International Journal of Safety and Security Engineering
Vol. 13, No. 5, October, 2023, pp. 893-902
Journal homepage: http://iieta.org/journals/ijsse

Design of Mutual Authentication Method for Deep Learning Based Hybrid Cryptography to Secure data in Cloud Computing

Anwar Ali Mohd[1], Sandhya Kummarikunta[2], Siva Kumar Thumboor Naga[3], Venkateswara Reddy Buthukuri[4], Phanikanth Chintamaneni[5], Ramesh Vatambeti[6*]

[1] Department of Information Technology, MLR Institute of Technology, Hyderabad 500043, India
[2] School of Computing, Mohan Babu University, Tirupati 517102, India
[3] Department of Artificial Intelligence and Data Science, Vishnu Institute of Technology, Bhimavaram 534202, India
[4] Department of CSE, Chalapathi Institute of Technology, Guntur 522016, India
[5] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522302, India
[6] School of Computer Science and Engineering, VIT-AP University, Vijayawada 522237, India

Corresponding Author Email: ramesh.v@vitap.ac.in

https://doi.org/10.18280/ijsse.130513

Received: 3 June 2023
Revised: 7 October 2023
Accepted: 24 October 2023
Available online: 10 November 2023

Keywords:
message authentication code, hybrid encryption procedures, convolutional neural network, information security, secure communication

ABSTRACT

In today's highly competitive environment, it might be difficult to keep sensitive data or information safe. Sharing sensitive information between parties in a cloud environment requires a high level of trust between them. There are numerous methods for achieving data and information security, including cryptography, steganography, etc. This research introduces a new approach to mutual authentication by using a pre-trained model of a convolutional neural network (CNN) to identify malicious activity on the internet. In this research paper, we present a novel approach for enhancing the security of data in cloud computing environments through the design of a mutual authentication method for deep learning-based hybrid cryptography. Our approach combines the strengths of hybrid cryptography and the power of deep learning to provide a robust and adaptable solution for securing data in the cloud. One of the key innovations of our approach is the integration of a pre-trained convolutional neural network (CNN) model. This CNN plays a pivotal role in identifying and mitigating malicious activities on the internet that could pose a threat to cloud-based data. By continuously monitoring network traffic and data patterns, the CNN contributes to the proactive defense mechanism of our system. Secure communication between the involved parties is ensured by combining cryptography with authentication for key agreements. However, no known security method has simultaneously provided a high level of security and a fast execution time. When compared to older encryption systems, hybrid encryption techniques are far superior in terms of providing peace of mind for users. In order to provide robust security, this paper presents hybrid encryption procedures (HEA) by combination of symmetric key (Message Authentication Code [MAC]) and asymmetric key cryptographic procedures (Modified and Enhanced Lattice-Based Cryptography [MELBC]). Results from experiments show that the suggested HEA algorithm offers more security than competing security algorithms.

1. INTRODUCTION

Data and information security in the context of today's electronic communication system is a major obstacle. There are three objectives that must be met in order to keep the data safe. Data, information, and computer services need to have three things in place: privacy, security, and availability [1]. For this reason, information security aims to prevent data from falling into the wrong hands (through confidentiality), prevent data from being tampered with (through integrity), and make data accessible (through availability) to the appropriate parties [2]. Multimedia distribution is becoming a crucial technique for global service delivery because of the prevalence of the internet. The frontiers of commerce, science, and pleasure, as well as social possibilities, have been expanded. Strong copyright protection mechanisms [3] have been developed due to the ease with which digital content may be parallelized and transmitted. The internet is now widely used for the rapid dissemination of vast volumes of crucial information. As a result, it can be damaged by a wide range of threats. This data is therefore vulnerable to unauthorised access and other privacy and security breaches. When it comes to computer security, tried-and-true methods of data protection include cryptography and steganography [4]. Research has been done to enhance the aforementioned data security measures, but there are still certain caveats that show how important it is to find a solution to this problem.

Cyberattacks, as seen in Figure 1, can damage the IoT cloud networks [5] or disrupt the channel connection between sensors and the gateway. When compared to other cryptographic solutions, the conventional methods (such as RSA, AES, and DES) are prohibitively resource-intensive. As a result, they can't be used with low-power sensors or technologies like the Internet of Things right away [6]. It is so challenging to develop efficient, fast, compact, and secure cryptographic algorithms for the Internet of Things. In addition, IoT networks should have a basic cybersecurity system to prevent sensitive data from falling into the wrong hands and to verify that users have the proper authorization to make use of IoT services (through authentication and access control) [7-9].

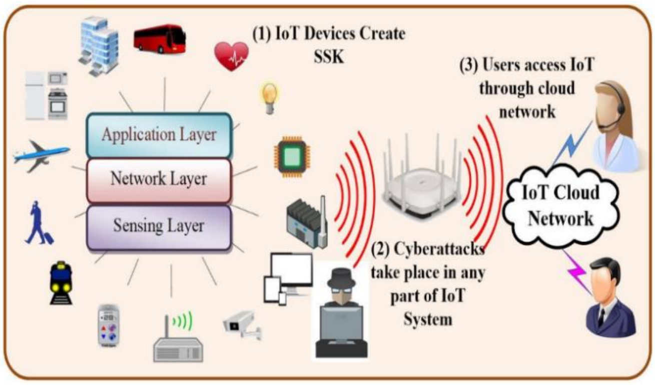
(1) IoT Devices Create SSK
(3) Users access IoT through cloud network
Application Layer
Network Layer
Sensing Layer
IoT Cloud Network
(2) Cyberattacks take place in any part of IoT System

Figure 1. Scenario of cyberattacks on the IoT network

Protecting the Internet of Things (IoT) from cyberattacks requires three main solutions: cryptography, digital signatures, and authentication. IoT Cryptography can make use of either symmetric (using a private key) or asymmetric (using a public key) encryption [10]. In symmetric encryption, both the sender and the receiver use the same key to perform the cryptographic operation. The security of symmetric encryption relies on how well the private key is disseminated among IoT nodes. In contrast to symmetric encryptions, which only require one key [11], asymmetric encryptions require two keys. The private key remains secret and is never divulged to any unauthorised devices, while the public key can be transmitted through a safe channel.

Device Authentication: In an IoT ecosystem, numerous devices communicate with each other and with central servers. Symmetric and asymmetric encryption can be used for device authentication. Symmetric keys can be used for initial authentication, ensuring that only trusted devices are allowed to join the network. Asymmetric encryption can be employed for more secure and ongoing authentication, enabling devices to verify each other's identities without sharing sensitive information.

Secure Data Transfer: IoT devices collect and transmit data, which often includes sensitive information. Symmetric encryption is well-suited for securing data in transit. IoT devices can encrypt data using a shared symmetric key before sending it to the cloud or another device. Only authorised recipients with access to the key can decrypt and read the data.

Public Key Infrastructure (PKI): Asymmetric encryption, specifically public-private key pairs, can be used to establish a public key infrastructure (PKI) in the IoT. IoT devices can have their own unique public keys, which are distributed widely, while the private keys are kept secure. This enables secure, end-to-end communication between devices without the need for shared secrets.

Secure Firmware Updates: IoT devices often require firmware updates for security enhancements and bug fixes. Asymmetric encryption can be used to sign firmware updates with a private key. Devices can verify the authenticity of updates using the corresponding public key, ensuring that updates come from trusted sources and have not been tampered with during transit.

Data Integrity: Both symmetric and asymmetric encryption methods can be used to ensure data integrity. Asymmetric signatures can confirm that data hasn't changed since the sender signed it, while symmetric encryption can prevent data tampering during transit.

Key Management: Managing encryption keys in an IoT environment is crucial. Discussing how symmetric and asymmetric keys are generated, distributed, and managed within IoT networks would provide insight into the practical challenges of IoT security.

In the same way that encryption may provide secrecy, message authentication can ensure that what was sent was indeed what was received. However, authentication and secrecy are essential for IoT systems. Although it's tempting to combine encryption and authentication into a single process, not all such schemes are secure [12]. Although it is possible to create a safe combination of cryptographic tools, doing so is notoriously challenging, and occasionally even excellent cryptographic tools are integrated in a way that creates an insecure combination [13]. To overcome these restrictions on online data exchange, our research integrates cryptography and deep learning methods at a trusted to provide a mutual authentication protocol with increased security and efficiency. For safe online communication, we suggest adequate registration, followed by correct session formation and password change phases, during which key agreement are implemented. The following is a brief summary of the main results of the planned effort.

- To facilitate the safe transfer of information between users in a federated cloud server situation, a new deep learning approach is presented.
- To protect against DDoS assaults and other security breaches, a reliable cloud server has created an online danger detector based on a pre-trained CNN model.
- The effectiveness of the suggested protocol was confirmed by formal verification utilising the ProVerif security analysis tool and a comparison to state-of-the-art solutions.

The remaining parts of the paper have the following layouts: In Section 2, we describe the relevant literature, and in Section 3, we offer a brief summary of the suggested model. Section 4 compares the suggested model's experimental analysis to the state-of-the-art methods. Finally, the study is summarised in Section 5.

2. RELATED WORKS

For the purpose of outlier identification in the IIoT, Sankaran et al. [14] suggested a technique for energy-efficient, protected data transfer. The primary goal of this method is to provide an industrial IoT data transfer technique that is both safe and private. Inputs include data from the industrial sector, such as electricity, loop sensors, and land sensors; a system network model is then developed and optimised using a multi-scale grasshopper algorithm. The information is then securely transported to the cloud, where it is kept. In order to determine the nature of a network assault, a classification method called Robust Multi-cascaded CNN (RMC-CNN) is used. Next, a dynamic honey pot encryption technique is used to encrypt the data utilising the key generation process. As a result, sensitive data may be encrypted before transmission and kept in the cloud until it is needed. The Throughput, latency, and detection rate are also compared and contrasted. Finally, estimates for encryption, decryption, and running time are provided and compared to current methods. According to the results, the suggested system outperforms the current approaches by a significant margin.

Awasthi et al. [15] suggest a new approach to e-health data analysis using cloud-based device-to-device communication through feature selection and categorization. The goal of this study is to examine the potential of integrating cloud and distributed computing into e-healthcare by conducting a thorough requirement investigation and user survey. After contrasting the smart healthcare scheme with database-centric healthcare approaches, a prototype system will be developed and implemented. The acquired e-health data is analysed using convolutional adversarial neural networks trained using transfer perceptron on the cloud. The proposed method achieved the following results: 98% accuracy in training, 93% accuracy in validation, 66% PSNR, 68% MSE, 72% precision, 63% quality of service, 58% latency.

To fortify the safety of monitoring systems, Diao et al. [16] provide an approved joint security strategy (EJSS). Based on those steps, this system acknowledges the use of electronic health records. Different security measures are used for each procedure to ensure the longest possible protection. Relative security, implemented by mutual key sharing between the accessing user and the EHR database, is required for entree control and storage change. Learning in this scenario reveals the ways in which various procedures might mitigate hostile interference. This technique employs a federated learning model to locate adversaries across several simultaneous operations. To improve mutual authentication using individual qualities, it is helpful to classify adversaries differently for each procedure. Therefore, for diverse and extensive EHR datasets, the efficiency of individual monitoring is enhanced by log inspection and adversary identification. In the context of privacy manipulation in group training, Madni et al. [17] explore scenarios involving compromised and malevolent participants in the Secure Learning (SL) environment. To safeguard the confidentiality of model parameters prior to sharing them among participants, who have been validated and registered using blockchain technology, we propose a technique known as Swarm-FHE, which stands for Swarm Learning with Fully Homomorphic Encryption. In Swarm Learning training, all participants exchange ciphertexts, which are encrypted versions of their parameters, with each other. We evaluate the effectiveness of our approach by training convolutional neural networks on the CIFAR-10 and MNIST datasets. Through extensive experimentation using a variety of hyperparameter settings, we demonstrate that our technique surpasses existing state-of-the-art alternatives.

A lightweight authentication approach, dubbed LAFED, is proposed by Ji et al. [18] for blockchain-enabled federated learning systems. LAFED introduces three new ideas to the field: an adaptive model aggregation algorithm based on the quality of the models and the contributions of the nodes to boost performance. Extensive experiments show that the proposed LAFED may accomplish lightweight authentication while guaranteeing a high model accuracy.

A blockchain and garlic-routing based safe data exchange system, i.e., GRADE, was suggested by Jadav et al. [19], which removes the security restrictions and keeps the connection steady in MTC. When a data request from an MTC user is determined to be safe, the Long-Short-Term Memory (LSTM)-based Nadam optimizer sends it to the Garlic Routing (GR) network. Each computer taking part in MTC is given a one-of-a-kind ElGamal encrypted session tag by the GR network. Then, the requests for MTC data are encrypted using the Advanced Encryption Standard (AES). The proposed GRADE system is also more scalable since the machine's session tags are stored in a blockchain based on the Inter-Planetary File System (IPFS). MTC network performance has also been improved according to the proposed framework's incorporation of the necessary advantages offered by the 6G network. Finally, the proposed GRADE framework is tested using a variety of performance measures, including scalability, packet loss, correctness, and the rate at which MTC data requests are compromised. In comparison to the baseline approaches, the results reveal that the GRADE framework achieves higher levels of accuracy (98.9%), a lower compromised rate (18.5%), more scalability (47.2%), and a lower packet loss ratio (24.3%).

According to research by Anusuya et al. [20], healthcare providers, lab workers, data scientists, and proprietors of machine learning models used for illness analysis all need to treat patients' personal health information with the utmost confidentiality. Medical records and imaging studies must be encrypted before disease analysis may begin. Chest x-rays of children have been taken for pneumonia diagnosis. Before the X-ray pictures can be categorised, they are encrypted. In this study, we focus on three different encryption algorithms and compare them based on their runtime performance. After data is encrypted, a Convolutional Neural Network (CNN) deep learning model is constructed for picture categorization. The VGG16 transfer learning model is used to classify the same dataset experimentally, revealing improved performance in shown that running the same model in a federated distributed learning context improves accuracy even more.

3. PROPOSED SYSTEM

3.1 Phase 1: Modified and improved lattice-based cryptography

Partitioning plaintext, creating encryption keys, encrypting data, re-encrypting data, and decrypting data are all components of the improved lattice-based cryptography.

3.1.1 Plaintext partitioning procedure

The following are the procedures of the projected Plaintext Partitioning Procedure (PPA):

Plaintext Partitioning Procedure

Input: Entire plain text

Output: Partitioning plaintext into two parts PT1, PT2

Step 1: In this phase the available plaintext is divided into two partitions. If “l” is the length of the message,then the first one-two part of the message goes to the encryption, Re-encryption, decryption processes and it ranges from 0 to l/2–1

$$p_{-1} = \frac{l}{2} - 1 \tag{1}$$

Step 2: “p1” is a first-half of plaint text partition. Then ciphertext of the first half of the plaintext partition is produced by the following Eq. (2)

$$c_1 = (MELB, C_{-enc}(MELB, C_{-en}(p_{-1}, k_{-1}))k_{Re}) \tag{2}$$

Step 3: “MELBC” is modified enhanced lattice-based cryptography, “enc” encryption, and 1 is plaintext first half partition, k1 is a key, k_Re is a re-generation key. Next second half of the plaintext message is allotted to the second part, and its range is from $\frac{n}{2}$ to $\frac{2n}{2}-1$, then the second part of the plaintext goes to encryption and decryption process only. The cipher text c2 is framed with the help of Eq. (4).

$$p_2 = \frac{I}{2} - 1 \tag{3}$$

$$c_2 = \left(MELBC_{-enc}(p_{-2\ldots,}k_{-1})\right) \tag{4}$$

The use of plain text partitioning is crucial to the proposed approach. Part 1 of the plaintext (PT1) and Part 2 (PT2) make up the entirety of the plaintext. The first using the improved, then the cypher text format of PT1 is encrypted once again using the same technique and key-value, k_{Re}.

3.1.2 MELBC Encryption process ($MELBC_{\text{enc}}$)

The Multi-Element Lattice-Based Cryptosystem (MELBC) is a public-key cryptosystem. Providing efficiency is what drives MELBC, and compared to other public key cryptosystems, it offers a superior method to ward off quantum assaults. The quotient ring is used to communicate with MELBC. The projected MELBC-Key Generation Procedure (MELBC-KGA) may be broken down into the following phases:

MELBC Key Generation Procedure

Input: Systems and security parameters.
Output: Public key and secret key
Step 1: The list of input parameters are $\{(I^m), (a, b, p, q), (E_n, D_e, M, S_v, R_v, F)\}$.
The symbol m is the security parameter, a is a row wise dimension parameter, b is the column wise dimension, p is a prime modulus, q is the identity space dimension, E_n represents an encoding function to map public identities, De is the bitwise decomposition of Ids, M is the uniformly random matrix $M \in Z_p^{a\times b}, S_v$ is the secret vector $S_v \in Z_p^a, R_v$ is the random vector $R_v \in Z_p^a, F$ is the function $F: Z^a \to Z^b$. Based on the public parameters, the public keys (E_n, R_v, M) are generated as follows

$$M \in Z_p^{a\times b}, F: Z_p^a \to Z_p^b, R_v \in Z_p^a, E_n \in Z_p^{a\times 1} = PU \tag{5}$$

Step 2: private key is generated based on sample vector (s_a) and error vector (e_r) as shadows

$$s_a \in Z_p^q \tag{6}$$

$$e_r \in Z_p^a \tag{7}$$

$$PR = s_a + e_r \tag{8}$$

Step 3: Public Key “PU'” and private key “PR” is generated

System and security characteristics such as row and column dimensions, vector, and error vector are used to derive the corresponding public and private keys. Due to their large computational and communication costs, the conventional public key cryptography techniques are inadequate. Public key cryptosystems vulnerable to quantum computers are those based on the integer factorization problem or the discrete logarithm problematic. Secure constructions against attacks from classical and quantum computers are only possible using Adapted Enhanced Lattice-Based Cryptography (MELBC).

3.1.3 Encryption phase
Plaintext divider (p1), ID, public key (PU’) → Cipher text (c1)
Cipher text (Cip1) = Encryption (c1)
Plaintext (p1) → Cipher text (Cip1)
Plaintext (p2) → Cipher text (c2)
Cipher text (C) → combine (cip1, c2)
Step1 Sender’s public key

$$M \in Z_p^{a\times b}, F: Z_p^a \to Z_p^b, R_v \in Z_p^a, E_n \in Z_p^{a\times 1} = PU' \tag{9}$$

Phase 2 Plaintext partition $p_{-1} = l/2 - 1$ (1)
Phase 3 Encryption

$$c_1 = p_1 . PU' + e_r \tag{10}$$

Phase 4 Re-key generation

$$R_k = (PU' + PR)\ XOR\ PU' \tag{11}$$

Phase 5 Re-Encryption

$$Cip1 = Re_{-Enc}(R_k(C_1)) \tag{12}$$

Phase 6 Plaintext partition $p_2 = l/2 - 1$ (3)
Phase 7 Encryption

$$c_2 = p_2 . PU' + er \tag{13}$$

Phase 8 Cipher Text C Combine (Cip1, c2)

Using the public key PU′ and the error vector er, we encrypt the plaintext partition p_1 to create the ciphertext c_1. Input is the ciphertext c_1 and the regeneration key R_k, which is computed using Eq. (11). Ciphertext Cip1 is the result of the re-encryption procedure and may be written using the formula (12). In this case, the ciphertext c_2 is formed by the encryption using Eq. (13), with the inputs of public key. Send the ciphertext C to the recipient by combining Cip1 and c_2 and then framing it.

Plaintext Partition:

Define what a "plaintext partition" is within the context of your research. Is it a data segmentation technique, and if so, how is it applied?

Explain the purpose of plaintext partitioning in your system and its significance in achieving your research goals.

Re-Encryption Key:

Define the term "re-encryption key" and its role in your proposed system. Is it used for secure data transmission or other cryptographic operations?

Clarify how re-encryption keys are generated or distributed in your system.

Other Key Concepts:

If there are additional terms or concepts unique to your research, provide clear and concise definitions for each of them.

Illustrative Examples:

Whenever possible, consider including illustrative examples or diagrams to help readers visualize how these concepts are applied in your system.

3.1.4 Decryption phase

It accepts the cypher text C, the identifier ID, the private key PR, the public key PU′, and the Re-generation key R_k as inputs during decryption, and returns the plaintext message p as an output.

Cipher text (C), ID, secluded key (PR), (PU′), Re-generation key R_k → plain text (p)

Step 1 Decryption process cipher text c2

$$p2 = decrpt_{PR}(c_2.PU\,' + e_r) \tag{14}$$

$$p2 = decrpt_{PR}(c_2.PU\,' + er)PU'^{-1} \tag{15}$$

Step 3 Where, U is a unimodular matrix

$$p_2 = p_2.PU\,'.PU'^{-1} + er.PU'^{-1} \tag{16}$$

Step 4 The Babai rounding procedure will be used to eliminate the term er. PU′ − 1

$$p_2 = p_2 + er.PU'^{-1} \tag{17}$$

Step 5 Total the plaintext partition p_1 p_2

Step 6 Decryption procedure cipher text Cip1

$$p_1 = decrpt(cip1)p_1 = decrpt(Re_{-decrpt}R_k(c_1)) \tag{18}$$

Step 7 Decryption process cipher text c1

$$p_1 = decrptPR(c_1.PU\,' + er) \tag{19}$$

Step 8

$$p_1 = decrpt_{PR}(c_1.PU' + e_r)PU'^{-1} \tag{20}$$

Step 9

$$p_1 = p_1.PU'.PU' - 1 + er.PU'^{-1} \tag{21}$$

Step 10 The Babai rounding procedure to remove the term

$$er.PU'^{-1}\, p_1 = p_1 + er.PU'^{-1} \tag{22}$$

Step 11 Compute the plaintext partition p1 1

Step 12 Finally combine the Plaintext (p) → syndicate (p_1, p_2)

Eqns. (18) and (22) represent the results of the decryption procedure. The plaintext p can be received by the receiver without a quantum attack if we merge the plaintext partitions p1 and p2.

3.2 Phase 2: Mutual authentication protocol

A Trusted Authentication (TA) scheme implemented at middleware linked to many cloud servers is used by a company that employs the services of numerous cloud servers. This middleware interface is hosted on a reliable, centralised server. The TA system combines encryption and deep learning to provide the safest possible session establishment between parties involved in the conversation. The mutual key agreement phase is demonstrated using a hybrid cryptography approach, and a deep learning based pre-trained model of CNN is used to identify and mitigate any threat or abnormality during the session setup phase. In the parts that follow, we'll go through each individual part in greater depth.

3.2.1 Deep Learning classifier-based anomaly detector

The TA server incorporates a deep learning classifier, in the form of a Capsule Network model (discussed in more detail in Section 3.2.3.) that learns to predict a target output (class) with high precision. P_final is the projected output and there are n characteristics, such as Cl_1, Cl_2, ..., Cl_n, that go into making this prediction. Classifiers are continually trained using both archived data and real-time information from the session setup phase (or key agreement). If an abnormality is found in the input, the anticipated output will read "malware," and otherwise, "benign". The TA server uses the key parameters and all the characteristics (received from the sender) to determine the likelihood of malicious intent behind session set-up when it gets a request from an entity (user/data owner). Only if the danger likelihood is 0% is the session setup allowed.

One-time passwords, security questions, access restrictions, and even full access to the material may be implemented in response to a detected anomaly.

In a federated cloud server setup, mutual key agreement demonstrates the recommended paradigm for mutual authentication and safe common session key (CSK) calculation. Here, we have two entities—the data owner (Ea) and the data receiver (Eb)—and a third, TA, which is a Trusted Authentication System implemented on middleware using cloud server federation. All of the company's personnel information is safely saved in TA's database.

The full procedure for authentication and session key agreements looks like this:

1. First, entity A (Ea) submits a login request to entity B (TA), including a secret identity parameter; TA checks this parameter against its record in its database of legitimate entities; TA then provides OTP back to Ea after successful authentication.
2. After inputting the OTP start parameters (i.e., the secret identities of the sender and recipient), Ea will issue a session establishment request.
3. Third, the criteria for establishing the session, such as the hidden identities of Ea and Eb and the nonce, are sent to TA. If an anomaly or danger is detected, TA will activate the voting classifier (described in the next section). If no danger is found, the session is set up by applying a Schnorr's signature and calculating masked IDs for both the sender and the recipient.
4. The fourth receiver, Eb, logs on to TA in the same way. Masked identifiers and other security settings are sent to Eb from TA.
5. Then, in a secure environment, Eb does certain computations to confirm that TA is a legitimate server and to generate common shared session keys (CSK). In response to TA, Eb generates a new nonce value but does not send the CSK.
6. Six, after some computational effort, TA checks Eb and sends Ea, Eb's answer and other security settings.
7. After some computational effort, Ea checks TA and Eb and accepts the reply in order to compute its mutual shared sitting key CSK.

Successfully completing the authentication process and setting up a secure session among Ea and Eb through TA and computing CSK without disclosing it on the public channel is achieved in this manner.

3.2.2 Phases of mutual authentication

The projected authentication arrangement works in three stages:
1. Registration stage
2. Login phase
3. Password renewal stage

Registration phase. To use the trusted authentication system TA, the entity selects a password (Pwd) and provides his unique employee (ID) details, including his official registered mail-ID, usb fingerprint as a usb-key, over a secure channel. Here, TA determines a secret numeral for each subscriber using the following formula:

$$C = H(ID)\|Pwd\|PhoneNumber\|p\|UsbLogin\|Email\|$$

where, p is TA selects large prime sum p, and elliptic curve E over finite field F_{p} as $a, b: y^2 = x^2 + ax + b$where $a, b \in Fp$ and $4_1^3 + 27b^2(modp) \neq 0, E(Fp)$ is set of all points on curve E. After that, TA sends Ea the public key y and the entity's secret number c, which it derives from the private key p using the formula y=H(cp). For further security, TA keeps a record of each user's p,c, and UsbLogin key in a central repository.

Login and authentication phase. 1. E_a initiates a login with a password of c'. The TA gets it, checks to see whether c'=c in the repository, and then drops the session if the two don't match. If not, it will send a temporary OTP (good for only 5 minutes) to E_a.

2. E_a User provides one-time password and USB key; TA verifies successful mutual authentication. Encrypts the sender's ID (Ea) and the recipient's ID (Eb) using the following key exchange:

3. $aa = Hash1(ID_a, t1, y)$

4. $ab = Hash1(ID_b, t1, y)$

5. E_a then directs aa, ab and nonce t1 (freshly shaped time stamp that can never be recurrent) to TA.

6. TA then computes masked independences for E_a and E_b to setup current meeting, by smearing Schnorr's signature as shadows:

7. $mid\_a = Mult(aa, Hash2(y, xt1)$

8. $mid\_b = Mult(ab, Hash2(y, t2)$

9. TA sends mid_a, mid_$b, t1, t2, aa, ab\ to\ Eb$. Then, E_b applies adapted Schnorr's signature verifying scheme to figure x1 and x2 as shadows:

10. $x1 = Mult(ab, Hash(y, t2))$

11. $x2 = Mult(aa, Hash(y, t1))$

12. If (x1=mid_b and x2=mid_a) then dismiss the session, then, TA is confirmed as trusted server and E_b generates new common shared session key CSK.

13. $r = Hash(Hash(t1, t2), t3)$

14. $csk = Hash(r, mid\_a, mid\_b)$

15. E_b replies TA with security parameters rand t3 TA receives r and new nonce t3, TA calculates cc to verify E_b as follows:

16. $cc = Hash(Hash(t1, t2), t3)$

17. Using its before known nonce value (i.e., t1 and t2) and newly conventional nonce t3. If cc= r then E_b by TA) to E_b.

18. E a receives masked ID from TA and confirms TA and Eb as shadows:

19. $xmid\ a = Mult(aa, Hash(y, t1))$

20. $xmid\ b = Mult(ab, Hash(y, t2))$

21. If xmid a≠mid a and xmid b ≠mid b then terminates the session, then, compute common key (CSK) as follows:

22. $csk = Hash(cc, mid\ a, mid\ b)$

Password renewal phase. Every customer is required to change his password at regular intervals (every week, every two weeks, every month, etc.). However, the user has the option to change their password at any moment. Since the secret number c, in addition to ID, mailID, etc., is dependent on the password Pwd, if the Pwd changes, c, mechanically vicissitudes; hence, the secret numeral is refreshed weekly or at the user's discretion.

Anomaly detection model. Incorporating CAL and the Adagrad optimizer, the CapsNet model is put to use as a classifier. In the suggested study, entropy is utilised to rank uncertain data and expose the signal's unpredictability by visualising the chaos in the system.

CapsNet construction. The CapsNet perfect is designed to model the hierarchical relationships and preserve the picture objects' locations and characteristics. In the CNN approach, the pooling layer receives the most relevant information first. It's possible that the network won't be able to learn fine-grained information when the data is passed on to the next pooling layer. In addition, the CNN approach produces a numeric number for neural output. The CapsNets' multi-neuron capsules create similarly sized vector output with different routing. The CNN uses activation functions on the input vector such as the Tangent, ReLU, and Sigmoid. The CapsNet, on the other hand, makes use of a vector activation function called squashing, which is described by Eq. (23).

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|} \tag{23}$$

where, v_j is the final output of capsule j. When an item is present in the image, the length of the vector v_j is reduced to one, and when none is present, the vector is truncated to zero. In CapsNet layers beyond the first, the predictive vector's weights are used to estimate capsule S_j's input values.

$(U_{\mathrm{j|i}})$ in the capsule as layer. The prognostic vector $(U_{\mathrm{j|i}})$ is projected by the produce of a capsule layer with their output (O_{i}) and weight matrix (W_{ij}).

$$S_j = \sum_i b_{ij} u_{j|i} \tag{24}$$

$$u_{j|i} = W_{ij} O_i \tag{25}$$

where, b_{ij} signifies constant distinct by lively routing process and it is assumed by,

$$b_{ij} = \frac{\exp\,(a_{ij})}{\sum_k exp(a_{ik})} \tag{26}$$

where, a_{ij} represents a logarithmic probability. Log prior likelihood is specified by Softmax, and the number of correlation coefficients between capsules i in the top layer is one. Margin loss for estimating the number of objects in a given class is as follows in CapsNet.

$$L_k = T_k max(O, m^+ - \|v_k\|^2 + \lambda(1 - T_k max(O, \|v_k\| - m)))^2 \tag{27}$$

When element k of class k is present, T_k has a value of 1. In addition, the hyper variables are represented by m+ = 0.9 and the weight of the loss by m- = 0.1. CapsNet, which represents the probability in the picture region, provides an estimate of the vector length.

Class attention layer apparatus. It is unable to learn the probabilistic dependence by recurrent eating with comparable features, even however the extracted features are at a higher level and are fed directly into FC layer to prediction. Therefore, defining class dependence and properly linking CapsNet for multi-label classification processes relies heavily on the extraction of discriminative class wise features. Feature discovery across all domains is thus conducted using CAL. First, a class attention map is generated using an 11 convolution layer with a stride of 1, and then, in a second step, each class attention map is vectorized to extract characteristics unique to that class. In model, the size of a feature map X is defined as W W K, where 1v_l is the l-th filter in the CAL layer. Class l's care mapping Ml may be calculated using the formula:

$$M_l = X * w_l \tag{28}$$

where, l is an arbitrary number between one and several classes and is a convolution function. To illustrate, a class care mapping M_l is just a linear integration of the entire channel in X, given a convolution filter of size 1 1. The provided CAL may be implemented in this way to develop separate mapping [21]. In Eq. (28), for instance, feature mapping X is considered to be the output of the model's convolutional block after input data is processed. Therefore, vectorization is used to transform the class mapping M_l into a class wise feature vector vl with W2 dimensions.

Adagrad optimizer The CapsNet model's hyperparameters are fine-tuned using the Adam optimizer. The Adagrad optimizer [22] is a gradient-oriented optimisation method that performs well on sparse gradients. Hyperparameter optimisation, also known as tuning, is the procedure of determining the optimal values for a learning algorithm's hyperparameters. A hyperparameter is a variable whose value is adjusted to affect training. Concurrently, we learn about other factors like the weight of nodes. The rate at which it learns would be automatically modified. Eq. (29) is the primary formula used in parameter updates.

$$\theta_{t+1} = \theta_t - \frac{a}{\sqrt{\varepsilon + \sum g_t^2}} \odot g_t \tag{29}$$

where, θ_t signifies the mutable at time t, α designates the learning rate, g_t means gradient estimation and $\odot$ signifies multiplication.

4. RESULTS AND DISCUSSION

A server computer built around two IntelVR XeonVR Silver 4114 CPUs, each with 40 cores and 2.20 GHz clock speed, runs the simulation tests. This computing system has 128 GB of main RAM and runs Ubuntu 16.04 LTS on a 64-bit architecture. Python 3.7 is used for the suggested work. The conditions for assessing the presentation of the projected plan are exposed in Table 1.

4.1 Modularity check

Table 2 displays the suggested model's module-based processor utilisation. Additionally, the following is the module-based execution period of the proposed model. The elapsed time for the hybrid encryption scheme is estimated. The amount of time needed by a cryptographic method to change plaintext into an encrypted message. The throughput of an algorithm may be estimated by looking at how long it takes to encrypt data. It is the determining factor in encryption speed. If throughput could be increased, electricity consumption might be reduced. As will be shown below, the results varied depending on the magnitude of the inputs used.

Table 1. Setup for experiments

Setup	Explanation
Stage	Visual C++ (Visual studio Community 2017)
Scheme	64-bit OS, X-64 based Processor
OS	Windows-10
Processor	Intel (R) Core (TM)i7-7500U CPU @2.60 Ghz 2.90GHz

Table 2. Processor utilization rate

	1st Trial	2nd Trial	3rd Trial	4th Trial
Proposed hybrid model	405	146	113	357
AES	464	859	996	986
Blowfish	488	980	988	1010
DES	523	240	443	340
RC5	530	650	708	980
3DES	416	190	371	301

Each round's CPU time was studied by altering the amount of the input data. In the suggested approach, the key transformation module often consumes the most processing time. The obtained results demonstrate the usefulness of the suggested paradigm. In this part, we examine the strengths and weaknesses of several of the most popular block cyphers in the context of HI security in the cloud. Understanding how much time a central processing unit (CPU) spends on a given task is called "processor usage". It indicates how much work is being done by the CPU. The more the use of central processing unit in the encipherment method, the greater the burden on the computer's CPU. The purpose of these tests is to examine the performance and impact of various input sizes and the influence of various stages on the estimation of processor time.

Furthermore, it is clear from the aforementioned experiments that the proposed model outperforms other widely-used procedures in processor utilisation rate, less memory utilisation, the highest degree of key rate, and that its utilisation makes it a more attractive excellent for mobile devices. Table 3 demonstrates additional, separate qualitative security guaranteeing techniques that allow the planned strategy to produce satisfactory experimental findings.

Table 3. Memory utilization

Encryption Scheme	MSE	AES	Blowfish	DES	RC5	3DES
Memory utilization (KB)	984	1357	1293	1785	1624	1896

Key varieties or sorts of these many systems (based on user-specified needs for a certain level of security) A qualitative comparative method was used here. Single-type-key support is provided by DES and IDEA; three-type-key support is provided by AES; and five-type-key support is provided by the suggested model, indicating a very requirement-centric approach. It follows that the suggested model has the greatest number of significant deviations. Since the keys include measures, in most cases each key alters the data twice every round, for a total of eighteen times the key includes information rather than nine times. Table 4 details a comparison of RC5, RC6, Blowfish, 3DES, and the proposed model from the perspective of single-round key subsumption; the proposed model outperforms IDEA, DES, 3DES, and AES by transforming data twice in each round.

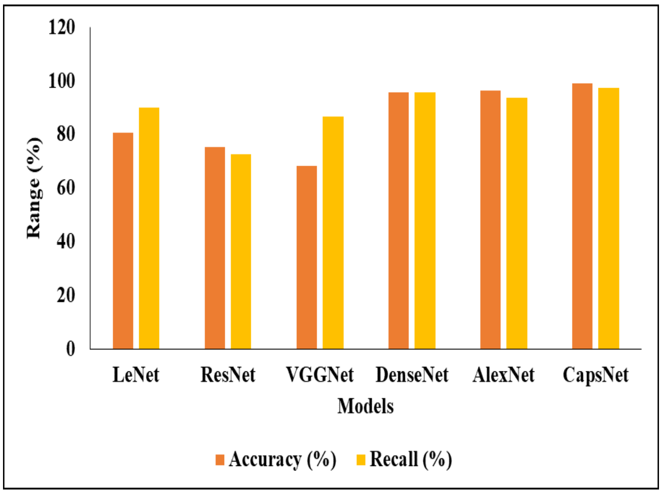

Figure 2. Analysis of proposed classifier

Table 5 represents the comparative analysis of a pre-trained classifier with existing models. In this analysis, the LeNet model reached an AUC score of 0.758, an accuracy level of 80.72, a precision rate of 78.14, a recall value of 89.92, and finally, an F-measure value of 88.67, respectively. And another method of ResNet model reached the AUC score of 0.854 and also the accuracy level of 75.17, the precision rate of 70.91, the recall value of 72.69, and finally the F1-score value of 72.33, respectively. And another method of VGGNet model reached an AUC score of 0.687, an accuracy level of 68.28, a precision rate of 64.17, a recall value of 86.66, and finally, an F-measure value of 80.24, respectively. And another method of DenseNet model reached an AUC score of 0.947, an accuracy level of 95.78, a precision rate of 91.94, a recall value of 95.61, and finally, an F-measure value of 76.86, respectively. And another method of AlexNet model reached the AUC score of 0.957 and also the accuracy level of 96.34, the precision rate of 92.45, the recall value of 93.78, and finally, the F-measure value of 94.36, respectively. And another method of CapsNet model reached an AUC score of 0.987, an accuracy level of 98.97, a precision rate of 96.84, a recall value of 97.24, and finally, a F-measure value of 94.13, respectively. In this comparison analysis, the CapsNet model achieved better performance than the other compared models, as shown in Figures 2-4.

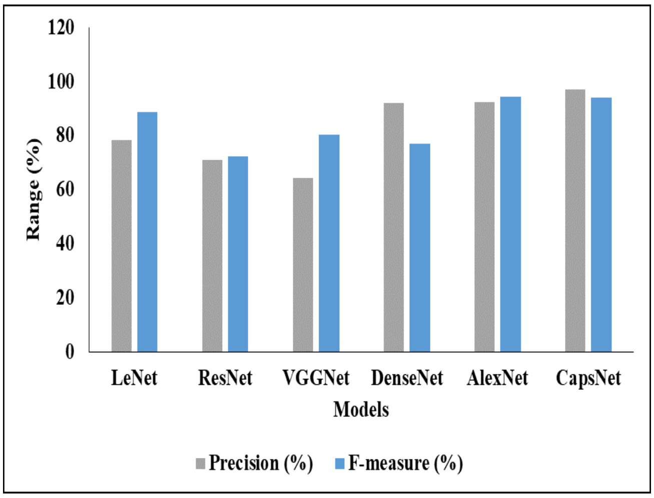

Figure 3. Graphical comparison between different DL models

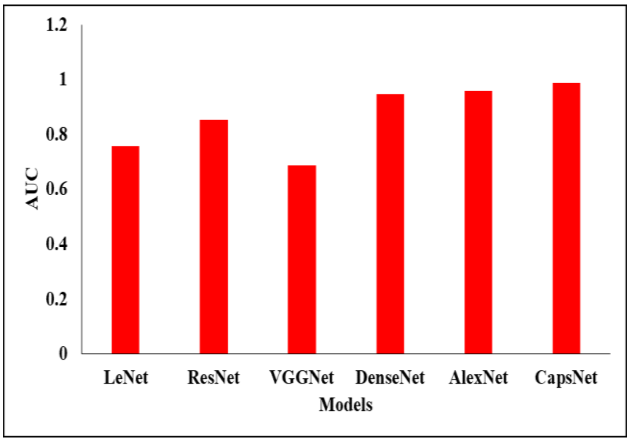

Figure 4. AUC validation

Table 4. Key-data colligation rate

Encryption scheme	Proposed Hybrid Model	AES	Blowfish	DES	RC5	3DES
Single round key colligation	2.11	1.04	1.02	1.03	1.02	1.04

Table 5. Comparative analysis of pre-trained classifier with existing models

Classification	AUC	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
LeNet	0.758	80.72	78.14	89.92	88.67
ResNet	0.854	75.17	70.91	72.69	72.33
VGGNet	0.687	68.28	64.17	86.66	80.24
DenseNet	0.947	95.78	91.94	95.61	76.86
AlexNet	0.957	96.34	92.45	93.78	94.36
CapsNet	**0.987**	**98.97**	**96.84**	**97.24**	**94.13**

5. CONCLUSION

For safe information exchange, we present a new mutual authentication system that utilises both encryption and deep learning. Applications benefit from robust security thanks to the deployment of a hybrid encryption model that encrypts communications with a combination of Modified and cryptography and a MAC procedure. Step one: amp up the force Methods of re-encryption are offered for use in MELBC in order to fortify the security of the ciphering procedure. Second, we offer plaintext segmentation approaches to reduce computational load, power consumption, and compromise while maintaining a high level of security. As a conclusion to the work provided, fresh formulae are developed for the processes of key creation, encryption, and decryption. This system allowed for highly authenticated data transport. Due to the fact that users' true identities directly shared on the network, this protocol is secure against a wide variety of security threats. Additionally, the mutual authentication scheme's security has been examined with the help of the ProVerif security analysis programme. The outcomes prove that the suggested protocol is computationally inexpensive and secure against several online data sharing security threats in a multi-cloud setting. More sophisticated mathematical and computational advancements in the future will require the incorporation of encryption algorithms with deep learning into our scheme.

REFERENCES

[1] Narayanan, U., Paul, V., Joseph, S. (2021). Decentralized blockchain based authentication for secure data sharing in Cloud-IoT: DeBlock-Sec. Journal of Ambient Intelligence and Humanized Computing, 13: 769-787. https://doi.org/10.1007/s12652-021-02929-z

[2] Narayanan, U., Paul, V., Joseph, S. (2022). A novel system architecture for secure authentication and data sharing in cloud enabled Big Data Environment. Journal of King Saud University-Computer and Information Sciences, 34(6): 3121-3135. https://doi.org/10.1016/j.jksuci.2020.05.005

[3] Rajasoundaran, S., Prabu, A.V., Routray, S., Kumar, S.S., Malla, P.P., Maloji, S., Mukherjee, A., Ghosh, U. (2021). Machine learning based deep job exploration and secure transactions in virtual private cloud systems. Computers & Security, 109: 102379. https://doi.org/10.1016/j.cose.2021.102379

[4] Ali, A., Pasha, M.F., Ali, J., Fang, O.H., Masud, M., Jurcut, A.D., Alzain, M.A. (2022). Deep learning based homomorphic secure search-able encryption for keyword search in blockchain healthcare system: A novel approach to cryptography. Sensors, 22(2): 528. https://doi.org/10.3390/s22020528

[5] Kotapati, G., Ali, M.A., Vatambeti, R. (2023). Deep learning-enhanced hybrid fruit fly optimization for intelligent traffic control in smart urban communities. Mechatron. Intell Transp. Syst, 2(2): 89-101. https://doi.org/10.56578/mits020204

[6] Prabhakaran, V., Kulandasamy, A. (2021). Hybrid semantic deep learning architecture and optimal advanced encryption standard key management scheme for secure cloud storage and intrusion detection. Neural Computing and Applications, 33(21): 14459-14479. https://doi.org/10.1007/s00521-021-06085-5

[7] Zhou, Z., Gaurav, A., Gupta, B.B., Lytras, M.D., Razzak, I. (2021). A fine-grained access control and security approach for intelligent vehicular transport in 6g communication system. IEEE Transactions on Intelligent Transportation Systems, 23(7): 9726-9735. https://doi.org/10.1109/TITS.2021.3106825

[8] Almuzaini, K.K., Sinhal, A.K., Ranjan, R., Goel, V., Shrivastava, R., Halifa, A. (2022). Key aggregation cryptosystem and double encryption method for cloud-based intelligent machine learning techniques-based health monitoring systems. Computational Intelligence and Neuroscience, 2022: 3767912. https://doi.org/10.1155/2022/3767912

[9] Vatambeti, R., Divya, N.S., Jalla, H.R., Gopalachari, M.V. (2022). Attack detection using a lightweight blockchain based elliptic curve digital signature algorithm in cyber systems. International Journal of Safety & Security Engineering, 12(6): 745-753. https://doi.org/10.18280/ijsse.120611

[10] Salvakkam, D.B., Pamula, R. (2022). MESSB–LWE: multi-extractable somewhere statistically binding and learning with error-based integrity and authentication for cloud storage. The Journal of Supercomputing, 78(14): 16364-16393. https://doi.org/10.1007/s11227-022-04497-1

[11] Singh, C.E.J., Sunitha, C.A. (2022). Chaotic and Paillier secure image data sharing based on blockchain and cloud security. Expert Systems with Applications, 198: 116874. https://doi.org/10.1016/j.eswa.2022.116874

[12] Tahir, M., Sardaraz, M., Mehmood, Z., Muhammad, S. (2021). CryptoGA: A cryptosystem based on genetic algorithm for cloud data security. Cluster Computing, 24: 739-752. https://doi.org/10.1007/s10586-020-03157-4

[13] Singh, A.K., Saxena, D. (2022). A cryptography and machine learning based authentication for secure data-sharing in federated cloud services environment. Journal of Applied Security Research, 17(3): 385-412. https://doi.org/10.1080/19361610.2020.1870404

[14] Sankaran, K.S., Kim, B.H. (2023). Deep learning based energy efficient optimal RMC-CNN model for secured data transmission and anomaly detection in industrial IOT. Sustainable Energy Technologies and Assessments, 56: 102983. https://doi.org/10.1016/j.seta.2022.102983

[15] Awasthi, A., Suchithra, R., Chakravarty, A., Shah, J., Ghosh, D., Kumar, A. (2023). Machine learning-based D2D communication for a cloud-secure e-health system and data analysis by feature selection with classification. Preprint. https://doi.org/10.21203/rs.3.rs-2653343/v1

[16] Diao, Z., Sun, F. (2023). A deep-learning neural network approach for secure wireless communication in the surveillance of electronic health records. Processes, 11(5): 1329. https://doi.org/10.3390/pr11051329

[17] Madni, H.A., Umer, R.M., Foresti, G.L. (2023). Swarm-FHE: Fully homomorphic encryption-based swarm learning for malicious clients. International Journal of Neural Systems, 33(8): 2350033. https://doi.org/10.1142/S0129065723500338

[18] Ji, S., Zhang, J., Zhang, Y., Han, Z., Ma, C. (2023). LAFED: A lightweight authentication mechanism for blockchain-enabled federated learning system. Future Generation Computer Systems, 145: 56-67. https://doi.org/10.1016/j.future.2023.03.014

[19] Jadav, N.K., Kakkar, R., Mankodiya, H., Gupta, R., Tanwar, S., Agrawal, S., Sharma, R. (2023). GRADE:

Deep learning and garlic routing-based secure data sharing framework for IIoT beyond 5G. Digital Communications and Networks, 9(2): 422-435. https://doi.org/10.1016/j.dcan.2022.11.004

[20] Anusuya, R., Oviya, S., Sangavi, R. (2023). Secured data sharing of medical images for disease diagnosis using deep learning models and federated learning framework. In 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, pp. 499-504. https://doi.org/10.1109/ICISCoIS56541.2023.10100542

[21] Hua, Y., Mou, L., Zhu, X.X. (2019). Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. ISPRS Journal of Photogrammetry and Remote Sensing, 149: 188-199. https://doi.org/10.1016/j.isprsjprs.2019.01.015

[22] Saravanan, S., Hailu, M., Gouse, G.M., Lavanya, M., Vijaysai, R. (2019). Optimized secure scan flip flop to thwart side channel attack in crypto-chip. In: Zimale, F., Enku Nigussie, T., Fanta, S. (eds) Advances of Science and Technology. ICAST 2018. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 274. Springer, Cham. https://doi.org/10.1007/978-3-030-15357-1_34