applied sciences

MDPI

Article

Design of a Lightweight Cryptographic Scheme for Resource-Constrained Internet of Things Devices

Nahla Ibrahim [1,*] and Johnson Agbinya [2]

[1] Computer Science and Information Technology, Sudan University of Science and Technology (SUST), Khartoum 11116, Sudan
[2] School of Information Technology and Engineering, Melbourne Institute of Technology, Melbourne, VIC 3000, Australia
* Correspondence: nahla480@outlook.com

Abstract: We propose an ultra-lightweight cryptographic scheme called "Small Lightweight Cryptographic Algorithm (SLA)". The SLA relies on substitution–permutation network (SPN). It utilizes 64-bit plaintext and supports a key length of 80/128-bits. The SLA cipher includes nonlinear layers, XOR operations, and round permutation layers. The S-box serves to introduce nonlinearity in the entire scheme design. It plays a vital role in increasing the complexity and robustness of the design. The S-box can thwart attacks such as linear and differential attacks. The scheme makes it possible to breed many active S-boxes in a short number of rounds, hindering analytical attacks on the cipher. When compared to other currently used ciphers, SLA has a higher throughput. Additionally, we demonstrate the SLA's performance as an ultra-lightweight compact cipher, and its security analysis. The SLA cipher's design is well suited for applications where small-scale embedded system dissipation is critical. The SLA algorithm is implemented using Python.

Keywords: Internet of Things (IoT); lightweight cryptographic scheme; SP-network; block cipher; encryption; resource-constrained

Citation: Ibrahim, N.; Agbinya, J. Design of a Lightweight Cryptographic Scheme for Resource-Constrained Internet of Things Devices. *Appl. Sci.* **2023**, *13*, 4398. https://doi.org/10.3390/app13074398

Academic Editor: Rashid A. Saeed

Received: 28 November 2022
Revised: 7 February 2023
Accepted: 10 February 2023
Published: 30 March 2023

1. Introduction

The current significant advancement in the network of intelligent objects is the Internet of Things (loT). It has various uses, such as radiation monitoring in nuclear power plants, smart cities, smart homes and smart environments in general, animal tracking, health monitoring, and many more applications. The main issues in IoT applications are energy management, IPv6 adoption, standardization, and security [1].

Any wireless cryptographic scheme has serious data security challenges. A cryptographic method is a crucial component of network security. For IoT systems, "Lightweight Cryptography (LWC)" is most suited. Lightweight cryptography is a protocol designed for use in constrained environments such as sensor networks, meters, healthcare, the Internet of Things, cyber-physical systems, intelligent energy systems, indicators, custom controls, etc. [2]. In addition, formal methods on IoT application layer protocols for improving security and detecting security issues remain an open challenge.

The motivation for developing SLA includes the following. Major standards organizations closely follow lightweight cryptographic development. Several lightweight cryptographic methods have been standardized by The International Organization for Standards (ISO) and the International Electro-technical Commission (IEC). These standard bodies are also reviewing more algorithms to include in their standards. Recently, the National Institute of Standards and Technology (NIST) made public its lightweight algorithms portfolio requirements [3]. This announcement came after the Institute held two workshops on lightweight cryptography [4,5].

Enforcing security against attacks on IoT systems is a challenging task. Once the required features of an IoT system have been added, cryptographic engineering aims to

harmonize conflicting requirements to create secure yet usable embedded IoT systems. The field of LWC for IoT applications combines cryptography, computer science, and electrical engineering. Solutions to IoT security problems are converging on cryptographic engineering. This paper attempts to reconcile these conflicting requirements. In this paper, we propose a new lightweight cryptographic scheme built from finite fields, using their underlying mathematical structure. In general, this choice does not influence the security of the scheme, but influences the performance of the resulting implementation. This approach has become a major design trend in cryptography, due to the increasing importance of small-scale embedded IoT devices. In a lightweight cryptographic scheme, the construction of a strong non-linear S-box (confusion layer) uses Galois field multiplication which meets cryptographic properties, and provides a novel method to construct diffusion layers by 32-bit binary matrix. The resulting proposed SLA scheme provides a sufficient security level against most of the well-known attacks on block ciphers, such as linear and differential cryptanalysis.

This paper is organized as follows. Section 2 is a presentation of research work. Section 3 is a depiction of the block cipher SLA. Section 4 is an assessment of the block cipher SLA, and Section 5 is the conclusion.

2. Related Work

A significant number of authors, in the current literature, have undertaken investigations into how to improve IoT security and privacy. This section derives input from current solutions for small cryptographic algorithms. They also discussed how to enhance the level of security for the IoT.

In [6] a new variant lightweight cryptography algorithm for the Internet of Things is proposed, which is called New Variant Lightweight Cryptography (NVLC). The main idea of the design of NVLC is to use a 4-bit S-box with a lower signal delay in comparison to an 8-bit S-box. Additionally, it used the Whitening key idea at the beginning and end of encryption to raise the difficulty of key search and the difficulty of attacking the cipher. However, the encryption methods investigated the goal of a high level of security in the low-resource device for NVLC block cipher design.

SFN [7], a new lightweight algorithm, employs a 96-bit key on a 64-bit block. The novel idea of the design is to use a different encryption method that takes both SP network structure and Feistel network structure to encrypt. Involution related properties of the nonlinear and linear components are employed into the design of SP network structure. The modified SP network structure enables the encryption and decryption program or circuit to work as the Feistel network structure. The encryption method satisfies the security requirements of different user levels. It gave a good performance in hardware at 1876 GEs.

A study in [8] proposed a simplified new version of the round function of the original SIMON by reducing its impact by changing the shift numbers, so the first rotation is removed to enhance the speed of SIMON and execution time.

The Feistel scheme is used to encrypt the lightweight block cipher LiCi [9]. LiCi has a 128-bit key, a 64-bit block, and 31 rounds. The LiCi design uses the substitution layer derived from the Karnaugh Map that applies 4×4 S-boxes, which has been employed to reduce the logic gates of the S-box, and use circular shift by (3, 7). The encryption method offers good performance, both on hardware at 1153 GEs and on software platforms.

BORON, a low-power cipher proposed in [10], boasts being ultra-weight and compact. It works with 128/80-bit keys and 64-bit plaintext over the SP network. Their methods used for encryption gave excellent performance of 1939 GEs in a small area. It performs efficiently on both hardware and software platforms.

In [11], a family of low energy block ciphers called Midori is proposed, which is composed of two variants: Midori64 and Midori128. The design of Midori is to make use of cell-permutation layers 4×4 involutory Binary MDS matrix to optimize diffusion speed, and two types bijective 4-bit S-boxes. The encryption methods satisfy the optimization goal of low energy for block cipher design.

The study in [12] presented Simeck, a lightweight block cipher designed from components of other ciphers, SIMON and Speck. The study proved the ability to design ciphers that have less power consumption and are relatively smaller in area.

The RECTANGLE algorithm [13] proposed new design criteria for the RECTANGLE S-box. The main idea of the design of RECTANGLE makes use of the bit-slice style in a lightweight manner, and was introduced for speeding up the software speed in the design of the DES and Serpent block cipher [14,15]. It offers a very low cost in hardware but also is very competitive in software speed.

In [16], a lightweight, versatile block cipher called TWINE is proposed. The global structure of TWINE is a type-2 generalized Feistel structure (GFS). A round function of TWINE consists of a non-linear single 4-bit S-box rather than multiple ones, which can contribute to smaller (serialized) hardware and software implementations and different block shuffle from the original (cyclic shift), which can greatly improve the diffusion speed of type-2 GFS. Despite the fact that bit-shifting operations are often used in the diffusion layer of many lightweight block ciphers (e.g., PRESENT and NOEKEON), they actually lose their efficiency in software implementations. Therefore, (PRESENT-like and NOEKEON-like) diffusion techniques [17,18] are not an option for TWINE.

Bogrof et al. [19] presented PRINCE, which provides a new dimension to lightweight cryptography by achieving low latency. It also focuses on hardware implementation. It utilized a 128 bits key and was comprised of 64 bits block with 12 rounds. The S-box of this cipher was non-linear i.e., Feistel structure. The main advantage of the Feistel structure is that the same program code can be used for the encryption and decryption process. It also helps in reducing memory usage. The cipher can however be susceptible to related-key attacks if the Feistel structure uses alternating keys. Some other noteworthy mentions from this generation are Humming-Bird, KASUMI, and Piccolo

In [20], a symmetric cryptographic algorithm, KLEIN, which has the benefit of better performance of software on legacy sensor platforms, is proposed. The fact that it uses a 4-bit S-box permutation via the algebraic normal form (ANF), rather than an 8-bit S-box, whether implemented in hardware or software, results in a tiny hardware implementation. KLEIN's design increases the available options of lightweight block ciphers for low-resource applications.

In [21], Leander et al. proposed a family of new lightweight variants of DES (data encryption standard), which are called DESL/DESX/DESXL (the lightweight modified versions of the well-known DES). The main idea of the new variants of DES is to use just one S-box recursively, instead of eight different S-boxes, to minimize the hardware implementation.

mCrypton [22] is designed by following the overall architecture of Crypton [23], but with redesign and simplifications of each component function to enable much more compact implementation in both hardware and software.

Based on the state of current small cypher results, it is essential to provide not only a small cypher footprint to fit into small memories, but also to enhance speed and cryptographic strength by making it difficult for linear and differential cryptanalysis.

3. Block Cipher SLA

To ensure difficulty in differential and linear cryptanalysis, SLA uses a substitution–permutation network [24], having 16 rounds with 16 keys. The block size is 64 bits with an 80- and 128-bit key size. The block diagram of the SLA cipher is shown in Figure 1, and Figure 2 shows the detailed SLA block cipher.

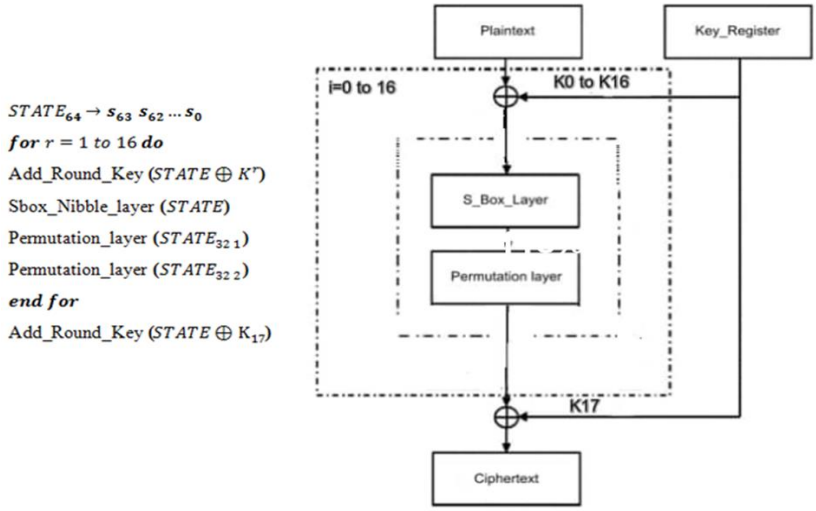

Figure 1. SLA Block diagram.

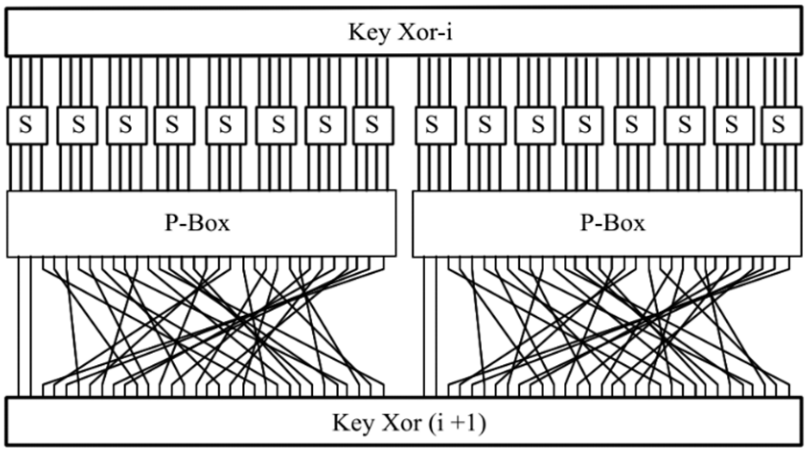

Figure 2. SLA Block cipher.

Each of the sixteen rounds includes an eXclusive OR ('XOR') logic operation to obtain new keys K_r for $1 \leq r \leq 16$. They are produced by the 80/128-bit key register. Finally, one additional key will then be produced and XOR-ed to obtain the ultimate ciphertext. The confusion layer represents a non-linear substitution (box) table. The diffusion layer represents one of the durable layers between extant LWC schemes. Figure 2 depicts the block cipher, including pseudocode, with each phase.

3.1. Add_Round_Key

The Add_Round_Key performs an eXclusive OR ('$\oplus$') on a 64-bit plaintext and with a 64-bit sub-key produced from the 80/128-bit key register. $K_i \rightarrow k^i_{63} \cdots k^i_0$ defines sub-keys for $1 \leq i \leq 16$, and the actual output $STATE_{64} \rightarrow s_{63}s_{62} \cdots s_0$ is given as

$$STATE \rightarrow STATE \oplus K^i$$

3.2. Substitution Box (S-Box_Nibble Layer)

The single S-box used in our scheme is $S : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^4$ S-box. The substitution layer is represented in a hexadecimal form in Table 1. Values in Table 1 are readily implemented using a table of sixteen four-bit values. This is a direct result of the search for a lightweight cryptographic algorithm.

Table 1. A Sbox_Nibble Layer of our scheme.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S(**X**)	f	8	3	e	0	7	b	a	5	d	9	c	6	4	2	1

3.3. Permutation Layer

The permutation layer creates a mixed 32-bit output from a 32-bit input. At the 32-bit number bit position, a 32-bit-sized bit is replaced by bit x. The diffusion function can be expressed as a bit permutation level of 1-bit words using the equations below. Appendix A contains a 32 by 32 one-to-one permutation matrix and its inverse. The permutation matrix performs the binary permutation operation in (1) as

$$P_{ij} = P_{ij}X_{ij} \tag{1}$$

The reason for using a one-to-one permutation matrix is to scramble the key. The desire is to rearrange the values in the key so that it looks like someone knows the original key. In a one-to-one permutation matrix, the values of the key do not change; the values only change position. A one-to-one permutation matrix is very fast because one-to-one swapping of the position is fast. For example, if the RaspberryPi device tests a 10×10 number, it takes more time to perform 10 multiplications by 10 in the microcontroller; however, a one-to-one permutation matrix is very fast.

Algorithm 1 summarizes the encryption process outlined in Sections 3.1–3.3.

Algorithm 1 Encryption

Input: Plaintext $STATE_{64} \rightarrow s_{63}s_{62} \cdots s_0, S[16], P[32]$
Output: Ciphertext C_{64}
for r = 0 *to* 16 ***do***
$STATE_{64} \rightarrow s_{63}s_{62} \cdots s_0$
$STATE \rightarrow STATE \oplus K^i$
$STATE_s \rightarrow S[STATE]$ // S-box
$STATE_{p1} \rightarrow P\left[STATE_{s\ high}\right]$ // High 32-bit P-box
$STATE_{p2} \rightarrow P[STATE_{s\ low}]$ // Low 32-bit P-box
$STATE_{round\ cipher} \rightarrow STATE_{p1} + STATE_{p2}$
end for
$C_{64} \rightarrow STATE \oplus K^{17}$

3.4. Key Schedule of 80- and 128-Bit Key Size

The key scheduling algorithm is one of the most crucial parts of any cryptographic scheme; it determines the cipher's intricacy. Cryptography has undoubtedly evolved since the days of Kerckhoff. The key schedule of SLA is inspired by the key schedule of PRESENT [17]. No attacks have been reported to date on the PRESENT scheme key scheduling. The SLA scheme key scheduling has a total of 16 sub-keys with a 64-bit key size.

1. Scheduling of 80-bit key

The key register KEY contains the 80-bit key provided by the user, specified as $KEY = k_{79}k_{78} \cdots k_0$. From round i, the 64-bit sub-keys least significant bit (LSB), $K_i = k_{63}k_{62} \cdots k_0$ obtained in (2):

$$K^i = k_{63}k_{62} \cdots k_0 \tag{2}$$

The register KEY is updated after obtaining the 64-bit key in (3–5):

$$KEY <<< 13; \tag{3}$$

$$[k_3k_2k_1k_0] = S[k_3k_2k_1k_0]; \tag{4}$$

$$[k_{63}k_{62}k_{61}k_{60}k_{59}] = [k_{63}k_{62}k_{61}k_{60}k_{59}] \bigoplus RC^i \quad (5)$$

For 0 to 16 rounds, five bits of the round counter i are XOR-ed with the five bits of key register KEY, i.e., from k_{59} to k_{63}.

2. Scheduling of 128-bit key

The key register KEY contains the 128-bit key provided by the user, specified as $KEY = k_{127}k_{126}\cdots k_0$. From the round i, 64-bit sub-keys least significant bit (LSB), $K_i = k_{63}k_{62}\cdots k_0$ obtained in (6):

$$K^i = k_{63}k_{62}\cdots k_0 \quad (6)$$

The register KEY is updated after obtaining the 64-bit key obtained in (7–10):

$$KEY <<< 13; \quad (7)$$

$$[k_3k_2k_1k_0] = S[k_3k_2k_1k_0]; \quad (8)$$

$$[k_7k_6k_5k_4] = S[k_7k_6k_5k_4]; \quad (9)$$

$$[k_{63}k_{62}k_{61}k_{60}k_{59}] = [k_{63}k_{62}k_{61}k_{60}k_{59}] \bigoplus RC^i. \quad (10)$$

3.5. The Decryption Process

The decryption process is the reverse of the encryption procedure. Each layer is reversible. The subkeys are generated in reverse order by the key schedule by using the transformation round keys. The decryption process involves the same number of rounds of encryption, where its processes are performed in each round. They are, however, the reverse of each other. In the add_round_key layer, the inverse is achieved by XORing the same round key to the block. In the S-box, and in the permutation layer, the inverse function is used in the decryption process, using the result that $A \bigoplus A \bigoplus B = B$.

4. Evaluation of SLA Scheme

4.1. Security Evaluation

1. Linear cryptanalysis

Linear cryptanalysis [25,26] is one of the most widely used techniques for breaking block ciphers. To evaluate the difficulty of the linear cryptanalysis of the SLA scheme, we present a minimal bound on the number of so-called "active" S-boxes defined in a linear characteristic. Table 3 presents the linear characteristic for the SLA scheme, and Table 4 shows the minimal number of active S-boxes in the linear characteristic.

Theorem 1. *For sixteen rounds of SLA, it features 48 active S-boxes, and a maximum probabilistic bias linear characteristic is* 2^{-55}.

Theorem 1 is formally proved in Appendix C.

Walsh transform. The Walsh transform of the Boolean function $f : \mathbb{F}_2^n \to \mathbb{R}$ with $n-$ variables is defined as

$$a \to \varepsilon(f + \varphi_a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+a.x} \quad (11)$$

The Walsh coefficient of f at point a is denoted by the value $\varepsilon(f + \varphi_a)$, and the Walsh spectrum of $\mathbb{F}$ is denoted by the multiset consisting of all Walsh coefficients of f.

Walsh transform. The bias (aka, correlation or imbalance) of a Boolean function f with $n-$ variables is defined as

$$\varepsilon(f) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)} = 2^n - 2\sqsupseteq\sqcup(f) \quad (12)$$

In other words,

$$\Pr_X[f(X) = 1] = \frac{\exists\sqcup(f)}{2^n} = \frac{1}{2}(1 - \frac{\varepsilon(f)}{2^n}) \tag{13}$$

2. Differential Cryptanalysis

Differential cryptanalysis [26,27] is the main form of attack on symmetric block ciphers. Differential paths are formed by considering the differences between inputs and outputs with a high probability for each round. The S-box where the differences between the inputs or the outputs are nonzero is called the active S-box. SLA presents a minimal bound on the number of so-called "active" S-boxes defined in a differential characteristic to assess the hardness of the SLA scheme differential cryptanalysis.

Appendix E shows the linear/differential relations SLA scheme S-box. Table 5 shows the differential characteristics SLA scheme, and Table 6 shows the minimal number of active S-boxes in the differential characteristic.

Theorem 2. *For sixteen rounds of SLA, it features 48 active S-boxes, and a maximum probabilistic differential characteristic for the sixteen rounds is* 2^{-96}.

Theorem 2 is formally proved in Appendix D.

Autocorrelation. The autocorrelation transform taken concerning to $a \in \mathbb{F}_2^n$, of Boolean function f with $n-$ variable is denoted by $\hat{r}_f(a)$ and defined as (14):

$$\hat{r}_f(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(x \oplus a)} \tag{14}$$

Differential uniformity. Given differential uniformity for any vectorial Boolean function $\mathbb{F} \in \mathcal{F}n, m$ for any $a \in \mathbb{F}_2^n$ into $b \in \mathbb{F}_2^m$, we defined in (15):

$$\delta(a, b) = \#\{x \in \mathbb{F}_2^n \ : S(x + a) + S(x) = b\} \tag{15}$$

Then, the differential spectrum of $\mathbb{F}$ is the multi-set $\{\delta\{a, b); a \in \mathbb{F}_2^n \backslash \{0\}; b \in \mathbb{F}_2^m\}$, and its maximum (16):

$$\delta_{\mathbb{F}} = \max_{a \neq 0, b} \ \delta(a, b) \tag{16}$$

3. Algebraic degree. An algebraic degree of a vectorial Boolean function $\mathbb{F} \in \mathcal{F}n, m$ is the number of variables in the longest item of its ANF, denoted by $\mathcal{D}\ \varepsilon\mathcal{G}(\mathbb{F})$ [28].

Nonlinearity. The nonlinearity of Boolean function $f \in \mathbb{F}_m$ is defined as the Hamming distance between f and the set $\mathcal{A}_n$ of all affine functions (or linear) [29] in (17)

$$\mathcal{NL}_f = \min_{\varphi \in \mathcal{A}_n} \exists\sqcup\left(f \bigoplus \varphi\right) \tag{17}$$

4. Nonlinearity. The nonlinearity of the vectorial Boolean function $\mathbb{F} \in \mathcal{F}_{n,m}$ is the minimal of all component functions of $\mathbb{F}$ [28], and the Walsh spectrum is used to calculate it in the manner outlined below in (18):

$$\mathcal{NL}(\mathbb{F}) = \min_{b \neq 0 \in \mathbb{F}_m} \ \mathcal{NL}(b, \mathbb{F}) = 2^{n-1} - \frac{1}{2}\max(WS(a, b)) \tag{18}$$

5. Correlation Immunity. The correlation immunity of a Boolean function $f \in \mathbb{F}_n$ is defined as a measurement of how uncorrelated outputs are with a certain subset of its inputs. If f is balanced, and $t - C\mathcal{I}$, then so-called $t -$ resilient [30]. This criterion is from the Walsh spectrum in the manner outlined in (19):

$$\hat{\theta}_{\mathbb{F}}(a, b) = 0, \ \forall\, a \neq 0 \in \mathbb{F}_n, 1 \leq \exists\sqcup \leq t, \forall\, b \neq 0 \in \mathbb{F}_m \tag{19}$$

6. Balancedness. The vectorial Boolean function $\mathbb{F} \in \mathcal{F}_{n,m}$ is balanced if its outputs are distributed uniformly over $\mathbb{F}_2^m$. According to the Walsh Spectrum, this property is evaluated as follows [31]:

$$\hat{\theta}_{\mathbb{F}}(0, b) = 0, \ \forall\, b \neq 0 \in \mathbb{F}_m \tag{20}$$

7. Algebraic immunity. Algebraic immunity of a Boolean function $f \in \mathbb{F}_n$ is defined as the least degree of all annihilators of f or $1 + f$ is designated by notation $(\mathcal{AI}(f))$ [32–34].
8. Global avalanche criterion (GAC). The global avalanche criterion is presented through two indicators [35].

   First, the absolute indicator, denoted by MAXAC.

$$AC_{max}(\mathbb{F}) = \max(|AC(\mathbb{F})(a, b)|) \forall\, a \neq 0 \in \mathbb{F}_n, \ \forall\ \neq 0 \in \mathbb{F}_m \tag{21}$$

   Second, the sum − of − squares indicator, denoted by σ.

$$\sigma(\mathbb{F}) = \sum_{(a,b) \in \mathbb{F}_n x \mathbb{F}_m} AC(\mathbb{F})(a, b)^2 = \frac{1}{2^n} \sum_{(a,b) \in \mathbb{F}_n x \mathbb{F}_m} WS(\mathbb{F})(a, b)^4 \tag{22}$$

When cryptographic functions have achieved low values of both indicators, they reach the best diffusion.

9. Propagation criterion. The propagation criterion of vectorial Boolean function $\mathbb{F} \in \mathcal{F}_{n,m}$ satisfies the $\mathcal{P}C(l)$. This property is from the Walsh spectrum in the manner outlined below [36,37]:

$$r_{\mathbb{F}}(a, b) = 0, \ \forall\, a \in \mathbb{F}_n, \ 1 \leq \exists \sqcup \leq l, \ \forall\, b \neq 0 \in \mathbb{F}_m \tag{23}$$

10. Linear potential. The linear potential of a vectorial Boolean function $\mathbb{F} \in \mathcal{F}_{n,m}$ is a metric of linearity that fulfills $2^{-n} \leq \mathcal{LP} \leq 1$ [38]. Therefore, the upper bound is met when $\mathbb{F}$ is linear or affine, whereas the tight bound holds if and only if $\mathbb{F}$ exhibits maximal nonlinearity ($\mathbb{F}$ is bent), and it is defined as

$$\mathcal{LP}(\mathbb{F}) = \frac{1}{2^{2n}}. \max(WS(\mathbb{F})(a, b)^2 \tag{24}$$

11. Differential potential. The differential potential of a vectorial Boolean function $\mathbb{F} \in \mathcal{F}_{n,m}$ is a gauge of resistance to differential attack where $2^{-m} \leq \mathcal{DP} \leq 1$, and the lower bound is valid if $\mathbb{F}$ is bent and the upper bound is met when $\mathbb{F}$ is linear or affine, and it is defined as

$$\mathcal{DP}(\mathbb{F}) = 2^{-n}\delta(\mathbb{F}) \tag{25}$$

12. Fixpoints and negated-fixpoints. A vectorial Boolean function $\mathbb{F} \in \mathcal{F}_{n,m}$ represents the fixpoints of $\mathbb{F}$, that is, $\{x|F(x) = x\}$ and negated-fixpoints of $\mathbb{F}$, that is $\{x|F(x) = \bar{x}\}$.

4.2. The Effect of the Avalanche

A cipher with a strong avalanche effect has a better chance of resisting most possible attacks, since even minor input changes have a big impact on the output. In SLA, by changing just one bit in the plaintext/key bits of the input in SLA, the output was observed. It was observed that more than half of the ciphertext bits are impacted by a single bit change in the SLA cipher's key. Tables 8 and 9 show the effect of the avalanche. This is the intended outcome in the design of SLA in this paper [39].

4.3. Performance Evaluations

Here, we analyzed the performance of SLA further. Based on the AMD Ryzen 45500U processor with 64-bit 4 GHz, Table 10 provides a thorough comparison between SLA and other contenders.

4.4. Results

An analysis of the probabilistic linear relations shows that the nonlinearity property (item $\mathcal{NL}$ in Table 2) is four, while the highest value for a Sbox_Nibble is six. The linear potential (item $\mathcal{LP}$ in Table 2) is 0.25 over the best known for a Sbox_Nibble with four input variables, which is 0.0625. The findings demonstrate that a 4×4 Sbox_Nibble Layer offers good resistance to linear attacks.

Table 2. Comparison with Respect to Cryptographic Criteria of Sbox_Nibble Layer for our design Approach and Mini AES.

Criteria	Lower Bound	Upper Bound	Our S-Box	Mini AES S-Box	Ref.
$\mathcal{D}$ $\varepsilon\mathcal{G}$	0 (constant functions)	**n**	3	2	[40]
$\mathcal{NL}$	0 (affine functions)	$\mathbf{2^{n-1}-2^{\frac{n}{2}-1}}$**(≥2m and n even)** $\mathbf{2^{n-1}-2^{\frac{n-1}{2}}}$ **(n<2m and n odd)**	4	2	
C$\mathcal{L}$	0	**n**	1	1	
S-box Balanced	-	-	Balanced	Balanced	
$\mathcal{AL}$	0	[32]$\lceil \frac{n}{2} \rceil$	2	2	
MAXAC	0 (bent functions)	$\mathbf{2^n}$(affine functions)	8	16	
σ	$\mathbf{2^{2n}}$ (bent functions)	$\mathbf{2^{3n}}$ (affine functions)	640	1408	
$\mathcal{LD}$	0 (if it has linear structures)	$\mathbf{2^{n-2}}$	2	0	
$\mathcal{P}$**C**	0	**n**	1	1	
$\mathcal{LP}$	0.0625 ($\mathbf{2^{-n}}$)	1	0.25	0.5	
$\mathcal{DP}$	0.0625 $\mathbf{(2^{-m})}$	1	0.25	0.5	

Investigation and analysis of the probabilistic differential relations of SLA were undertaken to establish its resistance to probabilistic differential attacks. The result shows that the linearity distance Sbox_Nibble (item $\mathcal{LD}$ in Table 2) is two over a maximal value of four. The differential potential (item $\mathcal{DP}$ in Table 2) is identical to 0.25 over the best known for a Sbox_Nibble with four input variables, which is 0.0625. These findings demonstrate that Sbox_Nibble has the best defense against differential attacks.

The robustness of our S-box design compared to the Mini AES S-box is shown in Table 2. For a 4×4 S-box, the Mini AES does not provide effective defense against linear attacks. In addition, it does not exhibit the best defense against differential attacks.

The algebraic degree of the Mini AES S-box is two. This number is too low for immunity against high order differential attacks. Consequently, algebraic attacks can be efficiently executed if a multivariate algebraic equations system is solved. The S-box Mini AES's absolute indicator reaches the upper bound of 16 and its sum−of−squares indicator is close to 4096, hence its inability to achieve a great diffusion. Table 2 provides an overview of the findings for these criteria.

The range of values accepted by the Walsh transform of the Sbox_Nibble are 8, 4, 0, −4, and −8.

The range of values accepted by the linear profile are 64, 16, and 0; the range of values accepted by the differential profile are 1, 024, 512, and 0; finally, the autocorrelation has three levels: 8, 0, and −8.

The results of the linear characteristic and the differential characteristic SLA scheme are shown in Tables 3 and 4.

Table 3. Linear Characteristics for the SLA scheme.

Rounds	Input to S-Box	Output of S-Box
First	0000 0000 0008 0000	0000 0000 0003 0000
Second	0000 0000 5000 5000	0000 0000 c000 c000
Third	0360 0060 0300 0360	04b0 00b0 0400 04b0
Fourth	2024 2004 0040 204c	5051 5001 0010 5012

Table 4. Minimal Number of Active S-boxes from the Linear Characteristics.

Rounds	Min. No. of Active S-Boxes
First	1
Second	3
Third	9
Fourth	18

The results of the minimal number of active S-boxes in the linear and differential characteristics are shown in Tables 5 and 6.

Table 5. Differential Characteristics for the SLA scheme.

Rounds	Input-to-S-Box	Output-of-S-Box
First	0000 0000 000e 0000	0000 0000 0009 0000
Second	0000 0000 2000 2000	0000 0000 c000 c000
Third	0360 0060 0300 0360	0840 0040 0800 0840
Fourth	0140 0045 0120 0100	0710 0012 07c0 0700

Table 6. Minimal Number of Active S-boxes in the Differential Characteristics.

Rounds	Min. No. of Active S-Boxes
First	1
Second	3
Third	9
Fourth	20

The comparison of the linear and differential attack of SLA with the other algorithms is shown in Table 7. The results show that 16 rounds of SLA are secure enough against differential and linear attacks.

Table 7. Comparison of Linear and Differential Attacks.

LWC Algorithm	No. of Rounds	No. of Active S-Boxes	No. of Known Plaintext	No. of Chosen Plaintext	Ref.
SLA	16	48	2^{110}	2^{96}	This paper
BORON	18	48	2^{98}	2^{96}	[10]
ANU	18	54/48	2^{110}	2^{96}	[41]
FEW	27	45	2^{90}	2^{90}	[42]
L-Block	15	32	2^{66}	2^{64}	[43]
PICCOLO	30	30	2^{120}	2^{120}	[44]
PRESENT	25	50	2^{102}	2^{100}	[17]

By changing just one bit in the input plaintext/key bits, the output seen in Tables 8 and 9 is produced. Each time a bit in the key is changed when using the SLA scheme; over half of the ciphertext bits are also changed.

Table 8. The Effect of the Avalanche on SLA-80.

Plaintext	Key	Ciphertext	No. of Bits Altered	Rate
0000 0000 0000 0000	0000 0000 0000 0000 0000	740434f796cff821		
	0010 0000 0000 0000 0000	0020313a0c9157ee	34	53%
	0000 0000 0000 0000 0010	bcc58124f3b581de	34	53%

Table 9. The Effect of the Avalanche on SLA-128.

Plaintext	Key	Ciphertext	No. of Bits Altered	Rate
0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000	858f96a55cc4f107		
	0000 0000 0000 0000 0000 0000 0000 0010	48987e3bd2bc193b	34	53%
	0000 8000 0000 0000 0000 0000 0000 0000	00f7818f94a2e296	39	60%

The SLA scheme is designed in such a way that it provides optimum performance. The performance of SLA and relevant LWC algorithms in software on AMD Ryzen 45500U processor are shown in Table 10. Interestingly, the execution times of SLA are small, typically less than 0.000659451 s for all of them, which would fit quite well in current IoT devices. They also require little power to process. SIMON and CLEFIA have the highest execution times of 0.02201274 s and 0.022619188 s, respectively. SLA has the highest throughput of 97,050.44167 kilobytes per second. This performance can further be improved by using faster processors in IoTs.

Table 10. The Performance of SLA and Relevant LWC Algorithms in Software.

Performance on AMD Ryzen 45500U Processor						
Structure	**LWC Algorithm**	**Block Size**	**Key Size**	**Execution Time**	**Block Size**	**Key Size**
SP network	SLA	64	80	0.000659451	97050.44167	1631127
	PRESENT	64	80	0.015804805	4049.401468	238283
	LED	64	64	0.020789735	3078.442315	11398908
	KLEIN	64	64	0.021051668	3040.139201	689344
	AES	128	128	0.016058415	3985.44933	727652
Feistel network	DES	64	56	0.015747033	4064.257677	37362051
	TWINE	64	80	0.02186244	2927.395138	2118524
	SPECK	48	96	0.021938369	2917.2634	228783
	SIMON	48	96	0.02201274	2907.407207	247285
	CLEFIA	128	128	0.022619188	2829.456145	1074972

In the autocorrelation coefficients, the absolute indicator for Sbox_Nibble is 8, and for the sum − of − squares indicator is 640. These results show that Sbox_Nibble obtains a reasonably acceptable diffusion since its absolute indicator is closer to the lower bound and is 0 compared to the upper bound, where 16 is. Similarly, the sum − of − squares indicator has theoretical bounds of 256 and 4096, and is extremely close to the 256 lower bound.

When the cryptographic functions attain lower bound for both indicators, the ideal diffusion will be attained. The nibble S-box is represented using the algebraic normal form (ANF):

$$\begin{aligned} S(x_1) &= 1 + x_0 + x_2 + x_3 + x_0x_1 + x_1x_2 + x_1x_3 + x_2x_3 + x_0x_1x_2 + x_0x_2x_3 \\ S(x_2) &= 1 + x_0 + x_3 + x_0x_1 + x_0x_3 + x_2x_3 + x_0x_1x_2 + x_0x_1x_3 \\ S(x_3) &= 1 + x_0 + x_1 + x_3 + x_0x_2 + x_0x_3 + x_1x_3 + x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_1x_2x_3 \\ S(x_4) &= 1 + x_1 + x_3 + x_1x_2 + x_2x_3 + x_0x_1x_3 + x_0x_2x_3 + x_1x_2x_3 \end{aligned} \tag{26}$$

These forms show that Sbox_Nibble, when compared with others with a maximum algebraic immunity of 2, has an algebraic immunity degree of 3, which is sufficiently high to protect it from higher-order differential attacks. As a result, carrying out algebraic attacks through solving a multivariate algebraic equation system is difficult.

The cipher structure has no visible flaws such as the absence of fixpoints/negated-fixpoints. In addition, a low-level cryptographic algorithm with a rising number of fixpoints or negated-fixpoints lacks the required randomness, so it is not considered to be well designed.

5. Conclusions

This paper has presented SLA, a lightweight scheme. Because SLA is based on the SP-network, it is faster than the Feistel-based cipher. Additionally, the proposed SLA scheme employs a novel encryption method, including finite field multiplication, to construct nonlinear S-box (confusion layer) and the effective one-to-one matrix linear permutation (diffusion layer), which leads to satisfactory security requirements without losing performance efficiency on both execution time and throughput. We exploited properties related to the nonlinear and linear components to design SP network structure. As we designed the SLA, we researched the minimal numbers of active S-boxes and good S-boxes. We also researched the hamming weight calculation for LAT and DDT entries. The proposed SLA design has achieved a small execution time, high throughput, and high level of security. This makes it suitable for small-scale embedded environments such as RFID tags and wireless sensor nodes. Advanced attacks can be used to examine the SLA scheme further. We expect our results to be applied in other domains as well.

Author Contributions: Conceptualization, N.I. and J.A.; Methodology, N.I. and J.A.; Software, N.I.; Validation, N.I. and J.A.; Formal analysis, N.I. and J.A.; Investigation, J.A.; Data curation, N.I.; Writing—original draft, N.I.; Writing—review & editing, J.A.; Visualization, J.A.; Supervision, J.A. All authors have read and agreed to the published version of the manuscript.

Funding: This PhD study is self-funded with no external funding.

Institutional Review Board Statement: Reviewed by my PhD supervisor.

Informed Consent Statement: I Johnson Agbinya PhD Supervisor for Nahla Ibrahim consent and approve to publish this research output as a requirement for examination of her PhD thesis.

Data Availability Statement: Contact Nahla Ibrahim, nahla480@outlook.com.

Acknowledgments: The authors would like to express their thanks for the infinite grace of the Almighty God of essential importance. I solemnly offer my regards to His grace, which enabled peace and harmony for this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Bit Permutation Layer

P_{ij} =

```
[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
```

Figure A1. 32 by 32 one-to-one permutation matrix.

P_{ij}^{-1} =

```
[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0;
0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0;
0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0;
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0;
0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,1,0,0 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0;
0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
```

Figure A2. Inverse of 32 by 32 one-to-one permutation matrix.

Appendix B. Test Vectors

Table A1. Test Vector (for 80-bit key).

Plaintext	Key	Ciphertext
0000 0000 0000 0000	0000 0000 0000 0000 0000	740434f796cff821
	FFFF FFFF FFFF FFFF FFFF	475a92fa61af749c

Table A2. Test vector (for 128-bit key).

Plaintext	Key	Ciphertext
0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000	858f96a55cc4f107
	FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF	0937f4ef5c91acfa

Appendix C

Proof of Theorem 1. In cryptanalysis, the Matsui's piling-up lemma is a principle used in linear cryptanalysis to construct linear relations to the action of block ciphers:

Lemma (Pilling-up lemma)

Let $\epsilon_{i_{1,i_{2,\ldots i_k}}}$ denote the bias of the random variable $X_{i_1} \oplus \ldots X_k$. Then

$$\epsilon_{i_{1,i_{2,\ldots i_k}}} = 2^{k-1} \prod_{j=1}^{k} \epsilon_{i_j}.$$

Let p be the probability of a linear characteristic. The correlation of the linear characteristic over S-box is given by $q = (2P-1)^2$ [45]. From the input–output correlation of S-box, it is straightforward that any linear characteristic over S-box has a correlation of at most $\left(2 \times \frac{4}{16} - 1\right)^2 = 2^{-2}$. The best way to resist against linear cryptanalysis is to increase the number of active S-boxes in the cipher scheme.

The maximum probabilistic bias linear characteristic of SLA S-box equal to 2^{-2} can be calculated similarly [18]. Therefore, the maximum probabilistic bias linear characteristic is estimated for three rounds as

$$2^8 \times \left(2^{-2}\right)^9 = 2^{-10}$$

When applied to sixteen rounds, the maximum probabilistic bias linear characteristic is estimated as

$$\varepsilon = 2^5 \times (2^{-10})^6 = 2^{-55}$$

To determine the hardness of linear attack, compute the number of known plaintext as follows:

$$N_L = \frac{1}{\varepsilon^2}$$

The number of known plaintext is specified for sixteen rounds of the SLA scheme as follows:

$$N_L = \frac{1}{\varepsilon^2} = \frac{1}{(2^{-55})^2} = 2^{110}$$

The available limit of known plaintext is 2^{64}. This number is lower than the desired number of known plaintext i.e., 2^{110}. So, the full number of rounds of the SLA scheme demonstrate solid resilience to linear attacks.

Appendix D

Proof of Theorem 2. The maximum probabilistic differential characteristic of SLA S-box is (2^{-2})**.** Therefore, the maximum probabilistic differential characteristic for the sixteen rounds is $P_d = (2^{-2})^{48} = 2^{-96}$. □

To determine the hardness of differential attack, compute the number of chosen plaintext as follows:

$$N_d = C/P_d$$

The number of chosen plaintext is 2^{96}, where $C = 1$ and $P_d = 2^{-96}$.

The number of chosen plaintext is 2^{96}, which exceeds the allowed bound $of\ 2^{64}$. So, excellent defense against differential attacks is seen in the complete rounds of SLA. Table 7 presents linear and differential attack comparisons.

Appendix E. Differential and Linear Relations of SLA Sbox_Nibble

Table A3. Differential relations of Sbox_Nibble.

a\b	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	2	2	2	0	0	4	2	0	0	0	2	0	0	0
2	0	2	0	0	2	2	2	0	0	2	0	0	4	0	0	2
3	0	0	2	0	2	2	2	0	4	0	0	2	0	0	2	0
4	0	4	0	2	0	2	0	0	2	0	2	2	0	0	0	2
5	0	0	4	2	0	0	2	0	0	0	0	2	2	2	0	2
6	0	0	0	2	4	2	2	2	0	0	2	0	0	2	0	0
7	0	0	0	4	2	0	0	2	0	2	0	2	0	0	2	2
8	0	0	2	0	2	0	0	0	2	2	2	0	0	2	0	4
9	0	0	0	0	0	0	2	2	2	0	2	0	2	0	4	2
a	0	2	0	2	0	0	4	0	2	2	0	0	0	2	2	0
b	0	2	2	0	0	0	2	2	0	2	4	2	0	0	0	0
c	0	0	2	2	0	4	0	0	0	2	2	0	2	0	2	0
d	0	2	2	0	0	2	0	2	0	0	0	0	0	4	2	2
e	0	0	0	0	0	2	0	2	2	4	0	2	2	2	0	0
f	0	2	0	0	2	0	0	0	0	0	2	4	2	2	2	0

Table A4. Linear relations of Sbox_Nibble.

a\b	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	−4	0	2	−2	2	2	−2	2	2	2	0	0	0	4
2	0	2	0	−2	2	4	2	0	−2	0	2	0	4	−2	0	−2
3	0	2	0	2	−4	2	0	−2	0	−2	4	2	0	2	0	2
4	0	−4	2	2	2	2	4	0	2	−2	0	0	0	0	−2	2
5	0	0	2	2	0	0	−2	−2	0	4	−2	2	4	0	−2	2
6	0	2	2	4	0	−2	2	0	0	2	2	−4	0	−2	2	0
7	0	−2	−2	0	−2	4	0	2	2	4	0	−2	0	2	2	0
8	0	−2	−2	4	−2	0	0	2	−4	−2	−2	0	2	0	0	−2
9	0	−2	−2	0	0	−2	−2	0	2	0	4	−2	2	0	−4	−2
a	0	0	2	−2	−4	0	2	2	−2	2	0	0	−2	−2	−4	0
b	0	0	−2	−2	−2	−2	4	−4	0	0	−2	−2	2	2	0	0
c	0	2	−4	2	0	2	0	−2	2	0	−2	0	−2	−4	−2	0
d	0	−2	0	−2	−2	0	−2	0	0	−2	0	−2	2	−4	2	4
e	0	0	0	0	2	2	−2	−2	−4	0	0	−4	−2	2	−2	2
f	0	−4	0	0	0	0	0	−4	−2	2	2	2	−2	−2	2	−2

References

1. Vinayaga Sundaram, B.; Ramnath, M.; Prasanth, M.; Varsha Sundaram, J. Encryption and Hash based Security in Internet of Things. In Proceedings of the ICSCN 2015, IEEE, Chennai, India, 26–28 March 2015; pp. 1–6. [CrossRef]
2. Kumar, V.K.; Mascarenhas, S.J.; Kumar, S.; Rakesh, J.P.V. Design And Implementation of Tiny Encryption Algorithm. *IJERA* **2015**, *5 Pt 2*, 94–97.
3. National Institute of Standards and Technology (NIST); Lightweight Cryptography. Available online: https://csrc.nist.gov/Projects/LightweightCryptography (accessed on 18 October 2018).
4. National Institute of Standards and Technology (NIST); Lightweight Cryptography Workshop 2015. Available online: https://www.nist.gov/newsevents/events/2015/07/lightweight-cryptographyworkshop2015 (accessed on 23 September 2016).
5. National Institute of Standards and Technology (NIST); Lightweight Cryptography Workshop 2016. Available online: https://www.nist.gov/newsevents/events/2016/10/lightweight-cryptography-workshop-2016 (accessed on 3 April 2017).
6. Al-Rahman, S.A.; Sagheer, A.; Dawood, O. NVLC: New Variant Lightweight Cryptography Algorithm for Internet of Things. In Proceedings of the AICIS 2018, IEEE, Fallujah, Iraq, 20–21 November 2018; pp. 176–181. [CrossRef]
7. Li, L.; Liu, B.; Zhou, Y.; Zou, Y. SFN: A new lightweight block cipher. In Proceedings of the MICPRO 2018, Opatija, Croatia, 21–25 May 2018; Elsevier: Amsterdam, The Netherlands, 2018; pp. 138–150. [CrossRef]
8. Alassaf, N.; Gutub, A.; Parah, S.A.; Al Ghamdi, M. Enhancing speed of SIMON: A light-weight-cryptographic algorithm for IoT applications. In *Multimedia Tools and Applications 2018*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 32633–32657. [CrossRef]
9. Patil, J.; Bansod, G.; Kant, K.S. LiCi: A new ultra-lightweight block cipher. In Proceedings of the ICEI 2017, IEEE, Pune, India, 3–5 February 2017; pp. 40–45. [CrossRef]
10. Bansod, G.; Pisharoty, N.; Patil, A. BORON: An ultra-lightweight and low power encryption design for pervasive computing. In *FITEE 2017*; Zhejiang University: Hangzhou, China; Springer: Berlin/Heidelberg, Germany, 2017; pp. 317–331. [CrossRef]
11. Banik, S.; Bogdanov, A.; Isobe, T.; Shibutani, K.; Hiwatari, H.; Akishita, T.; Regazzoni, F. Midori: A Block Cipher for Low Energy. In *ASIACRYPT 2015*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 411–436. [CrossRef]
12. Yang, G.; Zhu, B.; Suder, V.; Aagaard, M.D.; Gong, G. The Simeck Family of Lightweight Block Ciphers. In *CHES 2015*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 307–329. [CrossRef]
13. Zhang, W.; Bao, Z.; Lin, D.; Rijmen, V.; Yang, B.; Verbauwhede, I. RECTANGLE: A Bit-slice Lightweight Block Cipher Suitable for Multiple Platforms. In *Science China Information Sciences 2015*; Springer Nature: Berlin/Heidelberg, Germany, 2015; pp. 1–15. [CrossRef]
14. Biham, E. A Fast New DES Implementation in Software. In *FSE 1997*; Biham, E., Ed.; Springer: Berlin/Heidelberg, Germany, 1997; pp. 260–272. [CrossRef]
15. Anderson, R.; Biham, E.; Knudsen, L.R. Serpent: A Proposal for the Advanced Encryption Standard. *NIST AES Propos.* **1998**, *174*, 1–23.
16. Suzaki, T.; Minematsu, K.; Morioka, S.; Kobayashi, E. TWINE: A Lightweight Block Cipher for Multiple Platforms. In *SAC 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 339–354. [CrossRef]
17. Bogdanov, L.R.; Knudsen, G.; Leander, C.; Paar, A.; Poschmann, M.J.B.; Seurin, Y.; Vikkelsoe, C. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 450–466. [CrossRef]
18. Daemen, J.; Peeters, M.; Van Assche, G.; Rijmen, V. The Noekeon Block Cipher. The NESSIE Proposal, 2000. First Open NESSIE Workshop, November. Available online: http://gro.noekeon.org/ (accessed on 23 September 2016).
19. Borghof, J.; Canteaut, A.; Güneysu, T.; Kavun, E.B.; Knezevic, M.; Knudsen, L.R.; Leander, G.; Nikov, V.; Paar, C.; Rechberger, C.; et al. PRINCE—A low-latency block cipher for pervasive computing applications. In Proceedings of the ASIACRYPT 2012, Beijing, China, 2–6 December 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 208–225. [CrossRef]
20. Gong, Z.; Nikova, S.; Law, Y.W. KLEIN: A New Family of Lightweight Block Ciphers. In *RFIDSec 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–18. [CrossRef]
21. Leander, G.; Paar, C.; Poschmann, A.; Schramm, K. New Lightweight DES Variants. In *FSE 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 196–210. [CrossRef]
22. Lim, C.H.; Korkishko, T. mCrypton—A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In *WISA 2005*; Song, J., Kwon, T., Yung, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 243–258. [CrossRef]
23. Lim, C.H. A revised version of crypton: Crypton v1.0. In *FSE 1999*; Knudsen, L.R., Ed.; Spinger: Berlin/Heidelberg, Germany, 1999; pp. 31–45. [CrossRef]
24. Menezes, A.; van Oorschot, P.C.; Vanstone, S. *The Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996. [CrossRef]
25. Matsui, M. Linear Cryptanalysis Method for DES Cipher. In *EUROCRYPT 1993*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 386–397. [CrossRef]
26. Heys, H.M. A tutorial on linear and differential cryptanalysis. *Cryptologia* **2001**, *26*, 189–221. [CrossRef]
27. Biham, E.; Shamir, A. Differential Cryptanalysis of DES-like Cryptosystems. *J. Cryptol.* **1991**, *4*, 3–72. [CrossRef]
28. Nyberg, K. On the construction of highly nonlinear permutations. In *EUROCRYPT 1992*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 92–98. [CrossRef]
29. Pieprzyk, J.; Finkelstein, G. Toward effective nonlinear cryptosystem design. *IEE Proc. E-Comput. Digit. Tech.* **1988**, *135*, 325–335. [CrossRef]

30. Siegenthaler, T. Decrypting a class of stream ciphers using ciphertext only. *IEEE Trans. Comput.* **1985**, *34*, 81–85. [CrossRef]
31. Pommerening, K. *Linearitatsmafie fiir BOOLE sche Abbildungen*; Technical Report 2005; Fachbereich Mathematik der Johannes-Gutenberg-Universität: Mainz, Germany, 2005.
32. Courtois, N. Fast algebraic attacks on stream ciphers with linear feedback. In *CRYPTO 2003*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 176–194. [CrossRef]
33. Courtois, N.; Meier, W. Algebraic attacks on stream ciphers with linear feedback. In *EUROCRYPT 2003*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 345–359. [CrossRef]
34. Faugére, J.-C.; Ars, G. An Algebraic Cryptanalysis of Nonlinear Filter Generators Using Grobner Bases. Technical Report 2003. INRIA 4739. Available online: https://hal.inria.fr/inria-00071848 (accessed on 1 February 2023).
35. Zhang, X.-M.; Zheng, Y. GAC—The criterion for global avalanche characteristics of cryptographic functions. *J. Univers. Comput. Sci.* **1995**, *1*, 320–337. [CrossRef]
36. Preneel, B.; Van Leekwijck, W.; Van Linden, L.; Govaerts, R.; Vandewalle, J. Propagation Characteristics of Boolean Functions. In *EUROCRYPT 1990*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 161–173.
37. Webster, A.F.; Tavares, S.E. On the Design of S-boxes. In *Crypto '85*; Williams, H.C., Ed.; Springer: Berlin/Heidelberg, Germany, 1986; pp. 523–534. [CrossRef]
38. Chabaud, F.; Vaudenay, S. Links between differential and linear cryptanalysis. In *Advances in Cryptology (EUROCRYPT 1995)*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 356–365. [CrossRef]
39. Shi, Z.; Lee, R.B. Bit permutation instructions for accelerating software cryptography. In *ASAP'00 2000*; IEEE: Boston, MA, USA, 2000; pp. 138–148. [CrossRef]
40. Álvarez-Cubero, J.A.; Zufiria, P.J. Algorithm 959: VBF: A Library of C++ Classes for Vector Boolean Functions in Cryptography. *ACM Trans. Math. Softw.* **2016**, *42*, 1–22. [CrossRef]
41. Bansod, G.; Patil, A.; Sutar, S.; Pisharoty, N. ANU: An ultra-lightweight cipher design for security in IoT. *Secur. Commun. Netw.* **2016**, *9*, 4823–6411. [CrossRef]
42. Kumar, M.; Pal, S.K.; Panigrahi, A. FeW: A lightweight block cipher. MATDER 2019, Turk. *J. Math. Comput. Sci.* **2019**, *11*, 58–73.
43. Wu, W.; Zhang, L. L-block: A lightweight block cipher. In *ACNS 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 327–344. [CrossRef]
44. Shibutani, K.; Isobe, T.; Hiwatari, H.; Mitsuda, A.; Akishita, T.; Shirai, T. Piccolo: An Ultra-Lightweight Block cipher. In *CHES 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 342–357. [CrossRef]
45. Matsui, M. New structure of block ciphers with provable security against differential and linear cryptanalysis. In *FSE 1996*; Gollmann, D., Ed.; Springer: Berlin/Heidelberg, Germany, 1996; pp. 205–218. [CrossRef]