Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

A SECURE METHOD FOR MANAGING DATA IN CLOUD STORAGE USING DEDUPLICATION AND ENHANCED FUZZY BASED INTRUSION DETECTION FRAMEWORK

Hema S, Ph.D Research Scholar (Part Time), PG & Research Department of Computer Science, Govt. Arts College (Autonomous), Salem-7, TamilNadu, India

Dr.Kangaiammal A, Assistant Professor, Department of Computer Applications, Govt. Arts College (Autonomous), Salem-7. TamilNadu, India (Affiliated to Periyar University, Salem, Tami Nadu)

Abstract. Cloud services increase data availability so as to offer flawless service to the client. Because of increasing data availability, more redundancies and more memory space are required to store such data. Cloud computing requires essential storage and efficient protection for all types of data. With the amount of data produced seeing an exponential increase with time, storing the replicated data contents is inevitable. Hence, using storage optimization approaches becomes an important pre-requisite for enormous storage domains like cloud storage. Data deduplication is the technique which compresses the data by eliminating the replicated copies of similar data and it is widely utilized in cloud storage to conserve bandwidth and minimize the storage space. Despite the data deduplication eliminates data redundancy and data replication, it likewise presents significant data privacy and security problems for the end-user. Considering this, in this work, a novel security-based deduplication model is proposed to reduce a hash value of a given file size and provide additional security for cloud storage. In proposed method the hash value of a given file is reduced employing Distributed Storage Hash Algorithm (DSHA) and to provide security the file is encrypted by using an Improved Blowfish Encryption Algorithm (IBEA). This framework also proposes the enhanced fuzzy based intrusion detection system (EFIDS) by defining rules for the major attacks, thereby alert the system automatically. Finally the combination of data exclusion and security encryption technique allows cloud users to effectively manage their cloud storage by avoiding repeated data encroachment. It also saves bandwidth and alerts the system from attackers. The results of experiments reveal that the discussed algorithm yields improved throughput and bytes saved per second in comparison with other chunking algorithms.

Keywords: Cloud computing, De-duplication, Cloud Storage, Distributed Storage Hash Algorithm (DSHA), Improved Blowfish Encryption Algorithm (IBEA), enhanced fuzzy based intrusion detection system (EFIDS), Data Security

1. Introduction

Cloud computing is one of the developing technology, and it has helped several organizations to save money and time adding convenience to the end users [1-3]. With the increasing demand of data rates throughout the world, cloud storage systems are becoming a necessity for computer users. All the users are uploading their data onto the cloud storage and are accessing the remotely stored data upon need. But 70% of the data that is stored in cloud storage is redundant, this is not permitting us to utilize the storage space efficiently [4-5]. After the rapid development of cloud computing, users and enterprise would like to back up their data to cloud storage. Due to the exponential rise in digital data, deduplication approaches are used extensively for data backup and reduction in storage and network overhead through the detection and removal of repetition among data. The deduplication techniques are generally used in the cloud server for reducing the space of the server [6-7]. When a data is uploaded its hash value is formed and then compared with the existing hash

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

value, if duplicate value is found then that data is not uploaded and is replaced with pointer to the unique data else if no duplicate is found.

De-duplication are often realized by either In-Line strategy [8-9] or Post-Process strategy. In in-line technique, this is a time consuming process as deduplication occurs prior to the data being written to the memory device. To prevent the unauthorized use of data accessing the encryption technique to encrypt the data before stored on cloud server. Cloud Storage generally includes business-oriented data and processes [10]; therefore, superior security is the sole solution for maintaining a serious trust association between the cloud users and cloud service providers. Therefore, the scope of cloud storage is vast because the organizations can virtually store their data's without bothering the entire mechanism [11]. Cloud Computing provides key advantage to the end users like cost savings, able to access the data irrespective of location, performance and security. In this technical work, a novel security based deduplication framework is proposed to reduce the given file size and provide more security for cloud storage, which aiming to boost the storage efficacy and increasing security.

The remaining sections of the work is arranged as given; Section 2 describes the advantages and disadvantages of the recent deduplication techniques. Section 3 presents the projected deduplication techniques for cloud storage. Section 4 illustrates the results and analysis. Section 5 deals with the conclusion and work intended for the future.

2. Literature Review

In this section review few recent techniques in cloud deduplication for cloud storage.

Li et al [12] presented a variety of novel deduplication frameworks that facilitates authenticated duplicate check in a hybrid cloud infrastructure. Security evaluation shows that the approach is safe in terms of the definitions provided in the discussed security framework. In the form of a proof of concept, a model of the discussed authenticated duplicate check approach is implemented and test experiments are carried out employing this model. Here it is proven that the discussed authenticated duplicate check approach experiences reduced overhead in comparison with normal functions. Kim et al [13] introduced a group-based memory deduplication approach, which guarantees separation between customer groups co-located in a hardware machine. Along with providing support for isolation, this approach facilitates customization of per-group of memory deduplication in accordance with the memory requirement and workload fashion of each group.

Deng et al [14] introduced a memory sharing technique that depends on user groups. This mechanism guarantees separation between the different users assigned to the same host. Moreover, a sampling hash algorithm that improves the efficiency of make the memory scanning process is designed. And implement this approach in Linux by modifying the KSM scanning mechanism and dividing the global ksmd thread into per-group ksmds. The experiment results show the work can help in optimizing the memory-critical VMs, and speed-up the memory scanning process with efficiency. Ning et al [15] introduced a new group-based memory deduplication approach to prevent the inter-group covert channel attack. VMs present in the same group can share memory with one another if and only if their owners are aware of a shared secret groupID value. The discussed approach is realized on KVM/KSM virtualized environment. More analysis reveals that group-based approach can yield inter-group separation with acceptable effect on the memory utilization.

Chen et al [16] proposed a group-based secure page sharing model (or GSKSM), in which both VM processes and normal processes are considered. And its implementation has been done successfully in Linux kernel 3.6.6, and the results of experiments reveal that it performs better

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

with the overhead reduced. At last, in accordance with GSKSM, an effective memory management approach (or SEMMA) is presented, which integrates GSKSM and balloon approach for efficient management of the memory, and the early experimental results are acceptable. Rong et al [17] developed an effective and robust protocol of CCCMD (Cloud Covert Channel based on Memory Deduplication). And, the CCCMD working approach is first analyzed in a virtualized environment, and its prominent drawbacks and implementation challenges are unravelled. And, a model called as Wind Talker that overcomes these hurdles are built. The experiments reveal that the performance of Wind Talker is quite improved with reduced bit error rate and yields a tolerable transmission speed, well adjusted to noisy environment.

3. Proposed Methodology

This section illustrates the comprehensive description of the projected data deduplication for cloud storage as well as improving the privacy and security. The overall flow of the projected scheme is shown in Fig 1. At first, the proposed Distributed Storage Hash Algorithm (DSHA) used for identifying and eliminating the duplicated data in cloud and to provide security the file data is encrypted by using an Improved Blowfish Encryption Algorithm (IBEA).This framework also proposes the enhanced fuzzy based intrusion detection system (EFIDS) by defining rules for the major attacks and alert the system automatically. Finally the data deduplication and security encryption technique allows the cloud users to manage their cloud storage space effectively by avoiding storage of repeated data's and save bandwidth.

3.1. Distributed Storage Hash Algorithm (DSHA)

In this section, the proposed Distributed Storage Hash Algorithm (DSHA) applied and in this method the entire file is considered as a chunk. We first generate a fixed size hash value applying the MD5 [18] or SHA-1 [19] method, and then the length of the hash value is reduced by using our proposed method. The single server monitors the performance of the complete system here as well as maintains the storage nodes. Multiple storage nodes manages the hash value index and deduplicated file data. The function of the DSHA algorithm is not just reducing the length of the hash value and also utilized for storing the hash value in the appropriate storage node. The benefit of file level deduplication is that it takes less resource and subsequently saves more memory space just as lessens metadata lookup process and CPU usage.

3.2. Improved Blowfish Encryption Algorithm (IBEA)

Security has mostly remained a huge challenge to handle the sensitive information in cloud storage. In order to get over the challenges of security violations, several cryptographic algorithms are utilized such as: AES, DES, Triple DES, Blowfish, etc. This research work proposed a novel blowfish encryption algorithm which is enhance the security degree of blowfish algorithm and generated the symmetric key block which is utilized for both encryption and decryption technique. This algorithm is secure against malicious attack and runs quicker compared to the well-known algorithms that are available.

1) Modification of proposed blowfish Algorithm using 4 states

In this a new improvement on the Blowfish algorithm makes use of the operation '#' applied during each round in the original Blowfish algorithm defined in [20]. Here a new key is introduced and applied on this '#'operation at both sides, this key may come in binary form and convert to a 4-states key. Two keys will be utilized in every round of the actual Blowfish, the first key K1will be utilized with the xL and Pi to generate the next left part. The second key K2 will be utilized with F (xL) and xR to generate the right part. These three inputs to the '#'

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

operation must be firstly changed from 32 bits to a 16 digits each may be one of four states (0, 1, 2, 3), i.e., each two bits converted to its equivalent decimal digits; see figure 1.

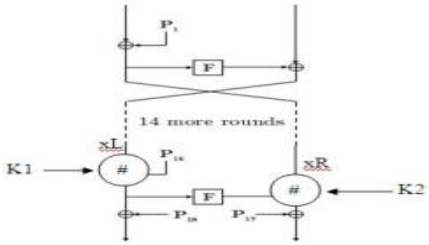


Fig.1. Inputs and Outputs of the # operation in proposed Algorithm

For example, the binary number: 10010111010100101010011111010001001 will be converted to the number: 2 1 1 3 1 1 0 2 2 2 1 3 2 2 0 2 1

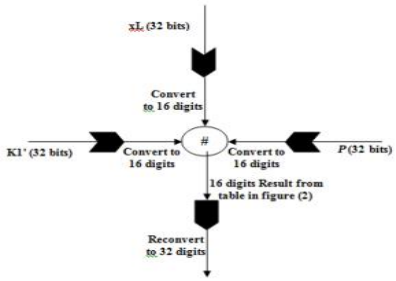


Fig. 2. New Structure of Each Round

Then the '#' operation will be applied to generate a new 16 digits that should be reconverted to 32 bits, see Figure 2.This work gives better encryption rate, security, adaptability as well as same key utilized for both process of encryption and decryption. Whenever the client download the information, the key is fundamental to coordinate for encryption key. On the off chance that the key is unmatched the client can't download the information. Test results reveal that the discussed blowfish algorithm works better in terms of security by lessening the probabilities of the presence of duplicate block which will leak information

The security of a cryptographic system frequently relies on the robustness of its secret key. Generally, each application has a special behaviour either in term of cache access or execution time. This cryptographic algorithm can be attacked more flexibly when target a cryptographic cipher, side-channel attacks threaten the security of this system by allowing the encryption key to be inferred.

Meanwhile, the Cloud Service Provider (CSP) is "honest-but-curious", implying that it is accurate in carrying out every operation, however it may attempt finding the actual content of uncommon data. And thesituations where the CSP acts good is not considered. As the objective of the intrusive CSP is to find the content of uncommon blocks, it has to extensively analyzed if (and how) confidentiality is ensured for uncommon data in every stage of the protocol. But, in case the user requires a file to be keptprivate even when its popularity increases, he may perform the encryption the file using a standard encryption protocol and upload it to the cloud without any protocol steps being followed. So this work concentrate on increasing security against those side channel attack with the help of Enhanced fuzzy based intrusion detection system (EFIDS). EFIDS integrates knowledge-based and behaviour-based approaches and

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

monitors to identify local events that could represent security violations. Due to the unique functionality of the EFIDS, intrusion detection elements are invisible and inaccessible to intruders.

2) Enhanced Fuzzy Based Intrusion Detection System (EFIDS)

Intrusion detection can be defined as "the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, which attempts to protect confidentiality, integrity, availability, or even bypassing the security mechanisms of a computer or network". Intrusion Detection System (IDS) can be grounded on software or hardware, which is tactically positioned at suitable detection points in a system or networks and helps in automatic detection of probable launch of attacks and prevent them from making any attacks in the future. With the aim of increasing the cloud system security, an effective Enhanced Fuzzy based IDS is proposed. The primary aim of this discussed cloud deduplication technique is to develop EFIDS for achieving security in the cloud system. For achieving the security in the system, in this work, an algorithm grounded on Fuzzy based IDS is developed. In this, along with the projected Fuzzy Logic System (FLS), the rules are optimally chosen with the aid of Ant Colony Optimization (ACO) algorithm.

- **Introduction to Fuzzy Logic Systems (FLS)**

An FLS introduces two important elements: 1) the Knowledge Base (KB), denoting the knowledge on the problem thatis getting solved taking the form of fuzzy linguistic IF-THEN rules, and 2) the Inference Engine, which brings in the fuzzy inference process required for getting an output out of the FRBS once an input is provided. The structure of a FLS . The KB comprises of the Rule Base (RB), which includes the set of linguistic rules that are conjoined using the connective also, and the Data Base (DB), constituting the term sets and the membership functions specifying their semantics. The structure of the fuzzy linguistic rule used in FLS is given by the following:

$$R_i: IF\ X_1 is\ A_{i1}\ and\ \ldots.\ and\ X_n\ is\ A_{in}\ THEN\ Y\ is\ B_j, \qquad (1)$$

with $X_1, \ldots\ldots X_n$ and Y referring to the input and output linguistic variables, correspondingly, and $A_{i1}, \ldots\ldots, A_{in}$ and $B_j$referring to the linguistic labels, with all of them showing an association o a fuzzy set specifying what it means.

- The Inference Engine consists of three components: a Fuzzification Interface, which exhibits the function of converting the crisp input data into fuzzy sets, an Inference System, which combines these together along with the KB to carry out the fuzzy inference process, and a Defuzzification Interface, which gets the crisp output finally from every inferred fuzzy outputs.

The Inference System depends on the application of the Generalized Modus Ponens, which extends the conventional logic Modus Ponens. It is carried out using the Compositional Rule of Inference, which staying in its simplest structure gets reduced to:

$$R_i(x_0, y) = \mu B_i'(y) = I\left(\mu A_i(x_0), \mu B_j(y)\right), \qquad (2)$$

with $x_0 = (x_1, \ldots, x_n)$referring to the present system input, $\mu A_i(x_0) = T(\mu A_{i1}(x_1), \ldots, \mu A_{in}(x_n))$indicating the matching degree between the rule antecedent and the input with $\mu A_{ik}(.)$referring to the membership function of the label A_{ik} and T indicating a conjunctive operator (a t-norm), and I referring to a fuzzy implication operator.

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

- **The Fuzzy Rule Learning (FRL) Problem**

All these methods depend on operating over an input-output data E={e1, . . . , eN},el= (xl1, . . . , xln, yl), indicating the behavior of the problem that is getting solved, and with an earlier specification of the DB comprising of the input and output primary fuzzy splits. In this case, symmetrical fuzzy partitions will be considered with a number of triangular membership functions that cross at a height of 0.5. Therefore, our FRL problem will be restricted to get the rules which combine the labels belonging to the antecedents and assigning a particular consequent to each antecedent combination.

- **Ant Colony Optimization Algorithms for Learning Fuzzy Rules**

To use ACO algorithms to a specific problem, the steps below must be followed:

1. Represent the problem in the form of a graph or an identical structure simply covered by ants.
2. Specify the means of designating a heuristic preference to every selection to be taken by the ant in every step for the solution generation.
3. Define a suitable means for the pheromone initialization.
4. Specify a fitness function requiring optimization.
5. Select an ACO algorithm and apply it to the problem.

In the following subsections, these steps will be introduced to solve the FRL problem.

- **Representation of the Problem**

In order to use ACO algorithms for the FRL problem, it would be better to consider it to be a combinatorial optimization problem capable of being denoted in the form of a graph. Like this, the problem considers a specific number of rules and interprets the FRL problem to be the means of designating consequents (i.e., labels belonging to the output fuzzy partition) to these rules according to an optimality condition. Therefore, in reality, working with an assignment problem and the representation of problem can be identical to the one utilized for finding a solution to the quadratic assignment problem (QAP), though with few differences. An analogy is drawn between rules and facilities and also between consequents and locations. But, in contrary the QAP, the set of probable consequents for every rule may be unique and there are chances to designate a consequent to multiple rules (two rules may be assigned to the same consequent). And deduce from these properties that the sequence of choosing each rule to be designated with a consequent is indeterminate, i.e., the order of assignment has no relevance. To construct the graph, the following steps are taken:

1. Determine the rules: A rule $R_i - i = 1, \ldots . , N_r$ defined by an antecedent combination,

$$R_i = IF\ X_1 is\ A_{i1}\ and\ \ldots . and\ X_n\ is\ A_{in}\ , \qquad (3)$$

will take part in the graph if and only if:

$$\exists_{el} = \left(x_1^l, \ldots \ldots \ldots, x_n^l, y^l\right) \in Esuch\ that\ \mu A_{i1}\left(x_1^l\right) \ldots . . \mu A_{in}\left(x_n^l\right) \neq 0. \qquad (4)$$

It implies that is, there exists at least one example positioned in the fuzzy input subspace defined by the antecedents considered in the rule.

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

2. Associate the rules to consequents: The rule R_i will be associated with the consequent $B_j - j = 1, \ldots \ldots, N_c$ (taken from the set of labels of the output fuzzy partition) if and only if it satisfies the condition below:

$$\exists_{el} = \left(x_1^l, \ldots \ldots \ldots, x_n^l, y^l\right) \in Esuch\ that\ \mu A_{i1}\left(x_1^l\right) \ldots . . \mu A_{in}\left(x_n^l\right). \mu B_j\left(y^l\right) \neq 0. \quad (5)$$

This implies that, at least one example exists in the fuzzy input subspace, which is enclosed by a consequent such as this. Figure 3 shows an example of a system with four rules and one output variable with three consequents. In Figure 3(a), the probable consequents for every antecedent combination are illustrated. To build a full solution, an ant iterates over every rule and selects a consequent having a probability based on the pheromone trail τ_{ij} and the socratic information η_{ij} , as earlier(see Figure 3(b)). Like it is mentioned, the sequence of choosing the rules has no relevance. In Figure 3(c) the possible paths that an ant can take in a specific example.

- **Socratic Information**

The socratic information on the probable decision over the selection of a particular fic consequent, B_j, in every antecedent combination (rule) is decided by considering covering criteria as follows (see Figure 8 for a graphical interpretation of the socratic assignment):

For each rule defined by an antecedent combination, $R_i = IF\ X_1 is\ A_{i1}\ and \ldots . and\ X_n\ is\ A_{in} - i = 1, \ldots . . , N_r\ do$:

1. Construct the set E_i'which includes the input-output data pairs positioned in the input subspace defined by R_i, i.e., $E_i' = \{e_l = (x_1^l, \ldots . . , x_n^l, y^l) \in E$ such that $\mu A_{i1}\left(x_1^l\right) \ldots . . \mu A_{in}\left(x_n^l\right) \neq 0\}$.

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

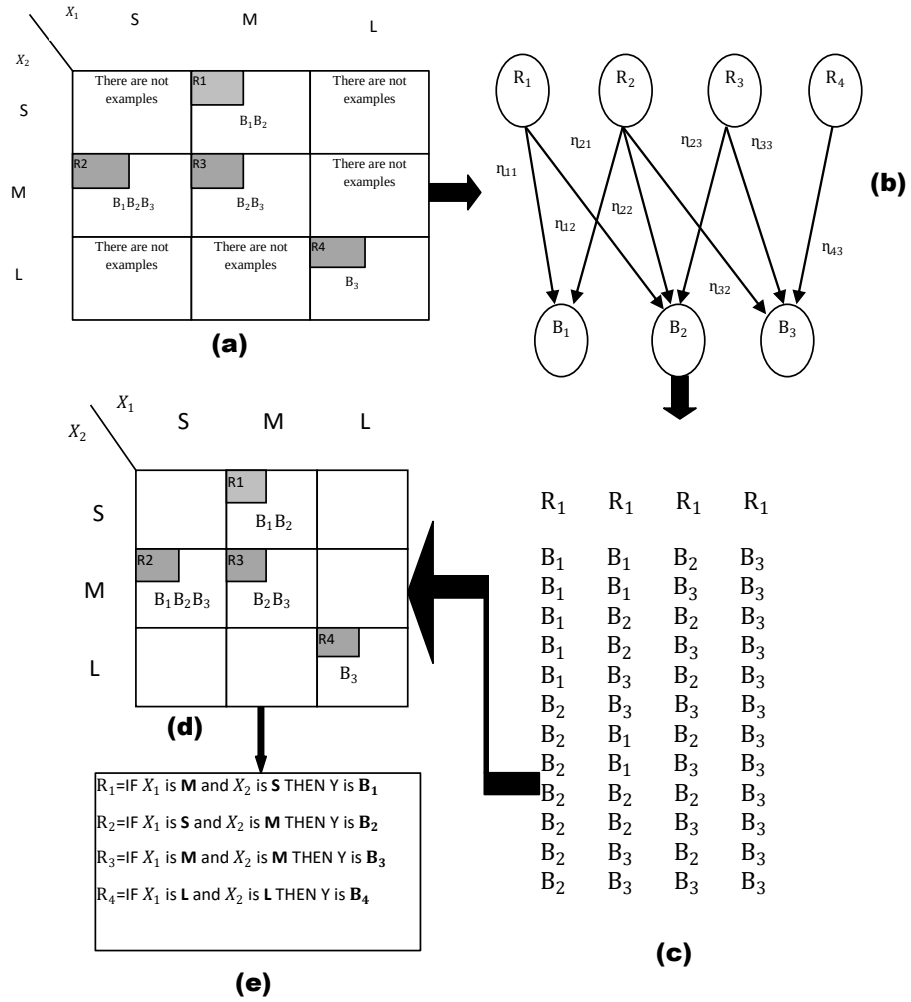


Figure 3. Learning process for an easy problem having two input variables (n = 2), four rules (Nr = 4), and three labels present in the fuzzy partition output (Nc=3): (a) Set of possible consequent for each rule (only the rules where at least one example is positioned in the corresponding subspace are considered); (b) Graph of paths where ηij 6= 0 except η13, η31, η41, and η42, which are zero; (c) It is possible to take twelve diverse paths (ensemble of consequents); (d) Rule decision table for the third ensemble; (e) RB produced from the third ensemble.

And an initialization function is used which is grounded on covering condition to provide a socratic degree of preference to every selection. Several diverse choices may be taken into consideration. In this work considering the covering of the example best covered criterion. Since the socratic information is based on covering criteria, it will be zero for a specific consequent when no examples positioned in the fuzzy input subspace are considered by it. This means that for a rule, only those links to consequents whose heuristic information is greater than zero will be considered. In Figure 3(b) can observe the consequent B3 cannot be assigned to the rule R1, the consequent B1 cannot be assigned to the rule R3, and the consequents B1 and B2 cannot be assigned to the rule R4 because their socratic information (covering degrees) are zero.

- **Pheromone Initialization**

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

The initial pheromone value of every assignment is computed as below:$\tau_0 = \frac{\sum_{i=1}^{N_r} max_{j=1}^{N_c} \eta_{ij}}{N_r}$. In this manner, the initial pheromone will be found from the average value of the path built using the best consequent in every rule as per the socratic information (implying a greedy assignment).

Socratic information considered:

$$\eta_{ij} = \underset{e_l \epsilon E_i'}{max} Min(\mu A_{i1}(x_1^l), \ldots\ldots, \mu A_{in}(x_n^l), \mu B_j(y^l)) \quad (6)$$

- **Fitness Function**

The fitness function defines the solution quality. The measure considered will be the function called mean square error (MSE), which defined as $MSE(RB_k) = \frac{1}{2.|E|}\sum_{e_l \epsilon E}(y^l - F_k(x_0^l))^2$with $F_k(x_0^l)$ being the output obtained from the FLS (built using the RB generated by the ant k, RB_k) when receiving the input x_0^l (input component of the example e_l), and y^lreferring to the known required output. If the measure is closer to zero, the better would be the solution.

- **Ant Colony Optimization (ACO) Algorithm**

In this work the FRL problem solved by using the ACO algorithm. The so-known solution construction and pheromone trail update rule considered by these ACO algorithms will be used. Only some adaptations will be needed to apply them to the FRL problem:

  - The set of nodes attainable from R_i (set of feasible neighbourhood of node R_i) will be $J_k(i) = \{j \; such \; that \; \eta_{ij} \neq 0\}$ in the transition rules considered by ACO algorithms when constructing the solution.
  - The amount of pheromone ant k puts on the couplings belonging to the solution constructed by it will be $1/MSE(RB_k)$, with RB_k being the RB generated by ant k.
  - In the local pheromone trail update rule of the ACO algorithm, the most usual way of calculating $\Delta\tau_{ij}$, $\Delta\tau_{ij} = \tau_0$, thus considering the simple-ACO algorithm.

- **Fuzzy system design**

Once the optimal rule is generated, align the fuzzy system. During the design of the fuzzy system, the fuzzy membership function (MF) specification and fuzzy rule base form the two primary steps. Membership function refers to the formula applied for computing the membership values given as follows. A MF stands for a curve assessing the way the mapping of every point in the input space is done onto a membership value (or degree of membership) in the range between 0 and 1. In addition, the MF alignment is done by selecting the right MF. In this, the triangular MF is chosen to modify the input data into the fuzzified value. The MF helps in complete evaluation of the fuzzy set.

1. A MF provides the measure of the similarity degree of an element with a fuzzy set.
2. MFs can be of any form, however there are few common examples depicted in actual applications.
3. The formula utilized for computing the membership values is explained as follows,

$$f(x) = \begin{Bmatrix} 0 & if\ x \leq a \\ \frac{x-a}{b-a} & if\ a \leq x \leq b \\ \frac{c-x}{c-b} & if\ b \leq x \leq c \\ 0 & if\ x \geq c \end{Bmatrix} \quad (7)$$

Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

- **Rule-based fuzzy score computation**

ACO optimization algorithm is used on the fuzzy rule set that is earlier generated. These rules are afforded to the fuzzy logic. The rule base includes a set of fuzzy rules taking the form of low, high, and medium distance values. At last, the score value is computed. Depending on the score value, it is verified if the provided data are perturbed or not. Here, in accordance to the score value, one threshold $T_h$is fixed. In case, the attained score value is higher than the threshold T_h, it implies that the data are perturbed; else the data can be considered normal. This way, the acquired score value meets the criterion expressed in equation (8),

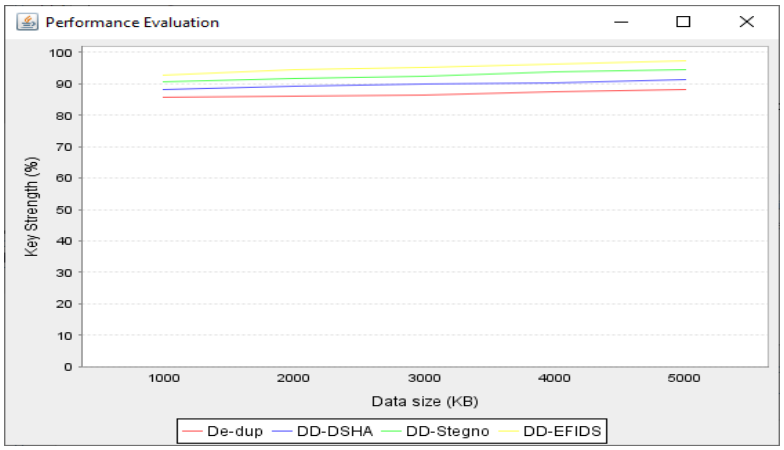
$$IDS = \begin{cases} T_h \geq score; & data\ are\ normal \\ T_h < score; & data\ are\ intruded \end{cases} \quad (8)$$

4. Results and Discussion

This section presents the results of proposed Enhanced Fuzzy Based Intrusion Detection System (EFIDS). To implement this system Java used in the front-end and MYSQL used in the backend. The performance evaluation has been carried out. Table 1 tabulates the test environment. SHA1 has been chosen for its collision tolerant characteristic for the hash value computation. Here, the results are analyzed and evaluated in terms of metrics such as Key strength and Attack detection accuracy.

Table 1. Test Setup

Hardware Setup	Processor - Intel(R)Core(TM)i5-5200U CPU @ 2.20GHz hard disk drive is 1TB Memory: 8 GB RAM
Software Setup	Operating System : 64-bit Windows 10 home version 1803 Programming Environment: Net Beans IDE 7.0.1, Java: 1.7.0; Database Server: 127.0.0.1 via TCP/IP, Software: MYSQL 5.5.27



Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 24-36
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.01

Figure.4. Performance comparison between the proposed and existing deduplication framework

Figure.4. shows the Performance comparison between the discussed and available deduplication framework. The x-axis denotes data size value and y-axis represents the key strength value. It conforms from the results of simulation that the projected DD-EFIDS technique provides the better results in key strength.

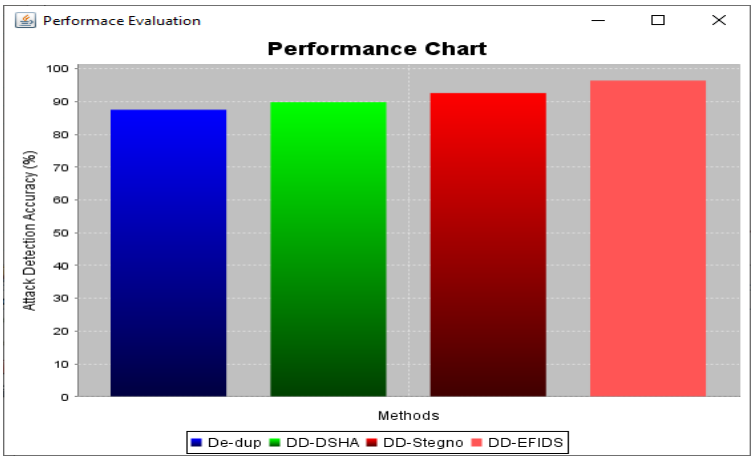


Figure.5. Performance comparison between the proposed and existing deduplication framework

Figure.5. shows the Performance comparison between the projected and available deduplication framework. To discover more attacks, DD-EFIDS incorporates knowledge-based and behaviour-based approaches and screens to distinguish nearby functions that that could represent security violations. In this criterion, the proposed system security compared to the existing technology. From the simulation results, DD-EFIDS issues more intensive number of attack warnings than other existing systems

5. Conclusion

In this proposed framework, system performance has been improved by using deduplication and provided a security framework to enhance cloud security. At first, the proposed Distributed Storage Hash Algorithm (DSHA) is employed for the identification and removal of replicated data in cloud. This research work exploits inline deduplication process, as its storage requirement is less compared to the post-process approach. The stored files are then encrypted by using Improved Blowfish Encryption Algorithm (IBEA). Also this framework proposes the enhanced fuzzy based intrusion detection system (EFIDS) by defining rules for the major attacks and alerts the system automatically. Here, separating the intrusion detection system from the system under monitoring makes the intrusion detection elements invisible and inaccessible to intruders. Because it uses a the combination of Improved Blowfish encryption and enhanced fuzzy based IDS , the proposed framework offers better deduplication and good security against the side channel attack, compared to existing chunk based deduplication. Currently, optimized cloud storage has been tested only for text files and pdf files. In future, it can be further extended to use files of other type i.e. video and audio files.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58.

Zhu, M., Zhang, K., & Tu, B. (2018, October). PCA: Page Correlation Aggregation for Memory Deduplication in Virtualized Environments. In *International Conference on Information and Communications Security* (pp. 566-583). Springer, Cham.

Deepu, S. R., Bhaskar, R., & Shylaja, B. S. (2014). Performance Comparison of Deduplication techniques for storage in Cloud computing Environment. *Asian Journal of Computer Science And Information Technology*, *4*(5), 42-46.

Akhila, K., Ganesh, A., & Sunitha, C. (2016). A study on deduplication techniques over encrypted data. *Procedia Computer Science*, *87*, 38-43.

Choudhary, B., & Dravid, A. (2014). A study on authorized deduplication techniques in cloud computing. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *3*(12), 4191-4194.

He, Q., Li, Z., & Zhang, X. (2010, October). Data deduplication techniques. In *2010 International Conference on Future Information Technology and Management Engineering* (Vol. 1, pp. 430-433). IEEE.

Bolosky WJ,Corbin S,Goebel D,Douceur JR.Single instance storage in Windows 2000.In:Proc.of the 4th Usenix Windows System Symp.Berkeley: USENIX Association,2000. 13-24.

Liu C, Lu Y, Shi C, Lu G, Du DH, Wang DS. ADMAD: Application-driven metadata aware de-duplication archival storage system. InStorage Network Architecture and Parallel I/Os, 2008. SNAPI'08. Fifth IEEE International Workshop on 2008 Sep 22 (pp. 29-35). IEEE.

Jibin Wang, Zhigang Zhao, Zhaogang Xu, Hu Zhang, Liang Li and Ying Guo, "I-sieve: An inline high performance deduplication system used in cloud storage," in Tsinghua Science and Technology, vol. 20, no. 1, pp. 17-27, Feb. 2015.

NetApp Deduplication and Compression. www.netapp.com/ us/products/platform-os/dedupe.html, April 2016.

Miao, M., Wang, J., Li, H., & Chen, X. (2015). Secure multi-server-aided data deduplication in cloud computing. *Pervasive and Mobile Computing*, *24*, 129-137.

Li, J., Li, Y. K., Chen, X., Lee, P. P., & Lou, W. (2014). A hybrid cloud approach for secure authorized deduplication. *IEEE Transactions on Parallel and Distributed Systems*, *26*(5), 1206-1216.

Kim, S., Kim, H., & Lee, J. (2011, August). Group-based memory deduplication for virtualized clouds. In *European Conference on Parallel Processing* (pp. 387-397). Springer, Berlin, Heidelberg.

Leesakul, W., Townend, P., & Xu, J. (2014, April). Dynamic data deduplication in cloud storage. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering* (pp. 320-325). IEEE.

Deng, Y., Hu, C., Wo, T., Li, B., & Cui, L. (2013, March). A memory deduplication approach based on group in virtualized environments. In *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering* (pp. 367-372). IEEE.

Chen, X., Chen, W., Long, P., Lu, Z., & Wang, Z. (2013, December). SEMMA: secure efficient memory management approach in virtual environment. In *2013 International Conference on Advanced Cloud and Big Data* (pp. 131-138). IEEE.

Zhu, M., Zhang, K., & Tu, B. (2018, October). PCA: Page Correlation Aggregation for Memory Deduplication in Virtualized Environments. In *International Conference on Information and Communications Security* (pp. 566-583). Springer, Cham.

Rong, H., Wang, H., Liu, J., Zhang, X., & Xian, M. (2015, August). Windtalker: An efficient and robust protocol of cloud covert channel based on memory deduplication. In *2015 IEEE Fifth International Conference on Big Data and Cloud Computing* (pp. 68-75). IEEE.

Opendedup, http://opendedup.org/

Nie, T., & Zhang, T. (2009, January). A study of DES and Blowfish encryption algorithm. In *Tencon 2009-2009 IEEE Region 10 Conference* (pp. 1-4). IEEE.