Journal of ICT, 10, pp: 67–83

CRYPTOGRAPHIC ALGORITHM USING MATRIX INVERSION AS DATA PROTECTION

Peter B. Zirra

Department of Mathematical Sciences
Adamawa State Universiti
Mubi, Nigeria

zirrapeter@yahoo.com

G.M. Wajiga

Department of Mathematics and Computer Science,
Federal Universiti of Technology
Yola, Nigeria

gwajiga@gmail.com

ABSTRACT

This study aimed at providing algorithms to secure sensitive information in store or in transit via unsecure channels from the hands of the Internet criminals by scrambling the sensitive information into a set of linear equations in matrix form and deciphering by solving the systems of the linear equations in conjunction with the principles of the delta encoding scheme, a formula and a lookup alphanumeric position table. The algorithms were tested with samples of real data and the result practically demonstrated how systems of non-linear equations turned into linear equations are indeed of relevance in providing strong and complex encryption and decryption keys to protect sensitive information in store or in transit via unsecure channels. The proposed algorithms ensured that unauthorized users do not have access to sensitive and classified documents stored or transmitted to an intended recipient without a secret key. This means that, there is confidentiality, non-repudiation and integrity of our sensitive and classified information from the hands of unauthorized users on the Internet because of the robustness of the proposed algorithms.

Keywords: Cryptographic algorithms, Delta encoding, Newton's method, Inversion matrix, Non-linear equations, Decryption key, Cryptanalyst.

INTRODUCTION

Information security has become a very critical aspect of modern computing systems. With the global acceptance of the Internet, virtually every computer in the world today is connected to every other. While this has created tremendous productivity and unprecedented opportunities in the world we live in, it has also created new risks and burden for the users of these computers. The users, business, education and organizations worldwide have to live with constant fear of threats from intruders (also known as terrorists, attackers, hackers, crackers, unauthorized users), who use a variety of techniques and tools to break into computer systems, steal information, change data and cause havoc (Kahate, 2008). If no security measures are taken, there is no doubt that such data and other sensitive information will continuously be faked, polished and formatted by the hackers.

There have been several high-profile incidents of loss of individual privacy and confidentiality of data in unsecure communication channels for example, the Russian attacker (called Maxim) managed to intrude into a merchant Internet site and obtained 300,000 credit card numbers from its database (Kahate, 2008); a Swedish hacker broke into Microsoft's Hotmail Website and created a mirror site. This site allowed anyone to enter any Hotmail user's email id and read their emails. These attributed to successful attacks on security (Kahate, 2008). An Australian long-time computer hacker called Julian Assange hacked into the master terminal that the Canadian telecom company Nortel, maintained in Melbourne, Australia. In 2006, he created a WikiLeaks, a website that published confidential government documents and images (Discoverthenetwork.org, 2010; The Times of India, 2011). There were also recent reports from the United States and Canada that indicate that 'hackers' penetrated the information database of the American Military System in 2009 to wreak monumental havoc (Olaoye, 2011).

Therefore, cryptographic algorithms which help prevent interception and enhance message security are now of primary importance but the challenging part of cryptography today is how to create a key that would prevent an unauthorized decryption of message. A key is simply a parameter to the algorithm that allows the encryption and the decryption processes to occur.

This paper provides cryptographic algorithms using high mathematical formulations that enciphered message into a form of systems of non-linear equations that are deciphered through the Newton's method of solving the systems of non-linear equations, and the principles of delta encoding scheme together with the help of a lookup alphanumeric position table which is notoriously difficult.

The idea behind these proposed cryptographic algorithms is to provide protection access to sensitive resources in store or transit, by ensuring that the resources required for revealing the secret information are strong and complex enough through a high mathematical formulation of the secret key. The more illegal trials attempted, the more difficult it is for the criminals to unveil the sensitive information. This may discourage the criminals based on the complexity of the decryption key.

BACKGROUND

Koblitz (1994) viewed cryptography as an algorithmic process of converting a plain text message to a cipher text message based on an algorithm that both the sender and the receiver know, so that the cipher text message can be returned to its original, plain text form. In its cipher form, a message cannot be read by anyone but the intended receiver. The act of converting a plain text message to its cipher text form is called enciphering (Potdar & Chang, 2004). Reversing that act (i.e., cipher text form to plain text message) is deciphering. Enciphering and deciphering are commonly based on a key referred to as encryption or decryption key, respectively.

There are a number of key-based algorithms for performing encryption and decryption (Buchmann, 2005; Stallings, 2006). Figg (2005) classified cryptography algorithm into two: symmetric and asymmetric key cryptography.

In symmetric key cryptography, the same key is used for both encryption and decryption (Yusuf, 2007). A shared key is vulnerable to fall into the wrong hands. Once a key becomes known to someone not authorized, then the security based on that key is compromised. In asymmetric key cryptography, one key is used for encryption and another, mathematically related key, is used for decryption (Beekman, 1999; Rudolf, 2000; Su, Zobel, & Iwu, 2003; Laudan & Traver, 2004). An asymmetric key can be much less secure than a symmetric key since it avoids the mechanism of key sharing. This study reports on a similar encryption technique that uses an asymmetric key due to its robustness.

Types of Key-based Encryption Algorithms

Currently, there are several kinds of key-based encryption algorithms categorized by their functions and target groups. The most common key based encryption techniques are given (Rudolf, 2000; Moore , 2001; Diaa, Hatem & Mohiy, 2010) but most of these algorithms were found to have weak keys or sometimes the keys could easily be guessed. There is large research literature

on their insecurity as found in the work of Williams (1982, Wiener 1990), Menezes, Ooschot, & Vanstone (1996), Boneh (1999), Obi (2000), Schneier (1996, 2003), Young & Yang, (2004), Murphy (2005), Stallings (2006), Goh (2007), Yusuf (2007) and Kak (2009).

Obviously, as observed from the forgoing, it seems that, there appears to be no stronger ciphers that can guarantee the safety of our vital information from the hands of the enemy.

Systems of Linear and Non-linear Equations

This refers to problems involving the solution of systems of linear and non-linear equations, possibly with a very large number of variables. In recent years considerable interest in systems of non-linear equations has been stimulated due to their numerous applications in the area of science and engineering (Biazar & Ghanbary, 2009).

Crina and Ajith (2008) defined a non-linear system of equation as:

$$f(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_n(x) \end{bmatrix} \quad (1)$$

where $x = x_1, x_2,...x_n$ refers to n equations and *n* variables, and $F_1, F_2,...,F_n$ are non-linear functions in the space of all real-valued continuous functions on $\Omega = \prod_{i=1}^{n}[a_i, b_i] \subset \Re^n$. Some of the equations can be linear, but not all of them. Finding a solution for a non-linear system of equations *f(x)* involves finding a solution such that every equation in the non-linear system is zero i.e. F(x)=0 (Burden & Faires, 2001). This system can be referred by:

$$\left.\begin{array}{l} f_1(x_1, x_2, x_3, \cdots, x_n) = 0 \\ f_2(x_1, x_2, x_3, \cdots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, x_3, \cdots, x_n) = 0 \end{array}\right\} \quad (2)$$

Biazar and Ghanbary (2008) assume that the system admits a unique solution.

Newton's Method

There several ways to solve systems of non-linear equations. Probably the most popular techniques are: Newton-type method, Trust-region method,

Broyden method, Secant method, and Halley method (Crina & Ajith, 2008). The most known iterative method for solving systems of non-linear equations is the classical Newton's method.

$$J(x_n)\delta_x = -F(x_n) \tag{3}$$

where $\delta_x = x_n - x_{n-1}$ is the solution vector for the set of simultaneous linear equations and $J(x_n) \epsilon\ \Re_{nn}$ denotes the Jacobian matrix to F evaluated at current approximation $x_n = x_{n,1}, x_{n,2}, \ldots, x_{n,n}$ and x_{n+1} the next approximation. The Jacobian matrix is defined as:

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \tag{4}$$

Explicit computation of is avoided by performing the operation in a two-step manner.

Non-linear problems are often treated numerically by reducing them to a sequence of linear problems (Burden & Faires, 2001).

Matrix Inversion of Square Matrix

A linear system of equations is a set of *n* linear equations in *k* variables (sometimes called "unknowns"). Linear systems can be represented in matrix form as the matrix equation.

$$\boldsymbol{Ax=b} \tag{5}$$

Where A = $\begin{pmatrix} \boldsymbol{a_{11}} & \cdots & \boldsymbol{a_{1n}} \\ \vdots & & \vdots \\ \boldsymbol{a_{n1}} & \cdots & \boldsymbol{a_{nn}} \end{pmatrix}$ is the matrix of coefficients, $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ is the column vector of variables, and $b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$ is the column vector of solutions. If $k<n$, then the system is (in general) over-determined and there is no solution.

If $k=n$ and the matrix $\boldsymbol{A}$ is non-singular, then the system has a unique solution in the $\boldsymbol{n}$ variables. In particular, as shown by Cramer's rule, there is a unique solution if $\boldsymbol{A}$ has a matrix inverse $\mathbf{A^{-1}}$. In this case,

$$x = A^{-1}b \quad (6)$$

If $b=0$, then the solution is simply $x=0$. If $\boldsymbol{A}$ has no matrix inverse, then the solution set is the translate of a subspace of dimension less than $\boldsymbol{n}$ or the empty set (Stroud and Dexter, 2007).

Delta Encoding

Delta encoding δ refers to several techniques that store data as the difference between successive samples (or characters), rather than directly storing the samples themselves (Steven, 2007) as shown Table 1.

Table 1

Example of Delta Encoding

Original data stream	05	10	05	12	20	15	15	01	23	52	45	…
Delta encoded (δ)	05	-05	05	-07	08	05	00	14	-22	-29	07	…

The first value in the delta encoded file is the same as the first value in the original data. All the following values in the encoded file are equal to the difference (delta) between the corresponding value in the input file, and the previous value in the input file.

More precisely,

$$\boldsymbol{\delta_x} = (\mathbf{x_n} - \mathbf{x_{n-1}}), \quad \text{where } n = 1, 2, 3, \ldots, \text{ and } x_0 = 0 \quad (7)$$

EXPERIMENTAL DESIGN

The proposed encryption and decryption algorithms consist of multilevel enciphers.

Words Compression Process

The first level is realized through the compression of words. The plain text to be ciphered is first compressed word by word as follows:

```
Algorithm_Words_Compression(input, output)
Step 1:     Read plain text (T)
Step 2:     Split plain text (T) into words
Step 3:     Determine the word length (L) from split words
Step 4:     Initialize variable K = 0
Step 5:     Set K = K + 1
Step 6:     If K = L then
            a:  set the first word  =  word token (K)
            b:  go to Step 7
            else Step 6
                  ouput the compressed text
            end of Step 6
Step 7:     Initialize variable J =0
Step 8:     a:  append K to index of words
            b:  set  J = J+1
Step 9:     Check if J < L then
            a:  set second word = word token (J)
            b:  repeat Step 10
            else Step 9
                  output the index of the words
                  return to Step 6
            end of Step 9
Step 10:    Compare the first word and the second word
Step 11:    If first word  =  second word then
            a:  check first word  against any empty value
            b:  if check is false then
            i:    append content of first word to
                  compressed word
            ii:   set the word token (J) to empty
            iii:  append J to the trailing of the
                  index of words
            iv:   go to Step 12
                  else Step 11
                  return Step 8(b)
                  end Step 11(b)
            else of Step 11
            c:  check second word against an empty value
            d:  if check is false
                  i:    append content of second word to
                        compressed text
                  ii:   return to Step 8(b)
                        else
                        return to Step 8(b)
                        end of Step 11(d)
            end of Step 11(d)
Step 12:    append content of compressed word to
            compressed text
            return to Step 8(b)
End Algorithm_Word_Compression
```

Algorithm 1: Word Compression

Transformation of Words to Linear and Non-linear Equation

The second level is achieved by transforming the compressed words (the number of words that were left from Algorithm (1) into the system of non-linear equations (2). The proposed mathematical formulation steps are as given below.

i. Count the words from Algorithm 1, i.e. the number of words that were left. Assuming we have p words, it requires p equations that must be within the permitted region of the alphabetic range coupled with the characters to be used.
ii. Turn each word into an equation whose variables must not be more than **p**, the number of equations allowed.
iii. Transform each word into the proposed equation starting with x_1 and next x_2, …, x_p and then adding or subtracting the 'space' of the current character from x (i.e adding a quantity denoting the position of the alphabet from the position of the variable being used helps to ensure the accomplishment of this).
iv. Multiply each term obtained in (iii) by a factor , sum all and then equate to the sum of the variable positions in the word. These give nonlinear systems of equations **Ax=b**.

Use Newton's Method

To further confuse the cryptanalyst, the system of non-linear equations is transformed again by using equation (3) as follows:

(a) set n =0,
(b) guess an approximation solution X_0,
(c) calculate $J(X_n)$ and $F(X_n)$, where $J(X_n)_{i,j} = \frac{\partial f_i(x)}{\partial x_j}$, for $1 \leq i, j \leq n$,
(d) solve the linear system , $J(x_n)\delta_x = -F(x_n)$,
(e) set , $\delta_x = x_{n+1} - x_n$,
(f) set , $\delta_x = \delta_x + 1$, and
(g) test for convergence.

For the purpose of this paper, we consider the following:

1. Explicit computation of the inversion of Jacobi (i.e. $J(x_n)^{-1}$) is avoided as this will involve additional iteration for determining $J(x_n)^{-1}$.
2. Instead a new approximation x_n, is obtained by adding δ_x *to* x_{n+1} to as the next iterate.
3. The starting point is chosen carefully with an approximate idea of the guess of the function in mind as the difference between the sums of the coefficients of and divided by of a given system of equations.

Inversion of Matrix

To disguise the message further, the matrix $\mathbf{A}^{-1}$ is obtained from matrix $\boldsymbol{A}$, and $\boldsymbol{x} = \mathbf{A}^{-1}\mathbf{b}$ is transmitted as cipher text.

Delta Encoding Scheme

This is the final level of the encryption:

1. We formulate a lookup table as shown in Table 2.

Table 2

A Lookup Alphanumeric Position

	2	3	4	5	6	7
0		0	@	P	`	p
1	!	1	A	Q	a	q
2	‘’	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	‘	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[	k	{
C	,	<	L	\	l	}
D	-	=	M	]	m	|
E	.	>	N	^	n	~
F	/	?	O	_	o	₦

2. Hide the positions of the variables that represent the original characters by encoding the distance of each variable solution of the variables of the text message from Table 2 as follows:

(a) for each variable index representing a character in a word, x_n, we write the difference between its actual position of the character it represents and its variable solutions obtained, and

(b) find $\delta_x = x_n - x_{n-1}$; with $x_0 = 0$, $n = 1, 2, 3, \ldots$ Thus, is the current quantity representing the position of each character from its variable solution and x_{n-1} is the previous quantity representing the position of each character from its variable solution.

This is also a step further to create more states of confusion for an unauthorized person.

Decryption Process

The decryption process is performed simply by:

1. Solving the systems of linear equations in Inversion of Matrix.
2. Using the variable solutions obtained in conjunction with the formula, $s_k = v_s + \sum_{n=1}^{k} \delta_x$, where is the variable solution.
3. Using Table 2 as a lookup table to finally locate the position of the characters of each word.
4. Writing, the characters (in deciphering) as they appeared coupled with spaces between them and where there are index values in place of the words that previously occurred, the index values are interpreted and written in the places they appeared.

APPLICATION OF THE EXPERIMENTAL DESIGN

The Encryption Process

Taking the text "*who is promising who*", it can be encrypted using the Algorithm 1 and the procedures as follows:

1. If the plain text "*who is promising who*" is input into Algorithm 1, it reduces the words to "*who is promising*" and one index value that indicated the re-occurrence of a word. In other words, it means that this algorithm produces n =3 number of words. This number will now serve as a guide to formulate the number of equations whose variable index must not also exceed the number of words.
2. Using the procedure in section Transformation of Word to Linear and Non-linear Equation, result of:

$$\left.\begin{array}{ll} \text{who:} & x_1^2 + x_1x_2 + x_1x_3 = 6 \\ \text{is:} & x_1^2 + x_1x_2 = 3 \\ \text{promising:} & 4x_1^2 + 2x_1x_2 + 3x_1x_3 = 17 \end{array}\right\} \quad (8)$$

This above representation can be written as:

$$\left.\begin{array}{l} x_1^2 + x_1x_2 + x_1x_3 - 6 = 0 \\ x_1^2 + x_1x_2 + 0x_1x_3 - 3 = 0 \\ 4x_1^2 + 2x_1x_2 + 3x_1x_3 - 17 = 0 \end{array}\right\} \quad (9)$$

3. Apply the principle of solving equation (3) on equation (9) to obtain:

$$J(x_n)\delta_x = -F(x_n) \Rightarrow \begin{bmatrix} 2x_1 + x_2 + x_3, & x_1 & x_1 \\ 2x_1 + x_2, & x_1 & \\ 8x_1 + 2x_2 + 3x_3, & 2x_1 & 3x_1 \end{bmatrix} \delta x = -\begin{bmatrix} x_1^2 + x_1x_2 + x_1x_3 - 6 \\ x_1^2 + x_1x_2 - 3 \\ 4x_1^2 + 2x_1x_2 + 3x_1x_3 - 17 \end{bmatrix} \quad (10)$$

4. We take an initial guess of X_0 = **(1, 1, 1)**^{t} and substitute in equation (10), to obtain $J(X_0)$ and $f(X_0)$ as in equation (11) in matrix notation.

Hence, from equation (10) we obtain:

$$\begin{bmatrix} 4 & 1 & 1 \\ 3 & 1 & 0 \\ 13 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \\ x_3 - 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 8 \end{bmatrix} \quad (11)$$

where $A = \begin{bmatrix} 4 & 1 & 1 \\ 3 & 1 & 0 \\ 13 & 2 & 3 \end{bmatrix}$, $x = \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \\ x_3 - 1 \end{bmatrix}$ and $b = \begin{bmatrix} 3 \\ 1 \\ 8 \end{bmatrix}$

5. Transform equation (11) to $\boldsymbol{x} = \mathbf{A}^{-1}\mathbf{b}$ as follows:

(a) compute |A| as , $-\frac{1}{4}$,

(b) calculate , $A^{-1} = -\frac{1}{4}\begin{bmatrix} 3 & -1 & -1 \\ -9 & -1 & 3 \\ -7 & 5 & 1 \end{bmatrix}$,

(c) then the expression $\boldsymbol{x = A^{-1}b}$ can be written as

$$x = -\frac{1}{4}\begin{bmatrix} 3 & -1 & -1 \\ -9 & -1 & 3 \\ -7 & 5 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \\ 8 \end{bmatrix} \quad (12)$$

Equation (12) is now transmitted as cipher text to the intended recipient.

The Decryption Process

1. Apply the steps in Decryption Process to equation (12), to obtain the approximate solution $x_1 = 1$, $x_2 = 2$ and $x_3 = 3$.
2. Obtain the lookup alphanumeric position in Table 2.
3. Find the δ_x values associated with each solution of the variable index of the formulated systems of non-linear equations calculated using equation (7) in conjunction with Table 2. The result is depicted in Table 3.

Table 3

Delta Encoding: Who is Promising

Virtual Position of Characters (x_n)	*76*	*66*	*6c*	*68*	*71*	*6d*	*71*	*6c*	*6b*	*68*	*71*	*68*	*6b*	*6b*
Solution of Equations (v_s)	*1*	*2*	*3*	*1*	*2*	*3*	*1*	*3*	*2*	*1*	*2*	*1*	*3*	*1*
δ_x	*76*	*-10*	*6*	*68*	*9*	*6d*	*04*	*-05*	*-01*	*-03*	*09*	*-09*	*03*	*-05*

4. Use the formula in conjunction with Tables 2 and 3. The effect of the formula is shown in Table 4.

RESULT

Table 4 shows the result of the deciphered text from the enciphered text obtained from the proposed algorithms.

Table 4

Plain Text Recovery

Position of Word	$s_k = v_s + \sum_{n=1}^{k} \delta_x$	From Table 2 (R.C)
1	$X_1 = 1 + 76 = 77$	*77 = w*
	$X_2 = 2 + 76 - 10 = 68$	*68 = h*
	$X_3 = 3 + 76 - 10 = 6f$	*6f = o*
2	$X_1 = 1 + 68 = 69$	*69 = i*
	$X_2 = 2 + 68 + 9 = 73$	*73 = s*
3	$X_3 = 3 + 6d = 70$	*70 = p*
	$X_1 = 1 + 6d + 4 = 72$	*72 = r*
	$X_3 = 3 + 6d + 4 - 5 = 6f$	*6f = o*
	$X_2 = 2 + 6d + 4 - 5 - 1 = 6d$	*6d = m*
	$X_1 = 1 + 6d + 4 - 5 - 1 - 3 = 69$	*69 = i*
	$X_2 = 2 + 6d + 4 -5 - 1 - 3 + 9 = 73$	*73 = s*
	$X_3 = 3 + 6d + 4 -5 - 1 - 3 + 9 - 9 = 69$	*69 = i*
	$X_1 = 3 + 6d + 4 -5 - 1 - 3 + 9 - 9 + 3 = 6e$	*6e = n*
	$X_1 = 1 + 6d + 4 -5 - 1 - 3 + 9 - 9 + 3 - 5 = 67$	*67 = g*
4	$X_1 = 1 + 76 = 77$	*77 = w*
	$X_2 = 2 + 76 - 10 = 68$	*68 = h*
	$X_3 = 3 + 76 - 10 + 6 = 6f$	*6f = o*

Hence the enciphered words are fully recovered.

DISCUSSION

It is clearly observed from the result of the proposed algorithm that any method of attack to find the decryption key by the unauthorized user required: solving the set of linear equations in equation (11), finding the values associated with

the variable index in Table 3, formulating the equation and understanding the position of the alphanumeric characters in Table 2. This is notoriously difficult. Hence, the proposed method provides strong protection of data from unauthorized access in storage or transmission. This is in agreement with an earlier study by Ritter (2007) who revealed that resources required for revealing a secret message should be strong and complex enough through a hiding key.

The result in Table 4 was obtained through the use of a public key (asymmetric cryptography). This avoided the risk of sharing keys (key falling into the wrong hands) associated with the symmetric cryptography. The findings are in accordance with the study conducted by Murdoch (2001) who found out that there is less risk associated with a public key than the symmetric key and the security based on that key is not compromised (Talbot & Welsh, 2006).

From the finding, the decrypted messages were not visible to an unauthorized person without the decryption keys. This proved how the proposed algorithms can fortify the security of our sensitive and classified information both in store and transit from the hands of Internet crooks. According to Wang (2002) it means that there is confidentiality, non-repudiation and integrity of our sensitive and classified information over the Internet from the hands of Internet terrorists. This is because of the robustness of the proposed algorithms. This view was also highlighted by Kahate (2008), and Kessler (2010).

The result in Table 4 has proved the validity, reliability and practicality of the proposed application of the systems of non-linear equations to encryption and decryption algorithms in handling real life situations in an unsecured environment. It was for this reason that Biazar and Ghanbary (2008) observed that in recent years, considerable interest in systems of non-linear equations has been stimulated due to its numerous applications in the area of Science and Engineering.

CONCLUSION

The proposed theoretical design provided strong protection of sensitive resources from illegal access by hackers while in store or transit, by ensuring that the resources required to reveal the secret information was strong and complex enough through a complex decryption key. The more illegal trials attempted, the more difficult it was for the criminals to unveil the sensitive information.

The findings of this study are consistent with the claims of researchers such as Biazar and Ghanbary (2009), and Kahate (2008) that the systems of non-linear equations are indeed of relevance in providing security to secure our sensitive information in store or in transit via unsecure channels from the hands of the eavesdroppers through the use of strong and complex encryption and decryption keys that are notoriously difficult to break.

Thus, these algorithms have proved their practicality in handling real life problems in an unsecured environment.

REFERENCES

Beekman, G. (1999). *Computer confluence*. Calfonia: Addison-Wesley.

Biazar, J., & Ghanbary, B. (2008). A new approach for solving systems of non-linear equations. *International Mathematical Forum, 3*(38), 1885–1889.

Biazar, J. & Ghanbary, B. (2009). A modification on Newton's method for solving systems of non-linear equations. *World Academy of Science, Engineering and Technology, 58*, 897–901.

Boneh, D. (1999). Twenty years of attack on the RSA cryptography. *Notices American Mathematic Society*, *46*(1), 203–213.

Buchmann, J. (2005). *Introduction to cryptograph* (2nd ed.). New York: Springer- Verlag.

Burden, R. J., & Faires, J. D. (2001). *Numerical analysis* (7th ed.). USA: Brooks/Cole.

Crina, G., & Ajith, A. (2008). A new approach for solving non-linear equations system. *IEEE Transaction on Systems, Man, and Cybernetic*, *38*(3), 698–714.

Diaa, S. A. M., Hatem, M. A. K., & Mohiy, M. H. (2010). Evaluating the performance of symmetric encryption algorithms. *International Journal of Network Security, 10*(3), 213–219.

Discoverthenetwork.Org. (2010). *A guide to the political left*. Retrieved from www.discoverthenetworks.org.

Figg, B. (2004). *Cryptography and network security*. Retrieved from http:/ www.homepages.dsu.edu/figgw.

Goh, E. (2007). *Encryption schemes from bilinear maps* (Unpublished doctoral dissertation). USA: Stanford University.

Kahate, A. (2008). *Cryptography and network security* (2nd ed.). New Delhi: Tata McGraw Hill.

Kak, A. (2009). *Classical encryption techniques. Lecture notes on "Computer and Network Security"*. Purdue University.

Kessler, G. C. (2010). *Handbook on local area networks: An overview of cryptography*. United Kingdom: Auerbach. Retrieved from http://www.garykessler.net/library.

Koblitz, N. (1994). *A course in number theory and cryptography* (2nd ed.). Berlin: Springer Verlag.

Laudan, C. K., & Traver, C. G. (2004). *E-commerce. Business. Technology. Society* (2nd ed.). New York: Pearson Education.

Menezes, A., Ooschot, P.V., & Vanstone, S. (1996). *Handbook of applied cryptography*. New York: CRS Press.

Murdoch, M. (2001). *Introduction to cryptography,* Part 1: *The broad view.*

Murphy S. (1990). The cryptanalysis of FEAL-4 with 20 chosen plaintexts. *Journal of Crytology*, *2*(1), 145–154.

Moore, G. W. (2001). *Cryptography mini-tutorial.* Lecture notes University of Maryland School of Medicine. Retrieved from http://www.medparse.com.

Obi, G. M. M. (2000). A generalization of the RSA algorithm. *Proceedings of Computer Association of Nigeria Conference Series, 11*(1), 203 – 207.

Olaoye, B. (2011). *ICT security training scholarship to university. A letter of Invitation.*

Potdar, V., & Chang, E. (2004). *Disguising text cryptography using image cryptography.* International Network Conference. United Kingdom: Plymouth.

Rudolf, D. (2000). *Development and analysis of block cipher and DES system*. Retrieved from http://www.cs.usask.ca.

Schneier, B. (1996). *Applied cryptography*. New York: John Wiley and Sons.

Schneier, B. (2003). *Practice cryptography*. New York: John Wiley and Sons.

Stallings, W. (2006). *Cryptography and network security* (4th ed.). Englewood (NJ): Prentice Hall.

Steven, W.S. (2007). *The scientist and engineer's guide to digital signal processing*. Califonia: Technical Publishing.

Stroud, K.A., and Dexter, J.B. (2007). *Engineering mathematics* (6th ed.). New York: Palgrave Macmillan.

Su, N., Zobel, R.N. & Iwu, F.O. (2003). *Simulation in cryptographic protocol design and analysis.* Proceedings of 15th European Simulation Symposium. University of Manchester, UK.

Talbot, J., & Welsh, D. (2006).*Complexity and cryptography: An introduction* New York: Cambridge University Press.

The Times of India. (2011). Retrieved from http://www.timesofindia. indiatimes.com.

Young, A., & Yung, M. (2004). *Malicious cryptography- exposing crypto virology*. New York: John Wiley & Sons.

Yusuf, M. A. (2007). *Data security: Layered approach algorithm* (Unpublished master's thesis). Abubakar Tafawa Balewa University, Bauchi, Nigeria.

Wang, H. (2002). *Security architecture for the teamdee system* (Unpublished master's thesis). Polytechnic Institution and State University, Virginia, USA.

Wiener, M. (1990). Cryptanalysis of short RSA secret exponents. *IEEE Transaction Information Theory*, *36*(1), 553–558.

William, H. C. (1982). A p+1 method of factoring. *Mathematic of Computation, l39* (3), 225–234.