A CRYPTOSYSTEMS ALGORITHM USING SYSTEMS OF NON-LINEAR EQUATIONS

S. E. ADEWUMI, Ph. D.
Management Information Systems (MIS)
University of Jos, Nigeria
email: adewumis@unijos.edu.ng

E. J. D. GARBA, Ph. D.
Mathematics Programme
Abubakar Tafawa Balewa University,
Bauchi, Nigeria

Abstract - Encryption is an important tool for managing and protecting information. Cryptography is the only practical means of sending information over an insecure channel. These channels may be computers, telephone lines, satellite, etc. The Increasing use of electronic means of data transfer from one point to another, coupled with the growth in networking and Internet communication, has extended the need to protect vital information- military, banking, academics (question papers), databases and the e-initiative. This paper proposes a new radical method for a secured means of encrypting and decrypting messages. The method uses data compression and systems of non-linear equations. With this, we hope that data transmission across networks will be more secured. This new proposal is in recognition of the peculiar nature of our environment where people want to break codes/messages not intended for them. The final stage is to turn the text message to a system of non-linear equations, which can be solved only when the text message gets to its intended destination.

Keywords - Cryptosystems Algorithm, Non-Linear Equations.

INTRODUCTION

Controlling and safeguarding the flow of information in this era, best called the era of Information and Communication Technology, is as important as controlling the flow of water in a desert. Encryption is an important tool for managing and protecting information. Its intention is to make it difficult for cryptanalyst to observe, corrupt or falsify messages in an unsecured media [1].

Cryptosystem defines all the elements that are involved with the provision of secured communication between any two points. The process involves the sender sending a plaintext, the plaintext is transformed into a form not readable or in a disguised form (called a Ciphertext). The process of transforming a plaintext to Ciphertext is called *encryption* or simply *enciphering*. When this Ciphertext arrives at its intended destination, it is transformed from the Ciphertext to the Plaintext (called *decryption* or *deciphering*). For the receiver to be able to read the message and, at the same time, avoid the plaintext being accessed by an unauthorized individual, the sender must transform the plaintext into Ciphertext using some specific parameter. This parameter is called the encryption Key K_e. The receiver, then, deciphers (interprets the Ciphertext) using the decryption Key K_d. In a public Key cryptosystem, K_e is made public but K_d is not. It must be kept secret, known only to the receiver. The key K for cipher and deciphering should be common to the sender and receiver.

In encrypting, the information is being transmitted and the encrypting key is fed into an encryption algorithm. When the encrypted data gets to the receiver, it is passed through a decryption algorithm so as to enable the reader to have access to the transmitted data in the plaintext.

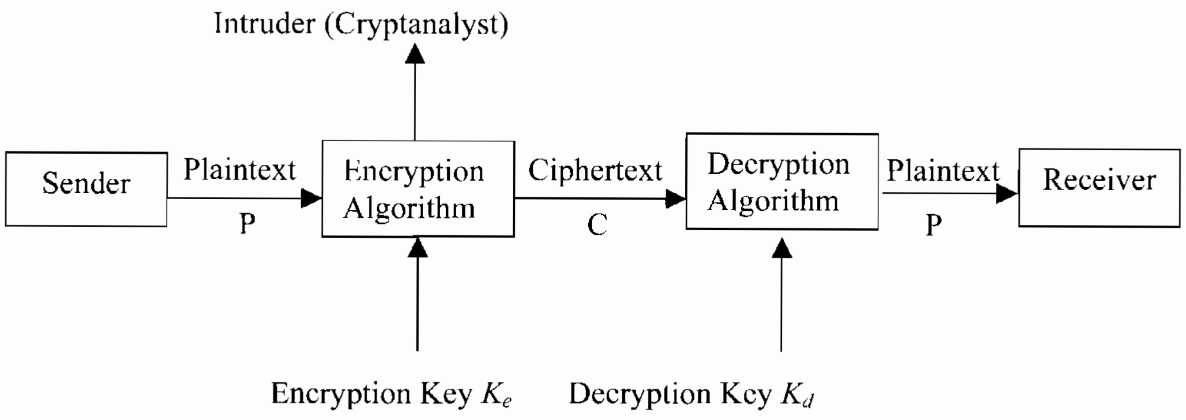

Figure 1: The Encryption Model

Figure 1 shows the encryption model for secured data transmission. It is assumed that during the transmission, the intruder (the enemy) hears and copies the Ciphertext, but he does not know what the decryption key is and, thus, fails to decipher the message.

The art of breaking Ciphers by the intruder (cryptanalyst) is called Cryptanalysis; while the art of devising Ciphers (cryptography) and breaking them (Cryptanalysis) is collectively known as Cryptology.

A state of confusion should always be created for the Cryptanalyst whose intention is to break in and have access to the data being transmitted. The Chinese proverb that says ***"it is sometimes good to be unclear"***, readily applies.

Many electronic frauds common in marketplace can be checked by encryption technique.

LITERATURE REVIEW

In the last few decades of their existence, computer networks were primarily used by university researchers to transfer information among Academia, and in corporate organizations to share printers, scanners, and other sharable resources in a distributed environment. Under those conditions, security was not on the priority since such shared information and resources did not pose any security risk to the sender or the receiver. Nowadays, the use of computer has gone beyond the early intentions. They are now used in the e-initiative (e-banking, e-commerce, e-shopping), etc. These phenomenal changes have brought about the need for tight security to data and information as they are transported from network to network.

In the last five decades, computer's processing speed has doubled every 1½ to 2½ years. It, therefore, shows that a code that took a thousand years to break with the computers available in 1960 would take a year to break with the computers available in 1980, but will take few hours to break in 2001 [1]. This calls for security consciousness on the part of users of computer systems.

Security is a broad topic that ranks almost first in a computer networked environment. In its simplest form, it is concerned with making sure that *busy bodies* cannot read, or worse still modify messages intended for other recipients. It is as a result of this concern for people trying to access remote services that they are not authorized to use, that the need for this work arises. Security also deals with the problems of modification on relayed messages and denial by people who want to claim that they have never sent certain messages. Most security problems are intentionally malicious by people who want to gain some benefit or cause harm [2].

If we represent the plaintext by the letter **P**, **C** as the Cipherxt, E_k as the encryption Key and D_k as the decryption key, then $C=E_k(P)$, meaning that the encryption of plaintext **P** using key *K* gives the ciphertext *C*. In the same manner, $\mathbf{P}=D_k(C)$ represents the decryption of **C** using key ***K*** to get back the plaintext **P**. In a more general form, it can be represented as $D_k(E_k(P))=P$.

THE RSA ALGORITHM

Rivest et al [3] at M.I.T. discovered an algorithm known as RSA (an acronym from Rivest, Shamir, Adleman). Their method was based on the principles of number theory. It is now summarised below:

i. Choose two large primes say p and q, (these numbers should not be less than (10^{100}).
ii. Compute $n = p \times q$ and $z = (p-1)\times(q-1)$.
iii. Choose a number relatively prime to z and call it d.
iv. Find e such that $e \times d = 1 \bmod z$.

These parameters are computed in advance. The encryption begins by dividing the plaintext into blocks so that each plaintext P, falls into the interval $0 \le P < n$. Encrypting message P, compute $C=P^e(\bmod n)$. Compute $P=C^d(\bmod n)$. To encrypt, the parameters e and n are required, while to decrypt, d and n are required. This implies that the public key consists of the pair (e,n) while the private key is made of the pair (d,n). The inability to factorize large numbers makes the system secure.

If the public known n can be factorized, then it is easy to reconstruct p and q, which will eventually lead to the formation of z. Once z and e are known, d can be found.

Rivest and colleagues have observed that factoring a 200-digit number requires 4 billion years as at 1978, this may be less as a result of increase in computer speed. Factoring a 500-digit will require about 10^{25} years.

As an example, we take p=3 and q=11 giving n=33 and z=20. A suitable value for d is 7, since 7 and 20 have no common factor. Therefore, e can be found by solving the equation 7e = 1 (mod 20), this will yield e=3.

Plaintext (P)		Ciphertext (C)				
Symbolic	Numeric	P^3	P^3 (mod 33)	C^3	C^7 (mod 33)	Symbolic
A	1	1	1	1	1	A
T	20	8000	14	105413504	20	T
T	20	8000	14	105413504	20	T
A	1	1	1	1	1	A
C	3	27	27	10460353203	3	C
K	11	1331	11	19487171	11	K

Figure 2: An example of RSA

RSA is disadvantageous in that it is slow when large volumes of data are to be encrypted.

TRANSPOSITION CIPHER (BLOCK CIPHER)

In this method, the plaintext is divided into blocks and, in turn, each block is enciphered independently. Under the control of a fixed key, different occurrences of a particular plaintext block will always be encrypted as the same ciphertext block. Figure 6 is an example of transposition cipher.

The general approach is that, the cipher is keyed by a word or phrase not containing any repeated letters. For example, if we take BAUCHI as the key, the entire columns can, then, be numbered according to their position in the alphabet. The A in BAUCHI will be 1 because A is the first (number 1) letter of the alphabet; B = 2 because B is the second (number 2) letter of the alphabet and so on. With this in mind, the plaintext will be written horizontally in rows. The ciphertext will, then, be read column by column starting, of course, from the column whose key letter is the lowest.

B	A	U	C	H	I
2	1	6	3	4	5
T	R	A	N	S	E
E	R	O	N	E	M
I	L	L	I	O	N
N	A	I	R	A	T
O	A	C	C	O	U
N	T	F	I	F	T
E	E	N	T	W	E

L	V	E	F	R	O
M	S	I	X	F	I
V	E	S	E	V	E
N	A	B	C	D	E

Plaintext:
transferonemillionnairatoaccountfifteentwelvefromsixfiveseven
Ciphertext:
RRLAATEVSEATEINONELMVNNNIRCITFXECSEOAOFWRFVDEMNTUTEOIE EAOLICFNEISB

Figure 3: A Sample Transposition Cipher

For a cryptanalyst to break a block cipher, he must first know that he is dealing with transposition cipher. One method for a crytanalyst is by observing the frequency of E, T, A, O, I, N to see if they fit the normal pattern of the plaintext. If they do, it is then deduced that the cipher is a transposition, reason being that in this kind of cipher, each letter represents itself.

Once this is established, the next ordinary thing to do is to guess the number of columns involved. This can be achieved, if, for example, the cryptanalyst suspects that a word like illionnaira occurs somewhere in the message. He can, then, observe that diagrams like IN, LA, LI, IR, OA will occur in the ciphertext as a result of the phrase wrapping round. If a key length of five was used, the digrams would have been LN, LA, II, OR, NA. Each key length will produce a different digrams from the ciphertext. With persistent hunting for the key length, the cryptanalyst may be able to determine the key length and thereby recover the plaintext from the ciphertext.

This cipher shown in Figure 3 produces a fixed length block of input and produces a fixed-length block of output. The cipher in Figure 3 can be seen as a 66-character-block cipher, with the following output 2, 8, 14, 20, 26, 32, 38, 44, 50, 56, 62, 1, 7, 13, 19, 25… 62. This is to say that the second input R, is the first to be the output followed by the eight, R, and so on.

PUBLIC KEY ALGORITHMS

In cryptosystems, key distribution has always been the main problem. However strong a cryptosystem is, once an intruder steals the secret key, the system is worthless.

The kind of cryptosystem used on the computer networks is called a *symmetric key system*. With this approach, the sender and the receiver use the same key, and they have to keep their shared key secret from other people. The biggest problem in this scheme is the shared key management.

In this method, if you want to communicate with several people and ensure that each person can read messages intended for others, then different secret keys are needed for each person. In the 1970s, Cryptographers developed *Public key Cryptography* in order for them to get around the problem of managing keys. Under this scheme, each person has two keys, Private key (secret) and Public key (freely available to any one who cares to know). In this case, the public key system is asymmetric – different keys are used for encryption and decryption [4].

In 1976, at Stanford University, a radical new kind of cryptosystem was proposed, one in which the encryption and decryption keys were different and the decryption key could not be derived from the encryption key. In their proposal, the (keyed) encryption algorithm, E, and the (keyed) decryption algorithm, D, had to meet the following requirements:

i. D(E(P)) = P.
ii. All (D, E) pains are distinct.
iii. It is exceedingly difficult to deduce D from E
iv. E cannot be broken by a chosen plaintext attack.

METHODOLOGY

Our proposal is a new radical means of sending messages through insecure channels. This method uses data compression algorithm designed by us. Complete encryption is done through an approach being proposed here which uses systems of non-linear equations. With these two algorithms in place, we hope that texts encrypted using this method will be difficult to crack as it has been done with most other available cryptographic systems.

The texts to be transmitted are compressed and the words that do not repeat are then turned into systems of non-linear equations before they are transmitted.

The plaintext is converted to systems of non-linear equations as ciphertext during encryption. When the receiver receives the ciphertext (encrypted text), solving the systems of non-linear equations decrypts it. Once the variables representing the characters in the text have all been recovered, the next step is to determine the meaning of each variable. To do this, the value of each variable $x_1, x_2, \cdots, x_n$ is manipulated with delta encoded values as shown in the examples. This is achieved by placing the recovered variables in their positions as represented in the delta encoded values transmitted with the equation. To further disguise the characters of the text, one single variable can be used to represent several characters at the same time. To distinguish them, we recover each character of each word from their distance from the origin as specified by the delta-encoding algorithm transmitted with the equation.

In formulating the equation, the alphabets are initialized; this initialization can be generated based on a criterion imposed by the number of word/character in the text to be deciphered.

The initialization is, thus, generated randomly with the formula $x_i = x_0 \mod k + j$, where i can take the place of 1,2,3…,n and X_0 called the SEED representing the number f letters in the text, k is the number of words in the text and j = X_0 *k. When this is done, the alphabets and the variables are not altered in any way. Xi is the origin, the starting value.

APPLICATION OF SYSTEMS OF NON-LINEAR EQUATIONS TO ENCRYPTION

The use of systems of non-linear equations provides another level security, as solving of non-linear equations is not as easy as solving systems of linear equations with which many are familiar.

Solving a system of non-linear equations is a problem that is avoided when possible. Non-linear equations describing the physical world may be algebraic, ordinary or partial differential.

The purpose of this section is to describe the methods available for solving this kind of problem and, then, applying them in our current work.

Consider the non-linear system of equations of the form:

$$f_1(x_1, x_2, \cdots, x_n) = 0$$
$$f_2(x_1, x_2, \cdots, x_n) = 0$$

$$\ldots$$

$$f_n(x_1, x_2, \cdots, x_n) = 0 \qquad (1.1)$$

With $x = (x_1, x_n, \cdots, x_n)'$ the initial guess.
Let $a = (a_1, a_2, \cdots, a_n)t$ be a solution.
Then $f_i(a) = 0, i = 1,2,\ldots,n$.

Applying Taylor's theorem for equation 1.1, expanding f(x) about x_k:

$$f_i(x) = f_i(x_k) + \sum_{j=1}^{n} (x_j - x_{j,k}) \frac{\partial f_i(x_k)}{\partial x_j} + o(2nd) \qquad for\ i = 1,2,\ldots,n$$

By letting x=a, using f(a) = 0 and dropping the second and higher order terms, we obtain,

$$0 = f_i(a \approx f_i(x_k) + \sum_{j-1}^{n} (a_j - x_{j,k}) \frac{\partial f_i(x_k)}{\partial x_j} + o(2nd) \qquad for\ i = 1,2,\ldots,n$$

or in matrix form:

$$f\ (x_k) + J(x_k)(a - x_k)$$

Where $f\ (x_k) = [f_1(x_k), f_2(x_k), f_3(x_k), \ldots f_n(x_k)]'$

$$J(x_k) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & & & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \qquad \text{Jacobian Matrix}$$

Newton's method for systems of non-linear equations replaces the derivative in the single variable case with *nxn* Jacobian matrix.

The method of **Steepest Descent** is generally used to find a sufficient accurate starting approximation for this kind of method. This method is similar to the bisection method use for the solution of a single equation.

The iteration formula for solving systems of linear equation can be written as
$J(x_k)(x_{k+1} - x_k) = -f(x_k)$ provided $|J| \neq 0$

If we substitute x_{k+1}-x_k by δ_k, the formula can, then, be written as $J(x_k)\delta_k = -f(x_k)$, where δ_k is a new improvement on the initial starting value of δ_0.

For systems of n simultaneous non-linear equations, n simultaneous linearized equations are solved in every iteration.

In this method, the systems are solved without finding the inverse of $J(x_k)$. The formula can be written as $x_{k+1}= x_k-f(x_k)/J(x_k)$, that is, $x_{k+1}= x_k-f(x_k)J(x_k)^{-1}$. The $J(x_k)^{-1}$ is avoided as this will involve additional iteration for determining the inverse of $J(x_k)$

GAUSS ELIMINATION ALGORITHM

The Gauss elimination algorithm generates a sequence of matrices $\{A^{(k)}\}$ (k = 1... N), where $A^{(1)} \equiv A$ and a sequence of right hand side $b^{(k)}$ such that $A^{(N)} = U$ being the upper triangular.

The first derived system $A^{(2)} = b^{(2)}$ is obtained by eliminating the subdiagonal components in the first column by subtracting suitables of the first row from each of the subsequent rows as shown below:

$$A^{(2)}= \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1N}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2N}^{(2)} \\ \vdots & & & \\ 0 & a_{N2}^{(2)} & \cdots & a_{NN}^{(2)} \end{bmatrix}$$

Where assuming

$$a_{11}^{(1)} \neq 0,$$
$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1} a_{1j}^{(1)}, \quad i,j = 2,\cdots,N$$
and
$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}$$

In general for
$$k = 1,2,\ldots,N-1, if\ a_{kk}^{k} \neq 0$$
$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad i,j = k+1,\ldots,N$$
with
$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$$

BACKWARD SUBSTITUTION

When a matrix has been reduced by the method enumerated above, into its upper triangular, it is possible to solve the components of x in the reverse order using the kth equation to solve for the kth component x_k.

$$x_N = b_N / a_{NN}$$
similarly

It follows that $x_k = (b_k - \sum_{j=k+1}^{N} a_{kj} x_j) / a_{kk}, \quad k = N-1,\ldots,1$

DELTA (δ) ENCODING

This technique is used to denote the change in a variable or code. It stores data as the difference between successive samples (or characters). With this method, data are not stored directly. The first value in the delta encoded is always the same as that in the original text. The delta represents the difference (delta) between the corresponding values. For example, if we have the following:

Original Data: 21 24 5 7 20 5 30…

Delta Encoded: 21 3 –19 2 13 –15 25…

We can now say that in order to recover the original text from the delta encoding, we can use the formula

$$C_n = \sum_{i=1}^{n} \delta_i$$

where $\delta_i = C_n - C_{n-1}$ and C_n represents the code at the point *n* of the original data stream. This method can be applied to encode the distance of each of the variables from the origin. This delta encoding will form part of the equation to be transmitted. The beauty of this algorithm is that a singular variable can be made to represent several other characters at the same time, rather than representing just one character per variable.

FORMULATING AN EQUATION

To formulate an equation, the following algorithm is used:

1. Count the number of words, if for example we have *n* words, then an *n x n* equation is to be formulated, which must fall within the allowed region of alphabetic range and other special characters that are used.
2. Each word is turned to an equation whose variables must not exceed *n*, the number of equations permitted.
3. To turn a word to non-linear equation, we start with x_1 and possibly add or subtract the distance of the current character form x_1. Adding a quantity representing the position of the alphabet from the position of the variable being used does this. For example, *E* can be represented by (x_1+4). This quantity is always multiplied either by $x_1, x_2, \ldots, x_n$ depending on the variable we have chosen from the left hand side for the purpose of encryption.
4. A starting value of the form $x_i = x_0 \bmod k + j$, where $i = 1, 2, \ldots, n$, x_0 is called the SEED and represents the number of letters in the text, k is the number of words in the text and is used as the starting value for the first variable, $n=x_0*k$ and $j=x_0*k$
5. When variables have been recovered, the value of x_1 is subtracted from the values of $x_2, x_3, \ldots x_n$.
6. As equations are formed, their equivalent delta encodings are also formed.

EXAMPLES/RESULTS

To explain this proposed method, we are going to use the following example to formulate an equation. The text below will lead to 4x4 non-linear equations.

Example 2: TRANSFER ONE MILLION NAIRA

T R A N S F E R

$$x_1(x_1+19)+x_1(x_2+16)+x_1(x_3-2)+x_1(x_4+10)+x_1(x_1+18)+x_1(x_2+4)+x_1(x_3+2)+x_1(x_2+16)$$

O N E

$$x_1(x_1+14)+x_1(x_4+10)+x_1(x_3+2)$$

M I L L I O N

$$x_1(x_1+12)+x1(x_2+7\ )x_1(x_3+9)+x_1(x_3+9\ )+x_1(x_2+7)+x_1(x_1+14)+x_1(x_4+10)$$

$$\begin{array}{ccccc} N & A & I & R & A \\ x_1(x_4+10)+ & x_1(x_3-2)+ & x_1(x_{2+}7)+ & x_1(x_2+16)+ & x_1(x_3-2) \end{array}$$

$x^2_1+x_1x_2+x_1x_3+x_1x_4+x_1{}^2+x_1x_2+x_2+x_1x_3+x_1x_2$
$x^2_1+x_1x_4+x_1x_3$
$x^2_1+x_1x_2+x_1x_3+x_1+x_1+x_1x_4$
$x_1x_4+x_1+x_3+x_1x_2+x_1+x_1x_3$

$2x^2_1 + 3x_1x_2 + 2x_1x_3 + x_1x_4-18 = 0$
$x^2_1 + x_1x_3 + \ + x_1x_4 - 8 = 0$
$2x^2_1 + 2x_1x_2 + 2x_1x_3 + x_1x_4 - 16 = 0$
$2x_1x_2 + 2x_1x_3 + x_1x_4 - 14 = 0$

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 1 & 2 & 3 & 2 \\ 19 & -3 & -18 & 12 & 8 & -14 & -2 & 14 \end{array}$$

$$\begin{array}{ccc} 1 & 4 & 3 \\ 14 & -4 & -8 \end{array}$$

$$\begin{array}{ccccccc} 1 & 2 & 3 & 3 & 2 & 1 & 4 \\ 12 & -5 & 2 & 0 & -2 & 7 & -4 \end{array}$$

$$\begin{array}{ccccc} 4 & 3 & 2 & 2 & 3 \\ 10 & -12 & 9 & 9 & -18 \end{array}$$

but $J(x)(x_{n+1} - x_n) = -f(x_n)$

$$J(x) = \begin{pmatrix} 4x_1 & 3x_1+3x_2 & 2x_1+2x_3 & x_1+x_4 \\ 2x_1 & & x_1+x_3 & x_1+x_4 \\ 4x_1 & 2x_1+2x_2 & 2x_1+2x_3 & x_1+x_4 \\ & 2x_1+2x_2 & 2x_1+2x_3 & x_1+x_4 \end{pmatrix}$$

$$\begin{pmatrix} 9x_1 & 3x_2 & 2x_3 & x_4 \\ 4x_1 & & x_3 & x_4 \\ 9x_1 & 2x_2 & 2x_3 & x_4 \\ 4x_1 & 2x_2 & 2x_3 & x_4 \end{pmatrix}$$

Applying Gauss elimination algorithm:

$$J(x_1) = \left(\begin{array}{cccc|c} 9 & 3 & 2 & 1 & 10 \\ 4 & 0 & 1 & 1 & 5 \\ 9 & 2 & 2 & 1 & 9 \\ 4 & 2 & 2 & 1 & 9 \end{array}\right)$$

$$= \begin{pmatrix} 9 & 3 & 2 & 1 & 10 \\ 0 & -^{4}/_{3} & ^{1}/_{9} & ^{5}/_{9} & ^{5}/_{9} \\ 0 & -1 & 0 & 0 & -1 \\ 0 & ^{2}/_{3} & ^{10}/_{9} & ^{5}/_{9} & ^{41}/_{9} \end{pmatrix}$$

$$= \begin{pmatrix} 9 & 3 & 2 & 1 & 10 \\ 0 & -^{4}/_{3} & ^{1}/_{9} & ^{5}/_{9} & ^{5}/_{9} \\ 0 & 0 & -1/12 & -5/12 & -^{17}/_{12} \\ 0 & 0 & ^{7}/_{6} & ^{5}/_{9} & ^{87}/_{18} \end{pmatrix}$$

$$= \begin{pmatrix} 9 & 3 & 2 & 1 & 10 \\ 0 & -4/3 & ^{1/}_{9} & ^{5}/_{9} & ^{5}/_{9} \\ 0 & 0 & -^{1}/_{12} & -^{5}/_{12} & -^{17}/_{12} \\ 0 & 0 & 0 & -5 & -15 \end{pmatrix}$$

$$\begin{array}{l} -5\,(x_4\text{-}1) = -15 \\ +\,5x_4 - 5 = +\,15 \\ 5x_4 \;=\; 20 \\ x_4 = {}^{20}/_{5} = 4 \end{array}$$

$$\left(\begin{array}{l} -^{1}/_{12}\,(x_3\text{-}1) - {}^{5}/_{12}(x_4\text{-}1) = -^{17}/_{12} \\ -^{1}/_{12}\,(x_3\text{-}1) - {}^{5}/_{12}\,(3) = -^{17}/_{12} \\ -^{1}/_{12}\,(x_3\text{-}1) \;\; - {}^{5}/_{4} = -^{17}/_{12} \\ -^{1}/_{12}\,(x_3\text{-}1) - {}^{5}/_{4} \;= -^{17}/_{12} \end{array} \right)$$

$$x_3 \;=\; 3$$

$$\begin{array}{l} -^{4}/_{3}\,(x_2-1) \;+ {}^{1}/_{9}\;\,(x_3\text{-}1) + {}^{5}/_{9}\,(x_4\text{-}1) = {}^{5}/_{9} \\ -^{4}/_{3}\,(x_2-1) \;+ {}^{2}/_{9} \;+ {}^{15}/_{9} = {}^{5}/_{9} \\ -^{4}/_{3}(x_2-1) \;+ {}^{17}/_{9} \;= {}^{5}/_{9} \\ -^{4}/_{3}\,(x_2-1) \;+ {}^{5}/_{9} \;- {}^{17}/_{9} \\ +^{4}/_{3}\,(x_2-1) \;+ {}^{4}/_{3} \\ 12(x_2-1) = +\,{}^{4}/_{3} \\ x_2 - 1 = 1 \\ x_2 = 2 \end{array}$$

$9(x_1 - 1) + 3(x_2 - 1) + 2(x_3 - 1) + 3(x_4 - 1) = 10$
$9(x_1 - 1) + 3 + 4 + 3 = 10$
$x_1 = 1$
$\therefore x_1 = 1, x_2 = 2, x_3, = 3, x_4 = 4$

Similarly, all the characters can be recovered using the value of each of the variables in conjunction with the delta encoded values.

CONCLUSIONS

The method proposed in this paper has been shown, through examples, to be practicable and one that can easily be adapted for use in our environment. Thc application of it will be easy, as it is *home breed.* Further improvement by the author is to be made in thc next few months in the areas of using c++ to fully develop the application to a standard that can be tested and applied. The use of Steepest Descent algorithm is to be employcd in determining the initial values for starting the solution algorithm.

REFERENCES

[4] Beekman, G., *Computer confluence* addison, Wesley Longman Inc., California, 1999, pp. 170-171, 290-300.
[1] Peha, J. M., *Encryption policy issues,* October 1998.
[3] Rivest, R. L., Shamir, A. and Adleman, L., *On a method for obtaining. digital signatures and public key cryptosystems*, commun. of the Acm vol. 21, February 1978, pp. 120-126.
[2] Tanenbaum, A. S., *Computer networks*, Prentice Hall, 1996, pp. 577-766.