








# An Enhanced Lightweight Speck System for Cloud-Based Smart Healthcare

Muyideen AbdulRaheem<sup>1</sup> , Ghaniyyat Bolanle Balogun<sup>1</sup>,  
Moses Kazeem Abiodun<sup>2</sup> , Fatimoh Abidemi Taofeek-Ibrahim<sup>3</sup>,  
Adekola Rasheed Tomori<sup>4</sup>, Idowu Dauda Oladipo<sup>1</sup> ,  
and Joseph Bamidele Awotunde<sup>1</sup>  

<sup>1</sup> Department of Computer Science, University of Ilorin, Ilorin, Nigeria  
{muyideen, balogun.gb, odidowu, awotunde.jb}@unilorin.edu.ng

<sup>2</sup> Department of Computer Science, Landmark University, Omu Aran, Nigeria  
moses.abiodun@lmu.edu.ng

<sup>3</sup> Department of Computer Science, Federal Polytechnic, Offa, Nigeria

<sup>4</sup> Directorate of Computer Sciences and Information Technology,  
University of Ilorin, Ilorin, Nigeria  
tomori@unilorin.edu.ng

**Abstract.** In the realm of information and communication sciences, the Internet of Things (IoT) is a new technology with sensors in the healthcare sector. Sensors are critical IoT devices that receive and send crucial bodily characteristics like blood pressure, temperature, heart rate, and breathing rate to and from cloud repositories for healthcare specialists. As a result of technical advancements, the usage of these devices, referred to as smart sensors, is becoming acceptable in smart healthcare for illness diagnosis and treatment. Data generated from these devices is huge and intrinsically tied to every sphere of daily life including healthcare domain. This information must be safeguarded and processed in a safe location. The term “cloud computing” refers to the type of innovation that is employed to safe keep such tremendous volume of information. As a result, it has become critical to protect healthcare data from hackers in order to maintain its protection, privacy, confidentiality, integrity, and its processing mode. This research suggested a New Lightweight Speck Cryptographic Algorithm to Improve High - Performance Computing Security for Healthcare Data. In contrasted to the cryptographic methods commonly employed in cloud computing, the investigational results of the proposed methodology showed a high level of security and an evident improvement in terms of the time it takes to encrypt data and the security obtainable.

**Keywords:** Cloud computing · Lightweight · Speck encryption · Smart healthcare system · Medical data · Security and privacy

## 1 Introduction

Technology and System for cloud applications have in the recent past evolved dramatically. A high number of architectural and infrastructural distributed models have

emanated from these technologies. Cloud computing as one of the emerging technologies also called computing network, is classically connected utilizing the online and shared a large number of services distributed around the region to meet the need of users [1]. The National Institute of Science and Technology of India (NIST) idea of computational cloud is a context that offer a standardized collection of reconfigurable cloud computing with quick access, and ubiquitous access to the internet as needed. The purpose of using cloud computing is not limited to its manageability, scalability, and affordability but also on-demand storage facility that is characterized by uniformity, simplicity, leased variety, dependability, and flexibility [2].

A cloud client like Smart Healthcare System can use the cloud tools on request to store its data and can be accessed anywhere, and on any device at any time. The NIST concept on cloud identified three service models which are made available by Cloud Service Provider (CSP) to various cloud users. It also recognizes four models for deployment which are based on Public, Hybrid, Private, and Community Cloud, highlighting the framework for distributed computing resources easy organization and management cloud [3].

With these models available on the cloud infrastructure, the most critical issue facing the cloud is the exposure of its data to vulnerability and a range of hazards being a technology that uses network connectivity for its communication and transmission. Cloud computing's widespread adoption is hampered by security concerns [4]. In reality, the sharing of cloud computing services poses the difficulty of keeping these services secure and secured against unauthorized access or usage. Mostly the cloud client data outsourced to it faces this challenge [5]. One of the most important security challenges in cloud computing is Smart Healthcare data security, which involves both direct and indirect threats. Providing a safe connection between Smart Healthcare and cloud providers has been created to protect the safety of data transmissions on the cloud network. Lightweight Cryptography is the most effective for data security of Smart Healthcare System. The transformation to ciphertext from plaintext is involved in the process and its tools are always send contents safely by guaranteeing that only the authorized parties are able to retrieve them [6].

Similar to Information Technology Management Systems, Cloud Computing requires fundamental security apparatus such as confidentiality, availability and integrity, authorization, accountability, authentication and privacy as part of essential cloud protection measures. In this paper, an Enhanced Lightweight Speck System for Cloud-based Smart Healthcare was presented. The proposed method called Enhanced Lightweight Speck System (ELSS) improve cloud computing data security for smart healthcare systems. It uses encryption set of rules according to lightweight symmetric cryptography.

The paper is structured as follows. Section one is the introduction to the study. Section to examines background to the study while section three is the methodology used. Lastly, section is the discussion of the result.

## 2 Background

Many researches have explored the security challenges facing cloud computing. This part, however, highlights a few current findings that looked into security of cloud computing. Some categories of symmetric algorithms such as Advanced Encryption Standard

(AES) algorithm, Data Encryption Standard (DES), and International Data Encryption Algorithm (IDEA) were employed to secure cloud computing [7]. A comparison of these symmetric algorithms and asymmetric algorithms such as Rivest-Shamir-Adleman were introduced based on Key size, Rounds, Degree of Safety, and Execution Time. In a cloud setting, the results are relatively efficient [8].

Blowfish, AES, RSA, DSA, and Eclipse IDA were used in a hybrid encryption technique to improve the security of data stored in multiple cloud servers. The research focuses on giving clients the power to choose how their data is to be encrypted rather than relying on third parties to do it [9]. To reduce latency and processing time, an efficient cloud computing architecture categorization has been developed. The information was divided into three tiers to take full advantage of the security of cloud computing. The study recommended using hybrid encryption approaches like RSA Digital Signature and Blowfish algorithm for encoding and decoding with Feistel structure and Exclusive-OR operations [10]. Furthermore, to overcome the security issues of Cloud Storage, [10] combined two independent encryption techniques. It conducted a survey of past studies devoted to cloud data security and then proposed a combined form of defense encryption system based on MD5 and Blowfish to increase cloud server security.

Furthermore, [11] investigated the combination of several cryptographic methods to protect cloud data using attribute-based, homomorphic, searchable contemporary cryptography. A hybrid encryption model was constructed to maximize each system's ability to safeguard cloud information and an assessment of data encoding and decoding of AES-256 SHA-512, IDAs, and 3DES was implemented by [11]. For large and small data files, the technique delivers a substantially greater level of safety and performance. These efforts have not examined the issue in relation to healthcare system.

A great number of studies on cryptography techniques that are symmetric were developed for various applications using lightweight encryptions such RECTANGLE, TWINE, CLEFIA and others [12]. A lightweight encryption system is a  $n$ -bit block size having  $m$ -bit key generated cryptosystem and repeated in a number of rounds, using mainly XOR combined with left or right rotations LR or RR operation. It serves the purpose of provide protection for ubiquitous resource constrained devices like RFID tags and wireless sensor nodes.

CLEFIA-128 is a symmetric block cipher developed by Sony for both hardware and software [13]. The 128-bit block size encryption uses 128-bit key size, with 28 rounds of Feistel structure. TWINE cryptosystem is a lightweight block encryption for multiple stage using wide range Feistel structure cipher. It has a 64-bit block size and 36 cycles of either 80-bit or 128-bit key size. Every round contains a layer with a 4-bit S-box combined with a 4-bit block permutation in addition to layer of nonlinear substitution for diffusion and confusion purposes.

Another lightweight encryption is RECTANGLE cryptosystem. It is a 64 bits block cipher with a key size of either 80-bit or 128-bit, but runs 25 rounds only for its encryption and decryption processes to achieve optimized protection [14]. Stable IoT (SIT) is a 64-bit block lightweight encryption algorithm using a 64-bit address to encrypt data.

The combination of Feistel structure and a substitution-permutation architecture enable the network algorithm protects the data. The lightweight encryption algorithm encrypts data in IoT devices utilizing a block cipher with a 64-bit size and an 80-bit

key length. As an alternative, Data Encryption Lightweight Systems (DESXL), rather than multiple S-Boxes without any initial and final permutations, a single S-Box is used to improve security with a 184-bit key. No successful case of attack has been reported against DESXL [15].

The benchmarking framework [16] compares the execution times, RAM footprints, and binary code size of 19 existing lightweight cryptographic algorithms such as AES, Chaskey, and Speck. [16] indicates that the best cipher is Chaskey, with Speck second, when using C and assembly languages to create block encryption on 8-bit AVR, 16-bit MSP430, and 32-bit ARM microprocessor platforms. This benchmarking tool is used to assess lightweight cryptographic methods for IoT-enabled devices. The lightweight cryptography standardization for Internet of Things devices are being given consideration [17]. RSA, Attribute Based Encryption (ABE), Identity-Based Encryption (IBE), Elliptic Curve Cryptography (ECC), and other cryptographic methods and protocols have been employed in IoT. RSA is a cryptographic algorithm that encrypts and transports encrypted symmetric keys for use with cryptography techniques that are symmetric such as AES [18]. To ensure the safety of communication between IoT devices, ECC is utilized for key agreement and authentication [19].

IBE [20] streamlines certificate administration in IoT systems, allowing, without accessing the public key certificate, a sender (smart node) to encrypt a message for an entity. Another technique that has been offered is ABE, which has been adjusted to be suited for IoT devices by utilizing a proxy [21]. These fundamental cryptography techniques are either changed or used in conjunction with protocols or other algorithms to achieve confidentiality in IoT, as detailed below. Maintaining data privacy at restricted devices and the Internet of Things (IoT) connectivity is a critical responsibility. The IoT's resource restrictions and asymmetric capabilities make it difficult to achieve this using the conventional cryptographic methods outlined above. [21] look at various specific algorithms that have been recommended for use in IoT security, including their architecture, benefits, and limitations. Communication devices over the integrated network suffer confidentiality difficulties in a smart home context.

Salami et al. (2016) [22] propose a lightweight cryptographic encryption strategy with two type of algorithms. The algorithm for key encryption, which only encrypts the session key currently in use once, and the algorithm for data encryption, which encrypts many messages using the encrypted session key. Without any additional processing or communication overheads, this system delivers secrecy services to devices and users. To provide flexible public key management, the stateful IBE approach is used. [22] does not, however, enable prior verification of the sender or receiver. This approach also necessitates more Diffie-Hellman (DH) key operations algorithm, particularly on the node that sends the message [23], and this is subject to the chosen plaintext attack [24]. Usman et al. [25] present a Secure IoT (SIT) algorithm with 64-bit size (2017) based on the combination of a Feistel and a uniform substitution permutation network (SPN). The SIT uses round of five encryptions on an 8-bit microcontroller, resulting in simpler code, lower memory usage, and shorter encryption and decryption cycles.

The literature reviewed of related works show the need for a secure system in health-care sector for the proper monitoring of patient data and information. This will also give a sense of belonging to patients that are using the IoT-based system for their wellbeing.

The speck encryption can greatly increase the security and privacy of healthcare data in IoT-based system.

### 3 Methodology

Speck belongs to a family of lightweight symmetric block ciphers having a pair of varying block and key sizes. A block cipher is an encryption algorithm that enables users to have a common key secretly and securely encrypt/decrypt blocks of data. Lightweight encryption is built for efficient implementation of extremely constrained platforms, such as RFID tags, smart sensors, microcontrollers, and other resource-limited devices.

The term “lightweight” does not mean how secured the algorithm is, but rather refers to its suitability for use on highly constrained devices. Thus, a secure lightweight block cipher having a given block and key size pairs offers the equivalent level of security as any other secure block cipher with that same block and key size.

Lightweight encryption addresses the request to secure resource constraint devices which the conventional encryptions could not adequately attend to, using algorithms and protocols designed to perform well on platforms for devices with resources constrained.

Speck has support for a block of various sizes such as 32, 48, 64, 96, and 128 bits, and up to three different key sizes to go along with each size of a block. Speck family has ten different algorithms with various block and key sizes, as shown in the Table 1. All values are in bits with  $n$  as the word size and  $M$  as the number of words.

**Table 1.** The speck parameters

Block size	Key sizes	$n$	$M$
32	64	16	4
48	72	24	3
48	96	24	4
64	96	32	3
<b>64</b>	<b>128</b>	<b>32</b>	<b>4</b>
96	96	48	2
96	144	48	3
128	128	64	2
128	192	64	3
128	256	64	4

Speck has operations that are highly efficient for software platforms of varying requirements. There are a quite few numbers of the operation such as NOT, OR, AND, XOR, rotations, modular addition, and subtraction use to achieve nonlinearity as against S-Boxes used in conventional encryptions. Since Speck is not using S-Boxes, then it is not Substitution-Permutation Networks (SPNs). Rather than being SPNs, Speck is an

Add–Rotate–XOR (ARX) cipher with Feistel Structure round functions, which provides an adequate equilibrium between operations of nonlinear confusion and linear diffusion. Using the bit permutation rotation operation may not achieve sufficient diffusion, but with additional functional rounds, an adequate level of security is achieved like in the S-Boxes permutation of SPN. Speck nonlinearity is achieved with modular addition. Thus, its functions are secured cryptographically and effectively suitable for IoT constraint devices software implementations.

### 3.1 The Proposed Model

The notations used in the definition of the proposed model is given in the table below.

n	A word size
2n	A block size
M	Number of words
R	Number of rounds
PT	2n-bit input plaintext
CT	2n-bit output ciphertext
K <sub>i</sub>	n-bit round subkey for round i
K	mn-bit Master key from which round subkeys are generated
⊕	Bitwise exclusive OR operation
S <sub>j</sub>	Left cyclic shift by n bits
S <sub>-j</sub>	Right cyclic shift by n bits
+	Mod 2n addition operation

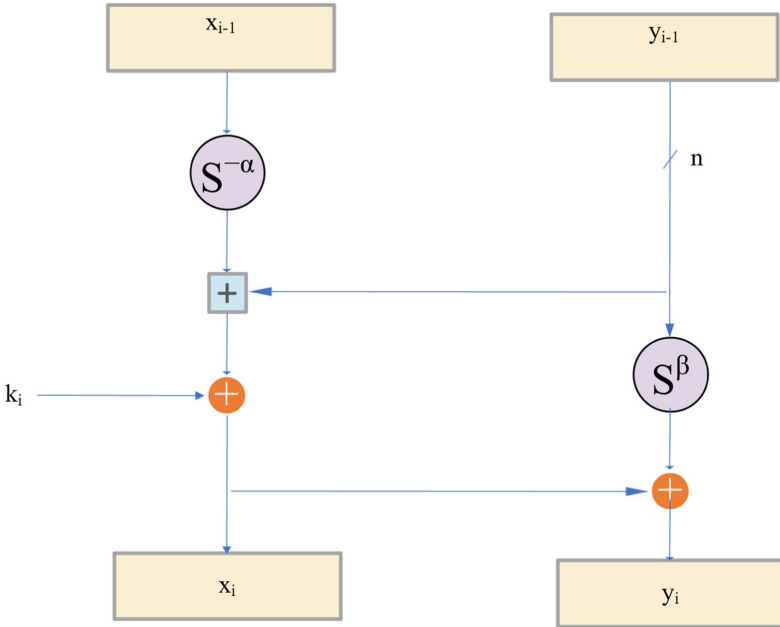
Speck block cipher denoted as Speck2n has an n-bit word, where n represents either 16, 24, 32, 48, or 64. Speck2n having an m-word (mnbit) key is denoted as Speck2n/mn. For instance, Speck64/96 denotes the Speck version having plaintext blocks of 64-bit and with a key having 96-bit (n = 32 and m = 3).

The round function of Speck2n encryption is denoted by the mapping.

$$R_k(x, y) = ((S^{-\alpha}x + y) \oplus k, S^{\beta}y \oplus (S^{-\alpha}x + y) \oplus k), \text{ as shown in Fig. 1.}$$

where x and y are n-bit halves of 2n-bit plaintext, and k is round key.

$x = \text{RCS}(x, \alpha)$	Right shift x by $\alpha$ and assign the result to x
$x = x + y$	modulo 2n addition of x and y and assign the result to x
$x = x \oplus k$	XOR x and round key k and assign the result to x
$y = \text{LCS}(y, \beta)$	Left shifty by $\beta$ and assign the result to y
$y = y \oplus x$	XOR y and x and assign the result to y



**Fig. 1.** Speck round function

For decryption, the inverse of the round function is used with modular subtraction instead of modular addition and is given by

$$R_k^{-1}(x, y) = (S^{\alpha}((x \oplus k) - S^{-\beta}(x \oplus y)), S^{-\beta}(x \oplus y))$$

The parameters  $\alpha$  and  $\beta$  are 7 and 2 respectively, for Speck32/64 and they are 8 and 3 for other variance of speck.

Figure 2 presents the speck encryption flow of the proposed model.

**Key Schedule**

Speck uses key schedules of 2-, 3-, and 4-word depending on Speck variance. Key schedules for Specks use around function, as given below.

Suppose  $m$  is the number of words for a key, and the key  $K$  can be written as  $(l_{m-2}, \dots, l_0, k_0)$ .

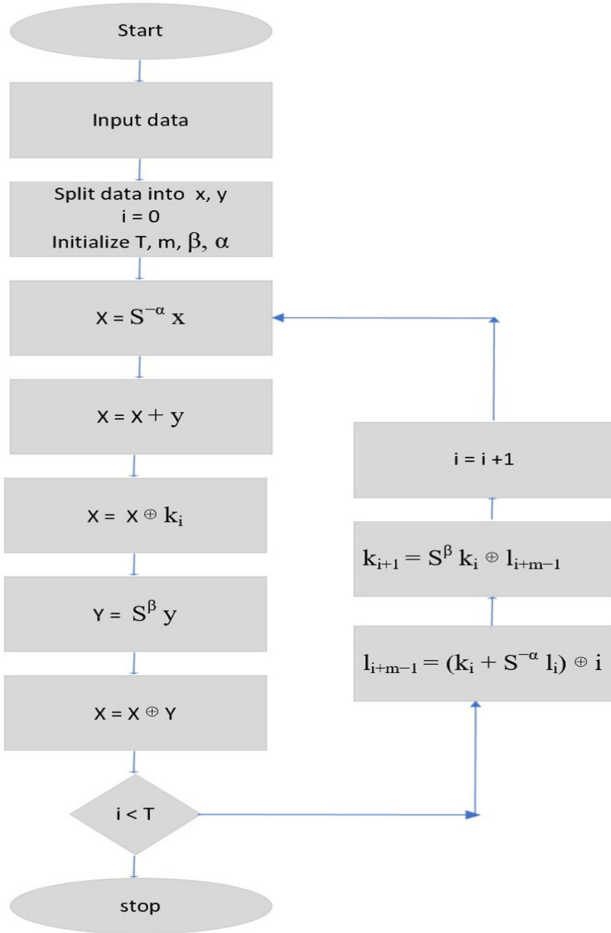
Then, to generate sequences of two words  $k_i$  and  $l_i$  by

$$l_{i+m-1} = (k_i + S^{-\alpha}l_i) \oplus i$$

and

$$k_{i+1} = S^{\beta}k_i \oplus l_{i+m-1}$$

The value  $k_i$  is the  $i$ th round key, for  $i \geq 0$



**Fig. 2.** The speck encryption flow

The Table 2 below shows the Speck rounds and the required parameters

The Speck key schedules receive an input key  $K$  and produce from it a sequence of  $T$  key words  $k_0, \dots, k_{T-1}$ , where  $T$  is the number of rounds. Encryption is the composition  $R_{k_{T-1}} \circ \dots \circ R_{k_1} \circ R_{k_0}$ , read from right to left.

Algorithm 1 below presents the pseudo-code of the SPECK encryption procedure. The key scheduling function forms the encryption key  $K = l_{m-2}, \dots, l_0, k_0$  and generates the  $T$ -rounds keys sequence  $k_0, \dots, k_{T-1}$  using the procedures presented by Algorithm 2.



**Table 2.** The speck rounds

Block size 2n	n	M	key size mn	Speck rounds	$\alpha$	B
32	16	4	64	22	7	2
48	24	3	72	22	8	3
48	24	4	96	23	8	3
64	32	3	96	26	8	3
64	32	4	128	27	8	3
96	48	2	96	28	8	3
96	48	3	144	29	8	3
128	64	2	128	32	8	3
128	64	3	192	33	8	3
128	64	4	256	34	8	3

**Algorithm 1: Speck Encryption**

input: plaintext, K encryption key

output: ciphertext

split plaintext into n-bit x, y

initialize T,  $\beta$ ,  $\alpha$ generate round keys  $k_0, \dots, k_{T-1}$ 

for i = 0 to T-1

$$x = (S^{-\alpha} x + y) \oplus k_i$$

$$y = S^{\beta} y \oplus x$$

end for loop

**Algorithm 2: Round Key generation**

Input: Encryption key

Output: sequence of keys

initialize m;

generate  $l_{m-2}, \dots, l_0, k_0$ 

for I = 0 to T-2

$$l_{i+m-1} = (k_i + S^{-\alpha} l_i) \oplus i$$

$$k_{i+1} = S^{\beta} k_i \oplus l_{i+m-1}$$

end for loop

## 4 Results and Discussion

The proposed encryption method was developed on the microcontroller (MCU) AVR 8-bit RISC (Reduced Instruction Set Computing) architecture that has programmable

on-chip flash memory, SRAM, IO storage space, and EEPROM, as shown in Table 3. Different block size data was processed for encryption in KB and our experimental findings are seen in Table 2. The execution time tests the amount of encryption and decryption block cycles used. This is the contrast between the beginning and the end of the process.

**Table 3.** Feature of the device

Name	Value
MCU	ATmega328PB
Program memory type	Flash
Program memory size (KB)	32
CPU speed (MIPS/DMIPS)	20
SRAM (B)	2,048
Data EEPROM/HEF (bytes)	1024
Digital communication peripherals	2-UART, 2-SPI, 2-I2C
Capture/compare/PWM peripherals	3 Input Capture, 3 CCP, 10PWM
Timers	2 x 8-bit
Number of comparators	1
Operating voltage range (V)	1.8 to 5.5
Pin count	32
Low power	Yes

**Table 4.** Performance analysis of the proposed model

File size (kilobytes)	Encryption time (mm)	Decryption time (mm)
0.82	0.121	0.120
1.65	0.216	0.215
12.32	0.893	0.891
36.50	2.014	2.012
50.2	3.142	3.138
100.7	5.461	5.459

Other block cipher techniques such as AES, DES, SIMON were compared to evaluate the analysis of memory use and speed. Table 4 presents the findings of the performance of the proposed system using encryption time and decryption time, from table 4 the results show that the high the file size (KB) the high the encryption and decryption time. The

findings as shown in Table 5 indicate that the memory use of AES, DES, and SIMON is greater than the proposed memory usage of the proposed system. In comparison, the time to encrypt and decrypt is less in the proposed system than in most of the lightweight ciphers.

**Table 5.** Comparison results of our proposed model with another model

Algorithm	Speed (Clockcycle /bytes)	Memory usage (bytes)	Encryption time (ms)	Decryption Time (ms)
AES	24695	1709	650	124
DES	28401	1608	797	281
SIMON	39313	755	700	300
SPECK	30957	1364	619	236

Table 5 uses various algorithms to display the different encryption times of the same input data. This illustrates that the suggested encryption solution takes the same amount of time as other regular block ciphers.

The proposed system was still tested using the EEG datasets taken from Bonn database a widely used dataset [30]. The dataset contains five various datasets labels as A, B, C, D and E. The dataset A, D, and E was analysed to measure the latency in both cloud and cloud computing using computational time and transmission delay. Dataset A was taken from a stable individual, while dataset D was taken during an interracial situation. During seizure activity, dataset E was collected from the epileptogenic region (ictal state). Each dataset contains 100 EEG epochs, each of which contains 4097 samples. The EEG data was collected at a sampling rate of 173.61 Hz. Tables 6 and 7 demonstrate the statistical parameters that were derived from the sub-bands [31]. The values of all statistical parameters are clearly higher for dataset E. The derived attribute estimates for datasets A and D are extremely similar. Dataset A, D, and E versus Dataset E were used to assess identification in this study.

On an Intel Celeron processor, for example, each cycle requires about 270 mA on average. For example, an encryption of 20,000 cycles would burn 7.7 J on a 700 MHz CPU operating at 1.35 V. As a result, program P’s energy consumption to achieve its aim (encryption or decryption) is provided by:

$$E = V_{cc} * I * N * \tau \tag{1}$$

Both  $V_{cc}$  and  $I$  are fixed for a given piece of hardware,

$$E \propto I \times N. \tag{2}$$

However, at the application level, talking about  $T$  rather than  $N$  is more useful, thus we write energy as  $E \propto I \times T$ . The duration time for encryption or decryption is replaced by the total number of clock cycles divided by the clock frequency. The amount of energy used by program P to achieve its purpose (encryption) is then calculated as follows:

$$\therefore E_{cost} = V_{cc} \times I \times T \text{Joules} \tag{3}$$

Where  $V_{cc}$  is the system’s supply voltage and  $I$  is the average current consumed from the power source in amperes. The number of clock cycles is denoted by the letter  $N$ .  $\tau T$  the clock period.  $T - N/\text{processor’s speed}$  (seconds).

**Table 6.** Coefficients of extracted features for dataset A

Coefficient	Variance	Standard deviation	Energy
D1	25.2164	5.0216	2.8564e + 04
D2	587.553	24.2395	3.0435e + 05
D3	5.3957e + 03	73.4555	1.4426e + 06
D4	9.9058e + 03	99.5279	1.9874e + 06
A4	1.5439e + 04	124.2539	4.0502e + 06

**Table 7.** Coefficients of extracted features for dataset E

Coefficient	Variance	Standard deviation	Energy
D1	1.4426e + 03	37.9819	1.8934e + 06
D2	6.4382e + 04	253.736	4.8707e + 07
D3	7.0151e + 05	837.560	3.0676e + 08
D4	6.9684e + 05	834.769	1.887e + 08
A4	1.7177e + 06	1.310e + 03	4.0854e + 08

## 5 Conclusion and Future Research Directions

The IoT-based system is an upcoming information and communication science technology. A large number of IoT technologies have been developed in the healthcare field to remotely detect, track, predict, and manage chronic diseases. The data generated by these devices is huge and the processing and transmissions need typical infrastructure to cope with them. Cloud computing can be applied to overcome the challenges of IoT-based systems. The cloud computing technology with lower latency, reduced computational time, and scalability will help IoT-based system devices in real-time data collection and processing. The deployment of cloud data is far away from the network; this causes the response time delay in real-time. Moreover, the cloud may cause significant overhead on the backbone network to the user application due to the huge amount of big data sent to the cloud. Hence, the application of cloud computing will bring the storage resources and computational closer to the end-user devices, thus reducing the burden on the cloud. The information transmitted by the smart sensor in smart healthcare systems (SHS) is personal, requiring secrecy, trustworthiness, and usability for healthcare practitioners to take timely and accurate decisions, so the data required to be transmitted and stored

safely. For the IoT based Smart Healthcare system, a lightweight ciphering technique was implemented. The proposed approach is based on basic operations of ARX Addition, Rotation XORing, swapping, slicing, among others. The outcome of the implementation reveals that the memory occupation is limited while the speed for producing a successful cipher is important. The results reveal that the longer the encryption and decryption time, the larger the file size (KB). The results show that the memory usage of AES, DES, and SIMON is higher than the proposed system's memory usage. In different constrained devices, the proposed technique can be applied and tested. It is also possible to perform differential and linear crypto-analysis of this algorithm in the future to ensure the cipher's robustness.

## References

1. Awotunde, J.B., Adeniyi, A.E., Ogundokun, R.O., Ajamu, G.J., Adebayo, P.O.: MIoT-based big data analytics architecture, opportunities and challenges for enhanced telemedicine systems. *Stud. Fuzziness Soft Comput.* **2021**(410), 199–220 (2021)
2. Maskeliūnas, R., Damaševičius, R., Segal, S.: A review of internet of things technologies for ambient assisted living environments. *Future Internet* **11**(12), 259 (2019)
3. Abiodun, M.K., Awotunde, J.B., Ogundokun, R.O., Adeniyi, E.A., Arowolo, M.O.: Security and information assurance for IoT-based big data. *Stud. Computat. Intell.* **2021**(972), 189–211 (2021)
4. Azeez, N.A., Van der Vyver, C.: Security and privacy issues in e-health cloud-based system: a comprehensive content analysis. *Egypt. Inf. J.* **20**(2), 97–108 (2019)
5. Abikoye, O.C., Ojo, U.A., Awotunde, J.B., Ogundokun, R.O.: A safe and secured iris template using steganography and cryptography. *Multimedia Tools Appl.* **79**(31–32), 23483–23506 (2020)
6. Ogundokun, R.O., Awotunde, J.B., Adeniyi, E.A., Ayo, F.E.: Crypto-Stegno based model for securing medical information on IOMT platform. *Multimedia Tools Appl.* 1–23 (2021)
7. Tabrizchi, H., Rafsanjani, M.K.: A survey on security challenges in cloud computing: issues, threats, and solutions. *J. Supercomput.* **76**(12), 9493–9532 (2020). <https://doi.org/10.1007/s11227-020-03213-1>
8. Awotunde, J.B., Chakraborty, C., Adeniyi, E.A., Abiodun, K.M.: Intrusion detection in industrial internet of things network based on deep learning model with rule-based feature selection. *Wirel. Commun. Mob. Comput.* **2021**, 1–17 (2021)
9. Thabit, F., Alhomdy, S., Jagtap, S.: A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions. *Int. J. Intell. Netw.* **2**, 18–33 (2021)
10. Abdurraheem, M., Awotunde, J.B., Jimoh, R.G., Oladipo, I.D.: An efficient lightweight cryptographic algorithm for IoT security. *Commun. Comput. Inf. Sci.* **2021**(1350), 444–456 (2021)
11. Mohammed, K.M.A.K.: Confidentiality of data in public cloud storage using hybrid encryption algorithms. Doctoral Dissertation, Sudan University of Science and Technology
12. Singh, P., Acharya, B., Chaurasiya, R.K.: Lightweight cryptographic algorithms for resource-constrained IoT devices and sensor networks. In: *Security and Privacy Issues in IoT Devices and Sensor Networks*, pp. 153–185. Academic Press
13. Makarenko, I., Semushin, S., Suhai, S., Kazmi, S.A., Oracevic, A., Hussain, R.: A comparative analysis of cryptographic algorithms in the internet of things. In: *2020 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC)*, pp. 1–8. IEEE, Oct 2020

14. Nayancy, Dutta, S., Chakraborty, S.: A survey on implementation of lightweight block ciphers for resource constraints devices. *J. Discrete Math. Sci. Cryptogr.* 1–22 (2020)
15. Saddam, M.J., Ibrahim, A.A., Mohammed, A.H.: A lightweight image encryption and blow-fish decryption for the secure internet of things. In: 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1–5. IEEE, Oct 2020
16. Dinu, D., Le Corre, Y., Khovratovich, D., Perrin, L., Großschädl, J., Biryukov, A.: Triathlon of lightweight block ciphers for the internet of things. *J. Cryptogr. Eng.* **9**(3), 283–302 (2019)
17. Turan, M.S., McKay, K.A., Çalik, Ç., Chang, D., Bassham, L.: Status report on the first round of the NIST lightweight cryptography standardization process. National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Rep. (NISTIR) (2019)
18. Kraft, J.S., Washington, L.C.: *An Introduction to Number Theory with Cryptography*. Chapman and Hall/CRC (2018)
19. Das, A.K., Wazid, M., Yannam, A.R., Rodrigues, J.J., Park, Y.: Provably secure ECC-based device access control and key agreement protocol for IoT environment. *IEEE Access* **7**, 55382–55397 (2019)
20. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) *CRYPTO 1984*. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). [https://doi.org/10.1007/3-540-39568-7\\_5](https://doi.org/10.1007/3-540-39568-7_5)
21. Fischer, M., Scheerhorn, A., Tönjes, R.: Using attribute-based encryption on IoT devices with instant key revocation. In: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 126–131. IEEE, Mar 2019
22. Al Salami, S., Baek, J., Salah, K., Damiani, E.: Lightweight encryption for smart home. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 382–388. IEEE, Aug 2016
23. Naoui, S., Elhdhili, M.E., Saidane, L.A.: Lightweight enhanced collaborative key management scheme for smart home application. In: 2017 International Conference on High Performance Computing Simulation (HPCS), pp. 777–784. IEEE, July 2017
24. Syal, R.: A comparative analysis of lightweight cryptographic protocols for smart home. In: Shetty, N.R., Patnaik, L.M., Nagaraj, H.C., Hamsavath, P.N., Nalini, N. (eds.) *Emerging Research in Computing, Information, Communication and Applications*. AISC, vol. 882, pp. 663–669. Springer, Singapore (2019). [https://doi.org/10.1007/978-981-13-5953-8\\_54](https://doi.org/10.1007/978-981-13-5953-8_54)
25. Awotunde, J.B., Jimoh, R.G., Folorunso, S.O., Adeniyi, E.A., Abiodun, K.M., Banjo, O.O.: Privacy and security concerns in IoT-based healthcare systems. *Internet Things*, 105–134 (2021)