Journal of Excellence in Computer Science and Engineering, Vol. 1(1) 2015, pp. 1-10

RESEARCH ARTICLE

An Algorithm for Encrypting/Decrypting Textual Messages

[*]S Umamaheswaran[1], K Senthil[1], R Rajaram[1]

[1]Department of Computer Science and Engineering, Vickram College of Engineering, Enathi, 630561, Tamil Nadu, India.

Received-20 August 2015, Revised-15 September 2015, Accepted-8 October 2015, Published-12 October 2015

ABSTRACT

This paper evolves an algorithm for encrypting and decrypting textual messages for transmission over an unsecured channel. The algorithm is based on the selection of a prime integer depending upon the size of the message. Then all its 'generators' or primitive roots are identified. The primitive roots are used for generating the elements for calculating the keys. Key generation follows a particular strategy, on a pre-arranged manner. The message constituting the alpha-numeric string of characters is permuted by 1-level railway fence. The cipher text is determined by XORing the ASCII decimal value of the message character with its corresponding key, expressed in ASCII decimal. Selected bits of the cipher text, are flipped. The enciphered text is then transmitted to the receiver. The prime integer and the primitive roots needed for key generation are transmitted to the receiver in a separate message. Upon receiving the cipher-text and the key generation parameters, the receiver reverses the whole sequence of operations to recover the plain-text. The algorithm scrambles the message during transmission through unsecured channels, and also safeguards data stored in the cloud. Applying 3 strategies for encryption/decryption is attempted for the first time and hence no comparison is attempted with prior works with regard to efficiency or versatility. Flipping bits on the date is the new concept introduced here.

Keywords: Prime integer, Primitive root, Discrete logarithm, Key, ASCII.

1. INTRODUCTION

This paper proposes an algorithm for encrypting and decrypting messages containing alpha-numeric characters. It can be used for messages of larger sizes, with a little bit more computational complexity. The strength of the algorithm lies in choosing an appropriate prime modulo value p. Generally it would be proper to choose a prime modulo value p>N, where N is the number of characters in the message to be encrypted/decrypted. Each prime integer gives rise to a series of primitive roots. For example, p=7 has 2 primitive roots. Similarly, p=101 has 40 primitive roots. The meaning and definition of primitive roots is discussed in section 3.

As the message becomes larger, the p value chosen must be larger. For example if we have a message exceeding 1024 characters, then p must be larger than 1024, say, 1031, 1033, 1039, 1049, 1051,1061,1063,1069,1087, 1091,1093, or 1097. These are prime modulo values. Anyone can be chosen and its primitive roots can be used for generating the key values. In this paper the message M consists of N number of alpha-numeric characters. It is subjected to 1-level of railway fence, as the first step towards introducing confusion and diffusion according to Shannon. The generators or primitive roots are used to generate initial elements for key generation.

Using these initial elements, the keys are calculated based on the positional value of the characters in the message. These key values are expressed as a matrix $q^{(z,p-1)}$, where z is the number of primitive roots of the chosen prime modulo p. The matrix is transposed:q^T. Now the position of a character in the plain message is used to determine the row and column to access q^T to read the element to be used for key generation. The details of this scheme are explained in Section 3.Using the

*Corresponding author. Tel.: +919443292499
Email address: mrrajaram15@gmail.com (R.Rajaram)
Double blind peer review under responsibility of DJ Publications
http://dx.doi.org/10.18831/djcse.in/2015011001

key, the single-byte message character (ASCII) is operated upon by the key to evaluate the cipher text. The operator used is exclusive-OR. $C_i=M_i \oplus K_i$. There are several ways of performing the encryption operation. For the purpose of this work $\oplus$ is considered. Thus each byte of the message is $\oplus$-ed with its corresponding key value to determine the cipher values. The cipher text is expressed in binary, and organized into an Nx8 matrix. Thereafter selected bits of the cipher texts are flipped.

The bits to be flipped, is based on the date D of transmission/reception of the cipher text. (D % 8) determines the position of bits to be flipped. Suppose the date is 23, the bit flipped is determined as 23%8=7. Hence every 7th bit is flipped in column-wise fashion. After every 7th bit has been flipped, the modified cipher-text is transmitted to the recipient. The prime modulo value p and the corresponding primitive root values (key generation parameters) are also send to the recipient in a separate message through an unsecured channel. The receiver can then generate the keys, and reverse the sequence of operations applied during encryption to recover the plain-text message. The closest literature that matches this work is [1], where the idea of $\oplus$ ing is adopted for encryption.

2. RELATED WORKS

Encryption and decryption are done by XORing the message byte and the key byte, as in one-time pad [2,3,4,5,6,7,8]. This work has no relationship to one-time pad at all. In [13] the authors use XOR for encryption/decryption. Each byte of plain-text is enciphered with one byte of the key stream. Each key byte is used only once. Hence the key stream must be a truly random. Each key byte can assume any value in the range 1 to 255.

An algorithm called MSA[9] for encryption and decryption of any file using a random key square matrix containing 256 elements was developed. In brute force attack it was necessary to try all the 256! cases to find the actual key matrix. This is quite impossible for the hacker. This is the strength of of MSA.

One-time pad is panacea for practical problems in cryptography. There are other ciphers which can be easily adopted. Moreover the key generation parameters needed for key generation and distribution is less. Public key cryptography overcomes most of the problems associated with one-time pad [10,11]. Many papers suggest that the keys should be generated by some mathematical algorithm, and thereafter ways be devised approximating the properties of a true-random key. Terry Ritter [12] suggests several such methods. However the lacunae in such methods are, even if a portion of the key stream can be found, the opponent can easily reconstruct the entire key stream. Shannon introduces the concept of confusion and diffusion in cryptography in his paper [13].

Vernam Cipher Method for all characters (ASCII code 0-255) with randomized keypad, introduced a feedback mechanism. The method has been closely monitored on different known plain text. This method is almost unbreakable. It allows multiple encryption/decryption [14]. It is an extremely secure block cipher method, applied to encrypt data. The present work makes judicious use of this art of confusion and diffusion in its encryption and decryption exercises. The first step towards attaining this goal is to subject the message to 1-level of railway fence [15, 16]. This itself scrambles the message sufficiently to cause some confusion.

With the rapid technological advances, possibility of tampering information has increased. Cryptography includes symmetric and asymmetric encryption, which avoids tampering attempts and guarantee data integrity[17]. [18] explains how the human need for privacy has manifested itself through cryptography. The accessible style and lucid explanations of complex algorithms penetrate through the mundane mathematical details without oversimplifying it.

This modified version of Generalized Vernam Cipher uses “feedback” effect and also reverses the file while encryption. This makes the encryption process very hard to decrypt by using any brute force method [19]. It was found that the encrypted text has huge difference for similar plain texts having minor difference even for the same text-key. The key generation relies solely on the prime modulo factor and their generators. A tinge of discrete logarithm concept [20, 21, 22, 26, 27] is involved in generating the keys used in the encryption/decryption tasks.

Public-key infrastructure based cryptographic algorithms are considered

slower than symmetric key based algorithms, due to their root in modular arithmetic. In the RSA public-key security algorithm, encryption and decryption is entirely based on modular exponentiation and modular reduction [23, 25, 28], performed on very large integers, typically 1024 bits. Due to this sequential implementation of RSA becomes computationally intensive, and takes lot of time and energy to execute. Moreover, it is very difficult to perform intense modular computations on very large integers, because of the limitation in size of basic data types available with GCC infrastructure.

This paper is organized as follows: Section 1 introduces the broad outline of the work. Section 2 discusses previous works. Section 3 elaborates on the tools and techniques used in various tasks of encryption and decryption. Section 4 discusses the strengths and weaknesses of the proposed algorithm. Section 5 concludes and discusses future scope for expanding the work. Section 6 includes the references. Section 7 gives the appendix where a numerical example is worked out.

3. PROPOSED ALGORITHM

In the present work the authors have used a bit manipulation method which include bit exchange, right shift and XOR operation on the incoming bits [24, 29,30]. To exchange bits the authors used a randomized key matrix $K^{(16\times16)}$. The present method allows multiple encryptions and multiple decryptions. The encryption process is initiated by entering a text-key of 16 characters long. Using these randomization numbers and the encryption number is calculated.

This paper discusses a novel algorithm for encrypting and decrypting a text message of reasonable size.

The plain text message M to be encrypted is expressed as $M=<M_1,M_2, M_3,.....M_N>$ There may be N alpha-numeric and special characters in the message. The strength of scheme relies on choosing a random prime integer p, satisfying p>N in the finite field. The generators or primitive roots of p are determined.

3.1. Definition 1: finite field

For a finite field GF(p), where p is a prime power, has a multiplicative subgroup which is cyclic. The elements g Є GF(p) constitute the subgroup, and are called its primitive elements. Suppose g Є GF(p) is a known primitive element, and u Є GF(p)* = GF(p) – [0], then we can identify the discrete logarithm of u with respect to g for any given integer k, $0\le k\le p-1$ as $u=g^k$ or expressing in terms of logarithm, $k= \log_g u$. Thus k is the discrete logarithm of u and is called the index of u.

3.2. Definition 2: primitive roots

A primitive root modulo n is any integer g with the property that any number co -prime to n is congruent to a power modulo n. For any integer which is co-prime to n, there is an integer k, such that $g^k=u(\text{mod } n)$, where k is the index or discrete logarithm of u to the base g modulo n. Consider $G=<Z_n^*,x>$, contains integers which are relatively prime to n. $G=<Z_p^*,x>$ is a special case where modulus is a prime. Apply the test $g^k=u(\text{mod } n)$, $1\le k\le p-1$. The integers that generate a unique and distinct sequence are declared as primitive roots of p. For example, for p=7, the test generates the following table 1.

Table 1.Table for p=7

g	g^1mod7	g^2mod7	g^3mod7	g^4mod7	g^5mod7	g^6mod7	Status
2	2	4	1	2	4	1	No
3	3	2	6	4	5	1	Yes
4	4	2	1	4	2	1	No
5	5	4	6	2	3	1	Yes
6	6	1	6	1	6	1	No

From table 1 we perceive that 3 and 5 generate a unique and distinct sequence of integers, which represent generators or primitive roots of 7.

Let p have primitive roots of $<g_1,g_2,g_3,.... g_z>$. Now applying $g_i^k \text{mod } p$, calculate the unique non-repetitive sequence, for all primitive roots, for $1\le i\le z$; $1\le k\le(p-1)$. The values are put in a matrix $q^{(z,p-1)}$. This matrix contains the elements to generate the key stream.

$$q = \begin{matrix} g_1^1 & g_1^2 & g_1^3 & \ldots\ldots & g_1^{p-1} \ (\text{mod } p) \\ g_2^1 & g_2^2 & g_2^3 & \ldots\ldots & g_2^{p-1} \ (\text{mod } p) \\ \ldots & \ldots & \ldots & \ldots\ldots & \ldots \ (\text{mod } p) \\ g_z^1 & g_z^2 & g_z^3 & \ldots\ldots & g_z^{p-1} \ (\text{mod } p) \end{matrix}$$

The q matrix is then transposed.

$$q^T = \begin{matrix} g_1^1 & g_2^1 & \dots & g_z^1 & (\text{mod } p) \\ g_1^2 & g_2^2 & \dots & g_z^2 & (\text{mod } p) \\ \dots & \dots & \dots & & (\text{mod } p) \\ g_1^{p-1} & g_2^{p-1} & \dots & g_z^{p-1} & (\text{mod } p) \end{matrix}$$

The transposed q^T matrix constitute the elements for generating the key for encryption.

Meanwhile consider the message: $M=<M_1,M_2,M_3, \dots M_N>$. The message array is subjected to 1-level of railway fence as shown

Level 0: M_1 M_3 M_5 M_{N-1}
Level 1: M_2 M_4 . M_N

Modified Message is M1,M3,M5,...Mn-1,M2,M4,...MN

Next determine the key for each character of the modified message array. Using row (r) and column (c) of matrix q^T, read off the element for evaluating the key, using the positional value of each character in M array.

$r=P \% (p-1) +1$ $c=P \% z+1$

Access $q^T_{r,c} = a$, where a is the element of matrix q^T at the intersection of r and c. $K_P=a^P \bmod 256$, where K_P is the key corresponding to the message character at position P in the message array.

Therefore $C_P= M_P \oplus K_P$. Both the M_P and K_P are represented by their ASCII decimal values. These steps are repeated for all characters in the message array. The cipher text is arranged in a matrix $\mathbf{C}^{Nx8}$, where each row represents the binary equivalent of C_P, expressed in ASCII decimal notation.

One more step towards introducing confusion and diffusion in the cipher matrix C remains. This is to flip select bits of the C matrix. The bits to be flipped are based on the date D of transmission/reception of the encrypted message. If the date D=28, we reduce it by applying modulus 8. 28 % 8 =4. This implies every 4th bit in the C matrix is flipped column-wise. This is shown in matrix C' below. F indicates the bits flipped. Leaving the first bit C_{11} intact, every 4th bit down the column 1 are flipped. From the last flipped bit in column 1, the next 4th bit will figure in column 2. Thus continuing this exercise, we find that every column will have some bits flipped.

$$C' = \begin{matrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} & C_{17} & C_{18} \rightarrow C_1 \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} & C_{27} & C_{28} \rightarrow C_2 \\ C_{31} & C_{32} & C_{33} & .. & .. & .. & .. & C_{38} \rightarrow C_3 \\ F & C_{42} & C_{43} & .. & .. & .. & .. & C_{48} \rightarrow C_4 \\ C_{51} & .. & .. & .. & .. & .. & .. & .. \\ C_{61} & .. & .. & .. & .. & .. & .. & .. \\ C_{71} & .. & .. & .. & .. & .. & .. & .. \\ F & .. & .. & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. & .. & .. & .. \\ C_{N1} & .. & .. & .. & .. & .. & .. & C_{N8} \rightarrow C_N \end{matrix}$$

Now each row of the matrix C' constitute one character of the cipher text. All the N cipher text characters in this matrix are transmitted to the recipient. $C'=<C_1,C_2,C_3, \dots C_N>$ Now the recipient works to reverse of the encryption process to recover the plain text message. For the benefit of the reader, let us summarize the steps leading to encryption and decryption.

1. Select prime modulo p, satisfying P>N, where N is number of characters in message.
2. Determine all generators g of p according to definition 2.
3. For each generator or primitive root g, generate the unique non-repetitive sequence. Organize these values into the matrix $q^{z \times (p-1)}$
4. Determine the transpose of q, q^T.
5. Define the message array, $M=<M_1,M_2, \dots M_N>$
6. Apply 1-level railway fence to M. Thus $M' = <M_1,M_3....M_{N-1}, M_2,M_4,.....M_N>$
7. Apply positional value of M_i to determine row and column to access matrix q^T
8. $q^T_{r,c} = a$. $K_P=a^P \bmod 256$
9. $C_P = M_P \oplus K_P$ Organize C matrix of size Nx8, each row representing a binary string.
10. Apply D % 8, to determine bits to be flipped column wise of the C matrix.
11. A —>B: $C=<C_1,C_2, \dots C_N>$; Elements of C matrix transmitted to recipient.
12. A —>B: p,g; Prime modulo p and its generators transmitted.
13. Recipient restores flipped bits using D % 8.
14. Using p, g values, recipient generates q.
15. Transposes q to yield qT matrix.
16. Accesses $q^T_{r,c}$ to find a. $K_P= a^P \bmod$

256

17. $M_P = C_P \oplus K_P$ This yields $M' = <M_1, M_3, \ldots M_{N-1}, M_2, M_4 \ldots M_N>$
18. Rearrange M' to yield $M = <M_1, M_2, M_3 \ldots\ldots M_{N-1}, M_N>$

4. DISCUSSION ON SECURITY OF THE PROPOSED METHOD

In normal one-time pad there are as many keys as there are characters in the message. Both the cipher-text and the key values have to be transmitted to the receiver whereas in this paper, only the prime modulo p and its generators or primitive roots g_i (key generation parameters) are transmitted, by a separate message through an unsecured channel. The key generation relies on the prime integer p and its primitive roots g_i. g_i is used to generate the key values in the range $1 \le i \le p-1$. It can be perceived that the discrete logarithm is the discrete exponentiation in the finite cyclic group. The $q^{\hat{a}}$ is evaluated as q^x=q,q,q.....q,q multiplied x times. In this paper it is calculated in discrete exponentiation within a group. The calculation is done fast entailing only O(log x) operations, by using fast exponentiation technique. If the prime number chosen is high, p>>N, then the opponent faces the difficult task of determining its primitive root values. He may have to requisition the use of high power computation machines.

4.1. On line guessing attack

This is made difficult as p is chosen so that p>N. There may be a large number of primes above N. Moreover each prime p has a long list of primitive roots. The attacker has to first decide on p and then determine their primitive roots to generate the keys. Even at an elementary level there are 55 prime values between 1 and 255. By a rough estimate the total primitive roots for these 55 primes work out to 1952. Thus the attacking probability is 1:1952 in the range 1 to 255. Thus the attacking probability is 1/1952 or 0.0005123. You can well imagine that if the range exceeds 255, the attacking probability becomes still lower.

4.2. Man-in-the-middle attack

The attacker captures the cipher-text and makes an attempt at guessing the key value. As you have seen guessing the key value is almost impossible, the attacker is able to monitor one or two links in the transmission path whereas in the public channel, embedded cipher blocks are routed through different links. The attacker may miss some of the vital blocks, which pass through other links. He may be successful in guessing the key value, by monitoring all links and gathering all the cipher blocks. This is impossible, and therefore man-in-the-middle attack is infeasible in our method.

Flipping of bits in cipher-text is based on date D % 8. Assuming that the opponent is able to guess D, he keeps wondering if flipping is done column-wise and row-wise.

Applying rail fence is another strategy to confuse the opponent. There are several levels which can be applied. For sake of simplicity, the authors use 1-level railway fence. There are other levels such as 2-level, 3-level and so on.

5. CONCLUSION & FUTURE SCOPE

This is a first time work where maximum exploitation of Claude Shannon's technique of confusion and diffusion is attempted for encryption and decryption. The scheme will be ideal for transmitting messages in secret at least interms of cost and minimum channel utilization. In one-time pad, the cipher-text and equal count of key material will have to be transmitted. In this scheme only the N number of cipher characters is send. The key generations parameters, namely prime modulo value p and its generators g_i are send in another message through a public unsecured channel. Here again the packets take different routes and the opponent will not be able to gather the entire set of cipher-text or the key material. The method proposed withstands several major attacks and is very secure for hiding transmitted cipher-text. The bit flipping can be based on other parameters such as receiving time, instead of the date D. One can do flipping on a pair of bits row wise rather than one bit column wise at a time. Higher level of railway fencing can be done. The q matrix containing the unique non repetitive sequence can be permuted as done in DES. The prime modulo value p can be chosen to be p>>N, instead of just p>N. This will give rise to more random choices of p and larger primitive roots. This is a pioneering work. Concept such as bit flipping is proposed for the first time. Therefore no comparison is done with existing works.

ACKNOWLEDGEMENTS

The work was taken up under the auspices of the CSE research centre of Vickram college of Engineering. The authors express their gratitude to Mr.S.Ayyanar for organizing the paper in its present form:

REFERENCES

[1] R.Rajaram, A new algorithm for encryption/decryption, Elsevier, Computers Standards and Interfaces, 2009, Vol. 31, No. 6, pp. 1069-1072, http://dx.doi.org/10.1016/j.csi.2008.09.037.

[2] Frank Rubin, One-time pad cryptography, Cryptologia, Vol. 20, No.4, 1997, pp. 359-364 http://dx.doi.org/10.1080/0161-11961885040.

[3] Y Dodi and J Spencer, On the (non) Universality of the One Time Pad, Poceedings of the 43rd Symposium on Foundations in Computer Science, Canada, 2002, pp. 376-385, http://dx.doi.org/10.1109/SFCS.2002.1181962.

[4] D.Raub, R.Steinwandt and J.Mueller Quade, On the Security and Composability of the One Time Pad, 31st Conference on Current Trends in Theory and Practice of Computer Science Liptovsky Jan, Slovakia, 2005, pp. 288-297, http://dx.doi.org/10.1007/978-3-540-30577-4_32.

[5] One Time Pad Tutorial, May 29, 2006.

[6] One time pad (Vernam's cipher), FAQ, 2006-05-12.

[7] Vivek Vaidya, Nithin Nagaraj, Prabhaker G.Vaidya, Re-visiting the One Time Pad, International Journal of Network Security, Vol. 6, No. 1, 2008, pp. 94-102.

[8] Bruce Schneier: Arguments Against One Time Pad

[9] D.Chatterjee, J.Nath, S.Dasgupta and A.Nath, A New Symmetric Key Cryptography Algorithm Using Extended MSA Method: DJSA Symmetric Key Algorithm, IEEE International Conference on CSNT, Jammu, 2011, pp.89–94, http://dx.doi.org/10.1109/CSNT.2011.25.

[10] Ralph Erskine Enigma's Security: What the Germans Really Know in Action in this Day, 2001, pp. 370-386.

[11] Robert Wallace and H.Keith Melton, Henry R.Schlesinger, Spycraft: The Secret History of the CIA's Spytechs, from Communism to Al-Qaeda, New York, Dutton, 2008.

[12] D.Coppersmith, Fast Evaluation of Logarithms in Fields of Characteristic Two, IEEE Transactions on Information Theory, Vol. 30, No. 4, 1984, pp. 587-594, http://dx.doi.org/10.1109/TIT.1984.1056941.

[13] Claude Shannon, Communications theory of secrecy systems, Bell System Technical Journal, Vol. 28, No. 4, 1949, pp. 656-715, http://dx.doi.org/10.1002/j.1538-7305.1949.tb00928.x.

[14] T.Chatterjee, T.Das, S.Dey, A. Nath and J.Nath, Symmetric Key Cryptosystem Using Combined Cryptographic Algorithms - Generalized Modified Vernam Cipher Method, MSA Method and NJJSAA Method: TTJSA Algorithm, IEEE International Conference on WICT, Mumbai, India, 2011, pp. 1175 – 1180, http://dx.doi.org/10.1109/WICT.2011.6141415.

[15] Devharsh Trivedi, Write a C program for railway fence, February 2012.

[16] James Lyons, Railway fence cipher, 2009-2012.

[17] Al Ahmad, Mohammad, Alshaikhli, Imad , Jumaah and Bashayer, Protection of the Digital Holy Quran Hash Digest by Using Cryptography Algorithms, IEEE International Conference on ACSAT, Malaysia, 2013, pp. 244–249, http://dx.doi.org/10.1109/ACSAT.2013.55.

[18] Singh Simon, The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography, New York, 2000.

[19] D.Das, Megholova Mukherjee, Neha Choudhary, Asoke Nath and Joyshree Nath, An Integrated Symmetric Key Cryptography Algorithm Using Generalised Modified Vernam Cipher method and DJSA Method: DJMNA Symmetric Key Algorithm, IEEE International Conference on WICT, Mumbai, 2011,

pp. 1199-1204, http://dx.doi.org/10.1109/WICT.2011.6141419.

[20] A.M.Odlyzko, Discrete Logarithms in Finite Fields and their Cryptographic Significance, Proceedings of EUROCRTPT 84 A Workshop on the Theory and Application of Cryptographic Techniques Paris, France, 1985, pp. 224-314, http://dx.doi.org/10.1007/3-54039757-4_20.

[21] Marcus Nilsson, Discrete logarithms in Cryptography, 2010.

[22] Don Coppersmith, Evaluating Logarithms in GF(2n), Proc 16th ACM Symposium on Theory of Computing, USA, 1984, pp. 201-207, http://dx.doi.org/10.1145/800057.808682.

[23] S.Saxena and B.Kapoor, An Efficient Parallel Algorithm for Secured Data Communications Using RSA Public Key Cryptography Method, IEEE International Conference on IACC, Gurgaon, India, 2014, pp. 850-854, http://dx.doi.org/10.1109/IAdCC.2014.6779433.

[24] N.Khanna, J.Nath, J.James, S.Chakraborty, A.Chakrabarti and A.Nath, New Symmetric Key Cryptographic Algorithm Using Combined Bit Manipulation and MSA Encryption Algorithm: NJJSAA Symmetric Key Algorithm, IEEE International Conference on CSNT, 2011, Jammu, India, pp. 125-130, http://dx.doi.org/10.1109/CSNT.2011.33.

[25] Y.Alkady, M.I.Habib and R.Y.Rizk, A New Security Protocol Using Hybrid Cryptography Algorithms, IEEE 9th International Conference on ICENCO, Giza, Egypt, 2013, pp. 109-115, http://dx.doi.org/10.1109/ICENCO.2013.6736485.

[26] S.N.Molotkov, Quantum Cryptography and V A Kotel'nikov's One Time Key and Sampling Theorems, PHYS-USP, Vol. 49, No.7, 2006, pp. 750-761, http://dx.doi.org/10.1070/PU2006v049n07ABEH006050.

[27] Frederik, Vercauteren, Discrete Logarithms in Cryptography, ESAT/COSIC - K.U. Leuven, ECRYPT Summer School location, 2008, Nederland.

[28] R.Rajaram, Network Security and Cryptography, Book, Scitech Publications (I) Pvt Ltd, Chennai, India, 2014.

[29] David Kahn, The Code Breakers, New York, 1930.

[30] Ritter and Terry, The Efficient Generation of Cryptographic Confusion Sequences, Crytologia, Vol. 15, No. 2, 1991, pp. 81-139, http://dx.doi.org/10.1080/0161-119191865812.

APPENDIX A

Numerical Example

M=HOW ARE YOU? There are 12 characters in the message. Select p=13.
Determine the primitive roots based on definition

$2^1 \bmod 13$	$2^2 \bmod 132$	$2^3 \bmod 13$	.....	$2^{N-1} \bmod 13$
$3^1 \bmod 13$	$3^2 \bmod 13$	$3^3 \bmod 13$	.....	$3^{N-1} \bmod 13$
..	..	..	.....	..
$12^1 \bmod 13$				$12^{N-1} \bmod 13$

Integers 2,6,7 & 11 alone generate the unique and non-repetitive sequence to qualify as primitive roots of 13. Using these values calculate matrix q of size 4 x 12

$$q = \begin{matrix} 2 & 4 & 8 & 3 & 6 & 12 & 11 & 9 & 5 & 10 & 7 & 1 \\ 6 & 10 & 8 & 9 & 2 & 12 & 7 & 3 & 5 & 4 & 11 & 1 \\ 7 & 10 & 5 & 9 & 11 & 12 & 6 & 3 & 8 & 4 & 2 & 1 \\ 11 & 4 & 5 & 3 & 7 & 12 & 2 & 9 & 8 & 10 & 6 & 1 \end{matrix}$$

$$q^T = \begin{matrix} 2 & 4 & 7 & 11 \\ 4 & 10 & 10 & 4 \\ 8 & 8 & 5 & 5 \\ 3 & 9 & 9 & 3 \\ 6 & 2 & 11 & 7 \\ 12 & 12 & 12 & 12 \\ 11 & 7 & 6 & 2 \\ 9 & 3 & 3 & 9 \\ 5 & 5 & 8 & 8 \\ 10 & 4 & 4 & 10 \\ 7 & 11 & 2 & 6 \\ 1 & 1 & 1 & 1 \end{matrix}$$

Apply 1-level railway fence to M.
HWAEYUObRbO? where b=blank.

Char	Pos(P)	R=P%12+1	C=P%4+1	$a=q^{T}_{R,C}$	$K_i=a^{P} \bmod 256$	$C_i=M_i?K_i$
H=72	1	2	2	10	10	72?10=66
W=87	2	3	3	5	25	87?25=78
A=65	3	4	4	3	9	65?09=72
E=69	4	5	1	6	16	69?16=85
Y=89	5	6	2	12	0	89?0=89
U=85	6	7	3	6	64	85?64=21
O=79	7	8	4	9	121	79?121=54
b=32	8	9	1	5	225	32?225=193
R=82	9	10	2	4	0	82?0=82
b=32	10	11	3	2	0	32?0=32
O=79	11	12	4	1	1	79?1=78
?=63	12	1	1	2	0	63?0=63

C=

0	1	0	0	0	0	1	0 = 66
0	1	0	0	1	1	1	0 = 78
0	1	0	0	1	0	0	0 = 72
0	1	0	1	0	1	0	1 = 85
0	1	0	1	0	1	0	1 = 89
0	0	0	1	0	1	0	1 = 21
0	0	1	1	0	1	1	0 = 54
1	1	0	0	0	0	0	1 = 193
0	1	0	1	0	0	1	0 = 82
0	0	1	0	0	0	0	0 = 32
0	1	0	0	1	1	1	0 = 78
0	0	1	1	1	1	1	1 = 63

D=23%8=7 Leaving C_{11} intact, every 7th bits are flipped column-wise.

$$
C' = \begin{matrix}
0 & 1 & 0 & 0 & F & 0 & 1 & 0 = 74 \\
0 & F & 0 & 0 & 1 & 1 & 1 & 0 = 14 \\
0 & 1 & 0 & 0 & 1 & F & 0 & 0 = 76 \\
0 & 1 & F & 1 & 0 & 1 & 0 & 1 = 117 \\
0 & 1 & 0 & 1 & 0 & 1 & F & 1 = 87 \\
0 & 0 & 0 & F & 0 & 1 & 0 & 1 = 5 \\
F & 0 & 1 & 1 & 0 & 1 & 1 & F = 183 \\
1 & 1 & 0 & 0 & F & 0 & 0 & 1 = 201 \\
0 & F & 0 & 1 & 0 & 0 & 1 & 0 = 18 \\
0 & 0 & 1 & 0 & 0 & F & 0 & 0 = 36 \\
0 & 1 & F & 0 & 1 & 1 & 1 & 0 = 110 \\
0 & 0 & 1 & 1 & 1 & 1 & F & 1 = 61
\end{matrix}
$$

A—>B:
{74,14,76,117,87,5,183,201,18,36,110,61}

A—>B: 13, 2,6,7,11

That completes steps 11, 12 of our algorithm. The recipient is expected to perform steps 13 to 18 and recover the original message.