Hindawi
Security and Communication Networks
Volume 2020, Article ID 8863345, 16 pages
https://doi.org/10.1155/2020/8863345

Research Article

Secure Framework Enhancing AES Algorithm in Cloud Computing

Ijaz Ahmad Awan,[1,2] Muhammad Shiraz,[3] Muhammad Usman Hashmi,[1] Qaisar Shaheen,[1] Rizwan Akhtar,[4] and Allah Ditta[5]

[1]*Department of Computer Science, Superior College, Lahore, Pakistan*
[2]*University of Engineering and Technology, Lahore, Pakistan*
[3]*Department of Computer Science, Federal Urdu University of Arts, Science and Technology, Islamabad, Pakistan*
[4]*School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, China*
[5]*Department of Information Sciences, Division of Science & Technology, University of Education, Lahore, Pakistan*

Correspondence should be addressed to Qaisar Shaheen; qaisar.shaheen2002@gmail.com

Received 17 June 2020; Revised 4 August 2020; Accepted 7 August 2020; Published 1 September 2020

Academic Editor: Umar M. Khokhar

The tremendous growth of computational clouds has attracted and enabled intensive computation on resource-constrained client devices. Predominantly, smart mobiles are enabled to deploy data and computational intensive applications by leveraging on the demand service model of remote data centres. However, outsourcing personal and confidential data to the remote data servers is challenging for the reason of new issues involved in data privacy and security. Therefore, the traditional advanced encryption standard (AES) algorithm needs to be enhanced in order to cope with the emerging security threats in the cloud environment. This research presents a framework with key features including enhanced security and owner's data privacy. It modifies the 128 AES algorithm to increase the speed of the encryption process, 1000 blocks per second, by the double round key feature. However, traditionally, there is a single round key with 800 blocks per second. The proposed algorithm involves less power consumption, better load balancing, and enhanced trust and resource management on the network. The proposed framework includes deployment of AES with 16, 32, 64, and 128 plain text bytes. Simulation results are visualized in a way that depicts suitability of the algorithm while achieving particular quality attributes. Results show that the proposed framework minimizes energy consumption by 14.43%, network usage by 11.53%, and delay by 15.67%. Hence, the proposed framework enhances security, minimizes resource utilization, and reduces delay while deploying services of computational clouds.

1. Introduction

It is observed that cloud technology is used in a number of architectures, services with further technologies, and various software design approaches [1]. Cloud service models include platform as a service (PaaS), software as a service (SaaS), and infrastructure as a service (IaaS). Architecture solutions for the public, private, community, and hybrid system depend on four cloud platform deployment models [2]. Advantages of cloud computing include flexibility, accessibility, and capacity when linked to traditional online computing or storage method [3]. However, a number of security concerns are associated with computational clouds including (i) privacy and security issues with cloud service providers and (ii) customer-related security issues [4]. In the literature, various types of attacks related to the strength of the AES (advanced encryption standard) algorithm have been proposed [5], for instance, different fault analyses which attack and introduce faults into the AES (advanced encryption standard) structure with the target to retrieving the secret information [6].

Furthermore, cloud computing standard can propose some feasible practices of service area, by means of computational resources on behalf of extraordinary performance in computing applications, telecommunication services, social networking, and web services [7, 8]. In addition, cloud storage in data centres is very valuable for users just before storing and

accessing their data distantly at any time without any further load [9, 10]. On the contrary, the main problem of cloud data storage is security. As a result, cloud data centres must have some mechanisms which are capable to ensure storage perfection and integrity of data that are stored on cloud [11].

Existing security systems employ one or two attributes at a time, i.e., low security and more time consumption to encrypt/decrypt the data. This makes the process more time-consuming and therefore increases the network use, power consumption, and delay in the network [12–16]. Cloud computing is that kind of platform which shares the data and resources efficiently, and therefore, security must be provided to the users as security is an important aspect of cloud computing. So, this is the responsibility of the cloud service providers to provide security with all attributes, such as less power consumption, delay of network, and time consumption [17–23]. Already, traditionally available methods are not able to quantify the security of cloud services effectively. Secure framework in cloud computing is a method that provides simplified management and accessing of computing resources, and a cost-effective approach is the need of the hour. The framework should use low power, time, and delay of network consumption with encryption and decryption that enhance the security of data in cloud computing.

The paper contributes towards the design of the security framework by implementing a new scheme of encryption/decryption. It also determines the serious components of the security framework within the cloud computing community. It would be helpful for those cloud users and cloud service providers who have similar requirements in terms of security during implementation. The framework helps in faster computing with lesser power consumption, network usage, and reduced network delay due to the smart algorithm. The framework employs a symmetrical encryption method to provide trust to users and enables trusted gateways. The proposed framework includes the key features including enhanced security and owner's data privacy. It modifies the 128 AES algorithm to increase the speed of the encryption process 1000 blocks per second by the double round key feature. However, traditionally, there is a single round key with 800 blocks per second. The proposed algorithm involves less power consumption, better load balancing, and enhanced trust and resource management on the network. The proposed framework includes deployment of AES with 16, 32, 64, and 128 plain text bytes. Simulation results are visualized in a way that depicts suitability of the algorithm while achieving particular quality attributes. Results show that the proposed framework minimizes energy consumption by 14.43%, network usage by 11.53%, and delay by 15.67%. Hence, the proposed framework enhances security, minimizes resource utilization, and reduces delay while deploying services of computational clouds.

The remainder of this paper is configured as the following sections: Section 2 details the literature review. Section 3 defines the framework architecture. Section 4 includes the experimental environment. Section 5 presents the performance results of both existing and proposed frameworks. Section 6 defines the forthcoming features associated with this paper exertion.

2. Literature Review

Several modifications were introduced in AES in order to enhance the performance speed and security by introducing some complexities in algorithms. These modifications are implemented on different software and hardware designs. However, preview framework security is always a concern due to some security constraints and problems with cloud computing. The security is provided to the information which is stored on the cloud by using cryptography algorithms. There are extensive security frameworks for cloud computing that uses enormous encryption techniques. Out of these, a few of them are presented here.

The security framework is based on the multicloud environment to store digital data at all. In order to prevent data disclosure, they practiced a segmentation approach to fragment the input appearance into several areas. The integrity of the outsourced clients' data helps to verify watermarking technique. Any accidental change to outsourced clients' data can be detected by the digital signature and watermarking methods [24]. This paper focuses on the computation of different methods which explain how to increase data security so that prevention from different security attacks and breaches can be made. Mitigation approaches used in this research on the HMAC (Hashed Message Authentication Code) were ECC and MD5. This proposed solution is based on different security levels; as a result, access control, authentication, confidentiality, integrity, and encryption are achieved in this work. The authors performed and checked the security solution in real-time as well as in real cloud computing environment and also concluded that the solution that is been provided has very low overhead for upload and download service time [17]. The framework presented in this study is more secure, and it provides more privacy to the data. This framework splits data into different blocks of bit. On every two blocks of bits, genetic algorithm is applied. Concluding output of each genomic algorithm procedure is a ciphertext along with two blocks of bits. Each ciphertext is stored on the cloud at a distinct location, and the location of the ciphertext is not secure. What makes it more secure from attackers to find the exact location of the ciphertext? The innovative security framework puts on a genetic algorithm on minor block size that increases the security. Furthermore, the framework uses the proficiency list aiming to secure and to access data [18].

In this paper, authors proposed a new framework that ensures the data security and integrity and also focused on the encryption and decryption approaches facilitating the cloud user with data security assurance. The proposed solution talked about the increased security along with the performance. Their solution has also included functioning of the forensic virtual machine, malware detection, and real-time monitoring of the system [25]. In this paper, the authors suggested a framework such that the objective is to store data in various clouds. The given framework is found based on 3DES and RSA encryption. On the contrary, this methodology is lacking in efficiency, privacy, and overload middleware through multiple functions [26]. In this paper, the authors studied, multilevel licensing framework approval

preservation cloud penetrating data. Safeguarding the familiar and delicate cloud data is obtainable by the three covers' framework. Those restrictions are being the security and privacy strategies, safety and approval policies which outcomes from the three films' security framework [19]. In this paper, the authors proposed quality metrics and details probe on instance cloud service broker frameworks are provided. These streak metrics help in enforcing standards on cloud service providers by using quality-based cloud service broker framework (QCSB). The algorithm and implementation of QCSB have been obsessing. At last, the authors concluded that the proposed material QCSB not only assists cloud computing to locate optimal CSP (cloud service provider) for cloud services but also affiliates candidate CSPs according to user quality preferences [20].

The complexity detects were an effect of dismiss logical purposes in the MixColumn conversion of AES. These reasonable tasks were eradicating in the modified version of AES. Afterward,on utilizing the modified AES, a 13.6% reduction in LUTs, 10.93% share discount, and a 1.19% reduction in interruption eating was attained. Likewise, the small dispersal rate met through the conservative AES at the initial nonentity, and important agenda sequences are spoken in [27]. In this research, they examined five metrics specifically: the graphic study, file size, radiance histogram, assessment by pixel, and show distance. In the file scopes, there were differences wherever it displays the regular worth of the fraction variations to −23.85% from the unique to the encrypt duplicate and −1.45% percentage worth from the innovative to the decrypt duplicate [28]. This paper showed an overview of the latest research studies that are going on in fog computing and the IoT and its uses; it also enlightened the research gaps and directions for further future research studies in the integration of fog computing and IoT (Internet of Things). A modern fog computing framework was presented [29]. The modified AES contained 10 series for encrypting, and the replacement and addition processes of the columns have been substituted by the line change and pixel standard summary. These processes not only decrease the spell complication of the algorithm but also improve the dispersal aptitude to the CCAES (combining the chaos and AES) algorithm.The encrypted descriptions by the CCAES algorithm remained unaffected to the variance occurrences. The project algorithm is protected alongside the entropy occurrences. The simulation consequences illuminate that the minor deviations in the unique appearance and consequences in the important fluctuations in the encrypt duplicate and the innovative appearance cannot be retrieved [30]. This paper described the CloudSim simulator counting its architecture, aces, convicts, and CloudSim forms. Likewise, it characterized exactly how to practice CloudSim demonstration and replication in the cloud environment. Furthermore, it also describe the way to calculate approximate presentation limits like regular reversal time, amount, implementation period, types pan and entire conclusion period, etc. [31].

This paper reported dissimilar data safety and privacy security concerns in a cloud calculating environment and suggested a technique for dissimilar security services such as verification, approval, and privacy along with observing in suspension. Cloud computing plans a different technique for obtaining cloud data in the actual environment. 128 bit AES encryption is recycled for privacy, genuineness, and contact controller [32]. In the future work, load balancer by means of My Load Balancer optimization method has been compared with the two greatest well-known weight balancer techniques, i.e., Round-Robin and Supper Present Implementation Freight, also recognized as Active Monitoring Load Balancer. All such Java-based virtual techniques are used to create Cloud analyst toolkit. Graph procedures have been recycling to prove the comparative analysis [33]. The procedure of cryptography involves two main methods which are encryption and decryption. In the encryption method, a basic manuscript is converted to an innovative text which the others cannot deliver and understand additional than the receiver. Blowfish and AES procedures are exploited for executing a hybrid approach connected to cryptography. This consequence in a cryptograph text which can merely be decrypted by the receiver this one [34]. In this paper, obtainable low-control AES architecture by exploiting humble shift catalogues and variation for key/data stored to decrease journey magnitude and control consumption. A low-power method, called clock gating is used to control exchangeable on S-box[35]. In the present study, Abikoye et al.'s modified AES algorithm [13] is presented which is also used in applications to make a comparison. K-L Tsai et al. presented the modified AES-based algorithm for power reduction in IoT using cloud computing applications [14]. In this paper, similarly, VM (virtual machine) allocation policy is used for security which is almost similar to the technique used in the previous work [36].

In general, the main purpose of all research studies related to the subject areas is to investigate the possible ways to improve the security of cloud computing services. Therefore, in this work, a secure framework has been proposed for securing confidential tasks being stored in cloud systems using AES encryption methods. Finally, a comparison of the results obtained through this proposed framework and traditional framework work formulated in the past is made which showed significant improvement of cloud computing using the proposed framework. The differences between our modified AES and previously developed or modified AES in the JAVA cipher-based security framework have been discussed in this manuscript. It is pertinent to mention here that our trust-based framework blocks the suspicious users from the network and maintains a queue for such users to protect the trusted users.

3. Architecture of the Proposed Secure Framework for Cloud Computing (SFCC)

The architecture of the proposed secure framework for cloud computing (SFCC) is presented in Figure 1.

Framework of secure cloud computing is proposed on the security architecture shown in Figure 1, which describes the information for each component and their applications which are required for secure technologies to operate between components in cloud computing. This framework acts in the

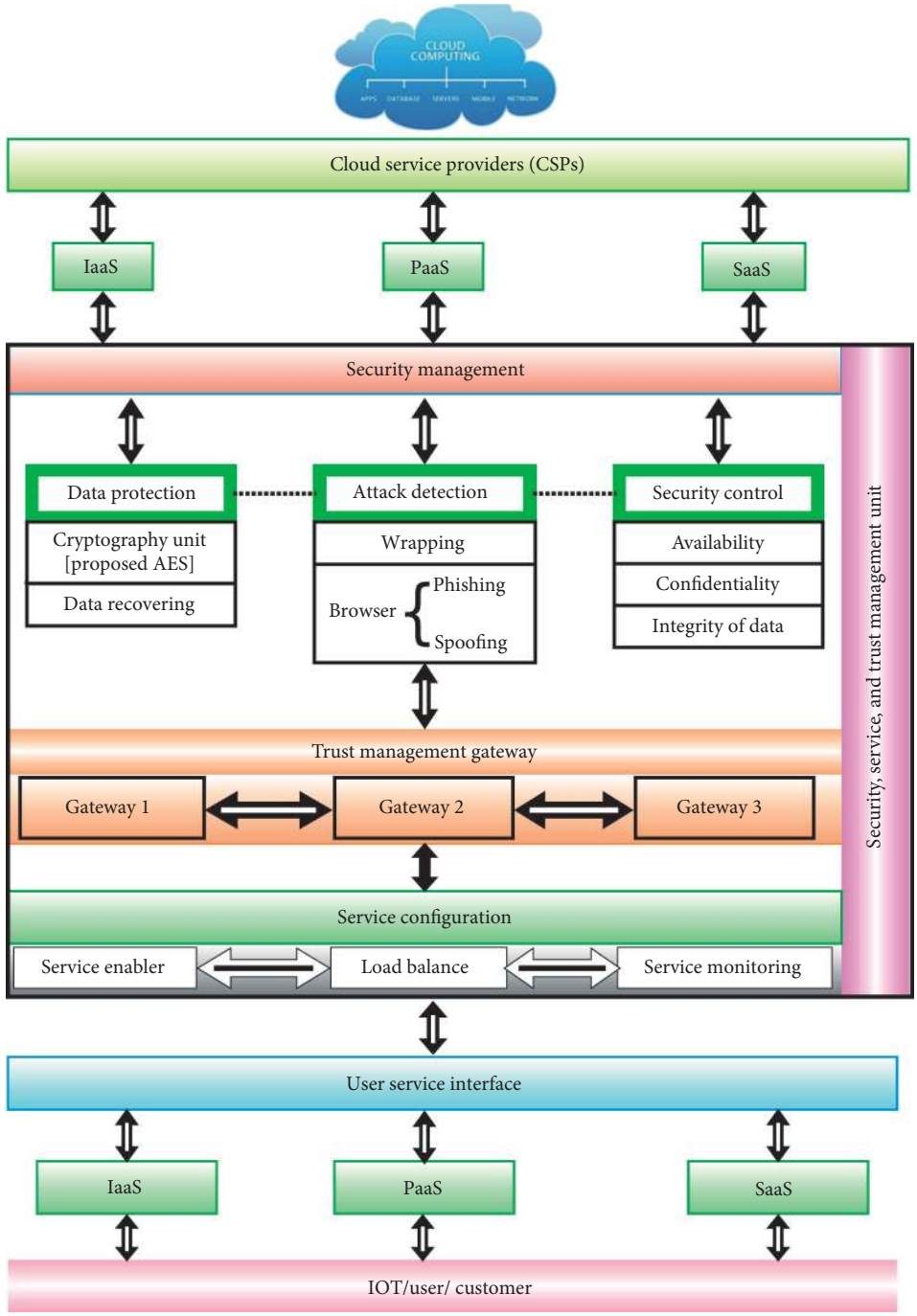

Figure 1: Secure framework for cloud computing (SFCC).

following conditions checking security, privacy, load balancing, and trust. When the user directs a demand to the cloud benefactor, it responds to the user's request and passes the data through framework gateways. The proposed framework includes the following components:

Cloud service provider (CSP) layer: the CSP controls the important sources and ability in construction and calculates the dispersed cloud storage servers processes and directs the live obscure work out method. Its main component is software as a service (SaaS); this is a model in which end users are provided software application (as a service). Platform as a service (PaaS): this model proposed an atmosphere for requests. Development tools that are essential for advanced applications are also provided in this model. Infrastructure as a service (IaaS): this is a platform that offers compulsory properties such as physical machines, virtual machines, and virtual storage.

Security service trust management unit: security service trust management controls all the units which include

security management; trust management gateways also control the service configuration, respectively. Further details of all units are described in the following.

Security management layer: the security management factor offers security and privacy details and implementation functionality. Security service has the following modules and their details.

Security control unit: availability is the percentage of time a customer can access the service. Confidentiality (authentication, authorization, and identification) is an integral component of security. It ensures that the information stored on the cloud is protected against the unintended or unauthorized access. Identification user is typically skilful by retaining usernames and passwords after utilizing web browser in order to admit in Cloud. Integrity of data security control is responsible for maintaining the accuracy of data computation that is coming from the combination of different files and is also responsible for its delivery.

Attack detection unit: ultimately, slightly usual activities that hover the cloud security necessities (e.g., integrity, confidentiality, and availability) are measured to be occurrences. Wrapping is when the attacker attacks by wrapping the communication between two people, while the users do not know this and think data are still coming from the actual root. Unethical browsing is to find bad actions happening, for example, phishing and spoofing and changing browser certificates.

Data protection unit: proposes the AES algorithm to enhance the data security by means of cryptography techniques using AES ciphers as they can encrypt 128 bits' data blocks within 1000 blocks per second with the double round key feature with less power consumption, load balancing, trust, and resource management on the network efficiently. We have used symmetric identification for security, i.e., the same key for encryption and the same key for decryption as identification of data streams in the form of security. It provides greater efficiency for software as well as hardware. The advantage of using symmetric key is to secure a large amount of data. Data recovery: if data is lost in a disaster that ithas a capability to regain it or restore it.

Trust management gateway layer: for the fourth layer, trusted gateways are implemented. These gateways get the encrypted data and decrypt only if the trusted source is connected with a valid internet protocol address of a given domain. These gateways support the issues of trust. There are three gateways in which two are in an alternative manner. In case of the normal gateway is being attacked and misused, other safe gateways shall be chosen to ensure data communication.

Service configuration layer: the service enabler makes provision for personalized cloud service using the user's profile for integration and interoperation. Load balancing can be implemented on hardware, software, or a combination of both. It is important in this configuration that all instances of identity server share the same directory server. Service monitoring: an automatic facility-checking system to assure an extraordinary level of facility presentation and obtainability.

User service interface layer: this layer provides different services to select the user via the internet: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).

Service configuration layer: the last unit for the user, IOT, and customer to send and receive data.

4. Experimental Setup and Implementation of SFCC

The SFCC can be implemented in real time. The results gathered from the simulations are very accurate. These results are theoretically consistent. Everything is implemented accordingly. Codes are very consistent with real-time mechanisms. The SFCC is developed using CloudSim and iFogSim simulators on the Eclipse integrated development environment. CloudSim is a very well-known and popular among simulators for cloud-based applications. It is responsible for the simulation and events handling at cloud. Some libraries are used for different purposes. Libraries used are JavaScript object notation (Json) data saver, common math, and JFreeChart.

The developed simulation comprises SFCC. The proposed framework is generic so that anyone could put one's idea or logic in this simulation and get the required results. It helps the user to test different scenarios under the proposed algorithm. The simulation has the ability to store and generate a large amount of data. It allows a user to measure the factors such as encryption, description, power consumption, network usage, delays, trusted devices, and service management. The advanced encryption standard for encryption and decryption for data protection is used. The comparison of the algorithm with the previous unmodified algorithms is discussed in later sections. The characteristics of the layers and devices are described in Tables 1–11.

4.1. Components. Data centre refers to on-premise hardware, while the cloud refers to off-premise computing. The cloud stores your data in the public cloud, while a data centre stores your data on your hardware. Data centre configuration is displayed in Table 1.

Infrastructure as a service (IaaS): this is a platform that offers compulsory resources such as physical machines, virtual machines, and virtual storage. Infrastructure-as-a-service configuration is displayed in Table 2.

Software as a service (SaaS): this is a model in which end users are provided software applications (as a service). Software-as-a-service configuration is displayed in Table 3.

Platform as a service (PaaS): this model proposed an atmosphere for requests. Development and deployment tools that are essential to advance applications are also provided in this model. Platform-as-a-service configuration is displayed in Table 4.

Table 1: Data centre characteristics of cloud.

Name of the device	Cloud
Level	1
Uploading bandwidth	5000
Downloading bandwidth	12000
Million instructions per second	130.0
RAM	45000
Rate per processing usage/MIPS	100000

Table 2: Data centre characteristics of infrastructure as a service.

Name of the device	Cloud IAAS
Level	2
Uploading bandwidth	4000
Downloading bandwidth	5000
Million instructions per second	50000
RAM	40000
Rate per processing usage/MIPS	400.0

Table 3: Data centre characteristics of software as a service.

Name of the device	Cloud SAAS
Level	2
Uploading bandwidth	4000
Downloading bandwidth	5000
Million instructions per second	60000
RAM	40000
Rate per processing usage/MIPS	400.0

Table 4: Data centre characteristics of platform as a service.

Name of the device	Cloud PAAS
Level	2
Uploading bandwidth	4000
Downloading bandwidth	5000
Million instructions per second	60000
RAM	40000
Rate per processing usage/MIPS	50000

Table 5: Data centre characteristics of security management.

Name of the device	Security management
Level	4
Uploading bandwidth	5000
Downloading bandwidth	5000
Million instructions per second	40000
RAM	35000
Rate per processing usage/MIPS	600.0

Security management: the security management factor offers the security and privacy details and implementation functionality table. Security management configuration is displayed in Table 5.

Table 6: Data centre characteristics of gateway1.

Name of the device	Trusted gateway1
Level	3
Uploading bandwidth	3000
Downloading bandwidth	4000
Million instructions per second	30000
RAM	20000
Rate per processing usage/MIPS	1000.0

Table 7: Data centre characteristics of gateway2.

Name of the device	Trusted gateway2
Level	3
Uploading bandwidth	3000
Downloading bandwidth	4000
Million instructions per second	30000
RAM	30000
Rate per processing usage/MIPS	400.0

Table 8: Data centre characteristics of gateway3.

Name of the device	Trusted gateway3
Level	3
Uploading bandwidth	4000
Downloading bandwidth	4000
Million instructions per second	50000
RAM	34000
Rate per processing usage/MIPS	600.0

Table 9: Data centre characteristics of service configuration.

Name of the device	Service configuration
Level	1
Uploading bandwidth	5000
Downloading bandwidth	5000
Million instructions per second	100000
RAM	40000
Rate per processing usage/MIPS	500.0

Table 10: Data centre characteristics of service provider.

Name of the device	Service provider
Level	1
Uploading bandwidth	5000
Downloading bandwidth	5000 Gbits/sec
Million instructions per second	50000
RAM	20000 gb
Rate per processing usage/MIPS	100.0

Table 11: Virtual machine configurations.

Virtual machine number level	Virtual machine number	Processing elements	Bandwidth (uplink)	Latency input
Level 0	2	20000	800	10
Level 1	4	18000	1000	6
Level 2	6	16000	1200	8

Gateway devices at the second-last level of the hierarchy gateway devices are created. These gateway devices are part of the layer responsible for communicating with proxy servers and cloud devices. Here are the characteristics of the gateway devices. Gateway device configuration is displayed in Tables 6–8.

Service configuration: This facility modifies the cloud service using the user's profile byintegrating service enabler, load balancing, and service monitoring. Service configuration is displayed in Table 9.

Service provider: this is the last unit for users and customers to send and receive data. Service provider configuration is displayed in Table 10.

Virtual machines are created and allocated to hosts to support processing and offloading the modules to support the load balancing mechanism. These virtual machines come with the proposed strong encryption algorithm to support the security and trust feature. The virtual machine configuration is displayed in Table 11

The materials and methods section should contain sufficient detail to repeat all procedures. It may be divided into headed subsections if several methods are described.

4.2. Physical Topology of SFCC. The physical topology shows the pattern of nodes and devices in the network. Physical entities are created, and their competence, capability, and configurations are specified. These entities include sensors, actuators, gateways, and cloud VM (virtual machines). The links between these entities and their configuration are also established. Physical network topology is important to understand the pattern of the network, how various network devices are organized, and how they communicate with each other. These configurations and capacity determine the load a network can tolerate and the amount of data it can transfer. The physical topology is shown in Figure 2.

4.3. Explanation Topology. The computing mechanism of cloud always happens at the top. Cloud stays at the top to manage the lower-level architecture [37]. The three different types of cloud stay below the top layer and act as CSPs [38] according to customers' need. For the third layer, the virtual machine allocation policy mechanism is implemented to support data offloading and privacy for security [39] in the proposed system. Offloading the modules not only provides load balancing but also solves the security issues of cloud by providing a new layer on the hosts. Virtual machines are created and allocated to the hosts to support processing and offloading of the modules to support the load balancing mechanism. These virtual machines come with a strong encryption algorithm to support the security and trust feature.

The virtual machine requires some storage and processing capabilities similar to a host H in nature. Equation (1) represents the conditions for creating a virtual machine. The Vm size is always smaller than the available host H and storage S, where the number of Vms depends on the size of load (β).

If $H = \{H1, H2, H3, \ldots, Hn\}$ and $V = \{Vm1, Vm2, Vm3, \ldots Vm\ N\}$, then

$$\exists Vm \in H \cup S: Vm \propto \beta \text{ where } H \cap S \gg Vm,$$
$$: Vm1, Vm2, Vm3, \ldots, Vm < H1, H2, H3, \ldots, H,$$
$$\cdot \forall V \exists Vm1, Vm2, Vm, \ldots, VmN \in H. \quad (1)$$

Equation (1) represents how VM creation is carried out under various rules and conditions

For the fourth layer, trusted gateways are implemented. These gateways get the encrypted data and decrypt only if a trusted source is connected with a valid Internet protocol address of a given domain. These gateways support the issues of trust [40]. There are 3 gateways in which 2 are alternate manner. In case of a normal gateway is being attacked and misused, other safe gateways shall be chosen to ensure data communication, as shown in Figure 3.

Trusted gateways put the blacklist users into the blocked users' category to ensure the security and privacy of trusted users. The fifth layer is responsible for 3 functions. These functions include service monitoring, load balancing, and service enabling/disabling. The bottom-most layer is based on the users of cloud, and it represents the Internet-of-Things layer in the proposed system. This is how all the aforementioned proposed frameworks work. The trusted customer stays as long as a mediator (trustee) stays. And a mediator stays as long as the cloud service providers are trustable. The chain of trust can be seen in Figure 4 [41].

4.4. Changesin Traditional AES Algorithm. The high-level flow of the proposed AES algorithm in a standard way is presented in Figure 5.

4.4.1. Changes in the Traditional AES Algorithm vs. the Proposed Algorithm. The cloud computing confidentiality framework is presented in this paper. In this framework, data integrity mechanism is used to enhance the data security by the means of cryptography technique. The modified AES (advance encryption standard) ciphers as it can encrypt 128 bit data blocks within 1000 cycles with low power, time, and delay of network consumption. The other work of the frameworks is load balancing, trust, and resource management on the network efficiently.

We have used symmetric identification for security, i.e., the same key for encryption and decryption as identification of data streams in the form of security. The difference between the proposed and previously developed AES is that we have also encrypted 1000 blocks per second with the double round key feature. Previously developed AES uses a single round key with 800 blocks per second. The advantage of using symmetric key is to secure a large amount of data.

4.5. AES Substitution Box (S-Box). The primary stage to around, remains to organize a byte by byte replacement through a lookup table called a substitution box or simply

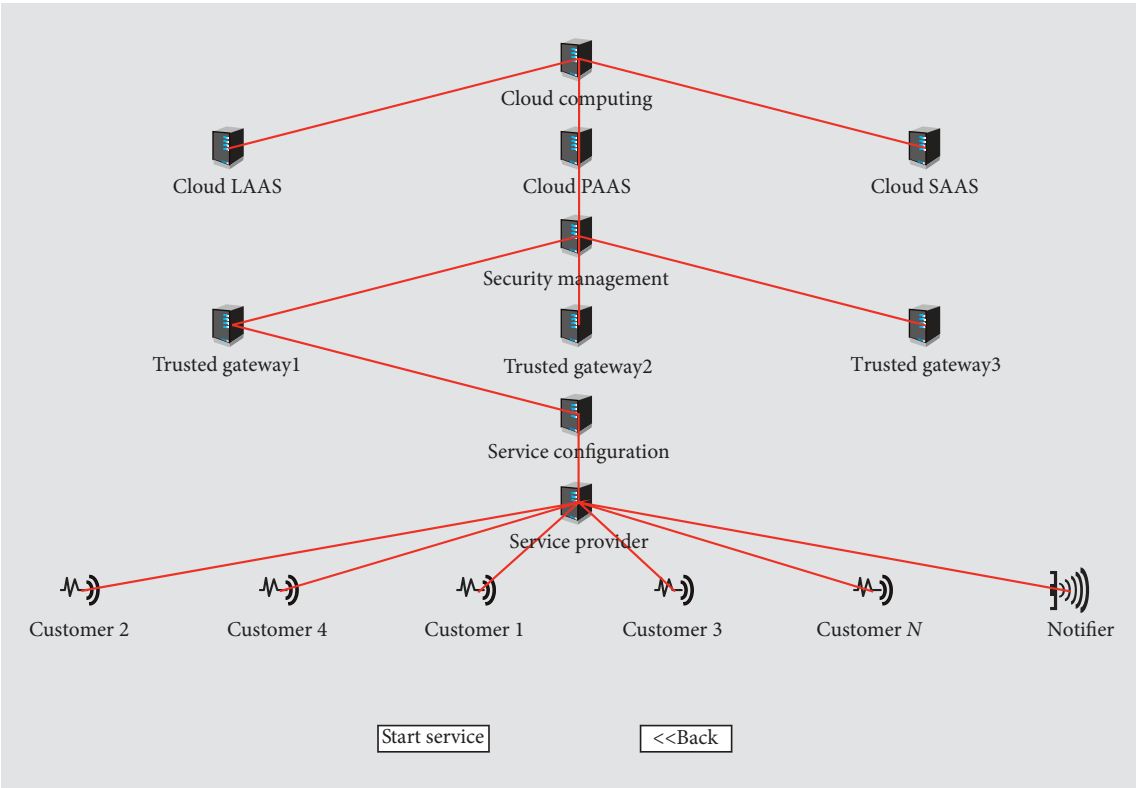

Figure 2: Physical network topology.

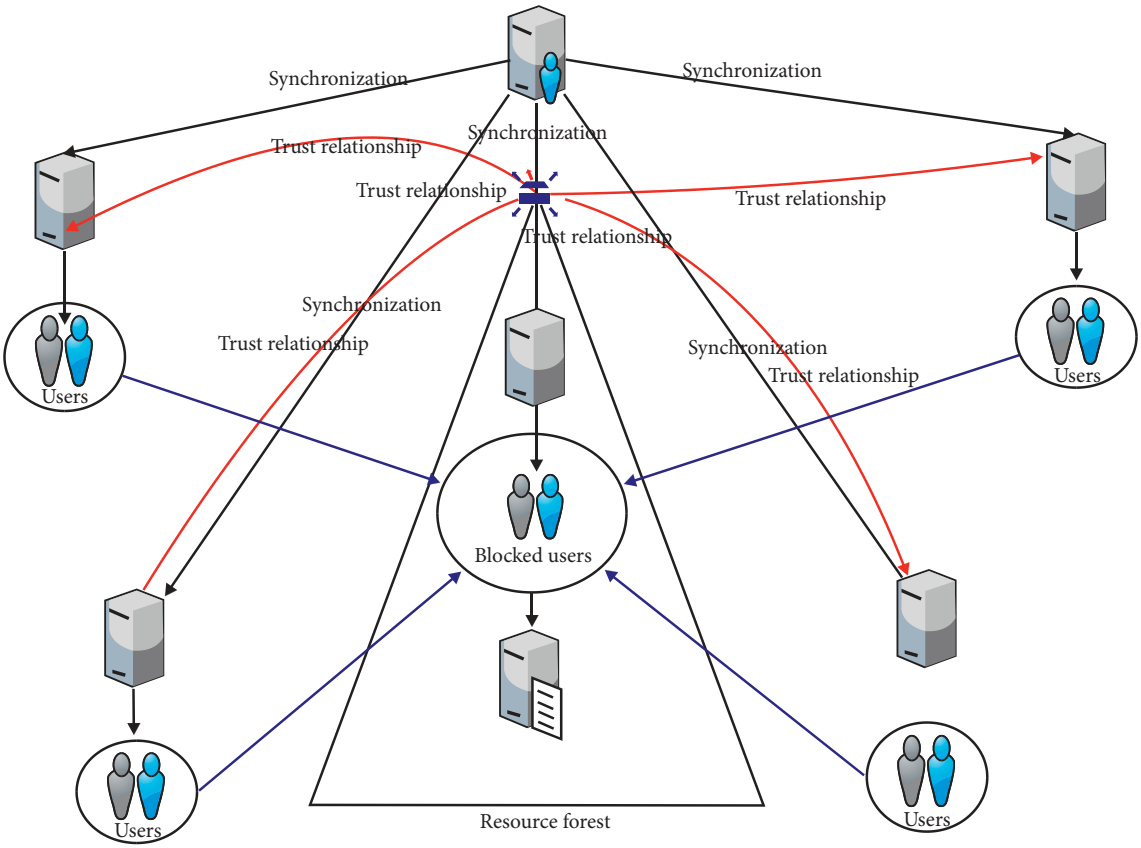

Figure 3: Trusted gateways.

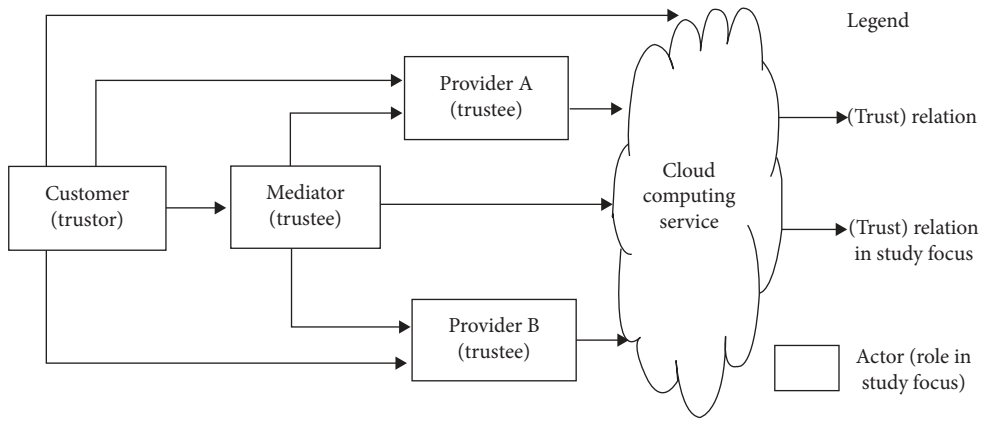

Figure 4: Mediator of cloud service providers' trusted chain.

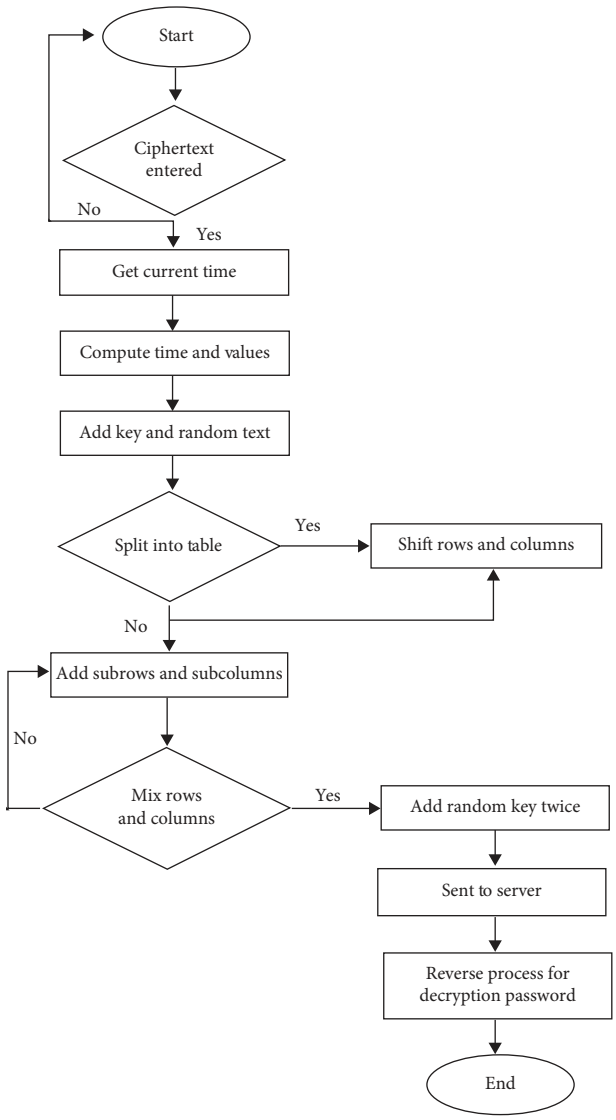

Figure 5: Flow diagram of the proposed algorithm.

S-box. The S-Boxescarry out one to one plotting for all byte values from 0 to 255 in 16×16 arrays. Replacement is a nonlinear conversion which achieves misperception of bits. A nonlinear revolution is vital for each current encryption algorithm and shown to be solid cryptographic original in contradiction to direct and disparity cryptanalysis. The S-box is shown Figure 6. All values are represented in hexadecimal notation [42]. The general substitution box for adding round keys is given in Figure 6.

Rows are represented by x, and columns are represented by y. The mixing process is done with XOR denoted by the symbol ⊕. The binary example in the following will illustrate the functionality of the XOR operator. The row and column mixing and shifting are done by Shift (x_row, y_clolumn) function. The transformed arrays x and y are converted into binaries using ASCII__ASCII 256 standard. Then, the XOR operator performs its ⊕ operation on the bits to generate the ASCII-(American Standard Code for Information Interchange-) generated ciphertext. The cryptographic technique used in SFCC is presented in the low-level language as follows:

$$CiT\,(\text{enc}) = \frac{1}{N}\sum_{i=0}^{1} X_r \oplus Y_c, \tag{2}$$

$$CiT\,(\text{dec}) = N\sum_{i=0} X_c \oplus Y_r. \tag{3}$$

⟶putfieldjavax.crypto.Cipher.spi: javax.crypto.CipherSpi

exec_0 [this] exec_2 [x__rows] (i)

⟶putfieldjavax.crypto.Cipher.provider: java.security.Provider

exec_0 [this] exec_3 [y__columns] (ii)

⟶putfieldjavax.crypto.Cipher.transformation: java.lang.String

exec_0 [this] (iii)

⟶getstaticjavax.crypto.CryptoAllPermission.INSTANCE:

javax.crypto.CryptoAllPermission (iv)

⟶putfieldjavax.crypto.Cipher.cryptoPerm: javax.crypto.CryptoPermission

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure 6: Substitution box [42].

exec_0 [this]aconst_null (v)

⟶putfieldjavax.crypto.Cipher.lock:java.lang.Object return $\prod$; (vi)

The example of $\oplus$ is given as follows.

Let $X = 1110_2$ and $Y = 1001_2$. Then, XOR of X and Y is represented by Z:

$$Z = X \oplus Y = 0111_2. \tag{4}$$

$Z = 0111_2$ is the result of this operand. Table 12 displays the result in the tabular form.

Table 12: XOR operations.

X	Y	Z (result)
1	1	0
1	0	1
1	0	1
0	1	1

5. Results and Discussion

In order to check and test the efficiency of the proposed algorithm, a simple code is used. This test helped us to prove that the proposed AES algorithm is better than any other AES algorithm, and after implementation of AES and advanced AES code on hardware will reduce the execution time. The SFCC results as performed and the implementation of this security framework for cloud computing. The period actsas an energetic character during the peers of key, encryption, and decryption procedure. Altogether, inquiries remain complete on Intel(R) Core-i3 with CPU 2.27 GHz processor, 4 GB RAM on Windows 10 at the work framework by using CloudSim with iFogSim as simulators on Eclipse integrated development environment. CloudSim is a very well-known and popular among simulators for cloud-based applications.

Various parameters such as encryption, decryption, energy consumption, network usage, network delay, trusted devices, and service management devices are compared. The same algorithms are implemented in real-time applications to solve the aforementioned issues. The results gathered from the simulations are very accurate. Codes are very consistent with real-time mechanisms. The simulators are redesigned according to the application need. The implementation period is a basic of the spell that is required to change a basic text to an encryption manuscript and vice versa, while encryption time that is referred to the time taken to change a basic text to a ciphertext and decryption which is referred to the time required to convert a cipher text to a plain text are both predicted to be short in instruction to take rapid and approachable system. Moreover, this execution time somehow is contingent on the layout of the system used. Table 13 offers the execution period in milliseconds (ms), which is obtained by computing the average encryption/decryption time after encrypting/decrypting the input text in 0.5 MB sizes while using the same key run on 16, 32, 64 and 128 bytes.

Table 13 presents the execution time test results in milliseconds (ms), which are attained by computing the average encryption/decryption time after encrypting/decrypting the input text in 0.5 MB sizes of the while using the same key run on 16, 32, 64, and 128 bytes. The results of Figures 7–9 specify that the existing AES has a minor rise in the encryption and decryption time after matched to the existing AES algorithm. Table 13 presents the time comparison between existing AES and different proposed AES algorithms for a string key.

Table 13: Execution time test result [13].

Plain text size (bytes)	AES	Avrg. encryption time (ms)	Avrg. decryption time (ms)
16	Existing AES	0.1658	0.1789
	Proposed AES	0.1190	0.1481
32	Existing AES	0.2976	0.3114
	Proposed AES	0.2507	0.2839
64	Existing AES	0.4564	0.4626
	Proposed AES	0.3916	0.4590
128	Existing AES	0.6984	0.5911
	Proposed AES	0.6014	0.5805
0.5	Existing AES	2359.65	2269.32
	Proposed AES	2159.8	2207.1

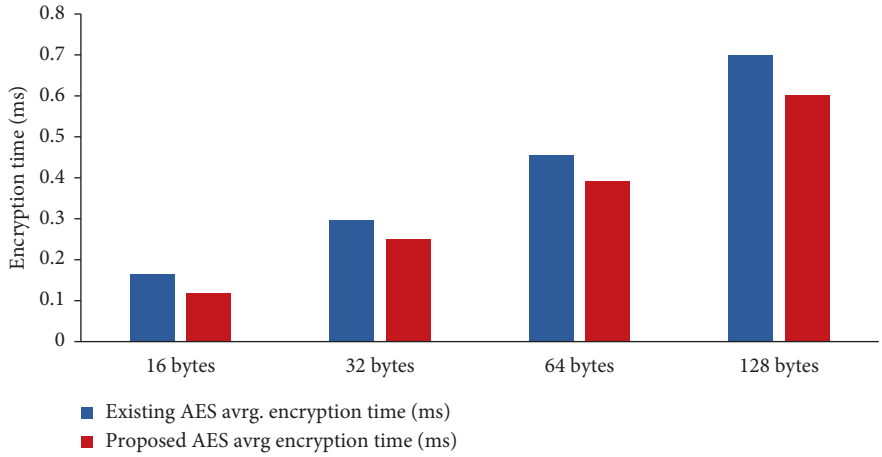

Figure 7: Encrypting time: existing AES vs. proposed AES.

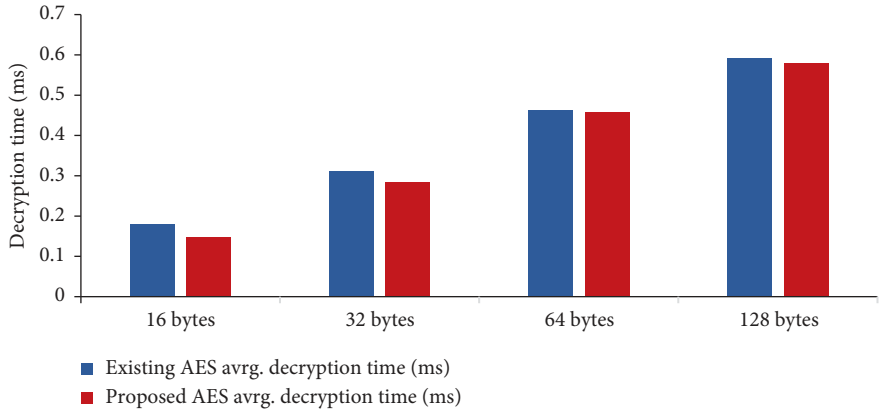

Figure 8: Decrypting time: existing AES vs. proposed AES.

5.1. Avalanche Effect. In cryptography, stuff called dispersal reproduces the cryptographic asset of an algorithm. If there is a small alteration in an input, the output changes meaningfully. This is also called the inundation effect. Avalanche consequence is leisurely by means of pretense reserve. Hamming reserve in material philosophy is the amount of variation. Playacting reserve is the amount of bit-by-bit XOR bearing in mind ASCII value as it develops informal to devise programmatically. A high gradation of dispersal, i.e., extraordinary avalanche consequence, is anticipated. Avalanche's conclusion reproduces the presentation of a cryptographic algorithm. The avalanche effect is described in Table 14.

The avalanche effect is described in Figure 10(simulation results from Table 14)

5.2. Comparative Analysis of Computed Results with the Existing Work. A comparative analysis of computed consequences with the current work is presented as

Table 14: Avalanche effect test result obtained after flipping a single bit in the plain text [13].

Execution program	Plain text	Secret key	Encryption and decryption time	Execution time
First time execution	I Love Unimorin!	H2 + 3S + MuePgIPK3h9SAHOtl6THtl8ak062IgB3ixEto	Encryption time Decryption time	0.05172414 0.03448276
Second execution	I Love Unimorin!	1mRVUf7lRS7W/K + BWFRkP3// KKjf0FtIaSnIGArvudY=	Encryption time Decryption time	0.06666667 0.044444446

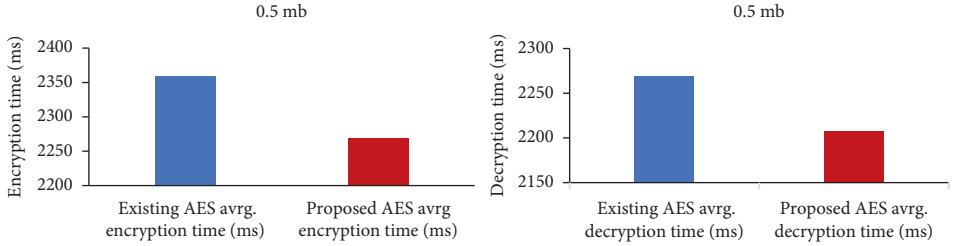

Figure 9: Encrypting and decrypting time: existing AES vs. proposed AES.

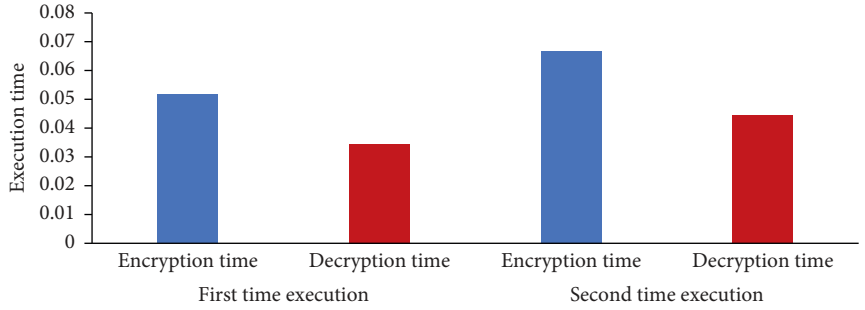

Figure 10: Avalanche effect test result.

follows. However, some researchers analyzed the performance of their advanced AES version. Meanwhile, many authors used encryption and description time as their performance metrics. The simulation environmental comparison between proposed AES and other AES using the CloudSim simulator is graphically represented in Figures 11 and 12.

5.3. Average Energy Consumed. By using the same technique described in [13], the energy consumption is being evaluated. These experiments shared that the proposed frameworks have 14% less energy consumption as compared to [13]. Actual cost taken is given by encryption and the average current that is used by every CPU clock cycle. Equation (5) is used to calculate energy cost per byte as well as various keys of AES encryption schemes:

$$: \sum E = E_c + \left(T_L - \frac{T_c}{T_u}\right) - \mathrm{P} * \mathrm{M}, \tag{5}$$

where C, L. and u represent the current, last, and updated, respectively.

The energy consumption E is the amount of work done on processing Mips M under a time frame T using power model P. The mathematical notation to represent the energy consumption is described in Figure 13.

5.4. Average Network Usage. Network usage is the overall network usage for the system. Network usage is represented in kilobytes. This parameter defines the usage of network resources. The length is reduced and approaches of requests to lower hierarchy by using service configuration so that the request could be processed in the lower hierarchy rather than sending it to cloud again and again. This algorithm reduces 3-hop communication to single-hop communication. Thus, low network usage is obtained through the proposed framework. The more the network is used, the more the expenditure. Efficient network topologies prefer to use minimal network. In these experiments, the network usage is evaluated using the same technique described in [13]. In the proposed framework, network resources are reduced by 11% as compared to [13]. The network while running the implemented encryption schemes is calculated using the following equation:

$$: \sum Nu = Ni + \frac{(L * D * B)}{T}, \tag{6}$$

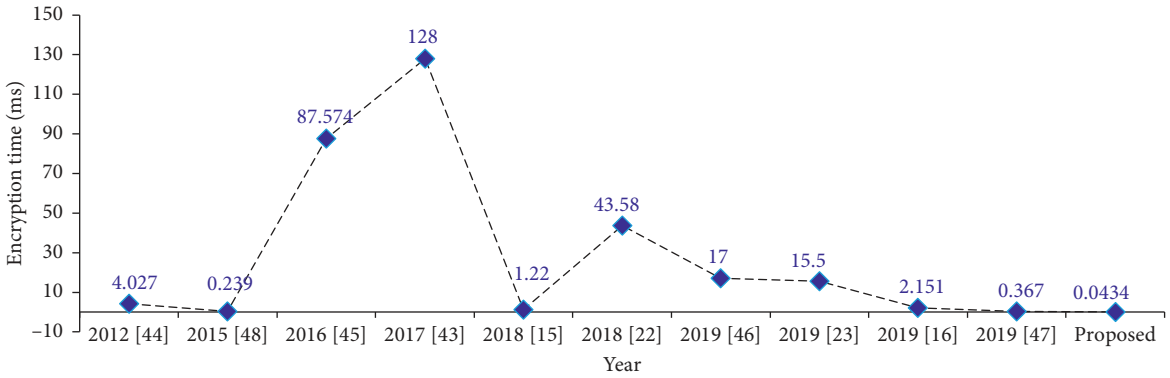

Figure 11: Encryption processing time factor in different AES.

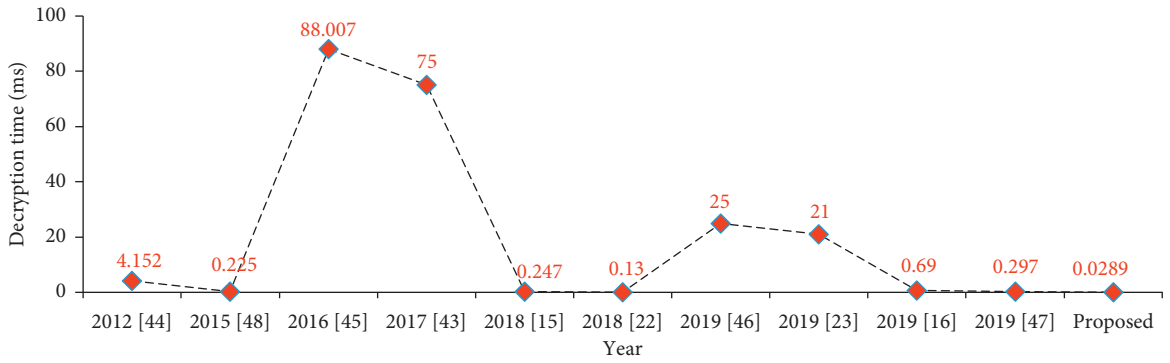

Figure 12: Decryption processing time factor in different AES.

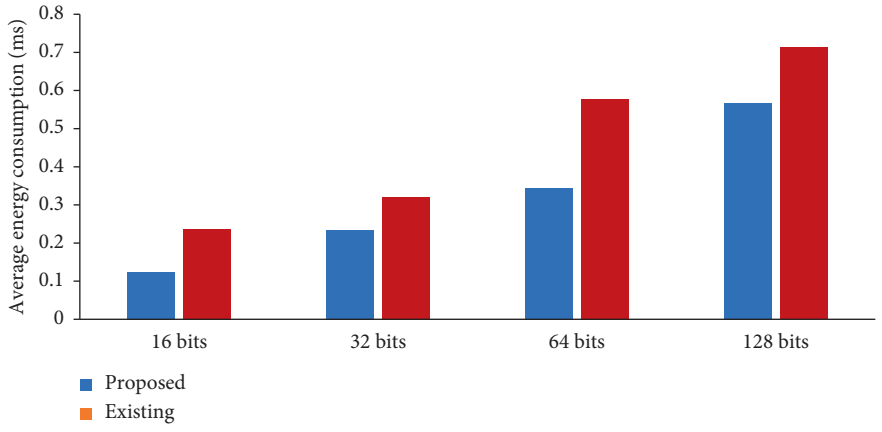

Figure 13: Energy consumption for different key AES encrypting and decrypting.

where *Ni* is the initial network usage (*Nu* at 0).

The network usage mathematical notation *Nu* is the number of bits *B* communicated in a certain time frame on devices under sets of data *D* with latency *L*. Simulation result is clear from Figure 14.

5.5. Average Networking Delay. In the calculation of testing and evaluating whether the data are secured, delay is likewise evaluated. In the local host cloud environment amount of consumers; the data traffic will develop tall, which will have influence on the scheme. In a real environment, numerous issues could cause delays, e.g., the size of the key and network speeds, which will cause suspensions and overcrowding. The larger numbers of key indicate increased delay due to the time when more data encrypt generate. When the key, it is originally split into dissimilar blocks formerly encryption. The scope of individual block may have contingent influence

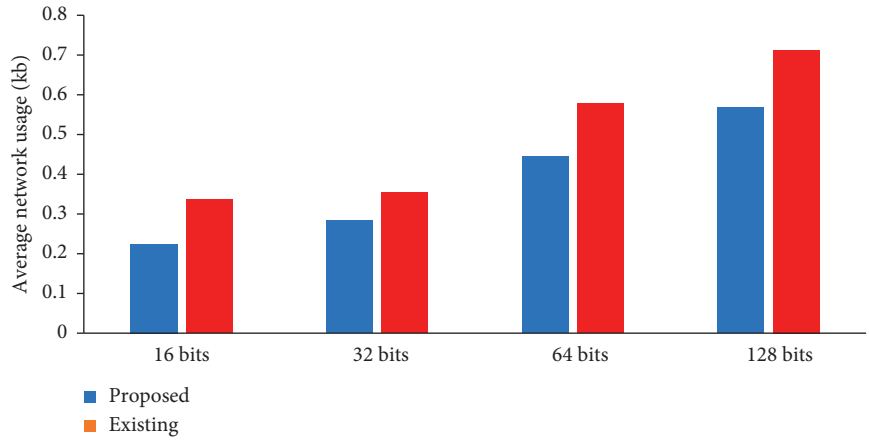

Figure 14: Network usage for different key AES encrypting and decrypting.

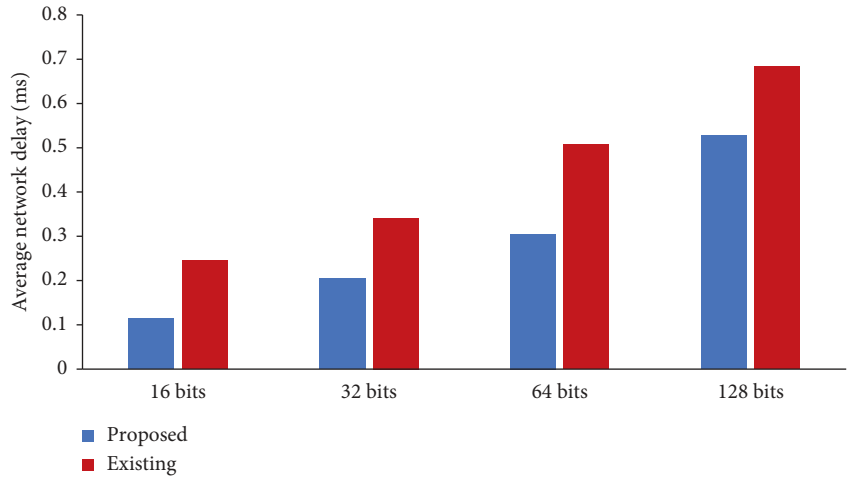

Figure 15: Networking delay for different key AES encrypting and decrypting.

on the scope. The delay comparison of the previous methodology [13] and the research shows that the significant differences in the delay indicate that the proposed framework is 15% better than the previous solution [13].

$$\sum D_n = B_s * \frac{L}{T} - B_d * \frac{L}{1} - \frac{T}{T_e}. \quad (7)$$

The delay D represents the time that the bits B take to reach a processing device from an end device under a certain latency L and connection time T. The observed delay is calculated using the equation. The mathematical notation to represent the delay is described below and by simulation result it is clear from figure. The delay calculation is shown in Figure 15.

6. Conclusion

To provide data confidentiality and information integrity of users' data in the cloud computing environment, an effective security framework is proposed that provides a mechanism through which communication is protected and unauthorized access is restricted. The proposed security framework allows cloud users to securely handle the privacy and integrity of data.

It also allows security, privacy, network usage, and storage in the cloud without depending on the plausibility of the cloud provider. The application of the AES algorithm provides a strong foundation that protects data stored in the cloud as well as authorizes access to data only on successful authentication and verification. The delays that occur in the actual environment vary in different situations all of which are not considered in this framework. Results show that the proposed framework minimizes energy consumption by 14.43%, network usage by 11.53%, and delay by 15.67%. Hence, the proposed framework enhances security, minimizes resource utilization, and reduces delay while deploying services of computational clouds.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G.S. Mahmood, J. H. Dong, and B. A. rahman Jaleel, "Achieving an effective, confidentiality and integrity of data in cloud computing," *International Journal of Network Security*, vol. 21, no. 2, pp. 326–332, 2019.

[2] S. Othman and A. S. Riaz, "A user-based trust model for cloud computing environment," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018.

[3] A. Firman, A. N. Hidayanto, and P. Harjanto, "Critical components of security framework for cloud computing community: a systematic literature review," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 3345–3358, 2018.

[4] K. V. Pradeep, V. Vijayakumar, and V. Subramaniyaswamy, "An efficient framework for sharing a file in a secure manner using asymmetric key distribution management in cloud environment," *Journal of Computer Networks and Communications*, vol. 2019, Article ID 9852472, 8 pages, 2019.

[5] Dr. Ramalingam Sugumar and K. Arul Marie Joycee, "FEDSACE: a framework for enhanced user data security algorithms in cloud computing environment," *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 4, no. 3, 2018.

[6] M. Kpelou and K. Kishore, "Lightweight security framework for data outsourcing and storage in mobile cloud computing," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, 2019.

[7] R. Ganga Sagar and N. Ashok Kumar, "Encryption based framework for cloud databases using AES algorithm," *International Journal of Research Studies in Computer Science and Engineering*, vol. 2, no. 6, 2015.

[8] J. R. Jain and A. Abu, "A novel data logging framework to enhance security of cloud computing," in *Proceedings of the SoutheastCon 2016,* IEEE, Norfolk, VA, USA, April 2016.

[9] J. Singh, "Framework for client side AES encryption technique in cloud computing," *IJIRMPS*, vol. 6, no. 5, 2018.

[10] J. Y. Gudapati Syam Prasad, S. sunil kumar, and A. Keerthi, "Integration of searching and AES encryption in cloud computing," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 4, 2019.

[11] I. A. Elgendy, W.-Z. Zhang, C.-y. Liu, and C.-H. hsu, "An efficient and secured framework for mobile cloud computing," *IEEE Transactions on Cloud Computing*, 2018.

[12] R. Saha, G. Geetha, G. Kumar, and T.-h. Kim, "RK-AES: an improved version of AES using a new key generation process with random keys," *Security and Communication Networks*, vol. 2018, Article ID 9802475, 11 pages, 2018.

[13] O. C. Abikoye, A. D. Haruna, A. Abubakar, N. O. Akande, and E. O. Asani, "Modified advanced encryption standard algorithm for information security," *Symmetry*, vol. 11, no. 12, p. 1484, 2019.

[14] K.-L. Tsai, Y.-L. Huang, F.-Y. Leu, I. You, Y.-L. Huang, and C.-H. Tsai, "AES-128 based secure low power communication for LoRaWAN IoT environments," *IEEE Access*, vol. 6, pp. 45325–45334, 2018.

[15] M.V. C. Suana, A. M. Sison, C. Aragon, and R. P. Medina, "Enhancement of advanced encryption standard (AES) cryptographic strength via generation of cipher key-dependent S-box," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 6, no. 4, 2018.

[16] S. NurRachmat, "Performance analysis of 256-bit AES encryption algorithm on android smart phone," *IOP Conf. Series: Journal of Physics: Conf. Series*, vol. 1196, 2019.

[17] J. Silki and V. Abhilasha, "An improved security framework for cloud environment using ECC algorithm," *International Journal for Research in Applied Science & Engineering Technology*, vol. 6, no. 1, 2018.

[18] A. Oussama and Z. Abdelha, "A security framework for cloud data storage (CDS) based on agent," *Applied Computational Intelligence and Mathematical Methods*, Springer, Berlin, Germany, 2019.

[19] H. J. Muhasin, R. Atan, M.A. Jabar, and S. Abdullah, "Cloud computing sensitive data protection using multi layered approach," in *Proceedings of the 2016 2nd International Conference on Science in Information Technology (ICSITech)*, pp. 69–73, Balikpapan, Indonesia, October 2016.

[20] K. Ravi and K. B. Rajesh, "Quality based cloud service broker for optimal cloud service provider selection," *International Journal of Applied Engineering Research*, vol. 12, no. 18, pp. 7962–7975, 2017.

[21] M. Adelmeyer, M. Walterbusch, B. Peter, and T. Frank, *Trust Transitivity and Trust Propagation in Cloud Computing Ecosystems*, Lawrence Erlbaum Associates, Mahwah, NJ, USA, 2018.

[22] F. Meng, R. Lin, Z. Wang, H. Zou1, and S. Zhou, "A multi-connection encryption algorithm applied in secure channel service system," *EAI Endorsed Transactions on Security and Safety*, vol. 5, no. 15, 2018.

[23] H. A. Al Essa and A. S. Ashoor, "Enhancing performance of AES algorithm using concurrency and multithreading," *ARPN Journal of Engineering and Applied Sciences*, vol. 14, no. 11, 2019.

[24] M. Marwan, A. Kartit, and H. Ouahmane, "A framework to secure medical image storage in cloud computing environment," *Journal of Electronic Commerce in Organizations*, vol. 16, no. 1, pp. 1–16, 2018.

[25] P. Sirohi and A. Agarwal, "Cloud computing data storage security framework relating to data integrity, privacy and trust," in *Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 4-5, Dehradun, India, September 2015.

[26] K. Subramanian, F. L. John, and F. L. John, "Dynamic and secure unstructured data sharing in multi-cloud storage using the hybrid crypto-system," *International Journal of Advanced and Applied Sciences*, vol. 5, no. 1, pp. 15–23, 2018.

[27] M. Edjie, D. L. Reyes, M. Ariel, Sison, and Dr.R. P. Medina, "Modified AES cipher round and key schedule," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 7, no. 1, March 2019.

[28] H. Talirongan, A. M. Sison, and R. P. Medina, "A new advanced encryption standard-butterfly effect in protecting image of copyright piracy," in *Proceedings of the Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, Hong Kong, China, December 2018.

[29] F. A. Hany, J. W. Robert, and B. W. Gary, "Fog computing and the internet of things: a review," *Big Data Cognitive Computer*, vol. 2, no. 2, 2018.

[30] A. Arab, M. J. Rostami, and B. Ghavami, "An image encryption method based on chaos system and AES algorithm," *The Journal of Supercomputing*, vol. 75, no. 10, pp. 6663–6682, 2019.

[31] M. Oqail Ahmad and R. Z. Khan, "Cloud computing modeling and simulation using CloudSim environment,"

International Journal of Recent Technology and Engineering (IJRTE) ISSN, vol. 8, no. 2, 2019.

[32] V. Surya, S. Ranichandra, and R. Ranjani, "Secure cloud storage using AES encryption," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 6, no. 6, 2018.

[33] A. Nair and S. S. SantoshAnand, "A performance booster for load balancing in cloud computing with my load balancer technique," *International Journal of Recent Technology and Engineering*, vol. 8, no. 1, 2019.

[34] D. Salama and A. Elminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *IJEIE*, vol. 8, no. 1, pp. 40–42, 2018.

[35] D.-H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X.-T. Tran, "Ultra low-power and low-energy 32-bit datapath AES architecture for IoT applications," in *Proceedings of the 2016 International Conference on IC Design and Technology (ICICDT)*, pp. 1–4, Ho Chi Minh City, Vietnam, June 2016.

[36] H. Jia, X. Liu, X. Di et al., "Security strategy for virtual machine allocation in cloud computing," *Procedia Computer Science*, vol. 147, pp. 140–144, 2019.

[37] B. T. Spiers, M. Halas, R. A. Schimmel, and D. P. Provencher, "Secure network cloud architecture," U.S. Patent 8,984,610, United States Patent (Justia Patents), 2015.

[38] E. Bertino, F. Paci, R. Ferrini, and N. Shang, "Privacy-preserving digital identity management for cloud computing," *IEEE Data Engineering Bulletin*, vol. 32, no. 1, pp. 21–27, 2009.

[39] S. Yi, Li Cheng, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, pp. 37–42, ACM, Hangzhou, China, June 2015.

[40] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proceedings of the 2014 International Conference on Future Internet of Things and Cloud*, pp. 464–470, IEEE, Barcelona, Spain, August 2014.

[41] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. &Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[42] G. N. Selimis, A. P. Kakarountas, A. P. Fournaris, A. Milidonis, and O. Koufopavlou, "A low power design for sbox cryptographic primitive of advanced encryption standard for mobile end-users," *Journal of Low Power Electronics*, vol. 3, no. 3, pp. 327–336, 2007.

[43] M. A. FaiqaMaqsood, M. M. Ali, and M. Ali Shah, "Cryptography: a comparative analysis for modern techniques", (IJACSA)," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, 2017.

[44] R. Paul, S. Saha, S. Sau, and A. Chakrabarti, "Design and implementation of realtime AES-128 on real time operating system for multiple fpga communication," 2012, http://arxiv.org/abs/1205.2153.

[45] D. Lohit Kumar, Dr.A. R. Reddy, and S. A. K. Jilani, "Implementation of 128-bit AES algorithm in MATLAB," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 33, no. 3, 2016.

[46] Dr. N. Suba Rani, Dr. A. Noble Mary Juliet, and K. Renuka Devi, "An image encryption & decryption and comparison with text - AES algorithm," *International Journal of Scientific & Technology Research*, vol. 8, no. 7, 2019.

[47] O. I. Omotosho, "A review on cloud computing security," *International Journal of Computer Science and Mobile Computing, IJCSMC*, vol. 8, no. 9, pp. 245–257, 2019.

[48] L. R1 and H. S2 Mohan, "Implementation and performance analysis of modified AES algorithm with key-dependent dynamic S-box and key multiplication," *Computer Applications Research*, vol. 5, no. 3, 2015.