Journal of Engineering and Sustainable Development
Vol. 20, No. 04, July 2016
ISSN 2520-0917
www.jeasd.org

IMPLEMENTATION AND ENHANCEMENT AFFINE CIPHER OF DATABASE

Dalal A. Hammood [1], Maitham A. Naji [2]

1) Lecturer, Computer Engineering Technology Department, College of Electrical and Electronic Techniques, Middle Technical University, Baghdad, Iraq
2) Asst. Prof, Power Engineering Technology Department, College of Electrical and Electronic Techniques, Middle Technical University, Baghdad, Iraq

Abstract: The first of this paper, Caeser cipher is used to encrypt the fields in data base. The keys have been used from different lengths of fields, then applying affine algorithm. This is given a layer of security for data base. The cipher text is stored as a raw data in text file, which is separated between fields by Semicolon. The process is continued to end of records. The second part is creating a new Data Access Object(DAO), tables, records, fields, through counting a number of semicolon in a raw. The program keeps going until obtaining original records.

Keyword: *database, Caesar, Affine cipher, and security*.

تطبيق وتحسين شفرة Affine لقواعد البيانات

الخلاصة:في الجزء الاول من هذا البحث، استخدمت طريقة القيصر Caesar في عملية تشفير جداول قواعد البياتات. تم استخدام مفاتيح التشفير باطوال مختلفة من طول القيد ويشفر القيد من خلال تطبيق خوارزمية Affine. هذا مما يوفر امنية عالية الجودة لقاعدة البيانات. بعد ذلك يخزن ناتج القيد المشفر كسطر في الملف النصي حيث يفصل ناتج تشفير كل حقل بفارزة منقوطة عن الحقل الاخر. تستمر هذه العملية الى اخر قيد في الجدول. اما الجزء الثاني للبحث فهو عملية خلق قاعدة بيانات جديدة باستخدام تقنية Data Access Object (DAO) ثم خلق الجداول وحقولها من خلال حساب عدد الفوارز المنقوطة في سطر الملف النصي ويستمر البرنامج باسترجاع النص المشفر الى النص الاصلي لكل قيد.

1. Introduction

Security is an important requirement in data transmission on the network. Cryptology: Is a study of techniques for ensuring the secrecy and/or authenticity of information.

---

*Corresponding Author alsaady_dalal@yahoo.com

Cryptology contains of two parts: Cryptography, which converts a plain

text into cipher text and versa depending on key. Cryptanalysis, which converts cipher text into plain text without known key, it depends on rules. Figure 1 shows the cryptosystem [1,2].

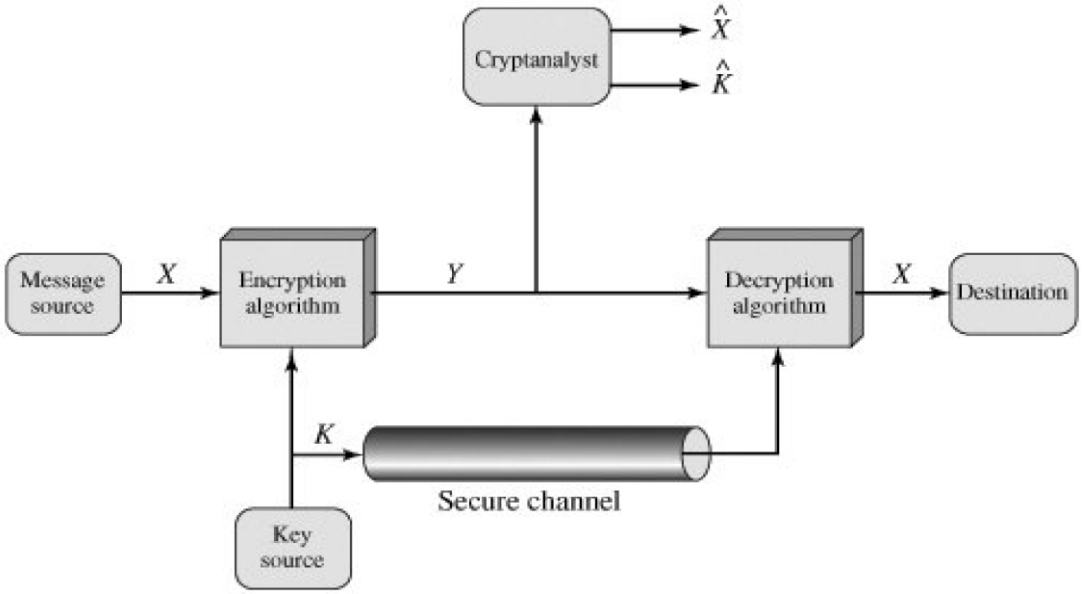

Fig. 1: Conventional Cryptosystem

Where X is plain text , Y is cipher text , and K is a key cipher.

There are a lot of types of cipher, some of them are:

1- Symmetric cipher

- Mono alphabetic cipher
- Poly alphabetic cipher
- Vigener cipher
- Caeser cipher
- Transposition cipher
- Hill cipher
- Playfir cipher
- Affine cipher
- One time pad cipher .. etc.

2- A symmetric cipher

- RSA cipher
- AES cihper
- DES cipher
- Knapsack cipher ..etc.

The Affine Cipher is a symmetric cipher. It works on twenty six letters which is determined by the following equations [1-3]:

$$\text{Cipher key}(x) = w\, s + t \mod 26 \quad (1)$$

Where $0 \leq w; t \leq 25$. w , s are key, and Cipher is cipher key(x) to encrypt the text. The variable w must be as : gcd(w,26)=1[1-3].

$$\text{Decipher key}(x) = w^{-1} (\text{cipher key}(x) - t) \mod 26 \quad (2)$$

In this paper Affine method is used (w,t) keys with a different values for each record.

Caeser method has been updated using a first word from first field, and city field. The result is stored in a text file, "Caeser enc.txt". an encryption file is encrypted by affine cipher, and the result is Affineenc. The original database is return by transforming Affinenc into Caeserenc file, then it converted to encryption database[1,4].

2. Database

The Database object is the most important object deal with because it allows to interface with the database. To look at the structure of a database, first look is needed at the objects, collections, and properties that exist in DAO hierarchy; figure (2) shows the relevant structure and properties [4].

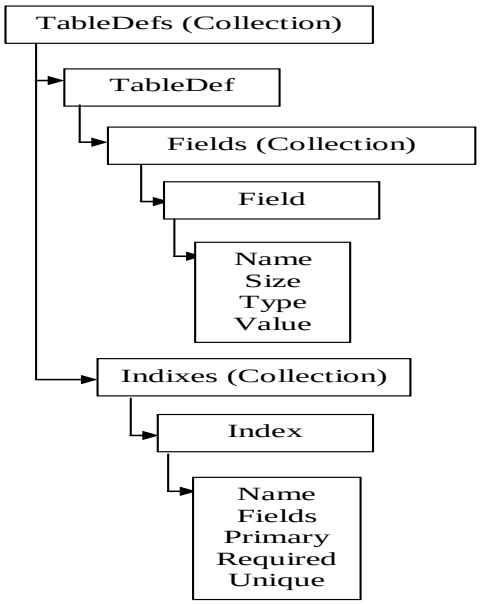

Fig.2 : The Tabledefs Collection Hierarchy [4]

2.1 TableDefs

Just as every object under the DBEngine object can be a collection or object, TableDefs is a collection of TableDef objects[4-6].

2.2 Fields

In addition, the table names are the fields that exist in the table. Each TableDef object contains a Fields collection, which in turn contains one or more Field objects. The most important properties are (name, size, source field, source table, type, city and value)[4-6].

2.3 Indexes

Just as you can look at the various fields you have in a table, you can also view the indexes on it[4].

There are various fields in a table, so It could be seen the indexes. Each TableDef object can have an associated Indexes collection with one or more Index Object in it. The most important properties are (fields, foreign, name, primary and unique)[4].

3- Literature Survey

In 2005, Y. M. Y. Hasan & E. M. Mohammed are submitted a new symmetrical key block cipher, with variable block and key lengths, referred as PATFC (Pseudorandom Affine Transformation-based Feistel Cipher), suitable for hardware and software uses [5].

In 2008, Noor. H. Arshad, Saharbudin. N Tahir , Azlinah.M, Abdul Manaf. M clear a new algorithm.This algorithm develops the impairment of the main affine cipher. the new alogrithm contains of a new encoding scheme and updating Cipher Block Chaining (CBC) mode of operation for block cipher. The result clear that the algorithm works appropriately, where the decryption process generated comparable output as the original plaintext and it ran through specific design and evaluated systematically with respect to database approach and algorithm technique to verify the adopted design[6].

In 2009, Kamaljit. K , K.S Dhindsa , Ghanaya. S, are introduced a new method and a strongly original of securing numerical data of databases. It introduces a functional solution to the problem where ciphered data was not possible to be stored in the existing numerical field and numerical data was converted to alphanumeric type. This work allowed explicit record level encryption that the data field type or fixed length could not change [7].

In 2010 Stephane Jacob is presented symmetric ciphers for database encryption since they are the single and common type of ciphers with satisfactory execution for a large amount of applications. The stream ciphers are the suitable type of encryption schemes. This work is presented a cryptanalyst a dedicated stream cipher proposed by Ge and Zdonic in 2007[8].

In 2013 and 2014 M., A., Naji is used Caesar method to generate one key to each record. The length of the key is computed from first word of record[9,10].

Whiles in this paper, two fields are used to determine the length of key. The proposed system also develops Affine method by using (n) keys for (n) records to encrypt the output of Caesar method.

4-Affine cipher

Affine cipher works on twenty six letters which is determined by the following equations[1]:

$$\text{Cipher key}(x)= w\, s + t \mod 26 \quad (3)$$

Where $0 \le w; t \le 25$. w , s are key, and Cipher is cipher key(x) to encrypt the text. The variable w must be as : gcd(w,26)=1[1-3,11,12].

$$\text{Decipher key}(x)= w^{-1} (\text{cipher key}(x) - t) \mod 26 \quad (4)$$

Firstly, assume that:

$$\gcd(w, 26) = d > 1 \quad (5)$$

Then the congruence:

$$ws \equiv 0 \pmod{26} \quad (6)$$

has (at least) two distinct solutions in $\mathbb{Z}_{26}$, i.e: $s = 0$ and $s = 26/d$.

In this case is not an injective function and hence not a valid encryption function as in equation 7.

$$e(x) = ws + t \bmod 26 \quad (7)$$

For example, since gcd(4, 26) = 2, it follows that $4s + 7$ is not a valid encryption function: s and $s + 13$ will encrypt to the same value, for any $s \in \mathbb{Z}_{26}$. Let's next suppose that gcd(w, 26) = 1. Assume for some s_1 and s_2 that in equation 8:

$$ws_1 = ws_2 \pmod{26} \quad (8)$$

Then:

$$w(s_1 - s_2) = 0 \pmod{26} \quad (9)$$

And thus $26 \mid w(s_1 - s_2)$. i.e., $s_1 = s_2 \pmod{26}$

The property of division is as the following:

if gcd (w, t) =1 and $w \mid t\,c$, then $w \mid c$. Since $26 \mid w(s_1 - s_2)$ and gcd (w, 26) = 1, we must therefore have that $16 \mid w(s_1 - s_2)$. i.e., $s_1 \equiv s_2 \pmod{26}$.

At this point we have shown that, if gcd(*a*, 26) = 1, then an equivalence of the form $ax \equiv y \pmod{26}$ has, at most, one solution in $\mathbb{Z}_{26}$. Thus, if we assume *x* vary over $\mathbb{Z}_{26}$, then *ax* mod 26 takes on 26 dissimilar values modulo. 26. That is, it takes on every value precisely once. It follows that, for any $y \in \mathbb{Z}_{26}$, the congruence $ax \equiv y \pmod{26}$ has a distinctive solution for *y*.

There is nonentity particular about the number 26 in this proof. The outcomes can be demonstrated in an equivalent mode [11, 12].

4-1 Theorem

The accordance $ws \equiv t \pmod{n}$ has a distinctive solution $s \in \mathbb{Z}_n$ for every $t \in \mathbb{Z}_n$ if and only if gcd(w, n) = 1.

Since 26 = 2 × 13, the values of $w \in \mathbb{Z}_{26}$ such that gcd(w, 26) = 1 are w = 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, and 25. The factor *t* can be any element in $\mathbb{Z}_{26}$. So, the Affine Cipher has 12 × 26 = 312 possible keys. (naturally , this is much too small to be safe) [11,12].

Let's now suppose the common adjustment where the modulus is *n*. We necessitate another description from number theory.

4-1-1 Example of encryption and deception process

Encryption function:

$$C = w\,s+t \mod 26 \qquad (10)$$

Table 1, clears encryption operation. The possible values that w could be are *1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, and 2,* As shown in table 3 .

Decryption function :

$$D = \text{inv}(w)\,(C- t) \mod 26 \qquad (11)$$

Table 2 clears decryption operation**.**

Example: the plaintext to be encrypted is "computer" w=5 t=8 as shown in table *1 ,and 2* [1-3].

Table 1: Encryption Operation

plaintext:	*c*	*o*	*m*	*p*	*u*	*t*	*e*	*r*	*s*	*y*	*s*	*t*	*e*	*m*
s:	2	14	12	15	21	19	4	17	18	24	18	19	4	12
5s+8	18	78	68	83	113	103	28	93	98	128	98	103	28	68

(5s+8) mod 26	18	0	16	5	9	25	2	15	20	24	20	25	2	16
cipher text:	S	A	Q	F	J	Z	C	P	U	Y	U	Z	C	Q

Table 2: Decryption Operation

ciphertext:	*S*	*A*	*Q*	*F*	*J*	*Z*	*C*	*P*	*U*	*Y*	*U*	*Z*	*C*	*Q*
C:	18	0	16	5	9	25	2	15	20	24	20	25	2	16
21(C-8):	210	-168	168	-63	21	357	-126	147	252	336	252	357	-126	168
(21(C-8)) mod 26:	2	14	12	15	21	19	4	17	18	24	18	19	4	12
plaintext:	c	o	m	p	u	t	e	r	s	y	s	t	e	m

According to equation (6) , the inverse of a is as shown in table (3).

Table (3): inverse of a

1	*3*	*5*	*7*	*9*	*11*	*15*	*17*	*19*	*21*	*23*	*25*
1	9	21	15	3	19	7	23	11	5	17	25

5- The Proposal Algorithm

The proposal algorithm are combined between two type of encryption. They are Caesar and affine cipher. In this paper, affine cipher is used after Caesar cipher.

5-1 Algorithm of data base creating

The system works on creating a new database, calculation number of semi colons, which it is compressed in text file, then the fields are created, that they are equal the same number of semi colons. Then the database and tables are stored as shown in fig. 3.

Start

1. calculation number of semicolons open access
2. open access
3. create file
4. create a new table
5. create a fields of table
6. define of each fields in a table by NAME of field and Type of field
7. save table in database
8. save a filename in DATA BASE
9. end

Fig. 3: Algorithm of Data Base

The original database is as shown in table 4.

Table (4): The Customer Table

Customer					
Customer No	Full Name	Title	Balance	Street	City
1	Kadhum Ahmed Ali	Mr	5000	Al Khdra	Baghdad
2	Naji Ali Naji	Mr	4500	Musum St	London
3	Asmaa Assam Kadhum	Miss	13000	Al Resoul	Erbil
4	Dalal Kamel Ali	Miss	7700	Al Beyaa	Baghdad
5	Sameer Mohammed Ahmed	Mrs	900	Dux St	Leeds

5-2 Algorithm of database encryption

Figure 4 clears the algorithm of affine cipher, it depends on a 3 parameters w,s,t. Where w is length of text.
Each number should be has inverse as shown in table 1[9].

In this algorithm, the filed is encrypted into a row in a text file, then the field has been converted from table, which separates the filed from other in semicolon in text file. The operation will keep on into end of table. As shown in Fig. 4. and Fig.

1. start
2. Open of text file
3. Checking and calculating fields number
4. Calculating length of field to represent a it should be has inverse number
5. Calculating length of b any length less than a
6. Applying encryption of affine cipher by the following equation C=ax-b mod 26
7. Save text file as a cipher text
8. List of field in database after affine algorithm
9. End

Fig. 4 : Algorithm of Encryption Affine Cipher

1. Text file
2. Length of file (number of fields)
3. Reading a first field=i
4. Begin top of database=filesize
    a. If file size(i) <>";" then go to 6
5. I=i+1
6. If i< filesize goto 4
7. Creating new table
8. Array= Split (Line1,";")
9. Phrase = Array(0)
10. Phrase
11. Key = Length (Item1(0))
12. Add New Record
13. I = I + 1
14. Plain-Text = CaesarDec-Function(Array(I),Key)
15. Input Plain-Text to Field(I)
16. I < = Fields Number
17. Read New Line1
18. NOT EOF (Text File) goto 8

Fig. 5 Fields Number Extraction

5-3 Algorithm of new database and table

A New database is created with a new table. Number of fields is created by system depending on number of semicolons in each row from text file, as shown in Fig.6.

1. Field no.
2. Delete all
3. Create new database
4. Create a new table in DB
5. I=0
6. I=i+1
7. F1 = Create Field (“F”+Ftr(I)),dbtext)
8. Append F1 to field collection
9. F1 = Nothing
10. I <= Field Number goto 3
11. Append new Table collection
12. End

Fig 6. Create Table Procedure

6-Results and Discussion

The system has been implemented by Visual Basic 6. The project main form has two basic commands "Table Encryption" and "Table Decryption" as shown in Figure(7).

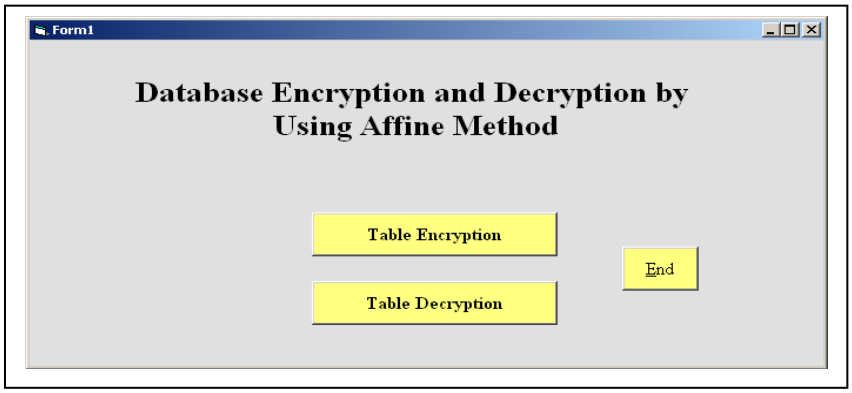

Fig. 7 Main Form Window

When the "Table Encryption" buttons is clicked input boxes is appeared to determine the database path and name shown in figure (8) and table name shown in Figures (9).

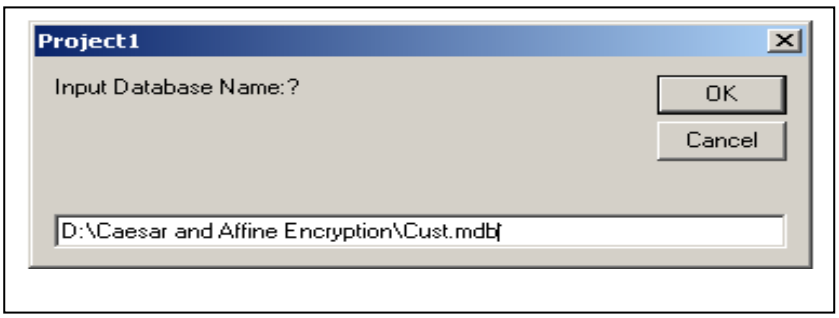

Fig. 8 Input Database Name

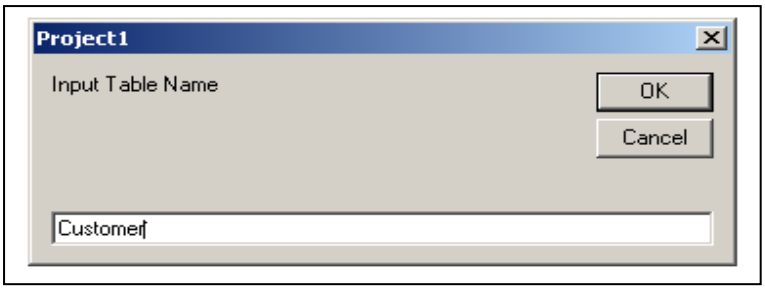

Fig. 9 Input Table Name

The command button "Table Encryption" is executed by click event. The system will process the "Customer" table and generate Caesar file is called "Caesarenc", according to encryption equation. As shown in Figure (10).

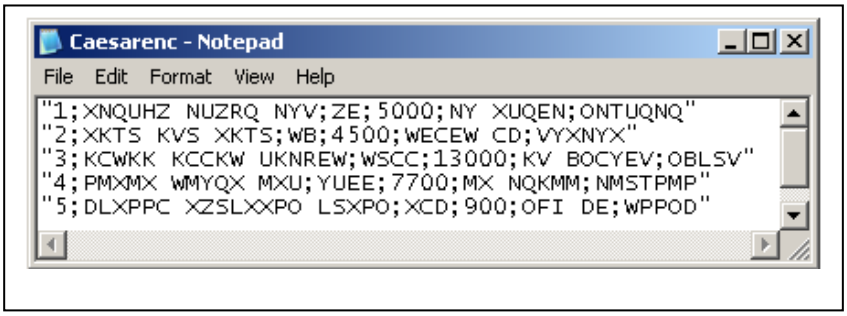

Fig.10 Caesar Encrypted File

The Caesar encrypted file will be an input to Affine encryption program. The system used row from Caesar file to produce line encrypted by Affine method. The process is continued until the ends of files in which the results saved in text file called "Affineenc" as shown in Figure (11). According to encryption equation:

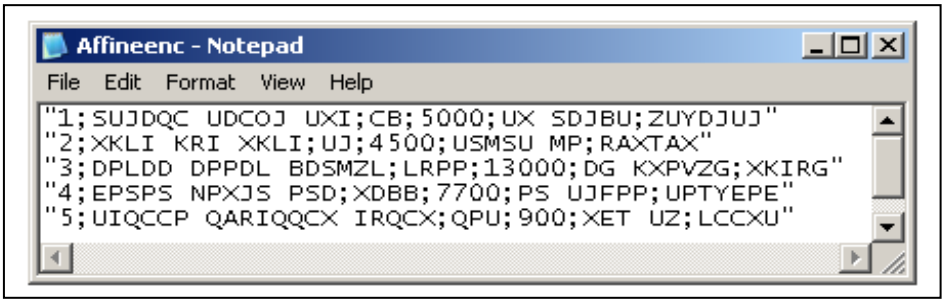

Fig.11 Affine Encryption File

The Decryption process is executed by click on "Table Decryption" button. The program processes Affine text file to produce Caesar cipher as shown in Fig.10. After that the Caesar decryption file convert to plan text and the result is saved in table as shown in table 5.

Table 5: Plain Table

CustT					
F 1	F 2	F 3	F 4	F 5	F 6
1	KADHUM AHMED ALI	MR	5000	AL KHDRA	BAGHDAD
2	NAJI ALI NAJI	MR	4500	MUSUM ST	LONDON
3	ASMAA ASSAM KADHUM	MISS	13000	AL RESOUL	ERBIL
4	DALAL KAMEL ALI	MISS	7700	AL BEYAA	BAGHDAD
5	SAMEER MOHAMMED AHMED	MRS	900	DUX ST	LEEDS

7. Conclusion

It is clear from results, can concluded the following:

1- The previous papers used Caesar cipher. This paper combines between two types of ciphers. They are Caesar and affine cipher. In this paper, affine cipher is used after Caesar cipher to get high security.
2- The capacity of text file is 2KB, it takes less than database, which it has 64KB.
3- Each record has double keys. This leads to obtain high security for table.
4- To retrieve plaintext , the key is generated from cipher text.
5- Finally, High security has been obtained from Conversion database into text file.

8. References

1. R.L. Rivest**,** 1997**,** " Handbook of Applied Cryptography: Foreword **",** CRC Press, Inc.
2. W. Stallings., 2011, *"Cryptography and Network Security Principles and Practices"*, 5th Edition, Prentice Hall.

3. R. Oppliger,2005, *"Contemporary Cryptography"*, computer security series, press artech house.
4. S. Nick,.2002, *"Oracle Programming with Visual Basic"*. SYBEXInc, USA, 715.
5. Y. M. Y. Hasan & E. M. Mohammed PATFC, 2005: *"Novel Pseudorandom Affine Transformation-Based Feistel-Network Cipher"*, IEEE International Symposium on Signal Processing and Information Technology, p811-816.
6. Noor. H. Arshad, saharbudin. N tahir ,Azlinah.M, Abdul Manaf. M, 2008, *"Database Encryption Using Enhanced Affine Block Cipher Algorithm"*, 10th wseas int. Conf. On mathematical methods and computational techniques in electrical engineering (mmactee'08), Sofia, Bulgaria, May 2-4, p71-76.
7. Kamaljit. K , K.S Dhindsa , Ghanaya. S, 2009, *"Numeric To Numeric Encryption of Databases: Using 3Kdec Algorithm"* IEEE International Advance Computing Conference (IACC 2009) Patiala, India, p1501-1505, 6-7 March.
8. Stephane. J, 2010, *"Cryptanalysis of a Fast Encryption Scheme for Databases"*. ISIT, Austin, IEEE . Texas, U.S.A., June 13 - 18, p2468-2472.
9. M., A., Naji, 2014, *"Implementation of Encryption Data Table by Using Multi-Keys"* IJSCI, India Vol. 4 Issue 2, P 14-17.
10. M., A., Naji., 2013, *"Implementation of Converting Text File to Data Access Table by Using Multi-Keys"* IJSER, USA Vol. 4 Issue 12, P 1090-1096.
11. Douglas R. Stinson, 1995, *"Cryptography: Theory and Practice"* , CRC Press LLC , Holmdel, New Jersey .
12. Alan G.Konheim,2007, *"Computer security and cryptography"*, WILEY , New Jersey.