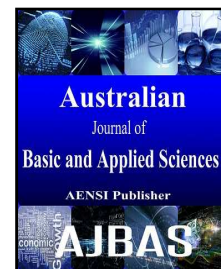




AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES

ISSN:1991-8178 EISSN: 2309-8414
Journal home page: www.ajbasweb.com



UNICODE Text Security Using Dynamic and Key-Dependent 16x16 S-BOX

¹Balajee Maram K. and ²J.M. Gnanasekar

¹Ph.D(CS) Scholar, Research and Development Centre, Bharathiar University, Coimbatore, Sr. Asst. Prof., Dept. of CSE, GMRIT, Rajam, INDIA

²Professor, Department of Computer Science & Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu.

Address For Correspondence:

Balajee Maram K., Ph.D(CS) Scholar, Research and Development Centre, Bharathiar University, Coimbatore, Sr. Asst. Prof., Dept. of CSE, GMRIT, Rajam, INDIA.

E-mail: balajee.journal@outlook.com.

ARTICLE INFO

Article history:

Received 10 November 2015

Accepted 30 December 2015

Available online 18 January 2016

Keywords:

UNICODE, S-box, UTF, ASCII, Cryptography.

ABSTRACT

The tremendous development in digital communication/data transmission over internet and severe threats from eavesdroppers/cyber attackers leads to concentrate on robust cryptography algorithms. In data transmission over internet, many symmetric cryptography algorithms have been introduced using S-box which provides security. S-box is very important component for some cluster of Cryptography algorithms. Some cryptography algorithms depend on static S-box, which yields insecurity to the digital data. The existing S-box generation algorithms are able to handle ASCII text only. This research paper presents the Substitution Box(S-box) that is dynamic and key-dependent. Dynamic and key-dependent S-box wraps the data with high security. The proposed S-box works with UNICODE text which includes UTF-8, UTF-16 and UTF-32 versions. It was tested on UNICODE text using PYTHON language. The results conclude that the proposed S-box is suitable for UNICODE text and shown better performance.

INTRODUCTION

In the public internet, information is being attacked and misused by hackers at different levels in the digital communication (William Stallings, 2004). Such type of attacks can be avoided through Data Encryption (Charles, P., 2004). There are two types of encryption are called as Symmetric-key encryption and Asymmetric-key encryption. Symmetric-encryption algorithms are 1000 times faster than Asymmetric-encryption algorithms. Still, Symmetric-key cryptography algorithms are being used to exchange data over insecure communication channels (Dragos Trinca, 2006). The block ciphers such as DES (Data Encryption Standard), AES (Advanced Encryption Standard) (<http://csrc.nist.gov/publications/fips/fips197/>), and EES (Escrowed Encryption Standard) (<http://csrc.nist.gov/publications/fips/fips1185/>) are being used by many companies in worldwide. The Tiny Encryption Algorithm (TEA) (Hernández, 2002; Hernández, 2003; Hernández,) is one of the fastest and efficient algorithm which uses operations from orthogonal algebraic groups like XOR, ADD and SHIFT. TEA satisfies Shannon's properties confusion and diffusion which are important for a security of block ciphers. But TEA suffers from equivalent keys, because its key size is only 126-bits (Kelsey, 1997). The Literature (Gupta, V. and S. Gupta, 2001; Daswani, N. and D. Boneh, 1999) suggests the traditional cryptography algorithms are not suitable to low processing devices like mobile devices, because low processing devices have slow processor, limited battery and limited memory.

Open Access Journal

Published BY AENSI Publication

© 2016 AENSI Publisher All rights reserved

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

To Cite This Article: Balajee Maram K. and J.M. Gnanasekar., Generation of a Dynamic Random 16x16 S-Box for Unicode Text Using Prime Numbers and Secret-Key. *Aust. J. Basic & Appl. Sci.*, 10(1): 26-36, 2016

Confusion and Diffusion is an important characteristic for Information Security. Confusion is being provided by different forms of existing Substitution Boxes (S-Box) S-Box is a building block of Information Security. S-Box is a desirable and important component in symmetric cryptography. S-Box is a mapping table which translates n-bits to m-bits. The design and analysis of a strong S-Box is a time consuming process, because it supports nonlinearity to the cryptosystems. But the limitations in the S-Box leads to break easily (Adams, C. and S. vTavares, 1990; Hussain, I., 2010). The S-Box design is being suffered from two major challenges. They are S-Box Searching and Verification of S-Box against the desired properties for an S-Box (Ahmed, N.,).

The block ciphers are depends on the Substitution-Permutation (SP) concept. S-Box is a nonlinear component which enables block cipher is resistant to various attacks such as linear and differential cryptanalysis. This is achieved in SP networks when S-Box satisfies the criteria like Avalanche Effect, Strict Avalanche (SAC) and Bit Independence Criteria (BIC), nonlinearity and maximum expected linear probability (MELP) etc. These are the desirable properties for S-Box design (Vergili, I.I., Yücel, M.D. (Eds.), 2000; Vergili, I., M.D. Yücel, 2001; Keliher, L., 1997; Keliher, L., 2005).

Instead static S-Box, random key-dependent S-Boxes are being generated for encryption process and S-Box are generated and checked until a strong S-Box is found (Mroczkowski, P., 2009). Recent technology supports static or dynamic S-box generation that supports ASCII values only. Many S-box generation techniques have been introduced that supports ASCII characters only. The traditional coding Scheme ASCII supports 256 characters and extended ASCII supports 512 characters only. Many countries developing their own character coding techniques which support their national languages. Latest technology introduces a consistent encoding system is called UNICODE (<http://www.unicode.org>). No S-box generation technique is available for UNICODE characters.

Design a robust encryption method for multilingual message is not a simple task. Multilingual languages are generated by UNICODE character set. UNICODE (Maram Balajee, 2011) is a new character encoding scheme. The main objective of UNICODE is to unify all the different encoding schemes and it reduces the confusion between different character encoding schemes. UNICODE has been used in data hiding (Shirali-Shahreza, H., M. Shirali-Shahreza, 2008) proposed by M. H. Shirali-Shahreza, and Mohammad Shirali-Shahreza proposed new method for hiding information in Persian and Arabic Unicode texts. Another method proposed by Lip Yee Por and *et al.* (2012) which worked on UNICODE space characters.

The proposed system is able to handle and solve the limitations in existing light-weight cryptography algorithms. It is based on two large prime numbers for generating Pseudo-Random Numbers. For different seeds (different large primes), it can generate different Pseudo-Random Numbers. The proposed system can use two large primes, bitwise-XOR operation and some mathematical methods. This paper works on the properties like Hamming-Distance, Balanced-Output and Avalanche-Effect. A preliminary result shows that the proposed algorithm supports the good cryptographic algorithm properties, with the added benefit that is resistant to linear and differential cryptanalysis.

This paper is organized as follows to explain the proposed work. Section 2 presents the properties of S-box. Section 3 presents Literature Survey. Section 4 presents the proposed algorithm for S-box construction. Section 5 gives the Results comparison between the existing and proposed algorithms. Section 6 concludes the benefits of the proposed systems and future enhancements.

2. Literature Survey:

In this section, some of the existing algorithms are presented to discuss:

A new algorithm for S-Box generation “DESIGNING THE S BOXES OF BLOWFISH ALGORITHM USING LINEAR CONGRUENTIAL GENERATOR” has been proposed. In this paper, the S-BOX is generated using a pi value which leads to easier cryptanalysis. The pi values are replaced with Linear Congruential Generator. The proposed algorithm yields better results than original blowfish algorithm. But the maximum period of Linear Congruential Generator is $M-1$, which is much too small for 32-bit generators where $M \leq 2^{32} \approx 10^9$, since this can be exhausted in a few minutes on a workstation.

“Efficient Implementation of AES By Modifying S-Box” (Vijay, L.,) has been proposed. The performance of the original AES algorithm using S-BOX which is based on polynomial is very good. In this paper, S-box and Inv S-box have been modified by swapping each word of S-box and Invs-box generated by new polynomial. The result shows the modified AES yields better results.

The Researchers concentrated on Random S-Box generation techniques. Random S-Box generation algorithms have been proposed and compared in “COMPARISON OF RANDOM S-BOX GENERATION METHODS” (Piotr Mroczkowski, 2009). The S-BOX is generated using Legendre’s sequences. The researches of nonlinearity of generated S-boxes show that, giving the maximum nonlinearity, it is possible to generate “good” S-boxes. It gives the possibility of making S-boxes used in block ciphers and makes them replaceable. This treatment secures cryptosystems against many crypto-graphical attacks, especially differential and algebraic cryptanalysis.

Dynamic AES-128 with Key-Dependent S-box (Eman Mohammed Mahmoud, 2013) presents a new algorithm that generates a dynamic AES with key-dependent S-Boxes. Any change of the secret key reflects the structure of the S-Box. Here the s-box is based on permutations only. Here s-box rows and columns are interchangeable is based on S1 vector and S2 vector. The result shows the proposed algorithm improves the AES security.

Cryptographically strong S-Box is proposed in "Construction of Cryptographically Strong 8x8 S-boxes" (Iqtadar Hussain, 2011). The construction of S-box is the action of PGL (2, GF (2⁸)) group on GF(2⁸), The proposed S-box is comparable with AES S-box, Affine Power Affine S-box, Gray S-box. And it is better than Prime S-box.

In this paper, a new algorithm "KEY-DEPENDENT S-BOX IN LIGHTWEIGHT BLOCK CIPHERS" (Sufyan Salim Mahmood Aldabbagh, 2014) to generate S-BOX based on the secret key is proposed. This research showed the intensive analysis for cost and security for 1bit, 2bits, 4bits and more than 4 bits. Depending on the application, the suitable method can be selected.

Key-dependent S-box has been proposed in "Key-Dependent S-Box Generation in AES Block Cipher System" (Kazys Kazlauskas, Jaunius Kazlauskas, 2009). The static s-boxes are vulnerable. So the dynamic s-boxes are resistant to different attacks. The proposed system is used to generate large number of s-boxes by changing the secret-key.

A new strategy has been proposed in "A novel design for the construction of safe S-boxes based on TDERC sequence" (Huss Ain, A., Alkh aldi, 2015) for developing cryptographically strong 8X8 S-boxes. These proposed S-boxes satisfy bijective and nonlinearity properties.

A new S-box was constructed in "Performance Efficiency of Modified AES Algorithm Using Multiple S-Boxes" (Felicisimo, V.) using XOR operation and affine transformation. The speed performance test in the encryption and decryption processes, the AES-2SBox was more efficient by 22.986% and 109.79% respectively than the original AES algorithm. From these results, we observed that the speed performance significantly increased in the modified AES algorithm using multiple S-Boxes, while the security side has slightly weakened.

The extra stage is known as S-Box Rotation has been added in "Enhancing Advanced Encryption Standard S-Box Generation Based on Round Key" (Julia Juremi Ramlan Mahmud Salasiah Sulaiman Jazrin Ramli,) and is introduced at the beginning of the round function. The rotation value is now dependent on the entire round key. This property holds for all possible 256 rotations, and this property can be used to make the S-box key-dependent.

A new way of generating the elements in and P arrays and S box has been proposed in (Jeyamala Chandrasekaran, B., 2011). Experimental results clearly show that the algorithm generates highly non linear S boxes and P arrays while preserving the same level of security as in Blowfish. Also the algorithm offers sufficient resistance towards Brute force attack and statistical crypt analysis of original and encrypted images.

The masked S-box has been proposed in (Lekshmi, R., Sajan Xavier, 2014) and has the ability to defend against DPA and glitch attacks, thereby offering high security level. The masked S-box maps the plaintext and masking values from GF (2⁸) to GF (2⁴) and vice-versa. Thus the implementation of masked s-box increases the system security and hence increases the algorithm's performance.

A New nonlinear transformation for AES S-box has been presented in (Vinoth John Prakash, S., A. Arun, 2013) which enhance the complexity of the S-box structure. The enhanced S-box structure provides a strong and expanded security. The alignment of the biometrics scheme with AES algorithm provides an additional protection in authentication system.

The concept of using key-dependent s-box manipulations to strengthen specific block ciphers against attacks which depend upon knowledge of the s-box contents has been proposed in (Sandy Harris and Carlisle Adams, 1999). Cryptographic strength may be substantially increased with hidden s-box contents.

TAYSEER S. ATIA et.al (2015) presents a novel S-box generation algorithm. It depends on initial value. It analyzes the performance in terms of time with existing algorithms.

Kazys KAZLAUSKAS, Gytis VAICEKAUSKAS, Robertas SMALIUKAS et.al (2015) explains, the key dependent S-box was proposed. This proposed algorithm is resistant to linear and differential cryptanalysis. It can generate huge number of S-boxes by changing the secret-key

TAYSEER S. ATIA et.al (2014) presents in this paper, a novel S-box generation algorithm was proposed. It depends on initial value. It analyzes the performance in terms of time with existing algorithms.

Razi Hosseinkhani, H. Haj Seyyed Javadi et.al (2012) proposes the S-box generation is based on cipher key. This proposed algorithm can generate as many as S-boxes which gives high security to AES algorithm.

C.P.Ronald Reagan, S.Selvi, Dr.S.Prasanna Devi, Dr.V.Natarajan et.al [46] presents the generation of dynamic S-box for AES algorithm is proposed. The time required to generate the dynamic S-box is less than 0.5 ms. It satisfies Avalanche Criteria.

M. Hamdi, R. Rhouma, S. Belghith et.al (2015) proposes a novel algorithm for generating Pseudo Random Number Generator (PRNG) has been proposed. It was found that the proposed algorithm is very fast and secure.

Grasha Jacob, Dr. A. Murugan, Irine Viola et al. presents the generation of S-box is based on dynamic key. It satisfies various S-box properties like bijection, non-linearity, Strict Avalanche Criterion, Balance etc.

Hristina Mihajloska, Danilo Gligoroski et al. (2011) explains the Quasi-groups based S-boxes has been proposed. This S-box satisfies the first characteristic of S-box is Linearity.

A. Hussain Alkhalidi et al. (2015) presents a novel method for generating cryptographically 8X8 S-boxes is proposed. The proposed S-box is bijective.

Dragan Lambić and Miodrag Živković (2013) presents the advantage is a possibility to achieve a large key space.

3. Proposed System:

Here Pseudo-Random Numbers are generated that are needed for Encryption/Decryption process. The generation process of Pseudo-Random Numbers is as follows:

3.1 Pseudo-code for Generation of Pseudo-Random Numbers:

- 1: Take two large prime number p and q.
- 2: for all $i=0, 1, \dots, 255$ do
- 3: $p=(p*q+1) \bmod 256$
- 4: $a(i) \leftarrow p$
- 5: end for

3.2 Pseudo-code for Generation of Random Numbers for positions in S-Box:

Here Pseudo-Random Numbers are generated that are needed for positions to keep the Pseudo-Random Numbers in S-box for Encryption/Decryption process. The generation process of Pseudo-Random Numbers for position is as follows:

- 1: Calculate next primes of p and q are p1 and q1
- 2: for all $i=0, 1, \dots, 255$ do
- 3: $p1=(p1*q1+1) \bmod 256$
- 4: $pos(i) \leftarrow p1$
- 5: end for

3.3 Algorithm for Key-Dependent Random S-Box1 Generation:

Input:

- a) The secret key $key[i], i=1, 2, \dots, n$ is the vector of n integer numbers from the interval [0..255].
- b) Array of Random Numbers i.e. a() for S-Box values
- c) Number of rounds 'rounds'

Output:

- a) The key-dependent substitution box $SBox(i)(j), i=0,1,\dots,15$ and $j=0,1,\dots,15$ is the 2-Dimensional vector of the different integer numbers(bytes) from the range [0,255].
- b) The key-dependent inverse substitution box $invS-Box(i)(j), i=0,1,\dots,15$ and $j=0,1,\dots,15$ is the 2-Dimensional vector of the different integer numbers(bytes) from the range [0,255].

Algorithm:

- Step 1: Make $S-Box(i)(j)$ from the array a(), $i=0,1,\dots,15$ and $j=0,1,\dots,15$ is the 2-Dimensional vector of the different integer numbers(bytes) from the range [0,255].
- Step 2: Each row in $S-Box()$ is circularly shifted to Left/Right according to the values of Key()
- Step 3: for round=1 .. rounds do
- Step 4: for all row=0, 1, ..., 16 do
- Step 5: if key(index) is even then row in $S-Box()$ is circularly shifted to Left of key(index) mod 16 positions
- Step 6: if key(index) is odd then row in $S-Box()$ is circularly shifted to Right. Of key(index) mod 16 positions
- Step 7: end of Step 4 for loop
- Step 8: end of Step 3 for loop
- Step 9: Now $S-Box()$ is ready.

3.4 Algorithm for Generation of Key-Dependent Random S-Box2:

Input:

- a) The secret key $key[i], i=1, 2, \dots, n$ is the vector of n integer numbers from the interval [0..255].
- b) Array of Random Numbers i.e. a() for S-Box values
- c) Array of Random Numbers i.e pos() for positions
- d) Number of rounds 'rounds'

Output:

a) The key-dependent substitution box $S\text{-Box}(i)(j)$, $i=0,1,\dots,15$ and $j=0,1,\dots,15$ is the 2-Dimensional vector of the different integer numbers (bytes) from the range $[0,255]$.

b) The key-dependent inverse substitution box $\text{invS-Box}(i)(j)$, $i=0,1,\dots,15$ and $j=0,1,\dots,15$ is the 2-Dimensional vector of the different integer numbers (bytes) from the range $[0,255]$.

Algorithm:

Step 1: Make $S\text{-Box}(i)(j)$ from the array $a()$, $i=0,1,\dots,15$ and $j=0,1,\dots,15$ is the 2-Dimensional vector of the different integer numbers (bytes) from the range $[0,255]$.

Step 2: Each row in $S\text{-Box}()$ is circularly shifted to Left/Right according to the values of $\text{Key}()$

Step 3: for $\text{round}=1 \dots \text{rounds}$ do

Step 4: for all $\text{row}=0, 1, \dots, 16$ do

Step 5: if $\text{key}(\text{index})$ is even then row in $S\text{-Box}()$ is circularly shifted to Left of $\text{key}(\text{index}) \bmod 16$ positions

Step 6: if $\text{key}(\text{index})$ is odd then row in $S\text{-Box}()$ is circularly shifted to Right. Of $\text{key}(\text{index}) \bmod 16$ positions

Step 7: end of Step 4 for loop

Step 8: end of Step 3 for loop

Step 9: All the elements in $S\text{-Box}()$ are permuted according to values in $\text{pos}()$ array.

Step 10: Now $S\text{-Box}()$ is ready.

3.5 Algorithm for Inverse S-Box

Step 1: Arrange all the elements from $S\text{-Box}()$ to $a()$

Step 2: for all $i=0,1,\dots,255$ do

Step 3: Calculate $\text{inva}(a(i)) \leftarrow i$

Step 4: end for

Step 5: Arrange all the elements from $\text{inva}()$ to $\text{invS-Box}()$

Step 6: Inverse of $S\text{-Box}()$ is ready.

All the elements in $S\text{-box}$ are rearranged from 2-dimensional array to 1-dimensional array 'a'. The inverse of 'a' is calculated using the following formula: $\text{Inva}(a[\text{index}]) = \text{index}$. Then all 256 elements are rearranged from 1-dimensional to 2-dimensional array is called invS-Box .

3.6 Encryption and Decryption technique for UNICODE using S-box, 3.3 & 3.4 Algorithms:**3.6.1 Encryption:**

Step 1: Take Input "Plain-text", which is UNICODE text

Step 2: Convert each UNICODE character into decimal form

Step 3: $\text{quotient} = (\text{Decimal value of UNICODE character}) / 256$

Step 4: $\text{remainder} = (\text{Decimal value of UNICODE character}) \% 256$

Step 5: $\text{Intermediate_value} = \text{pass remainder through S-box1 in 3.3}$

Step 6: $\text{Cipher-decimal} = (\text{quotient} * 256) + \text{Intermediate_value}$

Step 7: $\text{Cipher-character} = \text{UNICODE form of Cipher-decimal in Step 6}$

Step 8: if (characters in plain-text) then go to Step 3

Step 9: stop

3.6.2 Decryption:

Step 1: Take Input "Cipher-text", which is UNICODE text

Step 2: Convert each UNICODE character into decimal form

Step 3: $\text{quotient} = (\text{Decimal value of UNICODE character}) / 256$

Step 4: $\text{remainder} = (\text{Decimal value of UNICODE character}) \% 256$

Step 5: $\text{Intermediate_value} = \text{pass remainder through Inverse-S-box in 3.5}$

Step 6: $\text{Plain-decimal} = (\text{quotient} * 256) + \text{Intermediate_value}$

Step 7: $\text{Plain-character} = \text{UNICODE form of Plain-decimal in Step 6}$

Step 8: if (characters in Cipher-text) then go to Step 3, else "stop"

3.7 Experimental Results using S-box1:

In this section the results of analysis are given. The analysis includes the comparison of the properties of $S\text{-box}$ like Hamming-distance, Balanced-output and Avalanche Effects of proposed and existing algorithms. The proposed algorithm has been checked through PYTHON code.

Ahmed, N., Testing an S-Box for Cryptographic Use, *International Journal of Computer and Electrical Engineering*, 1-5.

Balajee Maram K, J M Gnanasekar, Construction Of A Dynamic Random S-Box For Data Security Using Large Prime Numbers And Secret-Key, *TEM Journal*, ISSN: 2217-8309,(Print). eISSN: 2217-8333 (Online), Vol.5, No.1.

Balajee Maram K, J M Gnanasekar, Light Weight Cryptographic Algorithm To Improve Avalanche Effect For Data Security Using Prime Numbers And Bit Level Operations, *International Journal of Applied Engineering Research*, ISSN 0973-4562 Volume 10, Number 21 (2015) pp:41977-41983.

Basavaraj, P., Halagali, Vijay, L. Hallappanavar, Veena V. Desai, Designing The S Boxes Of Blowfish Algorithm Using Linear Congruential Generator, *ASM's International e-journal of Ongoing Research in Management and IT*.

Charles, P., Pfleeger, Shari Lawrence Pfleeger, 2004. "Security in Computing", Pearson Education, 642-666.

Daswani, N. and D. Boneh, 1999. "Experimenting with Electronic Commerce on the PalmPilot", *Proc. Eurocrypt'99*, LNCS 1648, Springer Verlag, 1-16.

Data Encryption Standard: <http://csrc.nist.gov/publications/fips/fips46-3/fips-46-3.pdf>

Dragan Lambić and Miodrag Živković, 2013. comparison of random s-box generation methods in publications de l'institut mathématique, 109–115.

Dragos Trinca, 2006. "Sequential and Parallel Cascaded Convolution Encryption with Local Propagation: Toward future Directions in Cryptography," *Proceedings of The third International Conference on Information Technology-New Generations. (ITNG'06)*, 0-7695-2497-4 /, IEEE Computer Society.

Eman Mohammed Mahmoud, Ahmed Abd El Hafez, Talaat A.Elgarf, AbdelhalimZekry, 2013. dynamic AES-128 with Key-Dependent S-box, *International Journal of Engineering Research and Applications (IJERA)* ISSN: 2248-9622, 3(1): 1662-1670.

Escrowed Encryption Standard: <http://csrc.nist.gov/publications/fips/fips185/fips-185.txt>

Felicisimo, V., Wenceslao, Jr., Performance Efficiency of Modified AES Algorithm Using Multiple S-Boxes, *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 5(1): 1-9.

Grasha Jacob, Dr. A. Murugan, Irine Viola," Towards the Generation of a Dynamic Key- Dependent S-Box to Enhance Security", *Cryptology ePrint Archive*

Gupta, V. and S. Gupta, 2001. "Securing the Wireless Internet", *IEEE Communications*, 39(12): 68-74.

Hamdi, M., R. Rhouma, S. Belghith, 2015. "A Very Efficient Pseudo-Random Number Generator Based On Chaotic Maps and S-Box Tables", *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9-2.

Hernández, Julio César, Isasi, Pedro, Ribagorda, Arturo, 2002. "An application of genetic algorithms to the cryptanalysis of one round TEA". *Proceedings of the Symposium on Artificial Intelligence and its Application*.

Hernández, Julio César, Sierra, José María, Isasi, Pedro, Ribargorda. Arturo, 2003. "Finding efficient distinguishers for cryptographic mappings, with an application to the block cipher TEA". *Proceedings of the Congress on Evolutionary Computation*.

Hernández, Julio César, Sierra, José María, Ribagorda, Arturo, Ramos, Benjamín, J.C. Mex-Perera, " Distinguishing TEA from a Random Permutation: Reduced Round Versions of TEA Do Not Have the SAC or Do Not Generate Random Numbers", *Cryptography and Coding Volume 2260 of the series Lecture Notes in Computer Science*, 374-377.

Hristina Mihajloska, Danilo Gligoroski, 2011. A New Approach Into Constructing S-Boxes For Lightweight Block Ciphers, *8th Conference on Informatics and Information Technology with International Participation (CIIT 2011)*.

Huss Ain, A., Alkhaldi, 2015. A novel design for the construction of safe S-boxes based on TDERC sequence, *Alexandria Eng. J.*

Hussain Alkhaldi, A., 2015. A novel design for the construction of safe S-boxes based on TDERC sequence, *Alexandria Eng. J.*, <http://dx.doi.org/10.1016/j.aej.2015.01.003>

Hussain, I., T. Shah, H. Mahmood and M. Afzal, 2010. Comparative analysis of S-boxes based on graphical SAC, *International Journal of Computer Applications*, 2(5): 1-7.

Iqtadar Hussain, Tariq Shah, Muhammad Asif Gondal and Waqar Ahmad Khan, 2011. Construction of Cryptographically Strong 8x8 S-boxes, *World Applied Sciences Journal* 13 (11): 2389-2395, ISSN 1818-4952

Jeyamala Chandrasekaran, B., Subramanyan and G.S. Raman, 2011. Ensemble Of Blowfish With Chaos Based S-Box Design For Text And Image Encryption, *International Journal of Network Security & Its Applications (IJNSA)*, 3-4.

Julia Juremi Ramlan Mahmud Salasiah Sulaiman Jazrin Ramli, Enhancing Advanced Encryption Standard S-Box Generation Based on Round Key, *International Journal of Cyber-Security and Digital Forensics (IJCSDF)* 1(3): 183-188.

- Kazlauskas, K., 2009. Key-dependent S-box generation in AES block cipher system, *Informatica*, 20: 23–34
- Kazys Kazlauskas, Gytis Vaicekauskas, Robertas Smaliukas, 2015.” An Algorithm for Key-Dependent S-Box Generation in Block Cipher System”, *INFORMATICA*, 26(1): 51–65. Vilnius University, DOI: <http://dx.doi.org/10.15388/Informatica.2015.38>
- Kazys Kazlauskas, Jaunius Kazlauskas, 2009. Key-Dependent S-Box Generation in AES Block Cipher System, *INFORMATICA*, 20(1): 23–34.
- Keliher, L., 2005. Refined analysis of bounds related to linear and differential and linear cryptanalysis for the AES, (In: H. Dobbertin *et al.*, eds. *Advanced Encryption Standard AES S04*, Bonn, 2004, *Lect. Notes Comput. Sci.*, 42–57.
- Keliher, L., H. Meijer, S. Tavares, 1997. A new substitution-permutation network cryptosystem using key-dependent s-boxes, In: *Proc. SAC'97, Canada*, 13–26
- Kelsey, John, Schneier, Bruce, Wagner, David, 1997. "Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X NewDES, RC2, and TEA". *Lecture Notes in Computer Science*, 1334: 233–246.
- Lecture Notes on “Computer and Network Security” by AviKak.Pdf [http:// junicholl.org/Crypt analysis /Data / EnglishData.php](http://junicholl.org/Crypt%20analysis/Data/EnglishData.php)
- Lekshmi, R., Sajan Xavier, 2014. FPGA Based Design of AES with Masked S-Box for Enhanced Security, *International Journal of Engineering Science Invention* ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726, 3(5): 01-07
- Mar, P.P. and K.M. Latt, 2008. New analysis methods on strict avalanche criterion of S-boxes, *World Academy of Science, Engineering and Technology*, 48: 150-154.
- Maram Balajee, 2011. “UNICODE and Colors Integration tool for Encryption and Decryption “published in *International Journal on Computer Science and Engineering (IJCSSE)*. ISSN: 0975-3397, 3-3.
- Mroczkowski, P., 2009. Generating Pseudorandom S-Boxes – a Method of Improving the Security of Cryptosystems Based on Block Ciphers, *J. Telecommun. Inform. Technol.*, 74–79.
- Piotr Mroczkowski, 2009. Generating Pseudorandom S-Boxes – a Method of Improving the Security of Cryptosystems Based on Block Ciphers, *Journal of Telecommunications and Information Technology*.
- Por, L., K. Wong, K. Chee, 2012. UniSpaCh: A text-based data hiding method using Unicode space characters, *The Journal of Systems and Software*, 85(5), 1075– 1082.
- Razi Hosseinkhani, H., Haj Seyyed Javadi, 2012. “Using Cipher Key to Generate Dynamic S-Box in AES Cipher System”, *International Journal of Computer Science and Security (IJCSS)*, 6(1).
- Ronald Reagan, C.P., S. Selvi, Dr. S. Prasanna Devi, Dr.V.Natarajan, 2014.” Enhancing DES Using Local Languages”, *International Journal of Engineering Science and Innovative Technology (IJESIT)*, 3-1.
- Sandy Harris and Carlisle Adams, 1999. Key-Dependent S-Box Manipulations, *SAC'98, LNCS 1556*, 15–26, c Springer-Verlag Berlin Heidelberg.
- Schneier, B., 1994. Description of a new variable-length, 64-bit block cipher (Blowfish), (In: *Proc. Fast Software Encryption*, Springer, 191–204.
- Shirali-Shahreza, H., M. Shirali-Shahreza, 2008. Steganography in Persian and Arabic unicode texts using pseudo-space and pseudo-connection characters, *Journal of Theoretical and Applied Information Technology*, 4(8): 682-687.
- Sufyan Salim Mahmood Aldabbagh, Imad Fakhri Taha Al Shaikhli, Muhammad Reza Zaba, 2014. Key-Dependent S-Box In Lightweight Block Ciphers, *Journal of Theoretical and Applied Information Technology* 20th April, 62-2.
- Tayseer S., Atia, 2014.” Development Of A New Algorithm For Key And S-Box Generation In Blowfish Algorithm”, *Journal of Engineering Science and Technology*, 9(4): 432-442.
- The Unicode Consortium, The Unicode Standard, <http://www.unicode.org>.
- Vergili, I., M.D. Yücel, 2001. Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen $n \times n$ S-Boxes, *Turk J Elec Engin*, 9(2): 137-145.
- Vergili, I.I., Yücel, M.D. (Eds.), 2000. On Satisfaction of Some Security Criteria for Randomly Chosen S-Boxes, in *Proc. 20th Biennial Symp. on Communications*, Kingston, 64-68.
- Vijay, L., Hallappanavar, Basavaraj P. Halagali, Veena V. Desai, Efficient Implementation of Aes By Modifying S-Box, *IOSR Journal of Computer Science (IOSR-JCE)*, 35-39.
- Vinoth John Prakash, S., A. Arun, 2013. A Secure Software Implementation of Nonlinear advanced Encryption Standard, *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)* ISSN: 2319 – 4200, ISBN: 2319-4197, 1- 5: 44-48
- William Stallings, 2004. “Network Security Essentials (Applications and Standards)” Pearson Education, 2-80.