Secure Data Transmission using Goldbach Codes and RSA Algorithm

Jibendu Kumar Mantri, Rajalaxmi Mishra

Abstract – Information transmitted through the insecure network need to be secured by using different methods. There are several cryptographic methods to ensure secure data transmission. The compression algorithms are used to compress the information, then on the compressed information encryption algorithms can be applied so as to reduce the time of encryption. There are several data compression algorithms available to compress the data. Our proposed scheme is a combination of Goldbach Code Algorithm for data compression, and the RSA algorithm for data encryption.

Key words- Cryptography, RSA algorithm, Data compression, decompression, Goldbach codes

I. INTRODUCTION

Various cryptographic techniques are there to make the transmission of data secure. The algorithms of cryptography are classified broadly into private or symmetric key algorithms and public or asymmetric key algorithms [17]. For symmetric key cryptosystem the encryption key is the same as the decryption key. For public key cryptosystem, the encryption key is different from the decryption key. RSA cryptosystem is the most popular asymmetric cryptographic schemes.

To save storage requirements and transmission time, data can be compressed. There are several compression methods available, which can be broadly classified as " Lossy compression" and "Lossless compression". The information can be compressed by using lossless data compression methods such that no information is lost at the time of decompression of the compressed information. After decompression the information produced by Lossy data compression algorithms is not exactly same as the information before compression.

Some information loss is tolerable in these schemes. These compression reduce the file size by reducing some unnecessary data which is difficult to be acknowledged by human beings after decompressing [1,7]. In this paper we have presented an encryption scheme along with data compression scheme. For data compression, we have used Goldbah codes and for encryption we have used RSA algorithm.

Revised Manuscript Received on November 15, 2019

Dr. Jibendu Kumar Mantri, Department of Computer Application. North Orissa University, Baripada, Odisha, India. Email: Jkmantri@gmail.com

Rajalaxmi Mishra, Department of MCA, College of IT and Management Education, Bhubaneswar, Odisha, India Email: rajalaxmi_mishra@yahoo.com

Rest of the parts of this paper is arranged as, section II includes review of different associated literatures, the theory about Goldbach Codes and RSA algorithm is presented in section III, section IV has the results and discussions, section V includes Conclusion and future work and finally section VI includes references.

II. LITERATURE REVIEW

Now-a-days people are more dependent on internet to exchange confidential information in a secure manner. So the application of cryptography [4] is becoming essential in our life. Cryptography provides various mathematical formulas or to secure the network communications and data authentication. Cryptographic algorithms are broadly classified as Symmetric key or private key algorithms [9] where the keys used for encryption and decryption are the same and asymmetric key or public algorithms where one key is used for encryption and a different key is used for decryption[3]. Some popular symmetric algorithms include DES [6], TDES [13], Blow fish [20], IDEA [8], AES [6, 21], and RC6 [14, 16]. Some popular public key algorithms includes PGP[10], Diffie-Hellman keys [11], and SSH [22]. The most popular public key algorithm is RSA algorithm [6,15]. [7] Presents a survey on various data compression algorithms. [1] Discusses various algorithms for text data compression. [5] Provides various methods for data compression. [18] Discusses the Goldbach codes algorithm for data compression. [19] provides a method of data compression using Goldbach |Codes and data encryption using Vigenere Cipher Algorithm.

III. THEORITICAL OVERVIEW

We are presenting a scheme where the information is compressed using Goldbach Codes and then encrypted by using RSA algorithm. The encrypted message is then sent via Internet to the receiver. After receiving the message, it has to be decrypted first. Then the decrypted message has to be decompressed to get the original information.

A. RSA Algorithm

In this paper, the information is encrypted using RSA public key algorithm

KEY GENERATION FOR RSA ALGORITHM

- It is required to select two prime numbers p and q randomly.
- n = p * q and $\phi(n) = (p-1) * (q-1)$ be computed
- One arbitrary integer 'e' is to be selected with $1 < e < \phi(n)$ such that gcd $(e, \phi(n))=1$

- The integer d is to be computed which satisfies $1<d<\phi(n)$ and $ed \equiv 1 \bmod \phi(n)$
- Return (n,e,d)

Public key components are e and n where as Private key is d.

The information is enciphered by using public key of the receiver and deciphered by using private key of the receiver.

RSA ENCRYPTION SCHEME

Input: The plain text m ϵ [0, n - 1] and The public key i.e (n, e)

Output: Ciphered text c.

- $c = m^e \pmod n$ has to be computed
- The ciphered text, c has to be returned.

RSA DECRYPTION SCHEME

Input: RSA private key d and the ciphered text c

Output: Plain text m.

- $m = c^d \pmod n$ has to be computed.
- The deciphered text m has to be returned which is the original text.

B. Goldbach Codes Algorithm

The information is first compressed by using Goldbach codes then encrypted by using RSA algorithm.

In 2001, Peter Fenwick utilized Goldbach conjecture (presuming that it is true) to develop a completely novel prime number based class of codes. The basis of this class of codes is the prime numbers. An even integer can be represented as a sum of two prime numbers; it can be represented with exactly two one's. The even number 22 equals 5+17 and therefore it can be presented as 100010, it requires six bits to be assigned the prime weights (from left to right) 17,13,11,7,5,3. Now the bit stream is reversed to get the least significant bit 1, which gives 010001. A number in this form is easy to find and read from a very long bit string. One has to stop reading at the second occurrence of 1. This rule is similar to unary codes (a sequence of zeros followed by a single one), while reading the bits one has to stop at the first occurrence of 1. Hence the Goldbach Codes are treated as the expansion of unary codes [10].

The algorithm of Goldbach codes to compress the text data has the given steps:

1. The alphabets or symbols in the message are read to get the frequency of appearance of every symbol or alphabet. For the highest frequency symbol n=1, and so on.
2. A code word has to be found for each character by finding the prime numbers such that sum of two primes represent the number. The sequence of prime numbers are 3, 5,7, 11, 13, 17 and so on.
3. The code word representation has exactly two ones.

Table I gives Goldbach G0 codes

Table I: The Goldbach G0 Codes

n	2 (n+3)	Sum of primes	Code word
1	8	5 + 3	11
2	10	7 + 3	101
3	12	7 + 5	11
4	14	11 + 3	1001
5	16	11 + 5	101
6	18	11 + 7	11
7	20	13 + 7	101
8	22	17 + 5	10001
9	24	13 + 11	11
10	26	19 + 7	10001
11	28	17 + 11	101
12	30	17 + 13	11
13	32	19 + 13	101
14	34	23 + 11	10001
15	36	23 + 13	1001

IV. RESULT AND DISCUSSION

The message m= GOOD MORNING that is intended to be sent to the receiver. This message needs to be compressed first by means of Goldbach Codes. The process of compression is explained in Table II.

Table II: The Process of Compression

Character	Frequency	n	2 (n+3)	Primes	Code word
O	3	1	8	5 + 3	11
G	2	2	10	7 + 3	101
N	2	3	12	7 + 5	011
D	1	4	14	11 + 3	1001
Space	1	5	16	11 + 5	0101
I	1	6	18	11 + 7	0011
M	1	7	20	13 + 7	00101
R	1	8	22	17 + 5	010001

The resultant binary string 10111111001010100111100101011010001011101 formed by compression of message m=GOOD MORNING, where the total number of bits are 41.

To implement RSA algorithm we have chosen the parameters as

P=11 and q= 7 are the selected prime numbers so that $n=p*q=11X\ 7=77$

$\Phi(n)=(p-1)(q-1)=10X6=60$

e chosen as 7 such that GCD(7,60)=1

Public key of the receiver {e,n}={7,77}

d is calculated as $d = e^{-1} \pmod n = 43$

The receiver's Private Key { d, n}={ 43, 77}

RSA algorithm encrypts plain text in blocks; with every block having a binary value less than n. The size of the block must be at most log2 (n). Hence k is the size of the block where $2k < n \leq 2k+1$.

.

The value of n=77, hence the block size is 6 bits. There are total no. of 41 bits, so we have 6 blocks of size 6 bits and 1 last block of size 5 bits.
The block 1: 101111
Its decimal value =47
Encrypted value=$(47)^7$ mod 77=75

The block 2: 110010
Its decimal value = 50
Encrypted value =$(50)^7$ mod 77=8

The block 3: 101001
Its decimal value = 41
Encrypted value = $(41)^7$ mod 77=13

The block 4: 011101
Its decimal value = 29
Encrypted value = $(29)^7$ mod 77=50

The block 5: 000101
Its decimal value = 5
Encrypted value =$(5)^7$ mod 77=47

The block 6: 100110
Its decimal value = 38
Encrypted value =$(38)^7$ mod 77=3

The block 7 (last block) : 11101
Its decimal value = 29
Encrypted value =$(29)^7$ mod 77=50

The sequence of numbers 75,8,13,50,47,3,50 are sent from sender to receiver.
The receiver decrypts the numbers and presents them in a block of six bits except the last block.

The received number =75
The decrypted value=$(75)^{43}$ mod 77= 47
The data block = 101111

The received number =8
The decrypted value=$(8)^{43}$ mod 77= 50
The data block = 110010

The received number =13
The decrypted value=$(13)^{43}$ mod 77= 41
The data block = 101001

The received number =50
The decrypted value=$(50)^{43}$ mod 77= 29
The data block = 011101

The received number =47
The decrypted value=$(47)^{43}$ mod 77= 5
The data block = 000101

The received number =3
The decrypted value=$(3)^{43}$ mod 77= 38
The data block = 100110

The received number =50
The decrypted value=$(50)^{43}$ mod 77= 29
The data block = 11101

The receiver has the binary string 10111111001010100111100101011010001011101. Then the string is decompressed by following the arrangement in Table 2., the result of this decompression is GOOD MORNING.

As we are compressing the text then encrypting it, the cost of encryption and decryption time is saved.

V. CONCLUSION AND FUTURE SCOPE

In the present work, we have used Goldbach codes for compression of the information. Then used RSA algorithm, to encrypt the compressed data. This method ensures secure transmission of the information and it is also cost effective because of compression. In future, we may also use various compression and encryption schemes and compare the efficiency. Moreover, the proposed scheme is subjected to side channel and plain text attack, which needs to be countered effectively in future.

REFERENCES

1. A. V. Singh and G. Singh, "A Survey on Different text Data Compression Techniques," International Journal of Science and Research (IJSR), vol. 3, no. 7, pp. 1999-2002, 2014.
2. Annapoorna Shetty, Shravya Shetty and Krithika, “A Review on Asymmetric Cryptography RSA and ElGamal Algorithm”, International Journal of Innovative Research in Computer and Communication Engineering, 2014.
3. Bruen, A. A. &. Forcinito and M. A, Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century, John Wiley & Sons, 2011, p. 21.
4. Coron, J. S. “What is cryptography?”, IEEE Security & Privacy Journal, 12(8), 2006, pp. 70-73.
5. D. Salomon and G. Motta, "Handbook Of Data Compression Fifth Edition," Springer, 2010.
6. Gurpreet Singh, Supriya, “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security” International Journal of Computer Applications, pp. 0975 – 8887.
7. H. Jani and J. Trivedi, "A Survey on Different Compression Techniques Algorithm for Data Compression," International Journal of Advanced Research in Computer Science & Technology (IJARCST), vol. 2, no. 3, 2014.
8. Harivans Pratap Singh, Shweta Verma , Shailendra Mishra, “Secure-International Data Encryption Algorithm”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 2, February 2013.
9. Huang, Y., Chen, L., Tang, S, “Information security and encryption decryption core technology”, Electronic Press, 2001.
10. Huiping, H., Yan, R., Lan, Z, “Using PGP software to realize safe sending and receiving email”, Comput. Secur. **1**, 2011, pp. 52–54.
11. Li Xin, "An Improvement of Diffie-Hellman Protocol", Network & Computer Security, vol. 12, 2007, pp. 22-23.
12. Massey, J.L, "An Introduction to Contemporary Cryptology", Proceedings of the IEEE, Special Section on Cryptography, May 1988, pp. 533-549.
13. Nisha Rani, Mrs. Neetu Sharma, “Suspicious Email Detection System via Triple DES Algorithm: Cryptography Approach”, International Journal of Computer Science and Mobile Computing, Vol.4, Issue 5, May 2015.
14. R.L. pavan, M.J.B. Robshaw, R.Sidney, and Y.L. Yin, “The RC6 Block Cipher”, v1.1, August 1998.
15. Ronald L. Rivest, Adi Shamir, Len Adelman, "On Digital Signatures and Public Key Cryptosystems," MIT Laboratory for Computer Science Technical Memorandum 82, April 1977.
16. RSA Laboratories, “RC6 Block Cipher”, Historical: RSA Algorithm: Recent Results on OAEP Security: RSA Laboratories submissions, 2012.

17. S.William and W. Stallings, Cryptography and Network Security, 4/E : Pearson Education India, 2006.
18. S. D. Nasution and Mesran, "Goldbach Codes Algorithm For Text Compression," International Journal of Software & Hardware Research in Engineering (IJSHRE), vol. 4, no. 11, pp. 43-46, 2016.
19. S. D. Nasution, G. L. Ginting, Muhammad Syahriza and Robbi Rahim, "Data Security Using Vigenere Cipher and Goldbach Codes Algorithm" , International Journal of Engineering Research & Technology (IJERT), Vol. 6, Issue 01, January-2017
20. Saikumar Manku, and K. Vasanth, "Blowfish Encryption Algorithm for Information Security", ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 10, June 2015.
21. The web page gives the AES contains: http://www.nist.gov/ CryptoToolkit.
22. Ylonen, T. and Lonvick, C, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

AUTHORS PROFILE

Dr. Jibendu Kumar Mantri, Reader Dept. of Computer Application, North Orissa University. Odisha, India. Qualification : M.Sc M.Phil M.Tech Ph.D. 68 no. of research papers and 8 books already published. Areas of Interest : AI, Business Process Re-engineering, Computer Security. Life Member ISTE ,and ISCA.

Rajalaxmi Mishra, a faculty member in CIME, Bhubaneswar, Odisha, India.Qualification: M.Sc (Statistics), M.Phil(Statistics), M.Tech (Computer Science), Currently perusing PhD in Computer Application in North Orissa University, Odisha, India. Research Interest is Cryptography and Network Security.