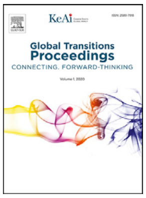
Global Transitions Proceedings 2 (2021) 91–99

Contents lists available at ScienceDirect

Global Transitions Proceedings

journal homepage: www.elsevier.com/locate/gltp



A new lightweight cryptographic algorithm for enhancing data security in cloud computing

Fursan Thabit[a,*], Associate Prof Sharaf Alhomdy[b], Abdulrazzaq H.A. Al-Ahdal[c], Prof Dr Sudhir Jagtap[a]

[a] *School of Computational Sciences, S.R.T.M. University of organization, Nanded, India*
[b] *Faculty of Computer and Information Technology (FCIT), Sana'a University, Yemen*
[c] *Faculty Computer Science & Engineering, Hodeidah University, Yemen*

ARTICLE INFO

Keywords:
cloud computing security
cryptography
feistel
lightweight algorithm

ABSTRACT

Data has been pivotal to all facets of human life in the last decades. In recent years, the massive growth of data as a result of the development of various applications. This data needs to be secured and stored in secure sites. Cloud computing is the technology can be used to store those massive amounts of data. . The rapid development of this technology makes it more critical. Therefore, it has become urgent to secure data from attackers to preserve its integrity, confidentiality, protection, privacy and procedures required for handling it. This paper proposed a New Lightweight Cryptographic Algorithm for Enhancing Data Security that can be used to secure applications on cloud computing. The algorithm is a 16 bytes (128-bit) block cipher and wants 16 bytes (128-bit) key to encrypt the data. It is inspired by feistal and substitution permutation architectural methods to improve the complexity of the encryption. The algorithm achieves Shannon's theory of diffusion and confusion by the involvement of logical operations, such as (XOR, XNOR, shifting, swapping). It also features flexibility in the length of the secret key and the number of turns. The experimental results of the proposed algorithm presented a strong security level and an apparent enhancement in measures of cipher execution time and security forces compared to the cryptographic systems widely used in cloud computing.

1. Introduction

The use of distributed computing systems and technologies has evolved dramatically in recent years. A large number of distributed network models, architectures and infrastructures, such as network, Pervasive, Autonomic, cloud, etc. have been created by this increase. Cloud computing refers to a computing network, typically linked via the Internet, which share a decentralized amount of services offered by a to access the user needs [1]. The private publication NIST concept of cloud computational is a framework for providing a common set of configurable computing resources (such as software) with omnipresent, fast, and on-demand network access. The benefits of cloud computing include manageability, scalability, and affordability. Additionally, cloud storage has infrastructure characteristics on demand, economy, universality, ease, leasing pluralism, reliability, and versatility.

A cloud client may use these tools on request to build, run and host services and applications that can be adapted anywhere, and on any device [2,3]. The NIST "concept emphasizes the three service models – Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS) which presenting cloud services to the client through a Cloud Service Provider (CSP) customers. It also briefs the four deployment models — Public Cloud and Hybrid Cloud, Private Cloud, Community Cloud, emphasizing the paradigm for distributing computing assets to coordinate the cloud administration.

On the other hand, cloud infrastructure faces a range of problems primarily related to interoperability, scalability, and multi-tenancy. But the most critical issues apply to protection as cloud infrastructure is exposed to a variety of threats as a system utilizing internet networks (such as embedded networks, grid computing, etc.)" [4]. Cloud computing security problems will hold up its widespread acceptance. In reality, the sharing of cloud computing services poses the difficulty of keeping these services secure and secured against unauthorised access or usage. Mostly the data outsourced to the cloud faces this challenge. Network security is one of cloud computing's key security issues, which relate to external and internal attacks [5,6]. Provide a safe link between cloud providers and users, and several systems and protocols have been developed to ensure data transfer security over networking, where cryptography is the most effective. The translation of plaintext into cipher-text is involved in cryptography. Generally, it is a tool used to securely transfer contents by ensuring that they can be found only by the intended recipient [7].

[*] Corresponding author.
E-mail address: thabitfursan@gmail.com (F. Thabit).

https://doi.org/10.1016/j.gltp.2021.01.013

Available online 23 January 2021

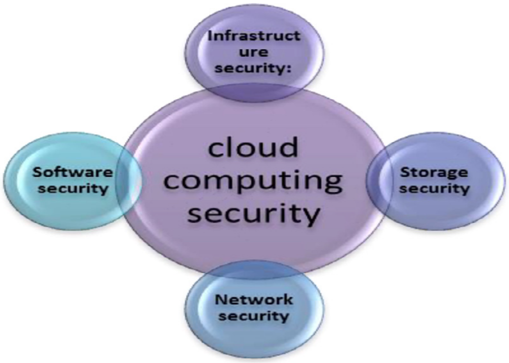


Fig. 1. Major cloud security.

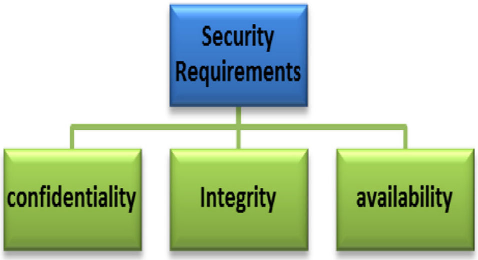


Fig. 2. Basic cloud security requirement.

In this paper, a new lightweight cryptographic algorithm has been proposed. It called a New Lightweight Cryptographic Algorithm (NLCA) for enhancing data security in a cloud computing environment. It encrypts data based on symmetric cryptography.

The rest of this paper is organized as follows. Section 2 provided a simple understanding of the security of cloud computing. Related work is given in Section 3. The robust encryption algorithm prerequisites are given in Section 4. Section 5 describes the proposed algorithm. The Implementation and results discussion is given in Section 6. Finally, the paper is concluded with further recommendations in Section 7.

2. Cloud computing security overview

The cloud security incorporates all the methodologies intending to safeguard, re-establish, and ensure the security of data in computer systems frameworks against various threats. The ElasticaQ2 2015 and CSA further look at how to create a viable cloud application security engineering, which gives control, deceivability, and remediation. Also, according to (ISO), describes Data Security concerns which can likewise be guided in regards to the cloud computing essential security requirements for a successful and secure innovation technology solution. Fig 1. Shows the significant security of cloud computing. In a cloud world, the truth of outsourcing data and software assigns their power to the service provider from the direct control of the user. As a result, Trust relies on the implementation model and the vendor of the cloud.

2.1. Key security requirements in cloud computing

According to NIST, confidentiality, availability and integrity are a fundamental necessity for cloud computing, just as any other information technology management system [2]. Further authorization, authentication, accountability and privacy as cloud protection criteria are introduced, as seen in Fig 2.

- **Confidentiality** means maintaining the information of customers and allowing unique individuals exclusively for access to data.
- **Integrity** means a commitment to ensure that information is not modulated or changed as it is processed or transmitted and that the approach of consumers to change, amend, repeat or remove information is merely allowed.
- **Authentication** involves ensuring the customer's integrity when providing access to details and this can be achievable with the use of such insurances on their accounts.
- **Availability** means that the information listed by the customer or the services it needs is continuously accessible wherever and wherever it is needed.
- **Authorization** means to guarantee that the options to get to it are allocated for clients who have submitted special details.

3. Related work

This section consists of two sub-sections; one presents cloud computing security-related work, and the other presents the available lightweight cryptographic systems related work.

3.1. Cloud computing security

There are many studies that have addressed the security issues of cloud computing; however, this section provides some recent studies which examined cloud computing cryptography. These are an advanced encryption algorithm (AES), international data encryption algorithm (IDEA), and data encryption standard (DES), it categorises common cryptographic algorithms. Comparison of symmetric algorithms (e.g. AES, DES, and 3DES) and asymmetric algorithms (e.g. RSA). The algorithms implemented based on Block Size, Protection Rate, Key Length, Rounds, and Execution Time was introduced. The results are more efficient in the cloud environment [8]. "The author Introduced the use of hybrid encryption approaches to enhance the privacy of the information stored on cloud servers such: Blowfish, RSA, AES, Eclipse IDA, DSA, This study focuses on not using third parties to encrypt customer data, but on allowing consumers the authority to decide ways of encrypting their data" [9]. A cloud computing model based on data classification at too has been proposed to decrease latency and processing time. The data were classified into three layers [10]. "The study suggested the use of hybrid encryption approaches to maximize the safety of cloud computing data such as: RSA Digital Signature, RSA algorithm, Blowfish algorithm encryption/decryption, Feistel, and XOR operating algorithms. and The Symmetric Data Encryption Standard (DES) were described by" [11]. Besides, demonstrated how to fuse two separate algorithms, such as DES and RSA, to remove Cloud Storage's security challenges. "The authors identified previous studies devoted to cloud data security and performed a survey. They suggested a hybrid protection encryption approach using Blowfish and MD5 to offer improved security on the cloud server" [12]. Some of common cloud service providers, such as Google (Google Drive), and Microsoft (Azure and One Drive) were studied by [14]. Moreover, they studied cryptographic algorithms mainly used in cloud computing: modern cryptography, searchable encryption, homomorphic encryption, and attribute-based encryption (e.g. DES, 3DES, AES, RC6, and BLOWFISH), as a fusion of two or more cryptographic techniques, they have also created a hybrid encryption concept to take advantage of the potential of each system to protect cloud data. A comparison of IDAs, SHA-512, 3DES, and AES-256 made by [13]. It consists of on premise data encoding and decoding. This algorithm achieves a much higher level of protection and performance for large and small data files [14]. The author investigated the classification of algorithms by algorithms commonly classified as Signature Algorithms, Symmetric Algorithms, Asymmetric Algorithms, and Hash Algorithms.

3.2. Lightweight cryptographic systems

There are many studied on symmetrical cryptographic algorithms for a lightweight were built and develop for suitable applications, such as L Block, LED. HIGHT, PRESENT, DESL, CLEFIA, TWINE, RECTANGLE, SIT etc. "A lightweight 64-bit block size cryptosystem with 128-bit key was created, iterated in 32 rounds, and carried two types of operations; XOR operation paired with left or right rotations. Its focus was to deploy hardware on ubiquitous devices, such as wireless sensor nodes and RFID tags, with about the same chip size as AES but running much faster" [15].

In [15] "the authors described a symmetric block cipher named CLEFIA-128, developed by Sony and built to be suitable for both software and hardware, 128-bit block size encryption in 128-bit key length, and 28 Feistel structure rounds. "The researcher constructed two different types of Data Encryption Lightweight systems, i.e. DESXL. DESL y DESL, on the other hand, in DESXL, instead of separate ones with no initial and final permutations, a single S-Box u is used to improve protection by using a 184-bit key. No attack was displayed against DESL and DESXL, as they said" [16]. The study identified the generalized multi-platform cryptosystem Feistel structure termed "TWINE." It is 64-bit block size running 36 rounds of either 80 or 128-bit key length, and each round contains a 4-bit S-box and 4-bit block permutation layer, nonlinear substitution layer [17].

In [18] stated the "RECTANGLE" cryptosystem optimized for a block size of 64 bits with a key length of either 80 or 128 bits, running just 25 rounds. In the [19] A Stable IoT (SIT) light-weight encryption algorithm. It is a 64-bit block cipher and the data has to be encrypted using a 64-bit address. A mixture of Feistel and a uniform substitution-permutation network is the architecture of the algorithm. In [20] this article, lightweight encryption algorithm which consists of a 64-bit block cipher and 80 bit key to encrypt data in IoT devices.

4. Powerful encryption algorithm prerequisites

Any encryption algorithm must satisfy some set of specifications to provide high security. The following criteria have been defined, based on the current literature that should be fulfilled by the newly developed algorithm:

- Encryption of the entire character set
- Encrypting each plain-text character into a special sequence
- There should be strong encoding
- The encryption approach should be complex

5. The proposed algorithm

To improve cloud computing protection with low processing, and high performance, a New Lightweight Cryptographic Algorithm (NLCA) for enhancing data security in cloud computing environment is proposed.

The algorithm is simple and highly secure encryption/decryption. The proposed algorithm gives an easy structure effective for the cloud environment. There are Some well-known popular ciphers including "Camelia [21], SF [22], Blowfish [23] and DES [24] use the Feistel structure". The major benefit of applying Feistel architecture is that the encryption and decryption operations are almost similar.

Also other popular block cipher including "3-Way [25], Grasshopper [26], AES (Rijndael) [27], PRESENT [28], SAFER [29] Use the Network SP" (Substitution-Permutation). The confusion is confronted with too many overlapping rounds of substitution and transposition and diffusion of Shannon's Characteristics that ensure the cipher text being altered in a Pseudo by a random manner. Therefore, the introduced algorithm is a symmetric key block cipher and the idea is inspired by a combination of Feistel and SP architectural methods to improve the complexity of the encryption.

The main idea of the **NLCA** is to use is a 16 bytes (128-bit) block cipher and want 16 bytes (128-bit) key for encrypt the data. The encryption process requires encryption rounds in a symmetric-key algorithm; each round always relies on mathematical functions to generate diffusion and confusion. Encryption algorithms are usually configured to take 10 to 20 rounds on average to keep the encryption process strong enough to meet device specifications. However, the proposed algorithm is limited to only five rounds to maximize energy efficiency results, as each round requires crypto mathematical operations involving 4 bits of data to work. It introduces mixed operations in multiple algebraic classes, including XOR and Addition operations, to generate difficulty for attackers. As follows the detailed steps of the procedures are described

Table 1
Notations.

Notation	Function
⊕	XOR
⊙	XNOR
‖,⧺	Concatenation

1) Key Generation Block.
2) Encryption Block.
3) Decryption Block.

In the following subsections, these blocks will be further clarified in detail, and some of the essential notes followed in the interpretation are presented in **Table 1**.

5.1. Key generation block

The key is the most important component of encryption and decryption methods. If this key is identified to an attacker, the confidentiality of the data is compromised. It is this key on which the whole integrity of the data rests. In order to generate confusion and diffusion, different operations are carried out to prevent the possibility of weak key as to strengthen key strength. The Feistel-based encryption algorithms depend on various rounds, requiring a different key for each round. The proposed algorithm has five rounds for encrypted/decrypted for such a reason, we need five unique keys. The algorithm is a block cipher of 16 bytes (128-bit) which allows a 16 bytes (128-bit) key to be taken from the user as an input, which is used as the input to the Generation Block key. Each block will produce five separate keys.

5.1.1. Key generation process

- The first step of key encryption, the128-bit cipher key ("Kc") is split into two segments 64-bits right and 64-bits left.
- In the Second step the 64-bits right and 64-bits left is split into the segments of 4-bits. After the split A shift row is made for every 4 bits, output of shifting to feed to f-function
- The f -function used 4 segments, each segment 4 bit (16 bit) as illustrated in Fig. 4. Substitution can generate for cipher key (Kc) by f-function as explain in Eq. (1).

$$Kb_i f = \Big\|_{j=1}^{5} K_{c4(j-\ 1)+i} \tag{1}$$

Where $\boldsymbol{i = 5}$;
$Ka_i f$ is outpu**t from Eq. (2)**

$$Ka_i f = f\left(b_i f\right) \tag{2}$$

f :Function. Linear and non-linear confusion and diffusion transformations composed of P and Q tables are seen in Tables 2 and 3, as seen in Fig. 3.

- Output, every f-function is created from (16 bit) one matrix 4 * 4. Eight called arrays are therefore created key matrix ('M1', 'M2', 'M3', 'M4', 'M5' 'M6', 'M7', and 'M8').
- The arrays generated ("K1, K2, K3, K4, K5, k6, K7, K8") are the round keys.
- The result of the rotation of the arrays is KM1, KM2, KM3, KM4, KM5 KM6, KM7, and KM8 respectively the generated key combination every two key to get the four public key **(kk1=k1+k2, kk2=k3+k4,**

Table 2
P table.

Kci	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F	3	F	E	0	5	4	B	C	D	A	9	6	7	8	2	1

Table 3
Q table.

Kci	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Q(Kci)	9	E	5	6	A	2	3	C	F	0	4	D	7	B	1	8

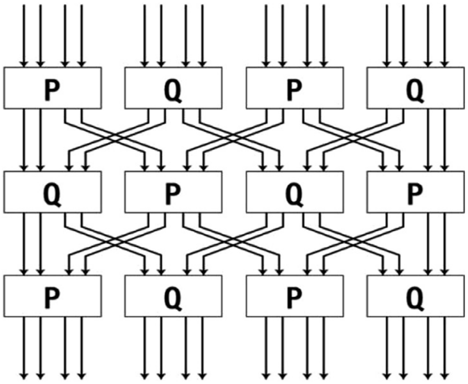


Fig. 3. F-Function of SIT algorithm.

kk3=k5+k6, kk4=k7+k8, Among the four round keys, an XOR operation is performed to obtain the fifth key, as seen in Eq. (3).

$$kkk = \bigoplus_{i=1}^{5} k_i \tag{3}$$

Moreover, in this process two keys would be generated according to the seed enter by user. The first one, is send by user for the decryption process, as described in section 5.3. The other key will be completely different from the first key which used to encrypt data as described in section 5.2 and in **Fig 5**, and it will be discussed below:

The 128-bit cipher key (Kc) is split into R0 (64 bit) and L0 (64 bit).

For M = 1 to 64
If RM = 0 then——Cr-R [M] = 1, Cr-L [M] = 1
Else Cr-R [M] = 0, Cr-L [M] = 1

To combine **Cr-L, Cr-R** to key send (Send Key 128 bit)
To combine **R0, Cr-R to Kc** (encryption Key 128 bit)

5.2. Encryption process

The encryption process takes place after producing the sub-keys (KK1, KK2 … KKK) from the Key Generation Process It can start with the purpose of creating cryptography as in **Fig. 9**, the encryption process takes place. Easy processes, like, XOR, XNOR, AND, OR left shift (LS), swapping and substitution (S boxes) methods, are conducted throughout the encryption process to create confusion and diffusion. These operations increase complexity and create confusion for the attackers.

Encryption Process steps

The message to be encrypted is separated into blocks of plain-text (each referred to as M) of 16 bytes (128-bit) length as shown in Fig. 9. Then the following steps encrypt each block:

- The 128-bit block input is generally divided into four sub-blocks of 32 bits each, namely, **P1, P2, P3, P4** ($\mathbf{P_{0-31}}$, $\mathbf{P_{32-63}}$, $\mathbf{P_{64-95}}$, $\mathbf{P_{96-127}}$) This is to produce segments (Ro11, Ro12, Ro13, Ro14)
- (Initial state steps): Each sub-block is addressed using working key sub-keys (KK1, KK2 … KKK) by combining operations from various algebraic groups, which are AND, OR, XOR, XNOR operations. As shown in Fig 9.
- **Ro11** is the output of **XNOR** between $\mathbf{P_{0-31}}$ and **K1**, The product (Ro11) feeds

  F-Function to produce **EFL1.**

- **Ro14** is the output of **XNOR** between $\mathbf{P_{96-127}}$ and **K1**, The product (Ro14) feeds

  F-Function to produce **EFR1**.

As well as the F-Function seen in Eq. (1), the F.Function contains the activity of substitute (S boxes), AND, OR, and left shift (LS).

F = F1+F2; → 32 Bit
F1 = OR (S-boxes (AND (LS (16 bits/4) 16 Bit.
F2 = OR (S-boxes (AND (LS (16 bits/4)16 Bit.

The output from the F function is then XOR

- **Ro12** is the output of **XOR** between $\mathbf{P_{64-95}}$, and **EFL1**.
- **Ro13** is the output of **XOR** between $\mathbf{P_{32-63}}$, and **EFR1**.
- Process of switching takes place during the encryption process between the two internal halves. Then, the switches are between the parts (**Ro11, Ro12**) and (**Ro13, Ro14**).

All the previous processes are to increase the complexity of the coding as shown in Fig. 9. The same steps for the rounds are repeated by Eq. (4).

$$Ro_{i,j} = \begin{cases} Px_{i,j} \odot k_i \quad ; \quad j = 1..4 \\ Px_{i,j+1} \oplus Ef_{li}; \; j = 2 \\ Px_{i,j-1} \oplus Ef_{ri} \quad ; j = 3 \end{cases} \tag{4}$$

After that, the encoded text is obtained by Eq. (5).

$$Ct = R_{51}\#R_{52}\#R_{53}\#R_{54} \tag{5}$$

A *Transformation round*
Also every round in NLCA algorithm includes numerous transformation tasks that involve XOR, XNOR, F, functions and swapping. In Fig. 6, a single round of the NLCA algorithm is shown.

B *Swapping function*
This is the encryption portion in the left half 32 bits are swapped to the right position and right half of 32 bits are swapped to half-place left as shown in Fig. 7. The main goal of switching Role is to adjust the initial data location to get the data a more complicated cipher.

C *F Function:* This is a central component of encryption the algorithm that causes data diffusion. The S Boxes, is conducted processes of AND & OR and left shifting (LS) Operations of 16 data bits, as seen in Fig. 4.

D *Left Shifting (LS):* 16 bits of data are in the LS process divided into 4-bit blocks and left shifting Conducted on each block. Outcomes of this activity full data Bit Mixing. The mechanism shown in Fig. 7

E *Operations of XOR & XNOR:* Simple logical logic Operations that is appropriate to produce confusion and diffusion for results.

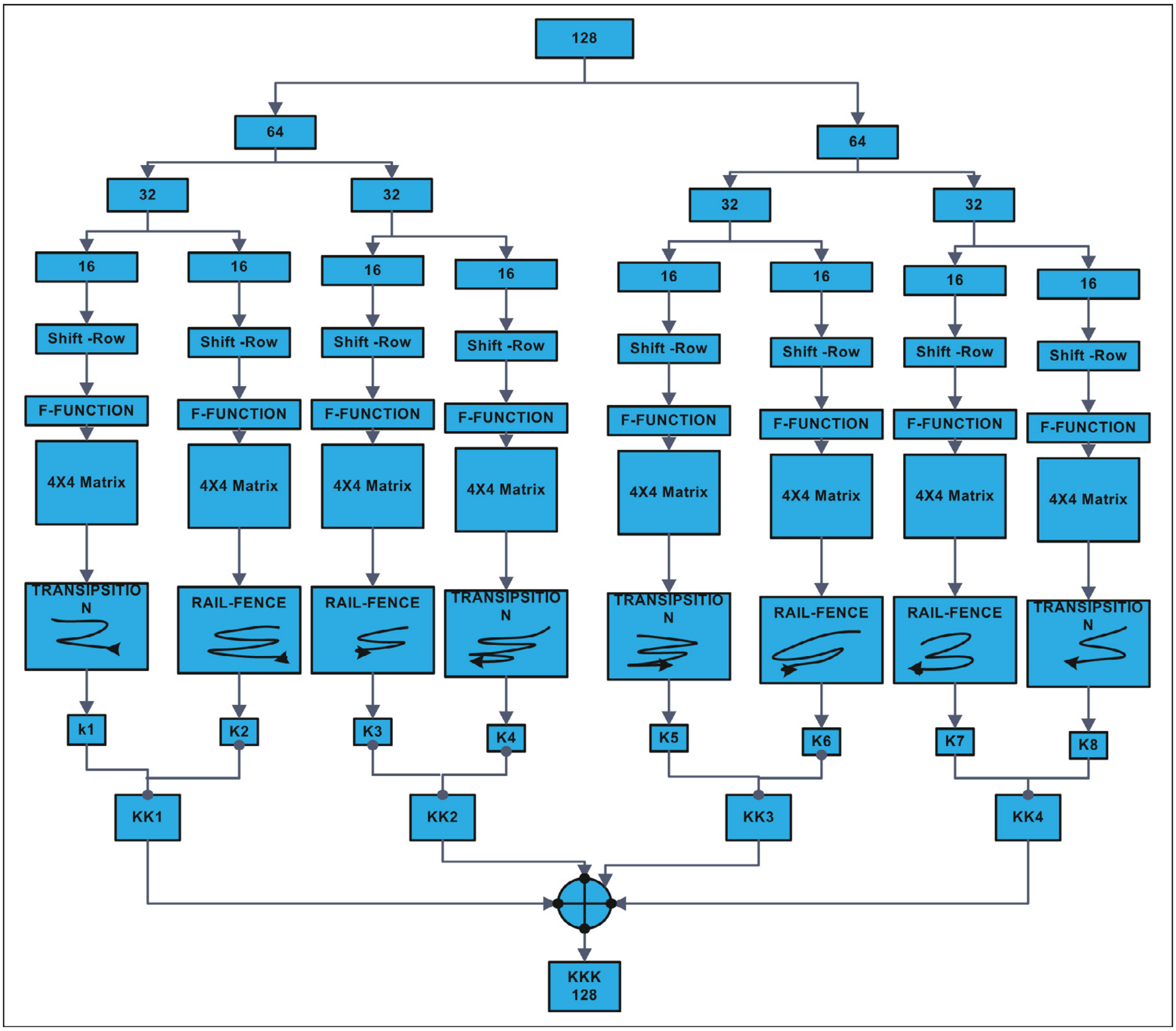


Fig. 4. Key generation process

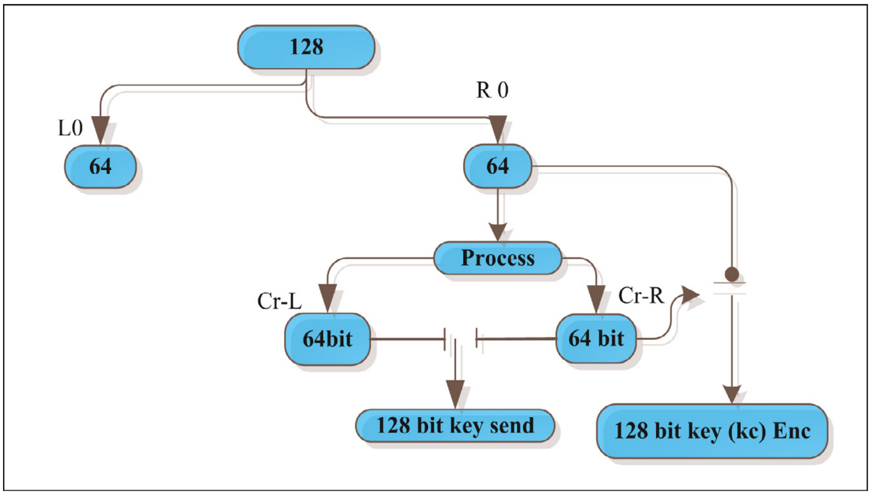


Fig. 5. Encryption Key.

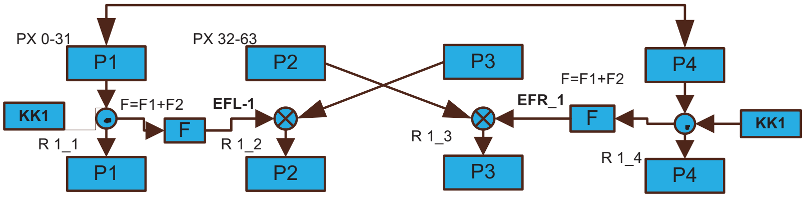


Fig. 6. Single round of encryption block.

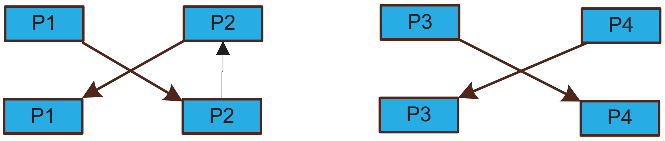


Fig. 7. Swapping process.

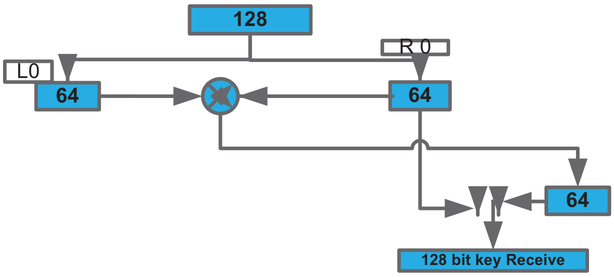


Fig. 8. Decryption key.

5.3. Decryption

In the decryption process, the encrypted key will be extracted from the key sent by the user. The process will be explained as in Fig 8. The 128-bit cipher key (send key), is divided into segment

R0 (46 bit)
L0 (64 bit).
K1 = R0⊕ L0
To combine R0 and K1 (decryption Key 128 bit).
Decryption process

The computational procedure used for the decryption of the Ci cipher-text block is exactly the same as the Mi block encryption method. The 128 -bit long Ci block is first split into 4 sub-blocks, and then handled with the same working keys using mixed XOR and Sub operations. And since they are the reverse of the encryption operation, the precise steps for the procedure would not need to be written.

6. Implementation and results discussion

This section described the implementation and results discussion for the proposed algorithm an d other algorithms.

6.1. Experimental environment

In order to analyze the efficiency of the proposed algorithm various kinds of symmetric cryptographic algorithms are used. The experimental setting consists of a cloud network consisting of the Xen Server hypervisor (6.1an Open stack middleware and a client that uses Citrix Desktop [30] to access the Xen-Server-hosted virtual machine). The cloud server Details as Core I7 (4.8 GHz) with 8 GB of RAM, and the client computer utilizes the Core I5 with 8 GB RAM.

6.2. Experimental results

This section, illustrates experimental studies are performed to display and check the feasibility of the NLCA algorithm. The experiment is performed on a 128-bit key size text data type and128 bits is the block size. The 128-bit key, which is expressed as a 4 × 4 matrix, is generated based on the key generation procedure.

Example:

The inputs are:

1 Data block = Original

0A 0B 0C 0D 0F 01 02 03 04 05 06 07 08 09 1A 2B

1 Encryption key (K) = Key cipher

4F 29 4C 71 D3 AB 29 D0 AB 79 AC 69 A2 73 AC 7B

1 Number of rounds = 5

Table 4. as an example of the encryption method lists the obtained values for encrypting a message block for four rounds. For all stages of the four rounds, the table displays the message block bytes, from plain text to cipher-text.

For the above example, the inverse method, i.e. the reproduction of the plaintext of the message block from its cipher-text is shown in Table 5.

Table 4
Example of NLCA encryption algorithm using 4 rounds.

	Value															
State	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
Original	0A	0B	0C	0D	0F	01	02	03	04	05	06	07	08	09	1A	2B
key	01	02	04	05	06	AA	BB	CC	44	DD	EE	88	09	04	05	06
Key cipher	4F	29	4C	71	D3	AB	29	D0	AB	79	AC	69	A2	73	AC	7B
Round 1	BA	DD	BF	83	AF	5B	FA	9A	2B	59	27	2P	P8	DF	A9	A5
Round 2	C3	0F	2C	B1	41	5P	2E	1D	F8	A1	E9	F0	01	0D	FA	04
Round 3	D7	89	7F	27	39	A9	1C	1D	A0	EB	00	D4	15	8B	A2	92
Round 4	64	25	AF	99	81	32	9A	53	A6	0D	A2	84	FD	67	53	50
Encrypted	64	25	AF	99	81	32	9A	53	A6	0D	A2	84	FD	67	53	50

Table 5
Example of NLCA decryption algorithm using 4 rounds.

	Value															
State	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
Key cipher	4F	29	4C	71	D3	AB	29	D0	AB	79	AC	69	A2	73	AC	7B
Encrypted	64	25	AF	99	81	32	9A	53	A6	0D	A2	84	FD	67	53	50
Round 4	64	25	AF	99	81	32	9A	53	A6	0D	A2	84	FD	67	53	50
Round 3	D7	89	7F	27	39	A9	1C	1D	A0	EB	00	D4	15	8B	A2	92
Round 2	C3	0F	2C	B1	41	5P	2E	1D	F8	A1	E9	F0	01	0D	FA	04
Round 1	BA	DD	BF	83	AF	5B	FA	9A	2B	59	27	2P	P8	DF	A9	A5
Original	0A	0B	0C	0D	0F	01	02	03	04	05	06	07	08	09	1A	2B

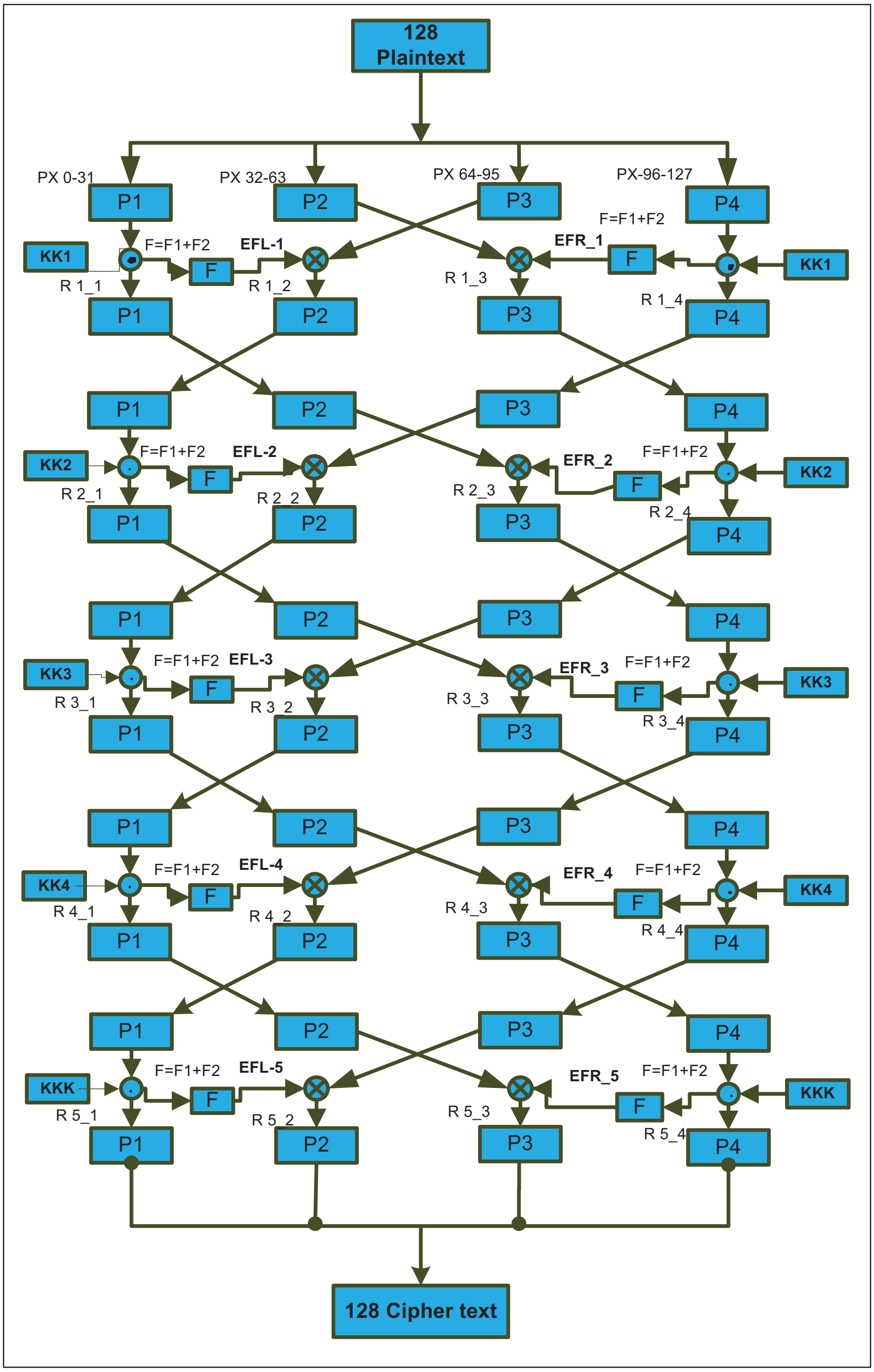


Fig. 9. Block diagram of encryption process.

6.3. Comparative analysis of parameters

This section describes analysed the differences between analysis the proposed algorithm namely (NLCA) and others symmetric algorithms.That are namely **DES** [31], **AES** [32], **HIGHT** [33], **Blowfish** [34], **and LED** [35], by executing them in the same cloud environment using the Block Size, Key Length, Possible Key, Mathematical Operations, Cipher Type and Security Power parameters as shown in **Table 6.** The results observe that the NLCA algorithm is high security strength with a highly flexible. The NLCA algorithm gives lowers computational complexity and decreases the power of processing by having a clear architecture which includes five rounds for encryption; each round requires simple mathematical procedures. In order to decrease the encryption strain, the dynamic key generation procedure is carried out at the encryption. AES, HIGHT, DES, on the other hand, consists ("12, 32, 32, and 20") rounds of encryption respectively.

In addition, the NLCA algorithm has been based on lightweight round functions. In such a way that computationally -complex multiplication is not used, Variable rotation as in Blowfish, S-boxes of large scale such in "AES", and variable rounds in "HIGHT". While mathematical operations number per round and S boxes used in NLCA is higher, the whole complexity is smaller. This is because most mathematical op-

Table 6
Some symmetric key algorithms' comparison in terms of flexibility, architecture, security & limitation.

	DES [29]	AES [30]	HIGHT [31]	Blowfish [32]	LED [33]	Proposed algorithm
Structure	Feistel	Substitution-Permutation	Festial	Festial	Festial	Festial+SP
Block size	64 bits	128 bit	64 bits	64 bits	64 or 128	128 , or 256
Key size	56 bits	128, 192, 256 bits	128 bits	32–448 bits	64 or 128	128,256
No. of Round	16	10, 12, 14	32	16	Variable	4
Possible key	2^{56} bits	2^{128} , 2^{192} Or 2^{256} bits	2^{128} bits	2^{32}–2^{448} bits	2^{64},2^{128} bits	2^{128} ,2^{256} bits
Mathematical Operations	Permutation, XOR, Shifting, Substitution. (6 bits input 4 output bits)	XOR, Mixing, Substitution, Shifting, Multiplication, Addition. (16 bits)	Modular Addition, XOR, Modular subtraction, Shifting. (8 bits)	XOR, Mixing, Substitution, Shifting ,the S-boxes accept 8-bit input and produce 32-bit output	XOR, rotations, 2n mod addition, substitution (6 bits)	XOR, XNOR, Shifting, Substitution (4 bits)
S-P Structure	8 S-Box	1 S-Box	N/A	4 S- Boxes	4 S- Boxes	4 S- Boxes
S-Box Size	16 * 16 (16 bits)	16 * 16 (16 bits)	N/A	8 × 4(32 bits)	4 × 4 (4 bits)	4 × 4 (4 bits)
Security rate	Proven inadequate.	Secure	Secure	Secure	Secure	Highly Secure

Table 7
Some symmetric key algorithms' comparison in terms of processing time.

Key Size	DES 56	3DES 192	AES 256	Blow fish 256	LED 64
Possible key	2^{56}	2^{192}	2^{256}	2^{256}	2^{64}
File 256KB	0.007	0.022	0.007	0.007	0.015
Size512KB	0.015	0.046	0.015	0.015	0.03
1MB	0.03	0.09	0.03	0.03	0.06
10MB	0.32	0.87	0.32	0.26	0.64
50MB	1.89	4.5	1.61	2.13	2.9

Table 8
Processing time of the new hybrid technique (NLCA).

File Size	256 KB	512 KB	1 MB	10 MB	50 MB
Time (s)	0.008	0.016	0.03	0.3	1.6
	0.01	0.021	0.04	0.4	2.11
	0.013	0.026	0.053	0.5	2.6
	0.015	0.03	0.061	0.62	3.058
Average t (s)	0.0115	0.02325	0.046	0.455	2.342

erations are based on (4-bit) data and (S boxes) are only 4 × 4 sizes. This differs from other mathematical schemes that operate on 8 or 16 bits of data.

The NLCA algorithm offers data size consistency in such a way that it allows various block and key lengths that are not provided by most other schemes mentioned and any of the other methods mentioned do not support it such in "AES" and "RC6″ the block length is fixed, while both block and key lengths are set in "HIGHT" in the NLCA algorithm flexibility feature aims to increase cloud storage security performance.

All these symmetric algorithms, such as ("AES, LED, DES, 3DES, Blowfish") were also implemented and checked in Table 7 using various sizes of input text files: **256 KB, 512 KB 1 MB, 10 MB**, and **50 MB**. The results described in **Table 7** are the average encryption time estimated three times after the experiments were conducted. Also Key size of different cryptographic algorithms was Shown Fig 10.

The data processing time generated by the encryption and the key generation time for the new algorithm are also faster than other algorithms with secure key generation described in table.

As you have seen in Table 8 and Fig. 11 Using the proposed algorithm, we can achieve fast data processing efficiency and an acceptable level of protection through this methodology. This NLCA improves the confidentiality and integrity of cloud-saved data while making it accessible on request. In terms of security the most known Threats such as Weak key attacks, Symmetric properties, Related-key attacks and Differential & Linear cryptanalysis [19,20] can be resisted by the NLCA algorithm.

Finally, on the basis data encryption enforcement properties of cloud storage, a brief description of NLCA algorithm is given.

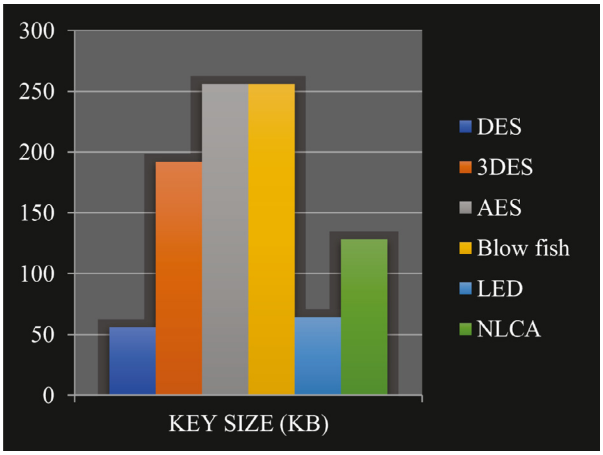


Fig. 10. Key size of different cryptographic algorithms.

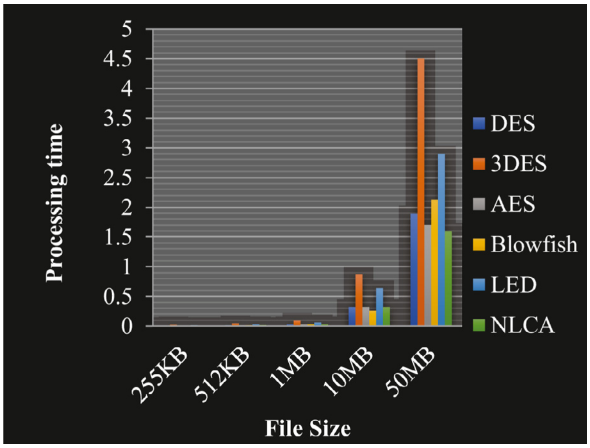


Fig. 11. Processing time of the new technique.

- **Security:** Because of the use of a complex structure and a mixture of Feistel and SP architectural approaches, NLCA is a secure algorithm.
- **Key Generation Process** NLCA provides an efficient key process that helps to avoid brute-force attacks due to the matrix and f-function extension of a key rather than a single extension key**.** NLCA since the **Key Generation** method is used, the security level will be increased.
- **Time Complexity:** There is no greater difficulty in time owing to the reduction of the demands of further rounds.
- **Storage:** The proposed algorithm is suitable for a distributed storage system in the context of cloud computing because it uses the principle of hidden sharing to provide secure access to data through insecure nodes independently
- **Reliability:** the algorithm more reliable and secure.
- **Integrity:** A minor shift in input data can bring a dramatic change in the ciphered output due to the use of the transpose and swap procedures.

7. Conclusion

The security of cloud computing has become the main of the core issues of cloud computing. Various processes and techniques have been proposed including cryptography, which is the most effective.

In this paper a new lightweight cryptographic algorithm has been proposed. It called a New Lightweight Cryptographic Algorithm (NLCA) for enhancing data security in cloud computing environment. It encrypts data based on symmetric cryptography. The algorithm is a 16 bytes (128-bit) block cipher and wants 16 bytes (128-bit) key to encrypt the data. The algorithm is simple and highly secure encryption-decryption. It is inspired by Feistel and SP architectural methods to improve the complexity of the encryption. The proposed algorithm compared the performance with some frequently cryptographic algorithms namely DES, AES, HIGHT, Blowfish, LED using various parameters that are block size, key length, possible key, mathematical operations, cipher type, and security power.

Experimental results show that the NLCA algorithm has demonstrated a powerful security level and a clear improvement in the encryption/decryption providing high security, and low computation cost. It is even more effective for the world of cloud computing, concerning its fast data collection and processing time. In future the NLCA algorithm can be implemented in hardware that may produce much better results.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was Partial supported by the Ministry of Higher Education – Yemen additionally; the authors would like to thank the anonymous reviewers for their comments and suggestions to enhance the quality of this manuscript. Also, I need to thank my guide to Prof. Dr. Sudhir B. Jagtap. For his guidance to my work.

References

[1] M. Köhler and S. Benkner, “VCE - A Versatile Cloud Environment for Scientific Applications,” 2011.

[2] P. Mell, T. Grance, The NIST definition of cloud computing - SP 800-145, NIST Spec. Publ. (2011), doi:10.1136/emj.2010.096966.

[3] B.D. Parameshachari, H.T. Panduranga, S. liberata Ullo, Analysis and Computation of Encryption Technique to Enhance Security of Medical Images, IOP Conference Series: Materials Science and Engineering, 925, IOP Publishing, 2020.

[4] W. Du, Y.S. Han, J. Deng, and P.K. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks,” 2003, doi: 10.1145/948117.948118.

[5] G.S. Pavithra, N.V. Babu, Energy efficient hierarchical clustering using HACOPSO in wireless sensor networks, Int. J. Innovat. Technol. Explor. Eng. 8 (12) (2019).

[6] S. Singh, Y.S. Jeong, J.H. Park, A survey on cloud computing security: issues, threats, and solutions, J. Netw. Comput. Appl. (2016), doi:10.1016/j.jnca.2016.09.002.

[7] A.N. Jaber and M.F. Bin Zolkipli, “Use of cryptography in cloud computing,” 2013, doi: 10.1109/ICCSCE.2013.6719955.

[8] D.S. Abd Elminaam, H.M.A. Kader, M.M. Hadhoud, Evaluating the performance of symmetric encryption algorithms, Int. J. Netw. Secur. (2010).

[9] M. Panda, “Performance analysis of encryption algorithms for security,” 2017, doi: 10.1109/SCOPES.2016.7955835.

[10] L. Tawalbeh, N.S. Darwazeh, R.S. Al-Qassas, and F. AlDosari, “A secure cloud computing model based on data classification,” 2015, doi: 10.1016/j.procs.2015.05.150.

[11] R. Arora, A. Parashar, Secure user data in cloud computing using encryption algorithms, Int. J. Eng. Res. Appl. (2013).

[12] S.S. Khan, P.R. Tuteja, Security in cloud computing using cryptographic algorithms, Int. J. Innov. Res. Comput. Commun. Eng. (2015), doi:10.15680/ijircce.2015.0301035.

[13] D.P. Timothy, A.K. Santra, A hybrid cryptography algorithm for cloud computing security, 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore (2017) 1–5, doi:10.1109/ICMDCS.2017.8211728.

[14] J.R.N. Sighom, P. Zhang, L. You, Security enhancement for datamigration in the cloud, Futur. Internet (2017), doi:10.3390/fi9030023.

[15] Z. Gong, S. Nikova, and Y.W. Law, “KLEIN: a new family of lightweight block ciphers,” 2012, doi: 10.1007/978-3-642-25286-0_1.

[16] G. Leander, C. Paar, A. Poschmann, and K. Schramm, “New lightweight des variants,” 2007, doi: 10.1007/978-3-540-74619-5_13.

[17] T.P. Berger, J. Francq, M. Minier, G. Thomas, Extended generalized feistel networks using matrix representation to propose a new lightweight block cipher: lilliput, IEEE Trans. Comput. (2016), doi:10.1109/TC.2015.2468218.

[18] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, I. Verbauwhede, RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms, Sci. China Inf. Sci. (2015), doi:10.1007/s11432-015-5459-7.

[19] M. Usman, I. Ahmed, M. Imran, S. Khan, U. Ali, SIT: a lightweight encryption algorithm for secure internet of things, Int. J. Adv. Comput. Sci. Appl. (2017), doi:10.14569/ijacsa.2017.080151.

[20] A.H.A. Al-ahdal, G.A. Al-rummana, G.N. Shinde, N.K. Deshmukh, A Robust Lightweight Algorithm for Securing Data in Internet of Things Networks, ustainable Communication Networks and Application. Lecture Notes on Data Engineering and Communications Technologies, vol 55. Springer, (2021) In press.

[21] K. Aoki et al., “Camellia: a 128-Bit block cipher suitable for multiple platforms – design and analysis,” 2001, doi: 10.1007/3-540-44983-3_4.

[22] M. Ebrahim and C.W. Chong, “Secure Force: a low-complexity cryptographic algorithm for Wireless Sensor Network (WSN),” 2013, doi: 10.1109/ICCSCE.2013.6720027.

[23] B. Schneier, “Description of a new variable-length key, 64-bit block cipher (Blowfish),” 1994, doi: 10.1007/3-540-58108-1_24.

[24] D. Coppersmith, Data encryption standard (DES) and its strength against attacks, IBM J. Res. Dev. (1994), doi:10.1147/rd.383.0243.

[25] J. Daemen, R. Govaerts, and J. Vandewalle, “A new approach to block cipher design,” 1994, doi: 10.1007/3-540-58108-1_2.

[26] A. Biryukov, L. Perrin, and A. Udovenko, “Reverse-engineering the S-box of streebog, Kuznyechik and STRIBOBr1,” 2016, doi: 10.1007/978-3-662-49890-3_15.

[27] Federal Information Processing Standards, National bureau of standards, federal information processing standards publication 197: announcing the advanced encryption standard (AES), Fed. Inf. Process. Stand. Publ. (2001), doi:10.1016/S1353-4858(10)70006-4.

[28] A. Bogdanov et al., “PRESENT: an ultra-lightweight block cipher,” 2007, doi: 10.1007/978-3-540-74735-2_31.

[29] J.L. Massey, “SAFER K-64: a byte-oriented block-ciphering algorithm,” 1994, doi: 10.1007/3-540-58108-1_1.

[30] D. Basak, R. Toshniwal, S. Maskalik, A. Sequeira, Virtualizing networking and security in the cloud, Oper. Syst. Rev. (2010), doi:10.1145/1899928.1899939.

[31] A.U. Rahman, S.U. Miah, S. Azad, Advanced encryption standard, Practical Cryptography: Algorithms and Implementations Using C++, 2014.

[32] M.A. Wright, The advanced encryption standard, Netw. Secur. (2001), doi:10.1016/S1353-4858(01)01018-2.

[33] D. Hong et al., “HIGHT: a new block cipher suitable for low-resource device,” 2006, doi: 10.1007/11894063_4.

[34] M.N. Valmik, P.V.K. Kshirsagar, Blowfish algorithm, IOSR J. Comput. Eng. (2014), doi:10.9790/0661-162108083.

[35] G. Bansod, N. Raval, N. Pisharoty, Implementation of a new lightweight encryption design for embedded security, IEEE Trans. Inf. Forensics Secur. (2015), doi:10.1109/TIFS.2014.2365734.