Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549563

INTELLIGENT DATA SECURITY FOR CLOUD COMPUTING USING PARTIAL HOMOMORPHIC ENCRYPTION

Sindhu V
Research Scholar, Bharathiyar university Coimbatore, India,

Rajeswari Mukesh
SRM Institute of Science and Technology, Ramapuram, Chennai, India.

Abstract
Security problems in computer networks are common nowadays due to various attacks that are initiated from outsiders and internal users. Cloud network resources are widely utilized for the benefit of customers to build their own services or products without having physical infrastructures. The massive count of cloud customers from public and private sectors is growing day by day to utilize the services. Notably, the distributed cloud networks are more convenient to manage the resources with optimal computation cost. At the same time, the customers need efficient security principles and secure cloud perimeter conditions to proceed with particular service. On the base, various research works are developed to provide appropriate distributed information security models in distributed cloud environment. Yet these research works are not significant in terms of providing content aware encryption models, data optimized confidentiality models and light weight intelligent encryption solutions. In this case, the proposed Intelligent Data Security for Cloud Computing using Partial Homomorphic Encryption (IDSHE) model is created to solve the issues. Under this proposed IDSHE model, procedures such as novel network modelling principles, data modelling principles, deduplication phases, Long Short Term Memory (LSTM) based data analysis principles and distributed and restricted homomorphic encryption algorithms are developed. The proposed model has been implemented and results are compared with crucial existing techniques. The result section shows that the proposed model overcomes the performance of exiting techniques by nearly 14% of growth rate.

Index terms—: cloud network, security, homomorphic encryption, deduplication and LSTM.

Introduction
Cloud computing and cloud resource optimization tasks are highly expected for various application specific services. The cloud services are provided for public services and private services depends upon user requests. The security metrics expected by private and public users are completely different and unique according to service level agreement. In addition, the cost of opting required cloud services are varying from user to user. In this domain, users are demanding the appropriate and optimal services for the costs.

The cloud network deployment models and logical configuration principles are deciding the benefit levels of each service. The services of cloud network are given for various users under

infrastructure level, software level and platform level principles. At any level, the expectations of security constraints are inevitable and more significant against malicious events.

Distributed cloud networks are deployed to ensure the even handling of workload and avoiding single node failure at any cost. Notably, distributed cloud networks are created using multiple physical systems that are operating various services at client level and server level. In addition, the installation of Virtual Machines (VM) is required to operate multi-level requests and responses. VMs of cloud network are installed at both physical server machines and physical client machines. The entire physical and virtual hosts are functioning to provide distributed cloud services at software points, platform solutions and infrastructure solutions.

On the platform, the data stored and retrieved in the cloud network are traversing around multiple shared resources. For example, multiple systems are providing data services and information management solutions in this cloud environment. At this situation, the information security models play a major role among shared cloud environment. The information security models and network security models are configured in the distributed systems using the strategies such as access control mechanisms, encryption techniques, trust model constructions, event validations, authentication functions, signature functions and multi-level security procedures.

Among these techniques, most of the existing works are developing security solutions and encryption procedures to the whole data packets and the files stored in cloud storages. Basically, the encryption techniques are categorized as either block chiper and stream chiper modes. These are suitable for improving stored data security and real-time data security respectively. In this case, homomorphic encryption techniques are emerging to make complex encryption phases through post-chiper computations in its encrypted mode. These homomorphic encryption techniques are classified as fully encryption functions and partial encryption functions.

Popović et al. [1] described the cloud service types and security principles. In addition, this work provided the details of protecting the data and functions of networks in terms of confidentiality, integrity, authentication and non-repudiation principles. At the same time, the involvement of multiple security features in to cloud environment shall be configured according to user level security agreements. From this work, the basic needs and deployment essentials of cloud security features are identified.

Harfoushi et al. [2] provides data security challenges in cloud environment and distributed network environment. According to this work, information security shall be incorporated through properly implemented encryption techniques and authentication techniques. The encryption and authentication solutions are termed as cryptography models such as block cipher encryption, stream cipher encryption in terms of data handling strategies. At the same time, the cryptography models are classified as public key frameworks and private key

frameworks that are applied in data protection mechanisms. Anyhow, the need for various crypto models is completely depending upon the cloud service limits to the respective user.

Khanezaei et al. [3] and Suthir et al. [4] are discussing the possibilities of modern encryption mechanisms enabled for cloud security principles. In this manner, Rivest Shamir Adelman (RSA), Advanced Encryption Standard (AES), Twofish, RC-4, RC-5, elliptic curve cryptography, digital signature models are discussed with notable technical features. The important point to note is to improve or customize the existing security techniques helps to provide more convenient data security solutions in cloud platforms.

Among these cryptography solutions, the application of Machine Learning (ML) and Deep Learning (DL) techniques with homomorphic encryption models helps to improve the quality and the benefit of cloud security features. In general, the conventional encryption techniques and homomorphic models are completely observing and encrypting the payload (files) without considering the legitimacy of cloud contents [5][6][7]. This is considered as major drawback and it can be noted as a crucial research problem under distributed cloud models. Moreover, the conventional homomorphic encryption techniques consume more time and computational resources. Since the need for intelligent frameworks for improving the efficiency of encryption techniques, the proposed model has been developed with more improvised deep learning based homomorphic encryption techniques and deduplication phases.

The proposed contributes the security of distributed cloud networks with following features.

- Data Extraction and Analysis Model-LSTM
- Multi-Level Authenticated Encryption and Homomorphic Solutions
- Effective Data management and Deduplication
- Fusion of AES, MAC and Homomorphic Principles

Based on the contributions, the proposed article is written from literature survey section that describes the details of exiting techniques and limitations. Section 3 holds the technical information and suitable security procedures of proposed model. Section 4 has been narrated with the implementation details, result analysis. Section 5 concludes the proposed article with its notable contributions and future demands.

Literature Survey

The importance of understanding and analyzing the contributions of existing techniques is mandatory to decide the model of proposed techniques. At the same time, improvising the quality of proposed functions happens through the proper identification of the limitations in existing techniques. Jayashri et al. [8] proposed partial homomorphic encryption technique and role dependent user control security policies in amazon web cloud platform. In this scheme, each user is categorized under multiple roles like end user, network administrator, system administrator, database administrator, local network monitor and others. Based on the user roles, each access is validated for legitimacy status to proceed safe cloud transactions. Anyhow, the role based access control mechanism is conventional approach against effective attacks.

Srividhya et al. [9] proposed multi-party homomorphic encryption technique and access control mechanisms to secure the network. In the same manner, multi-level threshold functions were provided to classify the normal user and attackers in the network. This approach is good on behalf of threshold evaluation models yet the computations methods were not efficient and fault tolerant against many attacks.

Sarkar et al. [10] applied homomorphic encryption techniques for enabling secure medical data prediction panels. In particular, this approach was developed to provide secure data management and prediction schemes for cancer disease. The importance of this model was stated on the basis of secure patient and disease data management portal for hospitals. On contrast, this methodology was not properly trained to classify the actual data and malicious data.

Asharov et al. [11] and Gennaro et al. [12] proposed multi-party encryption schemes and computation analysis procedures. On behalf of encryption procedures, computations and computation overload are increasing gradually around the nodes in the cloud network. Usually, the encryption and decryption techniques are carried out in edge devices or centralized computing devices. In this regard, these existing techniques were discussed on the load sharing and encryption principles for multi-party security schemes.

In the same manner, Antwi-Boasiako et al. [13] proposed distributed public key infrastructures with key generation engines and deep learning techniques. In this methodology, the security flaws happened due to improper third party authorities and vulnerable certificates. Against the security certificate flaws, this methodology was enriched with certificate-less adaptive homomorphic encryption techniques and deep learning algorithms. In this techniques, deep learning based secure datasets were used to observe the malicious events in certificate logs. At the same time, the effective observation of network and storage events were missed out and these were considered as major limitations.

On the stream, Chitrapu et al. [14] proposed new application of homomorphic encryption techniques for improving biometric data security using machine learning approaches. As growing technology reaches various domains such as medical, industrial, defense and other sectors, the security aspects are also being improved. In this connection, internet of medical things is required with completely automated and secured environment solutions. Hence, the data security and medical user validations were considered as major needs in this methodology. Moreover, this technique applied complex homomorphic encryption solutions on generating secure biometric data and validating the same for total protection.

Nivethitha et al. [15] and Jin et al. [16] developed the fusion of federated learning techniques and homomorphic encryption techniques to build a knowledgebase for improving the quality of security procedures. As most of the data security techniques are evolving with encryption methods, the importance of homomorphic encryption lies in the complex computations on encrypted data itself. At the same time, these techniques were developed for ensuring the use

of classification models, training procedures, knowledgebase models and integrated learning procedures in order to increase the toughness of cipher data along cloud environment. At the same time, the load increased in each system has to be noted seriously.

Similarly, Alzubi et al. [17] proposed multiple key generation and deep learning based homomorphic encryption techniques for secure data transmission. In particular, this technique was identifying the security techniques for medical data protection and validation. Generally, the medical data diagnosis models require more accurate data at each end to assure the state of disease and issue a treatment phase (L1). In this situation, a single data breach can violate the originality of medical data and create a serious issue in patient treatment and medical support solutions. Thus the importance of medical data security is improved using homomorphic encryption technique.

Kumar et al. [18] and Wibawa et al. [19] proposed deep learning based cryptographic techniques, role validation techniques, real-time security issues and user mapping techniques for developing the optimal security perimeter in medical cloud environment (L2 and L3). In the same manner, these techniques were not producing classified results on behalf of static and transit data security cases. On the ground, various security principles were introduced and developed using homomorphic encryption functions and deep learning functions [20][21][22]. Anyhow, the exiting techniques were not good enough for understanding and extracting the distributed cloud data before it reaches security engines and data security algorithms.

The proposed technique is modelled and developed to build an efficient and optimized homomorphic functions through authenticated encryption techniques. Particularly, each step in proposed model has been functioned at distributed yet controlled VMs on behalf of data packets validation procedures and file validation procedures. This proposed fusion model assures the existence of deep learning based homomorphic encryption techniques and valid data extraction phases to minimize the computation load at each session. Section 3 shows the crucial steps of proposed model.

Proposed Security Model

The proposed IDSHE model has been configured with multiple phases such as network model, data model, multi-channel data distribution model, LSTM-based packet extraction and file attribute extraction and authenticate homomorphic encryption. Equation (1) shows the data collection and distributed cloud model.

Data collection and distributed trust model,

$$DM^{CLO} = \begin{cases} TM^{CLO}(k,i) + M(c)\,, P_M \geq 1 \\ SM^{CLO}(k,i) + S(c), F_B \geq 1 \\ 0\ , \quad Null \end{cases} \tag{1}$$

$\boldsymbol{TM^{CLO}(k,i)}$: $\boldsymbol{Cloud\ Packet\ Data\ and\ identifier}$

$\boldsymbol{SM^{CLO}(k,i)}$: $\boldsymbol{Cloud\ Disk\ Data\ and\ idenfifier}$

Equations (2)→(3) provides the trust cost computations for packet data and file data block at each VM. In this case, VMs are implemented for processing both client and server actions depend upon distributed network principles.

$$M(c) = \begin{cases} t(P_s; P_t; S; D; T_s), & 1 \geq P \\ t(T_{lc}; T_{gc}), & 1 \geq P \\ t(m_a; T_m), & 0 \end{cases} \quad (2)$$

$$S(c) = \begin{cases} b(N, N_D, D_T, D_S), & 1 \geq B \\ b(TB_{lc}; TB_{gc}), & 1 \geq B \\ t(bm_a; bT_m), & 0 \end{cases} \quad (3)$$

$\boldsymbol{M(c)}$: $\boldsymbol{Packet\ Trust\ cost}$
$\boldsymbol{S(c)}$: $\boldsymbol{Data\ Block\ Trust\ Cost}$

According to this cost model, packet attributes such as sequence number, internal trust cost (local node computation), global or network cost, session time, source and destination address, timestamp and crucial priority bits are taken as the most values attributes for packet analysis. Similarly, data block, file attributes, authentication properties and owner details are considered as crucial storage properties for trust cost computation procedures.

$\boldsymbol{t(T_{lc}; T_{gc})}$: $\boldsymbol{Local\ cost\ and\ Global\ cost}$
$\boldsymbol{t(m_a; T_m)}$: $\boldsymbol{malicious\ event\ and\ time}$
$\boldsymbol{P}$: $\boldsymbol{Packet\ Trust\ Cost\ Threshold}$
$\boldsymbol{b(N, N_D, D_T, D_S)}$: $\boldsymbol{Data\ block\ number, Data\ fragments, Time\ and\ Data\ Source}$
$\boldsymbol{b(TB_{lc}; TB_{gc})}$: $\boldsymbol{Block\ local\ cost\ and\ global\ cost}$
$\boldsymbol{t(bm_a; bT_m)}$: $\boldsymbol{Malicious\ block\ and\ time}$
$\boldsymbol{B}$: $\boldsymbol{Block\ Trust\ Cost\ Threshold}$

On the base cloud network model, the identification of both client and server VMs is necessary to classify the nodes as indicated in equation (4).

$$Net^{CLO} = \begin{cases} VM\_c(k,i) + bc(c), & 1 \\ VM\_s(k,i) + bs(c), & 1 \\ 0, & Null \end{cases} \quad (4)$$

$\boldsymbol{VM_{c(k,i)}}$: $\boldsymbol{virtual\ client\ machines(i), k\ blocks}$
$\boldsymbol{bc(c)}$: $\boldsymbol{block\ computation\ function\ in\ VM\ client}$
$\boldsymbol{VM_{s(k,i)}}$: $\boldsymbol{virtual\ server\ machines(i), k\ blocks}$
$\boldsymbol{bs(c)}$: $\boldsymbol{block\ computation\ function\ in\ VM\ server}$

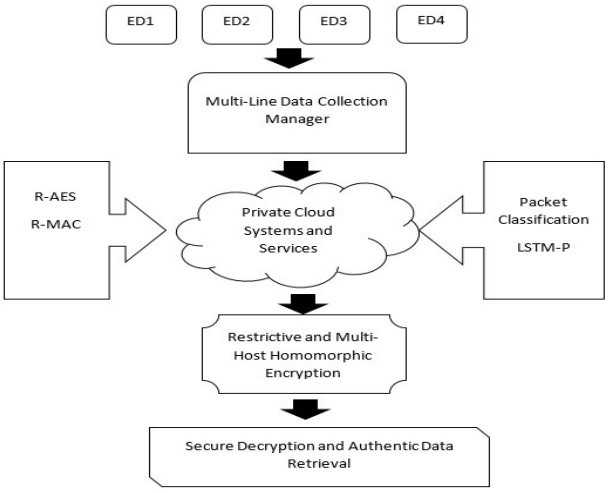

Fig. 1. System Architecture-IDSHE

As indicated in figure 1, the proposed IDSHE model phases are designed and implemented. As shown, the data are collected from multiple edge devices or users in to cloud environment. In the cloud, VMs are configured as either server units or client units. From these states, the proposed procedures are initiated using Restricted-AES and H-MAC procedures according to extracted features of packets and file data.

Cloud model and Data Modelling

P1: Cloud model and Data Modelling
Input: Cloud Nodes and Data Streams Output: Modelled Networks 1. Set all nodes (clients and servers) 2: Initiate VM client and VM server configuration metrics 3:Initiate node discovery process and logical deployment process 4: Call cloud model, $\boldsymbol{Net^{CLO}}$ 5: Initiate Data Model, $\boldsymbol{Data^{CLO}(M(c): S(c))}$ 6:Call $\boldsymbol{Data^{CLO}(M(c): S(c))}$ at $\boldsymbol{VM\_s(k,i)\ and\ VM\_c(k,i)\ \in Net^{CLO}}$ 7: Get data in to all $\boldsymbol{VM_{s(k,i)}\ and\ VM\_c(k,i)\ \in Net^{CLO}}$ 8: Call procedure 2 9. Repeat for all $\boldsymbol{VM_{s(k,i)}\ and\ VM\_c(k,i)\ \in Net^{CLO}}$
End

Procedure 1 illustrates data modelling and cloud network modelling based on separately configured VM clients and VM servers.. In this the all the VM clients and Servers are initialized. Procedure 2 shows the importance aspects of data collection points [24][25].Procedure 2 states the activation of interfaces to collect data from edge devices.

Similarly, the packet details are collected from the interfaces of interconnecting devices such as routers and switches.

Multi-line Data Distribution in to VMs

P2: Multi-line Data Distribution in to VMs

Input: $\boldsymbol{Data^{CLO}(M(c): S(c))\ at\ VM\_s(k,i)\ and\ VM\_c(k,i)}$

Output: $\boldsymbol{M(c,c): S(c,c)\ and\ M(s,c): S(s,c)}$:
client and server

1: Find the interfaces $\boldsymbol{I(c,e), I(s,e), I(VM,c), I(VM,s)}$ at computing points
$\boldsymbol{I(c,e)}$: $\boldsymbol{interfaces\ of\ edge\ client}$
$\boldsymbol{I(s,e)}$: $\boldsymbol{interfaces\ of\ edge\ server}$
$\boldsymbol{I(VM,c)}$: $\boldsymbol{Interfaces\ of\ vitual\ client}$
$\boldsymbol{I(VM,s)}$: $\boldsymbol{Interfaces\ of\ vitual\ server}$
2: Find the interfaces
$\boldsymbol{I(S,R)}$: $\boldsymbol{Interfaces\ of\ internal\ switches\ and\ routers}$
$\boldsymbol{I(G)}$: $\boldsymbol{Interfaces\ of\ cloud\ network\ gateways}$
3: Initiate cloud disk storage model,

$$\boldsymbol{D(cloud, interface)} = \begin{cases} \boldsymbol{D_i(S, VM(c,s))}, & \boldsymbol{Data \geq 1} \\ \boldsymbol{D_i(S,C)}, & \boldsymbol{Data \geq 1} \\ \boldsymbol{Others}, & \boldsymbol{Disable} \end{cases}$$

4: Enable all interfaces under secure port access principles
5: Do $\boldsymbol{If\ only\ I \in Secure\_Policy(Net^{CLO})}$
5.1: Get data from $\boldsymbol{I}$ and check credentials $\boldsymbol{M(c): S(c)}$
5.2: Call $\boldsymbol{TM^{CLO}(k,i)\ at\ I\ (Recall\ step\ 1\ and\ 2)}$
5.3: Call $\boldsymbol{SM^{CLO}(k,i)\ \ at\ I\ (D(cloud, interface), step\ 3)}$
5.4: Get the data in to VM clients and VM servers
6: Redo for all data at VMs and storages

End

Procedure 3 shows the implementation of LSTM to extract the packet attributes and storage file attributes. LSTM network of this proposed model has been trained to identify the legitimate and malicious features of packet data and file data. In addition, LSTM engines are configured with data preprocessing techniques to resolve the issues such as data duplication, noisy data, missing data and other data mismatches.

The data gathering and distributed cloud paradigm is demonstrated using LSTM-based packet extraction, file attribute extraction, and authentication homomorphic encryption. It provides the trust cost estimations for packet data and file data blocks at each VM. In this instance, VMs are used to process both client and server activities that rely on distributed network concepts.

For packet analysis, factors including sequence number, internal trust cost (calculated at the local node level), global or network cost, session duration, source and destination addresses, timestamp, and critical priority bits are considered to be the most important properties. Similar to data blocks and file attributes, authentication properties, owner information, and ownership details are regarded as essential storage aspects for trust cost computation processes.

LSTM-P-Data Dissemination and Packet Analysis

P3: LSTM-P-Data Dissemination and Packet Analysis

Input: $\boldsymbol{I\left(Data, TM^{CLO}(k,i)\right)}$: $\boldsymbol{I(Data, SM^{CLO}(k,i))}$

Output: Extracted and Classified Data

1: Get $\boldsymbol{I\left(Data, TM^{CLO}(k,i)\right)\ at\ network\ level}$

2: Get $\boldsymbol{I(Data, SM^{CLO}(k,i))\ at\ storage\ level}$

3: Make stream of packet data, $\boldsymbol{I\left(Data, TM^{CLO}(k,i)\right)}$ and collect at local buffer

4: Make storage level buffer for $\boldsymbol{Data, SM^{CLO}(k,i)}$

5: Call data pre-processing and detect duplications and suspicious events

6: Get trusted data through buffer, $\boldsymbol{TM^{CLO}(k,i)}$ and $\boldsymbol{SM^{CLO}(k,i)}$

7: Call $\boldsymbol{LSTM(I\left(Data, TM^{CLO}(k,i)\right), I\left(Data, SM^{CLO}(k,i)\right))}$

7.1: $\boldsymbol{Ip = sig(w(data) * I(Data) + h(t-1) * U + b(data)}$

7.2: $\boldsymbol{Op = tangent(w(out) * I(Data) + h(t-1) * U + b(out)}$

7.3: $\boldsymbol{hp = Op.tangent(c.state) : c.state: cell\ state}$

7.4: $\boldsymbol{c.state = Ip.c.state_1 + f.c.state(t-1)}$

8: Get de-duplicated data streams in buffer, $\boldsymbol{B(TM^{CLO}(k,i); Data, SM^{CLO}(k,i))}$

9: Call P4

10: Repeat for all data in cloud environment.

End

At the end of LSTM procedures, each packet data streams and file data streams are processed and malicious events are detected using KDD dataset features. At the same time, the event logs and other abnormal activities are classified effectively to start the proposed homomorphic encryption procedures [26].

Distributed Homomorphic Encryption and Decryption [R-AES:R-MAC]

Procedure 4 illustrates the authenticated AES and R-HMAC procedures with random homomorphic encryption functions on cipher text. In this proposed model, the malicious events, malicious packets, duplicated packets, malicious data injection and harmful data items are eliminated to minimize the computation load of encryption procedures.

VMs are set up as either server units or client units in the cloud. According to the properties of the collected packet and file data, the proposed operations are started from these states using Restricted-AES and H-MAC protocols. Following the completion of the R-AES:R-MAC

operations, the features from the KDD dataset are used to process each packet data stream and each file data stream and identify harmful events. The proposed homomorphic encryption techniques are launched concurrently once the event logs and other anomalous activity are effectively classified.

P4: Distributed Homomorphic Encryption and Decryption [R-AES:R-MAC]

Input: $\boldsymbol{B(TM^{CLO}(k,i); Data, SM^{CLO}(k,i))}$
Output: Encrypted and Authenticated data

1: Get $\boldsymbol{B(TM^{CLO}(k,i); Data, SM^{CLO}(k,i))}$
2: Call IDS engine to classify malicious log events in each buffer
3: Get alert report of IDS engine and remove malicious data at each buffer
4: Call $\boldsymbol{R-MAC(m) \forall\, TM^{CLO}(k,i)/Data, SM^{CLO}(k,i)}$
4.1: $\boldsymbol{R-MAC(m) = h^i \oplus pad^i}$
4.2: $\boldsymbol{R-MAC(m,K) = K \rightarrow (h^i \oplus pad^i)}$
4.3: $\boldsymbol{R-MAC(TM^{CLO}(k,i),k) = K \rightarrow [TM^{CLO}(k,i), k(h^i \oplus pad^i)]}$
4.4: $\boldsymbol{R-MAC(SM^{CLO}(k,i),k) = K \rightarrow [SM^{CLO}(k,i), k(h^i \oplus pad^i)]}$
5: Call $\boldsymbol{R-AES(m)}$
5.1: $\boldsymbol{AES(R-MAC(TM^{CLO}(k,i),k), K_A) \rightarrow Chiper\ Text}$: C
5.2: $\boldsymbol{AES(Chiper\ Text, K_A) \rightarrow Data}$
6: Invoke homomorphic encryption function
6.1: $\boldsymbol{S1: H(C,l) = l \rightarrow [C \oplus R_b \oplus Nonce]}$
6.2: $\boldsymbol{S2: H(S1,l1) = l1 \rightarrow [S1 + R-MAC(m) - Nonce]}$
7: Get encrypted and authenticated data from each interface
8: Verify the data at each virtual server and virtual client
9: Validate the packets at each core (router, switches and gateways) interfaces
10: Repeat and decrypt data based on authentic user request.

End

Finally, the proposed model has been implemented as given in procedure 4. The cloud environment setup and results are illustrated in section 4.

Cloud Testbed and Results

The experimental setup for comparing proposed IDSHE model and the existing techniques L1, L2 and L3 is simulated using cloud simulator. Under this environment, cloud simulator is used to build the network environment and python functions are used to implement the proposed models and existing techniques. Under this scenario, 50 VMs are created including both server editions and client editions under 120 physical machines. In this case, set of machines are acting as physical devices for managing storage points and monitoring tasks. The performance evaluation has been observed through the parameters such as data encryption time (milliseconds, msec), energy spent (joules), computation overload, secure data rate (Kilo Bits Per Seconds, Kbps), memory complexity (Kilo Bytes, KB) and internal IDSHE phase execution time (msec).

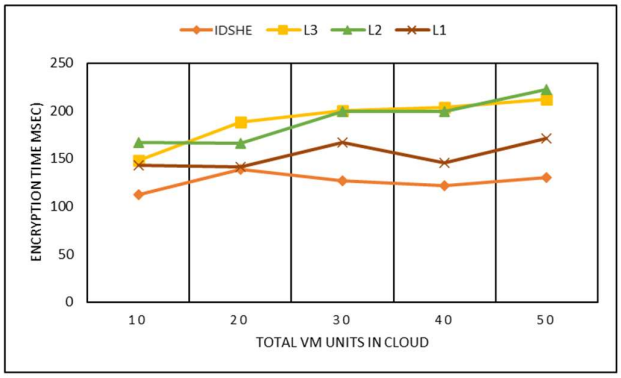

Fig. 2. Data Encryption Time

In this evaluation, the proposed IDSHE model has been compared with the existing techniques such as L1 (deep learning and mealy state model), L2 (deep learning based role validations) and L3 (Convolutional Neural Network with learning techniques). Figure 2 depicts data encryption time (msec) against the total number of VMs created in distributed environment. In this experiment, the encryption time of IDSHE varies from 110 msec to 125 msec as the VMs are changing gradually. At the same time, L1, L2 and L3 are varying from 15o msec to 220 msec which are more than proposed model. In the proposed model, LSTM and suitable pre-processing phases are applied to simplify the encryption load that are not available in existing techniques. Thus the proposed model achieves better time reduction rate than other techniques.

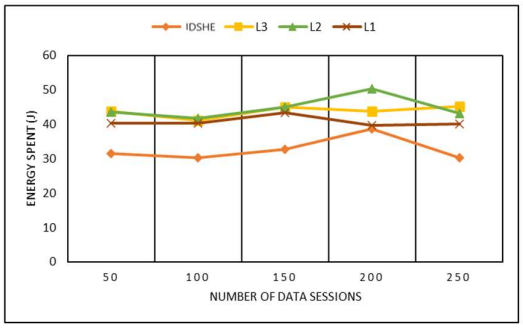

Fig. 3. Energy Consumption

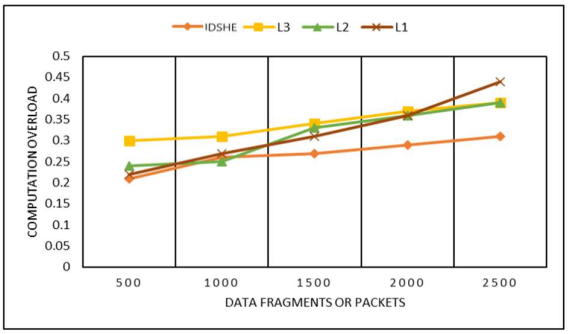

Fig. 4. Computation Overhead

Fig.3 gives the benefits of using IDSHE in energy domain than other works L1, L2 and L3. In this experiment, the energy spent by proposed model is minimal than other works (31 joules to 38 joules). As the existing techniques are not properly handling the data using preprocessing approaches and any data reduction techniques [26][27]. Thus the energy is not optimal for those techniques. In the same case, computation cost is also higher and directly match the energy consumption rate for the security systems. In the comparison of computation cost, the proposed model ensures optimal load (0.26) at maximum while number of data or packets increases during the sessions.

Fig.4 has depicted for describing the comparison of IDSHE and other techniques. As the number of computational procedures increases, the energy spent in each VM or physical machine increases which is unavoidable. Thus the importance of data reduction and lightweight procedures has to been taken seriously.

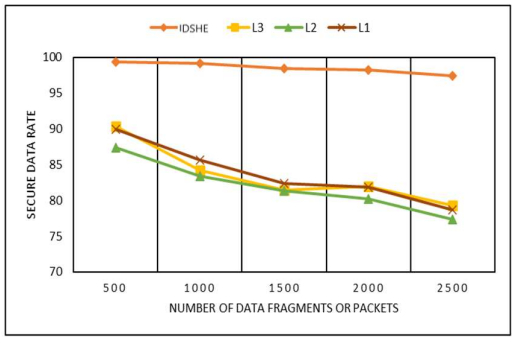

Fig.5. Secure Data Rate

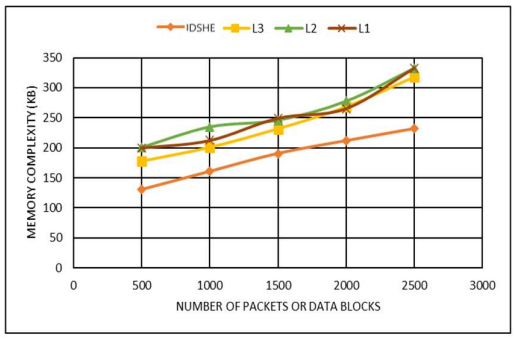

Fig.6. Memory Consumption

Fig.5 and Fig.6 are showing the details of secure data rate and memory expectations of the security systems respectively. In this comparison, secure data rate shall be optimized where the encryption procedures are customized and data is regulated for security procedures. These are achieved by proposed IDSHE model yet the technical aspects are limitted in existing techniques such as L1, L2 and L3. Similarly, the data reduction and lightweight encryption techniques are properly trained in proposed model than L1, L2 and L3. Thus the maximum secure data rate and memory consumption rate are optimal in IDSHE.

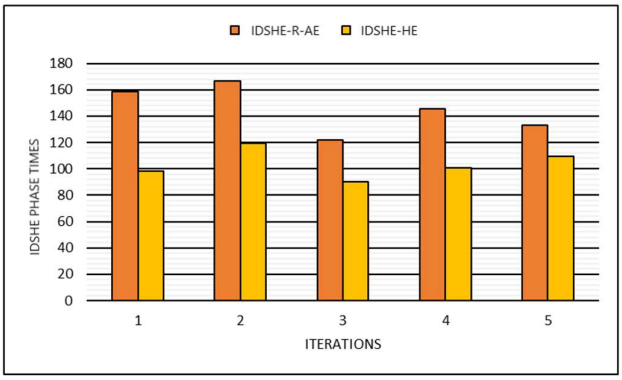

Fig.7. IDSHE Phase Execution Time

Figure 7 extracts the timing constraints of proposed IDSHE under the phases of authenticated encryption and random homomorphic encryption. The first phase includes the time taken to complete R-AES and R-MAC on packet or data encryption at the source VMs. In addition, the next phases involve with random homomorphic procedures to create complex cipher text blocks against attackers in the cloud. The fusion of random homomorphic and authenticated encryption makes complex cipher text generation and validation models in the proposed IDSHE model. For the benefit of security concerns, the time take to complete phase1 (162 msec maximum) and phase 2 (113 msec, maximum) are consumed as valid costs. Thus the proposed IDSHE model gets optimal results than the existing techniques.

Conclusion

The proposed IDSHE has been developed to provide more effective and restricted security mechanisms using novel homomorphic encryption techniques. Generally, most of the encryption techniques are applied in the cloud environment to provide common confidentiality and authentication solutions. In this case, energy wastages and computation load are increasing

rapidly. Against these issues, proposed IDSHE used well-trained LSTM engines, deduplication and preprocessing phases, R-AES and R-MAC functions to generate restricted and authenticated cipher text blocks. On these blocks, random homomorphic functions were applied to make the cipher text more complex against cipher text attackers and malicious events.

Due to numerous attacks launched by both external users and internal users, computer networks now frequently have security issues. Customers frequently make use of cloud network resources to create their own services or products without physical infrastructures. Utilizing the services, a sizable number of people from both the public and private sectors are using the cloud. Dispersed cloud networks, in particular, make it easier to manage resources at the lowest possible cost of computation. In order to proceed with a certain service, clients must also have effective security standards and a secure cloud perimeter. In order to provide suitable distributed information security models in a distributed cloud environment, many research works are built. Due to the novel aspects, the proposed IDSHE model achieved better performance than L1, L2 and L3 as shown in section 4. The future idea to improve the quality of proposed IDSHE has been identified in behalf of random active and passive attacks.

References

1. Popovic K, Hocenski Z. Cloud computing security issues and challenges. InThe 33rd international convention mipro 2010 May 24 (pp. 344-349). IEEE.
2. Harfoushi O, Alfawwaz B, Ghatasheh NA, Obiedat R, Abu-Faraj MA, Faris H. Data security issues and challenges in cloud computing: A conceptual analysis and review. communications and Network. 2014;6(01):15-21.
3. Khanezaei N, Hanapi ZM. A framework based on RSA and AES encryption algorithms for cloud computing services. In2014 IEEE conference on systems, process and control (ICSPC 2014) 2014 Dec 12 (pp. 58-62). IEEE.
4. Suthir S and Janakiraman S, "SNT Algorithm and DCS Protocols coalesced a Contemporary Hasty File Sharing with Network Coding Influence", Journal of Engineering Research, Vol. 6, Issue 3, pp.54-69, 2018
5. Thyagarajan C,et.al, "A Typical Analysis and Survey on Healthcare Cyber Security" in Int. Journal of Scientific and Technology Research, Vol.9, Issue.3, pp.3267-3270, 2020, ISSN: 2277-8616
6. Vijayaraj N, Arunagiri S, Demultiplexer design using photonic crystal ring resonator with high quality factor and less footprint for DWDM application. Opt Quant Electron 54, 465 (2022). https://doi.org/10.1007/s11082-022-03817-2
7. Prabhu D, et.al, "Privacy preserving steganography based biometric authentication system for cloud computing environment", Measurement: Sensors Journal, Vol 24, Dec 2022, https://doi.org/10.1016/j.measen.2022.100511
8. Jayashri C, et.al, "Big Data Transfers through Dynamic and Load Balanced Flow on Cloud Networks", 3rd IEEE International Conference on Advances in Electrical and Electronics, Information, Communication and Bio-Informatics, pp. 342–346, 2017, https://doi.org/10.1109/AEEICB.2016.7538376

9. Srividya M, Anusha N, et.al, "A contemporary network security technique using smokescreen SSL in huddle network server", 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016, pp 673-676. https://doi.org/10.1109/AEEICB.2016.7538376
10. Sarkar E, Chielle E, Gursoy G, Chen L, Gerstein M, Maniatakos M. Privacy-preserving cancer type prediction with homomorphic encryption. Scientific reports. 2023 Jan 30;13(1):1661.
11. Asharov G, Jain A, López-Alt A, Tromer E, Vaikuntanathan V, Wichs D. Multiparty computation with low communication, computation and interaction via threshold FHE. In Advances in Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings 31 2012 (pp. 483-501). Springer Berlin Heidelberg.
12. Gennaro R, Goldfeder S. Fast multiparty threshold ECDSA with fast trustless setup. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security 2018 Oct 15 (pp. 1179-1194).
13. Antwi-Boasiako E, Zhou S, Liao Y, Dong Y. Privacy-preserving distributed deep learning via LWE-based Certificateless Additively Homomorphic Encryption (CAHE). Journal of Information Security and Applications. 2023 May 1;74:103462.
14. Chitrapu P, Kalluri HK. A Survey on Homomorphic Encryption for Biometrics Template Security Based on Machine Learning Models. In2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) 2023 Feb 18 (pp. 1-6). IEEE.
15. Nivethitha, et.al., "Conceptual approach on smart car parking system for industry 4.0 internet of things assisted networks", in Measurement: Sensors, Volume 24, December 2022, https://doi.org/10.1016/j.measen.2022.100474
16. Jin W, Yao Y, Han S, Joe-Wong C, Ravi S, Avestimehr S, He C. FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System. arXiv preprint arXiv:2303.10837. 2023 Mar 20.
17. Alzubi JA, Alzubi OA, Beseiso M, Budati AK, Shankar K. Optimal multiple key-based homomorphic encryption with deep neural networks to secure medical data transmission and diagnosis. Expert Systems. 2022 May;39(4):e12879.
18. Kumar M, Zhang W, Fischer L, Freudenthaler B. Membership-Mappings for Practical Secure Distributed Deep Learning. IEEE Transactions on Fuzzy Systems. 2023 Jan 9.
19. Wibawa F, Catak FO, Kuzlu M, Sarp S, Cali U. Homomorphic encryption and federated learning based privacy-preserving cnn training: Covid-19 detection use-case. InProceedings of the 2022 European Interdisciplinary Cybersecurity Conference 2022 Jun 15 (pp. 85-90).
20. Chen Q, Wu Y, Wang X, Jiang ZL, Zhang W, Liu Y, Alazab M. A Generic Cryptographic Deep-Learning Inference Platform for Remote Sensing Scenes. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2023 Mar 23.

21. M.Bhavithra, V.Nivethitha – "Real Time Sectionalization of Enhanced Sharpness Video using FPGA" in Elysium Journal of Engineering Research and Management, Volume 3, Issue 4, Page No. 23 - 26, August-2016. ISSN: 2347-4408.
22. Sun D, Huang H, Zheng D, Hu H, Bi C, Wang R. Face security authentication system based on deep learning and homomorphic encryption. Security and Communication Networks. 2022 Apr 27;2022.
23. Ibarrondo A, Viand A. Pyfhel: Python for homomorphic encryption libraries. In Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography 2021 Nov 15 (pp. 11-16).
24. Nivethitha V, Sivasubramanian A "Intensification and Interpretation of Performance in 5G adopting Millimeter Wave: A Survey & Future Research Direction", International Arab Journal of Information Technology, Volume 20, Issue Number 4, July 2023.
25. Meftah S, Tan BH, Mun CF, Aung KM, Veeravalli B, Chandrasekhar V. Doren: toward efficient deep convolutional neural networks with fully homomorphic encryption. IEEE Transactions on Information Forensics and Security. 2021 Jun 21;16:3740-52.
26. Wood A, Najarian K, Kahrobaei D. Homomorphic encryption for machine learning in medicine and bioinformatics. ACM Computing Surveys (CSUR). 2020 Aug 25;53(4):1-35.
27. Behera S, Prathuri JR. Application of homomorphic encryption in machine learning. In 2020 2nd PhD Colloquium on Ethically Driven Innovation and Technology for Society (PhD EDITS) 2020 Nov 8 (pp. 1-2). IEEE.