2023 العدد 10 لسنة
No 10 August 2023
المجلة العراقية للبحوث الانسانية والاجتماعية والعلمية
Iraqi Journal of Humanitarian, Social and Scientific Research
Print ISSN 2710-0952-Electronic ISSN 2790-1254

TSFS Key Creation Utilizing Tent Chaos Map and Logistic

Haneen H. Alwan[1], Sabreen J. Bani [2,] Bushra K.Hilal [3], Salah A. Albermany [4] ,Ammar W.Altaher[5] .

Department of Information Technology, Technical College of Management/ Al-Furat Al-awsat Technical University [1,2,5]
Department of Computer information systems , College of Computer Science and information technology /University of Al- Qadisiyah[3]
Department of Computer Science, College of Computer Science and Mathematics / University of Kufa[4]

Haneen.hamza1@atu.edu.iq[1]
sabreen.bani@atu.edu.iq[2]
Bushra.k.h@qu.edu.iq[3]
Salah.albermany@uokufa.edu.iq[4]
dr.ammar@atu.edu.iq[5]

Abstract: The Transposition-Substitution-Folding-Shifting encryption technique (TSFS) is a well-organized, lightweight encryption method that is only utilized for sensitive data. It allows for efficient query implementation and quick user response. The symmetric-key block algorithm known as TSFS depends on the key length and uses identical keys for data encryption and decryption. The primary issue with the TSFS technique is that because the keys are produced statically, an attacker has a chance to crack it and obtain sensitive data from the database. Another issue with the TSFS algorithm is that it requires rapid key generation because it is used in IoT and other applications that demand speed. A suggested key generating mechanism is described in this paper. The logistic and tent maps, two of the chaotic maps, are used in the suggested adaptation. The keys were produced using the logistic chaos map, then the keys were expanded using the tent chaos map.

Keywords : TSFS Encryption Chaos map Logistic map.

إنشاء مفتاح TSFS باستخدام خريطة فوضى الخيمة واللوجستية

حنين حمزة علوان 1 ، صابرين جاسم باني 2 ، بشرى كامل هلال 3 ، صلاح عبدالهادي البيرماني 4 ، عمار وسام الطاهر5
1,2,5 قسم تقنيات المعلومات /الكلية التقنية الادارية / جامعة الفرات الاوسط
3 قسم نظم المعلومات الحاسوبية / كلية علوم الحاسوب وتكنولوجيا المعلومات / جامعة القادسية
5 قسم علوم الحاسوب / كلية علوم الحاسوب والرياضيات / جامعة الكوفة

الخلاصة : تقنية التشفير TSFS (Transposition-Substitution-Folding-Shifting) هي طريقة تشفير جيدة التنظيم وخفيفة الوزن تُستخدم فقط للبيانات الحساسة. يسمح بتنفيذ الاستعلام الفعال والاستجابة السريعة للمستخدم. تعتمد خوارزمية كتلة المفتاح المتماثل المعروفة باسم TSFS على طول المفتاح وتستخدم مفاتيح متطابقة لتشفير البيانات وفك تشفيرها. تكمن المشكلة الأساسية في تقنية TSFS

في أنه نظرًا لإنتاج المفاتيح بشكل ثابت ، فإن المهاجم لديه فرصة لتكسيرها والحصول على بيانات حساسة من قاعدة البيانات. هناك مشكلة أخرى في خوارزمية TSFS وهي أنها تتطلب إنشاء مفتاح سريعًا لأنها تُستخدم في إنترنت الأشياء والتطبيقات الأخرى التي تتطلب السرعة. تم وصف آلية توليد المفاتيح المقترحة في هذه الورقة. تُستخدم The logistic and tent maps ، وهما اثنتان من الخرائط الفوضوية ، في التعديل المقترح. تم إنتاج المفاتيح باستخدام the logistic chaos map ، ثم تم توسيع المفاتيح باستخدام the tent chaos map .

الكلمات الرئيسية TSFS :التشفير خريطة الفوضى الخريطة اللوجستية

1. Introduction

Database encryption is a crucial defense against intrusions and unauthorized access [1]. The Transposition-Substitution-Folding-Shifting (TSFS) encryption algorithm is a symmetric database encryption system that combines three keys and an expansion mechanism to ensure high security. By encrypting only the important data, it speeds up query execution time. It only applies, though, to alphanumeric characters [2]. Substitution and transposition ciphers are said to be the most important fundamental strategies in creating a new symmetric encryption algorithm. The TSFS algorithm has 12 rounds and four types of conversions. This cipher has the dual benefits of confusion and diffusion in terms of security and cryptology. Both the encryption and decryption processes used by TSFS follow the same order of these conversions. The ciphers are encryption methods, while the reverse ciphers are decryption techniques. Since each conversion or group of conversions should be reversible, TSFS algorithms are not Feistel ciphers. Moreover, the keys must be used in reverse. The TSFS algorithm is depicted (illustrated) in broad perspective in Figure (2) [3].

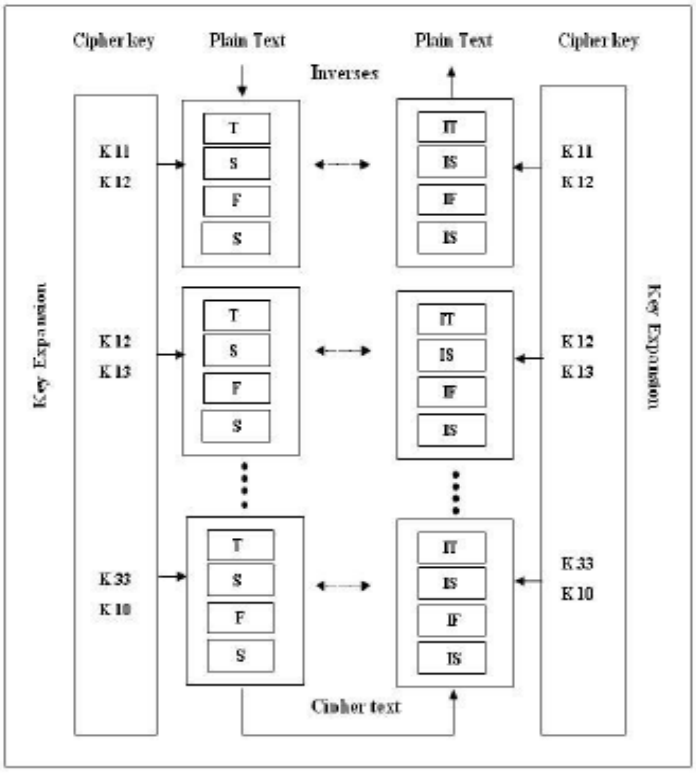

Figure (1): TSFS Algorithm

The logistic map is a polynomial mapping (or recurrence relation) of degree 2 that is frequently used as a paradigmatic illustration of how extremely straightforward non-linear dynamical equations can give rise to complicated, chaotic behavior [4][5].

$$x_{n+1} = r x_n (1 - x_n)$$

a piecewise linear, one-dimensional map with chaotic dynamics on the range [0,1], given by [6].

$$x_{n+1} = \mu\left(1 - 2\left|x_n - \tfrac{1}{2}\right|\right).$$

The first few iterations of (1) give

$$x_1 = -\mu\left(2\left|x_0 - \tfrac{1}{2}\right| - 1\right)$$
$$x_2 = -\mu\left(2\left|\mu\left(1 - 2\left|x_0 - \tfrac{1}{2}\right|\right) - \tfrac{1}{2}\right| - 1\right)$$
$$x_3 = -\mu\left(2\left|\mu\left(2 - \left|\mu\left(1 - 2\left|x_0 - \tfrac{1}{2}\right|\right) - \tfrac{1}{2}\right|\right) - \tfrac{1}{2}\right| - 1\right),$$

2. Related Work

A method of encrypting the database structure is implemented in [7] utilizing strong keys and subkeys using the Chinese Reminder theorem.

The proposed method adjustment focuses on the encryption of large data while taking into account a wide variety of unique characters, and an arbitrary generator is used for producing keys in the substitution stage. In [8] they imperative on information size and amount of exceptional characters.

For data capture from sensors, database encryption is offered in [9]. Three strong and distinct key matrices have been employed with the TSFS (Transposition, Substitution, Folding and Shifting) algorithm following two sub-steps for key generation using genetic algorithm and enlargement by column shifting. A reliable and effective database security solution for sensor data has been achieved by adopting the improved algorithm.

3. The Proposed Modification Description

The majority of the issues with the TSFS Encryption algorithm relate to the keys that are used. Since the attacker must be prevented from knowing these keys, they must be safeguarded and produced dynamically. Additionally, as the TSFS encryption technique is thought of as a lightweight encryption algorithm that requires speed in its operation, these keys must be created quickly. Moreover, the TSFS method is useful in IoT applications since it is fast. This document presents a suggested key generation method. The logistic and tent chaos maps are employed in the suggested keys generating approach. The ability to generate TSFS keys in a dynamic and unpredictable manner makes chaos maps useful. The proposed technique stages are shown in Algorithm (1) and Figure (1).

Algorithm (1): Proposed Keys Generation and Expansion Algorithm

Input: Logistic chaos map data ($X_{0(1)}$, $r_{0(1)}$, $X_{0(2)}$, r2, $X_{0(3)}$, r3)

Tent chaos map data ($X_{0(1)}$, μ_1, $X_{0(2)}$, μ_2, , $X_{0(3)}$, μ_3)

Output: Encryption Keys after Expansion

Process:

S1: For i=1 to 3

If i=1 then apply logistic chaos map ($X_{0(1)}$, $r_{0(1))}$ to generate 4*4 K1 matrix

Else If i=2 then apply logistic chaos map ($X_{0(2)}$, $r_{0(2))}$ to generate 4*4 K2 matrix

Else If i=3 then apply logistic chaos map ($X_{0(2)}$, $r_{0(2))}$ to generate

4*4 K3 matrix

End

S2: For i=1 to 3

If i=1 then apply tent chaos map ($X_{0(1)}$, μ_1) to generate 4*4 key 1 modification matrix

Else If i=2 then apply tent chaos map ($X_{0(2)}$, μ_2) to generate 4*4 key 2 modification matrix

Else If i=3 then apply tent chaos map ($X_{0(3)}$, μ_3) to generate 4*4 key 3 modification matrix

End

S3: For i=1 to 3

a. If i=1 then

i. Swap main diagonal with the second diagonal of key1 modification matrix

ii. Rearrange key1 matrix according to 3.1.1 to get k11

iii. Swap third row with the third row of key1 modification matrix

iv. Rearrange key1 matrix according to 3.1.3 to get k12

v. Swap first row with the second row of key1 modification matrix

vi. Rearrange key1 matrix according to 3.1.5 to get k13

vii. Swap third row with the forth row of key1 modification matrix

viii. Rearrange key1 matrix according to 3.1.7 to get k14

End

b. Else if i=2 then

i. Swap main diagonal with the second diagonal of key2 modification matrix

ii. Rearrange key2 matrix according to 3.1.1 to get

k21

iii. Swap third row with the third row of key1 modification matrix

iv. Rearrange key2 matrix according to 3.1.3 to get k22

v. Swap first row with the second row of key1 modification matrix

vi. Rearrange key2 matrix according to 3.1.5 to get k23

vii. Swap third row with the forth row of key1 modification matrix

viii. Rearrange key2 matrix according to 3.1.7 to get k24

End

c. Else if i=3 then

i. Swap main diagonal with the second diagonal of key3 modification matrix

ii. Rearrange key3 matrix according to 3.1.1 to get k31

iii. Swap third row with the third row of key1 modification matrix

iv. Rearrange key3 matrix according to 3.1.3 to get k32

v. Swap first row with the second row of key1 modification matrix

vi. Rearrange key3 matrix according to 3.1.5 to get k33

vii. Swap third row with the forth row of key1 modification matrix

viii. Rearrange key3 matrix according to 3.1.7 to get k34

End

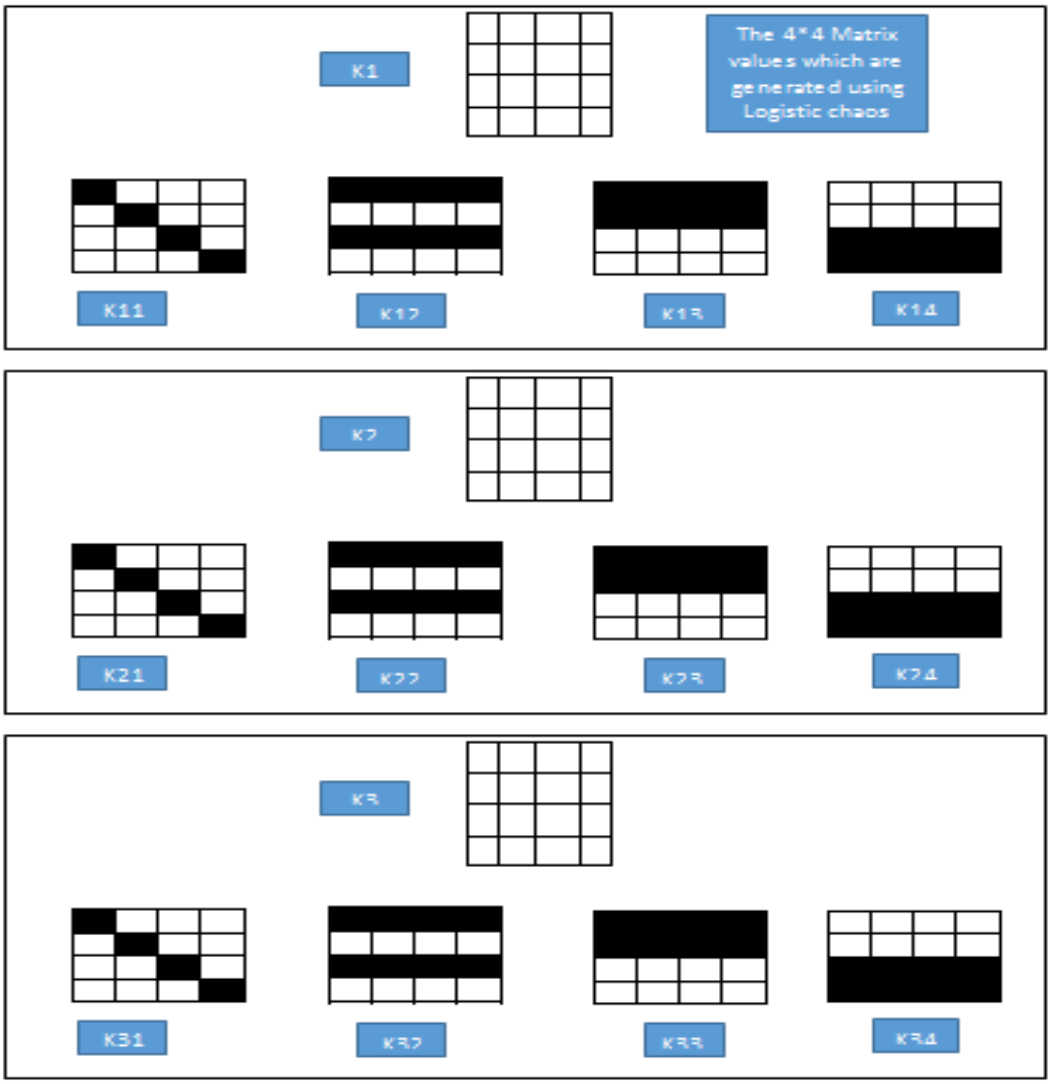

Figure (2) : The Proposed TSFS Keys generation Process

4. Analysis of the proposed modification

4.1 Sensitivity to the starting condition:

Any modifications to the initial circumstances will result in modifications to the system's state. Sensitive on starting conditions is what this is.

We used two initial circumstances in logistic and tent maps, x0 = [0.08, 0.37] and x00 = [0.2, 0.07], which varied slightly from x0, to demonstrate how sensitive the keys are to initial conditions. Plotting shows the solutions for both initial conditions.

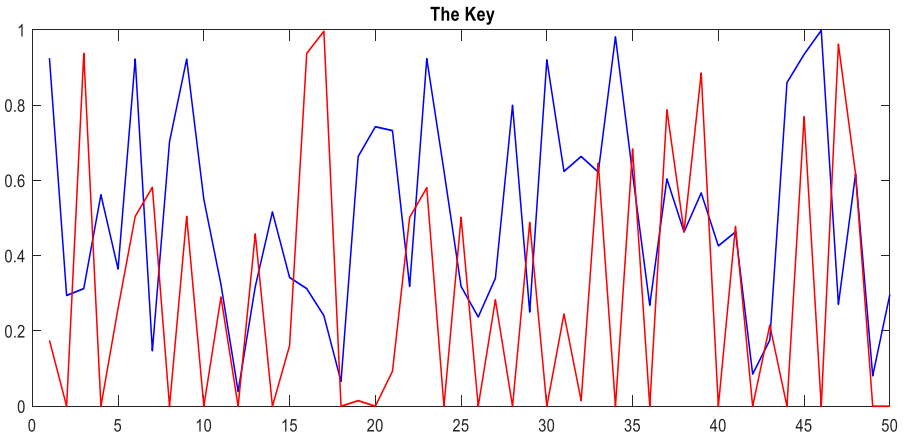

Figure (3) : Sensitivity to initial condition

4.2 Parameter variation

Use chaotic maps to produce random numbers because they are sensitive to parameter changes and depend on them. As a result, any change in the parameter will quickly affect the properties of the map. We utilized two parameters in the logistic and tent maps, r0 = [2, 1] and x00 = [4, 0.4], which differ somewhat from r0, to demonstrate how sensitive the keys are to parameters. Plotted are the solutions for both parameters.

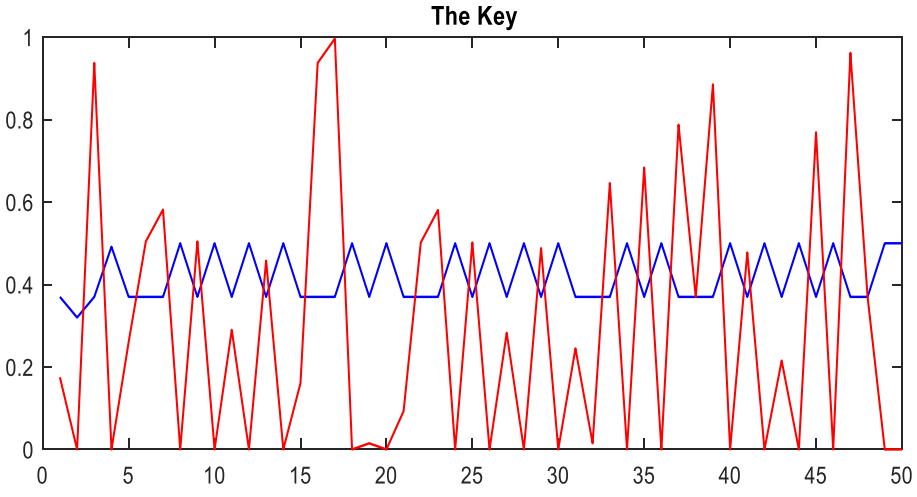

Figure (4) : Sensitivity to Variation of parameters

4.3 Complexity of keys

The complexity of chaotic systems can be used to describe their dynamics. The difficulty of a key chaotic sequence is calculated using the spectral entropy (SE) algorithm and the C0 algorithm.

The Fourier transform is the foundation of the spectral entropy (SE) technique, which is used to analyze chaotic systems' complexity and EEG signals. The fast Fourier transform (FFT)-based C0 method is used to analyze the complexity of biomedicine, the electrical signal in the brain, the optical images of the brain, and chaotic systems.

When a chaotic sequence is more complicated, it is more secure since it is closer to a random sequence.

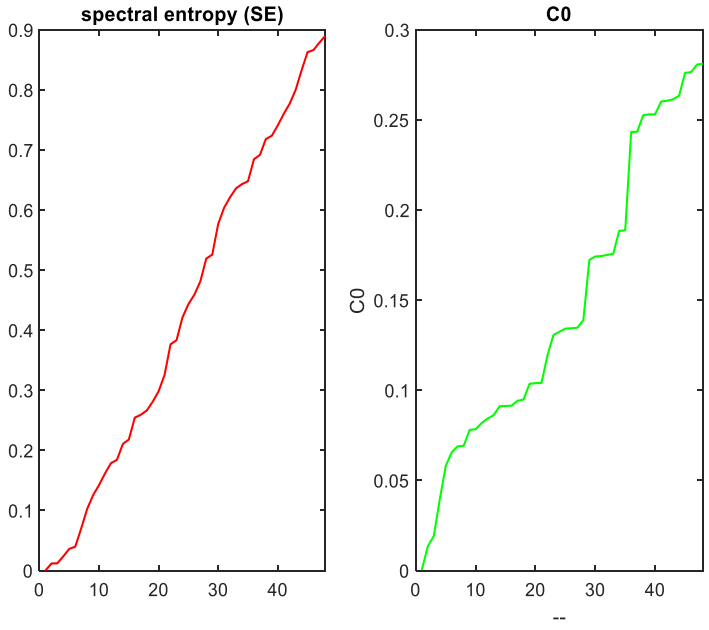

Figure (5) : Complexity of keys.

4.4 Poincaré section

The Poincaré section is thought of as a typical approach for testing chaotic dynamics. When a dynamic system has just one fixed point or a sparse collection of discrete points, its motion is periodic. When a dynamic system has a closed curve, its motion is quasi-periodic. When a dynamic system contains chunks of dense points that are known to have a fractal structure, their motion is chaotic.

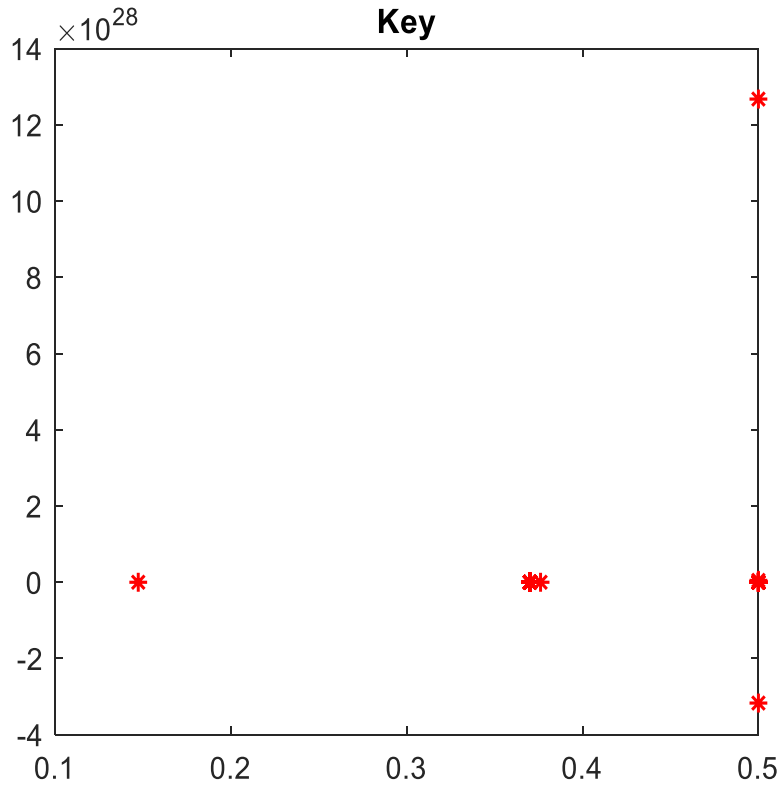

Figure (6) : Poincaré section of key when parameter r=[2,1]

5. Conclusions

The primary issue with the TSFS is that the utilized keys are static, making it possible for the taker to know what they are. The proposed keys in this work are generated dynamically, avoiding the static elements of key generation. The study of the generated keys demonstrates that the suggested approach is effective against the attacks that enable the use of this TSFS in a variety of applications.

References

[1] Yong-Xia, Zhao. "The technology of database encryption." 2010 Second International Conference on Multimedia and Information Technology. Vol. 2. IEEE, 2010.

[2] H. A. Al-Souly, A. S. Al-Sheddi and H. A. Kurdi, "Enhanced TSFS algorithm for secure database encryption," 2013 Science and Information Conference, London, UK, 2013, pp. 328-334.

[3] Manivannan, D., & Sujarani, R. (2010). *Light weight and secure database encryption using TSFS algorithm. 2010 Second International Conference on Computing, Communication and Networking Technologies.*

[4] J. Ayad et al., "Image Encryption using Chaotic Techniques: A Survey Study," 2021 International Conference in Advances in Power, Signal, and Information Technology (APSIT), Bhubaneswar, India, 2021, pp. 1-6.

[5] A. Sahay and C. Pradhan, "Multidimensional comparative analysis of image encryption using gauss iterated and logistic maps," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2017, pp. 1347-1351.

[6] H. G. Bhaskar, S. Rohith and M. R. Mahesh, "Two level image encryption scheme using Arnold map and combined key sequence of logistic map and tent map," 2015 Twelfth International Conference on Wireless and Optical Communications Networks (WOCN), Bangalore, India, 2015, pp. 1-5.

*[7] Wireless survey. sensor Jennifer network Yick, Biswanath Mukherjee, Dipak Ghosal *. Department of Computer Science, Computer Networks 52 (2008) 22922330.*

*[8] Nandkishor B., et al, "*Implementing ETSFS Algorithm for Database Security*",* International Research Journal of Engineering and Technology (IRJET) ,Volume: 04 Issue: 05 | May -2017.

[9] *Shatha Habeeb, Rehab F. Hassan, "Sensors data encryption using TSFS Algorithm", Journal Of Madent Alelem College Vol 10 , No 1, Year 2018.*