International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

PCF-LSTM for Dynamic Data Portability with CSHHC-Based Deduplication in Distributed Cloud

Anesa Al-Najeh[1], Sana Abouljam[2]
[1]Department of Computer, Faculty of Science, Sabratha, Sabratha University, Sabratha, Libya, +218
[2]Department of Computer, Faculty of Science, Alajelat, Zawia University, Sabratha, Libya, +218
Email address: [1]anesa.alnajeh@sabu.edu.ly, [2]s.abouljam@zu.edu.ly

Abstract*— *With the increase in duplicate data in cloud storage, the storage overhead and data loss have become a major concern. Thus, to remove the duplicate data, the DD technique in the distributed cloud is developed. However, due to provider lock-in and various security threats, the existing models are unreliable. Thus, to provide secure DD and DP in Distributed Cloud (DC), the article proposes CSHHC and PCF-LSTM. Initially, keys are generated for the users while registration. When the authenticated user uploads a file in a DCS, to mitigate security threats and DD shrinkage, the file is split, compressed, and encrypted. Then, for the DP, PCF-LSTM based on the dynamically updated lookup table is used. Then, for DD, the CSHHC technique is proposed. The experimental validation for the security and performance analysis revealed the superiority of the proposed scheme over the existing and state-of-the-art techniques.

Keywords*— *Distributed Cloud Server (DCS), Elliptic Curve Cryptography (ECC), security, Data Deduplication (DD), Data Portability (DP).

I. INTRODUCTION

Cloud service providers cater to the need of individuals by allowing them to store, and backup their ever-increasing amount of data at low cost [1]. Although cloud computing offers a huge amount of storage space, data duplication decreases the performance of cloud storage, and results in poor data management [2]. Thus, capacity optimization in Cloud computing is used to progress storage use by dwindling stored data. Crucial methodologies used for capacity optimization in cloud computing are data compression and data DD [3].

Data DD is an efficient method of data reduction, which aims at eliminating redundant data and improving the storage efficiency of cloud servers [4]. The classic data DD framework consists of a Key Server (KS), a Cloud Storage Provider (CSP), and users to ensure security depending on the trusted KS [5]. The backup DD structures effectively spread the fragments of each datum document through several distributed servers and create privacy [6]. However, outsourcing data to semi-trusted distributed cloud servers increases further security challenges [7] and DP problems. Thus, to provide security, the data were encrypted and uploaded to the DCS. But, the existing encryption solutions for DD in cloud storage either suffer brute-force attacks or incur large time overheads [8].

Portability is the capacity of exchanging workloads and information between cloud providers when data cannot be ported crosswise over a heterogeneous Platform [9]. However, there is a lack of awareness and motivation amongst users and practices for data export and import as core problems [10]. Thus, to solve these problems, the PCF-LSTM based portability with CSHHC based DD is performed.

A. Problem Statement

The existing works for DP and DD have the following drawbacks,

- Existing works neglected the unsuitability in the versions of the cloud when data is ported to the DCs, which makes the file as corrupted file during the downloading.
- In existing work, the semi-trusted CSP could cheat by providing wrong DD check results in order to increase the cost of storage.
- In existing works, the DD performed for large data, such as multi-media causes the DD shrinkage.

By analyzing these setbacks, the objective of the proposed framework is to develop secure DD and portability in the DC and its contributions are,

- To solve the version deviation problem, the most suitable cloud is selected with the PCF-LSTM, which is trained with a dynamically updated lookup table.
- Thus, to provide integrity, secure CSH based hash chaining to verify the duplication in the cloud is proposed.
- To avoid DD shrinkage, LZW compression is introduced.

The rest of this paper is organized as follows; Section II analyses the related works of the proposed model. Sections III and IV describe the proposed methodology and its corresponding experimental results. Section V concludes the paper.

II. RELATED WORKS

[11] implemented a secure DD and DP scheme in DCS using hash chaining and Levy Flight–Wind Driven Optimization techniques. The experimental results revealed a higher potential of the developed model. But, there was a lack of confidentiality without proper authentication.

[12] developed multi-key management for secure data DD in DCS. The system was developed based on the modified Residual Network (M-ResNet). The M-ResNet-based model achieved superior results in experimental analysis. However, the model took more time to process the data.

[13] recommended secure encrypted data DD based on data popularity. Here, bilinear mapping checked whether the encrypted data was from the same plain text. The experimental results proved the practicability and efficiency of the scheme.

Anesa Al-Najeh and Sana Abouljam, "PCF-LSTM for Dynamic Data Portability with CSHHC-Based Deduplication in Distributed Cloud," *International Research Journal of Advanced Engineering and Science*, Volume 9, Issue 1, pp. 61-66, 2024.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

When large data was used, the possibility for DD shrinkage was more.

[14] introduced a secure DD and recovery model based on public key encryption. The data users' re-encryption key about the file was appended to the cipher-text chain table. The security and efficiency analysis proved the reliability of the model. The DD cannot be done properly when there was a slight change in the new input file.

[15] presented optimized Ciphertext-Policy-Attribute-Based-Encryption for secure data transmission on DCS. The DP and DD were performed by the Hybrid Meerkat Clan Algorithm and SHA512. The experimentation outcomes revealed the better performance of the technique. But, the DD process took more time.

[16] suggested Enhanced Symmetric Key Encryption Algorithm (ESKEA) DD for secure data storage. The keys were selected based on spider monkey optimization. The results revealed that the ESKA scheme was more effective. However, the data stored in the semi-trusted CSP caused a security threat to user data.

[17] provided a verifiable data DD scheme with integrity and duplication proof. The correction of data DD was checked by a User Determined Duplication Check Protocol. The security analysis proved the correctness of the provided schemes. However, with increased users, the computational cost got increased.

III. PROPOSED SECURE DD WITH DP METHODOLOGY

The storing of duplicate data in distributed cloud servers causes storage problems along with security concerns. Thus, to solve this problem, CSHHC deduplication with PCF-LSTM based portability in DCSP is proposed and the architecture was given in Figure 1.

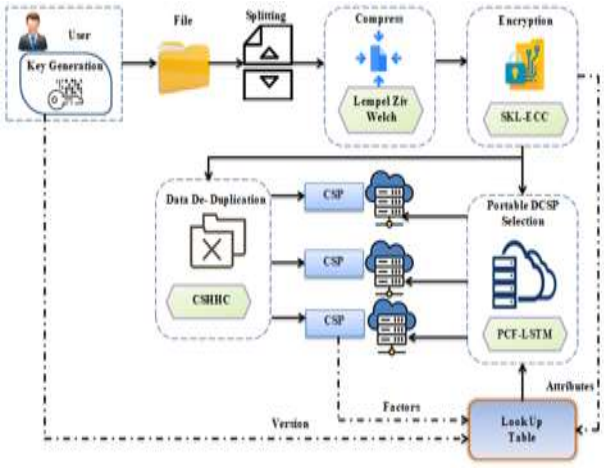

Fig. 1. Architecture of the proposed framework.

A. Key Generation

Initially, users upload the data to the DCSPs after registration with the corresponding network. During registration, the keys are generated by Symmetric Kullback-Leibler-based Elliptic Curve Cryptography (SKL-ECC). Here, the ECC is selected due to higher security. But, the exploitation of public parameters in the ECC causes the side-channel attack. Thus, to solve this problem, an individual key is generated for the users with the SKL technique.

Generator point (P) selection: Initially, a generator point is selected from the elliptic curve equation,

$$n = m^3 + d.m + j$$

Where, m, n depicts the coordinates of the curve, and d, j depicts the integer values.

Individual key computation: Thus, with P and the private keys (K_1, K_2) at the user and server side, the public keys are formed (ρ_1, ρ_2). Where, (K_1, K_2) are the random prime numbers and the public keys (ρ_1, ρ_2) are obtained by,

$$\rho_1 = K_1.P$$
$$\rho_2 = K_2.P$$

Then, the individual key of the user (I_{user}) using the SKL technique is,

$$I_{user} = \rho_2.\log\frac{\rho_2}{K_1} + K_1.\log\frac{K_1}{\rho_2}$$

B. Data Splitting

After the keys are obtained, the user stores a file in the DCSPs by splitting the input file (f) to x datum files. The partitioned data is denoted as,

$$f \rightarrow [f_1, f_2, ..., f_x] or\ f_Q,\ \ Q = 1, 2, ..., x$$

C. Data Compression

Then, to reduce the deduplication shrinkage, f_Q lossless compression of the file with the Lempel-Zev-Welch (LZW) technique is performed. The file f_Q is compressed using a table-based lookup model in the LZW algorithm by performing encoding the information in the file. The table formed is called a dictionary or code table. The single byte from the input file f_Q is coded with the codes of 0-255. While encoding, only the first 256 entries are present in the dictionary. The compression is achieved by using the 256 codes through 4095 entries to represent the sequence of bytes.

During compression, LZW identifies repeated sequences in f_Q and then adds them to the dictionary. Suppose the string in the file f_Q is denoted as $WYS*WYGWYS*$, it is compressed with LZW in encoded form as,

$$WYS*WYGWYS* = 8789834225671256258$$

Then, Decoding is achieved by taking each code from the compressed file and translating it through the code table by finding the representation of those characters. Thus, the compressed file is depicted as C_Q.

D. Data Encryption

After file compression, to provide security of the corresponding file in the DCSP, encryption with the SKL-ECC is performed.

Anesa Al-Najeh and Sana Abouljam, "PCF-LSTM for Dynamic Data Portability with CSHHC-Based Deduplication in Distributed Cloud," *International Research Journal of Advanced Engineering and Science*, Volume 9, Issue 1, pp. 61-66, 2024.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

During encryption, cipher texts $e_Q = (\tau_1, \tau_2)$ are formed as,

$$\tau_1 = P.k$$
$$\tau_2 = C_Q + I_{user} + k.\rho_2$$

Where, k depicts a random integer number.

E. Portable DCSP Selection

After the data is secured, the encrypted data is uploaded to the best DCSP with the help of Parameterized Clipping Function-based Long Short Term Memory (PCF-LSTM). Here, the LSTM is selected due to its advantage of predicting the output with previous memory. But, the existing LSTM takes more time to train its parameters. Thus, to solve this problem, the gates are activated with Parameterized Clipping Function (PCF) activation. The architecture of PCF-LSTM is in Figure 2.

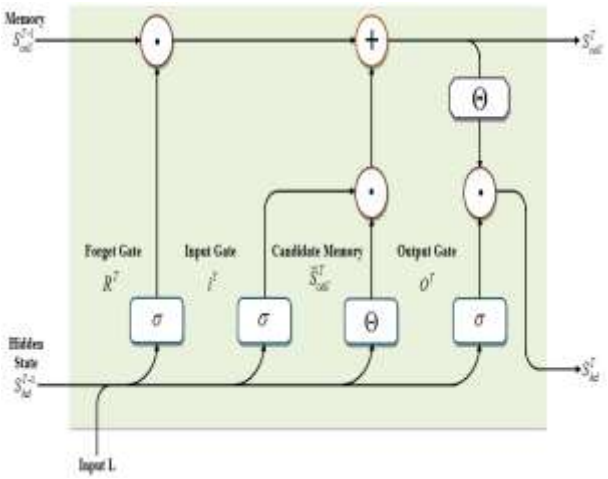

Fig. 2. Architecture of PCF-LSTM.

Lookup Table: The versions of the cloud and the user device might get unsuitable to port the data to the most suitable cloud. Thus, the PCF-LSTM is dynamically updated with the lookup table, which contains the factors $(F_b, b = 1, 2.., M)$ of the cloud, such as bandwidth, versions, and response time; attributes (a_α) of the encrypted file, such as size; version (V_β) of the device the user used. These factors are dynamically updated on the lookup table (L), which is mathematically represented as,

$$L = \{F_b, a_\alpha, V_\beta\}$$

Input: The PCF-LSTM receives the L, along with spontaneously generated two vectors within the PCF-LSTM (i.e. hidden state (S_{hd}) and cell state (S_{cell})), which is taken as the input at time instant (T). Given the three input vectors (L, S_{hd}, S_{cell}), the PCF-LSTM regulates vectors (L, S_{hd}, S_{cell}) through the forget, input, and output gates $(\aleph_g, i_g, o_g)$ as,

Cell state (S_{cell})***:*** The state combines the data from all the previous time steps $(T-1)$ in each memory cell of PCF-LSTM by,

$$S_{cell} = \Re_g . S_{cell}^{T-1} + i_i . \bar{S}_{cell}$$

Where, S_{cell}^{T-1} is the previous cell state, and $\bar{S}_{cell}$ is the generated new memory in the cell state,

$$\bar{S}_{cell} = \Theta\left(\varpi_{S_{cell}} . \left[S_{hd}^{T-1}, L\right] + B_{S_{cell}}\right)$$

Here, ϖ represents weight value, B depicts bias, Θ represents the PCF activation, which is denoted as,

$$\Theta(S) = \begin{cases} 0, & if\ S_{cell} \in (-\infty, 0) \\ S_{cell} & if\ S_{cell} \in [0, \delta) \\ \delta & if\ S_{cell} \in [\delta, +\infty) \end{cases}$$

Where, δ limits the dynamic range of activation to $[0, \delta]$.

Output: Finally, the PCF-LSTM output is determined as,

$$S_{hd}^T = o_g \times \Theta\left(S_{cell}^T\right)$$

Where, S_{hd}^T represents the final output of PCF-LSTM. The pseudocode of PCF-LSTM is given as,

Pseudocode of PCF-LSTM

Input: Lookup table
Output: Optimal CSP
Begin
  Initialize states (S_{hd}, S_{cell}), $i_g, \Re_g, o_g, T$
  Set $S_{hd} = 0, S_{cell} = 0$
  For time $(Time \le T)$ **do**
    Estimate gates output $S_{cell}, i_g, \Re_g$
    Update cell state with $\bar{S}_{cell}$
    Activate states with PCF Θ
    Determine o_g result
  End For
  Return output S_{hd}^T
End

After the PCF-LSTM is updated with the lookup table, when encrypted data is received, the PCF-LSTM selects the most suitable DCS.

F. Data Deduplication

Meanwhile, when encrypted data is uploaded in the DC, the data duplication is performed with Chaos System HAVAL-based Hash Chaining (CSHHC). The HAVAL hashing is selected to form hash chaining due to its efficiency in producing hashcodes of different lengths in less time. But, the manual intervention for specifying the number of passes in the HAVAL reduces its efficiency. Thus, to solve this problem, the Chaos System (CS) technique is included to select the number of passes.

Anesa Al-Najeh and Sana Abouljam, "PCF-LSTM for Dynamic Data Portability with CSHHC-Based Deduplication in Distributed Cloud," *International Research Journal of Advanced Engineering and Science*, Volume 9, Issue 1, pp. 61-66, 2024.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Message padding: For obtaining hash values, the input message (i.e., encrypted file e_Q) value is padded in N blocks, which is a multiple of 1024. The last block of the padded message (b_{N-1}) contains the number of bits in the unpadded message, the required number of bits in the output hash value, and the number of passes each message block is processed. The padded message of N is denoted as,

$$b_{pad} \to \{b_{N-1}, b_{N-2}, ..., b_0\}$$

Here, each b_{pad} is a 1024-bit block.

Compression: The Chaos System HAVAL (CSH) hashing starts from N_0 and a random 256-bit (8-word) constant string $(\aleph_{word})$,

$$\aleph_{word} = \omega_{0,7}\omega_{0,6}...\omega_{0,0}$$

The ω value is taken from the fraction of π. The ω compresses the b_{pad} in a block-by-block manner by,

$$\omega_{word+1} = \Delta\left(\aleph_{word}, b_{pad}\right)$$

Here, $\Delta(\ )$ signifies the basic compression function, whose working steps are given as follows,

The $\Delta(\ )$ applies 3 to 5 passes, which are denoted as, $\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5$. The output of $\Delta(\ )$ is the compressed output $(\aleph^{out}_{word})$, the internal computation parameter (c) as,

$$c_0 = \aleph_{word}$$
$$c_1 = \Delta_1\left(c_0, b_{pad}\right)$$
$$c_2 = \Delta_2\left(c_1, b_{pad}\right)$$
$$c_3 = \Delta_3\left(c_2, b_{pad}\right)$$
$$c_4 = \Delta_4\left(c_3, b_{pad}\right) \; if \;\; pass = 4,5$$
$$c_5 = \Delta_5\left(c_4, b_{pad}\right) \; if \;\; pass = 5$$

Then, the number of passes is selected with the CS technique as,

$$\aleph^{out}_{word} = \begin{cases} c_3 \oplus c_0 & pass = 3 \\ c_4 \oplus c_0 & pass = 4 \\ c_5 \oplus c_0 & pass = 5 \end{cases}$$

The number of passes is selected with the cs technique as,

$$pass_n = \gamma.z\left(b_N - b_{N-1}\right) * m_N.y\left(1 - b_N\right) * u.\omega_0$$

Where, $z, y, u, \gamma \; and \; m_N$ are the chaos parameter and chaos condition, respectively. Then, by performing the rounding functions, $\aleph^{out}_{word}$ for $\aleph_{word}$ is obtained.

Folding: The length of the hashed output (h) is determined using the folding technique, which adjusts the length of the hashcode as,

$$h = \text{mod}(l) \sum_{word=0}^{N-1} \aleph^{out}_{word}$$

Where, l indicates the bit length. Thus, the output hash value of CSH h is obtained.

Hash Chaining: After h is obtained, Hash Chain (HC) is created to perform deduplication. In the HC, the hash value of each file is stored in each block.

If a hashcode gets into the HC, the CSP verifies whether the data is in the HC. If the hash value is present in the hash chain, the CSP provides the reference link of the already available data to the data user. Else, the encrypted file will be stored in the DC. Thus, by performing this process, the DD is performed.

G. Data Retrieval

After the data is stored in a cloud, if the user device is updated and the cloud remains in the same version, the next optimal cloud is selected with PCF-LSTM. Then, for the deduplication, the authentication code (h) handover between the CSPs takes place. Thus, the users retrieve the file effectively without misjudging it as a corrupted file. During the data retrieval, the decryption takes place as,

$$C_Q = (\tau_2 - k.\tau_1) - I_{user}$$

As the individual key is generated with the parameters of users, the data is retrieved more securely.

IV. RESULTS AND DISCUSSION

In this section, the experimental analysis of the proposed model is performed in comparison with the baseline techniques on the PYTHON. The data is collected by sample real time network deployment.

A. Performance Analysis

In this section, the performance of the proposed algorithms is analyzed with conventional techniques.

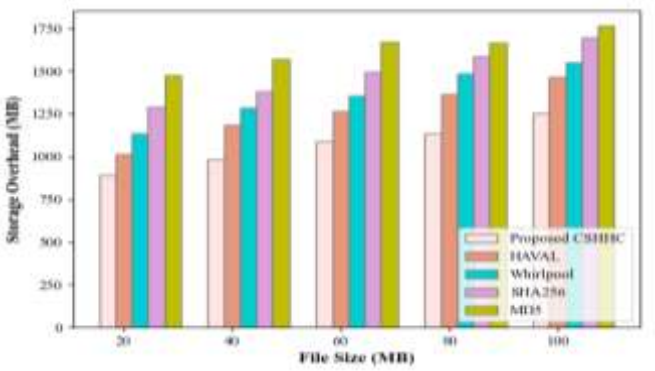

Fig. 3. Deduplication Efficiency.

In Figure 3, the deduplication efficiency is analyzed based on the storage overhead for the varied size of the files. Here, for the 40MB of data, the overhead was 985, whereas the storage overhead of Simple Hashing Algorithm (SHA256) and Message Digest (MD5) was 1378 and 1569MB, respectively. This shows that with the CSHHC, the deduplication is performed effectively.

Anesa Al-Najeh and Sana Abouljam, "PCF-LSTM for Dynamic Data Portability with CSHHC-Based Deduplication in Distributed Cloud," *International Research Journal of Advanced Engineering and Science*, Volume 9, Issue 1, pp. 61-66, 2024.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

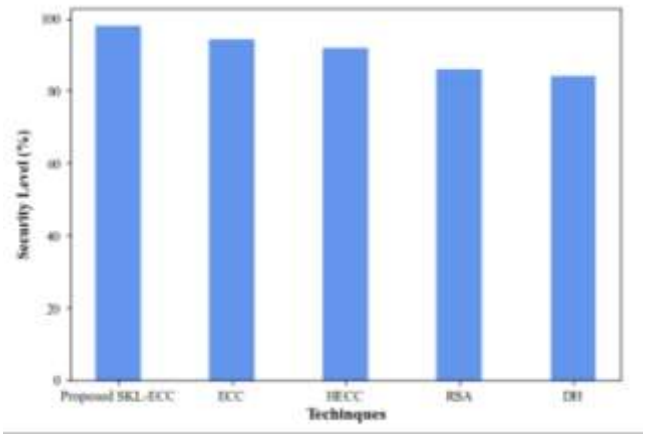

Fig. 4. Security Level Analysis.

The security analysis for the proposed SKL-ECC in comparison with the traditional ECC, Hyper ECC, Rivest Shamir Adleman (RSA), and Diffie Hellman (DH) is shown in Figure 4. Here, the security attained by the SKL-ECC is 3.57% and 13.83% higher than the ECC and DH techniques, respectively. This shows that with SKL-ECC, more security for the data is provided.

TABLE I. Response time analysis.

Algorithms	Response Time (ms)
Proposed PCF-LSTM	793
LSTM	826
GRU	984
Bi-LSTM	1194
RNN	1376

Table I unveils the response time attained by the proposed and baseline classifiers in experimental analysis. The response time of the proposed PCFLSTM is 33ms, 221ms, and 583ms lesser than the LSTM, BiLSTM, and RNN, respectively. This ensures that the DP is performed in less time with the proposed PCFLSTM technique.

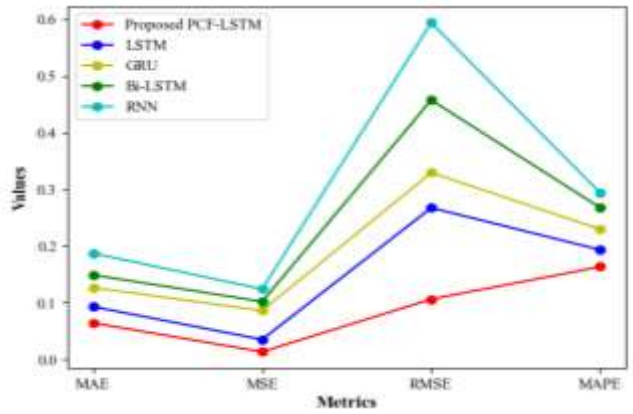

Fig. 5: Error Analysis.

Figure 5 analyses the Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) for the proposed PCF-LSRM, LSTM, Gated Recurrent Unit (GRU), Bidirectional LSTM (Bi-LSTM), and GRU. Here, among the existing algorithms, the LSTM achieved less MAE (0.0935), MSE (0.0355), and RMSE (0.2635). But, with the PCF technique in the LSTM, the errors are further reduced. Thus, it is concluded that with PCF-LSTM, DP in DC performed with more accuracy in the proposed framework.

TABLE III. Attack level analysis

Algorithms	Attack Level (%)
SKL-ECC	6
ECC	10
HECC	15
RSA	17
DH	20

The attack level of the proposed and the existing systems was analyzed in table II. Here, the attack level of the ECC is 10%, which is a lesser attack level than the other conventional techniques. But, with the SKL technique in the ECC technique, the attack level gets reduced by 4%. This shows that the SKL-ECC technique was more resistant to the attacks than the other algorithms.

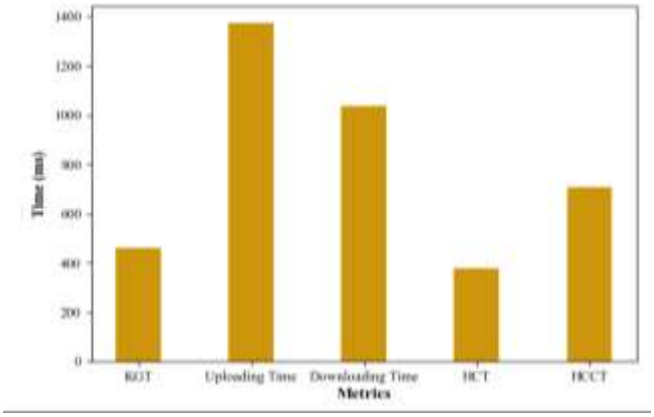

Fig. 6: Time Analysis of the Proposed Model.

In Figure 6, the proposed model for Key Generation Time (KGT), Hashcode Creation Time (HCT), HC Creation Time (HCCT), file uploading time, and downloading time are analyzed. Here, the KGT, HCT, HCCT, uploading time, and downloading time of the proposed model is 461ms, 379ms, 706ms, 1371ms, and 1034ms, respectively. This shows the time efficiency of the proposed model.

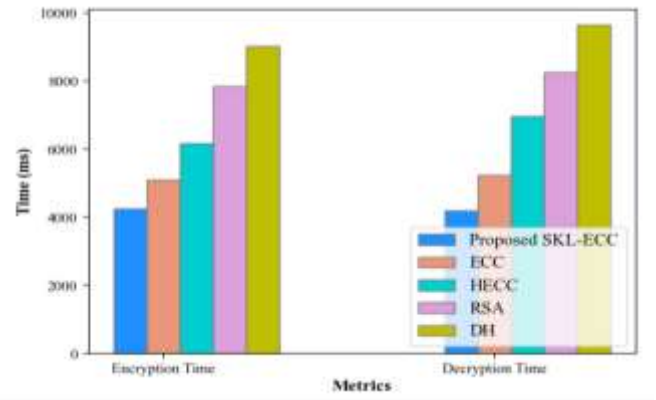

Fig. 7: Encryption and Decryption Time Analysis.

Anesa Al-Najeh and Sana Abouljam, "PCF-LSTM for Dynamic Data Portability with CSHHC-Based Deduplication in Distributed Cloud," *International Research Journal of Advanced Engineering and Science*, Volume 9, Issue 1, pp. 61-66, 2024.

In Figure 7, it is unveiled that the existing ECC technique takes less time of 5105ms and 5243ms to encrypt and decrypt the data. This time was much less than that of the prevailing HECC, RSA, and DH algorithms. But, the proposed SKL-ECC attained 859ms and 1036ms less encryption and decryption time than the ECC technique, which makes SKL-ECC more suitable for data encryption and decryption in the proposed model in less time.

B. *Comparative Analysis*

Here, the proposed models' security over existing techniques is analyzed in comparison with [16], [15], [11].

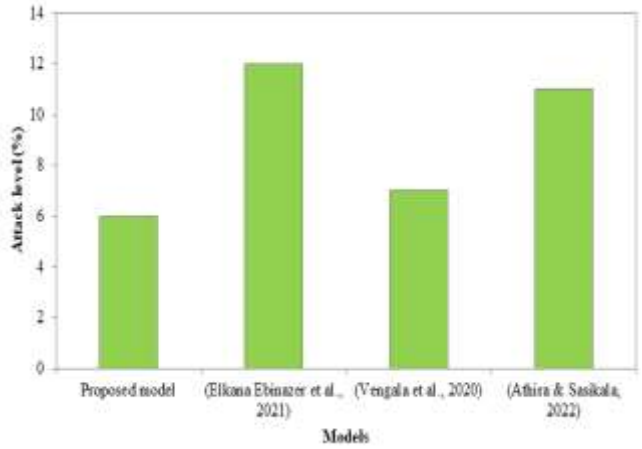

Fig. 8: Comparative Analysis.

The security of the deduplication and portability models is analyzed based on the attack level in the network. Here, for the proposed model, the attack level was 6%. But, the [16], [15], [11] models attained 50%, 16.66%, and 45.45% higher levels of attacks than that of the proposed model. This shows that the security of the proposed deduplication and portability model was better than any other technique.

V. CONCLUSION

This article proposed a secure data DD and portability scheme in the DC based on the dynamically updated lookup table. The security and efficiency of the proposed methodology are experimentally analyzed. During the experimental analysis, the proposed CSHHC attained less storage overhead. The SKL-ECC attained a higher security level of 97.94% in less attack level in less time. Then, the PCF-LSTM revealed a reliable DP with a better response time of 793ms. Finally, the comparative analysis revealed the superiority of the proposed model over existing models for data DD and portability based on system security. But, in this model, the user authorization is preserved based on the individual key only. Thus, to enhance the integrity of data, blockchain-based authorization can be developed in the future along with the proposed model.

REFERENCES

[1] Prajapati, P., & Shah, P. (2022). A Review on Secure Data Deduplication: Cloud Storage Security Issue. Journal of King Saud University - Computer and Information Sciences, 34(7), 3996–4007. https://doi.org/10.1016/j.jksuci.2020.10.021.

[2] Vijayalakshmi, K., & Jayalakshmi, V. (2021). Analysis on data deduplication techniques of storage of big data in cloud. Proceedings - 5th International Conference on Computing Methodologies and Communication, 976–983. https://doi.org/10.1109/ICCMC51019.2021.9418445.

[3] Magesh Kumar, S., Balasundaram, A., Kothandaraman, D., Auxilia Osvin Nancy, V., Sathish Kumar, P. J., & Ashokkumar, S. (2020). An approach to secure capacity optimization in cloud computing using cryptographic hash function and data de-duplication. Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, 1256–1262. https://doi.org/10.1109/ICISS49785.2020.9315892.

[4] Zheng, X., Zhou, Y., Ye, Y., & Li, F. (2020). A cloud data deduplication scheme based on certificateless proxy re-encryption. Journal of Systems Architecture, 102, 101666. https://doi.org/10.1016/j.sysarc.2019.101666.

[5] Zhang, D., Le, J., Mu, N., Wu, J., & Liao, X. (2021). Secure and Efficient Data Deduplication in JointCloud Storage. IEEE Transactions on Cloud Computing, 7161, 1–12. https://doi.org/10.1109/TCC.2021.3081702.

[6] Chavhan, S., Pati, P., & Patle, G. (2020). Scheme for Distributed Cloud Storage. Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020, 1406–1410).

[7] Olakanmi, O. O., & Odeyemi, K. O. (2021). Faster and efficient cloud-server-aided data de-duplication scheme with an authenticated key agreement for Industrial Internet-of-Things. Internet of Things (Netherlands), 14, 100376. https://doi.org/10.1016/j.iot.2021.100376.

[8] Fu, Y., Xiao, N., Chen, T., & Wang, J. (2022). Fog-to-MultiCloud Cooperative Ehealth Data Management with Application-Aware Secure Deduplication. IEEE Transactions on Dependable and Secure Computing, 19(5), 3136–3148. https://doi.org/10.1109/TDSC.2021.3086089.

[9] Ramalingam, C., & Mohan, P. (2021). Addressing semantics standards for cloud portability and interoperability in multi cloud environment. Symmetry, 13(2), 1–18. https://doi.org/10.3390/sym13020317.

[10] Kuebler-Wachendorff, S., Luzsa, R., Kranz, J., Mager, S., Syrmoudis, E., Mayr, S., & Grossklags, J. (2021). The Right to Data Portability: conception, status quo, and future directions. Informatik-Spektrum, 44(4), 264–272. https://doi.org/10.1007/s00287-021-01372-w.

[11] Athira, A. R. (2022). Secure Data Deduplication and Data Portability in Distributed Cloud Server Using Hash Chaining and LF-WDO. Data Analytics and Artificial Intelligence, 2(1), 7–13. https://doi.org/10.1007/s11277-022-09735-6.

[12] Nandha Kumar, R., Sathiya, T., Mohanta, H. C., Ahmad, S. S., Kumar, A., Alshammri, G. H., & Atiglah, H. K. (2022). Secure Data Deduplication System with Cyber Security Multikey Management in Cloud Storage. Security and Communication Networks, 2022, 1-11. https://doi.org/10.1155/2022/9790398.

[13] He, Y., Xian, H., Wang, L., & Zhang, S. (2021). Secure Encrypted Data Deduplication Based on Data Popularity. Mobile Networks and Applications, 26(4), 1686–1695. https://doi.org/10.1007/s11036-019-01504-3.

[14] Li, L., Zheng, D., Zhang, H., & Qin, B. (2023). Data secure de-duplication and recovery based on public key encryption with keyword search. IEEE Access, 11, 1-11. https://doi.org/10.1109/ACCESS.2023.3251370.

[15] Vengala, D. V. K., Kavitha, D., & Kumar, A. P. S. (2020). Secure data transmission on a distributed cloud server with the help of HMCA and data encryption using optimized CP-ABE-ECC. Cluster Computing, 23(3), 1683–1696. https://doi.org/10.1007/s10586-020-03114-1.

[16] Elkana Ebinazer, S., Savarimuthu, N., & Mary Saira Bhanu, S. (2021). ESKEA: Enhanced Symmetric Key Encryption Algorithm Based Secure Data Storage in Cloud Networks with Data Deduplication. Wireless Personal Communications, 117(4), 3309–3325. https://doi.org/10.1007/s11277-020-07989-6.

[17] Yu, X., Bai, H., Yan, Z., & Zhang, R. (2023). VeriDedup: A Verifiable Cloud Data Deduplication Scheme with Integrity and Duplication Proof. IEEE Transactions on Dependable and Secure Computing, 20(1), 680–694. https://doi.org/10.1109/TDSC.2022.3141521.