IEREK PRESS
Academic Research Community

http://www.press.ierek.com

ARChive

CrossMark

ISSN (Print: 2537-0154, online: 2537-0162)

International Journal on:

The Academic Research Community Publication

DOI: 10.21625/archive.v3i3.520

# Energy Efficient Encryption Algorithm for Low Resources Devices

**Bassam. W. Aboshosha[1], Mohamed. M. Dessouky[2], Ayman Elsayed[2]**

[1] *Computer Department, Higher Institute of Engineering, Elshorouk Academy, Cairo, Egypt*

[2] *Faculty of Electronic Engineering, Menoufia University, Menofia, Egypt*

## Abstract

Saving energy is one of the most challenging aspects in the wireless network devices. Such devices are connected together to perform a certain task. A well-known example of these structures is the Wireless Sensor Network (WSN). Distributed WSN consists of several spread nodes in a harsh area. Therefore, once network has been established sensors replacement is not a possible option before at least five years which called network lifetime. So, it is a necessity to develop specific energy aware algorithms that could save battery lifetime as much as possible. Security and Privacy are the vital elements which need to be addressed to hold up to the trust of users in WSN environment. Because the majority of modern cryptographic algorithms were designed for desktop/server environments, many of these algorithms cannot be implemented in the constrained devices used by these networks. Symmetric key algorithms are a typically efficient and fast cryptosystem, so it has significant applications in many realms. For a WSN with constraint computational resources, the cryptosystem based on symmetric key algorithms is extremely suitable for such an agile and dynamic environment. Therefore, a Simple Lightweight Encryption Algorithm (SLEA) based on addition and subtraction operations and compact Substitution-boxes (S-boxes) is proposed for wireless networks due to its low energy consumption, simple hardware requirements and suitable level of security. In addition, the algorithm tries to overcome the limitations of both public- and symmetric-key protocols. It relies on a smart version of Feistel structure.



***Keywords***

Wireless networks; security; privacy; Wireless Sensor Networks(WSNs); cryptosystem; SLEA; Feistel structure

## 1. Introduction

Recently, Wireless network applications are widely spread in different fields due to the massive increase in the number of connected devices. Near to ten billion network devices are communicated together to accomplish different tasks. These devices serve about billions of users. Services offered can be used for different civil applications such as smart houses/buildings/cities, communications, entertainment, traffic monitoring, healthcare, and even in human bodies for patient monitoring, sharing knowledge, financial transactions, and many other purposes. Besides these applications, it also used for military purposes such as boundary observation, vechile movements monitoring, environment monitoring, torrerists following and arrested and so on. In addition to a more powerful node called aggregation or base station node can communicate with conventional devices (laptops, smart phones, etc.) locate nearly around (Atzori, Iera & Morabito, 2010).

Wireless network environment is extremely open to attacks, for the reasons that wireless communications, the

risk of interception is greater than with wired networks. If the message is not encrypted, or encrypted with a weak algorithm, the attacker can read it; there by it compromises confidentiality. In addition, due to the wireless communication medium, there is a physical attack threats on its devices as they often deployed unsupervised for long time (Gubbi, Buyya, Marusic & Palaniswami, 2013; Want & Dustdar, 2015). Selection of a suitable security scheme is critical in wireless networks because of the open media broadcasts messages with limited energy (Mariona, Hallman, Kline, Miguel, Major & Kerr, 2016). To achieve the security requirements, several cryptographic algorithms have been proposed for server/desktop devices which are not compatable with wireless networks due to its constrained devices (Suo, Wan, Zou & Liu, 2012; Ho, Leung, Mishra, Hosseini, Song & Wagner, 2016). The implementation of such conventional encryption algorithms is not effective as they are decreasing the performance of the energy constrained devices and network lifetime. Therefore, the lightweight cryptographic algorithm is one of the most suitable solutions to safeguard against such kinds of attacks by ciphering the packets transmitted by wireless nodes Ho, Leung, Mishra, Hosseini, Song & Wagner, 2016). Beside that, it is an energy efficient ciphers that minimize the energy consumption in both the encryption and decryption processes and achieves a suitable level of security Ho, Leung, Mishra, Hosseini, Song & Wagner, 2016). This feature is the reason of saving the nodes power and increasing the overall network lifetime. Since the interest of this paper is to design an effective encryption algorithm suitable wired as well as wireless networks. It has been designed to meet the limited resources criteria including encryption and decryption consumed energy, program memory, temporary memory for runtime registers, execution time, encryption/ decryption key memory (Mariona, Hallman, Kline, Miguel, Major & Kerr, 2016; Suo, Wan, Zou & Liu, 2012).

A SLEA algorithm is proposed in this paper due to its ease implementation in both software and hardware. The SLEA algorithm has been designed to be a general purpose algorithm and to handle most of the limitations and constraints associated with other algorithms such as key size, speed, complexity, software and hardware implementations while focusing the attention on speed and complexity, power consumption has long been ignored. Dealing with power is, therefore, rapidly becoming one of the most important issues in encryption algorithms system design. This situation is aggravated by the selection of a suitable function with simple implementation, which achieves confusion and diffusion. SLEA is designed to be secure energy-efficient algorithm in terms of its selected functions. This algorithm is mainly concerned with energy-efficient, high-throughput and computationally intensive technique with a suitable level of security. Although energy efficient and security in wireless networks have been studied in the literature, to the best of our knowledge this paper is the first study to introduce a new design encryption technique specified in wireless network without compromising network lifetime and wasting network resources.

The rest of the paper is organized as follows. Section 2 presents the related work, Section 3 introduces overview of SLEA, Section 3 describes the SLEA encryption, Section 4 presents the security consideration of SLEA, Section 5 describes SLEA implementation consideration, Section 6 presents SLEA analysis, Section 7 presents recommendations, and concluding remarks are made in Section 8.

## 2. Related work

In general terms, there are two types of key-based cryptogiphic algorithms: Symmetric and Asymmetric key cryptogiphic algorithms. Symmetric algorithms, which are the ones of interest in this paper, are these algorithms in which both encryption and decryption keys are identical. Symmetric algorithms can be further divided in two other categories which are stream and block ciphers. The first category includes the ones that operate on the plaintext a single bit at a time. The second category includes those algorithms which operate on the plaintext in groups of bits (usually 64, since this length is considered long enough to preclude analysis and small length to be workable). The latter category is the one of interest in this research.

Many encryption systems were designed in the last two decades. These contributions can be classified into three groups:

– ***Classical Class:*** Conventional algorithms have been compared against each others and ranked according to its ability to be applicable in low resources devices applications, including Data Encryption Standard (DES) (National Bureau of Standards, U.S. Department of Commerce, 1977), Triple DES (TDES) (Stallings, 2009), International Data Encryption Algorithm (IDEA) (Lai & Massey, 1990), Blowfish (Schneier, 1994), and AES (Rijndael, Daemen & Rijmen, 1999; Nechvatal, Barker, Bassham, Burr, Dworkin, Foti & Roback, 2001). The most widely used encryption scheme is DES in which it was adopted in 1977 by the National Bureau of Standards (National Bureau of Standards, U.S. Department of Commerce, 1977). The original DES was designed for hardware implementation and does not produce efficient software code. DES has two major weaknesses which are the relatively short key length of only 56 bits and it may not provide adequate security (Stallings, 2009). The second weakness is that the S-boxes that may have hidden trapdoors that causes highly successful cryptanalysis attack (Coppersmith, 1994; Biham & Shamir, 1991). TDES is a trick to reuse DES implementations by cascading three instances of DES (with distinct keys). TDES is believed to be secure up to at least three times as DES (Stallings, 2009). However, it is slow, especially if it is implemented as a software to work along with computer applications. International Data Encryption Algorithm (IDEA) was created in its first form by Xuejia Lai and James Massey in 1990 (Lai & Massey, 1990). In 1991, Lai and Massey strengthened the algorithm against differential cryptanalysis and called the result Improved Proposed Encryption Standard (IPES). The name of IPES was changed to IDEA in 1992. IDEA is designed to facilitate both software and hardware implementation. IDEA approach suffers from low speed and throughputs caused due to the modulo multiplication operations. The multiplication module is the most computational intensive module and it needs too much effort to design it efficiently. In each round of IDEA, four such modulo multipliers are needed. Therefore, the performance of IDEA in hardware i.e. the throughput rate and the area and cost efficiency depends on efficient design of the multiplier. Blowfish was designed by Bruce Schneier in 1993 (Schneier, 1994). It uses a variable key length and valid keys have between 32- and 448-bits. It was marketed as a replacement for DES and IDEA that could be immediately dropped-in. Blowfish can use huge keys and is believed secure, except with regards to its block size, which is 64 bits, just like DES and TDES. Blowfish is one of the fastest block ciphers in widespread use, except when changing keys. Each new key requires pre-processing equivalent to encrypting about 4 kilobytes of text, which is very slow compared to other block ciphers. This prevents its use in certain applications. Blowfish is efficient in software, at least on some software platforms (it uses key-dependent lookup tables; hence performance depends on how the platform handles memory and caches). Blowfish suffers from weak keys problem also. Finally, the Rijndael (Rijndael, Daemen & Rijmen, 1999) proposal for Advanced Encryption Standard (AES) defined a cipher in which the block length and the key length can be independently specified to be 128,192, or 256 bits. The AES specification uses the same three key size alternatives but limits the block length to 128 bits. A number of AES parameters depend on the key length.

– ***Adaptation Class:*** Conventional algorithms have been modified to be suitable to wireless networks requirements. One example of these is Lightweight Data Encyption Standard (DESL). It is a smart version that mainly based on the classical DES algorithm (Leander, Paar, Poschmann & Schramm, 2007). Unlike DES, DESL uses a single S-box instead of 8 S-boxes of DES (Bogdanov, Knudsen, Leander, Paar, Poschmann, Robshaw, Seurin & Vikkelsoe, 2007). The design criteria of the single DESL S-box make DESL resistant to most common cryptanalytic attacks specially linear and differential attacks (Leander, Paar, Poschmann & Schramm, 2007). This allows saving a part of ROM for tables storage, more than 85% from memory space occupied by traditional DES has been saved. Another research team suggest to replace each S-boxes by in traditional algorithms that use such large boxes to save the memory space as well. The most famous algorithm that use S-boxes like these is AES. Finally, other researchers suggest to deal work with a reduction version of classsical algorithms by reducing number of round.

– ***Progressive Class:*** Suggest new techniques suitable to limited resource devices. PRESENT (Silverlight, 2009) is one of the first lightweight block cipher designs that was proposed for low resourses devices. SIMON and SPECK (Beaulieu, Shors, Smith, Clark, Weeks, & Wingers, 2013) are families of lightweight

block ciphers that were designed to be simple, flexible, and perform well in hardware and software. There are also algorithms from the 1990s such as RC5 (Rivest, 1994), TEA (Wheeler & Needham, 1994) and XTEA ((Wheeler & Needham, 1994), which consist of simple round structures that make them suitable for constrained software environments.

In this paper the proposed SLEA has been designed to mitigate some of the previous work limitations and constraints such as key size, block size, speed, lookup tables, S-boxes, P-boxes, complexity, software and hardware implementations. The main purpose of designing a new encryption algorithm is to be efficiently utilized in wireless networks.

## 3. Simple Encryption Algorithm (SLEA)

This section introduces the concept behind SLEA. In general terms, symmetric algorithms, which are the ones of interest here, are those algorithms in which the encryption key can be calculated from the decryption key and vice versa while in most of them both (encryption and decryption) keys are identical.

In fact, SLEA takes into consideration two important factors which are security and simplicity. SLEA accomplishes a suitable level of security by using two keys (inner and outer keys) beside achieving both confusion and diffusion concepts. Rigid S-boxes with high nonlinearity properties are used to fulfil confusion. A combination of operations beside the nature of the Feistel structure are used to diffuse the data. On the other hand, simplicity achieved in terms of the compact size of S-boxes as well as the used internal simple operations.

SLEA are based on a new approach of Feistel structure. SLEA block cipher operates with n-bit plaintext and ciphertext blocks and is controlled by an 2n-bit key. The fundamental feature in the design of this algorithm is the use of operations from different algebraic groups over the set of all (n/2)-bit blocks which are:

- XOR, denoted as $\oplus$.
- Addition Modulo $2^{n/2}$, denoted by $\boxplus$.
- Subtraction Modulo $2^{n/2}$, denoted by $\boxminus$.
- One's complement, denoted by $\sim$ .

These operations are applied on a (n/2)-bit sub-blocks. The cipher structure was designed to be easily implemented in both software and hardware. In addition, SLEA consists of m rounds using 4m (n/2)-bit outer sub-keys that are generated from the 2n-bit key and m (n/2)-bit inner sub-keys that are generated also from the 2n-bit key. The Basic architecture of SLEA encryption algorithm is depicted in Fig.1. As can be seen in this figure, SLEA architecture is based on the new approach of Feistel structure. The input is divided into right and left elements that go through number of rounds along with the generated sub-keys. The input could be n-bits as well as the key size is 2n-bits. The decryption structure is the same as the encryption with the reverse of the used keys.

### 3.1. Single Round Processing

More detailed architecture of SLEA can be obtained by investigating the internal structure of a single round. First, the n-bit input is divided into two equal halves (n/2) denoted as Li and Ri, respectively. The overall processing at each round can be summarized by equations (1) and (2) where the left half of the input $L_{i-1}$ and the outer sub-key $K_{i+2}$ are manipulated using subtraction binary operation. The output of the subtraction operation is complemented then Xored with function of the right half and both the inner and outer sub-keys S($R_{i-1}$, $K_i$, $Key_i$). Finally, the overall output is added to the outer sub-key $K_{i+3}$ became the right half input of the next round.The right half of the input $R_{i-1}$ and the sub-key $K_i$ are manipulated using addition binary operation. The output of the addition

operation and the sub-key $K_{i+1}$ are manipulated using subtraction binary operation became the left half input of the next round.

$$L_i = (R_{i-1} \boxplus K_i) \boxminus K_{i+1} \tag{1}$$

$$R_i = [(L_{i-1} \boxminus K_{i+2}) \oplus S(Key_i \oplus R_{i-1} \boxplus K_i)] \boxplus K_{i+3} \tag{2}$$

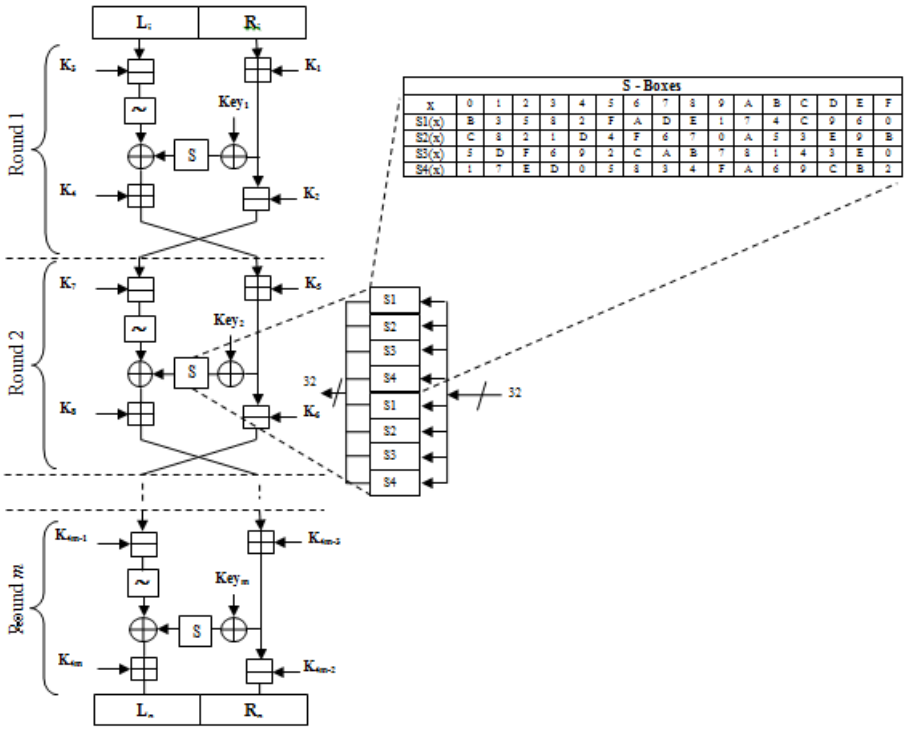


Figure 1. Feistel structure of SLEA

### 3.2. Key Generation

As shown in Fig. 2, for 16 rounds and block of 64 bits, it is required 64 (32-bits) outer sub-keys which are generated from the 128-bits encryption key. The scheme for generation is as follows. The first four sub-keys, labeled K1,K2,. . . .,K4 are taken directly from the key, with K1 being equal to the first (most significant) 32-bits, K2 corresponding to the next 32-bits, and therefore on. Then, a circular left shift of 19-bits positions is applied to the key, and the next four sub-keys are extracted. This procedure is repeated until all 64 outer sub-keys are generated. In addition, a 16 (32-bits) inner sub-keys are generated from another the 128-bits encryption key by the same way the only difference that the circular left shift of 13-bits positions is applied to the key. Then the total key length is 256-bits. Selection of shift value has been chosen to avoid repetition of sub-keys. To ensure that each sub-key is unique the following condition has to be satisfied, greatest common divisor between the shifted value and the key length equal one, GCD (Shift Value, Key Length) = 1.

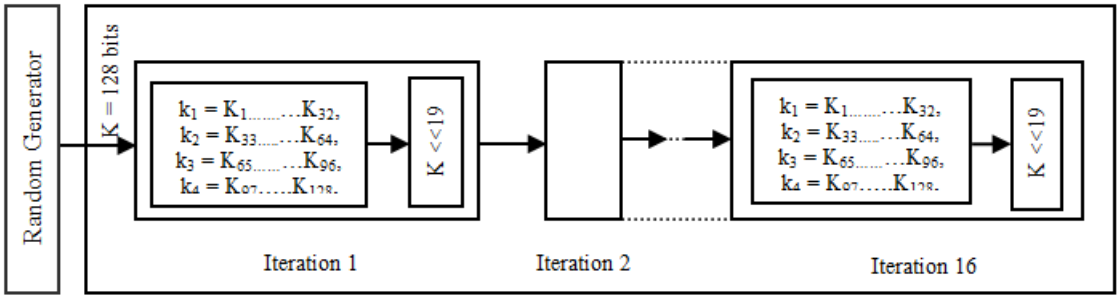


Figure 2. Key Generation Algorithm Diagram

### 3.3. Detailed SLEA Decryption Structure

Again, the beauty of the decryption process is to be the same as the encryption process. SLEA decryption is achieved by using the ciphertext as input to the same overall SLEA structure, but with a different selection of keys, the decryption sub-keys are derived from the encryption sub-keys where they are applied in reverse order. The reversing process for both encryption and decryption for any round i, can be seen in Fig. 3. As any symmetric encryption algorithm, the structure of the decryption round is similar as in encryption process. The left-hand side of this figure shows the encryption process using sub-keys (Ki, Ki+1, Ki+2, Ki+3) and keyi are produced by the left circular shift process. The output of the round consists of n-bits, which represent the coded message. On the other hand, the right-hand side of the Fig.3 shows the decryption process using sub-keys (Ki, Ki+1, Ki+2, Ki+3) and keyi are applied in reverse order. The output of the round consists of n-bits, which represent the original message. Similarly, it is simple to prove that the overall reverse processing at each round can be summarized by equations (3) and (4)

$$R_{i-1} = (L_i \boxplus K_{i+1}) \boxminus K_i \tag{3}$$

$$L_{i-1} = [(\overline{R_i \boxminus K_{i+3}}) \oplus S(Key_i \oplus (L_i \boxplus K_{i+1})] \boxplus K_{i+2} \tag{4}$$

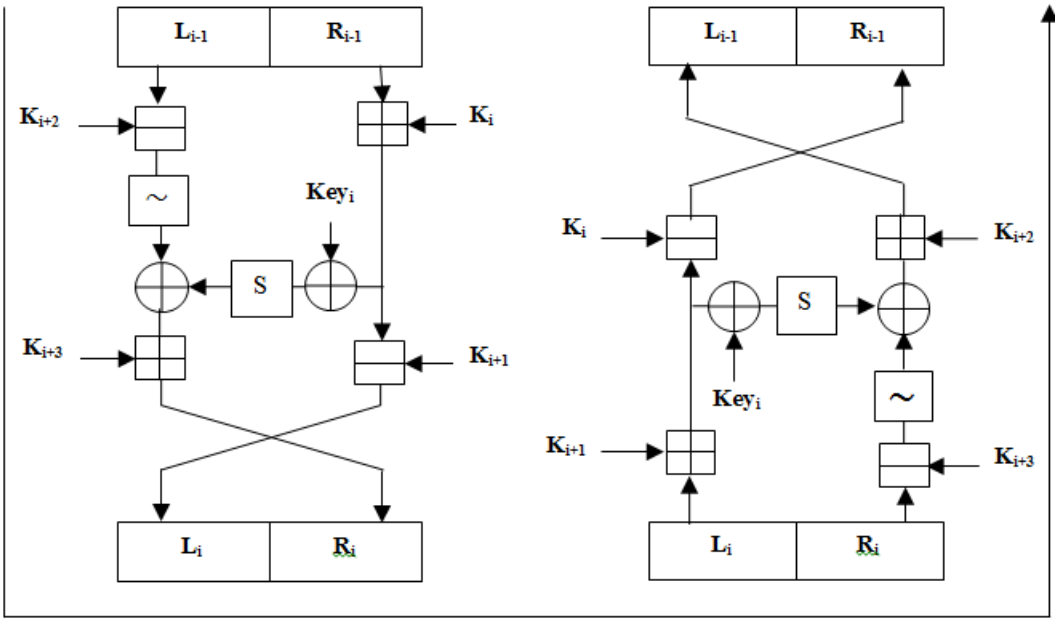


Figure 3. The reversing process for both Encryption and Decryption Algorithm for any Round i

## 4. Security considerations for the cipher

National Institute of Standards and Technology (NIST) launched a lightweight cryptography project that was tasked with understanding more about the issues and developing a scheme for the standardization of lightweight cryptographic algorithms. This project provides an overview of the lightweight cryptography project at NIST, and describes plans for the standardization of lightweight cryptographic algorithms. In this paper, NIST recommendations have been taken in consideration.

### 4.1. Substitution Boxes

The S-boxes S1, S2, S3 and S4 are the only nonlinear components of the cipher. It has been recommended to replace the usage of 8×8 S-boxes by a smaller 4×4 for use in constrained environments where the occupied memory space of current NIST cryptographic standards is not acceptable. On the other hand, Biham and Shamir proved that DES could be broken if poor S-boxes were used. For the DES-like cryptosystems to be cryptographically strong, so must their S-boxes. In other word, the security of DES like cryptosystems depends heavily on the strength of the S-boxes used. Therefore, the choice of cryptographically strong S-boxes is an important concern in design of secure cryptosystems. S1, S2, S3and S4 are chosen to satisfy the following cryptographically strong properties, Balancing, completeness, avalanche criteria, Strict Avalanche Criterion (SAC), non-linearity and robustness (XOR table distribution, and Linear Approximation Table (LAT).

The GOST block cipher is a Soviet and Russian government standard symmetric key block cipher which is the basis of most secure information systems in Russia (De Canni‘ere, 2005). The GOST block cipher has several versions. In this section, both GOST 28147-89 encryption algorithm, also known as simply GOST, and the most recent version of the standard, GOST R 34.12-2015, specifies that it may be referred to as Magma approaches have been observed. The GOST block cipher uses eight 4×4 S-boxes. These S-boxes have been analyzed to select the most rigid and powerful S-boxes (Courtois & Miształ, 2011; Courtois, 2012). On other words, sixteen S-boxes have been tested to select the most secure four S-boxes that exceed the security design criteria. Also, the criteria for selection of the S-boxes have been considered security against different types of cryptanalysis. Each box is used twice as shown in Fig. 1. Table 1 shows the selected S-boxes.

– **Substitution Boxes (S1, S2, S3 and S4)**

Table 1. The S-boxes of SLEA algorithm

| S-boxes | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| S1(x) | B | 3 | 5 | 8 | 2 | F | A | D | E | 1 | 7 | 4 | C | 9 | 6 | 0 |
| S2(x) | C | 8 | 2 | 1 | D | 4 | F | 6 | 7 | 0 | A | 5 | 3 | E | 9 | B |
| S3(x) | 5 | D | F | 6 | 9 | 2 | C | A | B | 7 | 8 | 1 | 4 | 3 | E | 0 |
| S4(x) | 1 | 7 | E | D | 0 | 5 | 8 | 3 | 4 | F | A | 6 | 9 | C | B | 2 |

#### 4.1.1. Security design criteria of substitution boxes

In cryptographic algorithms especially S-boxes, there are desirable properties which reflect the strength of the cryptographic transformation. These properties are stated as the following:

– **Balancedness Property**

Each of the Boolean functions of the S-box should be balanced, i.e., the number of zeros and ones in the truth table of the Boolean function must be equal.

– **Completeness Criterion**

For a Given one-one function f: {0,1}n –> {0,1}n, f is said to be complete if, for every i,j ∈ {1,...,n}, there exist two n-bits vectors X1, X2 such that X1 and X2 differ only in the ith bit and f(X1) differs from f(X2) at least in the jbit. In other words, if the input changes slightly (flipping a single bit) the output has to be changed at least in a single bit (flipping a single bit). Each output bit depends on all input bits this criteria is called completeness.

– **Avalanche effect**

For instance, one of the most important characteristics of an S-box is an avalanche criterion that is a bit change in the input byte of an S-box must result in a change in the output byte at least by 50% of bits.

– **Strict Avalanche Criterion**

The concepts of completeness and the avalanche effect can be combined to define a new property which we shall call the strict avalanche criterion. If a cryptographic function is to satisfy the strict avalanche criterion, then each output bit should change with a probability of one half whenever a single input bit is complemented.

Table 2 shows the Strict Avalanche Criterion (SAC) of the best four S-boxes. It is important to notify that the S-box S5 of the Russian Central Bank Federation has a SAC value better than the following boxes. But, the selection took place related to the overall evaluation of each box separately; these boxes have better nonlinear properties than the Russian Central Bank Federation S-box S5. Therefore, S5 has been rejected.

Table 2. The Strict Avalanche Criterion (SAC) of the SLEA S-boxes

| S - Boxes of most recent version of GOST | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SLEA-Box | GOST-Box | S[3] | S[2] | S[1] | S[0] | Max | Min | | SACAvg |
| S1 | S3 | 0.5625 | 0.5625 | 0.4375 | 0.5 | 0.5625 | 0.4375 | 0.25 | 0.515625 |
| S2 | S4 | 0.5625 | 0.5625 | 0.5 | 0.5 | 0.5625 | 0.5 | 0.125 | 0.53125 |
| S3 | S6 | 0.5625 | 0.5 | 0.625 | 0.5 | 0.625 | 0.5 | 0.25 | 0.546875 |
| S4 | S8 | 0.5 | 0.4375 | 0.5 | 0.5625 | 0.5625 | 0.4375 | 0.25 | 0.5 |

In S-boxes of the most recent version of GOST, S8 has the best average SAC value, but S4 has the smallest relative error this mean that the value of average SAC only may lead to inefficient result. It has to be taken in account both values of average SAC and relative error.

– **Robustness**

Seberry (Seberry, Zhang & Zheng, 1993) defines an expression for the robustness, *R*, of an S-box. The robustness (R) is a measure of the resistance of an S-box to differential cryptanalysis. Robustness is based on two features of the Differential Distribution Table (DDT). The first is the number of nonzero elements, *N*, in the first column of the DDT (excluding the first element). These denote instances when a change in the input results in no change in the output. Such occurrences are a weakness as they reduce the complexity of an algorithm and play an important part in differential cryptanalysis. The other feature is the largest value found in the DDT, *L*, other than the top left element. The robustness is given by equation 5:

$$R = (1 - N/2^n)(1 - L/2^n) \tag{5}$$

where $n$ is the number of input bits. The higher $R$ is, the more difficult differential cryptanalysis is to perform. Table 3 and Table 4 show the differential distribution and the linear approximation of the S-box (S1-box) respectively.

Table 3. The differential distribution (XOR distribution) of S1 (S3 of new GOST S-box)

| I/P - O/P XOR Difference | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 2 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 4 | 0 | 2 |
| 2 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 2 | 4 | 2 | 2 | 0 | 0 | 2 | 0 |
| 3 | 0 | 0 | 0 | 2 | 0 | 2 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 4 |
| 4 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 0 | 0 | 2 |
| 5 | 0 | 2 | 4 | 0 | 2 | 0 | 0 | 4 | 2 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| 6 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 0 | 2 | 0 | 2 | 2 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 4 | 0 | 0 | 0 | 4 | 0 |
| 8 | 0 | 0 | 4 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 4 | 2 | 2 | 0 |
| 9 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 0 | 2 | 0 | 2 |
| 10 | 0 | 0 | 0 | 0 | 4 | 0 | 2 | 2 | 0 | 2 | 0 | 2 | 2 | 0 | 0 | 2 |
| 11 | 0 | 2 | 2 | 2 | 4 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 |
| 12 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 0 |
| 13 | 0 | 2 | 2 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 4 | 2 |
| 14 | 0 | 2 | 0 | 2 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 2 | 2 | 2 | 0 | 0 |
| 15 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 2 | 0 | 0 | 4 | 2 | 0 | 0 | 0 |

Table 4. The linear approximation of S1 (S3 new GOST S-box)

| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | -4 | -2 | 0 | -2 | 0 | -2 | -2 | 0 | -2 | 0 | -2 | 4 | 2 | 0 |
| 0 | -2 | -2 | 2 | -2 | 0 | -4 | 0 | 0 | 0 | 4 | -2 | -2 | 2 | -2 | 0 |
| 0 | -2 | 0 | 0 | 2 | 2 | -2 | -4 | 0 | 2 | -2 | 2 | 0 | 0 | -2 | -4 |
| 0 | -2 | -2 | 2 | -2 | 0 | 4 | 0 | -4 | 0 | 0 | 2 | -2 | -2 | -2 | 0 |
| 0 | 4 | 2 | 0 | -2 | 2 | 0 | 2 | -2 | 0 | 2 | 2 | 0 | 2 | 0 | -4 |
| 0 | -4 | 0 | -2 | -2 | 0 | 2 | 2 | 2 | 2 | 0 | 2 | 2 | 4 | 0 | 0 |
| 0 | 0 | 2 | 4 | 2 | -2 | 0 | 2 | -2 | 4 | -2 | -2 | 0 | 2 | 0 | 0 |
| 0 | 0 | -2 | 2 | 0 | -2 | -2 | 0 | 0 | 2 | 2 | 4 | 2 | -2 | 4 | 0 |
| 0 | 0 | -4 | 0 | 4 | 4 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 2 | 2 | 0 | 2 | -2 | 0 | 0 | -2 | -2 | 4 | -2 | 2 | 0 | 4 |
| 0 | 2 | -2 | 0 | -4 | 0 | -2 | 2 | 2 | 2 | -4 | 0 | 0 | -2 | -2 | 0 |
| 0 | -2 | 0 | 2 | 0 | -2 | 0 | 2 | 2 | -4 | -2 | 0 | -2 | 0 | 2 | -4 |
| 0 | -2 | 2 | -4 | 0 | 0 | -2 | 2 | -2 | 2 | 0 | 0 | -4 | -2 | 2 | 0 |
| 0 | -2 | 0 | 0 | -2 | 2 | -2 | 0 | -4 | -2 | -2 | -2 | 4 | 0 | 2 | 0 |
| 0 | 0 | 0 | 2 | -2 | 4 | 2 | -2 | 2 | 2 | 0 | -2 | -2 | 0 | 4 | 0 |

The maximum values found in the XOR distribution and the linear approximation tables for all SLEA S-boxes are summarized in Table 5. The DPPmax is the same for all of the S-boxes, at 4/16=2^(-2), and the maximum bias probability of ±4/16.

Table 5. Summaryof the non-linear properties of GOST block cipher S-boxes

| S - Box of most recent version of GOST | | | |
|---|---|---|---|
| SLEA-Box | GOST-Box | Dmax(XOR Dist. Table) | Lmax(Linear App. Table) |
| S1 | S3 | 4 | ±4 |
| S2 | S4 | 4 | ±4 |

*Continued on next page*

*Table 5 continued*

| | | | |
|---|---|---|---|
| S3 | S6 | 4 | ±4 |
| S4 | S8 | 4 | ±4 |

The values of robustness for all of the SLEA S-boxes are calculated by applying equation 5 and are shown in Table 6. The results indicate that all of the S-boxes are approximately have the same robustness.

Table 6. Robustnessof GOST block cipher S-boxes

| S - Boxes of most recent version | | | | |
|---|---|---|---|---|
| SLEA-Box | GOST-Box | | L | RS |
| S1 | S3 | 0 | 4 | 0.75 |
| S2 | S4 | 0 | 4 | 0.75 |
| S3 | S6 | 0 | 4 | 0.75 |
| S4 | S8 | 0 | 4 | 0.75 |

## 5. Implementation Considerations

As shown before, SLEA is designed to facilitate both software and hardware implementation. Hardware implementation achieves high speed while the software implementation has the advantage of flexibility and low cost. The design principles for software implementation are stated as follows:

- ***Utilization of sub-blocks :*** Cipher operations should operate on sub-blocks that are natural for software, such as 8, 16, or 32-bits. SLEA can achieve that easily because it can be adapt to use 32 -bits sub-blocks.
- ***Use simple operations:*** Cipher operations should be easily programmed using addition, subtraction, shifting,complement, Xoring and so on. The elements of SLEA meet this requirement as well.

At the same time the design principles for hardware implementation are stated as follows:

- ***Similarity of encryption and decryption:*** Encryption should differ only in the way of using the key, so that the same device can be used for both encryption and decryption. SLEA has a structure that satisfies this requirement.
- ***Using a compact S-boxes:*** it can save the memory space and the implementation costs.
- ***Regular structure:*** The cipher should have a regular structure to facilitate VLSI implementation. SLEA is constructed from only one basic modular building blocks repeated multiple times.

## 6. SLEA Analysis

In this section, the characteristics of the SLEA relate to its cryptographic strength are illustrated as follow,

- ***Block length:*** The block length should be long enough to deter statistical analysis. This is valid for SLEA as well where 64-bit block is used.
- ***Key Length:*** The key length should be long enough to effectively prevent exhaustive key searches. With length 256-bits, SLEA seems to be secure in this area far into the future.
- ***Confusion:*** The ciphertext should depend on the plaintext and key in a complicated and involved way. The objective is to complicate the determination of how the statistics of the ciphertext depend on the statistics of

the plaintext. SLEA achieves this goal by using the four 4-bits S-boxes S1, S2, S3 and S4 and by mixing the different operations with the signal.

– ***Diffusion:*** Each plaintext bit should influence every ciphertext bit, and each key bit influence every ciphertext bit, the spreading out of the single plaintext bit over many ciphertext bits hides the statistical structure of the plaintext. In SLEA, the diffusion is provided by the four 4-bits S-boxes and the basic building block of the algorithm, known as the Subtraction/Complement/Xor/Addition (SCXA) structure as shown in Fig. 4 besides interchanging the two halves of the plaintext every round. This structure takes two (n/2)-bits values derived from the plaintext as input and two (n/2)-bits sub-keys derived from the key and produces (n/2)-bits output. Each output bit of the first round depends on every bit of the plaintext-derived input and on every bit of the sub-keys. This particular structure is repeated m times in the algorithm, providing very effective diffusion.

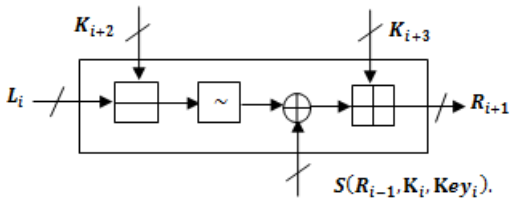


Figure 4. Subtraction/Complement/XOR/Addition(SCXA)

## 7. Conclusions

A proposed cryptography algorithm entitled “SLEA” has been introduced in this paper. The goal of our proposed algorithm is to provide a practical and secure cipher for low-resource applications. The benefit of the proposed SLEA its simplicity in terms of the used functions as well as the compact S-boxes that have been used. In addition, SLEA could be very beneficial to the wireless networks especially in networks with limited resources such as WSNs.

## 8. References

1. Atzori, L., Iera, A. & Morabito, G. (2010). The internet of things: a survey. *Comput Netwok,54*(15), 2787–2805.

2. Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems,29*(7), 1645–1660.

3. Want, R., & Dustdar, S. (2015). Activating the internet of things. *IEEEComputer Society,48*(9), 16–20.

4. Mariona, J., Hallman, R., Kline, M., Miguel, J., Major, M. & Kerr, L. (2016). Security in the industrial internet of things. *International Conference on Internet of Things and Big Data.* At Rome, Italy

5. Suo, H., Wan, J., Zou, C., & Liu., J. (2012). Security in the internet of things: A Review. *IEEE International Conference on Computer Science and Electronics Engineering*.

6. Ho, G., Leung, D., Mishra, P., Hosseini, A., Song, D., & Wagner, D. (2016). Smart locks: Lessons for securing commodity internet of things devices. *The11$^{th}$ ACM on Asia Conference on Computer and Communications Security* .

7. Nation*a*l Bureau of Standards, U.S. Department of Commerce. (1977, January). Data Encryption Standard. *Federal Information Processing Standard (FIPS).*

8. Stallings, W. (2009). *Cryptography and network security*: *Principals and Practices.*

9. Lai, X. & Massey, J. (1990). *A proposal for a new block encryption standard. In Proceedings of the EUROCRYPT 90 Conference,* 389-404.

10. Schneier. (1994, April). The Blowfish encryption algorithm.*Dr. Dobb's Journal of Software Tools,19*(4), 38-40.

11. Daemen & Rijmen, V. (1999, September 3). AES Proposal: Rijndael, AES algorithm submission. *International Journal of Computer Applications.*

12. Nechvatal, James., Barker, Elaine., Bassham, Lawrence., Burr, William., Dworkin, Morris., Foti, James., & Roback, Edward. (2001, May– June). Report on the Development of the Advanced Encryption Standard (AES). *International Journal of Computer Applications,* 106 (3).

13. Coppersmith, D. (1994, May). The Data Encryption Standard (DES) and Its Strength Against Attacks. *IBM Journal of Research and Development, 243 - 250.*

14. Biham, E., Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology,4*(l).

15. Leander, G., Paar, C., Poschmann, A., Schramm, K. (2007, March). New Lightweight DES Variants. Fast Software Encryption 2007, *LNCS, Springer-Verlag,* 26.-28.

16. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., ... Vikkelsoe, C. (2007, September 10). PRESENT: An Ultra-Lightweight Block Cipher. *International Workshop on Cryptographic Hardware and Embedded Systems,* Vienna, Austria: LNCS, Springer-Verlag.

17. Korte, T., & Silverlight. (2009, February). *Implementation of PRESENT. M.sc. thesis, Embedded Security Group.* Ruhr University Bochum.

18. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2013). The SIMON and SPECK Families of Lightweight Block Ciphers. *IACR Cryptology ePrint Archive.*

19. Rivest, R.L. (1994, December 14–16). The RC5 Encryption Algorithm. *Second International Workshop on Fast Software Encryption (FSE 1994),* Leuven, Belgium: LNCS 1008.

20. Wheeler, D.J., & Needham, R.M. (1994, December 14–16). TEA, A Tiny Encryption Algorithm. Proc. *Second International Workshop on Fast Software Encryption (FSE 1994),* Leuven, Belgium: LNCS *1008*, 363-366.

21. Christophe, De Canni'ere. (2005). GOST article. *Encyclopedia of Cryptography and Security,* 242-243.

22. Courtois, Nicolas, T., Miształ, Michał. (2011). Differential cryptanalysis of GOST . Cryptology ePrint Archive, from http: //eprint.iacr.org/2011/312.

23. Courtois, Nicolas T. (2012). An improved differential attack on full GOST. *Cryptology ePrint Archive,* from http: // eprint.iacr.org/2012/138.

24. Seberry, Jennifer., Zhang, Xian-Mo., Zheng, Yuliang. (1993, August 10). Systematic generation of cryptographically robust s-boxes. *The First ACM Conference on Computer and Communications Security.* New York, USA.