

# Symmetrical cryptographic algorithms in the lightweight internet of things

Akshaya Dhingra, Vikas Sindhu, Anil Sangwan

Department of Electronics and Communication Engineering, University Institute of Engineering and Technology,  
Maharshi Dayanand University, Rohtak, India

## Article Info

### Article history:

Received May 6, 2024  
Revised Aug 14, 2024  
Accepted Sep 22, 2024

### Keywords:

Cryptography  
Cybersecurity  
IoT  
LCAs  
SLCAs

## ABSTRACT

The internet of things (IoT) has emerged as a prominent area of scrutiny. It is being deployed in multiple applications like smart homes, smart agriculture, intelligent surveillance systems, and even innovative industries. Security is a significant issue that needs to be addressed in low-power IoT networks. This paper aims to describe symmetrical lightweight cryptographic algorithms (SLCAs) for lightweight IoT networks. The article focuses on discussing the principal difficulties of using cryptography in lightweight IoT devices, exploring SLCAs and their types based on structure formation throughout the literature survey, and comparing and evaluating different LCAs proposed in recent research. The main goal is to demonstrate how to solve the issues associated with conventional cryptography techniques and how lightweight cryptography algorithms aid limited IoT devices in achieving cybersecurity objectives.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Akshaya Dhingra  
Department of Electronics and Communication Engineering  
University Institute of Engineering and Technology, Maharshi Dayanand University  
Rohtak, Haryana, India  
Email: akshaya.rs.uiet@mdurohtak.ac.in

## 1. INTRODUCTION

The internet of things (IoT) has emerged as a prominent area of scrutiny due to its many potential uses in areas like transportation, industry 4.0, homes, offices, healthcare, farms, malls, and surveillance. IoT refers to a network of interconnected, individually identifiable items that can communicate and gather data over the Internet, whether it involves human contact or not [1]. One of the essential components of an IoT system, application, or solution is IoT devices. IoT devices are divided into two categories, i.e., conventional IoT devices (which are rich in resources) and light-weight IoT devices (LWID) (which are resource-constrained), as shown in Figure 1.

Over the last few decades, there has been an intensification in demand for IoT applications that use resource-constrained nodes. This rise in the number of connected LWIDs on diverse platforms gives birth to multiple unprecedented security threats in IoT networks [2]. So, cybersecurity is a significant issue of concern for LWID that requires regular system updates, privacy and regulation standards, availability, confidentiality, data integrity, authentication, and authorization, as shown in Figure 2 [3].

Cryptography refers to converting plain text into cipher text using specific algorithms. There are two types of cryptographic algorithms: conventional algorithms (designed as per traditional IoT device specifications) and lightweight cryptographic algorithms (LCAs). LCAs are explicitly intended for LWID-based applications. LCAs are further categorized into symmetrical lightweight cryptographic algorithms (SLCAs) and asymmetrical lightweight cryptographic algorithms (ALCAs). Both are based on the exchange

of cipher keys between the transmitter and the receiver. This paper will mainly focus on the importance of security in IoT networks and how SLCA can address cybersecurity objectives in limited IoT devices. Our study compares and evaluates different LCAs proposed in recent research. The study also describes the challenges faced by the designers of lightweight IoT devices, including restricted processing power and memory, power resources, and data security [4].

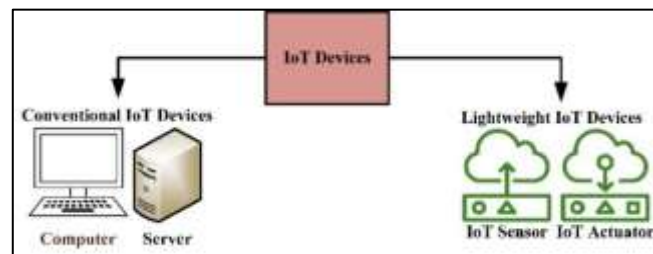


Figure 1. IoT devices



Figure 2. Cybersecurity challenges in lightweight IoT

## 2. METHOD

This section reviews previous research articles on lightweight cryptography for IoT devices. The review includes articles discussing the problems associated with conventional cryptography in lightweight IoT devices and proposing lightweight cryptography algorithms that can overcome these problems. McKay *et al.* [2], challenges posed by IoT devices, the need for LCAs to secure such devices, and the various parameters by which lightweight block ciphers are evaluated. The authors also described measures to form a hybrid cryptosystem for securing IoT devices. Thakor *et al.* [5], have deliberated on the importance of lightweight cryptography in minimizing new security threats imposed by IoT devices, which are becoming more common in different platforms. It also compares the performance of proposed LCAs, specifically for lightweight block ciphers. Gupta and Kumar [6] discusses the need for security in communication, particularly in solid encryption methodology for transmission, and the issues and possible countermeasures in secure transmission for IoT-enabled devices. Srinivas *et al.* [7], discuss the importance of securing IoT devices with lightweight cryptography algorithms. It emphasizes the limitations of traditional cryptography algorithms for IoT devices with restricted processing power, memory, and battery life. LCAs are designed to provide strong security while minimizing the computational and memory resources required for implementation. Rajesh and Prabha [8] discusses securing data in the IoT ecosystem and proposes a lightweight cryptographic solution using elliptic-curve cryptography (ECC). The paper systematically surveys previous work and provides information on ECC, requirements, and solution architecture. Goyal *et al.* [9], describe energy-efficient LCAs for IoT devices, focusing on exploring algorithms that can work within the constrained limits of these devices. The study presents the findings of the hardware implementation of the cryptography algorithms PRESENT, AES, ECDH, DH, and rivest-shamir-adleman (RSA), with each algorithm requiring different crypt-analysis techniques for “difficulty to break” measurement. For a 128-bit key length, the temporal complexity of the suggested PRESENT algorithm’s break attack is 2127. Table 1 summarizes previous works on SCAs with each algorithm’s structure, key size, rounds, and weaknesses.

The aforementioned articles provide a comparative evaluation of LCAs in IoT networks. This section mainly explains how LCAs can optimize the performance of constrained IoT devices by overcoming the limitations of traditional cryptography. The section concludes by emphasizing IoT integration and transmission of high data securely through IoT devices, highlighting future research gaps that should be explored for optimal utilization of lightweight cryptographic primitives. The rest of the paper is arranged as follows: the section 1 gives an overview of IoT and key challenges in implementing cryptography in lightweight IoT devices, section 2 analyses prior research articles related to LCAs. Section 3 discusses the need for cryptography in lightweight IoT networks. Section 4 discusses the types of LCAs. Finally, section 5 concludes the paper.

Table 1. Summary of previous works related to SCAs

Types of symmetric LCAs	Structure	Algorithm	Key size (in bits)	Rounds	Weakness	
Stream	ARX	Salsa20 [10]	128, 256	12 or 8	Relies on simple ARX operation	
		ChaCha [11]	128	64	No	
		Rabbit [12]	128	1	No	
		RC4 [12]	1-256		It relies on XOR operation only.	
	LFSR	A5/1 [1]	64	2 <sup>64</sup>	Prone to active and time-memory tradeoff attack	
		LILI-128 [13]	128	--	Prone to correlation and stream cipher attacks	
		MICKEY 2.0 [14]	0 -80	2 <sup>40</sup>	Problem in data communication and sharing	
	NFSR	Trivium [15]	64	1152	Prone to devastating attacks	
		Trivium-SC [16]	128	1000	Prone to cube attacks	
		Kreyvium [17]	128	872	Prone to distinguishing attack	
	FCSR	F-FCSR [12]	80	--	An attacker can quickly break the keystream	
	Block	SPN	AES [18]	128	128	Prone to man-in-the-middle (MITM) and side-channel attacks
			PRESENT [19]	128	64	MITM and key-related attacks
			GIFT [20]	64	28	Differential and square cryptanalysis attacks
			PRINCE [20]	128	64	Differential cryptanalysis attacks
		FN	DESL [21]	56	64	
TEA [22]			128	64	Key related attacks	
GFN		CLEFIA [23]	128	128	Integral cryptanalysis attack	
		Piccolo [23]	80	64	Differential and linear cryptanalysis attack	
ARX		SPECK [24]	96	48	Blique attack	
		HIGHT [25]	128	64	Linear cryptanalysis attack	
NFSR		KATAN [26]	80	32	Key related and MITM	
		Halka [27]	64	80	--	
Hybrid		Block and Stream	Hummingbird [18]	128	16	Key related attacks
			Hummingbird-2 [18]	128	16	Key related attacks
			PRESENT-GRP [28]	64	64	Multiple attacks

### 3. WHY THERE IS A NEED FOR CRYPTOGRAPHY IN LIGHTWEIGHT IOT NETWORKS?

Most lightweight IoT devices have minimal network resources, so there is a need to protect the information being transferred to/from these devices. So, cryptography is considered one of the most straightforward techniques to protect information by converting it into cipher text using keys. However, in a lightweight IoT network, the nature of IoT devices is constrained, which is why traditional cryptographic algorithms do not apply to this network [2], [29]. Hence, LCAs are deployed in these networks. The three primary features of LCAs are physical cost, performance, and security. LCAs also have each of these characteristics further observed in terms of physical space occupied, memory demand and energy consumption as implementation costs, processing power in terms of latency and through as performance (speed), length of a block or key, and several attack models involving fault-injection and side-channel attacks as a safeguard measure [30]. Table 2 shows all the characteristics a LCA must offer to protect the network.

Table 2. Characteristics of an LCA

Characteristics	What does an LCA offer?
Physical (like energy consumption, memory, and area)	Generation of a small key with the most minor computational capability
Performance	Best with low overheads
Security	Strong Structure for mitigation of attacks

#### 4. TYPES OF LIGHTWEIGHT CRYPTOGRAPHY ALGORITHMS

An algorithm is considered “lightweight” if it requires less memory, has a more minor key, and takes less time to execute than a conventional heavyweight algorithm [31]. Just like traditional encryption methods, LCAs are generally divided into two categories: symmetric and asymmetric methods. The algorithms that use the same key for encrypting and decrypting data are known as symmetric or private key cryptographic algorithms. The cryptographic algorithms that work with two keys are asymmetric or public key cryptographic algorithms [4], [32]. Figure 3 shows the structure-wise classification of LCAs.

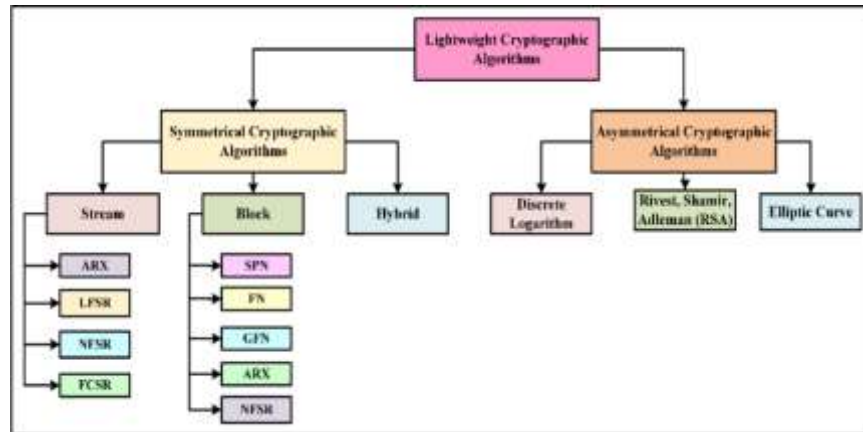


Figure 3. Structure-wise classification of lightweight cryptographic algorithms

##### 4.1. Symmetrical lightweight cryptographic algorithms

In SLCAs, the sender and recipient share a unique key. By encrypting the communication using a secret key, the sender creates a message that people cannot interpret. Conversely, the original text is found outside the communication once the recipient decrypts it using the same secret key [3]. SLCA is a popular choice for users because of its quick computation times, low algorithm complexity, and straightforward computational procedures. However, asymmetric lightweight cryptography algorithms (ALCAs) are preferred over SLCAs as they use public and private keys and are more secure in real-time applications. SLCAs are less secure since they cannot achieve non-repudiation and authentication. SLCAs are divided into stream, block, and hybrid cryptographic algorithms [33].

##### 4.1.1. Stream cipher cryptographic algorithms (SCCAs)

The data are encrypted bit by bit using this kind of encryption. Diffusion and confusion features are not accomplished because each encrypted bit is independent of the others. In this kind of cipher, the operators are kept as basic as feasible in the encryption process. It includes algorithms based on structures like addition-rotation-XOR (ARX), linear feedback shift registers (LFSR), nonlinear feedback shift register (NFSR), and shift register with carry feedback (FCSR) [34], [35].

##### A. ARX-based SCCAs

To attain the necessary security strength, several contemporary stream ciphers are presented as ARX-based algorithms, whose round function only requires the three hybrid operations of module addition ( $\boxplus$ ), interword-rotation ( $\gg$ ), and XOR ( $\oplus$ ). ARX-based stream ciphers and the security analysis technique are designed to protect lightweight IoT that has advanced structure. The ARX structure-based SCCAs include ChaCha, Salsa20, Rabbit, and RC4 [12]. Salsa20 [10] and the ChaCha [11] family are closely associated and focus on 32-bit ARX operation-based core hash functions. Salsa20 was the original cipher selected in the eSTREAM software profile; ChaCha is a 2008 version that uses a new round function to boost diffusion. Salsa20 and ChaCha function using an internal state comprising sixteen  $4 \times 4$  matrixes of 32-bit words. A 256-bit key (a 128-bit key can also be used with Salsa20), a 64-bit nonce, and a 64-bit counter map to the keystream of 512-bit blocks. Whereas ARX-based Rabbit [12] is selected in the eSTREAM software profile, up to 264 keystream blocks are generated using a 128-bit key and a 64-bit IV. The internal state comprises 513 bits, eight 32-bit counters, eleven 32-bit state variables, and one counter-carry bit. One of the most often used ARX-based SSCA is RC4 [12]. It's also sometimes referred to as ARCFOUR or ARC4. This method uses a stream independent of the plaintext and has a customizable key size ranging from 1 to 256 bytes.

## B. LFSR-based SCCAs

In LFSR structure-based algorithms, the driving part and nonlinear part are used in stream cipher designing, where the nonlinear component is mainly used to cover the linear qualities of source sequences to build keystreams with high nonlinearity, and the driving part is utilized to generate fundamental source sequences with good properties, such as the m-sequence [18]. This includes algorithms like A5/1, LILLI, Mickey 2.0, Approximately 130 million GSM users in Europe use the encryption method A5/1 [1] to safeguard their cellular voice and data communication privacy over the air. The generator's starting state, which yields 228 pseudorandom bits, is created for each frame by combining the 64-bit session key with a 22-bit publicly available frame counter. LILI-128 [13] was created for the NESSIE project and comprises two subsystems for data production and clock control, each with two filter functions and two binary LFSRs with lengths of 89 and 39 bits. Another LFSR-based hardware-focused eSTREAM-based algorithm is MICKEY 2.0 [14]. It transfers a variable length IV (0 to 80 bits) and an 80-bit secret key to a keystream with a maximum length of 240 bits. Two registers, each with a length of 100 bits, make up the generator.

## C. NFSR-based SCCAs

NFSR-based stream ciphers use non-linear output and non-linear feedback to offer strong sequence security and characteristics. Some NFSR structure-based algorithms are Trivium, TriviaA-SC, and Kreyvium [10]. Trivium [15] is part of the eSTREAM final portfolio, which produces up to 264 keystream bits using an 80-bit key and an 80-bit IV. Its internal state is 288 bits long. The stream cipher based on Trivium, which has been updated to have a considerably bigger internal state, is called Trivia-SC [16]. Trivia-SC employs three NFSRs with sizes of 132, 105, and 147 bits each and is loaded with a 128-bit key and 128-bit IV. Kreyvium [17] is focused on compressing homomorphic ciphertexts efficiently, using five registers. Although the 128-bit top and bottom registers have been modified from Trivium, the three middle registers with lengths of 93, 84, and 111 bits are Trivium-corresponding.

## D. FCSR-based SCCAs

Within this configuration, 2-adic rational numbers power the FCSR, an extra shift register. After adding carries to the intrinsic nonlinearity from the quadratic transition function, the sequence formed from an FCSR has the same acceptable statistical qualities as the LFSR sequence, such as equal distribution of patterns, balancedness, and a known period. A linear filter can extract the keystream from the internal state, as FCSR sequences are predictable by synthesizing techniques and hence not suitable for direct output [12], [36].

### 4.1.2. Block cipher cryptographic algorithms

Block cipher cryptographic algorithms (BCCAs) perform encryption and decryption on a fixed-size block (64 bits or more) simultaneously, while stream ciphers process the input parts bit by bit (or word by word). Claude Shannon established two fundamental aspects of cryptography, confusion and dispersion, to fortify the cipher. Diffusion employs permutation to distribute the plaintext's statistical structure throughout most of the ciphertext, and confusion uses substitution (S-box) to produce the most complex interaction between the ciphertext and the key [28]. In contrast to the block cipher, which has a simpler architecture than the stream cipher, the block cipher uses both confusion and diffusion properties. A stream cipher uses XOR function(s) to encrypt data that can be readily decrypted back to its original form. In contrast, a block cipher requires a complex encryption process to retrieve the original content. The BCCA includes algorithms based on a structure like substitution permutation network (SPN), feistel network (FN), generalized feistel network (GFN), ARX, and NFSR.

## A. SPN-based BCCAs

SPN operates on plaintext blocks, applies a key, and then uses substitution boxes (S-boxes) and permutation boxes (P-boxes) in many rounds. Bitwise rotation is commonly employed to accomplish the operations, and the operation rounds introduce portions of the key [9]. It includes algorithms like AES, PRESENT, GIFT, and PRINCE. An iconic example of an SPN-based algorithm is AES [18], which NIST standardized. It operates on a 128-bit block with 128, 192, and 256-bit vital variants. The S-boxes and P-boxes function in reverse. PRESENT [19] is another highly efficient SPN-based algorithm authorized by ISO/IEC and works well with software and hardware. It uses 64-bit blocks on two fundamental variants: 80-bit and 128-bit keys. GIFT [20] was introduced at CHES-2017 as an enhanced version of the PRESENT. It provides a more compact, lighter S-Box. With a more straightforward and faster critical schedule, fewer rounds result in high throughput. The two GIFT versions are GIFT-128, a 40-round with a 128-bit block size, and GIFT-64, a 28-round with a 64-bit block size. Both use a 128-bit key. PRINCE [20] algorithm is a lightweight and hardware-efficient method that operates on a 64-bit input, utilizing a 128-bit critical twelve times.

## B. FN-based BCCAs

The FN operates by dividing the input into two equal parts. One-half of the input will then be subjected to diffusion at each round, with the two halves being switched at the start of each subsequent round. Poschmann *et al.* [21] data encryption standard lightweight (DESL) is an FN-based BCCA that works with a block size of 64 bits, a key size of 56 bits, and an equivalent number of rounds as DES. DESL differs from DES because it uses a multiplexer and fewer S-boxes (eight to one). Israsena and Wongnamkum [22] tiny encryption algorithm (TEA) is an FN-based algorithm that provides appropriate ciphers for low-cost, small-sized, and computationally weak hardware. It does 32 rounds using a 128-bit key on a 64-bit input.

## C. GFN-based BCCAs

GFN is an improved version of FN that divides the input into many sub-blocks. Feistel functions are applied to each sub-block, followed by a proportionate cyclic shift. In that order, CLEFIA [23], a GFN-based BCCA developed by Sony and authorized by NIST, provides a 128-bit block with 128, 192, or 256-bit keys through 18, 22, and 26 rounds. Although relatively expensive, it exhibits excellent immunity against attacks and high performance. Another GFN-based extremely light BCCA that works with very constrained environmental devices (RFID and sensors) is Piccolo [23]. It uses two key sets, 80-bit and 128-bit, respectively, to process 64-bit input and execute two iterations, 25 and 31.

## D. ARX-based BCCAs

The NSA-designed SPECK [24] is a software-oriented ARX-based BCCA. The smallest hardware implementation of SPECK has a 48-bit block and a 96-bit key. Another extremely lightweight ARX-based BCCA is HIGHT [25] 64-bit data 32 times using a 128-bit key. It uses basic computing techniques to perform compact round functions (no S-boxes). It utilizes 128-bit, 192-bit, and 256-bit keys to process 128-bit input and carry out 24, 28, and 32 iterations.

## E. NFSR-based BCCA's

The NFSR structure can be used with stream and block ciphers. Its implementation is based on stream cipher blocks, where the current state is derived from the previous one. The cipher family KATAN/KTANTAN [26] is an NFSR-based BCCA that uses the 80-bit key on several block sizes (32-bit, 48-bit, and 64-bit) via 254 rounds. These could be used with small-scale hardware, primarily sensor networks and RFID tags. Halka [27] is another NFSR-based BCCA with suitable software and hardware performance. To complete 24 iterations in this scheme, 64 bits of input and an 80-bit key are required.

### 4.1.3. Hybrid SCAs

Hybrid SCAs include a structure combining both block and stream cipher techniques. Hummingbird [18] is an ultra-lightweight hybrid SCA that does 20 rounds using a 256-bit key and 16-bit input. It could have been attacked multiple times. So, a 128-bit key accepts 64-bit input (the starting vector) in Hummingbird-2 [18], which is intended for low-end microcontrollers. Both hardware and software platforms exhibit good performance. Although it uses 4-bit microcontrollers, it performs better than the present and has a few downsides. Since encryption (or decryption) relies on its stream characteristic, initialization is required before 2; diverse encryption and decryption features make the whole version 70% heavier than the encryption alone. Additionally, processing brief messages causes it to perform worse. PRESENT-GRP [28] is another hybrid SCA that executes 31 iterations using a 64-bit input and a 128-bit key. In place of the permutation table, the substitution-permutation approach from PRESENT is substituted with a group (GRP) operation for additional confusing features.

### 4.2. Asymmetrical lightweight cryptographic algorithms

Secret key algorithms use different keys to decrypt ciphertext and encrypt plaintext. It has two keys: a public key and a private key. Utilizing the public key allows the widely known sender text to be encrypted, and the private key unlocks the recipient's encrypted message. One of the primary benefits of asymmetric ciphers is that they share distinct keys (public and private) in contrast to symmetric ciphers. It now has the robustness of the first type. However, asymmetric encryption's primary drawback is its excessive energy consumption and slower speed than symmetric encryption. ECC and RSA are two well-known asymmetric critical methods utilized in cloud computing [36].

## 5. CONCLUSION

This article discusses the challenges of implementing cryptography in lightweight IoT devices and presents the concept of lightweight cryptography. It presents the idea of lightweight cryptography, a subset of

conventional cryptography, which addresses traditional encryption issues with features such devices have, like minimal memory and low processing power. The article focuses on SLCAs and their types based on structure formation. Overall, the article highlights the importance of SLCAs for achieving cybersecurity objectives for resource-constrained IoT devices.




## REFERENCES

- [1] M. Jabeen and K. Ishaq, "Internet of things in telecommunications: a technological perspective," *Journal of Information Technology Teaching Cases*, vol. 13, no. 1, pp. 39–49, May 2023, doi: 10.1177/20438869211067808.
- [2] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography," Gaithersburg, MD, Mar. 2017. doi: 10.6028/NIST.IR.8114.
- [3] A. Banafa, "3 major challenges IoT is facing | OpenMind," *OpenMind*, no. March 2017, pp. 1–9, 2017, [Online]. Available: <https://www.bbvaopenmind.com/en/technology/digital-world/3-major-challenges-facing-iot/>.
- [4] M. Rana, Q. Mamun, and R. Islam, "Lightweight cryptography in IoT networks: a survey," *Future Generation Computer Systems*, vol. 129, pp. 77–89, Apr. 2022, doi: 10.1016/j.future.2021.11.011.
- [5] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, "Lightweight cryptography for IoT: a state-of-the-art," *arXiv*, 2020.
- [6] D. N. Gupta and R. Kumar, "Lightweight cryptography: an IoT perspective," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 8, pp. 700–706, 2019.
- [7] T. A. S. Srinivas, A. D. Donald, I. D. Srihith, D. Anjali, and A. Chandana, "Small but mighty: the power of lightweight cryptography in IoT," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 52–67, Apr. 2023, doi: 10.48175/ijarsct-9008.
- [8] S. M. Rajesh and R. Prabha, "Lightweight cryptographic approach to address the security issues in intelligent applications: a survey," in *IDCIoT 2023 - International Conference on Intelligent Data Communication Technologies and Internet of Things, Proceedings*, Jan. 2023, pp. 122–128, doi: 10.1109/IDCIoT56793.2023.10053412.
- [9] T. K. Goyal, V. Sahula, and D. Kumawat, "Energy efficient lightweight cryptography algorithms for IoT devices," *IETE Journal of Research*, vol. 68, no. 3, pp. 1722–1735, May 2022, doi: 10.1080/03772063.2019.1670103.
- [10] C. De Canniere and B. P. Trivium, *New Stream Cipher Designs*, vol. 4986. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [11] Hellman, "ChaCha, a variant of Salsa20," *Workshop Record of SASC*, pp. 1–6, 2008, [Online]. Available: <http://cr.yp.to/chacha/chacha-20080120.pdf>.
- [12] L. Jiao, Y. Hao, and D. Feng, "Stream cipher designs: a review," *Science China Information Sciences*, vol. 63, no. 3, p. 131101, Mar. 2020, doi: 10.1007/s11432-018-9929-x.
- [13] L. R. Simpson, E. Dawson, J. D. Golić, and W. L. Millan, "LILI keystream generator," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2012, 2001, pp. 248–261.
- [14] S. Babbage and M. Dodd, "The stream cipher MICKEY 2.0," *ECRYPT Stream Cipher Project, Report*, pp. 1–12, 2006.
- [15] C. De Canniere and B. Preneel, "TRIVIUM specifications," *ECRYPT Stream Cipher Project, Report*, vol. 30, p. 2005, 2005.
- [16] A. Chakraborti, A. Chattopadhyay, M. Hassan, and M. Nandi, "TrivA: a fast and secure authenticated encryption scheme," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9293, 2015, pp. 330–353.
- [17] A. Canteaut *et al.*, "Stream ciphers: a practical solution for efficient Homomorphic-Ciphertext compression," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9783, 2016, pp. 313–333.
- [18] 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," *IEEE Access*, vol. 9, pp. 28177–28193, 2021, doi: 10.1109/ACCESS.2021.3052867.
- [19] A. Bogdanov *et al.*, "PRESENT: an ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, vol. 4727 LNCS, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 450–466.
- [20] S. Banik. *et al.*, "GIFT-COFB," *IACR Cryptology ePrint Archive*, 2020.
- [21] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "New light-weight crypto algorithms for RFID," in *Proceedings - IEEE International Symposium on Circuits and Systems*, May 2007, pp. 1843–1846, doi: 10.1109/iscas.2007.378273.
- [22] P. Irasena and S. Wongnamkum, "Hardware implementation of a TEA-based lightweight encryption for RFID security," in *RFID Security*, Boston, MA: Springer US, 2008, pp. 417–433.
- [23] J. Hosseinzadeh, "A comprehensive survey on evaluation of lightweight symmetric ciphers: hardware and software implementation," *ACSIJ Advances in Computer Science*, vol. 5, no. 4, pp. 31–41, 2016.
- [24] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proceedings - Design Automation Conference*, Jun. 2015, vol. 2015-July, pp. 1–6, doi: 10.1145/2744769.2747946.
- [25] D. Hong *et al.*, "HIGHT: a new block cipher suitable for low-resource device," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4249 LNCS, 2006, pp. 46–59.
- [26] C. De Cannière, O. Dunkelman, and M. Knežević, "KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5747 LNCS, 2009, pp. 272–288.
- [27] S. Das, "Halka: a lightweight, software friendly block cipher using ultra-lightweight 8-bit S-box," *IACR Cryptology ePrint Archive*, vol. 2014, p. 110, 2014, [Online]. Available: <http://dblp.uni-trier.de/db/journals/iacr/iacr2014.html#Das14a>.
- [28] G. Bansod, N. Raval, and N. Pisharoty, "Implementation of a new lightweight encryption design for embedded security," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 142–151, Jan. 2015, doi: 10.1109/TIFS.2014.2365734.
- [29] Z. A. Mohammed and K. A. Hussein, "Lightweight cryptography concepts and algorithms: a survey," in *2023 Second International Conference on Advanced Computer Applications (ACA)*, Feb. 2023, pp. 1–7, doi: 10.1109/ACA57612.2023.10346914.
- [30] Cryptrec, "CRYPTREC cryptographic technology guideline (lightweight cryptography)," *CRYPTREC: Lightweight Cryptography Working Group*, no. March, pp. 1–127, 2017, [Online]. Available: <https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016en.pdf>.
- [31] A. J. Acosta, E. Tena-Sánchez, C. J. Jiménez, and J. M. Mora, "Power and energy issues on lightweight cryptography," *Journal of Low Power Electronics*, vol. 13, no. 3, pp. 326–337, Sep. 2017, doi: 10.1166/jolpe.2017.1490.




- [32] L. M. Shamala, G. Zayaraz, K. Vivekanandan, and V. Vijayalakshmi, "Lightweight cryptography algorithms for internet of things enabled networks: An overview," *Journal of Physics: Conference Series*, vol. 1717, no. 1, p. 012072, Jan. 2021, doi: 10.1088/1742-6596/1717/1/012072.
- [33] M. K. Hasan *et al.*, "Lightweight cryptographic algorithms for guessing attack protection in complex internet of things applications," *Complexity*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/5540296.
- [34] S. A. Jassim and A. K. Farhan, "A survey on stream ciphers for constrained environments," in *1st Babylon International Conference on Information Technology and Science 2021, BICITS 2021*, Apr. 2021, pp. 228–233, doi: 10.1109/BICITS51482.2021.9509883.
- [35] M. Abujoodeh, L. Tamimi, and R. Tahboub, "Toward lightweight cryptography: a survey," in *Computational Semantics*, IntechOpen, 2023.
- [36] A. Kant, A. Dhingra, A. Sangwan, and V. Sindhu, "Building trust in the IoT: a comprehensive review of device authentication approaches," in *Proceedings of International Conference on Contemporary Computing and Informatics, IC3I 2023*, Sep. 2023, pp. 2287–2293, doi: 10.1109/IC3I59117.2023.10397655.

## BIOGRAPHIES OF AUTHORS






**Akshaya Dhingra**    is pursuing Ph.D. degree from the Department of Electronics and Communication Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University Rohtak, Haryana, India. She received an M.Tech. degree in Electronics and Communication Engineering in 2019 and a B.Tech. degree in Electronics and Communication Engineering in 2017 from Maharshi Dayanand University, Rohtak, Haryana, India. She has published more than 12 articles in reputed journals and conferences. Her areas of interest are communication networks, the IoT, and security. She can be contacted at email: akshaya.rs.uiet@mdurohtak.ac.in.



**Vikas Sindhu**    is an associate professor at Maharshi Dayanand University, Rohtak. He received a B.Tech. degree in Electronics and Communication Engineering in 2004, an M.Tech. degree in Electronics Instruments and Control Engineering in 2006, and Ph.D. from Maharshi Dayanand University Rohtak, Haryana, India. He has published around 30 research papers in national and international journals and conferences. His areas of interest are electronic devices, cognitive radio, electric vehicles, and the IoT. He can be contacted at email: vikassindhu.uiet@mdurohtak.ac.in.



**Anil Sangwan**    is an associate professor at Maharshi Dayanand University, Rohtak, Haryana. He received a bachelor's degree in Electronics and Communication Engineering and a master's degree in Electrical Instrumentation and Control Engineering from Maharshi Dayanand University, Rohtak, Haryana, India. He received his Ph.D. degree from Maharshi Dayanand University, Rohtak, Haryana, India. He has published around 30 research papers in national and international journals and conferences. He can be contacted at email: anilsangwan.uiet@mdurohtak.ac.in.