Feistel Network Assisted Dynamic Keying based SPN Lightweight Encryption for IoT Security

Krishna Priya Gurumanapalli[1]
Research Scholar, Department of Computer Science and Technology
Sri Krishnadevaraya University
Anantapuramu, India

Nagendra Muthuluru[2]
Professor, Department of Computer Science and Technology
Sri Krishnadevaraya University
Anantapuramu, India

Abstract—**In the last few years Internet-of-Things (IoT) technology has emerged significantly to serve varied purposes including healthcare, surveillance and control, business communication, civic administration and even varied financial activities. Despite of such broadened applications, being distributed, wireless based systems, IoTs are often considered vulnerable towards intrusion or malicious attacks, where exploiting the benefits of loosely connected peers, the attackers intend to gain device access or data access un authentically. However, being resource constrained in nature while demanding time-efficient computation, the majority of the classical cryptosystems are either computationally exhaustive or limited to avoid attacks like Brute-Force, Smart Card Loss Attack, Impersonation, Linear and Differential attacks, etc. The assumptions hypothesizing that increasing key-size with higher encryption round can achieve augmented security often fail in IoT due to increased complexity, overhead and eventual resource exhaustion. Considering it as limitation, in this paper we proposed a state-of-art Generalized Feistel Network assisted Shannon-Conditioned and Dynamic Keying based SSPN (GFS-SSPN) Lightweight Encryption System for IoT Security. Unlike classical cryptosystems or even substitution and permutation (SPN) based methods, we designed Shannon-criteria bounded SPN model with Generalized Feistel Network (SPN-GFS) model that employs 64-bit dynamic key with five rounds of encryption to enable highly attack-resilient IoT security. The proposed model was designed in such manner that it could be suitable towards both data-level security as well as device level access-credential security to enable a "Fit-To-All" security solution for IoTs. Simulation results revealed that the proposed GFS-SSPN model exhibits very small encryption time with optimal NPCR and UACI. Additionally, correlation output too was found encouragingly fair, indicating higher attack-resilience.**

Keywords—Internet-of-Things; dynamic programming; lightweight encryption; generalized Feistel network; substitution and permutation network

I. Introduction

The exponential rise in advanced software computing and low-cost hardware has broadened the horizon for industries to provide more-efficient and productive systems, serving healthcare, monitoring and surveillance, home-automation, smart-city planning and management, business communication, smart-factory, etc. Amongst the major such innovations the recently evolved IoT technology has gained wide-spread momentum to serve major aforesaid application environment. With low-cost, decentralized communication and control ability, IoT has become one of the most sought-after technology to meet the major aspects of human activities and expectations [1]. Contemporarily, it not only enables optimal decision making, productivity but also facilitates luxurious life signifying accomplished human objectives in modern era. In general, the inception of IoT can be hypothesized to be with machine-to-machine (M2M) communication systems that later evolved as internet enabled query driven systems or services [1]. Architecturally, IoT systems comprise multiple distributed (autonomous) nodes and a sophisticated software computing enabled data-processing unit, where the first collects the real-world data, while the later executes query driven tasks or instructions to meet expected demands [1]. Interestingly, in early days of evolution, though no sophisticated communication platforms were defined; however, the contemporary IoT eco-systems operate over certain wireless channel with unavoidable network dynamism and uncertainty [2]. Undeniably, in the era of IoT-ecosystem(s), the interaction between human and machines is more frequent and increase with high-pace. This is because the machines getting smarter and capable to perform many human tasks on the basis of certain targeted instructions provided. This as a result has been broadening the horizon of IoT systems across industries as well as socio-scientific periphery [1][2]. Putting a glance around, one can find that IoT has been applied in varied critical purposes as well including healthcare, finance, manufacturing, research, surveillance etc.; however, being operated over certain unpredictable network condition its reliability and data-infra-procedural (DIP) safety always remains questionable. Ironically, with increase in high-pace IoT demands, the events of security breaches too have increased severely. The security breaches in IoT can be at the different level, i.e., device level, infrastructure level, network level (say, channel) and data level [3][4][6][38]. Such security breaches can cause detrimental impacts in terms of financial loss, data-loss, tack-manipulation and procedural destruction [2]. Consequently, such IoT-security breaches can have adverse impact on major aspects of human life and its decision systems [5].

Recalling the typical architecture of the IoT systems that encompasses Low-power Lossy (sensor) Networks (LLNs), gateways, transmission channel and data-warehouses or storage possessing undeniably security related vulnerabilities. Typically, the poorly interfaced nature of IoT devices and their inferior connection, especially operating online can undergo security breaches and resulting losses [5-7]. On the other hand, attack probability gets increased manifolds over channel where different man-in-the middle attack (MITM), linear and differential attacks, impersonation, etc. are more common. Similarly, once compromising the security access level of IoT devices due to poorly configured setup or keys, an intruder can manipulate the device and its functioning, thus obstructing overall performance [7]. However, looking into depth it can be found the key reason behind such attack possibility is poor encryption systems with limited keying capacity and inferior cipher-space [8]. Such limitations are common in IoT systems where to meet computational efficiency demands; authors often apply lower-sized keys with smaller round of encryption. In fact, the threat of higher computation due to large key size and more complex confusion or cipher randomness limit efficacy of such systems [8]. It indicates the need of a robust (say, attack-resilient) and lightweight encryption system for IoTs [5][6]. An interesting fact that majority of the classical security systems employ different encryption modalities to perform device (access-level) security and data security; however, such systems under real-time (hardware) realization imposes significantly large computational overheads and resource (power and memory) exhaustion. Practically, the methods addressing IoT-network security or access-level security are different than the one used for data-security. Approaches pertaining to the data security predominantly consider attack-resilience, irrespective of the cost of computation (over large data size) and allied hardware realization [9]. On the contrary, the security measures designed towards access-control focus on retaining large key-size and encryption rounds to enable attack-resilience [10-12]. In sync with above inferences, to cope up with contemporary or even NextGen IoT ecosystems, designing a robust and "Fit-To-All (FTA) lightweight encryption system is inevitable [12]. Such security models are required to retain higher attack-resilience even at reduced computational cost, time, and hardware utilization.

Considering above the potential approaches available, despite of the different cryptosystems or crypto-algorithms like AES [13], RSA [14], DES [18], Huffman Coding, Homomorphic encryption etc. IoT demands more effective and more specifically lightweight solution [12][15-17][19]. Majority of these encryption methods focus on increasing key-size and corresponding encryption rounds to retain higher level of attack-resilience; however, fail in hardware realization due to significantly large computational overheads, memory and power exhaustion [21]. Similarly, a few researches suggested to use multi-factor authentication (MFA) by strategically amalgamating multiple cryptosystems to achieve IoT-security [1][11][13][14][17][19][32]. Such approaches often fail when delivering resource efficient and computationally efficient performance, especially with hardware realization [11][20]. Unlike classical cryptosystems, in the last few years lightweight encryption systems [12][13][16-18][33] have emerged as a potential solution for IoT security. The ability to retain higher level of confusion in cipher even at the significantly lower computation enables lightweight encryption system a viable solution towards IoT-security. Amongst the major approaches towards lightweight encryption based IoT-security, SPN-based algorithms are well-known [15][21]; however, retaining higher level of confusion even at reduced encryption cycle and relatively lower key-size has remained a challenge [8][22]. Most of the existing SPN encryption models employs 64, 128 and even larger size keys which operate over tens of encryption rounds to achieve higher cipher-confusion [23-28]. Unfortunately, these all approaches turned inferior due to higher demands of gate-elements (GE), power consumption and computational time [21][33][37]. Moreover, such methods mainly focused on retaining higher level of confusion (in cipher), irrespective of the fact that doing arbitrarily it may impact bit error rate or correlation performance. This as a result can impact different IoT-enabled services such as healthcare sector, device access control, etc. which are highly sensitive to the single-bit and /or pixel changes. In such circumstance, developing a robust lightweight encryption system with conditional confusion (say, limit constrained confusion in SPN) with lower key-size and minimum encryption rounds can be of great significance [29][30]. Undeniably, SPN based lightweight encryption methods have gained wide-spread attention towards IoT security; however, have always been criticized for limited S-box generation [29][34]. Moreover, their suitability with single key assisted encryption [30] has also been under suspicious lens. It indicates the need of dynamic keying concept with improved S-box function so as to yield better cipher at the end. Furthermore, the efficacy of Feistel Network towards SPN realization [31] too has broadened the horizon for further improvement in lightweight encryption based IoT systems. The hypothesis that "the strategic implementation of SPN with improved FN architecture [31][37] can enable improved lightweight encryption system for IoT" can be considered as the key driving force behind this research.

In sync with above stated problem and allied scopes, in this research paper a state-of-art new and robust Generalized Feistel Network assisted Shannon-Conditioned and Dynamic Keying based SPN Lightweight Encryption System for IoT Security. Our proposed lightweight encryption model, GFS-SSPN intends to improve attack-resilience or cipher generation while retaining lower key size and even significantly small encryption rounds. Architecturally, GFS-SSPN model employed SPN with Generalized Feistel Network (GFN) to perform block-cipher based encryption for IoT device security. Here, being a block-cipher based encryption, our proposed GFS-SSPN model considered 64 bits key size with merely five encryption rounds to perform data encryption. The overall proposed model encompassed dynamic programming based "Dynamic Key Generation and Expansion System", 64-bit block-cipher encryption and decryption. The proposed GFS-SSPN was designed in such manner that it retained higher level of confusion even at the reduced key-size and encryption rounds that enable to resilient to the major IoT attacks like Smart Card Loss Attack, Brute-Force Attack, Impersonation Attack, Linear and Differential attack etc. Unlike classical approaches the use of Shannon criteria for SPN-GFS encryption enabled quality-preserving encryption thus helping GFS-SSPN to have sufficient correlation and possibly minimum bit-error rate at the decryption. It enables it to be used for broad IoT applications including access-control, sensitive data logging and actuation control, telemedicine etc. [38]. The performance assessment in terms of NPCR, UACI, correlation and execution time exhibited that the proposed

GFS-SSPN model exhibits optimal performance even with significantly small key-size and encryption round. Additionally, the use of GFS enabled reduction in decryption programming and hence can be visualized as reduced hardware utilization or allied power exhaustion.

This research paper is organized as follows: Problem formulation is provided in Section II, which is followed by the system model in Section III. The results and discussions are presented in Section IV and conclusion and future scope of this research are presented in Section V.

II. Problem Formulation

Most of the classical security systems hypothesize that increasing key-size, encryption rounds and eventual cipher-randomness can help achieving optimal security; however, fail in addressing the resulting complexity and resource (i.e., power, memory, etc.) exhaustion. Additionally, these approaches undergo significantly large delay as well thus making it unsuitable towards realistic IoT-ecosystem. Undeniably, existing IoT-security systems employed cryptosystem(s) whether as standalone architecture or multi-factor authentication [35]. It eventually imposes computational overheads and complexity, thus making it inappropriate for IoT systems. To alleviate such issues, developing a lightweight encryption system has always been the dominant solution. However, retaining optimal key-size, number of rounds etc. has always remained challenge for industries, as any inappropriate design could cause loosely-coupled systems or channel inviting attacker to intrude it. Despite of the fact that the majority of the lightweight encryption methods developed so far are mainly focused towards multimedia data security in IoT systems [6]. On the contrary, a complete IoT ecosystem needs security system to ensure device-level security as well as data-level (within channel) security. To cope up with such demands, designing a “Fit-To-All” solution is must. Considering it as objective, in this paper a state-of-art new and robust Lightweight encryption-based block-cipher technique is proposed for IoT-systems. In GFS-SSPN the key goal is focused on retaining a lightweight encryption solution with smaller key-size, very small encryption and allied computational cost. Moreover, realizing the fact that IoT-systems encompass varied applications including access-control, data-logging, query-based data-retrieval, financial transactions, telemedicine information exchange etc., we focused on retaining optimal data quality along with uncompromising security. This as a result can enable a Fit-To-All solution for both access-control security as well as IoT-data security.

Literatures reveal that SPN based lightweight encryption is more hardware-friendly and computationally efficient towards IoT-security system; however, majority of the related work have applied large key-size with single key-based encryption. Additionally, such approaches have considered higher number of encryption-rounds to introduce encryption. Despite of the lightweight computation ability such approaches are often criticised for higher computational cost and delay that confines its suitability for IoT systems. It indicates the need of a solution employing lower key size with minimum encryption round of computation. Moreover, reduction in programming cost and allied overhead can help reducing memory consumption or allied gate-element exhaustion. Towards data-quality centric IoT systems, maintaining optimal cipher quality with minimum possible error too is equally significant. Considering these facts as scope or motivation, in this research we designed a novel Generalized Feistel Network (GFN) assisted Shannon-Conditioned and Dynamic Keying based SPN (SSPN) lightweight encryption system for IoT security. As the name indicates, GFS-SSPN model encompasses Shannon-Conditioned SPN encryption system which is employed over GFS encryption structure for plaintext encryption. Architecturally, GFS-SSPN employs 64-bit key which is executed over merely five rounds of encryption to generate the cipher results. Unlike major classical SPN models where authors mainly focus on introducing randomness, without considering its detrimental impact on the cipher generated and eventual decryption results, our proposed SSPN model is designed in reference to the Shannon-condition. Here, Shannon condition states that for an arbitrarily selected input, if one flips the i th bit, then likelihood that the j -th output would be changed must be one half has been taken into consideration, which is also called Strict Avalanche Condition for SPN. This approach can enable sufficiently large confusion in cipher, while maintaining higher association between input plain text and the cipher generated and hence would be advantageous towards error-free decryption. The proposed SSPN model has been applied with GFS architecture that enables its implementation as block-cipher model by partitioning input data into multiple chunks. This method not only helps in enhancing computational efficiency (i.e., time) but also reduces the separate program demands for decryption. It can be vital towards resource efficient and delay-resilient encryption for IoT. Recalling the overall architecture, SSPN model can enable higher confusion with optimal randomness and resource as well as delay efficiency while preserving data-quality. Here, SSPN helps achieving optimal S-box and P-box generation, while GFS would enable dynamic key management of keying. In the proposed model, SPN block cipher has been applied in iterative and alternating rounds of substitution and permutation (say, transposition) while ensuring that it fulfils the demands of Shannon’s Confusion and Diffusion characteristics to ensure higher attack-resilience. To achieve it, a novel dynamic key management has been developed to assure that the cipher has been processed in pseudo random manner. To introduce higher level of confusion and security-structure the proposed key-expansion block is implemented over five rounds, where in each round it intends to meet above stated Shannon’s Confusion and Diffusion (SCD) conditions. Cumulatively, these approaches can be robust enough to retrieve attack-resilient encryption even at the reduced cost and time. Here, to introduce a non-linear layer for better lightweight encryption, the proposed model applied S-Box. The proposed S-box has been applied in such way that it enables optimal diffusion over each round and thus making it more imperceptible and hence higher attack-resilient. Being a block-cipher concept, it splits 64-bit data into four chunks and performs five rounds of key generation and encryption that introduces sufficiently large randomness in cipher to avoid any attacks including SCLA, Brute-force, impersonation, linear and differential attacks etc.

III. System Model

As discussed in the previous section, our proposed lightweight encryption-based block-cipher model amalgamated SSPN and GFS optimistically designed to cope-up with high parallelism so as to achieve better time-efficiency. In the proposed model, complete input block is split into four equally partitioned 16-bit blocks. Noticeably, unlike classical SPN based approaches, we improved the overall structure by introducing numerous enhancements such as Shannon-Conditioned SPN, GFN, Dynamic Keying, etc. To enable our proposed GFS-SSPN model for swift computing IoT security system, unlike classical Feistel Network (CFS) which partitions the input into two-equal blocks, we applied GFS which split it into multiple equal blocks for further encryption. This mechanism at first helped SSPN to achieve time as well as computationally efficient parallelized encryption. Moreover, the use of GFS helps reducing additional programming cost that makes it hardware and power efficient when realized in real-world. Before discussing the proposed lightweight encryption model, a snippet of the proposed SPN model, in sync with Shannon criteria is given as follows:

A. Shannon's Confusion and Diffusion Theory

Being an SPN based encryption the proposed GFS-SSPN model applied two consecutive methods, confusion and diffusion for encryption. A snippet of these mechanisms is given as follows:

1) Confusion: Typically, in encryption system domain, confusion mechanism states that "each binary bit of the cipher must rely on varied other fractions of the key, representing obscured connection between the two (i.e., cipher and key). Functionally, it hides the associations in between the plaintext and the key applied, and consequently avoids any possible security breaches due to key-loss, such as SCLA or linear and differential attacks. Confusion makes it difficult for attacker to extract key and hence can alleviate any unauthorized retrieval of the original data from cipher. Functionally, for an efficient encryption model with single bit change or manipulation entire cipher bits must be changed. This approach enhances the level of ambiguity of the cipher and hence helps in retaining higher attack-resilience.

2) Diffusion: Unlike confusion, diffusion states that in case a single bit is manipulated in input plaintext, then nearly half of the bits in the cipher might be statistically changed. Similarly, if a single bit is manipulated in the cipher then the half of the plaintext bit would change. Since, a single bit may have only two-states, approximately half of the bit can have their state manipulated. In our proposed GFS-SSPN encryption scheme, the predominant concept behind diffusion is to hide the relationship between the input plaintext and the corresponding cipher which make it difficult for attacker(s) to get access of the plaintext. Diffusion is further accomplished because of the increased level of randomness in the plaintext over the different rows and columns. Functionally, it is accomplished by performing transposition. A snippet of the Shannon Condition of SPN, being applied in the proposed GFS-SSPN lightweight encryption model is given as follows:

Typically, Shannon condition for SPN encryption defines confusion as the process for transforming the associations or the relationship between the key and the cipher as complex as possible. It enables depletion or reduction in the native statistical structure of the input plaintext over cipher. In our proposed SSPN model, such complexity is introduced by perform repetitive substitutions and permutations function, where in substitution, a part of bits in each block is replaced or substituted with other parts following the Shannon criteria of substitution. Permutation on the other hand indicates the mechanism of changing the bit's order as per certain mathematical approach, also called transformation rules. In GFS-SSPN model to achieve high non-linearity, the input bits are distributed across the structure of the cipher text which makes it difficult to detect either key or the plaintext data. In sync with the Shannon condition for SPN, for any randomly selected input, if one changes the ith bit, probability that the j-th output would be changed remains the half. This condition is also known as the Strict Avalanche Condition. It states that it is important to assure that flipping of a definite set of bits must change each output bit in cipher with the probability of minimum one-half. A vital purpose of the use of confusion was to inculcate such randomness that an attacker to identify the key, despite the fact that the one has the significantly large plaintext-cipher pairs generated with the same key. Therefore, each bit of the cipher depends on the entire key in the different ways on the varied bits of the key. In other words, manipulating a single bit can change the entire cipher text. Thus, implementing the Shannon conditioned SPN (say, SSPN) we developed a state of art new Dynamic Keying and Expansion (DKE) mechanism over with GFS over five consecutive rounds. Here, we designed DKE in such way that it ensures the Shannon condition where the cipher gets manipulated in pseudo random manner. The proposed DKE assisted encryption with GFS-SSPN model was applied over five consecutive rounds so as to induce higher confusion and hence attack-resilience, where in each round of computation if follows the Shannon's criteria of confusion and diffusion. The detailed discussion of the overall system implementation is given in the sub-sequent sections.

B. System Implementation

For system implementation, to maintain lower computational cost and allied (possible) hardware resource exhaustion, we designed GFS-SSPN as a 64-bit block cipher model. In this approach we considered 64-bit key to perform encryption of each block encompassing 64-bit input plaintext. Additionally, to perform GFS-SSPN encryption we applied only five round of encryption. In this process, each encryption-round is performed over DKE function which helps in generating confusion and diffusion matrix over different rounds of encryption. Though, maintaining higher encryption round can introduce higher randomness; however, at the cost of increased computational overhead, delay and resource exhaustion. Therefore, to design the proposed encryption model suitable for IoT-applications, we maintained only five round of encryption, while to introduce higher cipher randomness and minimum correlation we introduced a state-of-art new Dynamic Keying and (key) Expansion concept in which each split of input 16-bit was processed for GFS-SSPN over five rounds, where each round gave rise to a new set of keys for encryption. Thus, performing keying expansion over each round of computation finally we obtained a concatenated 64-bit key to be used for decryption. Architecturally, in GFS-SSPN, at first the input of 64-bit was split into four equally chunks, which was possible by the use of GFS architecture. In other words, by applying the proposed GFS architecture as shown in Fig. 1, we applied SSPN over each chunk of 16-bits. To further enhance randomness in cipher, we split each 16-bit data into four distinct sub-parts each of 4-bits size. To further introduce higher confusion and diffusion over consecutive rounds of encryption, the proposed GFS model applied logical XOR function over each round of confusion and diffusion to result cipher results per block. To achieve the overall implementation our proposed model applies three key functions.

1) Multi-round Dynamic Keying, Expansion and Update.
2) Block-Cipher Encryption and Cipher Generation, and
3) Block-Cipher Decryption.

The detailed discussion of each functional component is given in the subsequent sections.

1) Multi-round dynamic keying, expansion and update: In sync with the real-world IoT systems where each node can be assumed to act like a key generator as well as decoder, it becomes vital to minimize cost caused by key generation and update. To enable lightweight encryption, in GFS-SSPN model different mathematical models by using varied logical functions like XOR and XNOR has been developed. Here, our proposed DKE model serves dual purpose; first it acts as a key generation and expansion (KGE) unit while secondly it helps concatenating cipher iteratively. To achieve key generation and expansion, GFS-SSPN applied GFS assisted encryption. In GFS-SSPN, GFS is applied over five encryption rounds, where each round employs distinct key to perform substitution and permutation tasks. Realizing the fact that the proposed model intends to apply minimum (here, only five) round of encryption which is undeniably lower than the classical AES, RSA, ECC, Diffie-Hellman, etc. kind of encryption systems or even many existing SPN based encryption model, our proposed model intended to introduce a concept of dynamic keying. Unlike classical SPN methods or other cryptosystems that apply merely single key to perform encryption, GFS-SSPN to introduce higher randomness or confusion in cipher, we applied dynamic keying concept. In the GFS-SSPN, we estimated a set of keys over each round and thus, we retrieved a final concatenated key of 64-bits. Noticeably, to retain higher attack-resilience a lightweight encryption model requires maintaining sufficiently large key size k_t which can prohibit any intruder to perform $2^{k_t - 1}$ encryption so as to gain key information for data access or retrieval. Our proposed model applied 64-bit key to perform bit-permutation instruction-based encryption for the 64-bit block sized input data. We applied 64-bit's cipher key k_c as the input of the proposed DKE function to perform a SSPN confusion and diffusion task, as per Fig. 2. In this manner, it provides five distinct keys one for each round of encryption.

In DKE (Fig. 1) we implement the mathematical approach suggested by Barreto et al. [68]. The Khazad cipher model as proposed in [68] applied Broad-Trial-Mechanism (BTM) to perform multiple linear and non-linear transformation to perform encryption. On the contrary, we applied GFS architecture which enabled a definite relationship and inter-dependency between the input bits and the output cipher in a predefined complex manner. As depicted in Fig. 1 the 64-bit input k_c is equally-split into four equal-blocks, each of 16-bits. Moreover, to perform SSPM, we split each 16-bit block into 4-bit chunks, which were later processed for substitution and permutation as depicted in Fig. 2. Thus, performing DKE over each 16-bit input, our proposed model obtained a matrix of 4-bits each. Noticeably, each block of 16-bits generated four 4-bits chunks to be processed for further GFS enabled SSPN. The detail of the proposed SSPN model is given in the subsequent section.

a) Shannon Constrained SPN (SSPN)

As depicted in Fig. 1, once estimating the 16-bits data post DKE, we performed the initial substitution of the chunks of k_c (1).

$$Km_{i \in 1,2,3,4} f = ||_{j=1}^{4} Kn_{4(j-1)+i} \tag{1}$$

The above derived model (1), enabled estimation of a 16-bits output for each DKE block, applied over all four 16-bit blocks. Once estimating the 4×4 matrix from each 16-bit input (i.e., post DKE), we performed circular shifting which helped us to generate four different keys.

Now, estimating all four keys for $Km_{i \in 1,2,3,4} f$ as depicted in Fig. 1, we concatenated these keys and retrieved a set of keys $Kc_i f$ for round of computation using (2).

$$Kc_i f = f\ (Km_i f) \tag{2}$$

To further introduce higher confusion and diffusion over the consecutive SSPN layers, we applied a state-of-art new Transformation coder containing two different functions, say, “Linear (LF)-Non-Linear Function (NF)”, as depicted in Fig. 2.

To enable a computationally efficient SSPN implementation, we applied a predefined transformation coder, encompassing two distinct functions, LF and NLF, as given in Table I. Though, in GFS-SSPN k, we applied LF and NLF transformation coder as predefined value; however, different other transformational values can also be assigned for the different test cases or custom encryption demands.

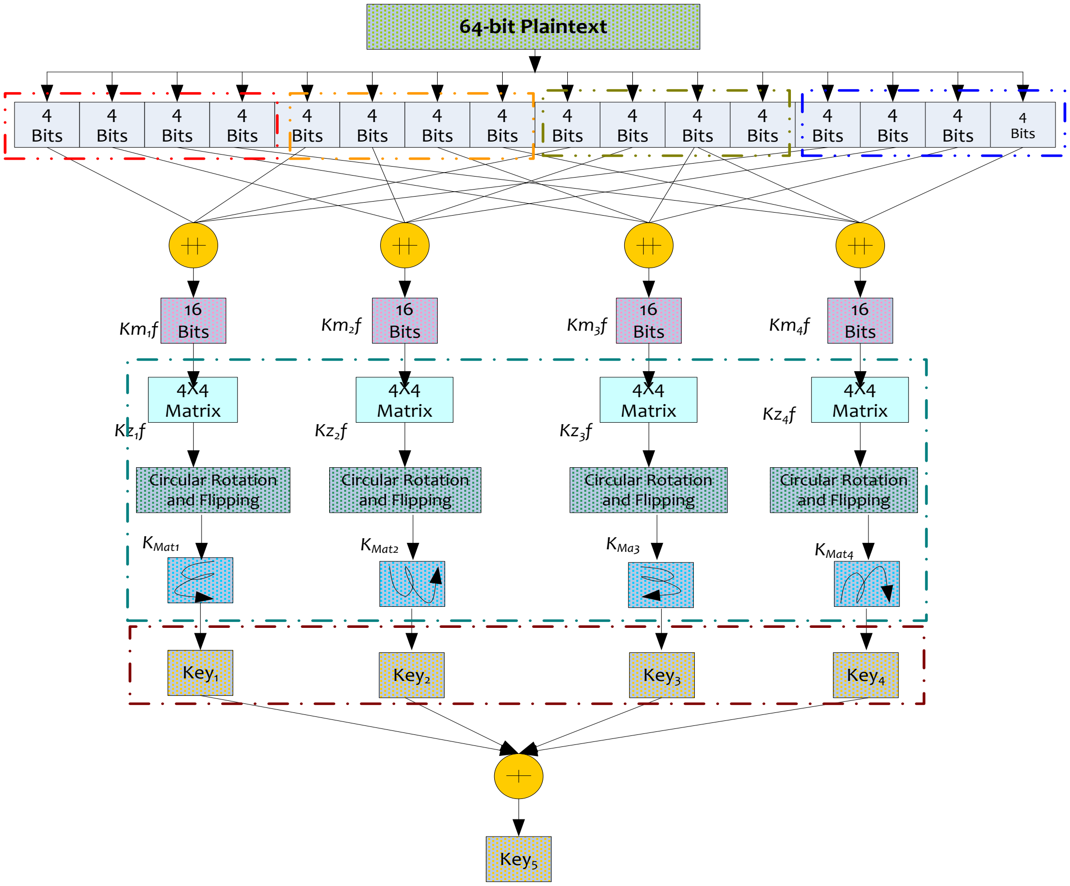

Fig. 1. Proposed Dynamic Key Generation and Expansion.

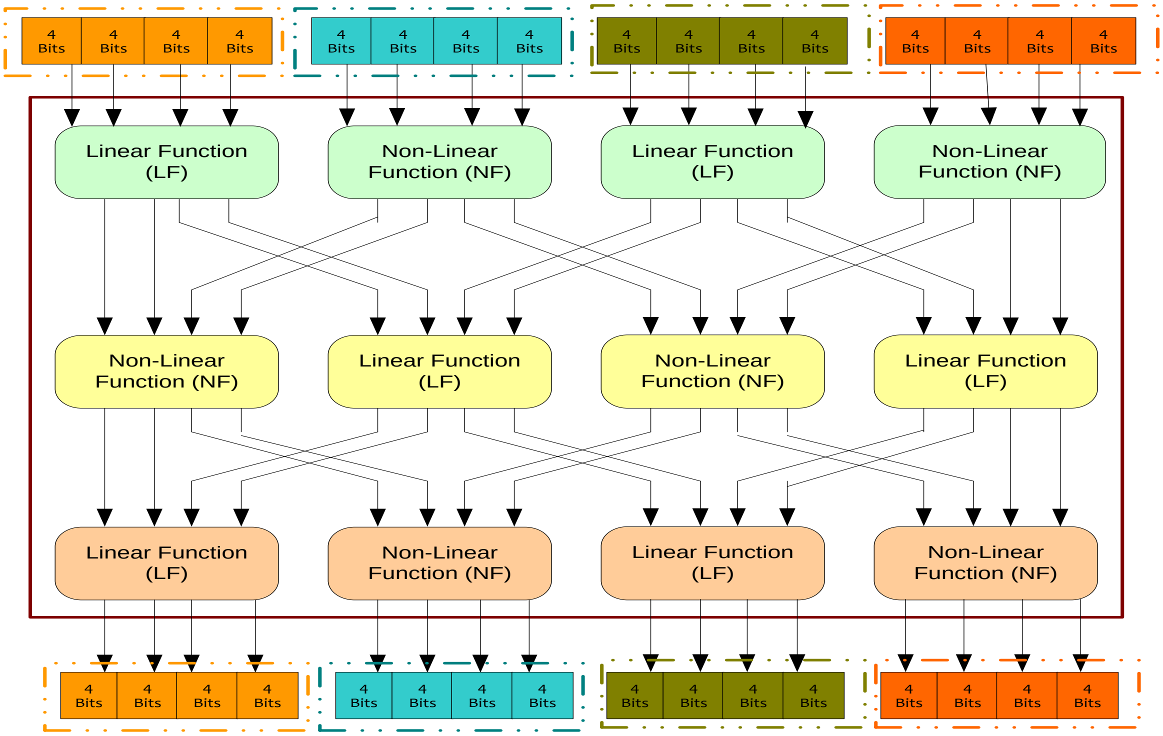

Fig. 2. SSPN Implementation.

TABLE I. TRANSFORMATION CODER

$k_{n,i\in1,2,3,4}$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$LFp(k_{n,i\in1,2}$	3	F	F	0	4	1	E	B	A	F	7	8	6	8	1	1
$NLFp(k_{n,i\in}$	9	D	4	4	F	2	3	E	D	2	6	E	7	C	2	7

Thus, with the different inputs, we applied transformation coder (Table I) values. This as a result enabled distinct keys per 16-bit blocks. Estimating the outputs from the DKE model, we obtained 4×4 matrix which was later processed for the "circular-Shifting". This mechanism re sampled the input 4×4 matrix, per 16-bit of input and eventually estimates four distinct keys, as per the equations (3-6). The matrix generated for each 16-bit block is given in equations (3-6).

$$K_{Mat1} = \begin{bmatrix} Kc_1f_1 & Kc_1f_2 & Kc_1f_3 & Kc_1f_4 \\ Kc_1f_5 & Kc_1f_6 & Kc_1f_7 & Kc_1f_8 \\ Kc_1f_8 & Kc_1f_{10} & Kc_1f_{11} & Kc_1f_{12} \\ Kc_1f_{13} & Kc_1f_{14} & Kc_1f_{15} & Kc_1f_{16} \end{bmatrix} \quad (3)$$

$$K_{Mat2} = \begin{bmatrix} Kc_2f_1 & Kc_2f_2 & Kc_2f_3 & Kc_2f_4 \\ Kc_2f_5 & Kc_2f_6 & Kc_2f_7 & Kc_2f_8 \\ Kc_2f_8 & Kc_2f_{10} & Kc_2f_{11} & Kc_2f_{12} \\ Kc_2f_{13} & Kc_2f_{14} & Kc_2f_{15} & Kc_2f_{16} \end{bmatrix} \quad (4)$$

$$K_{Mat3} = \begin{bmatrix} Kc_3f_1 & Kc_3f_2 & Kc_3f_3 & Kc_3f_4 \\ Kc_3f_5 & Kc_3f_6 & Kc_3f_7 & Kc_3f_8 \\ Kc_3f_8 & Kc_3f_{10} & Kc_3f_{11} & Kc_3f_{12} \\ Kc_3f_{13} & Kc_3f_{14} & Kc_3f_{15} & Kc_3f_{16} \end{bmatrix} \quad (5)$$

$$K_{Mat4} = \begin{bmatrix} Kc_4f_1 & Kc_4f_2 & Kc_4f_3 & Kc_4f_4 \\ Kc_4f_5 & Kc_4f_6 & Kc_4f_7 & Kc_4f_8 \\ Kc_4f_8 & Kc_4f_{10} & Kc_4f_{11} & Kc_4f_{12} \\ Kc_4f_{13} & Kc_4f_{14} & Kc_4f_{15} & Kcf_{16} \end{bmatrix} \quad (6)$$

Now, once retrieving the values of the key-matrix ($K_{Mat1}, K_{Mat2}, K_{Mat3}andK_{Mat4}$), to obtain the keys for each 16-bit block, we transformed these metrics into four distinct arrays of 16 bits as derived in (7-10). These 16-bits arrays provided the encryption key per round. Noticeably, in (7-10) the operator $\mathbin{+\!\!+}$ represents the concatenation function.

$$Key_1 = a_4 \mathbin{+\!\!+} a_3 \mathbin{+\!\!+} a_2 \mathbin{+\!\!+} a_1 \mathbin{+\!\!+} a_5 \mathbin{+\!\!+} a_6 \mathbin{+\!\!+} a_7 \mathbin{+\!\!+} a_8 \mathbin{+\!\!+} a_{12} \mathbin{+\!\!+} a_{11} \mathbin{+\!\!+} a_{10} \mathbin{+\!\!+} a_9 \mathbin{+\!\!+} a_{13} \mathbin{+\!\!+} a_{14} \mathbin{+\!\!+} a_{15} \mathbin{+\!\!+} a_{16} \quad (7)$$

$$Key_2 = b_1 \mathbin{+\!\!+} b_5 \mathbin{+\!\!+} b_9 \mathbin{+\!\!+} b_{13} \mathbin{+\!\!+} b_{14} \mathbin{+\!\!+} b_{10} \mathbin{+\!\!+} b_6 \mathbin{+\!\!+} b_2 \mathbin{+\!\!+} b_3 \mathbin{+\!\!+} b_7 \mathbin{+\!\!+} b_{11} \mathbin{+\!\!+} b_{15} \mathbin{+\!\!+} b_{16} \mathbin{+\!\!+} b_{12} \mathbin{+\!\!+} b_8 \mathbin{+\!\!+} b_4 \quad (8)$$

$$Key_3 = c_1 \mathbin{+\!\!+} c_2 \mathbin{+\!\!+} c_3 \mathbin{+\!\!+} c_{10} \mathbin{+\!\!+} c_{11} \mathbin{+\!\!+} c_{12} \mathbin{+\!\!+} c_{16} \mathbin{+\!\!+} c_{15} \mathbin{+\!\!+} c_{14} \mathbin{+\!\!+} c_{13} \quad (9)$$

$$Key_4 = d_{13} \mathbin{+\!\!+} d_9 \mathbin{+\!\!+} d_5 \mathbin{+\!\!+} d_{11} \mathbin{+\!\!+} d_7 \mathbin{+\!\!+} d_3 \mathbin{+\!\!+} d_4 \mathbin{+\!\!+} d_8 \mathbin{+\!\!+} d_{12} \mathbin{+\!\!+} d_{16} \quad (10)$$

Thus, estimating the four distinct keys; Key_1, Key_2, Key_3 and Key_4 , we performed XOR logical function as per (11) to achieve the final encryption key.

$$Key_{Fused} = Key_1 \oplus Key_2 \oplus Key_3 \oplus Key_4 \quad (11)$$

2) Lock-Cipher encryption and cipher generation: Once estimating the complete 64-bit of encryption key (11), we performed plaintext encryption. To perform encryption, we applied the schematic given in Fig. 3. As depicted in Fig. 3, the input 64-bit plaintext was equally split into four 16-bit blocks (i.e., Px_{0-15}, Px_{16-31}, Px_{32-47} and Px_{48-63}). Subsequently, executing bit-permutation instruction over each 16-bit block (each round), GFS-SSPNexhibited bit-wise swapping so to reduce traceability. To achieve it, we performed bit's order alteration, where the sub-blocks of 16-bit block were changed. Subsequently, we performed bitwise XNOR logical operation between the round key Key_i and the input plaintext Px_{0-15}. This process was repeated four rounds in K_i to Px_{48-63} and eventually the corresponding outputs Ro_{11} and Ro_{14} were obtained. Now, estimating the output from XNOR logical operator (Fig. 3) we fed the results to the DKE component which generated two distinct outputs Ef_{l1} and Ef_{r1}. Now, once estimating the values of Ef_{l1} and Ef_{r1}, we took Ef_{l1} and Px_{32-47} , and performed bitwise-XOR to estimate Ro_{12} . Similarly, the bitwise-XOR between Ef_{r1} and Px_{16-31} gave rise to Ro_{13}. We applied equation (12) to estimate encrypted cipher outputs.

$$Ro_{i,j} = \begin{cases} Px_{i,j} \odot K_i \ ; & j = 1 \ and\ 4 \\ Px_{i,j+1} \oplus Ef_{li} \ ; & j = 2 \\ Px_{i,j-1} \oplus Ef_{ri} \ ; & j = 3 \end{cases} \quad (12)$$

Thus, the above discussed transformations were performed in such way that for each consecutive round, Ro_{11} turned out to be Px_{16-31}, while Ro_{12} became Px_{0-15}, Ro_{13} as Px_{48-63}. Similarly, Ro_{13} became Px_{32-47}. This process was performed for all rounds by using (12). The outputs of the final round were concatenated as per (13) to generate the cipher output.

$$CT = R_{51} \mathbin{+\!\!+} R_{52} \mathbin{+\!\!+} R_{53} \mathbin{+\!\!+} R_{54} \quad (13)$$

3) Block-Cipher decryption: As already stated, unlike classical SPN based encryption or other cryptosystems, our proposed GFS-SSPN lightweight encryption model applied GFS assisted SSPN, and therefore, it avoids any additional programme for decryption. The use of GFS enabled decryption in the same way as performed towards encryption using the different logical functions, as discussed above. This characteristic enabled significantly lower computational overheads and resource exhaustion. Thus, implementing the above discussed methods, we performed block-cipher encryption for the different set of inputs characterizing a small size authentication credential or a large plaintext data to be communicated over the IoT-centric LLNs channels or even to be stored on local memory [70]. The detailed discussion of the overall simulation results and allied inferences is given in the subsequent sections.

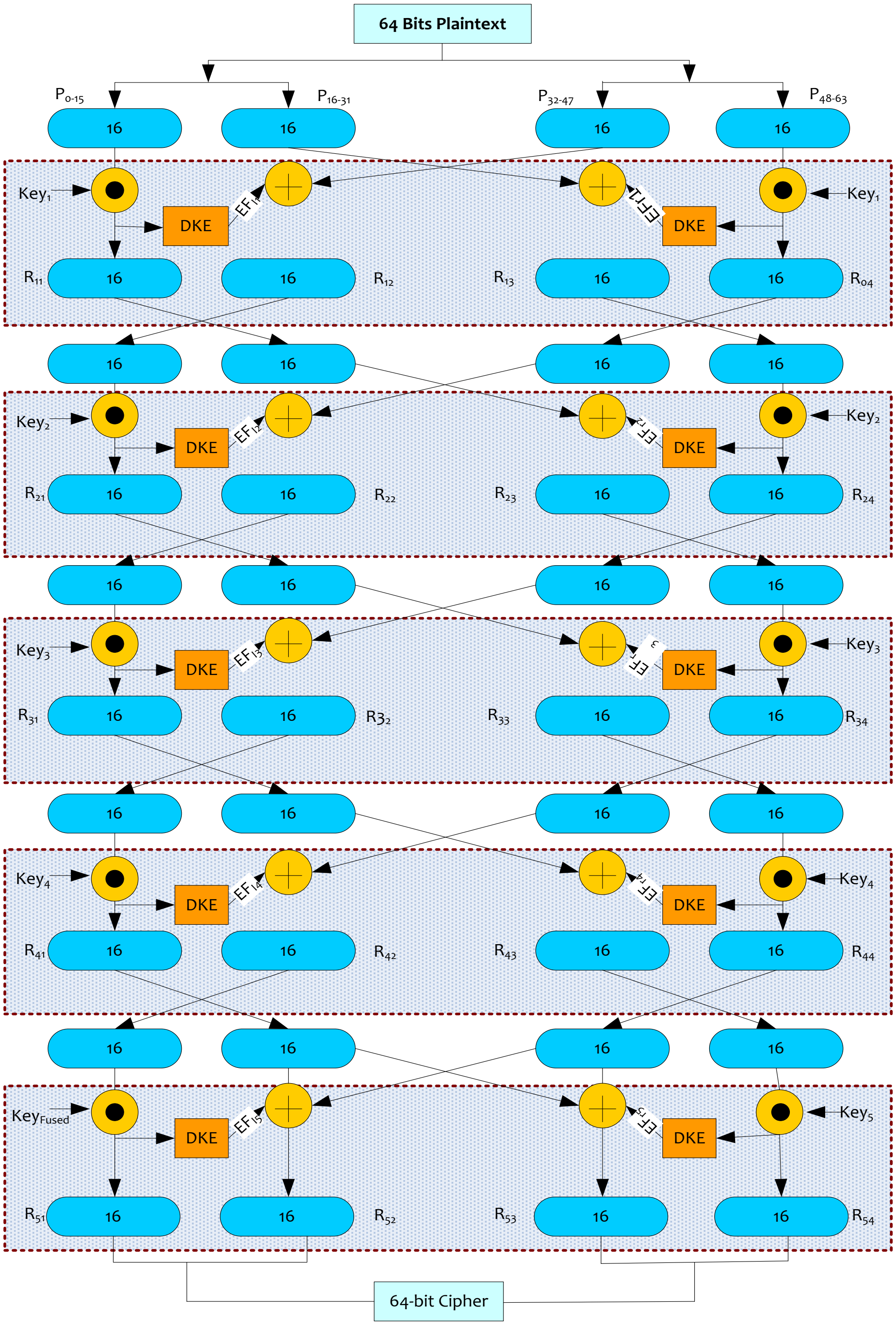

Fig. 3. Proposed Block-Cipher Encryption.

IV. Results and Discussions

As stated, and discussed in details, the predominant emphasis of this research was to contribute a state-of-art new and robust lightweight encryption system or the block-cipher technique for IoT security. Zeroing down the belief that whether a device level or the data level security in IoT, maintaining seamless and attack-resilient cipher is must, the proposed work focused on introducing sufficiently large and optimal randomness (say, confusion) in cipher. On the other hand, to retain computational efficacy and optimality, we considered block cipher method with 64-bit key and merely five round of encryption. In sync with hypotheses that maintaining or applying higher encryption rounds can enable higher randomness in cipher, unlike classical methods, we applied different state-of-art logical functions implemented over GFS assisted SSPN to perform encryption. In other words, we strategically implemented GFS with Shannon-Conditioned SPN to perform encryption using 64-bit key and only five round of encryption. Noticeably, the key purpose of using GFS was to assist block-cipher generation, while SSPN was targeted to accomplish encryption with sufficiently large confusion or randomness. Moreover, the use of GFS enabled reduction in programming cost as it avoided any separate decryption program like other classical SPN methods or cryptosystem algorithms. Noticeably, unlike conventional SPN methods, the use of Shannon-conditioned SPN not only intended to achieve optimal confusion and diffusion, but also intends to support optimal decryption at the receiver without bit-loss. This is the matter of fact that not much significant works are done towards lightweight encryption based IoT security, especially by using SPN and/or Feistel network. A few efforts employing technologies have mainly focused on multimedia data security [6]; however, the use of higher encryption key and rounds make them computationally exhaustive. On the other hand, a few methods, targeting hardware efficient lightweight encryption too are found limited and exhaustive due to bit-by-bit encryption and supplementary tool/programme-based randomness insertion (in cipher). In our research we tried to alleviate such limitations and designed GFS-SSPN in such manner that it could retain higher confusion even with minimum encryption round and lower key-size. The use of dynamic keying concept over the multiple rounds helped accomplishing an untraceable keying strategy, where SSPN helped in maintaining sufficiently large randomness as well as associations with the input plaintext. This as a result intended to achieve high attack-resilience.

Repeating the structural details, our proposed GFS-SSPN model encompassed three key components including Dynamic Key Generation, Expansion and Update (DKE), GFS-SSPN encryption and GFS-SSPN decryption. Being a bloc-cipher method, the proposed model at first split 64-bit input into four equal blocks of 16-bit each. Subsequently, applying an initial key value, with input 16-bit plaintext we perform SSPN that eventually performs logical operations and generated each round output. The outputs obtained for each 16-bit blocks were processed for the different logical operations and flipping, circular rotation etc., that eventually generated 64-bit keys (16-bit key per block, which were concatenated together to result a final 64-bit key for encryption) over five round of encryption. Thus, with the final generated key, we performed GFS-SSPN encryption followed by decryption. The overall proposed GFS-SSPN model was developed using C-programming language which was simulated over $DEVC++$ compiler. We simulated the proposed lightweight encryption system onto a central processing unit (CPU) with 8 GB RAM, and Intel-i5 processor. To assess performance of the proposed GFS-SSPNmodel, we examined its efficacy in terms of encryption time, correlation, NPCR and UACI. To be noted, as already stated IoT-ecosystem encompasses the different activities including data logging, device access, live streaming, etc., and hence for both device level security as well as data level or network level, ensuring plaintext (say, data) security is must. Retaining minimum encryption time, negligible or very small correlation between the cipher and key is must. Moreover, there are many IoT applications such as data-streaming, access-credentials, telemedicine in healthcare etc., where maintaining higher NPCR and suitable UACI is must. Considering this fact, we examined the proposed of the proposed GFS-SSPN model in terms of encryption time, correlation, NPCR and UACI.The performance assessment has been performed under two broad umbrellas; first quantitative assessment and second, the qualitative assessment. Here, quantitative assessment discusses empirical simulation outputs and allied inference, while qualitative assessment discuses different attack-resilience. The detailed discussion is given as follows:

A. Quantitative Assessment

This section primarily discusses some of the key empirical analysis outcomes in terms of correlation, encryption time, etc.

1) Correlation: Realizing the fact that to ensure high attack-resilience and imperceptibility in cipher, it is must to retain minimum correlation between cipher generated, original text and the encryption key. There are many IoT-attack conditions such as linear and differential attack analysis, impersonation etc., where the attacker often exploits cipher details to retrieve the key information and allied original data. To avoid such problems, retaining minimum or negligible correlation is must. On the other hand, correlation being the statistical dependency between the two different variables requires to be maintained as minimal as possible so as to alleviate any attack conditions, as stated above. To assess performance of the proposed GFS-SSPN model, we obtained correlation coefficient value γ between the cipher text and the original text using (1). Noticeably, for any encryption-based security system maintaining γ near zero is considered as an ideal condition.

$$\gamma_{x,y} = \frac{cov(x,y)}{\sqrt{D(x)\sqrt{D(y)}}} \tag{14}$$

In (14), the parameter $cov(x,y)$ signifies the covariance, while the variance for $x$and $y$are given as $D(x)$ and $D(y)$, respectively. Typically, the distribution of the variance of any single dimension random variable can be estimated as per (15).

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)^2 \tag{15}$$

In (15), $D(x)$states the variance for x. Now, to calculate covariance in between x and y, we used (16) as given below.

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right) \tag{16}$$

In (16), the parameters $E(x)$ and $E(y)$ signifies the targeted values for x and y, respectively. Here, the expectation values for x is obtained as per (17).

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{17}$$

In (18), N presents the total number of bits in the data, while x signifies the N-dimensional vector. Noticeably, the component x_i presents the $i-$th value of the original plaintext.To assess the performance, we simulated proposed model with the input plaintexts of different sizes (here, we call sample(s). The results obtained towards the correlation outputs between input sample and resulting cipher is given in Table II.

TABLE II. CORRELATION ANALYSIS

Input Samples	Original data	Cipher data
1	0.9919	0.0076
2	0.9971	0.0072
3	0.9829	0.0036
4	0.9801	0.0060
5	1.0000	0.0071
6	0.9672	0.0129
7	0.9721	0.0131
8	0.9408	0.0083
Average Correlation	**0.9790**	**0.0082**

The average correlation obtained over a total of 8 different samples encryption reveals that the proposed GFS-SPNN model accomplishes the minimum correlation coefficient of 0.0082. This is significantly small that it can enable seamless encryption for attack-avoidance. In above depicted results (Table III), we considered the different samples of varied sizes for encryption. Table II presents the correlation results between the original test and the cipher data obtained post GFS-SSPN. Observing the results, it can easily be inferred that post-encryption the correlation value significantly decreases. It indicates minimum correlation signifying high attack-resilience.

2) Number of Changing Pixel Rates (NPCR): In addition to the correlation performance, we examined the efficacy of the proposed lightweight encryption system in terms of other statistical performance variables such as the number of pixels changing rate (NPCR)and unified averaged changed intensity (UACI). Typically, those encryption algorithms or methods which hypothesize that merely including randomness in cipher can enable attack-resilience often under a detrimental effect called avalanche effort or plaintext sensitivity [39-41]. Though, the use of Shannon criteria of SPN alleviated such problem, however, maintaining higher changes or sensitivity over encryption is must. In this relation, we measured NPCR performance signifying the total number of characters which are different in cipher than the equivalent plain-text (say, w_1 and w_2). Mathematically, NPCR was estimated as per (18), where its value is often presented in the form of percentile (%).

$$NPCR = \frac{\sum_{i=1}^{l} W(i)}{l} \times 100 \tag{18}$$

In (18), the variable l states the length of the text to be encrypted, and

$$W(i) = \begin{cases} 0, & if\ E_1(i) = E_2(i) \\ 1, & if\ E_1(i) \neq E_2(i) \end{cases} \tag{19}$$

In majority of the lightweight encryption models, the ideal NPCR is hypothesized to be near 99.6%. In this reference our accomplished average performance with NPCR=99.63% affirms suitability towards a real-time IoT security system.

TABLE III. NPCR PERFORMANCE OVER DIFFERENT PLAINTEXT SAMPLES

Input Samples	NPCR (%)
1	99.60
2	99.483
3	99.21
4	100.00
5	100.00
6	99.63
7	99.52
8	99.62
Average NPCR	**99.63**

3) Unified Averaged Changed Intensity (UACI)L: UACI represents the difference of the mean values between the cipher texts. Typically, the 100% of UACI states that both texts are completely different in amplitude. Mathematically, we used (20) to estimate the UACI performance.

$$UACI = \frac{100}{l \times 95} \sum_{i=1}^{l} |E_1(i) - E_2(i)| \tag{20}$$

In (20), l represents the text-length, while $E_1(i)$and $E_2(i)$ signify the symbol value of the cipher texts. For a single text encryption method, (i.e., a unit plaintext is processed at once), $E_i(i)$ can be applied for UACI estimation. The UACI values for the different samples are given in Table IV.

TABLE IV. UACI RANDOMNESS ASSESSMENT

Input Samples	UACI (%)
1	35.77
2	38.32
3	43.81
4	45.00
5	44.00
6	41.06
7	44.87
8	43.31
Average UACI	**42.01**

4) Encryption time analysis: In sync with IoT systems or allied applications, maintaining minimum encryption time is must. As already discussed, IoT system often performs delay-resilient real-time communication and therefore whether to get access of system or data, or even transmits the data, achieving minimum encryption time is inevitable. Taking into consideration of this fact, we examined encryption time (in ms) performance over the different samples. Table V presents the encryption time results by the proposed GFS-SSPN lightweight encryption system. Noticeably, we used (21-23) to estimate the encryption and decryption time (ms) [35].

$$Encryption\ Time\ (ms) = \frac{\text{Time required to encrypt i−th data}}{\text{Total number of data}} \tag{21}$$

$$Decryption\ Time\ (ms) = \frac{\text{Time required to Decrypt i−th data}}{\text{Total number of data}} \tag{22}$$

$$Execution\ time = \\ Encryption\ time\ (ms) + Decryption\ time\ (ms) \tag{23}$$

TABLE V. EXECUTION TIME

Plaintext Input Samples	Execution Time (ms)
1	1.67
2	6.02
3	6.31
4	9.82
5	6.98
6	11.41
7	12.14
8	23.28
Average Exec time (ms)	**9.70**

To be noted, the sample sizes (in bits) considered in this study were varied (here, increasing order) from 1 to 8. The initial sample size we considered for sample-1 was 256 bits, while we kept increasing the size from 1 to 8 samples. In sync with increasing data volume, the encryption and decryption time too increased. However, the average time for execution (including both encryption as well as decryption) was 9.70 ms. This time is sufficiently small to accommodate real-time encryption purposes. To be noted, typically the time efficiency of an algorithm depends on computer's efficacy as well. Since, the proposed model was simulated over Intel i5 processor, execution over superior CPU configuration can yield even more affirmative results.

5) Inter-Model performance assessment: This is the matter of fact that the proposed GFS-SSPN lightweight encryption model exhibited significantly well towards delay-resilient and quality-sensitive encryption for IoT. Undeniably, the performance outcomes in terms of correlation, NPCR, UACI and execution time (ms) is encouraging; however, to validate efficacy of the proposed model towards IoT ecosystem we performed an inter-model performance assessment. In this approach we compared the performance of the proposed GFS-SSPN model with the other state-of-art existing methods. Factually, a very few researches have addressed lightweight encryption based IoT security system, and a few researches applying this method are developed for multimedia data security to be used in IoT applications [6]. Researchers [20][21] [40[41] emphasized their lightweight encryption model towards image data security over IoT platforms [6]. Furthermore, these authors have applied different random inputs to assess respective performance; and therefore, comparing performance over the same data is difficult. Considering this fact, we examined relative performance as average performance by the different algorithms.

Authors [21] developed a dynamic structure based lightweight encryption system for image encryption for IoT communication. Unlike our proposed model, authors designed a lightweight encryption concept with forward and backward chaining blocking concept (FBC) that in conjunction with a permutation block performed encryption. Though, similar to our approach authors too suggested applying multi-layered dynamic keying concept to introduce higher randomness and hence high attack-resilience. Noticeably, their proposed cipher layer comprised a binary diffusion matrix, S-box and P-Box, in sequence. Authors [15][42][43] affirmed their efficacy backed by a binary diffusion matrix, substitution and permutation table. Considering about the NPCR performance by [43], the highest NPCR obtained was 50.10, while UACI for the same algorithm was obtained as 99.64. In sync with the inference in [39][41], for a robust lightweight encryption model retaining maximum possible NPCR (near 100%) is must. Therefore, [43] fails in delivering the expected performance. Murillo-Escobar et al., [39] too designed a lightweight encryption model using Chaotic map based symmetric text-cipher generation [36]. Though, authors applied 128-bits key, the use of two logistic maps with optimized pseudorandom sequences were considered for encryption, made it too computationally exhaustive. Authors employed single round of permutation diffusion to reduce overhead of encryption. A snippet of the recent approaches and their corresponding performance is given in Table VI.

Noticeably, the work proposed by Murillo-Escobar et al. (2014) applied 128-bit key. On the contrary we employed only 64-bit key for encryption. A snippet of the different lightweight encryption models and their configurations is given in Table VII.

Table VI presents some of the key lightweight encryption systems and allied functional architectures like block size, key size and number of encryption rounds. Observing the overall results, it can be found that the proposed GFS-SPNN model employs minimum key size (here, 64) with minimal encryption rounds (here, five) and even retains optimal performance. Though, a few works like KHAZAD [68][29][52], etc. applied low encryption round; however, retained 128 as key size. Similarly, the approaches employing 64-bit of encryption key have applied higher encryption round to retain higher confusion or randomness in cipher. The comparative assessment also reveals (Table VI) that most of the existing lightweight encryption models for IoT have applied SPN as encryption technologies. Though, a few works like [29][45][60] and [67] applied FN assisted SPN for encryption; however, the key size used were 128 or 256 bits. These approaches can easily be visualized to be highly computationally complex and exhaustive, demanding higher gate element (GE) for hardware realization. Another lightweight encryption models such as SPARX [49], LAX [49], Chaskey [52] and LEA [57] employed SPN with ARX-based S-boxes for encryption. However, these approaches demanded higher key-size (128 minimum) that makes them computationally as well as resource exhaustive.

TABLE VI. RELATIVE PERFORMANCE ANALYSIS

Technique	**NPCR (%)**	**UACI (%)**	**Correlation**	**Encryption Time (ms)**
[43]	50.1029	99.6460	-	-
[39]	98.85	33.31	-	140 ms for 1126 symbols or characters
GFS-SSPN	99.63	43.01	0.0082	6.70 ms

TABLE VII. DIFFERENT LIGHTWEIGHT CRYPTOSYSTEMS FOR IOT SECURITY

Ciphers	**Technique**	**Key Size**	**Block size**	**No. of Rounds**
Improved Lilliput [44]	EGFN	80	64	30
GIFT [23]	SPN	128	64/128	28/40
SIT [45]	Feistel + SPN	64	64	5
DLBCA [46]	Feistel	80	32	15
LiCi [48]	Feistel	128	64	31
SKINNY [24]	SPN	64-384	64/128	32-56
MANTIS [24]	SPN	128	64	10/12
SPARX [49]	SPN with ARX-based S-boxes	128/256	64/128	24-40
LAX [49]	SPN with ARX-based S-boxes	128/256	64/128	24-40
RoadRunneR [50]	Feistel	80/128	64	10/12
PICO [25]	SPN	128	64	32
RECTANGLE [51]	SPN	80/128	64	25
Chaskey [52]	SPN with ARX-based S-boxes	128	128	8
OLBCA [23]	SPN	80	64	22
ITUBee [53]	Feistel	80	80	20
HISEC [47]	Feistel	80	64	15
LAC [54]	Feistel	80	64	16
SIMON [55]	Feistel	64/ 72/ 96/ 128/ 144/ 192/ 256	32/48/64/96/128	32/36/42/44/52/54/68/69/72
SPECK [55] [69]	Feistel	32/ 64/ 72/ 96/ 128	64/ 72/ 96/ 128/ 144/ 192/ 256	22/ 23/ 26/ 27/ 28/ 29/ 32/ 33/ 34
FeW [56]	Feistel	80/128	64	32
LEA [57]	SPN with ARX-based S-boxes	128/192/256	128	24/28/32
SCREAM [26]	SPN	128	128	10/12
PRINCE [27]	SPN	128	64	12
Hummingbird-2 [29]	SPN + Feistel	128	64	4
TWINE [58]	GFN	80/128	64	36
LED [59]	SPN	64/128	64	32/48
LBlock [60]	Feistel + SPN	80	64	32
PICCOLO [61]	GFN	80/128	64	25/31
KLEIN [28]	SPN	64/80/96	64	12/16/20
CLEFIA [62]	Feistel	128/192/256	128	18/22/26
PRESENT [63]	SPN	80/128	64	31
SEA [64]	Feistel	96	96	93
mCrypton [65]	SPN	64/96/128	64	12
TDEA [66]	Feistel	64	64	48
Camelia [67]	Feistel + SPN	128/192/256	128	18/24/24
KHAZAD [68]	SPN	128	64	3
Proposed GFN-SPNN	**GFS+SSPN**	**64**	**64**	**5**

B. Qualitative Assessment

In addition to the above discussed performance assessment, to assess attack-resilience of the proposed GFS-SSPN lightweight encryption system we performed qualitative assessment. In this method we examined the robustness and attack-resilience of the proposed lightweight encryption system in IoT-ecosystem [70]. In real-world IoT communication environment, an attacker can intercept the cipher stored (using SCLA), communicated (Linear and differential attack or impersonation) etc. and can attack the same to retrieve the targeted content or credential. Functionally, a cipher can be stated to be breached in case the attacker gets access or becomes able to retrieve the secret key. The encryption method applied in the proposed model was robust enough to alleviate the different kinds of attack-vulnerability. In this reference, we examine its robustness theoretically.

1) Differential and linear cryptanalysis: The proposed GFS-SSPN block-cipher model employs DKE component that functionally applies different linear and non-linear mathematical functions, substitution a permutation, along with cyclic flipping concepts to introduce higher randomness in cipher without using large key or higher encryption rounds. This approach strengthens it to avoid any cryptanalysis or linear and differential attack probabilities. Moreover, as discussed above, the correlation in between the plaintext and the ciphertext is significantly low exhibiting higher imperceptibility that can be vital to avoid any linear attacks in IoT-ecosystem. Moreover, GFS assisted confusion and diffusion over five consecutive rounds too strengthened the proposed model to achieve higher attack-resilience. Additionally, since the (each) round transformation is maintained uniform which enables treating each bit similar and hence facilitates resilience to the differential attack. The results obtained in terms of NPCR and UACI, as discussed above reveals that the proposed block-cipher model can have sufficient resilience to avoid any kind of differential attacks probability over IoT ecosystem.

2) Weak key combination: In major operating conditions, users make a common mistake by keeping poor or weak key combination that helps attacker to get easy access to the ciphers. On contrary, the cipher information where the non-linear operations usually rely on the key value maps the block cipher in such manner that it causes detectable weakness. On the other hand, looking into the proposed security model where it avoids using the same (actual) key in the cipher (due to multiple round key manipulation and/or exchange by XORing the actual key followed by DKE for five rounds). It makes our proposed GFS-SSPN model robust enough to avoid any kind of week-key attack probability.

3) Related keys combination trial attack: The attack can be made with the help of certain partially known or unknown keys as well. The related keys primarily depend on either slow diffusion or possessing symmetry in key expansion block, as discussed in the previous section. In our proposed security model, we crafted the key expansion mechanism in such manner that it retains fast computation and non-linear diffusion, especially for the cipher key difference in comparison to the round keys that makes significant confusion to assess related key for credential level or under transit data attack.

4) Square-attack: To assess efficacy of a security model, different attack modules are applied to investigate attack-resiliency by the proposed approach. Some of the key approaches applied in cloud-sensitive security models are the RS-Analysis and Square Attack. Considering Square Attack condition, it is capable enough to retrieve one byte of the last key combination and intends to retrieve or recover rest of the keys by repeating the attack iteratively. Let, such repetition be eight times, then also to achieve above stated information, the attacker needs to identify 28 keys precisely by 28 plaintexts which is equivalent to 216- S-box lookups. This becomes highly complicate and thus the proposed model can avoid such attack efficiently.

5) Interpolation attack: In general, such kinds of attacks primarily rely on the generic architecture of the cipher components which could generate certain rational expression with relatively low complexity. However, as already discussed the S-box expression of the proposed security system with diffusion characteristics strengthen it to avoid such limitations and thus makes it impracticable enough to avoid attack.

The above-mentioned relative performance assessment (Table VIII and Table IX), affirms suitability of the proposed GFS-SSPN lightweight security model towards realistic IoT ecosystems.

TABLE VIII. COMPARISON OF TWO FACTOR AUTHENTICATION PERFORMANCES FOR A CONVENTIONAL MODEL WITH OUR PREVIOUS WORKS

Security property	**Amin [71]**	**Tan [72]**	**Xie [73]**	**TFA-PUF-IoT [74]**	**TFA-RPUF-IoT [76]**	**Proposed GFS-SSPN**
Resilience to the impersonation attack	Yes	Yes	Yes	Yes	Yes	Yes
Anonymity and un traceability	Yes	No	Yes	Yes	Yes	Yes
Resilience to the password guessing attack	No	Yes	Yes	Yes	Yes	Yes
Prevents clock synchronization problem	No	Yes	No	Yes	Yes	Yes
Device security	No	No	No	Yes	Yes	Yes
Deployed security algorithm	ECC	ECC	ECC	PUF and FE	RPUF-FE	GFS-SSPN
Random response for every clock cycle	No	No	No	No	Yes	Yes
SCLA	Yes	No	No	No	Yes	Yes
Brute Force	No	No	No	No	No	Yes

TABLE IX. COMPARISON OF SECURITY PERFORMANCES

Comparison matrices	Aman [75]	TFA-PUF-IoT [74]	TFA-RPUF-IoT [76]	GFS-SSPN
Mutual authentication	Yes	Yes	Yes	Yes
Two factor secrecy	No	Yes	Yes	Yes
Privacy of the IoT devices	No	Yes	Yes	Yes
Consideration of noise in the PUF	No	Yes	Yes	Yes
Protection against physical attacks	Yes	Yes	Yes	Yes
Random response for every clock cycle/encryption round	No	No	Yes	Yes
Block cipher	No	No	No	Yes

V. CONCLUSION

This paper primarily focused on designing a state-of-art new and robust dynamic programming assisted lightweight encryption system for IoT security. Unlike classical cryptosystem-based approaches which often undergo increased computational overhead, latency and more importantly resource exhaustion, this research contributed a state-of-art new lightweight encryption system. The proposed model intended to exploit efficacy of both SPN with GFS, where the first enabled optimal confusion and diffusion (in cipher) while the later enabled reduced computation with significantly smaller decryption cost. Additionally, towards “Fit-To-All” IoT security solution for both device-level security as well as data-level security, the proposed model employed Shannon Criteria based SPN (SSPN) which helped in error-free decryption. On the other hand, the use of GFS enabled block-cipher encryption helped in enhancing computation time as well as cost. Architecturally, the proposed GFS-SSPN model employed 64-bit dynamic keying with five rounds of encryption which retained minimal computation and allied cost. It can be vital towards resource efficient hardware realization. A key notable contribution of the proposed model can be the use of dynamic keying concept was derived by processing four 16-bit inputs by means of the SPN followed by cyclic rotation and concatenation. The use of 64-bit key with merely five round of encryption helped maintaining minimal computational overheads and time, which can be vital for IoT security systems. The overall proposed GFS-SSPN model was realized as block-cipher model and hence unlike bit-by-bit encryption it reduced computational overhead and delay significantly. Moreover, the use of predefined transformation coder for SPN (substitution purpose) enabled time-efficient confusion and diffusion process to assure IoT-centric efficacy. Noticeably, being the Feistel Network based Shannon Conditional GFS-SSPN based lightweight encryption model avoided distinct decryption program and therefore can be more resource and computationally efficient. In this reference, the proposed GFS-SSPN model was examined with the different inputs’ credentials, often in plaintext to perform encryption. To assess performance by the proposed GFS-SSPN model, different plaintext samples were taken into consideration where performance in terms of correlation, encryption time, NPCR and UACI confirmed superiority over the existing approaches, including other lightweight cipher models as well as symmetric key cryptographic concepts. The depth analysis revealed that the correlation in between the original plaintext and the encrypted cipher was significantly low 0.006, while it maintained NPCR of 99.6% and UACI of 27%. The encryption time observed was near 2.6 ms which is fairly in sync with major IoT systems and allied real-time encryption demands. The overall results affirmed that the proposed system can be well suited for IoT security system, where it can be applied as a device middleware to safeguard access-level security, while the same can be applied to encrypt the input to ensure attack-resilience during transmission. Qualitative assessment too revealed that the proposed model can be effective enough to alleviate the attacks of MITM type, SCLA, Interpolation as well as linear and differential attacks. Though, GFS-SSPN model exhibited satisfactory performance and efficacy towards IoT security, computationally efficient performance etc. Considering the hardware compatibility, the proposed system is developed using C programming language. However the power and resource profile could not be examined. In feature authors can implement the proposed method in a hardware efficacy so as to examine the power and resource utilization for real word realization.

REFERENCES

[1] V. A. Thakor, M. A. Razzaque and M. R. A. Khandaker, "Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities," in IEEE Access, vol. 9, pp. 28177-28193, 2021.

[2] Barki A., Bouabdallah A., Gharout S. and Traoré J., "M2M Security: Challenges and Solutions," in IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1241-1254, Secondquarter 2016

[3] J. Sun, H. Xiong, X. Liu, Y. Zhang, X. Nie and R. H. Deng, "Lightweight and Privacy-Aware Fine-Grained Access Control for IoT-Oriented Smart Health," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6566-6575, July 2020.

[4] B. Aboushosha, R. A. Ramadan, A. D. Dwivedi, A. El-Sayed and M. M. Dessouky, "SLIM: A Lightweight Block Cipher for Internet of Health Things," in IEEE Access, vol. 8, pp. 203747-203757, 2020.

[5] Philip M. A. and Vaithiyanathan, "A survey on lightweight ciphers for IoT devices," 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), Kollam, 2017, pp. 1-4

[6] X. Guo, J. Hua, Y. Zhang and D. Wang, "A Complexity-Reduced Block Encryption Algorithm Suitable for Internet of Things," in IEEE Access, vol. 7, pp. 54760-54769, 2019.

[7] U. Hijawi, D. Unal, R. Hamila, A. Gastli and O. Ellabban, "Lightweight KPABE Architecture Enabled in Mesh Networked Resource-Constrained IoT Devices," in IEEE Access, vol. 9, pp. 5640-5650, 2021.

[8] S. Banerjee et al., "A Provably Secure and Lightweight Anonymous User Authenticated Session Key Exchange Scheme for Internet of Things Deployment," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8739-8752, Oct. 2019.

[9] E. Uchiteleva, A. R. Hussein and A. Shami, "Lightweight Dynamic Group Rekeying for Low-Power Wireless Networks in IIoT," in IEEE Internet of Things Journal, vol. 7, no. 6, pp. 4972-4986, June 2020.

[10] V. Odelu, A. K. Das, M. Khurram Khan, K. R. Choo and M. Jo, "Expressive CP-ABE Scheme for Mobile Devices in IoT Satisfying

Constant-Size Keys and Ciphertexts," in IEEE Access, vol. 5, pp. 3273-3283, 2017

[11] Hameed A. and Alomary A., "Security Issues in IoT: A Survey," 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, 2019, pp. 1-5.

[12] M. Ali, M. Sadeghi and X. Liu, "Lightweight Revocable Hierarchical Attribute-Based Encryption for Internet of Things," in IEEE Access, vol. 8, pp. 23951-23964, 2020.

[13] W. Yu and S. Köse, "A Lightweight Masked AES Implementation for Securing IoT Against CPA Attacks," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 11, pp. 2934-2944, Nov. 2017.

[14] D. Vergnaud, "Comment on “Efficient and Secure Outsourcing Scheme for RSA Decryption in Internet of Things”," in IEEE Internet of Things Journal, vol. 7, no. 11, pp. 11327-11329, Nov. 2020.

[15] F. Noura; L. Sleem; M. Noura; M. M. Mansour; A. Chehab; R. Couturier; “A new efficient lightweight and secure image cipher scheme”, Multimed Tools Application (springer), 2017.

[16] Sehrawat D. and Gill N. S., “Lightweight Block Ciphers for IoT based applications: A Review”, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 5 (2018) pp. 2258-2270.

[17] Salami S. Al, Baek J., Salah K. and E. Damiani, "Lightweight Encryption for Smart Home," 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, 2016, pp. 382-388,

[18] Leander G., Paar C., Poschmann A., and Schramm K, “New Lightweight DES Variants” FSE 2007, LNCS, vol. 4593, pp. 196-210. Springer, 2007.

[19] Kane L. E., Chen J. J., Thomas R., Liu V. and Mckague M., "Security and Performance in IoT: A Balancing Act," in IEEE Access, vol. 8, pp. 121969-121986, 2020.

[20] Elhoseny M., Ramírez-González G., O. Abu-Elnasr M., S. A. Shawkat, N. Arunkumar and A. Farouk, "Secure Medical Data Transmission Model for IoT-Based Healthcare Systems," in IEEE Access, vol. 6, pp. 20596-20608, 2018.

[21] Noura H., Couturier R., C. Pham and Chehab A., "Lightweight Stream Cipher Scheme for Resource-Constrained IoT Devices," 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, 2019, pp. 1-8.

[22] Standaert F.-X., Piret G., Rouvroy G., Quisquater J.-J., and J.-D. Legat. ICEBERG: an Involutional Cipher Efficient for Block Encryption in Reconfigurable Hardware. In B. Roy and W. Meier, editors, Fast Software Encryption — FSE 2004, pages 279–298. Springer- Verlag, 2004.

[23] Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M. and Todo, Y., 2017, September. GIFT: a small PRESENT. In International Conference on Cryptographic Hardware and Embedded Systems (pp. 321-345). Springer, Cham.

[24] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P. and Sim, S.M., 2016, August. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Annual Cryptology Conference (pp. 123-153). Springer Berlin Heidelberg.

[25] Bansod, G., Pisharoty, N. and Patil, A., 2016. PICO: An Ultra Lightweight and Low Power Encryption Design for Ubiquitous Computing. Defence Science Journal, 66(3).

[26] Grosso, V., Leurent, G., Standaert, F., Varici, K., Journault, A., Durvaux, F., Gaspar, L. and Kerckhof, S., 2015. SCREAM Side-Channel Resistant Authenticated Encryption with Masking. CAESAR submission.

[27] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C. and Rombouts, P., 2012, December. PRINCE–a low-latency block cipher for pervasive computing applications. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 208-225). Springer, Berlin, Heidelberg.

[28] Gong, Z., Nikova, S. and Law, Y.W., 2011. KLEIN: A new family of lightweight block ciphers. RFIDSec. Springer, 7055, pp.1-18.

[29] Engels, D.W., Saarinen, M.J.O., Schweitzer, P. and Smith, E.M., 2011. The Hummingbird-2 Lightweight Authenticated Encryption Algorithm. RFIDSec, 11, pp.19-31.

[30] Mohd B. J. and Hayajneh T., "Lightweight Block Ciphers for IoT: Energy Optimization and Survivability Techniques," in IEEE Access, vol. 6, pp. 35966-35978, 2018.

[31] Mohammed, A.A. and Ibadi, A.O., 2017. A Proposed Non Feistel Block Cipher Algorithm.

[32] S. Tan, K. Yeow and S. O. Hwang, "Enhancement of a Lightweight Attribute-Based Encryption Scheme for the Internet of Things," in IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6384-6395, Aug. 2019.

[33] S. Roy, U. Rawat and J. Karjee, "A Lightweight Cellular Automata Based Encryption Technique for IoT Applications," in IEEE Access, vol. 7, pp. 39782-39793, 2019.

[34] E. Uchiteleva, A. R. Hussein and A. Shami, "Lightweight Dynamic Group Rekeying for Low-Power Wireless Networks in IIoT," in IEEE Internet of Things Journal, vol. 7, no. 6, pp. 4972-4986, June 2020.

[35] S. Atiewi et al., "Scalable and Secure Big Data IoT System Based on Multifactor Authentication and Lightweight Cryptography," in IEEE Access, vol. 8, pp. 113498-113511, 2020.

[36] N. Tsafack et al., "A New Chaotic Map With Dynamic Analysis and Encryption Application in Internet of Health Things," in IEEE Access, vol. 8, pp. 137731-137744, 2020.

[37] B. Aboushosha, R. A. Ramadan, A. D. Dwivedi, A. El-Sayed and M. M. Dessouky, "SLIM: A Lightweight Block Cipher for Internet of Health Things," in IEEE Access, vol. 8, pp. 203747-203757, 2020.

[38] L. Zhang, X. Gao and Y. Mu, "Secure Data Sharing with Lightweight Computation in E-Health," in IEEE Access, vol. 8, pp. 209630-209643, 2020.

[39] Murillo-Escobar M. A., Abundiz-PÈrez F., Cruz-Hern·ndez C., LÛpez-GutiÈrrez R. M., “A novel symmetric text encryption algorithm based on logistic map”, Proceedings of the 2014 International Conference on Communications, Signal Processing and Compute, pp. 49-53. 2014.

[40] Gan Z.-H. Chai X.-L., Han D.-J., and Chen Y.-R., “A chaotic image encryption algorithm based on 3-D bit-plane permutation," Neural Comput. Appl., vol. 31, no. 11, pp. 7111-7130, Nov. 2019.

[41] Zhou J., Li J. and Di X., "A Novel Lossless Medical Image Encryption Scheme Based on Game Theory with Optimized ROI Parameters and Hidden ROI Position," in IEEE Access, vol. 8, pp. 122210-122228, 2020.

[42] Noura F. Z. H, Mostefaoui A., “An efficient and secure cipher scheme for images confidentiality preservation”, Signal Process Image Communication, vol 42, pp.90–108, 2016.

[43] F. Noura; L. Sleem; M. Noura; M. M. Mansour; A. Chehab; R. Couturier; “A new efficient lightweight and secure image cipher scheme”, Multimed Tools Application (springer), 2017.

[44] Mumthaz Pookuzhy Ali, Geethu T George, 2017. “Optimised Design of Light Weight Block Cipher Lilliput with Extended Generalised Feistal Network (EGFN).” International Journal of Innovative Research in Science, Engineering and Tech., 2017.

[45] Usman, M., Ahmed, I., Aslam, M.I., Khan, S. and Shah, U.A., 2017. SIT: A Lightweight Encryption Algorithm for Secure Internet of Things. arXiv preprint arXiv:1704.08688.

[46] AlDabbagh, S.S.M., 2017. Design 32-bit Lightweight Block Cipher Algorithm (DLBCA). International Journal of Computer Applications, 166(8).

[47] AlDabbagh, S.S.M., Shaikhli, A., Taha, I.F. and Alahmad, M.A., 2014, September. HISEC: A new lightweight block cipher algorithm. In Proceedings of the 7th International Conference on Security of Information and Networks (p. 151). ACM.

[48] Patil, J., Bansod, G. and Kant, K.S., 2017, February. LiCi: A new ultra-lightweight block cipher. In Emerging Trends & Innovation in ICT (ICEI), 2017 IEEE International Conference, pp. 40-45.

[49] Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J. and Biryukov, A., 2016. Design strategies for ARX with provable bounds: Sparx and LAX. In Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology

and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22(pp. 484-513). Springer Berlin Heidelberg.

[50] Baysal, A. and Şahin, S., 2015, September. Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors. In International Workshop on Lightweight Cryptography for Security and Privacy (pp. 58-76). Springer, Cham.

[51] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B. and Verbauwhede, I., 2015. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences, 58(12), pp.1-15.

[52] Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B. and Verbauwhede, I., 2014, August. Chaskey: an efficient MAC algorithm for 32-bit microcontrollers. In International Workshop on Selected Areas in Cryptography (pp. 306-323). Springer, Cham.

[53] Karakoç, F., Demirci, H. and Harmancı, A.E., 2013, May. ITUbee: a software oriented lightweight block cipher. In International Workshop on Lightweight Cryptography for Security and Privacy (pp. 16-27). Springer, Berlin, Heidelberg.

[54] Zhang, L., Wu, W., Wang, Y., Wu, S. and Zhang, J., 2014. LAC: A lightweight authenticated encryption cipher. CAESAR competition.

[55] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B. and Wingers, L., 2013. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptol ogy ePrint Archive, Report 2013/404.

[56] Kumar, M., Pal, S.K. and Panigrahi, A., 2014. FeW: A Lightweight Block Cipher. IACR Cryptology ePrint Archive, 2014, p.326.

[57] Hong, D., Lee, J.K., Kim, D.C., Kwon, D., Ryu, K.H. and Lee, D.G., 2013, August. LEA: A 128-bit block cipher for fast encryption on common processors. In International Workshop on Information Security Applications (pp. 3-27). Springer, Cham.

[58] Suzaki, T., Minematsu, K., Morioka, S. and Kobayashi, E., 2011, November. Twine: A lightweight, versatile block cipher. In ECRYPT Workshop on Lightweight Cryptography.

[59] Guo J., Peyrin T., Poschmann A., and Matt Robshaw M, Preneel B. and Takagi T., 2011. The LED Block Cipher. CHES 2011, In International Association for Cryptologic Research, LNCS 6917 (pp. 326–341).

[60] Wu, W. and Zhang, L., 2011. LBlock: a lightweight block cipher. In Applied Cryptography and Network Security (pp. 327-344). Springer Berlin/Heidelberg.

[61] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T. and Shirai, T., 2011, September. Piccolo: An ultra-lightweight blockcipher. In CHES (Vol. 6917, pp. 342-357).

[62] Shirai, T., Shibutani, K., Akishita, T., Moriai, S. and Iwata, T., 2007, March. The 128-bit block cipher CLEFIA. In FSE (Vol. 4593, pp. 181-195).

[63] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J., Seurin, Y. and Vikkelsoe, C., 2007, September. PRESENT: An ultra-lightweight block cipher. In CHES (Vol. 4727, pp. 450-466).

[64] Mace F., Standaert F.-X., and Quisquater J.-J, “ASIC Implementations of the Block Cipher SEA for Constrained Applications”, In RFID Security-RFID sec 2007, Workshop Record, pages 103 – 114, Malaga, Spain, 2007.

[65] Lim C. and Korkishko T. mCrypton - A Lightweight Block Cipher for Security of Low-cost RFID Tags and Sensors. In J. Song, T. Kwon, and M. Yung, editors, Workshop on Information Security Applications-WISA 2005, volume 3786 of Lecture Notes in Computer Science, pages 243–258. Springer-Verlag, 2005.

[66] Barker, W.C. and Barker, E., 2012. Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher: NIST Special Publication 800-67, Revision 2.

[67] Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J. and Tokita, T., 2000, August. Camellia: A 128-bit block cipher suitable for multiple platforms-design and analysis. In Selected Areas in Cryptography (Vol. 2012, pp. 39-56).

[68] Barreto, P.S.L.M. and Rijmen, V., 2000. The Khazad legacy-level block cipher. Primitive submitted to NESSIE, 97.

[69] Ray B., Douglas S., Jason S., Stefan T., Bryan W., and Louis W., “The simon and speck families of lightweight block ciphers,” Cryptology ePrint Archive, Report. /404, Tech. Rep., 2013.

[70] Urunov K., Namgung J. and Park S., "Security analysis based on Trusted Environment (TRE) of M2M/IoT," 2015 17th Asia-Pacific Network Operations and Management Symposium, Busan, 2015, pp. 554-557,

[71] R. Amin, S. K. Islam, M. K. Khan, A. Karati, D. Giri, and S. Kumari, “A two-factor RSA-based robust authentication system for multi-server environments”, Security and Communication Networks, 2017.

[72] J. Qu, and X. L. Tan, “Two-factor user authentication with key agreement scheme based on elliptic curve cryptosystem”, Journal of Electrical and Computer Engineering, 2014.

[73] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, “Provably secure dynamic ID- based anonymous two-factor authenticated key exchange protocol with extended security model”, IEEE Transactions on Information Forensics and Security, Vol. 12, No. 6, pp. 1382-1392, 2017.

[74] P. Gope and B. Sikdar, “Lightweight and privacy-preserving two-factor authentication scheme for IoT devices”, IEEE Internet of Things Journal, Vol. 6, No. 1, pp. 580-589, 2018.

[75] M. N. Aman, K. C. Chua, and B. Sikdar, “Mutual authentication in IoT systems using physical unclonable functions”, IEEE Internet of Things Journal, Vol. 4, No. 5, pp. 1327-1340, 2017.

[76] K. P. Gurumanapalli, N. Muthuluru, “ A Non Linear PUF Circuit Design for Two Factor Authentication in IoT Cryptography”, International Journal of Intelligent Engineering and Systems, Vol.14, No.1, 2021.