possible keys different combinations are possible which varies from a lesser value to higher values.

Conversion of plain text to cipher text.

Table 4 : Conversion of plain text to cipher text for a 5 character text.

Case 1:

Cipher text for the developed model	Vbp0vvjspvdj0ex

Table 5 : Conversion of plain text to cipher text for a 20 character text.

Case 4:

Plain Text for a 20Character text	The reqiurementsof inf
Cipher text for Developed model	V0pdvyacpybal0yspxzkdlday00dqycchdoyelpyappdolpdbydqaasrxdxa

Table 6 : Multiple Cipher Texts to one Plain Text, for a 3 to 1 mapping,

From the above analysis we can see that for the same plain text, multiple cipher texts can be generated.

Plain text for a 5 character value	The re
Cipher texts	Vbp0vvjspvdj0ex
	Xapapy0izyoedkv
	Ydp0ppmbby00dt
	V0pdvyacpybal0y

Table 7 : Multiple Cipher Texts to one Plain Text, for a 2 to 1 mapping.
Note: Considering Case 3 of Table1, one plain text character being mapped to two cipher text characters.

From the above analysis we can see that for the same plain text, multiple cipher texts can be generated.

Plain text for a 5 character value	The re
Cipher texts	Gbab0yf0ay
	Kcajkygkky
	Tjackyt00y
	Ljacdyzdky

6. CONCLUSION

The developed algorithm is trained for different combinations of keys. From the above study, it has been observed that the generated model is trained for a [3*3] and [4*4} keys on a input ternary vector of 27 & 81 values with a base of 3, and on 100 values with a base of 4. The model is trained for a slight variations in the key values and observed the variations in the sequence generated, and the basins formed from these sequences. It has been observed that the more the values of the vector, it does not guarantee in the increase of basins. But increases in the values of basins are observed. It is also observed that for a certain variations in the key, the model is generating more basins. Thus for a certain random generated key, the number of basins is evaluated. By slightly varying the key values, the difference in number of sub keys(basins) formed is identified. A better metric is provided to a random key which provides more number of basins. Since the number of basins form the characters of the alphabet, the more the number of basins, and the more the characters of the alphabet. A combination of basins will increase the number of characters of the alphabet. It is also observed that the more the number of basins, the less will be the data overhead.

7. REFERENCES

1. Georg J.Fuchsbauer: An Introduction to Probabilistic Encryption, 'Osjecki Matematicki List 6(2006), pp37-44.
2. Guo D, Cheng L.M., Cheng L.L: A New Symmetric Probabilistic Encryption Scheme Based on Chaotic Attractors of Neural Networks, Applied Intelligence, Vol 10, No.1, Jan 99, pp 71-84.
3. Henry Baker and Fred Piper : Cipher systems(North wood books, London 1982).
4. I.Chien-Chiang: Efficient improvement to XTR and two padding schemes for probabilistic trapdoor one way function, 2005-12-05.
5. 5.J.William stalling **:**Cryptography and network security (Pearson Education,ASIA1998)
6. Josh Benaloh: Dense Probabilistic encryption, Feb. 99.
7. Krishna A.V.N.: A new algorithm in network security, International Conference Proc. Of CISTM-05, 24-26 July 2005, Gurgoan, India.
8. Krishna A.V.N., Vishnu Vardhan.B.: Utility and Analysis of some Encryption algorithms in E learning environment, International Convention Proc. Of CALIBER 2006, 02-04 Feb. 2006, Gulbarga, India.
9. Krishna A.V.N., S.N.N.Pandit: A new Algorithm in Network Security for data transmission, Acharya Nagarjuna International Journal of Mathematics and Information Technology, Vol: 1, No. 2, 2004 pp97-108
10. Krishna A.V.N, S.N.N.Pandit, A.Vinaya Babu: A generalized scheme for data encryption technique using a randomized matrix key, Journal of Discrete Mathematical Sciences & Cryptography, Vol 10, No. 1, Feb 2007, pp73-81
11. Krishna A.V.N., A.Vinaya Babu: Web and Network Communication security Algorithms, Journal on Software Engineering, Vol 1,No.1, July 06, pp12-14
12. Krishna A.V.N, A.Vinaya Babu: Pipeline Data Compression & Encryption Techniques in e-learning environment, Journal of Theoretical and Applied Information Technology, Vol 3, No.1, Jan 2007, pp37-43
13. Lester S. Hill, Cryptography in an Algebraic Alphabet, The American Mathematical Monthly 36, June-July 1929, pp306–312.
14. 14.Lester S. Hill, Concerning Certain Linear Transformation Apparatus of Cryptography, The American Mathematical Monthly 38, 1931, pp135–154.
15. Maybec.J.S. (1981), Sign Solvability, Proceedings of first symposium on computer assisted analysis and model simplification, Academic Press, NY.
16. Pandit S.N.N (1963): Some quantitative combinatorial search problems. (Ph.D. Thesis).
17. Pandit S.N.N (1961): A New matrix Calculus, J Soc., Ind. And Appl. Math. Pp 632-637.
18. Pascal Pallier: Public key Cryptosystem Based on Composite degree Residuosity Classes, Advances in Cryptology- EUROCRYPT' 99, vol. 1592 of lecture notes in computer science, pp.223-238, Springer –Verlag, 1999.
19. Phillip Rogaway : Nonce Based Symmetric Encryption, www.cs.ucdavis.edu/rogeway.
20. R.S.Thore & D.B.Talange: Security of internet to pager E-mail messages (Internet for India-1997IEEE Hyderabad section) pp.89-94.
21. Terry Ritter: Substitution Cipher with Pseudo-Random Shuffling.

Article received: 2008-05-27