Sensors data encryption using TSFS Algorithm

Shatha Habeeb and Dr. Rehab F. Hassan

University of Technology, Computer Sciences Department

E-mail: shathahabeeb@yahoo.com

Abstract

Sensors data security has an increasing importance with the use of sensors in all kinds of real world applications, after being collected from sensors, huge amount of data will be stored in a database, with different types of analysis. It should be protected from all attack kinds considering sensor data as sensitive data. Several security algorithms are implemented to present database protection, and to have powerful protection, the sensitive data should be covered from the risk of being attacked and not stored as original text only, therefore, the key solution for protecting database is to encrypt it. This paper provides database encryption for data acquisition from sensor. TSFS (Transposition, Substitution, Folding, and Shifting) algorithm has been used with three strong and different keys matrices after two sub-steps for key generation using genetic algorithm and expansion by column shifting. By implementing the enhanced algorithm, an efficient and strong database security system has been achieved to data received from sensor.

Keywords: TSFS, Cryptographic**,** Key expansion, Database Security, Sensor Data Encryption.

تشفير بيانات اجهزة الاستشعار باستخدام خوارزمية TSFS

الجامعة التكنولوجية قسم علوم الحاسوب

الخلاصة: أمن بيانات أجهزة الاستشعار لها أهمية تتزايد مع استخدام أجهزة الاستشعار في جميع أنواع التطبيقات في العالم الحقيقي، بعد تجميعها من أجهزة الاستشعار، سيتم تخزين البيانات الضخمة في قاعدة بيانات. ينبغي حمايتها من جميع أنواع الهجمات و تعتبر بيانات الاستشعار بيانات حساسة. يتم تنفيذ العديد من خوارزميات الامنية لتوفير الحماية لقاعدة البيانات, ولها خصائص قوية في حماية البيانات، وينبغي حجب البيانات الحساسة من خطر التعرض للهجوم وليس تخزينها كنص أصلي فقط ، وبالتالي ، فإن الحل الرئيسي لقاعدة البيانات المحمية هو التشفير.

هذا البحث يقدم تشفير قاعدة البيانات المستلمة من أجهزة الاستشعار. وقد استخدمت خوارزمية TSFS مع ثلاثة مفاتيح قوية ومختلفة كما استخدمت المصفوفات لتسهيل التعامل مع هذه المفاتيح بعد ان تم الاستعانة الخوارزمية الجينية لتوليد المفاتيح الثلاثة ولاضافة قوة اكبر تم توسع عن طريق تحويل العمود. ومن خلال تنفيذ الخوارزمية المطورة، تم تحقيق نظام فعال وقوي لأمن قواعد البيانات للبيانات الواردة من أجهزة الاستشعار.

الكلمات المفتاحية: أمن بيانات, التشفير, خوارزمية TSFS, أجهزة الاستشعار

I. Introduction

The growing use of sensors in real world applications and linking these sensors to the Internet has its results obvious online on the Web, moreover, tens of devices can be communicating with each other in a real-time manner. Real world applications may connect and control several sensors and actuators with their data and services to be available at all times. [1]

The sensor can sense the physical world and convert it to electrical signals. There are many existing types of sensors that sense the environmental factors such as light power, temperature, audio waves, pictures, etc. [2] A number of sensor nodes (few tens to thousands) working simultaneously to observe a region to gain data about the environment.[3]

With data gained from sensors a database system can be build. The protection systems of database become critical, any damages or misusing will affect the sensitive data stored in that database and will also affect the entire system and the risk has been increased with the increasing number of database developments.

For implementing database security systems, there are four techniques: DBMS, OS physical security systems and data encrypting. The security techniques are not completely acceptable solutions. A well-ordered light-weight encrypting techniques TSFS algorithm will be used for sensitive data only, an effective implementation of queries will afford and a fast response to the users. TSFS is the symmetric-key block algorithm, similar keys are used for data encrypting and decrypting and its power dependents on the key length. [4]

II. Related work

Several schemes have more efficient and creative executions proposed for database security domain. An efficient light-weight database encryption techniques have been used, TSFS algorithm used with ordinary data and randomly generated key. [4] H. Al-Souly *et al*, used TSFS algorithm by spreading the data-set to separate characters, and by correcting shifting and substitution

operations, by supporting several modulo factors and four sixteen arrays correspondingly to avoid the inaccuracy that arises in the decryption steps [5].

Based on the Chinese Reminder theorem and using strong keys and sub-keys an implementation of encrypting the database structure in [3], a record encryption system for all the levels of columns and rows are implemented. Multilevel access control had been proposed in [6] to improve the security level.

A solution for confidentiality problem, a chip secured data access had been proposed in [7]. It is a quite effective and secure solution, however, still the cost is expensive and it is rather a complicated method.

III. TSFS Algorithm:

With 12 rounds, four types of conversions in TSFS algorithm: Substitution and transposition ciphers are considered the most significant essential techniques in building a new symmetric encryption algorithm. The advantage of this cipher is having the two aspects of security and cryptology, confusion and diffusion. TSFS uses the same sequence of these conversions for both encryption and decryption. Encryption algorithms are known as the ciphers and the opposite ciphers are the decryption algorithms. TSFS algorithms are a not Feistel ciphers, each conversion or set of conversions should be reversible. The keys have to be used in the reverse order also. Figure1 shows (illustrate) TSFS algorithm in general view. [4]

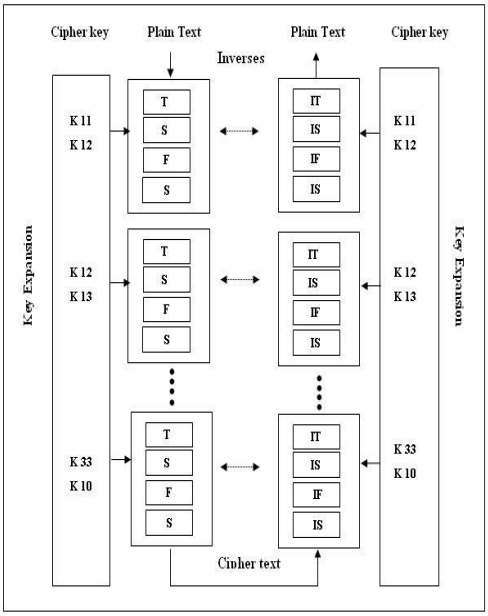

Figure 1: TSFS Algorithm

V. Proposed System

The proposed system collects data from the physical world by a set of sensors and stores it in a database. Then to secure the database by adapting the data in a form that cannot be stated by illegal people. The proposed system is an enhanced form of TSFS algorithms by using genetic algorithm to generate the key and extending it in a way to get strong key matrix. Besides adjusting its transposition and substitution steps, to avoid the error taking place through the decryption process. Experiment results demonstrate the power of the proposed system in the encryption of the sensitive data only**.**

To provide high security, the proposed system is a symmetric algorithm for database encryption with three keys after key expansion technique. For more security and effective system, these three keys had been expanded in twelve sub-keys by using the key expansion method.

1. Cryptographic Key:

Cryptography is often used in security environment to protect data that is sensitive, has a high value, or is susceptible to unauthorized disclosure or unnoticed modification through transmission or storage. It relies upon two basic components: a cryptographic algorithm (methodology) and a cryptographic key. [8]

1.1- Key Generation:

The proposed system generates three keys by using Genetic algorithm and with the help of random number generator to make the key complex. Key generation will go through a number of process and main criteria for key selection will be the fitness value of the population. [9]

Input: Threshold Number

Output: Key Matrix

Process

Generate population number random

Do

Step one

Selection:

1- To breed a new generation selected is a portion of the existing population.

2- Convert sample of the population into binary.

Step two

Crossover: used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, Crossover is a process of taking more than one parent solutions and producing a child solution from them.

Step three

Mutation: alters one or more gene values in a chromosome from its initial state

Until Get all element of key matrix

Algorithm 1: Key Generation

1.2- Key Expansion:

The proposed algorithm generates three keys to be used in 12 rounds. In each round, each key is expanded to several sub-keys. The extension operations for the keys are done by changing the column positions and by using add round key procedure. Here, to get these keys, an arbitrary key generator have to be used only for getting the key values on the first step. Then the key is changed to numbers based on the sequence in the alphabets and stored in 4x4 matrix shape.

The following algorithm shows key extension:

Input Three keys of Array 4x4

Output 12 keys of Array

Process :

Do

Key I

Expanded into key (i1,i2,i3,i4)

For keyI1 - column 1 is not shifted,
column 2 is shifted one position,
column 3 is shifted two position
column 4 is shifted three positions.

For keyI2 - column 1 is shifted one position,
column 2 is shifted 2 positions,
column 3 is shifted three positions,
column 4 not shifted.

For keyI3 - column 1 is shifted two positions,
column 2 is shifted three positions,
column 3 is not shifted,
column 4 is shifted one position.

For keyI4 - column 1 is shifted three positions,
column 2 is not shifted,
column 3 is shifted one position,
column 4 is shifted two positions.

Until End keys

Algorithm 2: Key Expansion

Original Key Expansion to key I ,1	0	1	2	3
Original Key Expansion to key I ,2	1	2	3	0
Original Key Expansion to key I ,3	2	3	0	1
Original Key Expansion to key I ,4	3	0	1	2

Figure 1 Role of shift algorithm of three key

Key1

9	5	4	5
5	1	4	1
9	1	4	2
2	3	1	4

11

9	1	4	4
5	1	1	5
9	3	4	1
2	5	4	2

12

5	1	1	5
9	3	4	1
2	5	4	2
9	1	4	4

13

9	3	4	1
2	5	4	2
9	1	4	4
5	1	1	5

14

2	5	4	2
9	1	4	4
5	1	1	5
9	3	4	1

Key2

6	8	5	5
3	2	4	3
4	9	1	2
5	4	3	0

21

6	4	1	0
3	8	3	5
4	2	5	3
5	9	4	2

22

3	8	3	5
4	2	5	3
5	9	4	2
6	4	1	0

23

4	2	5	3
5	9	4	2
6	4	1	0
3	8	3	5

24

5	9	4	2
6	4	1	0
3	8	3	5
4	2	5	3

Key3

9	2	2	9
8	0	2	9
4	5	6	6
7	7	0	6

31

9	7	4	8
8	9	7	4
4	8	9	7
7	4	8	9

32

7	4	8	9
9	7	4	8
8	9	7	4
4	8	9	7

33

4	8	9	7
7	4	8	9
9	7	4	8
8	9	7	4

34

8	9	7	4
4	8	9	7
7	4	8	9
9	7	4	8

Figure.2 examples view key expansion process

2. Transposition

Transposition cipher is a significant type of traditional ciphers. It does not perform a replacement of one representation with another, alternatively modifies the position only. The symbol in the first position of the plain-text may occur in different position of the cipher text. That means transposition ciphers rearranges symbols. TSFS algorithm uses ***zigzag*** diagonal transposition for storing data into 4 x 4 matrix forms. Figure 3 shows transposition procedure with example.

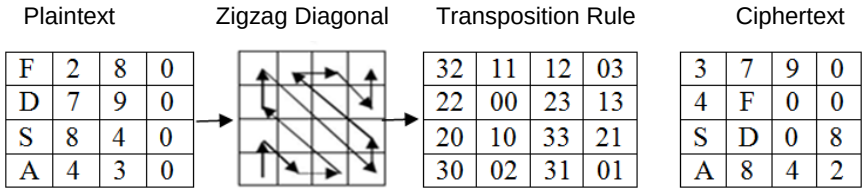

Figure 3: Transposition Procedure

INPUT: In data is 4x4 matrixes that contain the data should be encrypted.

OUTPUT: matrix changing symbols location.

```
Process Matrix out;
        Out_data [0,0] = In_data[3,2];
        Out_data [0,1] = In_data[1,1];
        Out_data [0,2] = In_data [1,2];
        Out_data [0,3] = In_data [0,3];
        Out_data [1,0] = In_data [2,2];
        Out_data [1,1] = In_data [0,0];
        Out_data [1,2] = In_data [2,3];
        Out_data [1,3] = In_data [1,3];
        Out_data [2,0] = In_data [2,0];
        Out_data [2,1] = In_data [1,0];
        Out_data [2,2] = In_data [3,3];
        Out_data [2,3] = In_data [2,1];
        Out_data [3,0] = In_data [3,0];
        Out_data [3,1] = In_data [0,2];
        Out_data [3,2] = In_data [3,1];
        Out_data [3,3] = In_data [0,1];
        return Out_data;
End transposition
```

Algorithm 3: Transposition

3. Substitution

The next stage is substitution conversion. It changes each data matrix component with another one by implementing equation (1). Each character is replaced with another character only, and if the element is a number, it should be changed with a number. The encryption Equation (1) *E(x)* for every given character x is:

$$E(x) = ((key1 * p) \bmod M + key2) \bmod M \qquad (1)$$

Where p is the plaintext, key1 and key2 represent the keys that should be in the same location as p, M is the size of modulo process. TSFS algorithm takes two values for M unlike the original TSFS algorithm which takes one value only. Substitution procedure illustrated in equation (1) has confusion if the data is collected of numeric digits and alphabetic and M is equal to 26 for any digit, as exemplified in the next example. If plain data was 6, key1=1, key2=7, M = 26, then the result is 13. M in the proposed TSFS algorithm is equal to 26 if p represents aliphatic, and 10 if p equal to any numerical value. [5]

Decryption Equation (2) ***D(E(x))*** can be written as follows:

$$D(E(x)) = (((E(x) - key2) \bmod M) / key1) \bmod M \qquad (2)$$

Substitution procedure can be considered as poly-alphabetic or mono-alphabetic ciphers. [4]

3	7	9	0
4	F	0	0
S	D	0	8
A	8	4	2

→

17	9	14	9
18	J	5	6
C	k	8	11
L	14	12	8

Figure 4 Substitution

4. Folding

The next stage use folding procedure. Folding is considered as a transposition cipher, the matrix elements are folded vertically, horizontally, and diagonally. The folding procedure mixes up the data from one location to other one. Figure 5 shows the result of folding procedure. [4,5]

17	9	14	9
18	J	5	6
C	k	8	11
L	14	12	8

→

8	14	12	L
6	11	K	18
8	5	J	C
9	9	14	17

Figure 5 Folding

```
Algorithm folding (Matrix data)
Input: In_data is 4x4 matrix get from substitution technique.
Output: data is data matrix after applying folding technique.
Matrix Out_data;
        Out_data [0,0] = In_data [3,3];
        Out_data [0,1] = In_data [3,1];
        Out_data [0,2] = In_data [3,2];
        Out_data [0,3] = In_data [3,0];
        Out_data [1,0] = In_data [1,3];
        Out_data [1,1] = In_data [2,2];
        Out_data [1,2] = In_data [2,1];
        Out_data [1,3] = In_data [1,0];
        Out_data [2,0] = In_data [2,3];
        Out_data [2,1] = In_data [1,2];
        Out_data [2,2] = In_data [1,1];
        Out_data [2,3] = In_data [2,0];
        Out_data [3,0] = In_data [0,3];
        Out_data [3,1] = In_data [0,1];
        Out_data [3,2] = In_data [0,2];
        Out_data [3,3] = In_data [0,0];

Return Out_data;
End folding
```

Algorithm 4 folding

5. Shifting

The final stage of TSFS algorithms is shifting conversion, which offers an easy method for data encryption using a 16 element in array of numeric digits to replace the symbol with other one. Each array element contains a numeric representation of the data. Every digit must only appear once in each array element in any order. [4,11] In shifting procedure, within its array element, every element is replaced in the data matrix by its location.[5]

Each array element contains 26 numeric digits from 0 to 25. For shifting decryption procedure, the location is given as an input considered on the position of the data, the data represents the plain text of the given cipher-text.

This process is illustrated in the following figure 6.

I/p	Array element	O/p
3	0 1 2 3 4 5 6 7 8 9 10.....22 23 24 25	3
17	1 2 3 4 5 6 7 8 9 10 11 ..... 23 24 25 0	16
7	2 3 4 5 6 7 8 9 10 11 12.....23 24 25 0 1	5
•	•	•
•	•	•
•	•	•
F	13 14 15 16 17 18 .....7 8 9 10 11 12	S
14	14 15 16 17 18 19 ..... 9 10 11 12 13	0
O	15 16 17 18 19 20 ..... 10 11 12 13 14	Z

Figure.6 Shifting

All above encryption procedures form the first round only from TSFS algorithm. And output of the first round is the input to the second round, output of the second round is the input to the third round, and so on. The round is repeated 12 times, the cipher-text of the given plain text is the last round output and that cipher-text is stored in the database. Decryption Algorithm is the opposite procedure of the Encryption. [4,10]

VI. Conclusion

The exchanging and storing of sensor data between networks is increasing rapidly around the world. Attacks at that data raise the danger of data discovery. Many organizations must deal with regulation and legislation on data confidentiality. In this environment, the proposed security development contains an approach for protecting sensitive data against any attacker. The experimental result shows that TSFS algorithm with key generated by using genetic algorithm profitably important cipher, in addition to alphanumeric data. Improved performance approaches without compromising query processing time or database size.

Reference

1- **Sensor Similarity Search in the Web of Things**. Cuong Truong, Kay Römer, Kai Chen. Institute of Computer Engineering,2012.

2- **Application based Study on Wireless Sensor Network**. Kiran Maraiya, Kamal Kant, Nitin Gupta, International Journal of Computer Applications *Volume 21– No.8, May 2011*.

3- **Wireless sensor network survey**. Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal *. Department of Computer Science, Computer Networks 52 (2008) 2292–2330.

4- **Light Weight and Secure Database Encryption**. Using TSFS Algorithm. D. Manivannan1, R.Sujarani2. 1Asst. Professor, School of Computing, International Conference on Computing Communication and Networking 2014.

5- **Lightweight Symmetric Encryption Algorithm for Secure Database**. Author 1: Hanan A. Al-Souly. Author 2: Abeer S. Al-Sheddi. Author 3: Heba A. Kurdi. *Science and Information Conference 2013.*

6- **Wireless underground sensor networks: research challenges,** Akyildiz, E.P. Stuntebeck, Ad-Hoc Networks 4 (2006) 669–686.

7- **System-Architectures for Sensor Networks Issues, Alternatives, and Directions"**J. Feng, F. Koushanfar, M. Potkonjak, IEEE International Conf on Computer Design (ICCD), Germany, 2002. pp. 226- 231.

8- **Recommendation for Cryptographic Key Generation,** NIST SP 800-133...http:// csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general, 2012.

9- **Key Generation Using Genetic Algorithm** for Image Encryption. Aarti Soni1, Suyash Agrawal2RESEARCH ARTICLE. © 2013, IJCSMC All Rights Reserved. 376.

10- **Secure Database Encryption in Web Applications,** Amandeep kaur1 , Mrs. Shailja Kumari2 Oct 13, 2016.