A Chaos-Based Encryption Algorithm for Database System

Ekhlas Abbas Albahrani[1], Sadeq H. Lafta[2,*] and Naeem Howrie Ghayad[3]

[1]*Department of Computer Science, Mustansiriyah University, Baghdad, Iraq*
[2]*Department of Applied Science, University of Technology – Iraq*
[3]*Ministry of Labour and Social Affairs, Iraq*
E-mail: akhlas_abas@uomustansiriyah.edu.iq1; sadeq.h.lafta@uotechnology.edu.iq2; naeamhowrie@gmail.com
[*]*Corresponding Author*

Received 04 February 2022; Accepted 05 January 2023;
Publication 03 March 2023

Abstract

This paper introduces a new Database Transposition, Substitution and XORing Algorithm (DTSXA) based on using chaotic maps. It is based primarily on two well-known security properties: confusion and diffusion. A random number generator was depended on to produce the keys for the algorithm of encryption and decryption. The encryption of the Arabic language in addition to the English language was done, besides it can encrypt a table, individual row and individual column. The suggested algorithm was obeyed and analyzed by different tests involving brute force attack analyses, statistical attack analyses (security analysis histogram, correlation coefficient analysis and information entropy analysis), key sensitivity analysis, differential attack analysis, and mean square error analysis. This algorithm passed all the applied analyses well-deservedly, which indicates that the

Journal of Cyber Security and Mobility, Vol. 12_1, 25–54.
doi: 10.13052/jcsm2245-1439.1212

presented encryption algorithm has a high security level due to its large key space and high sensitivity to the change in the cipher keys.

Keywords: Database, encryption, security, chaos maps, Kaplan–Yorke map, Arnold Cat Map.

1 Introduction

In present time, database contains dynamic data for all life fields such as marketing, learning, and medicine. Security is at the beginning of the list of the biggest dangerous issues encountering the database system. Database information is usually utilized and exchanged by different operators, users and beneficiaries. The valuable data that stored in some database are considered to be as an object of attacks and parasitical interfering from internal and external the establishment. Database ciphering provides safer and more secure for the significant information to be in a high level of trust. Therefore, database protection is one of the most in demand challenges in computer science research [1]. Encryption techniques are based on a theoretical or algebraic concepts. Chaos is an exploratory model that points to some complex and unpredictable dynamic phenomena. There are some common features of chaos and encryption. The most notable feature is the sensitivity to the changing in variables and parameters. The remarkable difference between cryptography and chaos is actuality that the chaotic systems are defined only in real numbers [2], while integer numbers are the input of cryptography systems. Anyway, it is believed that these disciplines can benefit from each other. There are many database studies within the safety subject involved functional applications. Despite a number of having a good potential but they still need more developments. W. Xing-hui et al. 2010 [3] suggested a database cipher system depends on the integration of IDEA & RSA hybrid. The role of IDEA is to execute data encryption and decryption operations. The execution is accomplished via 8 iterations of the whole program involving sub-key production. The role of the RSA algorithm is to encrypt the key of IDEA via the public key of the RSA cryptosystem that is passed to the other end. Manivannan et al. 2010 [4], focused especially on protecting the databases having sensitive data by TSFS algorithm containing alteration, replacement, folding and shifting. It's the symmetric-key block encipherment algorithm, with three keys. The generated keys were widened to 12 subkeys through utilizing the key expansion. S.M. Darwish et al. 2014 [5] define a new fuzzy chaos algorithm and cellular automata technique for database encryption. It generates a set of random passwords and uses a

fuzzy logic approach to choose the best password from a set of produced passwords. Then, it generates a key using Cellular Automata. After that, the encrypted of sensitive database fields is chosen by Pseudo-Random Number Generators (PRNGs). The algorithm contains a new fuzzy chaos theory by Takagi-Sugeno fuzzy models to transform discrete-time chaotic systems to separate linear systems. The encryption intends to hide the message signal by making the chaotic carrier hide the encrypted password. V. Galushka et al. 2018 [6] proposed end-to-end data encryption. It was performed in the final nodes of an interaction of the information system using symmetric encryption algorithms. Its key arrangement was intended to be used in multi-user systems. The encryption process is based on username and password, where the password is transformed into an altered password of 128 bits in length by the md5 algorithm. Depending on the distributed key representation model, the first fraction of the key is kept in the database, while the second fraction is gotten by transforming the user's password. After that, it performs the bitwise exclusive-OR operation for the encryption process.

In the present study, a new chaotic DTSXA algorithm for database encryption/decryption is suggested. The suggested algorithm consists of three methods (transposition, substitution, and XORing) which are implemented based on the chaotic system and can encrypt/decrypt English & Arabic database systems.

The rest sections of the paper are arranged as follows: the contribution and feasibility of the proposed system, the basic theory of the chaotic functions, key generation method, the proposed database encryption and decryption algorithm, implementation, and results.

2 Contributions and Feasibility

1. A new, chaotic DTSXA algorithm for database encryption/decryption is proposed.
2. The proposed algorithm consists of novel three methods (Transpose, Substitute, and XORing) implemented based on the chaotic system.
3. The proposed system is capable of encrypting/decrypting the English and Arabic database systems.
4. In addition, the proposed system can encrypt specific tables with N rows and M columns, encrypt only one row in a table, encrypt only one column in a table, encrypt a sequence of rows and columns, and lastly encrypt the query. Therefore can be very useful for handling single records, which can lead to higher accuracy in times.

Feasibility: the proposed DTSXA takes into consideration the feasibility of the practical situations by allowing the secure performance of the three main security properties (confidentiality, integrity, and availability). In general, confidentiality property imposes predefined restrictions on access to protected data and prevents disclosure to unauthorized persons. The integrity property ensures that data cannot be invisibly corrupted. Finally, the property of availability ensures reliable access at an appropriate time to the database. All these properties are provided in the proposed system. Database encryption satisfies the first property via that the encryption process is done only for the users who have the right to do that. As for the second property, it is implicitly verified where only the authorized user can decrypt the data, which ensures the integrity of the data. The last requirement is that all data are available for authorized users, and this feature is provided by the system because the processes of encryption and decryption are carried out only by authenticated users.

3 Chaotic Maps

Chaotic behavior represents a complex dynamic behavior with certain properties, which are possible to be linked to the substitution and permutation attributes in a perfect cipher. Chaotic systems have the following features [7, 8]:

- The system is very critical to initial conditions (Aperiodicity).
- The chaotic system is complex and unpredictable (Nonlinearity).
- System's manner will be altered for any small change in the input (Sensitivity).
- Chaotic system generates identical output if it receives identical input.

In this paper, we used two chaotic maps: 2D Kaplan–Yorke Map and 2D Arnold Cat Map.

3.1 2D Kaplan–Yorke Map

The Kaplan–Yorke map can be defined as a discrete-time dynamical system that shows chaotic behavior. 2D Kaplan–Yorke system maps a point (x_{n+1}, y_{n+1}) depending on another known point (x_n, y_n) through the following equation [9]:

$$\frac{x_{n+1} = 2x_n \ (mod\ 1)}{y_{n+1} = \alpha y_n + cos(4\pi x_n)} \tag{1}$$

Where (mod) is modulo operator. This system relies on just one parameter (α). To prevent the operator to be zero after a relatively few number of iterations, mod 0.99995 is taken instead of mod 1.

3.2 2D Arnold Cat Map

It's a chaotic and reversible map with two-dimensional, which was presented by Vladimir Arnold. The word "cat" was given because of utilizing an image of a cat to show its chaotic behavior. Its equation is given by [10]:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ba+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \ mod(n) \tag{2}$$

Where, (x_{i+1}, y_{i+1}) is the new value of (x_i, y_i), a and b represent the control parameters and n is the number that is used for mod. The control parameters additionally work as secret keys. The inverse of Equation (2) is [11]:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ba+1 \end{bmatrix}^{-1} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} \ mod(n) \tag{3}$$

4 Key Generation Method

The key generation algorithm that generates the key for the proposed database encryption/decryption algorithm is called Chaotic Key Stream Generator (CKNG), which is previously designed based on a method given in [12]. The core of the CKNG is a 2D Henon map and a 2D Rational map. The products of the 2D Hénon and 2D Relational chaotic maps, which are double point digits, are linked by CKNG. These products are transformed into binary sequences of 64-bit. XOR process is used to link the sequences to yield only one binary sequence. The reason behind selecting Hénon chaotic map that has a large key space with the Relational chaotic map was to broaden the complexity of the system and reduplication the difficulty for an attacker to obtain important data.

5 The Proposed Database Encryption and Decryption Algorithm

A new database encrypting and decrypting algorithm called DTSXA (Transposition, Substitution and XORing Algorithm) was proposed based on using

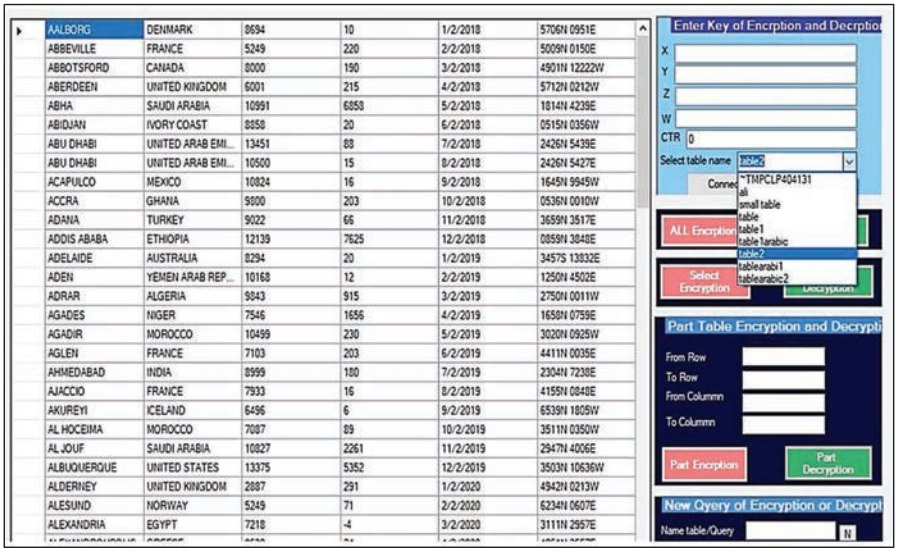


Figure 1 Selecting a table that needs to be encrypted from the database.

chaotic maps. The DTSXA algorithm has two main algorithms: database encryption and database decryption.

A. *Database Encryption Algorithm:* – The suggested DTSXA encryption algorithm consists of the following steps:

1. **Preprocessing step:** open the specified database file and select a table that needs to be encrypted. For example, for a hospital database, the table that needed to be encrypted is a medicine table as shown in Figure 1. The selected table is auto-saved in memory by the Data Table. Data Table (denoted by DT) represents one table of in-memory relational data and temporary storage in RAM and it was treated as a 2D array. Each cell value in DT is converted to its ASCII value and saved into 2D Database Array denoted by (DA[m x n]) as shown in figure.
2. **Key generation step:** in this step, the keys that are needed for the encryption algorithm in all operations are generated and scheduled based on CKNG algorithm. This step involves:
   - First, using the CKNG algorithm with four initial parameters (x1, x2, x3, x4) to generate the key array1 (KA1) with size equal to the size of DA. These initial parameters are double

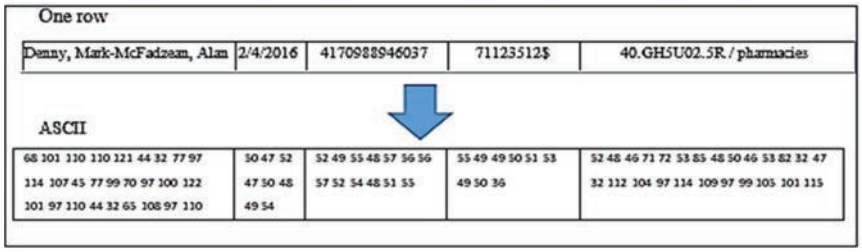


Figure 2 The ASCII values of a row in a certain table.

numbers with accuracy of 10^{-16} and they are treated as the algorithm keys.

- Second, the CKNG algorithm is applied to generate the key array2 (KA2) with size equal to DA size. This is done through using the same initial parameters (x1, x2, x3, x4) after adding any number to them. KA1 and KA2 elements are used to perform the Xoring operation in encryption step.
- Third, the CKNG algorithm is applied with new parameters (newx1, newx2, newx3, newx4) to generate the keys that needed for transposition and substitution operation in encryption step. The parameters (newx1, newx2, newx3, newx4) are created based on the initial parameters (x1,x2,x3,x4) using the Equation (4):

$$\begin{aligned} new_{x1} &= x_1 \oplus x_3 \\ new_{x2} &= x_2 \oplus x_4 \\ new_{x3} &= (x_1 + x_1) \oplus (x_3 * 0.7) \\ new_{x4} &= (x_2 * 0.2) \oplus (x_4 * 0.5) \end{aligned} \tag{4}$$

3. **Encryption step:** For each row in DA array, do the following operation:
   - *Transposition method*: the transposition method permuted the input row values according to Kaplan–Yorke map values. Here, the algorithm iterates the 2D Kaplan–Yorke map in Equation (1) for a number of cycles by as same as the size of the input row. In each iteration, two double numbers are generated and converted to two integer numbers in the range [1 .... Input row length]. The repeated number will be ignored. These numbers represent the new indexes that will be used to

permute the input row, in which every two numbers (as two indexes), exchange their values. This operation needs that the input row length must be an even number. The reason behind that can be explained in the following example. Let the input row is [34, 78, 89, 102, 205, 88, 155, 65, 90]. While the input length is odd, so the value 32d which is the ASCII code of the space bar will be added to the input row as shown [34, 78, 89, 102, 205, 88, 155, 65, 90, 32]. The generated indexes are [4, 7, 2, 6, 9, 1, 3, 10, 8, 5], where the first pair of indexes, 4 and 7, are swapped so the resulted row will become [, , , 155, , , 102, ,]. The second pair of indexes, 2 and 6, are also swapped so the resulted row will become [, 88, , 155, , 78, 102, ,] and so on. The resulted permuted row will be [90, 88, 32, 155, 65, 78, 102, 205, 34, 89]. This process can provide diffusion property. The Pseudo-Code algorithm 1 shows the transposition method.

- *Substitution method*: Arnold's cat map substitutes the pair values of permuted row and generates the substituted row. Each pair values of the permuted row are mapped into a new pair. The substitution method receives permuted row to produce substituted row. Continuing with previous example, one can find the first pair values in the permuted row are v1 = 90 and v2 = 88 and the control parameters for the Arnold cat map are a = 3 and b = 2. The values v1 and v2 are the inputs of Arnold cat map Equation (2) where each value will be inputted to one equation as shown:

$$\begin{aligned} \mathrm{nv}_1 &= (\mathrm{v}_1 + \mathrm{v}_2 * \mathrm{a}) \bmod 256 \\ &= (90 + 88 * 3) \bmod 256 = 98 \\ \mathrm{nv}_2 &= (\mathrm{v}_1 * \mathrm{b} + \mathrm{v}_2 * (\mathrm{a} * \mathrm{b} + 1)) \bmod 256 \\ &= (90 * 2 + 88 * (3 * 2 + 1)) \bmod 256 = 28 \end{aligned}$$

And so on for the rest values in the permuted row. These steps are repeated for all values in the permuted rows to result in the substituted row. To apply the steps to the Arabic language, the range of Arnold cat map [0 ... 255] the English language is exchanged in by a new range [0 . . . 1792] for Arabic language. The pseudo-code is shown in the algorithm 2.

- *XORing method*: – The XORing process is done with twice phases as follows:-
  (a) XORing1: – Each value of the substitution row is XORed with the value of the KA1, which is generated using the CKNG algorithm and producing cipher row1 (CR1).
  (b) XORing2: – Each value of CR1 is XORed with the value of the KA2 which is generated using the CKNG algorithm and producing cipher row2 (CR2) that will be saved in a cipher table.

Algorithm 1 Transposition method

Input: x_0, y_0, a, b // initial and control parameters of Kaplan–Yorke map
M // input vector of dimension (1.. n)
Output: P // output permuted vector of dimension (1.. n)
Begin
Processing:

Step 1: $i = 0$
Step 2: $n =$ length(M)
Step 3: **if** mod (n, 2) $\neq 0$ ▷ // if row length is odd
n=n+1
M[n]=32 ▷ // add ASCII value of space bar
End if
Step 4: Iterate n times
$x_{i+1} =$ mod (2*x_i, 0.99995)
$y_{i+1} =$ mod (a*y_i + cos(4*π*x_0))
V[i] = integer(x_{i+1}* n) ▷ // convert to integer in rang[1..n]
V[i + 1] = integer(y_{i+1}* n) ▷ // convert to integer in rang[1..n]
$i = i + 2$
$x_i = x_{i+1}$
$y_i = y_{i+1}$
End Iteration
Step 5: $i = 0$
Iterate n times
$t_1 =$ V[i]
$t_2 =$ V[i + 1]
P [t1] $\leftarrow$ M [t_2]
P[t2] $\leftarrow$ M [t_1]
i = i+2
End Iteration
End

Algorithm 2 Substitution method

Input: a,b // control parameters of Arnold's cat map
m // the modularity where in English lang. is 256 and in Arabic is 1793.
P // permuted vector of dimension (1.. n)
Output: S // output substituted vector of dimension (1.. n)
Begin
Processing:

Step 1: i = 0
Step 2: Iterate n times
S[i] = mod (P[i] + P[i + 1]*a, m)
S[i + 1] = mod(P[i]*b + P[i + 1](a*b + 1), m)
i = i + 2
End Iteration

B. ***Database Decryption Algorithm***

The decryption algorithm of the proposed DTSXA is an inverse of the encryption algorithm where each operation is reversible. First, for the XORing method, the inverse is achieved by XORing the same key with the cipher data table. Second, for the substitution, the method, inverse Arnold cat map is used in the decryption algorithm. The inverse of the permutation method is archived by returning every value in AD to its real position. The basic steps of the decryption algorithm are:

1. **Preprocessing Step:** Connecting to the database and selecting the cipher table that needs to be decrypted. The selected table is auto-saved in memory by DT and treated as a 2D array. Each value in DT is converted to its ASCII value and saved into 2D array referred as Database Array (DA[m x n]).
2. **Generation of key step:** In this step, the keys that needed in all decryption operations of the algorithm are generated and scheduled based on CKNG algorithm in the same way as the encryption algorithm.
3. **Decryption step**: For each row in DA array, do the following operation:
   - *XORing method*: – The XORing process is performed in the same way as in the encryption except that the ASCII values of each cipher row are XORed first with KA2 and then the result is XORed with KA1. The values of KA1 and KA2 are generated using CKNG in the same way as in the encryption algorithm.

- *Inverse Substitution Method*:- Each pair value in the resulted XORed array is substituted using the inverse 2D Arnold cat map Equation (3). Inverse substitution method is shown in algorithm 3.
- *Inverse transposition method*:- Each pair value in the substituted array returns to its original position. The inverse transposition method is same as the transposition method.

Algorithm 3 Invers substitution method

Input: a,b // control parameters of inverse Arnold's cat map
m // the modularity where in English lang. is 256 and in Arabic is 1793.
P // Xoring vector of dimension (1.. n)
Output: S // output inverse substituted vector of dimension (1.. n)
Begin
Processing:

Step 1: i = 0
Step 2: Iterate n times
S[i] = mod (P[i]*(a*b + 1) − a* P[i + 1]), m)
S[i + 1] = mod (P[i + 1] − b* P[i], m)
i = i + 2
End Iteration

End

C. **Implementation with Examples**

The proposed algorithm was designed to encrypt any English or Arabic database. All the examples of English database are given from Northwind Plus Database through the link https://docs.yugabyte.com/preview/sample-data/northwind/. While we built the Arabic database for the purpose of implementing the encryption process. The proposed DTSXA encrypts the database in five options as shown:

1. Table Encryption: – The encryption algorithm in this option encrypts a whole table, Figures 3 and 4 shows an example of table encryption that has English and Arabic data. To encrypt the whole table, first the initial values and control parameters of the key must be entered to the DTSXA and secondly, the specified table must be selected and then click the Encryption button.
2. One Row Encryption: – After entering the initial values and control parameter of the key, the specified table must be

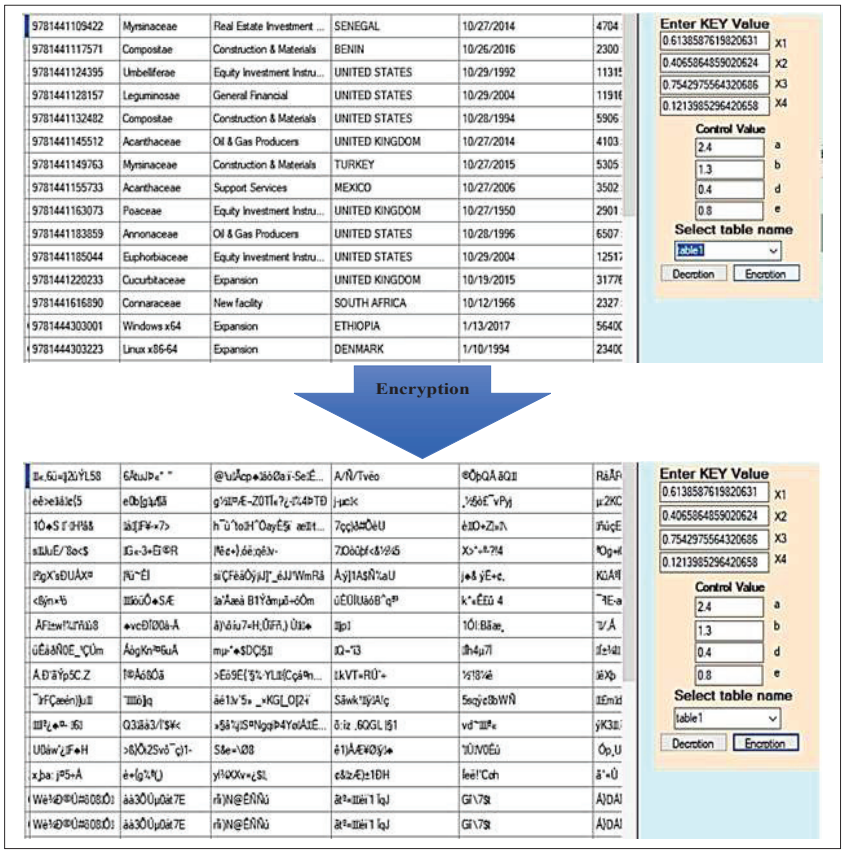


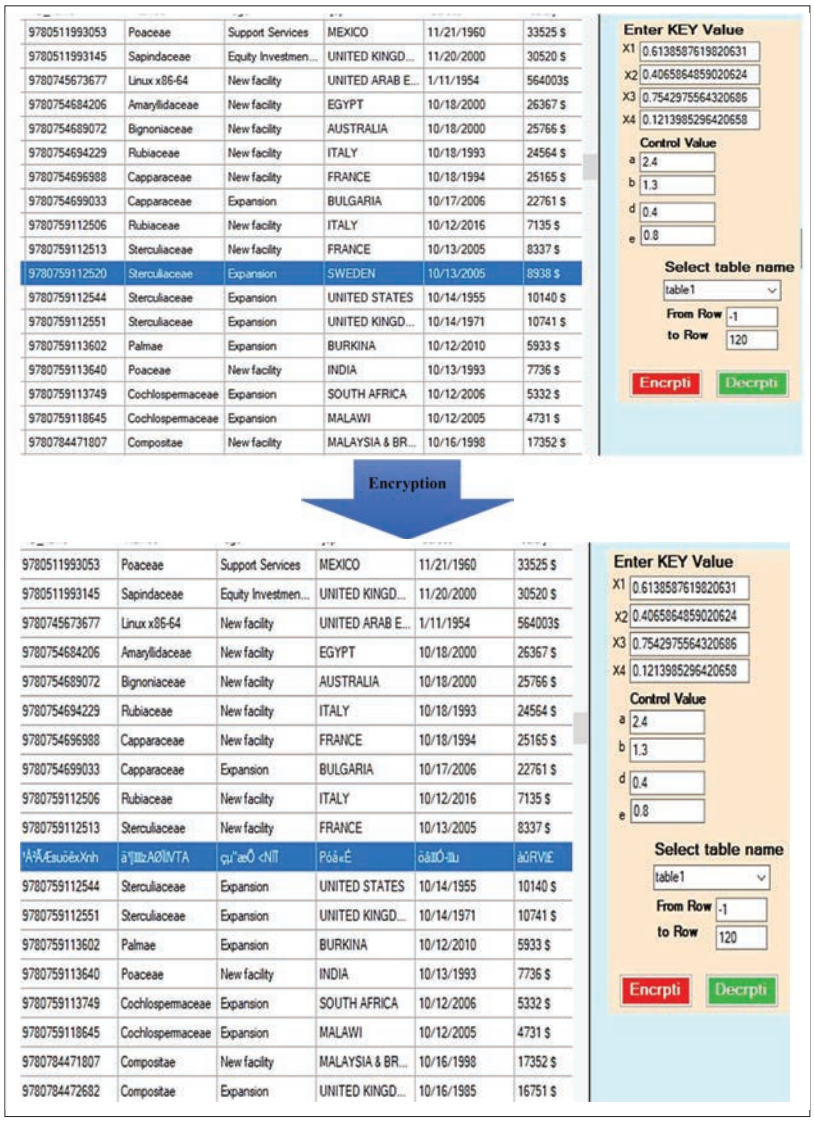
Figure 3 Example of encryption of English whole table.

selected, then one must choose the row and click on the Encryption button. Figure 5 shows the encryption process of a certain selected row in a table.

3. One Column Encryption: – First of all, the initial values and the control parameter of the key are entered then the specified table must be selected. Later the column number must be entered finally click on the Encryption button. Figure 6 shows an example of the column encryption process.
4. Encryption of Rows and Column Sequences: – This option allows to encrypt a number of rows or columns. If one wants to encrypt a sequence of rows, he must select the specified table after entering the initial values and the control parameter

Figure 4 Example of encryption of Arabic whole table.

of key, the range of rows required for encryption will be highlighted. Furthermore, one also needs to press the Encryption button. Figure 7 shows an example for encryption of Row Sequences. Encryption a sequence of columns is conducted in the same way, i.e. the number of columns to be encrypted must be specified and press the encryption button as shown in Figure 8. To encrypt a specific set of columns and rows,



9780511993053	Poaceae	Support Services	MEXICO	11/21/1960	33525 $
9780511993145	Sapindaceae	Equity Investmen...	UNITED KINGD...	11/20/2000	30520 $
9780745673677	Linux x86-64	New facility	UNITED ARAB E...	1/11/1954	564003$
9780754684206	Amaryllidaceae	New facility	EGYPT	10/18/2000	26367 $
9780754689072	Bignoniaceae	New facility	AUSTRALIA	10/18/2000	25766 $
9780754694229	Rubiaceae	New facility	ITALY	10/18/1993	24564 $
9780754696988	Capparaceae	New facility	FRANCE	10/18/1994	25165 $
9780754699033	Capparaceae	Expansion	BULGARIA	10/17/2006	22761 $
9780759112506	Rubiaceae	New facility	ITALY	10/12/2016	7135 $
9780759112513	Sterculiaceae	New facility	FRANCE	10/13/2005	8337 $
9780759112520	Sterculiaceae	Expansion	SWEDEN	10/13/2005	8938 $
9780759112544	Sterculiaceae	Expansion	UNITED STATES	10/14/1955	10140 $
9780759112551	Sterculiaceae	Expansion	UNITED KINGD...	10/14/1971	10741 $
9780759113602	Palmae	Expansion	BURKINA	10/12/2010	5933 $
9780759113640	Poaceae	New facility	INDIA	10/13/1993	7736 $
9780759113749	Cochlospermaceae	Expansion	SOUTH AFRICA	10/12/2006	5332 $
9780759118645	Cochlospermaceae	Expansion	MALAWI	10/12/2005	4731 $
9780784471807	Compositae	New facility	MALAYSIA & BR...	10/16/1998	17352 $

9780511993053	Poaceae	Support Services	MEXICO	11/21/1960	33525 $
9780511993145	Sapindaceae	Equity Investmen...	UNITED KINGD...	11/20/2000	30520 $
9780745673677	Linux x86-64	New facility	UNITED ARAB E...	1/11/1954	564003$
9780754684206	Amaryllidaceae	New facility	EGYPT	10/18/2000	26367 $
9780754689072	Bignoniaceae	New facility	AUSTRALIA	10/18/2000	25766 $
9780754694229	Rubiaceae	New facility	ITALY	10/18/1993	24564 $
9780754696988	Capparaceae	New facility	FRANCE	10/18/1994	25165 $
9780754699033	Capparaceae	Expansion	BULGARIA	10/17/2006	22761 $
9780759112506	Rubiaceae	New facility	ITALY	10/12/2016	7135 $
9780759112513	Sterculiaceae	New facility	FRANCE	10/13/2005	8337 $
¹Ä³ÅÆ±üöëxÄnh	ä¶¶¶zA0¶VTA	çµ"æÖ <N¶	Póå«É	öâ¶Ô-¶u	àúRVÉ
9780759112544	Sterculiaceae	Expansion	UNITED STATES	10/14/1955	10140 $
9780759112551	Sterculiaceae	Expansion	UNITED KINGD...	10/14/1971	10741 $
9780759113602	Palmae	Expansion	BURKINA	10/12/2010	5933 $
9780759113640	Poaceae	New facility	INDIA	10/13/1993	7736 $
9780759113749	Cochlospermaceae	Expansion	SOUTH AFRICA	10/12/2006	5332 $
9780759118645	Cochlospermaceae	Expansion	MALAWI	10/12/2005	4731 $
9780784471807	Compositae	New facility	MALAYSIA & BR...	10/16/1998	17352 $
9780784472682	Compositae	Expansion	UNITED KINGD...	10/16/1985	16751 $

Figure 5 Example of encryption of one row.

the bundle of columns and rows is selected for encryption followed by pressing the Encryption button.

5. Query Encryption: – A query is a request for data or information from a database table or combination of tables. This data

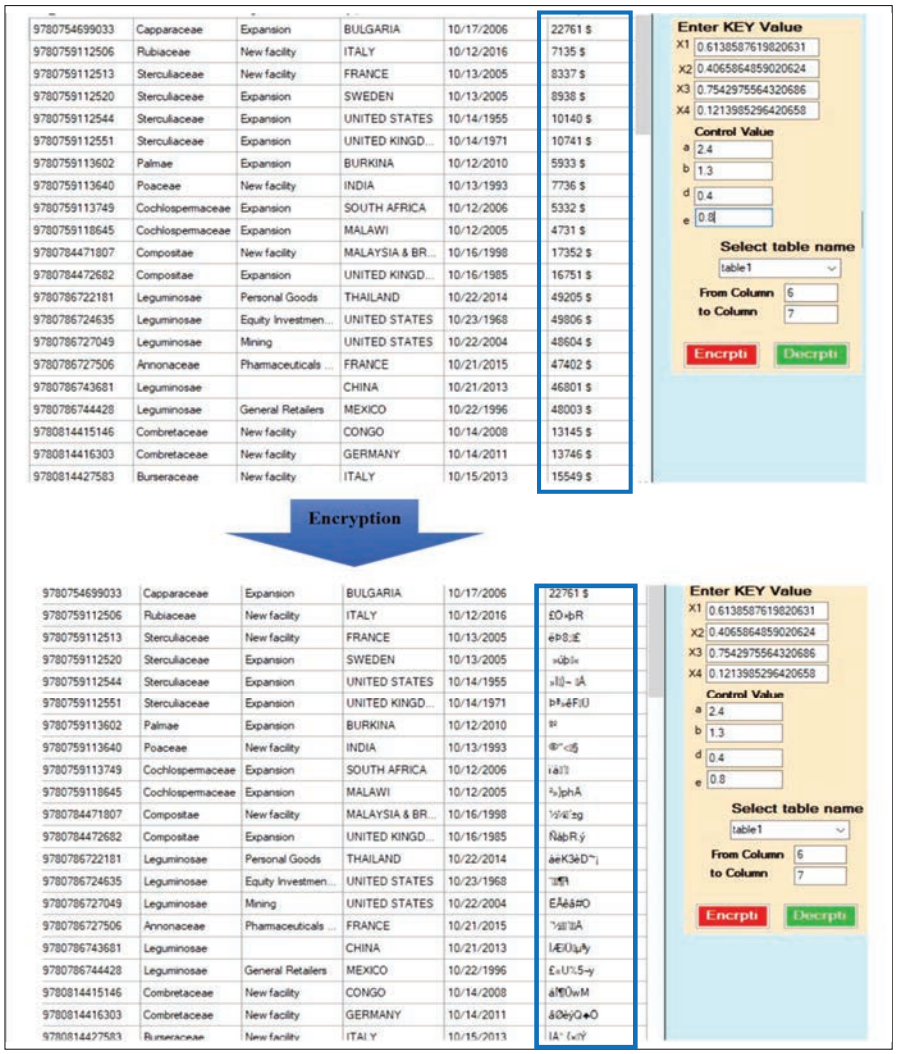


Figure 6 Example encryption one column.

may be generated as results returned by Structured Query Language (SQL), pictorials, graphs or complex results. Figure 9 illustrates the encryption method of query yield. The process begin from the blue field of "New Query" followed by typing query SQL instruction.

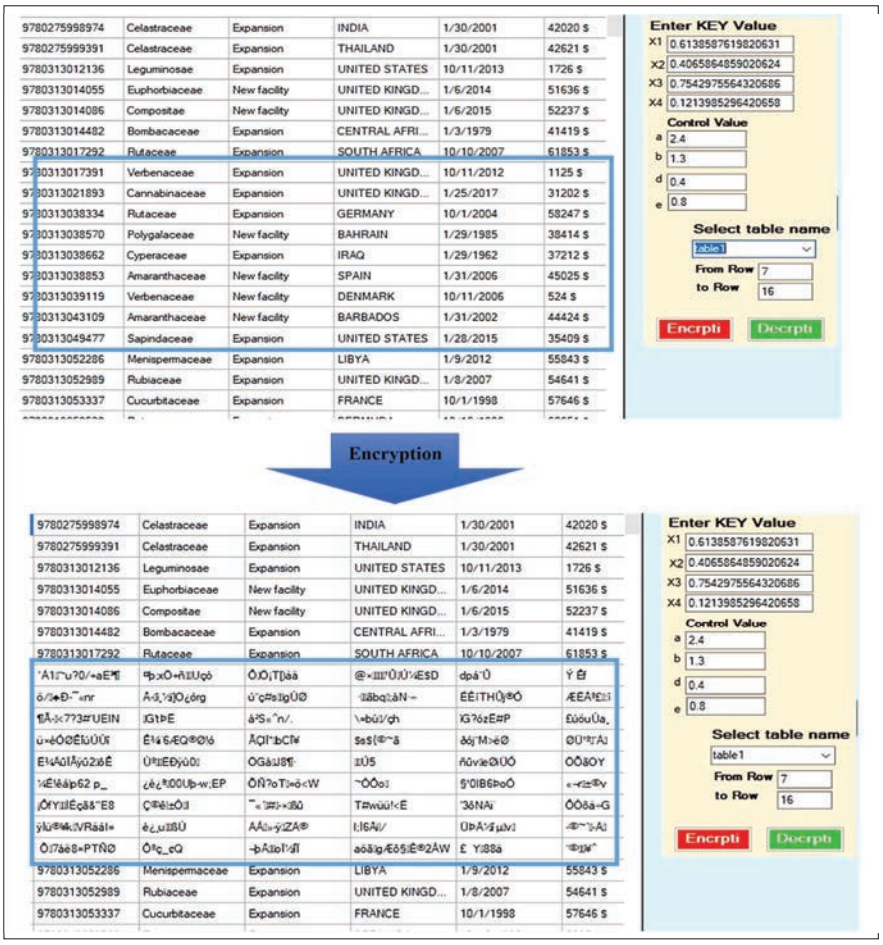


Figure 7 Example encryption of row sequences.

6 Performance Analysis of the Proposed DTSXA

The results of the proposed DTSXA were analyzed through a number of statistical and security tests to sure its performance.

6.1 Brute Force Attack Analyses

Any system can get good resistance to the brute-force attack by making its key space sufficient big (i.e. as a rule, the main space has a key space lower than 2^{128} is not reasonable to be secure enough), otherwise in a limited period

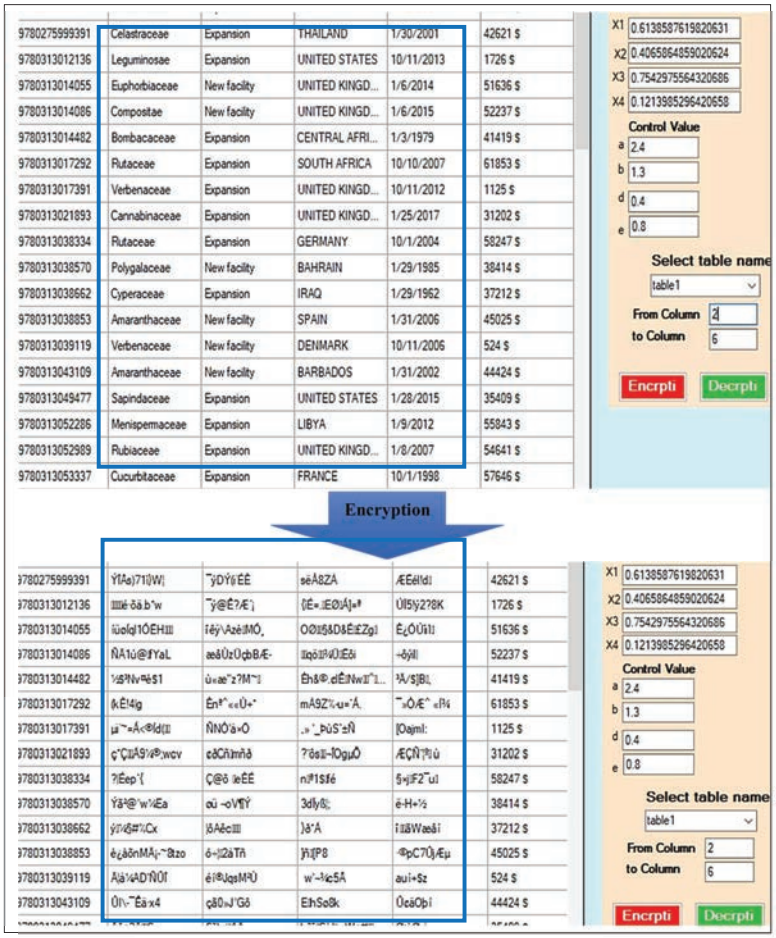


Figure 8 Example encryption of columns sequences.

of time, it will be discovered by somewhat long inspection to obtain the secret key [13, 14]. In this encryption system, the key space is constructed from the initial and control values that are needed for generating the key (CKNG). These parameters are double point number (x_1, x_2, x_3, x_4) and have precision of 10^{-16}. Therefore, the present work has a final key space of 2^{213}. A well prepared database encryption program must have a powerful resistance to different kinds of attacks like brute-force attack and statistical attack. So the security of the proposed DTSXA is analyzed through a number of tests include brute force attack analysis, key sensitivity, statistical attack

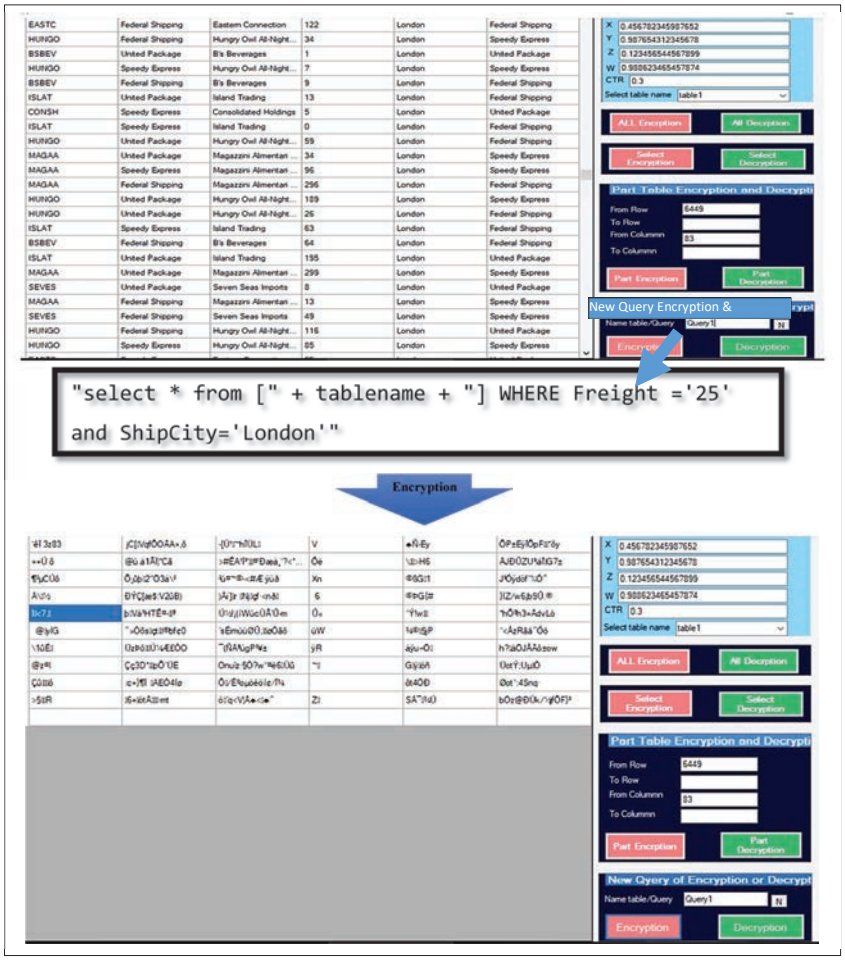


Figure 9 Example of query encryption.

analysis, differential attack analysis, information entropy analysis and mean square error.

6.2 Statistical Attack Analyses

1. Histogram Analysis: – For optimal database encryption, no characteristic distribution mode must be shown on the cipher text histogram. In other words, the cipher text letters must be distributed in an equal way for the total span of the ASCII codes. The cipher crypto-analysis uses the

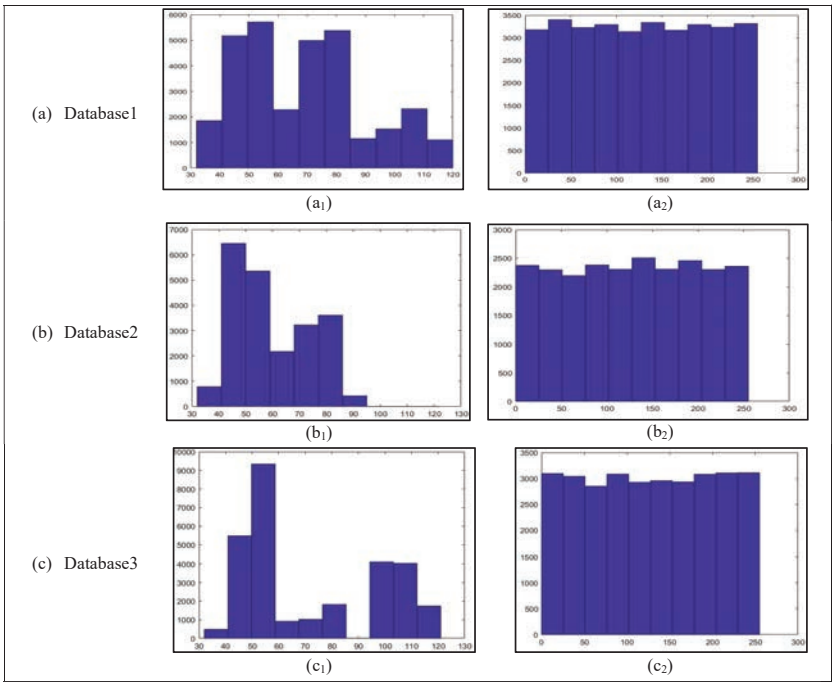


Figure 10 Histogram analysis of the English language. (a_1), (b_1) and (c_1) histograms of plain table that were chosen randomly. (a_2), (b_2) and (c_2) corresponding histograms of cipher table.

histograms gauges to look at the distribution of letters and/or symbols of cipher text and plain-text. We have used various databases in Arabic and English language. Each database contains several types of data like (text, Number, Date) with different sizes of data.

Figure 10 shows the histogram of separate databases in English language that were chosen randomly. Figure 11 shows the histogram of several databases in Arabic language that were also chosen randomly. For example, in Figures 10.a_1 and 11.a_1, one can see that high peaks start at (40 – 60, 70 – 85). While at the lower peaks, differences are also observed. On other the hand, the cipher texts have nearly uniform character distributions.

2. Correlation Coefficient Analysis: – Correlation test is useful to know how to specify the strength of the linear relationship between two sequences. The ways for investigating the correlation of the strings are given below [13, 15]:

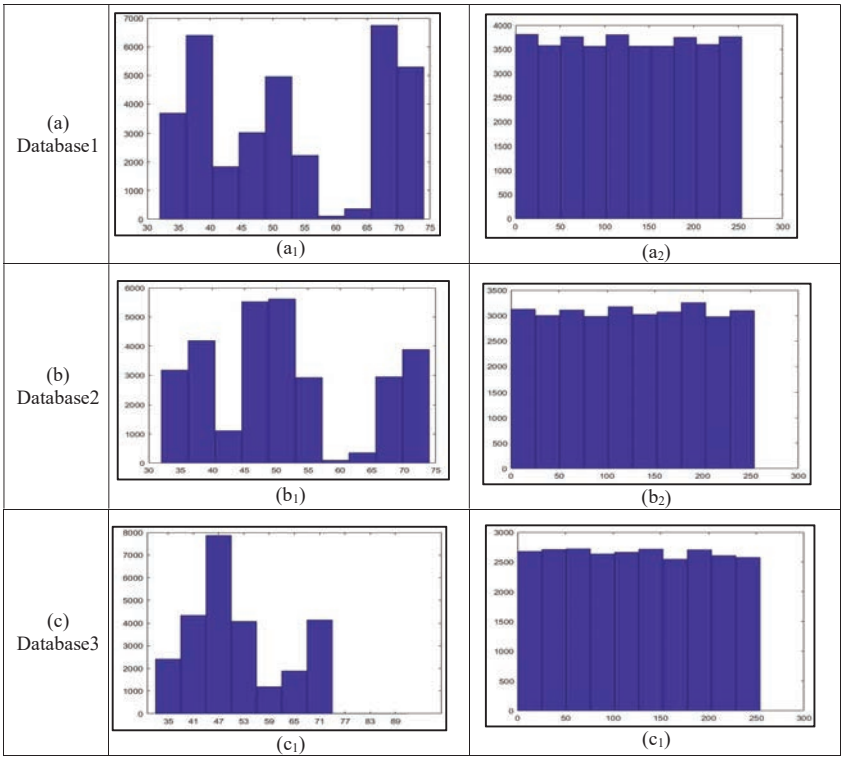


Figure 11 Histogram analysis of the Arabic language. (a_1), (b_1) and (c_1) histograms of plain table that were chosen randomly. (a_2), (b_2) and (c_2) corresponding histograms of cipher table.

- **Pearson's Correlation Coefficient**: The test aims to examine the presence of a relevance that links the sequences in the generated pseudo-random numbers by measuring the Pearson correlation coefficients of every string sequences of the pair. If the chosen pair are: str1 = [x1…,xn] and str2 = [y1…,yn] then:

$$(str1, str2) = \frac{\sum_{i=1}^{n}(x_i - x)(y_i - y)}{\sqrt{[\sum_{i=1}^{n}(x_i - \overline{x})^2][\sum_{i=1}^{n}(y_i - \overline{y})^2]}} \tag{5}$$

$$\text{Where } \overline{x} = \sum_{i=1}^{n} \frac{x_i}{n} \quad \text{and} \quad \overline{y} = \sum_{i=1}^{n} \frac{y_i}{n}$$

- **Hamming Distance:** the test aims to find the difference between two binary strings having same size. Here, hamming distance calculates

Table 1 Results of correlation analysis between plain table and cipher table

	Database	Pearson's correlation Value	Hamming Distance Value
1	English database1	−0.0113	0.9961
2	English database2	−0.0079	0.9958
3	English database3	0.0150	0.9964
4	Arabic database1	0.0118	0.9961
5	Arabic database2	−0.0016	0.9960
6	Arabic database3	−0.0012	0.9960

the numbers of sites that the string bits are different, for example, the number of places containing (0) and that containing (1). It is given by Equation (6) where the series involves the strings (String S1) and (String S2):

$$d(S_1, S_2) = \sum_{j=0}^{m-1} (X_j y_j) \tag{6}$$

The correlation coefficients between the original table and encrypted table, for different tables of different data types, were taken and the results are shown in Table 1. The original tables use databases in English and Arabic. The encrypted database shows that the correlation coefficients approach zero and so the correlation is nearly trivial. These correlation analyses prove that all algorithms correspond to zero correlation and so the attacker could never get any significant information by employing a statistical contravention.

6.3 Information Entropy Analysis

Entropy is the rate at which information is produced by a stochastic source of data. To build a perfect encryption system, the entropy of enciphered data of a table approaches the optimal state is required, and can be calculated according to the equation [13, 15, 16]: –

$$H(m) = \sum_{i=1}^{2^{n-1}} \left(p \, log \frac{1}{p(m_i)} \right) \tag{7}$$

Here, n is the count of bits that given by (m_i) where $m_i \in$ m. p (mi) is the probability of m_i and log () is a logarithm of base 2, therefore the equation outcome will be in bits. For an actual random source producing 2n symbols, the entropy will equal to (n). Table 2 shows the entropy of diverse Arabic and English databases where all values having an entropy close to the ideal value.

Table 2 Information entropy analysis result of cipher table

	Data Base	Entropy Value
1	English database1	8
2	English database2	8
3	English database3	8
4	Arabic database1	7.9
5	Arabic database2	7.9
6	Arabic database3	7.9

6.4 Key Sensitivity Analyses

A simple change in the key will be selected to evaluate the system sensitivity of the proposed encryption system. The procedure will be done for ten different databases that were first encrypted using the encryption key based on the following initial values:

$$\mathrm{X}_1 = 0.6138587619820631,\ \mathrm{X}_2 = 0.4065864859020624,$$
$$\mathrm{X}_3 = 0.7542975564320686,\ \mathrm{X}_4 = 0.1213985296420658.$$

After that, the database will be encrypted using the encryption key with a minimal change. The change is only for the number of rank of 16 and the other fifteen numbers remain unchanged, i.e.:

$$\mathrm{X}_1 = 0.6138587619820632,\ \mathrm{X}_2 = 0.4065864859020625,$$
$$\mathrm{X}_3 = 0.7542975564320687,\ \mathrm{X}_4 = 0.1213985296420659$$

Pearson's correlation coefficient and Hamming distance are calculated between two encrypted databases where they are encrypted using two slightly different key values. Table 3 illustrates the results of the Pearson's correlation coefficient and hamming distance which indicate that the correlation is very small despite the slight change in the basis of key values. This means that the proposed algorithm is sensitive to the secret key and it has the ability to resist mass attack.

6.5 Differential Attack Analysis

It exhibits the effect of variation in the input on the variation in the output. It is a try to get the decryption key using a simple variation in the input. Here, the large enough output variation, the more difficult for an attacker to find the decryption key [13–15]. A reliable cipher system must distribute the effect of

Table 3 Results of the sensitivity of the key between the cipher tables (Cipher1 table and Cipher2 table) which encrypted with slightly different keys

	Database	Pearson's Correlation Value	Hamming Distance Value
1	English database1	0.0230	0.9851
2	English database2	0.0089	0.9882
3	English database3	−0.0071	0.9936
4	Arabic database1	0.0302	0.9574
5	Arabic database2	0.0129	0.9960
6	Arabic database3	0.0139	0.9959

Table 4 Results of differential attack analysis (SAD test) between the Cipher1 table and Cipher2 table

	Data Base	SAD Value
1	English database1	0.329829817
2	English database2	0.331284375
3	English database3	0.274028125
4	Arabic database1	0.2745255208
5	Arabic database2	0.324802734
6	Arabic database3	0.330626171

a single plaintext character on the maximum possible portion of the cipher text. This property makes the statistical entity of the plaintext does not be observed easily and in turn avoids the known-plaintext attack and the chosen-plaintext attack. That implies, if a little chance in the plaintext table can bring about a huge change in the cipher-table, regarding diffusion and confusion, so the differential attack really no longer has its effectiveness and becomes reasonably pointless.

The differential attack is analyzed by calculating the sum of the absolute difference (SAD) between each pair of encrypted databases. The SAD is determined using the following equation [14–16]:

$$SAD(F, F') = \frac{1}{N} \sum_{j=1}^{N} \left(\frac{|F - F'|}{2^8} \right) \tag{8}$$

Where F and F′ are two encrypted databases with the same length N. Through the change of one character in the data of the table, the SAD test is calculated. The result of SAD is shown in Table 4. Since the sum of absolute difference is so approaching the ideal value of 1/3. So, the results on various databases in Arabic and English demonstrate that the proposed system has a high efficiency of resistance to differential attack.

Table 5 Results of MSE for six different databases between plain table and Cipher table

	Data Base	MSE Value in dB
1	English database1	46.96
2	English database2	47.15
3	English database3	48.24
4	Arabic database1	47.09
5	Arabic database2	45.65
6	Arabic database3	47.18

One can observe that SAD value is different for each used database. This variation relates to the initial values that have an important effect on the chaotic map, control parameters, and content of the database where their correlation is another reason for such variation.

6.6 Mean Square Error (MSE)

(MSE) is a statistical method to evaluate a hidden quantity. It calculates the mean of the errors upped to power 2. The error here is the difference between the observed and predicted values. MSE is a risk function, compatible with the expected value of the squared error loss [17, 18]. The encrypted text should have a significant difference with the plain text. This difference is measured by the mean square error function as in Equation (8):

$$MSE = \frac{1}{mn} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \|f(i-j) - g(i-j)\|^2 \tag{9}$$

Where f is the original database table of length m and g the encrypted table of length n. if the MSE value $\geq$ 30 dB, quality variation between the plain and cipher databases is clear. Table 5 shows the MSE for six encrypted databases where all values are close to the ideal value so we can conclude that the difference between cipher and plain data table is evident.

6.7 The Time Complexity for Encryption and Decryption Algorithms

A very important metric for the measurement of encryption algorithms where the Time complexity is the number of operations an algorithm performs to accomplish its task (taking into account that each operation takes the same

amount of time). The most efficient algorithm is the one that performs tasks within the smallest number of operations. However, the complexity of time is also influenced by factors such as the operating system and hardware, but we will not include them in this discussion.

Let us consider a table of N rows and M columns, then the complexity of the encryption algorithm will be considered based on the complexity of the three steps which are Preprocessing step, the Key generation step, and the Encryption step. The complexity of Preprocessing step is O(1). The complexity of the Key generation step can be calculated first for one iteration which is O(64) and for N rows will be O(64N). The complexity of the Encryption step consists of the complexity of the transposition method and substitution method. The transposition method complexity for a row data of size N is O(N), but the complexity for M rows in a table will be $O(N^3)$. A table's substitution method will also be $O(N^3)$. The whole encryption complexity of a table of N rows and M columns is $O(1) + O(64N) + O(N^3) + O(N^3) = \mathbf{O(1)} + \mathbf{O(N)} + \mathbf{O(N^3)}$.

7 Conclusion

In this study paper, a database encryption method was designed depending on a chaotic two-dimensional functions. The chaotic encryption operations lead to enhance two properties (diffusion and confusion) which produce highly secure algorithm. The whole test results proved that high security through its resistance to the different attacks. The proposed database encryption can be applied to any type of database administration language such as SQL Server or Microsoft Access and any size of data. The proposed work treats Arabic language, which suffers from large lacks in cryptographic techniques, as well as dealing with English language. Besides that, DTSXA algorithm can treat individual rows or columns that make it more applicable for diverse users and diverse organizations. Hence the proposed algorithm has high strength and efficiency in database encryption.

Acknowledgement

The authors are grateful and appreciative to their institutes, Mustansiriyah University and University of Technology in Baghdad/Iraq, due to institute roles in supporting and providing academic time for performing the research.

References

[1] P. Singh and K. Kaur, "Database security using encryption," 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015, pp. 353–358, DOI: 10.1109/ABLAZE.2015.7155019.

[2] J J. S. Teh, M. Alawida, and Y. C. Sii, "Implementation and practical problems of chaos-based cryptography revisited," Journal of Information Security and Applications, 2020, vol. 50, no. 102421, pp. 102421, DOI: 10.1016/j.jisa.2019.102421.

[3] X.-H. Wu and X.-J. Ming, "Research of the Database Encryption Technique Based on Hybrid Cryptography," 2010 International Symposium on Computational Intelligence and Design, 2010, pp. 68–71, DOI: 10.1109/ISCID.2010.105.

[4] Manivannan and R. Sujarani, "Light weight and secure database encryption using TSFS algorithm," 2010 Second International conference on Computing, Communication and Networking Technologies, 2010, pp. 1–7, DOI: 10.1109/ICCCNT.2010.5591778.

[5] S. M. Darwish, A. A. El-Zoghabi, and M. A. Abdewi, "Database encryption using fuzzy chaotic," International Journal of Future Computer and Communication, 2014, vol. 3, no. 6, pp. 436–443, DOI: 10.7763/IJFCC.2014.V3.343.

[6] V. V. Galushka, A. R. Aydinyan, O. L. Tsvetkova, V. A. Fathi, and D. V. Fathi, "System of end-to-end symmetric database encryption," Journal of Physics: Conference Series, 2018, vol. 1015, p. 042003, DOI: 10.1088/1742-6596/1015/4/042003.

[7] R. N. AL-Zubaidy and E. Al-Bahrani, "New Key Generation Algorithm based on Dynamical Chaotic Substitution Box," 2018 Al-Mansour International Conference on New Trends in Computing, Communication, and Information Technology (NTCCIT), 2018, pp. 93–98, DOI: 10.1109/NTCCIT.2018.8681187.

[8] A. A. Maryoosh, "A new block cipher algorithm for image encryption based on chaotic system and S-Box", International Journal of Civil Engineering and Technology , 2018, vol. 9, no. 13, pp. 318–327.

[9] M. H. P. Ranmuthugala and C. Gamage, "Chaos theory based cryptography in digital image distribution," in 2010 International Conference on Advances in ICT for Emerging Regions (ICTer), 2010, pp. 32–39, DOI: 10.1109/ICTER.2010.5643275.

[10] F. Chen, K.-W. Wong, X. Liao, and T. Xiang, "Period distribution of generalized discrete Arnold cat map," Theoretical Computer Science,

vol. 552, pp. 13–25, 2014, DOI: https://doi.org/10.1016/j.tcs.2014.08.002.

[11] X. Jin, Y. Chen, S. Ge, K. Zhang, X. Li, Y. Li, Y. Liu, K. Guo, Y. Tian, G. Zhao, X. Zhang, and Z. Wang, Color Image Encryption in CIE L*a*b* Space, International Conference on Applications and Techniques in Information Security (ATIS), 2015, vol. 557, pp. 74–85, DOI: https://doi.org/10.1007/978-3-662-48683-2_8.

[12] N. H. Ghayad and E. A. Albahrani, "A combination of two-dimensional hénon map and two-dimensional rational map as key number generator," in 2019 First International Conference of Computer and Applied Sciences (CAS), 2019 pp. 107–112, DOI: 10.1109/CAS47993.2019.9075731.

[13] Albahrani, A. A. Maryoosh, and S. H. Lafta, "Block image encryption based on modified playfair and chaotic system," Journal of Information Security and Applications, 2020, vol. 51, no. 102445, pp. 1–9, DOI: https://doi.org/10.1016/j.jisa.2019.102445.

[14] K. A. Hussein, S. A. Mahmood, and M. A. Abbass, "A new permutation-substitution scheme based on Henon chaotic map for image encryption," in 2019 2nd Scientific Conference of Computer Sciences (SCCS), 2019, pp. 63–68, DOI: 10.1109/SCCS.2019.8852590.

[15] S. Mahmood and M. S. M. Rahim, "Novel method for image security system based on improved SCAN method and pixel rotation technique," Journal of Information Security and Applications, 2018, vol. 42, pp. 57–70, DOI: https://doi.org/10.1016/j.jisa.2018.08.001.

[16] Albahrani and R. N. Kadhum, "A New Cipher Based on Feistel Structure and Chaotic Maps", Baghdad Science Journal, 2019, vol. 16, no. 1, pp. 270–280, DOI: https://doi.org/10.21123/bsj.2019.16.1(Suppl.).0270.

[17] B. Norouzi, S. M. Seyedzadeh, S. Mirzakuchaki, and M. R. Mosavi, "A novel image encryption based on hash function with only two-round diffusion process," Multimedia Systems, 2014, vol. 20, no. 1, pp. 45–64, DOI: https://doi.org/10.1007/s00530-013-0314-4.

[18] B. D. Parameshachari, Kiran, P. Rashmi, M. C. Supriya, Rajashekarappa, H. T. Panduranga, "Controlled partial image encryption based on LSIC and chaotic map," in Proceedings of the 3rd International Conference on Cryptography, Security and Privacy – ICCSP '19, 2019, pp. 60–63, DOI: https://doi.org/10.1145/3309074.3309107.

[19] J. Xu, B. Zhao, and Z. Wu, "Research on color image encryption algorithm based on bit-plane and Chen chaotic system," Entropy (Basel), 2022, vol. 24, no. 2, p. 186, DOI: 10.3390/e24020186.

Biographies

Ekhlas Abbas Albahrani was born at 1974. She had the PhD Degree in Computer science/Data Security in 2016. The Msc. in Computer science/Data Security in 2001, and Bachelor Degree in Computer science 1996 at the University of Technology/Iraq. Work teams are at University of Technology and Mustansiriyah University, Baghdad, Iraq on Cryptography research field. Academic member from 1996 to 2001 at the University of Technology/Iraq, from 2001 to 2009 at Sebha University and at Mustansiriyah University/Iraq from 2009 to 2021. She participated in different committees such as organizing symposium and conferences in computer Sciences at Mustansiriyah University and IEEE conferences. She participated in different seminars, Training courses and lectures inside Iraq. She reviewed more than 22 articles in data security for Clarivate and Scopus journals. She participated in different organizing and scientific committees of conferences in College of Education at Mustansiriyah University/Iraq. She has about 20 published research papers in Cryptography and Data Security.

Sadeq H. Lafta was born at 1972 he had his PhD Degree in Magnetic Material Science in 2016, the Msc. in Laser Physics 1998, and Bachelor

Degree in Applied Physics 1995 University of Technology/Iraq. His work teams are at University of Technology in semiconductor, nanomaterials and sensor fields, Duisburg-Essen University in nano-magnetic material field and Mustansiriyah University in Cryptography research field. He was an academic member from 1998 to 2009 in Sebha University, researcher at Nanotechnology Centre and Applied Science Department/University of Technology/Iraq from 2010 to 2021. He participated in different committees such as organizing symposium and conferences in Nanotechnology Centre/University of Technology. He participated in revealing the validity of laboratory equipment and suitability for work. He participated in different seminars, Training courses and lectures inside and outside Iraq. He has different social and scientific articles in Applied Science Dep. Website. He reviewed more than 50 articles in different applied physics for Clarivate and Scopus journals. He participated in different organizing and scientific committees of conferences in Nanotechnology Centre Applied Science Dep. At University of Technology/Iraq. He has about 25 published research papers in Applied Physics and Cryptography. He has a Science Day Medal/Ministry of Higher Education/Iraq in 2016 and Inventors Medal/Ministry of Science and Technology/Iraq in 2018.

Naeem Howrie Ghayad, born in 1978 in Baghdad, B.Sc. 2004 in computer science. MSc. In Computer Science from College of Education at Mustansiriyah University/Baghdad in 2019. He has 3 published papers in the field of cryptography. Now, he works in the Ministry of Labour and Social Affairs/Iraq.