A Lightweight Cipher for Balancing Security Trade-Off in Smart Healthcare Application

K. N. Sandhya Sarma, E. Chandra Blessie, and Hemraj Shobharam Lamkuche

Abstract IoT is a dynamic, emerging sector characterized by using existing Internet infrastructure to connect devices and data centers worldwide. The internet of healthcare things (IoHT) is a subclass of IoT that includes high-tech medical equipment that plays a crucial role in data monitoring, processing, storage, and transfer. It has novel challenges in data security. Several cryptographic techniques have been devised to safeguard the system against data misuse, alteration, and node tempering. Such cryptographic algorithms are inadequate because of the device's diminutive size, limited computing power, memory, and power resources. A lightweight cryptographic scheme based on symmetric cryptosystems powered by the Feistel structure is required to secure such a system. A lightweight cryptosystem is proposed in this paper for the security of the e-healthcare system. The proposed algorithm encompasses a compact S-box function, less round and addition substitution, and an XOR operation to generate encrypted information from plaintext. Additionally, a secure authentication mechanism based on the previously mentioned method is suggested for secure communication in the healthcare system. The cryptanalysis process and result state that our proposed system resists several cryptanalytic attacks which include brute force, avalanche effect, linear cryptanalysis, and biclique attack when compared with standard AES algorithm.

Keywords Cryptography · Lightweight algorithms · Internet of Things · Internet of healthcare things

K. N. Sandhya Sarma (✉)
Nehru College of Management, Thirumalayampalayam, India
e-mail: sandhyasvivek@gmail.com

E. Chandra Blessie
Coimbatore Institute of Technology, Coimbatore, India
e-mail: blessie@cit.edu.in

H. S. Lamkuche
School of Computing Science and Engineering, VIT Bhopal University, Kothrikalan Sehore 466114, India

A. E. Hassanien et al. (eds.), *International Conference on Innovative Computing and Communications*, Lecture Notes in Networks and Systems 731,
https://doi.org/10.1007/978-981-99-4071-4_42

1 Introduction

The Internet of Things, which links physical objects to the internet, has recently seen a sharp rise in internet usage. The Internet of Things has significantly enhanced task completion. Many applications of IoT can be found across a variety of industries, including personal use, manufacturing, education, home automation, environmental monitoring, and health care. Smart cities extensively use Internet of Things (IoT) that make people in smart cities access services quickly from anywhere, saving time and money. Although we might call IoT a gift, it can also pose a threat if suitable security measures are not implemented. The Internet of Things demands a continuous network. What if there is a network or power outage, for example? As a result, prophylactic actions are urgently required. IoT device's sensors provide data to the cloud, where it is analyzed by software and sent without the involvement of human or computer over a network. The data collected from the patient's body through sensors are highly sensitive and we must ensure that the information is kept secure. Conventional cryptographic methods cause high computational costs and are impractical for IoT devices. Lightweight cryptographic approaches are introduced to overcome these issues and making more applicable to these devices. A novel approach is described in this study with fewer rounds and addition substitutions, which lowers the computational cost and makes it more relevant to sensor devices.

2 Literature Review

Various strategies for developing cryptographic techniques for securing data communication in IoT devices have been proposed in numerous research studies. To hide the digital information given in medical images, cryptography methods and hybrid encryption algorithms were suggested by Elhoseny et al. [1]. The system incorporates asymmetric encryption (RSA) and symmetric encryption (AES). The article also intends to improve medical data security. Data can be sent safely by combining steganography and hybrid encryption methods. A healthcare system that enables secure transmission of medical data to Wireless Personal Area Networks from Wireless Body Area Networks was presented by Huang et al. [2]. To provide anonymity, homomorphic encryption based on a matrix technique is utilized. To provide security, the sensitive healthcare data are separated and scrambled using a little amount of data, as suggested by Bao et al. [3]. The scrambled data are stored in the cloud, while the tiny data are stored locally. The Internet of Things and cloud computing have combined to form the "cloud of things." The CoT designs and platforms, as well as CoT application in smart health care, are examined in an article by Mahmoud et al. [4]. The paper examines various approaches that are of energy efficiency in CoT for healthcare industry.

IoMT is a novel concept coined by Limaye et al. in their study [5]. The article sets the path for future studies into new optimization strategies that will allow for

the efficient execution of upcoming IoMT. Bandwidth bottlenecks are solved via edge computing. For IoMT, a benchmark suite called HERMIT is introduced, which includes applications from several sectors in health care. Fog computing was the main topic of Al Hamad et al.'s [6] research as a means of protecting sensitive healthcare data on the cloud. By utilizing bilinear pairing cryptography, he suggested a triple-party one-round authenticated key agreement mechanism. To store and access confidential healthcare data, the decoy approach is used. Attribute-based signature (ABS) is a useful technique for protecting user privacy. It is most appropriate approach for privacy access control and anonymous authentication. To ensure that users' sensitive information is kept private, LPPMSA system as suggested by Liu et al. [7] is utilized which is also based on multi authority ABS. The work done by Leila et al. [8] suggests a lightweight blockchain design for managing healthcare data that consume less computational resources and low latency as compared to the Bitcoin network. PCML [9], a unique technique suggested by Fengwei Wang and colleagues, addresses security issues. It also improves the accuracy of online medical diagnosis services.

Healthcare facilities can use the Paillier cryptosystem with distributed skyline computation and threshold decryption. Using these approaches with local diagnosis models, the global diagnosis model is learnt by system with the aid of cloud. As suggested by Yang et al. provide LIST [10], which enables rapid keyword searches while encrypting patient data, it creates an end-to-end channel from a patient's mobile device to data users. Attribute-based encryption is used in encrypted data to enable fine-grained access.

3 IoT-Enabled Healthcare System

3.1 Healthcare System Objectives

IoT technology can be used to create a variety of smart health applications, accomplishing the objectives of the healthcare system. Figure 1 shows the basic elements of the healthcare architecture as well as the integrated technologies that make up a healthcare system. Creating high-quality services that assist people with a range of healthcare requirements is one of the most crucial objectives. This calls for better system performance, effective resource management, and tool optimization. Automation and artificial intelligence can increase the precision of outcomes and accelerate routine, simple processes. Furthermore, instant medical care and ongoing monitoring can be provided with the use of remote access and real-time replies. Last but not the least, the development of relevant databases with comprehensive and conveniently available medical records may result in many more individualized treatment and better diagnosis.

Increasing operational effectiveness while maintaining low costs, resource usage, and energy consumption is another crucial objective. As a result, IoT devices with low resources and energy can be used to create healthcare applications. However,

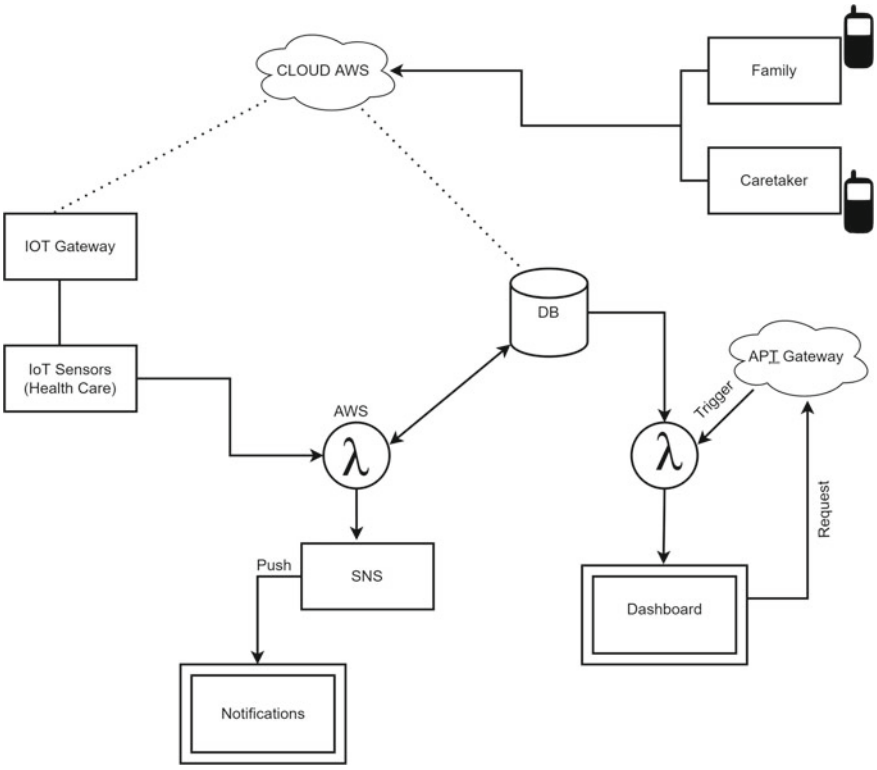


Fig. 1 Lightweight encryption-based AWS IoT healthcare system

the system requirements govern how throughput and resource consumption must be balanced. Because IoT devices often have limited resources and energy, healthcare applications can be integrated in them. The important balance between throughput and resource use is determined by the system requirements. Diagnoses are made simpler and more precise due to the real-time transmission, analysis, and storage of these data to cloud services. If expensive health check-up routines are substituted by less expensive, easily available, user-friendly, and quickly responsive alternatives, the demand for healthcare staff and resources will be lowered. Finally, information will be quickly and easily shared between different healthcare facilities and providers.

3.2 Connected Device's Architecture

The IoT architecture is described as a three-layer structure. The application layer followed by network layer and physical/perception layer forms the three tiers. In-depth representations of the IoT layers in a healthcare architecture are shown in Fig. 2.

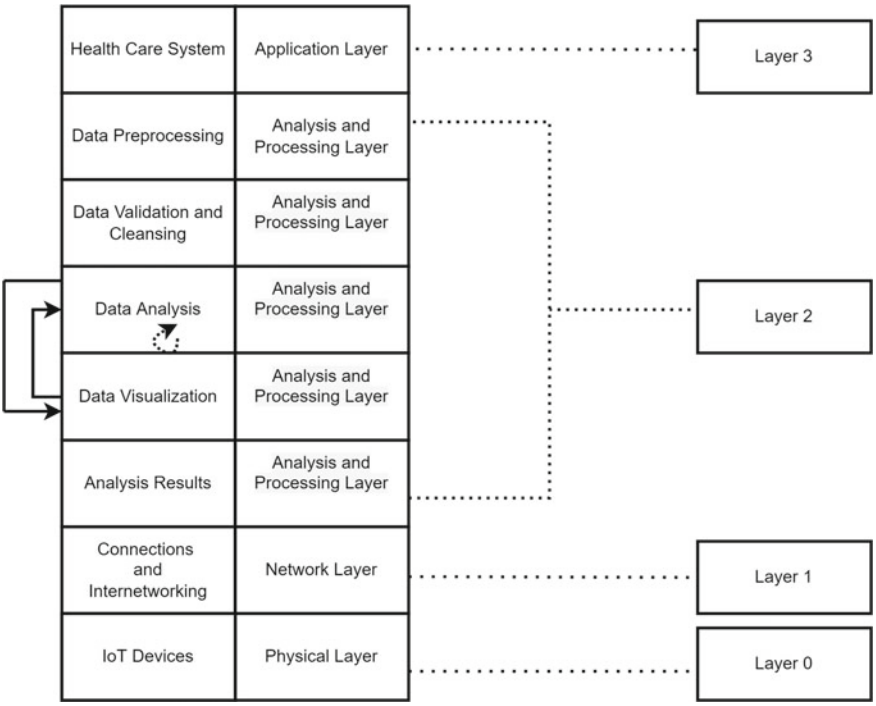


Fig. 2 Layered view of IoT healthcare system

The application layer links the items to the IoT network and forms the top layer in the architecture. It comprises several healthcare apps that offer the e-health features and services to the user. The network layer's communication protocols support the IoT components to connect with one another and share data that have been collected by physical layer devices. Some of the most widely used networks include Bluetooth, 5G, Wi-Fi, Radio Frequency Identification (RFID), ZigBee, and Lora WAN. Another popular node network incorporated into the IoT is the Wireless Sensor Network. The last layer of the architecture is the physical/perception layer. All of the tangible components utilized in IoT systems, such as sensors, wearables, actuators, smartphones, antennas, and CPUs are included in this. The objective of this layer is to collect health signals and covert them to information that the network layer can communicate.

3.3 IOT-Enabled Healthcare System Design

IoT technology has now made feasible to provide healthcare services outside the hospital. The key applications which include telemedicine, ambient-assisted living

(ALL) for the elderly or crippled, and remote health monitoring are some of the important applications that benefit the health care. By reducing the demand on hospital resources, they can, for instance, enhance the efficacy and accessibility of health care. Several writers have also studied the Internet of Things, ambient-assisted living, and remote healthcare monitoring systems. The researchers additionally proposed a decentralized approach for an IoMT-based smart health system. The data producer, hybrid computing, and data consumer layers make up this design. IoT sensors make up the first layer, which produces health data that are then gathered and sent to a hybrid computing system that makes use of both edge and cloud paradigms. Blockchain technology and the Distributed Data Storage System (DDSS) approach enable decentralized data processing. Furthermore, three cryptography algorithms are used to establish system privacy and security. To monetize acquired health data and offer data security and privacy rules, another decentralized healthcare architecture was exhibited. IoT, AI, Big Data, and Blockchain technologies were all completely described, as were all of the architecture's elements. There was a careful examination of each of its layers and critical technologies. Services like telemedicine, emotion interaction, and smart garment monitoring have all made use of this concept. Multiple sensors were coupled to a primary CPU in a hardware-based implementation developed by researchers, and the system was utilized to continuously monitor health-related factors.

3.4 Security Consideration in Smart Healthcare System

As was already said, security is a crucial factor to consider while implementing IoT-based healthcare applications. However, as demonstrated in Fig. 3 (SSH—Secure Healthcare Architecture), the most vulnerable component of IoT networks used in general smart health infrastructure is the IoT device. Through the IoT network, an attacker can quickly get any personal information shared by devices. The two most common attacks on data privacy are eavesdropping and data transmission/traffic monitoring. Furthermore, user authentication is hampered in the absence of effective data protection. Unauthorized devices may gain access to these data, which they may subsequently process or modify to harm others. Also, they can generate bogus medical information and communicate it to other IoT devices. As a result, the patient and the healthcare professional communicate in an unreliable manner and receive false health diagnoses.

Researchers are very interested in creating secure communication networks. A common technique for protecting privacy of data and verifying user authentication is cryptography. It uses cryptographic techniques, specifically ciphers, to encrypt and then decrypt data to hide the content of different messages. However, due to the device resource limitations, known cryptographic primitives cannot be applied in the IoT system. The encryption that is being used must not take away resources from other vital tasks that provide healthcare services. To fully correlate to IoT hardware restrictions, a more lightweight version must be created. Additionally, the delayed

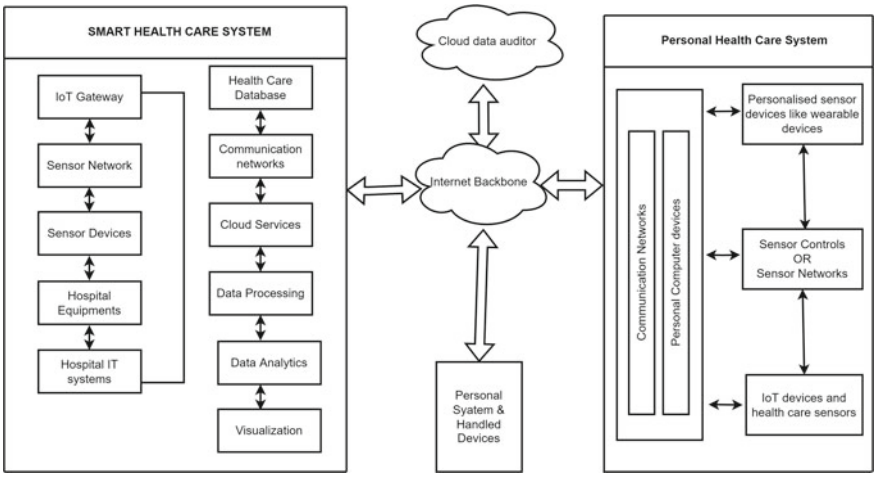


Fig. 3 SSH: secure healthcare architecture

implementation of the algorithms can have fatal results in crucial circumstances when real-time applications are affected. The speed of encryption and decryption as well as the quick response time of the system must be considered. Finally, the system must provide a variety of options for each feature, supporting various security and networking requirements, based on the current requirements of the application. As a result, the system must be expanded with mechanisms for flexibility and scalability. Overall, a security plan and a lightweight cryptographic primitive must be offered to sufficiently safeguard the health data of a smart health application. Before the acquired data can be exchanged across the IoT devices, it must first be encrypted by the encryption technique. As a result, hostile assaults cannot access the patient's personal information. The received data must also be decrypted by the decryption method before it can be used in the healthcare application. Hence, all communication networks employed, notably IoT networks and the cloud-based services connected to the Internet, entirely protect the transferred data. Figure 4 depicts a generic intelligent health infrastructure that might apply a lightweight security plan while meeting all the above goals (an upgraded lightweight cipher-based security scheme).

4 SSH—A Lightweight Cipher

SSH, an improved lightweight security primitive that proficiently encrypts and decrypts the gathered information while providing better flexibility for key size and operational speed, is used in the suggested lightweight-based security system. This method, which safeguards data on communication networks and the Internet, is built into every sensor node used in a medical system, including smart hospitals and secure

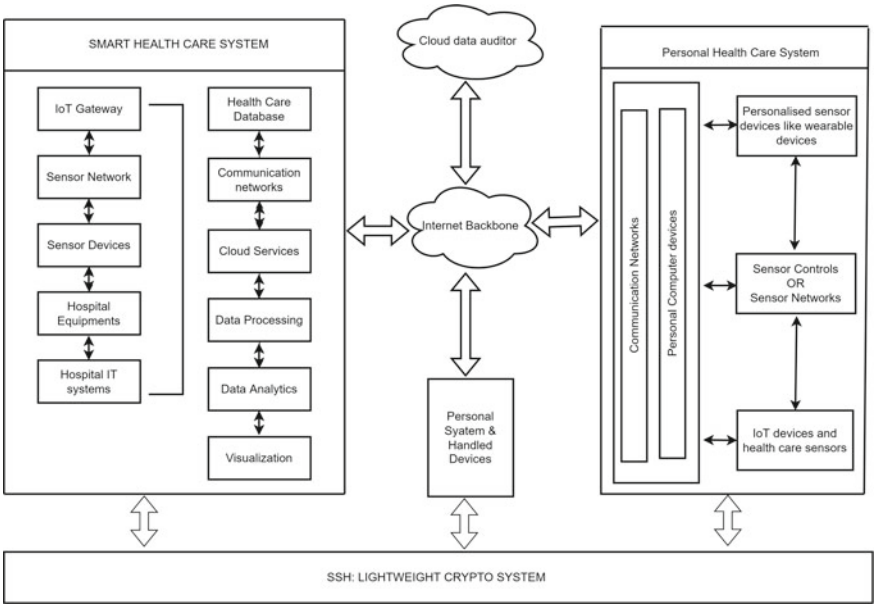


Fig. 4 SSH: an enhanced lightweight cipher base security scheme

patient setups. In conclusion, the suggested lightweight-based security method, as depicted in Fig. 4, can be implemented to safeguard sensitive data via the internet.

4.1 SSH—A Lightweight Cipher

Compared to other widely used encryption primitives, SSH is a basic fundamental for generating keys, encrypting data, and decrypting data. Compatibility with the complex communication requirements of the IoT-based healthcare system is one advantage of a lightweight design. It specifically offers rapid hardware resource-efficient end-to-end communication. SSH is entirely based on symmetric encryption schemes, with the goal of preserving trade-offs such as cost, performance, and security in hardware and software implementations. Symmetric cryptography improves operational speed while requiring fewer computing resources. Figure 5 depicts the detailed process of the SSH cryptosystem scheme.

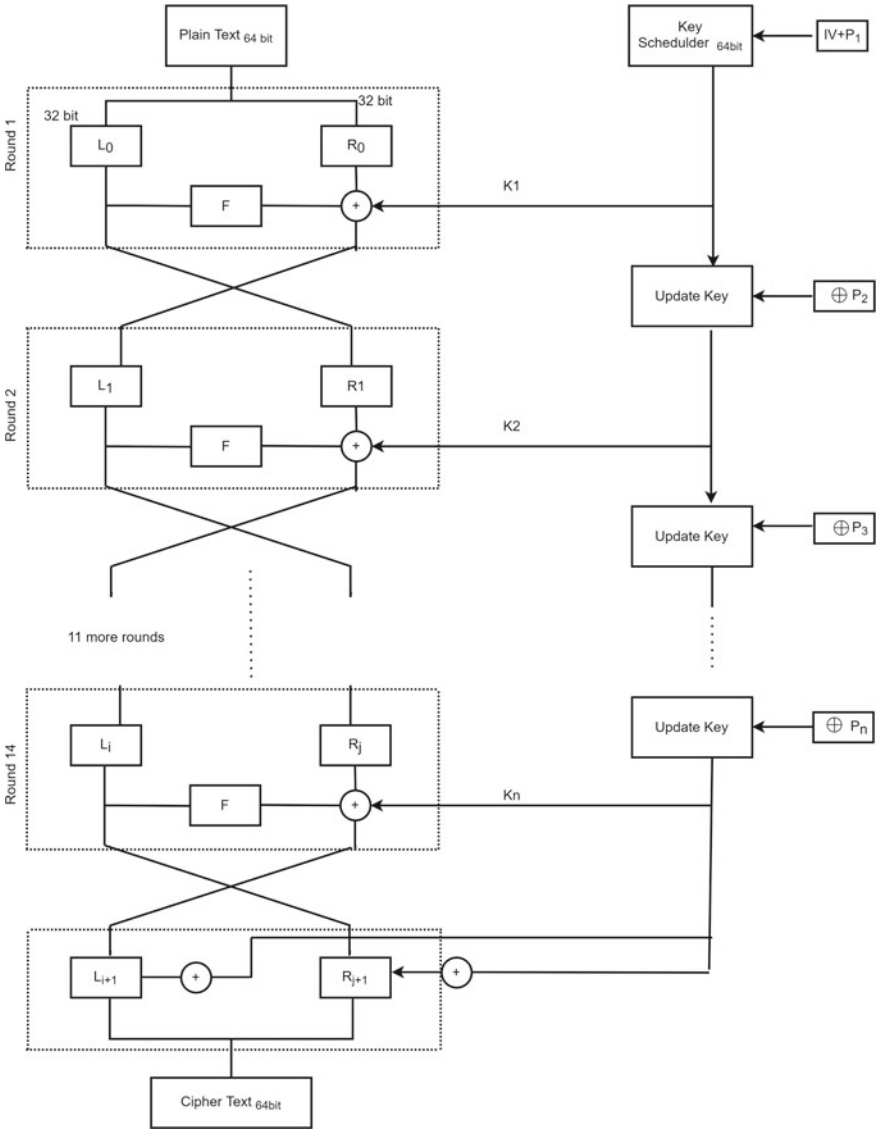


Fig. 5 SSH: Feistel structure

4.2 Implementation of SSH

SSH is a lightweight cryptographic primitive based on Horst Feistel structure. It takes the input of 64-bit block size of plaintext information and converts it into 64-bit block size of ciphertext by performing 14 rounds in Feistel structure. Each round is composed of key schedule operations, F-function which integrates S-box

operations on the plaintext, then a unique key is XORed in each round of Feistel round to generate a complex ciphertext as output of the round. The same cipher is again forwarded to next round as input text to generate ciphertext by applying the above process consecutively for straight 14 rounds. Starting with a stationary 64-bit block of plaintext, the encryption process splits it into two 32-bit shards. In each cycle of the encryption system, the Feistel function F() and a private key with a size ranging from 64 to 128 bits are employed. A 32-bit cipher is produced as the output of four 8-bit s-boxes at the end of each round of the Feistel function by combining an optimized F-function with a lightened 8-bit s-box function at electronic speed. At the conclusion of the 14th round of the Feistel structure in cipher block chaining mode, the resultant output, two 32-bit halves, is then swapped and combined to obtain the required 64-bit ciphertext.

Algorithm 1: SSH-Encryption Pseudocode

Let T be the 64-bit plaintext input
Divide the plaintext T into TLEFT, TRIGHT, 4 bytes each
for i varying from 1 to 14: do
TLEFT = TLEFT XOR P(i)
TRIGHT = TRIGHT XOR (P (i) XOR F (TLEFT))
Swap TLEFT and TRIGHT
end for
Interchange TLEFT and TRIGHT (Reverse the most recent switch)
TLEFT = TLEFT XOR P15
TRIGHT = TRIGHT XOR P16
Switch TLEFT and TRIGHT
TLEFT = TLEFT XOR P17
TRIGHT = TRIGHT XOR P18
TLEFT and TRIGHT should be combined again to create 64-bit ciphertext
Function Box-F ():
F(PT): ((S1(a, b) + S2(a, b)) XOR (S3(a, b) + S4(a, b)))

End Encryption Algorithm
The pseudocode for SSH-decryption is given below:

Algorithm 2: SSH-Decryption Pseudocode

Ciphertext input if 64-bit (T)
Divide the ciphertext T into TLEFT and TRIGHT, 4bytes each
TLEFT = TLEFT XOR P18
TRIGHT = TRIGHT XOR P17
Swap TLEFT and TRIGHT
TRIGHT = TRIGHT XOR P16
TLEFT = TLEFT XOR P15
for i varying from 14 to 1: do
TLEFT = TLEFT XOR (P (i) XOR F (TRIGHT))
TRIGHT = TRIGHT XOR P (i)
Swap TLEFT and TRIGHT (Undo the last swap)

End For
TLEFT and TRIGHT should be combined again to create 64-bit original plaintext
Function Box–F ():
F(T) = ((S1(a, b) + S2(a, b)) XOR (S3(a, b) + S4(a, b)))

End Decryption Algorithm

The proposed methodology also uses the concept of homomorphic cryptosystem in which any encryption algorithm that exhibits homomorphic properties can be called a homomorphic encryption algorithm. Homomorphic properties include addition, multiplication, or both. A single operation like multiplication or addition but not both can be executed in Partial Homomorphic Encryption Scheme [PHE]. Despite being able to perform several operations, Somewhat Homomorphic Encryption (SWHE) approaches can only support a certain amount of addition and multiplications. Fully homomorphic encryption (FHE) refers to a cryptosystem that supports addition and multiplication as well as the computation of any function [11].

4.3 Paillier Cryptosystem

Pascal Paillier created a partial homomorphic encryption in 1999. For public key encryption, it is a probabilistic asymmetric algorithm. E-voting system and threshold schemes are just a couple of the uses for the Paillier cryptosystem's additive homomorphic characteristic [12].

4.4 Pseudo Code for Paillier Algorithm

Step-1 Compute the product, $n = p*q$ after choosing two big primes p and q.
Step-2 : A semirandom, nonzero integer, g in Z_n^* must be selected, such that the order of g is a multiple of n in Z_n^{2*}.
Step-3 : Let msg be a message to be encrypted where msg $\in Z_n$.
Step-4 : Select a random integer r where $r \in Zn^*$
Step-5: Create the ciphertext as $c = g^{\text{msg}}. r^n. \bmod n^2$.
Step-6: Public key used is (n, g).
Step-7: Private key used is (χ, μ).

Where $\chi = \text{LCM}\ (p^{-1}, q^{-1})$ and $\mu = (L(g^{\chi} \bmod n^2))^{-1} \bmod n$.
Where function L is defined as $L(x) = (x-1)/n$.
Decryption in Paillier cryptosystem is one exponentiation modulo n^2 [13].

Novelty: The novelty of our proposed system is to use hybrid approach using Paillier cryptosystem and SSH cryptosystem. The proposed system encompasses of quad steps to execute secure communication of sensitive information over cloud. Paillier cryptosystem uses different keys for encoding and decoding, which is comparable to

RSA. It is bendable and resistant to specific plaintext assaults. It is used in e-voting, e-cash, and storing sensitive healthcare information over cloud.

By including encryption techniques for data transmission in cloud-based frameworks, the foreseen protocol for hybrid cryptography aims to develop a strong and reliable encryption algorithm. The proposed hybrid system compares the input and output of several encryption approaches to the current hybrid method. The proposed hybrid approach considers the following encryption algorithm combinations: Paillier, as well as the SSH cryptosystem. Cloud storage of encrypted data is utilizing the Paillier–SSH hybrid approach with no attack blocking constraints. Paillier–SSH encryption is used to collect encrypted data without compression through cloud storage. Attacks can be stopped using a firewall while accumulating encrypted data in the cloud utilizing Paillier–SSH encryption with compression and blocking rules.

The data are always partitioned by its type before the user stores his data in the cloud at the time of uploading. The suggested hybrid cryptographic scheme is divided into two phases: encryption and decryption. Each user's partitioned files are secure thanks to the offered encryption technique. The hybrid cryptographic algorithm is used to encrypt the partitioned files. The individual data components are merged at the server and forwarded to the decryption block. The section on implementation and results shows how the encryption process was carried out using both hybrid techniques. The files that have been partitioned are decrypted during the decryption process to increase security. The size of the file and a related hardness index of the selected systems are considered as comparative criteria, together with the encryption and decryption times. The decryption procedure converts ciphertext back to its original clear text in the opposite direction. The system's overall design is shown in the implementation if a server is attacked by an intruder. The dependable third party can be reached from both the client side and the server end. Integrity verification is one of the trusted third-party features, and it is done by creating a hash value for the data using the BLAKE-3 algorithm and comparing it to a hash value created at the client end. Every inconsistency between the two entries alerts the client. User-created firewall rules and policies were used to block attacks by employing iptables.

For this research, we believe trusted DBMS which is a promising goal for storing structured data in DynamoDB using AWS cloud services. As previously stated, one technical basis for the trust could be a moving target defense. A trusted cloud DBMS should not be expected to be secure once and for all. A DynamoDB that is trusted by some may not be trusted by others. Stronger security will almost certainly come at a cost. As for universally trusted operating systems, browsers, and virtual machines, the trusted DynamoDB technology will eventually face an endless race between proposals, exploits, patches to the exploits, and so on. The trusted DynamoDB may send the selected data as ciphertext, plaintext, or a combination of the two. The client must decrypt the ciphertext after it has been encrypted. In contrast to the traditional paradigm, a client of a trusted DynamoDB may request the entire decryption at the cloud. Avoiding the decryption burden securely may clearly make many clients happy. In methodology, even popular browsers (such as Chrome, Edge, and Brave)

should be capable of acting as clients for a trusted DynamoDB. All of these appear to be a benefit of the new approach.

Personal information captured by IoT devices is transmitted to the network via smart phones. When uploading data to the cloud, instead of putting the entire dataset into the cloud, use the AWS services. To store these data, AWS provides a cloud-based infrastructure platform that is highly reliable, scalable, and low cost. A compute service without a server which is AWS Lambda automatically manages resources on demand and executes code, lowering costs. Lambda functions, unlike traditional servers, do not run indefinitely. A change in the AWS environment is an event that can cause the function to be triggered. In the case of health care, a critical value in a patient's medical data can trigger an event and an action in Amazon Simple Notification Service (SNS). The data in the database can be monitored and AWS IoT events' alarms can be set for changes. Alarms that send notifications when a threshold is breached can be set.

IoT in health care itself is a vital area. IoT devices are power-constrained devices. When they transmit the data to devices like activators, they consume more energy. We must implement algorithms which consume less power in providing security to such devices. The whole process can be articulated in layers as shown in figure below. Layer 0, device layer where the data is recorded. Layer 1, Communication layer transmits the data to the cloud. The data transmitted should be protected from cyberattack like man in middle, DDOs, intruders intercepting or sniffing the data and successfully fabricating the data. Here, the data are related to health care, and integrity of data should not be compromised. NIST states that whatever data have been transmitted should strictly abide CIA trait—confidentiality, integrity, and availability. The next layer 2 is where data are processed and analyzed before transmitting to the doctors or health care. Layer3, application layer is where the data are received. The patient's details need not be revealed to the healthcare system. According to the guideline of WHO, the identity of a patient should be kept confidential and there is no need to reveal the identity of any person in any scenario.

5 Result and Analysis

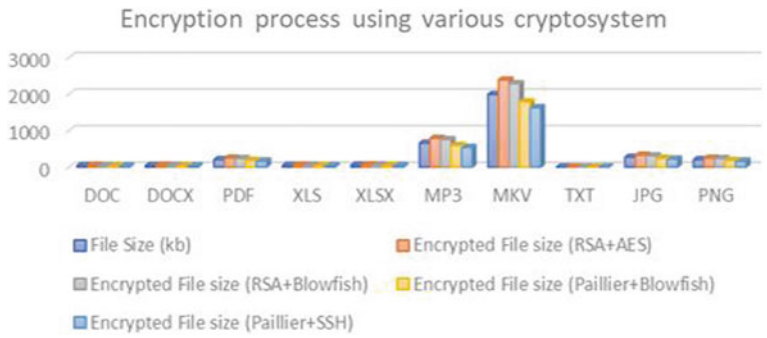
We have implemented a test version of proposed work and evaluated the data using Ubuntu OS on VMware Workstation Pro 16. Python3 has been used to implement the SSH cryptosystem. Due to their accessibility and policies that are adapted to the infrastructure offered in free-tier service level agreements, AWS's solutions were chosen for the installation of real-time cloud security. The hardening index is a unique indicator that assesses how well security vulnerabilities have been reduced in the system. It is calculated by using an information security software called Lynis. Lynis is an open-sourced shell script that supports various plugins, customs, and compliance checks and provides warnings and suggestions, as well as detailed system logs based on the security tests. ArcherySec4 has been used to assess and manage

vulnerabilities. The Paillier–SSH cryptographic implementation was tested using Apache NetBeans IDE 12.5.

Clients are thought to receive enough security assurances from the specified protocol if cloud service providers take strict action against malicious or illegal users. In the face of both internal and external threats, the truthfulness, obtainability, and privacy security restrictions were validated. The Lynis security auditing program was also utilized to comprehend specific general data, vulnerable software packages, and any configuration concerns in both suggested hybrid ways. It gives a general overview of the system components that represent the biggest security threats and are thus top priorities for initiatives aimed at hardening them. Using a set of parameters, we evaluated the performance of several methods. We divided them into two categories: computational overhead parameters and performance parameters. The computation overhead settings aim to minimize the time complexity and space complexity in software and hardware implementation of Paillier–SSH cryptosystem, while the performance parameter helps us to obtain low latency and high throughputs. The data shown in Tables 1 and 2 depict Paillier–SSH cryptosystem perform better when compared with standard algorithms like AES, RSA, and Blowfish encryption and decryption schemes.

Table 1 Encryption process using various cryptosystem

File type	File size (kb)	Encrypted file size (RSA + AES)	Encrypted file size (RSA + Blowfish)	Encrypted file size (Paillier + Blowfish)	Encrypted file size (Paillier + SSH)
DOC	38.6	46.32	44.39	34.74	31.652
DOCX	42.5	51	48.875	38.25	34.85
PDF	202.6	243.12	232.99	182.34	166.132
XLS	46.3	55.56	53.245	41.67	37.966
XLSX	52.7	63.24	60.605	47.43	43.214
MP3	655.5	786.6	753.825	589.95	537.51
MKV	1985.5	2382.6	2283.325	1786.95	1628.11
TXT	2	2.4	2.3	1.8	1.64
JPG	269.5	323.4	309.925	242.55	220.99
PNG	202.5	243	232.875	182.25	166.05



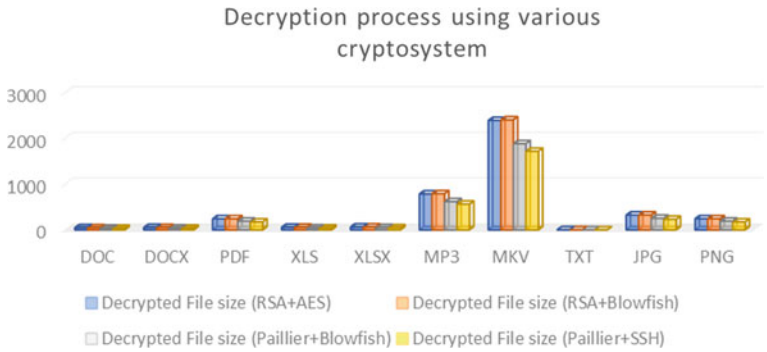


Table 2 Decryption process using various cryptosystem

File type	Decrypted file size (RSA + AES)	Decrypted file size (RSA + Blowfish)	Decrypted file size (Paillier + Blowfish)	Decrypted file size (Paillier + SSH)
DOC	47.38	46.6095	36.477	33.2346
DOCX	52.65	51.31875	40.1625	36.5925
PDF	244.12	244.6395	191.457	174.4386
XLS	56.72	55.90725	43.7535	39.8643
XLSX	63.94	63.63525	49.8015	45.3747
MP3	787.96	791.51625	619.4475	564.3855
MKV	2384.85	2397.49125	1876.2975	1709.5155
TXT	2.8	2.415	1.89	1.722
JPG	324.76	325.42125	254.6775	232.0395
PNG	244.45	244.51875	191.3625	174.3525

The file execution time while employing the Paillier and Blowfish, Paillier–SSH with and without compression, and RSA–AES cryptosystem approaches is shown in Tables 3 and 4. The amount of time required to encode data so that only authorized users may access it is known as the encryption time. Decryption time is the amount of time needed to undo encryption or to change encoded data back into its native format. We used calculations designed for the selected encryption technique to compute time. Execution time for Paillier–Blowfish, RSA–AES, and Paillier–SSH encryption and decryption (with and without compression) is also tabulated.

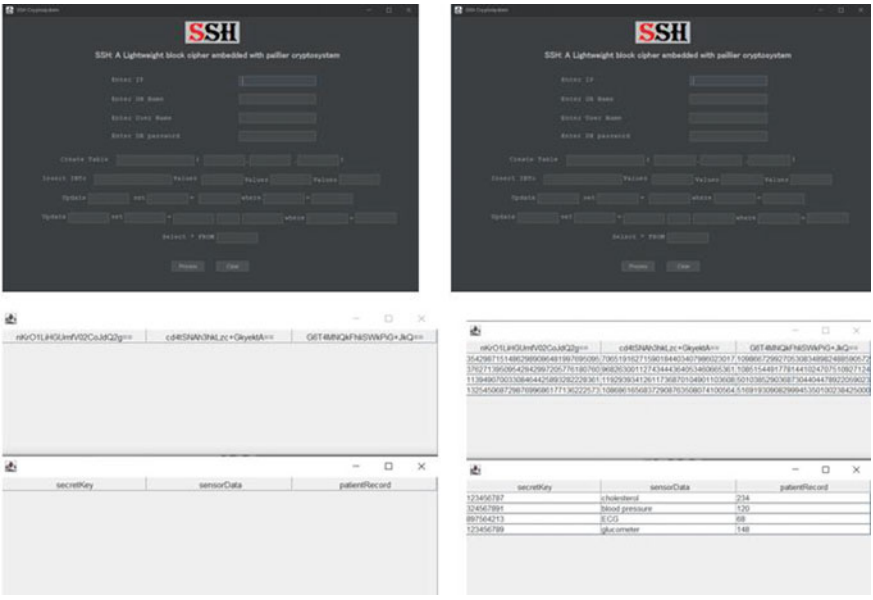


Table 3 Table throughput (kbps) comparison of various cryptosystems

Modes	AES	Blowfish	Paillier	SSH
ECB	77	78	78	85
CBC	72	71	71.5	74.65
CFB	70	69	70.65	73.65
OFB	69	68	69.5	71.45
CTR	74	73	74	78.35

Table 4 Execution time (encryption and decryption)

		RSA–AES			RSA–Blowfish			Paillier–Blowfish			Paillier–SSH		
File type	File size	Encryption	Decryption	Total	Encryption	Decryption	Total	Encryption	Decryption	Total	Encryption	Decryption	Total
DOC	38.6	46.32	47.38	93.7	44.39	46.6095	90.9995	34.74	36.477	71.217	31.652	33.2346	64.8866
DOCX	42.5	51	52.65	103.65	48.875	51.31875	100.19375	38.25	40.1625	78.4125	34.85	36.5925	71.4425
PDF	202.6	243.12	244.12	487.24	232.99	244.6395	477.6295	182.34	191.457	373.797	166.132	174.4386	340.5706
XLS	46.3	55.56	56.72	112.28	53.245	55.90725	109.15225	41.67	43.7535	85.4235	37.966	39.8643	77.8303
XLSX	52.7	63.24	63.94	127.18	60.605	63.63525	124.24025	47.43	49.8015	97.2315	43.214	45.3747	88.5887
MP3	655.5	786.6	787.96	1574.56	753.825	791.51625	1545.34125	589.95	619.4475	1209.3975	537.51	564.3855	1101.896
MKV	1985.5	2382.6	2384.85	4767.45	2283.325	2397.49125	4680.81625	1786.95	1876.2975	3663.2475	1628.11	1709.5155	3337.626
TXT	2	2.4	2.8	5.2	2.3	2.415	4.715	1.8	1.89	3.69	1.64	1.722	3.362
JPG	269.5	323.4	324.76	648.16	309.925	325.42125	635.34625	242.55	254.6775	497.2275	220.99	232.0395	453.0295
PNG	202.5	243	244.45	487.45	232.875	244.51875	477.39375	182.25	191.3625	373.6125	166.05	174.3525	340.4025

6 Conclusions

Our problem was to identify a gap in preserving the security and privacy of data generated in sensor devices which are then transmitted to remote systems or intermediate systems and later to the cloud. It is our responsibility to protect the security and privacy of such gadgets. This study suggests an encryption strategy to maintain the security of private healthcare data stored in the cloud. The proposed SSH symmetric encryption scheme solves and balances the trade-offs related to cost, performance, and security. The SSH lightweight encryption cryptosystem practically consumes less power, area, and cost when implemented on IoT controller like Arduino, Raspberry Pi, Beagle bone Black Rev C, and FPGA devices. The proposed approach is capable of resisting against known cryptanalyst attacks like Avalanche, Brute force, Plaintext attack, linear cryptanalysis, differential cryptanalysis, rectangle attack, side channel attack, and meet-in-the-middle attack. This system can further be improved to large scale by integrating emerging technologies like blockchain and artificial intelligence.

References

1. Elhoseny M, Ramirez-Gonzalez G, Osama M, AbuElnasr, Shawkat SA, ArunKumar N, Farouk A (2018) Secure medical data transmission model for IoT—based healthcare systems. IEEE Access Inform Secur Telecommun Appl 6
2. Huang H, Gong T, Ye N, Wang R, Dou Y (2017) Private and secured medical data transmission and analysis for wireless sensing health care system. IEEE Trans Indus Inform 13(3)
3. Bao S-D, Chen M, Guang-Zhong (2017) A method of signal scrambling to secure data storage for healthcare applications. IEEE Trans Biomed Health Inform 21(6)
4. Mahmoud MM, Rodrigues JJPC, Ahmed SH, Shah SC, Al-Muhtadi JF (2018) Enabling technologies on cloud of things for smart healthcare. IEEE Access Cyber Threats, Countermeasures in Healthcare Sector 6
5. Limaye A, Adegbija T (2018) HERMIT: a benchmark suite for the internet of medical things. IEEE Trans Internet of Things 5
6. Al Hamid HA, Md Mianur Rahman SK, Shamim Hossain M, Almogren A, Alamri A (2017) A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing based cryptography. IEEE Access 5
7. Liu J, Tang H, Sun R, Du X, Guizani M (2019) Ligtweight and privacy—preserving medical service access for healthare cloud . LPP-MSA.IEEE Access 7
8. Ismail L, Materwala H, Zeadally S (2019) Lightweight blockchain for healthcare. IEEE Access 7
9. Wang E, Zhu H, Liu X, Lu R, Hua J, Li H (2019) Privacy-preserving collaborative model learning scheme for EHealthcare. IEEE Access 7
10. Yang Y, Liu X, Deng RH, Li Y (2020) Light weight shareable and traceable secure mobile heath system. IEEE Trans Dependable and Secure Comput 17
11. Ogburn M, Turner C, Dahal P (2013) Homomorphic encryption. Elsevier
12. Kouichi S, Takagi T (2002) On the security of a modified paillier public key primitive. Information security and privacy. Springer, Berlin, Heidelberg
13. Pascal P (1999) Public—key cryptosystems based on composite degree residuosity classes. In: Advances in cryptology—EUROCRYPT'99, Springer, pp 223–238

14. Mehmood A, Natgunanathan I, Xiang Y, Poston H, Zhang Y (2018) Anonymous authentication scheme for smart cloud based healthcare application. IEEE Access 6
15. Kwabena O-A, Qin Z, Zhuang T, Qin Z (2019) MSCryptoNet: multischeme privacy—preserving deep learning in cloud computing. IEEE Access 7
16. Ali M, Sadeghi M-R, Liu X (2020) Lightweight revocable hierarchical attribute-based encryption for internet of things. IEEE Access 8:23951–23964. https://doi.org/10.1109/access.2020.2969957
17. Chen X, Liu Y, Chao HC, Li Y (2020) Ciphertext-policy hierarchical attribute-based encryption against key-delegation abuse for IoT-connected healthcare system. IEEE Access 8:86630–86650. https://doi.org/10.1109/ACCESS.2020.2986381
18. Goodman N, Zwick A, Spicer Z, Carlsen N (2020) Public engagement in smart city development: lessons from communities in Canada's smart city challenge. The Canadian Geographer/Le Géographe Canadien. https://doi.org/10.1111/cag.12607
19. Jawhar I, Mohamed N, Al-Jaroodi J (2018) Networking architectures and protocols for smart city systems. J Internet Serv Appl 9(1). https://doi.org/10.1186/s13174-018-0097-0
20. Kalmeshwar M, K S, APD, Prasad N (2017) Internet of things: architecture, issues and applications. Int J Eng Res Appl 07(06):85–88. https://doi.org/10.9790/9622-0706048588
21. Liao TL, Lin HR, Wan PY, Yan JJ (2019) Improved attribute-based encryption using chaos synchronization and its application to MQTT security. Appl Sci (Switzerland) 9(20). https://doi.org/10.3390/app9204454
22. Peralta G, Cid-Fuentes RG, Bilbao J, Crespo PM (2019) Homomorphic encryption and network coding in IoT architectures: advantages and future challenges. Electronics (Switzerland) 8(8):1–14. https://doi.org/10.3390/electronics8080827
23. Rasori M, Perazzo P, Dini G (2020) A lightweight and scalable attribute-based encryption system for smart cities. Comput Commun 149:78–89. https://doi.org/10.1016/j.comcom.2019.10.005
24. Sarma KS, Larnkuche HS, Umarnaheswari S (2013) A review of secret sharing schemes. Res J Inf Technol 5(2):67–72
25. Lamkuche HS, Pramod D, Onker V, Katiya S, Lamkuche G, Hiremath GR (2019) SAL–a lightweight symmetric cipher for internet of things. Int J Innov Technol Explor Eng 8(11):521–528
26. Murthy H, Lamkuche H (2022) Harnessing the power of ML and NLP for decision making in education sector from social media data. Cardiometry 22:415–420
27. Lamkuche HS, Pramod D (2020) CSL: FPGA implementation of lightweight block cipher for power-constrained devices. Int J Inf Comput Secur 12(2–3):349–377
28. Lamkuche HS, Kondaveety VB, Sapparam VL, Singh S, Rajpurkar RD (2022) Enhancing the security and performance of cloud for E-governance infrastructure: secure E-MODI. Int J Cloud Appl Comput (IJCAC) 12(1):1–23
29. Ramesh A, Pradhan V, Lamkuche H. (2021) Understanding and analysing resource utilization, costing strategies and pricing models in cloud computing. J Phys: Conf Ser 1964(4):042049. IOP Publishing
30. Gaikwad D, Lamkuche H (2021) Segmentation of services provided by e-commerce platforms using PAM clustering. J Phys: Conf Ser 1964(4):042036. IOP Publishing
31. Kumar S, Kumar D, Lamkuche HS (2021) TPA auditing to enhance the privacy and security in cloud systems. J Cyber Secur Mobility 10(3):537–568
32. Lamkuche HS, Singh K, Shirkhedkar K (2022) A lightweight block cipher for cloud-based healthcare systems. In: Panda SK, Rout RR, Sadam RC, Rayanoothala BVS, Li KC, Buyya R (eds) Computing, communication and learning. CoCoLe 2022. Communications in computer and information science, vol 1729. Springer, Cham. https://doi.org/10.1007/978-3-031-21750-0_1
33. Jhanwar N, Goel P, Lamkuche H (2022) Telecommunication stocks prediction using long short-term memory model neural network. In: Panda SK, Rout RR, Sadam RC, Rayanoothala BVS, Li KC, Buyya R (eds) Computing, communication and learning. CoCoLe 2022. Communications in computer and information science, vol 1729. Springer, Cham. https://doi.org/10.1007/978-3-031-21750-0_26