International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6S, August 2019

Minimizing the Security Risks in Hybrid Cloud Networks with the Aid of Optimal Triple Data Encryption Standard Algorithm

Subhash Chand Gupta, Vikas Kumar

Abstract*: Security is one of the challenging problems in the hybrid cloud, because of the combination of private and public cloud. A novel algorithm has been presented in this paper, to secure the hybrid cloud networks during information storage and recovery. The algorithm consists of four modules: (i) data collection module, (ii) data de-duplication module, (iii) encrypted de-duplication module and (iv) user interface module. Initially, the document is collected using the data collection module and duplicate data is removed using the matching score mechanism. This de-duplicated data is encrypted using the Optimal Triple Data Encryption Standard (OTDES). To enhance the performance of TDES, oppositional monarch butterfly optimization algorithm (OMBOA) is used in hybrid combination with TDES and this is referred as OTDES. This encrypted data is then stored on the cloud. The stored information can be accessed through the user interface module. Performance evaluation of the proposed security system has been carried out on the basis of different evaluation metrics and have been found in good order.

Keywords*: Security, Optimal Triple Data Encryption Standard, monarch butterfly optimization algorithm, hybrid cloud, encryption.

I. INTRODUCTION

With the tremendous growth of technology, activities of the people, and organisations have changed (Al). Lot of activities utilise the internet of service for attaining their daily activities and this causes abundance of data everywhere (Kumar & Bhardwaj, 2018). Traditionally, people used to buy and maintain hardware devices on their premises to store this data, however this has become hard to maintain in the present scenario. This problem is being tackled effectively by using the cloud storage (Vidhyalakshmi & Kumar, 2016). Using this storage, the data can be stored in the third party servers, using the pay-per-use model (Singh & Kumar, 2013). The Cloud Service Provider (CSP) provides the service to the users in terms of their request (Lin et al., 2018) and the user need to pay for the amount to the service they use (Al). The process of cloud computing is based on virtual machines (VMs), which are used for provisioning the user's applications into the service provider's datacenters (Mao et al., 2018, Yang et al., 2017). The services of cloud can be further extended by federated hybrid cloud model, which uses a collaborative service of various cloud service providers. Federated cloud facilitates the sharing of infrastructure, enhancing the capacity, load balancing, and savings on cost and energy.

The federated cloud is highly recommended as effective services for cloud deployment (Esposito et al., 2016, Abdi et al., 2017). A hybrid cloud model can be utilized based on two perspectives such as privacy and bursting. A privacy perspective means that sensitive data can be utilized only in the private network rather that in public network, whereas a bursting perspective means the resources of the neighborhood are deficient, consequently additional open cloud assets are needed for dealing with the workload peaks (Labba et al., 2018). Hybrid cloud additionally gives different administrations to the cloud applications by upgrading the memory, the efficiency of processors and the facilities of networking. The storage of data and recovery algorithms were utilized in the ordinary databases, the fundamental networks and the operating system which can be adjusted to be compatible with the hybrid cloud. Furthermore, the data security is still questionable as the users are not sure about their data storage location by the third party servers (Vidhyalakshmi & Kumar, 2017). In addition, they also fear about the issues of data migration, modification and backup etc. So, security guarantee should be given for the user to enable their data to the cloud Some of the security issues are highly risky and require high concentration, for example: application security, encryption and key administration, physical access control etc. There are various systems that utilize the hybrid cloud to perform redistributed calculation with privacy preservation (Sturrus & Kulikova, 2014). In terms of providing highly secured platform for cloud user, the cloud service provider developed service level agreement. However, unfortunately there is no standard procedure to construct an SLA (Kumar and Pradhan, 2013). Along with this, traditional technique are utilized for security purposes, such as firewalls, host-based antivirus software and intrusion detection system etc. However, all of these techniques can not guarantee the security in hybrid cloud environment (Subramanian & Jeyaraj, 2018). This work is focused on developing an effective technique towards the hybrid cloud security. A security mechanism during information storage and retrieval from the cloud database has been proposed in the present work.

Manuscript published on 30 August 2019.
***** Correspondence Author (s)
Subhash Chand Gupta, Mewar University, Chittorgarh – 312901, Rajasthan, India.
Vikas Kumar, Chaudhary Bansi Lal University, Bhiwani – 127021, Haryana, India.

II. LITERATURE SURVEY

Lot of researchers have explained the data security and duplication process in cloud. An overview of the research issues in cloud security have been presented by Singh and Kumar (2014). Zheng et al., 2016) have explained a data de-duplication process in cloud environment. They have performed the de-duplication process on encrypted data and combined this with access control. Mohammad et al. (2018) have explained a signcryption algorithm based security. They have utilized 6LoWPAN mechanism to take care of the fragment duplication attack. To face the fragment duplication attack, the lightweight Offline-Online SignCryption (OOSC) scheme was presented. Many techniques and tools are being utilized for providing a highly levelled security with the quick advancement of Internet. A split algorithm has been presented by Priyansha et al. (2018) to transfer data on cloud servers, which increases the security of the system. They have presented a concept of data encryption, utilizing the steganography and stego-image splitting, in addition to using split algorithm for more security.

Moreover, Prabu et al. (2018) have explained a privacy preserving mechanism for cloud using the fully homomorphic–elliptic curve cryptography algorithm. They have encrypted the original information using ECC algorithm and again encrypted the data using FH algorithm. The access control unit was used to verify the user information and secure the data. Sridevi et al. (2018) have explained the networked medical data sharing on a secure medium. Developing interest for e- healthcare insurance over the globe has raised worries towards the safe and confirmation upgraded restorative image sharing. Here, a security upgraded DICOM picture sharing over a LAN addressing confidentiality, integrity, and authentication was clarified. At first, the AES scrambled patient history was joined alongside the thumb impression and Quick Response (QR) code of patient ID as a watermark. This watermark was scrambled utilizing the Integer Wavelet Transform (IWT), tumultuous guide and attractors with disarray dispersion activities. Then, the encoded watermark was inserted in the chosen Region of Non-Interest (RONI) pixels of DICOM picture. Username and extraordinary secret word certifications, Face distinguishing proof and FPGA created One Time Password (OTP) frame the three-layer validation conspire to protect DICOM picture access through the LAN. Web distributing medium of storing secured DICOM pictures in the cloud was tended to in this work.

Fouad et al. (2018) have presented secure and privacy-preserving data sharing in the cloud based lossless image coding. Methodology used compressing the picture, utilizing lossless image coder with the end goal to make space before encryption. This space is then loaded up with a haphazardly created succession and joined with an encoded adaptation of the compressed bitstream to frame a full goals scrambled picture in the pixel area. The CSP utilizes this extra space in the scrambled picture to include extra information and produces an encoded picture containing extra information in a comparative mold. Surveyed with the lossless Embedded Block Coding with Optimized Truncation (EBCOT) tech on characteristic pictures, the plan has been successful to surpass the limit of 3 bpp of extra information, while keeping up information security and privacy. Similarly, Varsha & Shedge (2018) have explained a Secure Data Deduplication and Auditing for Cloud Data Storage. UR-MLE2 scheme helps to check deduplication whereas data auditor is responsible for data integrity checking. To improve the system performance, dynamic binary decision tree is used to check data deduplication. Dynamic binary tree updates the tree data as per user modification or deletion of user data. The system performance will be evaluated based on execution time.

III. PROPOSED DATA SECURITY IN HYBRID CLOUD

The primary aim of the proposed methodology is to store the document in the cloud with high security in the way of encryption and decryption by the formation of De-duplication dataset utilizing various approaches. The security and redundancy are the major risks in hybrid cloud. To reduce the security risk, encryption algorithm have been used in the present work. Whereas, to reduce the redundancy, de-duplication process has been used. The process of de-duplication is to avoid the repeated data or multiple times of same data and the level of De-duplication achievable depends on a number of factors. Figure 1 provides the overall information flow of the proposed methodology of data security. The proposed methodology uses three modules:

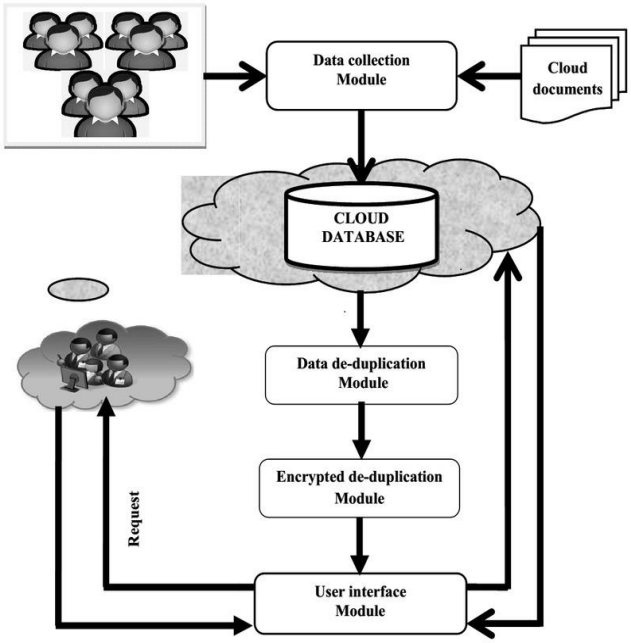

Figure 1: Proposed Data Security Information Flow

3.1 Data collection module

Data collection module is the first module in the process that collects the documents to be kept secure in the hybrid cloud. These documents will be collected from the cloud users from different organisations and places. For the purpose of calculations, let us consider that N_i is the total number of documents collected and in the cloud data bas D.

3.2 Data De-duplication Module:

The collected cloud database may contain number of duplication documents. These duplicated documents would increase the storage requirement and risks of the cloud. Therefore, removal of the duplicate document from cloud database is very much essential. The data de-duplication method is similar to a data compression process that is utilized to remove the repeated document in cloud storage. This method can be also utilized to improve the use of storage and to transfer data which is utilized to minimize the quantity of bytes primarily that must be sent. Only a single copy of the content is maintained by de-duplication method rather than maintaining the numerous copies of the data.

A new algorithm has been proposed for de-duplication process. At first, we select the first document D_1 from cloud database D. Then, we select the first line L_1 from first document D_1 for calculating similarity using cosine similarity formula which is given in equation (1). If the similarity score S^S is greater than threshold T^H means, we remove the line from particular document. Similarly, we can remove the document which has more similarity score S^s. Finally, we store the less similarity documents. The steps involved in data de-duplication process is explained in table 1

$$Similarity\ S^S = \cos\theta = \frac{\vec{A}.\vec{B}}{\|\vec{A}\|.\|\vec{B}\|} \quad (1)$$

Where, $\vec{A}, \vec{B}$ are the term frequency vectors of the documents.

Table 1: De-duplication Process

Input: Documents Output : de-duplicated document Step 1: consider the cloud data base D and from the database D, select the first document D_1 Step 2: From the first document D_1, select the first line L_1. Step 3: After that verify the resemblance for the first line L_1 with other lines $(L_1, L_2, ..., L_n)$ of the document D_1. Step 4: Find the similarity score S^S utilizing equation (1). Step 5: If the similarity score S^S is greater than T^H then from the document, dismiss the extract line (L_i). Step 6: The documents D_i that are containing more contents which are alike is to be dismissed Step 7: De-duplicated document

3.3 Encryption module

After the de-duplication process, we have to encrypt the documents using optimal Triple Data Encryption Standard algorithm (OTDES) for reduce the security risk in hybrid cloud. The OTDES algorithm is a combination of triple DES and oppositional monarch butterfly optimization algorithm (OMBOA). The idea of Triple DES algorithm is same as DES algorithm with the exception of we apply it three times. The DES is a Symmetric encryption algorithm, which requires 2 input sources: a key and a plaintext. The plaintext's length is 64 bits, and the key is likewise 64 bits (among them, just 56 bits are ever utilized). However, the length of 56-bit is the impediment of DES. So as to upgrade the encryption's quality, triple DES algorithm is designed to expand the key length of DES. The TDES is additionally called EDE (Encrypt-Decrypt-Encrypt) which is a more secure version of DES. Also, a TDES algorithm has three keys with 168 bits as key length (three 56-bit DES keys). In TDES approach three keys are chosen i.e. Key 1, Key 2 and Key 3 (K^1, K^2&K^3). TDES algorithm uses three cycles of normal DES cipher. It receives a mystery 168-piece key that is isolated into three 56-bit keys. The key process of TDES include:

- Encrypt the plain test with K^1 that is PK^1 (d)
- Decrypt the encrypted text with K^2 that is OK^2 (d)
- Encrypt the cipher text with K^3

TDES runs three times slower than DES; however is significantly more secure whenever used genuinely. The method for decoding information is equivalent to the methodology for encryption, aside from it is executed in reverse. For improving the execution of TDES process, in this paper, we optimally select the three keys such as K_1, K_2 and K_3. To optimize the keys, here, we utilize the oppositional monarch butterfly algorithm (OMBA) which is the combination of oppositional based learning (OBL) and monarch butterfly algorithm (MBA). To increase the searching ability of the MBA, in this paper we hybrid OBL with MBA. The encryption module consist of two processes namely, optimal key generation process and OTDES based encryption process.

3.3.1 Optimal key generation process

In TDES algorithm, normally we utilize three keys namely K^1, K^2 and K^3 which are randomly selected. To increase the system security, in this paper we optimally select the keys using oppositional monarch butterfly algorithm (OMBA). etaheuristic algorithm is MBO.

It is enlivened by the conduct of the monarch butterfly amid movement. In MBO, every one of the monarch butterfly people are glorified and situated in 2 lands only: Land 1 and Land 2. Similarly, the locations of the monarch butterflies are updated in two different methods: the movement operator and the butterfly altering operator.

a) Movement Operator

When monarch butterflies remain at Land 1 and Land 2, their quantities in Land 1 and Land 2 can be refered as $ceil(q*SP)$ $(SP_1, Subpopulation1)$ and a $SP-SP_1$ $(SP_2, Subpopulation2)$, respectively according to the time. Here, $ceil(y)$ rounds y to the closest integer greater than or equal to y ; the population size is denoted as SP ; the ratio of monarch butterflies in Land 1 is q . Formulated below is this migration process,

$$y_{u,n}^{v+1} = y_{r_1,n}^{v} \quad (2)$$

Where $y_{u,n}^{v+1}$ indicates the nth element of y_u at generation $t+1$. Likewise, $y_{r_1,n}^{v}$ denotes the nth element of y_{r_1}. The current generation number is v. Butterfly r_1 is randomly chosen from $Subpopulation1$. When $r \le q$, $y_{u,n}^{v+1}$ is generated by equation (3). Here r can be calculated as,

$$r = rando * perio \quad (3)$$

Where $perio$ denotes migration time and is set to 1.2 in the basic MBO method. Random number is represented by $rando$. If $r > q$, $y_{r_1,n}^{v}$ is generated by,

$$y_{u,n}^{v+1} = y_{r_2,n}^{v} \quad (4)$$

Where $y_{r_2,n}^{v}$ indicates the nth element of y_{r_2} , and butterfly r_2 is randomly selected from $Subpopulation2$. Based on the analyses as introduced above, the Movement operator can be summarized as shown in Algorithm 1.

Table 2: Movement Operator

```
Begin
For u = 1 to SP1 (for all monarch butterflies in Subpopulation1 ) do
    For n = 1 to d (all the elements in ith monarch butterfly) do
        Randomly generate a number rando using equation (3)
        If r ≤ q then
            Randomly select a monarch butterfly in Sub population 1;
                Generate y(u,n)^(v+1) by equation (2)
        Else
            Randomly select a butterfly in Subpopulation2;
            Generate y(u,n)^(v+1) by equation (4)
        End if
    End for n
End for u
End.
```

a) Butterfly altering Operator

The locations of monarch butterflies except migration operator can be updated by the upcoming butterfly adjusting operator. The following describes the procedure of butterfly adjusting operator. For every elements in butterfly t, if $rando \le q$ it can be updated as,

$$y_{t,n}^{v+1} = y_{best,n}^{v} \quad (5)$$

Where $y_{t,n}^{v+1}$ indicates the nth element of y_t at generation $v+1$. Similarly, $y_{best,n}^{v}$ indicates the nth element of the fittest butterfly y_{best}.

If $rando > q$, it can be updated as

$$y_{t,n}^{v+1} = y_{r_3,n}^{v} \quad (6)$$

Where $y_{r_3,n}^{v}$ indicates the nth element of y_{r_3}. Here, $r_3 \in \{1,2,...,SP_2\}$.

Under this condition, if $rando > BAR$, further it can be updated as follows,

$$y_{t,n}^{v+1} = y_{t,n}^{v+1} + \alpha(dy_n - 0.5) \quad (7)$$

Where butterfly adjusting rate is denoted by BAR. dy is the walk step of butterfly t which can be calculated by the following equation,

$$dy = (y_t^v) \quad (8)$$

In equation (6), the weighting factor is α as shown in equation (8),

$$\alpha = S_{\max} / v^2 \quad (9)$$

Where $S_{\max}$ is the maximum walk.

OMBO based key generation

In this section, we optimally select the key value using OMBO algorithm. Step by step procedure of key generation is as follows:

Step 1: Solution initialization

Solution initialization is an important process for all the optimization. In this section, we optimize the key that is present in the TDES. In TDES, we utilized three different key values. These, three key values are optimally selected using OMBA. Here, at first we randomly initialize the key values.

Step 2: Oppositional solution generation:

After the initial solution generation, we create an opposite solution using equation (10). Here, every solution S_{ij} has a unique opposite S'_{ij} solution. The opposite solution $OP(S'_{11}, S'_{12}, ..., S'_{1n})$ is calculated based on the equation;

$$S'_{ij} = L_i + U_i - S_{ij} \quad (10)$$

Where; the lower bound coefficient is denoted by L_i, the upper bound coefficient is denoted by U_i, the old solution is denoted by S_{ij}. Then, we combine the initial and opposite solution for further processing.

Step 3: Fitness calculation:

Fitness function of security process as minimum number data retrieved in decryption process with triple keys K 1, K2 and K 3. With help of optimal key, only the below condition satisfied and also initial solution generation process using multiple key sets in TDES process.

$$F_i = \max(T)$$
$$T = Key\ breaking\ time \quad (11)$$

Step 4: Updation using MBO algorithm

Utilizing MBO algorithm, we update the solution after the calculation of fitness. The Updation is based on two stages. If the data are belongs to land 1 means we update the solution using movement operator. Similarly, the data are present in land 2 means the solutions are updated using alter operator.

Step 5: Termination criteria:

The algorithm terminates its implementation only if a highest quantity of iterations is reached and the solution that is having the best fitness value is selected by utilizing OMOA and it is provided as a best solution to TDES.

3.3.2 OTDES based document encrypion

The main objective of this section is to encrypt the de-duplictaed data using OTDES algorithm. In this the optimal keys are taken from OMBO algorithm. The encryption process is explained below.

- Here, at first we encrypt the document using DES algorithm with the help of first key K^1 which is taken from OMBO
- Then, the encrypted document is decrypted using DES algorithm with the help of second key K^2 which is derived from OMBO.
- Finally, we encrypt the decrypted document DES Algorithm with the help of the third key K^3.

$$S^{doc} = P3(O1(E1(d))) = P3(u) \quad (9)$$

$$S^{doc} = P3(O3(P1(d))) = P1(u) \quad (10)$$

Where, S^{doc} represents the cipher document, O1 represents decryption with K^1, P^1 defines encryption with K^1 and u refer the Secure Data. 3DES cipher can be used with a secret 112-bit key. Third and first secret keys are same in this case.

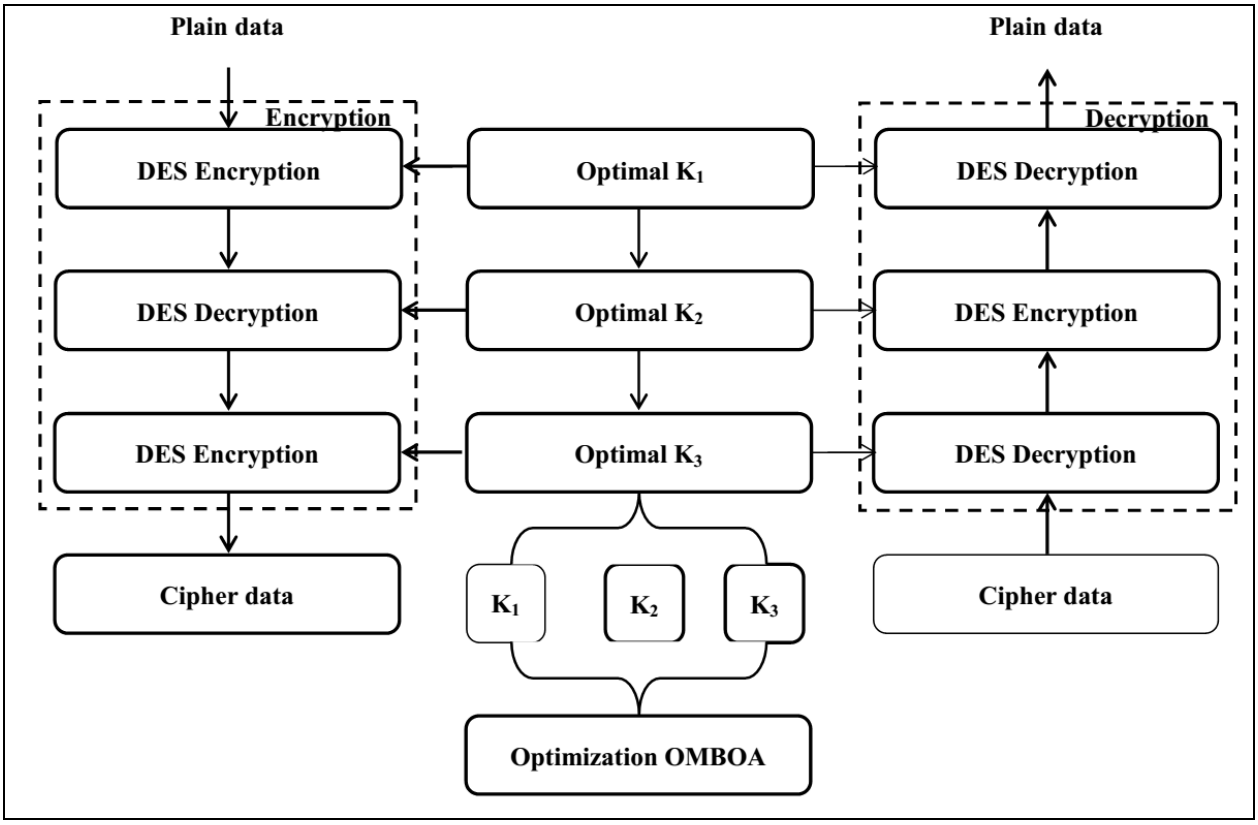

Figure 2: Optimal TDES algorithm

3.4 User interface module

After the encryption process, we store encrypted document on cloud database via user interface module. Then, any cloud user wants to access the document; they at first send the request to user interface module. The user is already a member means, the Cloud send the data to the user otherwise ignore the request. The confidentiality of information can be inferred directly because of the optimal encryption scheme. The benefit of the proposed strategy is the other attackers cannot produce a substantial signature or legitimate message authentication.

IV. RESULT AND DISCUSSION

This proposed data security with optimal encryption along with data de-duplication model is executed in java programming with JDK 1.7.0 out of a windows machine containing configurations Intel (R) Core i5 processor, 1.6 GHz, 4 GB RAM and the operation framework stage. The proposed methodology is evaluated with different metrics. We analyzed the experimental results of presented methodology in this section.

4.1 Evaluation metrics

Following metrics have been used for evaluation:

Total Time consumption: - This is the total time consumed by each method on their respective iterations.

Encryption time: - Time consumed by the system for encrypting the data from the dataset.

Files uploaded: - This is the total number of files uploaded per unit time. The files are uploaded on various iterations.

MIM attack: - MIM attack here represents the percentage of files that can be attained by the MIM attack from the considered dataset.

4.2 Performance analysis of proposed methodology

The aim of proposed methodology is to securely store the document in to the cloud and retrieve the data without any information loss. In this paper we analyze security risk in hybrid cloud system. Here, we try to reduce the two types of risk using de-duplication algorithm and encryption algorithm. Here, at first we collect the document from different user and stored the data into hybrid cloud. Then we remove the repeated document in cloud database using score matching mechanism. After that, we encrypt the document using OTDES algorithm. Then the encrypted data are stored in the cloud. Then in retrieval process, the cloud user get the information through user interface module and other user are not allowed to retrieve the data. In this section, efficiency of the system is analyzed in terms of different evaluation metrics. We compare the presented work with three different methods such as presented by Praveena and Rangarajan (2018), GA and PSO to prove the efficiency of the proposed methodology. Praveena and Rangarajan (2018), have encrypted the file using RSA algorithm and they developed a re-duplication algorithm and novel access control mechanism.

Even though, this method does not provide high level of data security because of the limitations of RSA algorithm. Also RSA algorithm require key of at least 1024 bit for good security in encryption and decryption. But in our paper, for encryption process we utilize only 64 bit key. And further increase the security of the system we optimally select the key for encryption.

Table 3: Time consumed with varying number of files

Number of files	Methods			
	Proposed	Proposal by Praveena & Rangarajan (2018)	GA	PSO
25	5412	6215	6138	5941
50	7965	8625	8103	8019
75	12364	13548	13261	13224
100	15695	17224	16844	16927

Figure 3 shows the proposed performance analysis based on consumed time of number of file 25, 50, 75 and 100 for the proposed method in comparison to the existing methods. Overall, the proposed method performs much better than Existing algorithms in all the iterations.

Table 4 depicts the analysis of performance depending on encryption time of number of file 25, 50, 75 and 100 for the proposed method vs. Existing vs. GA vs. PSO.

Table 4: Encryption Time with varying number of files

Number of files	Methods			
	Proposed	Proposal by Praveena & Rangarajan (2018)	GA	PSO
25	2145	1968	1845	1893
50	3547	2974	2863	2901
75	6587	6124	6022	6094
100	8967	8459	8149	8265

Table 4 depicts the analysis of the performance depending on encryption time for the proposed method, Existing, GA and PSO in four different iterations. It is clearly seen that the proposed method achieves the more efficient result when compared to other methods.

Table 5 shows the examination of performance of the presented method with the existing methods using MIM attack by varying iteration. The proposed method is compared with three different methods here such as existing, GA and PSO. At present lot of attacks namely denial of service (DOS), brute force attack (BFA) and Man in the middle (MIM) are used for get the original information from secure document. In this paper, we apply MIM attack on secure document to check the confidentiality. It is clear from the result that the presented method achieves the more efficient result when compared to other methods.

Table 5: MIM attack resolution time with varying number of files

Number of files	Methods			
	Proposed	Proposal by Praveena & Rangarajan (2018)	GA	PSO
25	7.68	11.54	12.13	12.84
50	6.15	12.24	13.64	14.03
75	5.24	9.63	10.32	11.28
100	7.63	14.58	15.21	16.49

Also, for the encryption process, we have utilised the OTDES algorithm. Here, for key generation process we use oppositional monarch butterfly algorithm. To prove the effectiveness of proposed OTDES algorithm based encryption process, we compare our proposed algorithm with DES, ECC, and AES algorithm. The encryption process comparison has been given in table 6. The methods involve OTDES, Data encryption standard (DES), Elliptic Curve Cryptography (ECC) and advance encryption standard (AES). The AES algorithm is mainly used for encryption process which is hide the original information from malicious. This algorithm was very simple structure. But the counter mode is complex. The ECC algorithm was also used for encryption process. This algorithm was faster and it need less computing power. But this algorithm more complex and more difficult compare to AES, ECC and DES algorithm. Similarly, in DES algorithm the initial and final permutation is not exactly clear and seems confusing. To overcome the problems in this paper, we proposed a OTDES algorithm which is highly secure the data compare to other methods. These methods are analyzed under iterations 25, 50, 75 and 100. The comparison shows that OBDES performs well in terms of cryptography. Therefore OBDES performs better than DES, ECC and AES.

Table 6: Comparison of Different Cryptographic Methods

Iterations	Cryptographic Methods			
	OTDES	DES	ECC	AES
25	7.68	9.48	11.94	11.63
50	6.15	9.12	11.86	10.25
75	5.24	8.96	10.49	10.52
100	7.63	10.31	12.22	11.54

The fitness function calculation of various algorithms such as GA, PSO and OMB are shown in figure 8. The fitness function has been calculated using these algorithms on various iterations. When the iteration is set at 10, time takes break the key using GA is 6841ms, the time takes break the key using PSO is 5864ms and the time takes break the key using proposed OMB 7682. When the iteration is set at 30, the time takes break the key using GA is 10037ms, the time takes break the key using PSO is 9142ms and the time takes break the key using OMB is 12867ms. These iterations show that the fitness function is higher for OMB when compared with GA and PSO. This is same for all the iterations. Therefore OMB provides efficient performance on fitness function calculation.

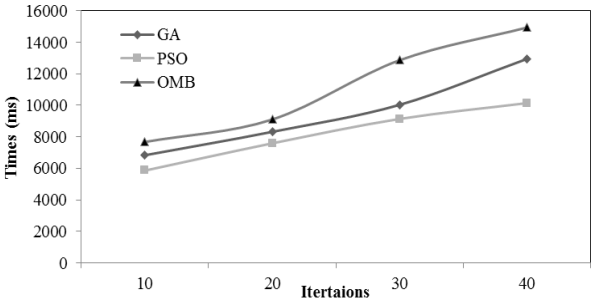

Figure 3: Fitness function of various algorithms for various iterations

V. CONCLUSION

Oppositional monarch butterfly algorithm with TDES has been presented for hybrid cloud security during information storage and retrieval. The work has focused upon the de-duplicated data and data encryption, to increase data security on cloud. The encryption process was done with the help of TDES algorithm and the key generation process was done with the help of OMBA. For experimental analysis, the hybrid cloud networks' security level when saving and recovering data has been analyzed. This method mainly decreases the security hazards and for limiting the time and spacebased data access. The algorithm has been evaluated for the metrics using the total time consumed, encryption time, number of files uploaded and MIM attack resolution time. The developed algorithm has been found good in comparison to the existing algorithms.

REFERENCES

1. Abdi, S., PourKarimi, L., Ahmadi, M., & Zargari, F. (2017). Cost minimization for deadline-constrained bag-of-tasks applications in federated hybrid clouds. Future Generation Computer Systems, 71, 113-128. doi: 10.1016/j.future.2017.01.036
2. Arumugham, S., Rajagopalan, S., Rayappan, J., & Amirtharajan, R. (2018). Networked medical data sharing on secure medium – A web publishing mode for DICOM viewer with three layer authentication. Journal Of Biomedical Informatics, 86, 90-105. doi: 10.1016/j.jbi.2018.08.010
3. Esposito, C., Castiglione, A., & Choo, K. (2016). Encryption-Based Solution for Data Sovereignty in Federated Clouds. IEEE Cloud Computing, 3(1), 12-17. doi: 10.1109/mcc.2016.18
4. Garg, P., Sharma, M., Agrawal, S., & Kumar, Y. (2018). Security on Cloud Computing Using Split Algorithm Along with Cryptography and Steganography. International Conference On Innovative Computing And Communications, 71-79. doi: 10.1007/978-981-13-2324-9_8
5. Gordon, A. (2016). The Hybrid Cloud Security Professional. IEEE Cloud Computing, 3(1), 82-86. doi: 10.1109/mcc.2016.21
6. Hwang, Jinho. "Toward Beneficial Transformation of Enterprise Workloads to Hybrid Clouds." IEEE Transactions on Network and Service Management, Vol. 13, No. 2, pp. 295-307, 2016.
7. Keshanchi, B., Souri, A., & Navimipour, N. (2017). An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: Formal verification, simulation, and statistical testing. Journal Of Systems And Software, 124, 1-21. doi: 10.1016/j.jss.2016.07.006
8. Khelifi, F., Brahimi, T., Han, J., & Li, X. (2018). Secure and privacy-preserving data sharing in the cloud based on lossless image coding. Signal Processing, 148, 91-101. doi: 10.1016/j.sigpro.2018.02.016
9. Kumar, V., & Pradhan, P. (2013). Role of Service Level Agreements in SaaS Business Scenario. IUP Journal of Information Technology, 9(1).
10. Labba, C., Bellamine Ben Saoud, N., & Dugdale, J. (2018). A predictive approach for the efficient distribution of agent-based systems on a hybrid-cloud. Future Generation Computer Systems, 86, 750-764. doi: 10.1016/j.future.2017.10.053
11. Lin, W., Wang, W., Wu, W., Pang, X., Liu, B., & Zhang, Y. (2018). A heuristic task scheduling algorithm based on server power efficiency model in cloud environments. Sustainable Computing: Informatics And Systems, 20, 56-65. doi: 10.1016/j.suscom.2017.10.007
12. Lu, P., Sun, Q., Wu, K., & Zhu, Z. (2015). Distributed Online Hybrid Cloud Management for Profit-Driven Multimedia Cloud Computing. IEEE Transactions On Multimedia, 17(8), 1297-1308. doi: 10.1109/tmm.2015.2441004
13. Mao, L., Li, Y., Peng, G., Xu, X., & Lin, W. (2018). A multi-resource task scheduling algorithm for energy-performance trade-offs in green clouds. Sustainable Computing: Informatics And Systems, 19, 233-241. doi: 10.1016/j.suscom.2018.05.003
14. Nikravan, M., Movaghar, A., & Hosseinzadeh, M. (2018). A lightweight signcryption scheme for defense against fragment duplication attack in the 6LoWPAN networks. Peer-To-Peer Networking And Applications, 12(1), 209-226. doi: 10.1007/s12083-018-0659-8
15. Prabu Kanna, G., & Vasudevan, V. (2018). A fully homomorphic–elliptic curve cryptography based encryption algorithm for ensuring the privacy preservation of the cloud data. Cluster Computing. doi: 10.1007/s10586-018-2723-9
16. Praveena, D., & Rangarajan, P. (2018). A machine learning application for reducing the security risks in hybrid cloud networks. Multimedia Tools And Applications. doi: 10.1007/s11042-018-6339-0
17. Singh, A., & Chatterjee, K. (2017). Cloud security issues and challenges: A survey. Journal Of Network And Computer Applications, 79, 88-115. doi: 10.1016/j.jnca.2016.11.027
18. Singh, J., & Kumar, V. (2013). Implementation of User-End Broker Policy to Improve the Reliability of Cloud Services. International Journal of Cloud Applications and Computing (IJCAC), 3(4), 13-27.
19. Singh, J., & Kumar, V. (2014). Multi-disciplinary research issues in cloud computing. Journal of Information Technology Research (JITR), 7(3), 32-53.
20. Sturrus, E., & Kulikova, O. (2014). Orchestrating Hybrid Cloud Deployment: An Overview. Computer, 47(6), 85-87. doi: 10.1109/mc.2014.159 13