Computer Systems Science & Engineering
DOI:10.32604/csse.2022.020486
Article

Tech Science Press

Protecting Data Mobility in Cloud Networks Using Metadata Security

R. Punithavathi[1,*], M. Kowsigan[2], R. Shanthakumari[3], Miodrag Zivkovic[4], Nebojsa Bacanin[4] and Marko Sarac[4]

[1]Department of Information Technology, M.Kumarasamy College of Engineering, Karur, 639113, India
[2]Department of Computer Science and Engineering, School of Computing, SRM Institute of Technology, Kattankulathur Campus, Chennai, 603203, India
[3]Department of Information Technology, Kongu Engineering College, Erode, 638060, India
[4]Singidunum University, Belgrade, 11000, Serbia
*Corresponding Author: R. Punithavathi. Email: punitha.vathi@yahoo.com
Received: 26 May 2021; Accepted: 19 July 2021

Abstract: At present, health care applications, government services, and banking applications use big data with cloud storage to process and implement data. Data mobility in cloud environments uses protection protocols and algorithms to secure sensitive user data. Sometimes, data may have highly sensitive information, leading users to consider using big data and cloud processing regardless of whether they are secured are not. Threats to sensitive data in cloud systems produce high risks, and existing security methods do not provide enough security to sensitive user data in cloud and big data environments. At present, several security solutions support cloud systems. Some of them include Hadoop Distributed File System (HDFS) baseline Kerberos security, socket layer-based HDFS security, and hybrid security systems, which have time complexity in providing security interactions. Thus, mobile data security algorithms are necessary in cloud environments to avoid time risks in providing security. In our study, we propose a data mobility and security (DMoS) algorithm to provide security of data mobility in cloud environments. By analyzing metadata, data are classified as secured and open data based on their importance. Secured data are sensitive user data, whereas open data are open to the public. On the basis of data classification, secured data are applied to the DMoS algorithm to achieve high security in HDFS. The proposed approach is compared with the time complexity of three existing algorithms, and results are evaluated.

Keywords: Data mobility; data security; cloud computing; big data; DMoS algorithm

1 Introduction

Big data are processed in cloud storages using the Hadoop file system. However, providing security to big data in cloud databases is challenging. Content delivery networks in cloud environments are used by service providers and numerous content users, who are connected to the system. Thus, sensitive data in

the Internet must be protected from intruders. Researchers have presented and demanded for security algorithms for data protection. Content delivery networks include edge servers, caches, IoT devices, and end users. User data are placed in any edge servers or cloud storage by using Hadoop Distributed File System (HDFS) allocation. When users request data, they must not be identified as intruders but rather secured users. After a user has conformed, data must be served with high availability and performance with minimum time.

Big data and cloud storage are emerging technologies that are used in intelligence, data mining, and industrial data coordination, which are new topics for future business systems [1,2]. Meanwhile, cloud computing is a dynamic advanced model that allocates all Internet resources as a cloud pool and shares resources for different data processing applications. Compared with traditional distribution systems, the cloud system brings more elasticity, scalability, performance, and efficiency in data execution. Moreover, cloud computing reduces the execution cost of smart applications, supercomputing infrastructure, and grid environments. Despite the above advantages, security concerns on sensitive data when storing personal information remain as a challenge. Still now we have regular compliance on security in sensitive data processing from particular domain to wide distributed domain [3].

Big data platforms offer HDFS tools for computing big data by making decisions quickly and minimizing the risk of human intruders [4]. HDFS, which is commonly used in big data and cloud processing, involves splitting data into small files; this system supports parallel computation and has high reliability, reduced redundancy, and improved scalability in distributed systems [5]. In addition, HDFS is designed to process large datasets and data types (i.e., structured, unstructured, and semistructured). Scheduling algorithms in map reduce [6] used in big data clustering in the network. Various security problems are addressed by HDFS and cloud storage in big data environments. New security challenges are addressed in big data cloud environments [7–11] to distribute sensitive data, such as business secrets and personal data. The main objective of this project is to protect sensitive data from high risks.

The secure socket layer (SSL) between clients and storage devices uses nodes during cloud data transmission for security purposes. The encryption technique is used by creating hash functions for data transfer. However, a major problem in single-certificate authorization is that it can disturb cloud side security, that is, intruders can easily identify certificates. While transmitting data in the cloud system, data might cross different levels of security in network nodes and may provide sensitive information under great threats. For example, data may be hacked during data exchange. Thus, the security algorithm for data mobility must be identified in the cloud networks. Furthermore, performance must be ensured in data mobility networks. To increase the standards, techniques that guarantee safety must be introduced to reduce risks in data analysis and aggregation. In our study, we introduce a security algorithm, namely, data mobility security algorithm (DMoS), to overcome the above challenges in cloud networks. The proposed algorithm reduces threats during data transfer in cloud environments. Our work mainly focuses on boosting security and privacy in handling big data in the cloud.

Our proposed DMoS algorithm provides an integrated methodology for sensitive data by combining data classification and security. In this process, data are transferred and duplicated after implementing encryption and decryption algorithms on sensitive information.

The important contributions of this study are as follows:

1. A data classification technique for cloud networks is designed on the basis of data description and metadata to identify the sensitivity level of data. On the basis of data importance and how much confidential data must be preserved, a classification technique is built. This technique beats the challenges of handling big data in open-source platforms with a distributed HDFS platform. This platform uses the above classification technique to perform speedy operations in data classification.

2. In accordance with data security level (i.e., on the basis of a client's predefined policies), data in clouds are classified into public, private, or special data. In contrast to private data, public data do not need security. Meanwhile, special data are highly confidential and can only be sent to specific receivers. This methodology helps identify data and provide security to required data. In this manner, we can reduce the cost of security processes for all data files transmitted between nodes in the cloud network.
3. Here, we introduce an encryption technique for data that require security by considering two aspects: time and efficiency.
4. Security-based data classification software is designed to handle security issues and help cloud administrator systems in data mobility.
5. Security and classification are integrated into a single methodology in cloud systems.
6. Data are accomplished with safety, integrity, and high availability.

The remainder of this paper is organized as follows: The next section consists of a literature survey on big data and cloud computing in security analysis. Security risks are discussed in Section 3. Classification and security integration are explained in Section 4. Performance evaluation and results are provided in Section 5. Lastly, the conclusions of this study and plans for future work are presented in Section 6.

2 Literature Survey

With the emergence of big data technology, cloud security has become crucial, and architecture for big data with risks is assessed. Security mechanisms and data governance are missed to address in this section. Data security is difficult to monitor when data are transferred between service providers in the cloud, whereas enterprises are difficult to monitor because various data do not need the same security level. A major potential risk in cloud data protection comes in two types of attacks: insider and outsider. Insider attacks are more tedious than outsider ones. To overcome this type of attack, a hidden Markov model is used to detect whether the edge devices in the cloud are sensitive, hacked, legitimate, or under attack. In this manner, edge devices act as virtual devices for identifying hackers and attackers in the network, thus preventing further transmission by intruders. In security information, event management and data loss prevention are encryption tools for protecting sensitive data from attackers in cloud security networks. Indicators for identifying critical and sensitive data are not induced in big data classification; thus, providing security to big data processing is difficult. Fake mails are created by intruders to gain valuable data from clients. At present, no phishing security technique has been introduced to big data processing, although some integrity methods have been proposed.

The need for cyber security in cloud computing has increased [12,13] due to the increase in cloud storage use, which has become a target for intruders. Traditional security algorithms can handle databases, but these algorithms may not be suitable for big data processing environments. Therefore, providing mobile data security in cloud storage has become challenging. Sometimes, sensitive data are controlled through fog computing [14,15]. Big data transfer between clouds and fog requires checking the data type of storage devices. Moreover, this task needs a clustering algorithm for data sorting. Here, security algorithms are required to prevent outsiders from accessing data.

Recently, research has focused on data transmission security as it has become a major problem in recent technologies, such as cloud storage, data science [16,17], and big data. However, some of the existing security algorithms can only work on fixed databases, not on big data platforms. Thus, a highly secured algorithm for data in clouds and big data must be established [18,19]. Data that are under threat the most are confidential data in social media. In [20], the authors discussed that in big data infrastructure, the emails of users are under threat. Nowadays, providing security for cloud storage and data transfer in big

data-based cloud computing systems is challenging [21,22]. Big data storage and transmission management require high data security and privacy [23]. Comprehensive security algorithms must meet secrecy in data with integrity and reliability. In [24], multiple encryption techniques were used for data privacy in the cloud to construct big data infrastructure. This technique protects data from unofficial users while storing and processing data [25]. Here, data are treated with the same priority and time complexities for improved performance.

Cloud and big data security has two interdependent aspects, namely, security and data process control. In cloud computing, the main problems are the smart management and classification of big data. In various levels, data security is ensured by using the Kerberos policy, which ensures security in data transmission, data communications, and stored data authorization. The transport secure layer is designed by Kerberos for data encryption. In the cloud, data may be received from different sources with various governance policies, making the security process difficult. Functions, such as "know data," "prevent unknown," "detect intruders," "respond to client," and "recover lost data," are practiced in big data security to find leaked data. Files in cloud data are processed using an access control mechanism. The logging method is used in access control where user overlapping occurs. In data, own attributes are utilized for ranking based on its relevance. Empirical data are used to weigh the various levels of sensitive data. Data security is managed by using empirical information in cloud computing.

3 Security Risk Assessment in Cloud and Big Data

At present, information in social media is processed using big data. Thus, the main goal is to protect the sensitive data of users. Big data analytics proves how difficult it is to provide security for NP-hard problems. The cloud infrastructure must provide data integrity, availability, and data secrecy. The main problem in security is identifying authorized users in the network while transmitting data in the cloud network. Only authorized users are allowed to enter and modify data in the network. In this manner, the sensitivity of data is protected. Data availability means that data are available to authorized users without interruption and with easy access. Another problem is that storing big data in one place leads to high security risks. Most institutions have a single server for processing, storing, and fetching the sensitive data of their customers. Sensitive data may include patient's record, trading details, and financial details in the same server; this setup may cause information leakage and data hacking. During service invocation, intruder attacks lead to denial-of-service and availability problems. To overcome these problems, some researchers use classification techniques based on risk assessment to avoid security risks in cloud-based big data processing. Some risks in asset management, vulnerability prediction, threat level classification, and calculation of the likelihood of threat are handled. Measurement is performed between zero and five levels: negligible threat (zero to one), low-risk threat (one to two), medium-level threat (two to three), high risk (three to four), and high threats (four to five).

The equation for data threat and data vulnerability is

$$\text{DTDV} = \text{threat value} + \text{vulnerable value}. \tag{1}$$

Impact level of risk (ILR) is introduced to control security level in big data protection. The value of ILR is calculated on the basis of risk assessment as critical data protection needed or public data does not require protection. Finally, ILR is product of asset, DTDV, likelihood data threat value as LDTR.

$$\text{ILR} = \text{asset} * \text{DTDV} * \text{LDTR} \tag{2}$$

On the basis of Eq. (2), ILR is calculated as

If data security has high risk, then the value is (4–5) and calculated as ILR= (very high) * (very high) * (very high) = (4–5) * (4–5) * (4–5).

Data security has low risk, then value is (0–1), ILR is calculated as ILR= (very low) * (very low) * (very low) = (0–1) * (0–1) * (0–1).

ILR > 0–5, then it is special data, which need additional protection. These data are only sent to one secured user through the encryption technique.

Based on combinations of data component risk values, the value of ILR ranges from zero to five. This value helps the proposed approach to classify the level of security needed by the data during the processing, transferring, and copying of data over the cloud data.

4 Proposed Combined Methodology of Security and Classification

Here, we introduce a combined method of classification and security-based framework for providing security during data mobility over the cloud network. The architecture of the method is illustrated in Fig. 1, and the entire process is discussed below.

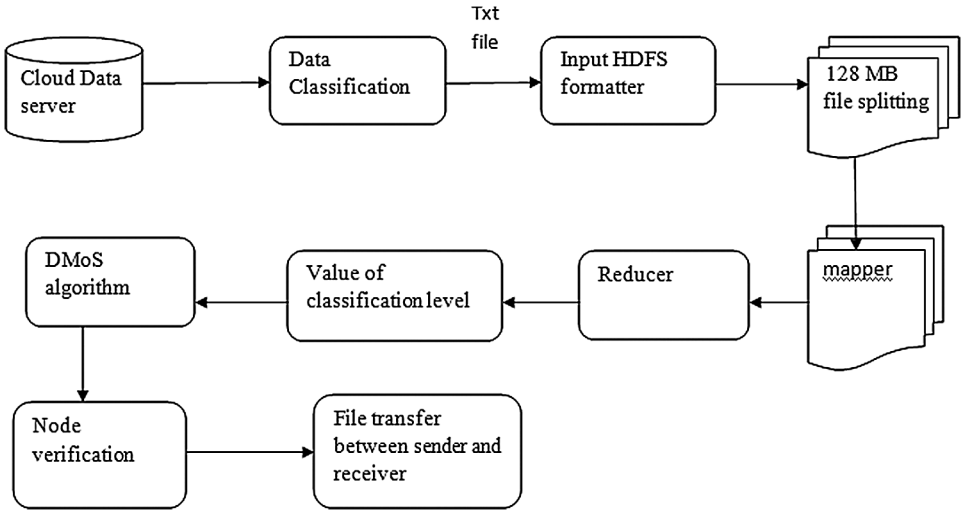

Figure 1: Cloud security architecture

4.1 Cloud Data Classification

Once data are entered in the cloud storage, they are classified and placed under a particular security strategy. Big data are classified on the basis of data sensitivity, safety, protection, and degree of the security it required in the cloud network during transfer. Data are classified into three categories (private, public, and special) based on the impact level of risk. It can be defined as follows:

1. ILR represents the impact level of risk based on metrics assessment (0–5). This measurement is considered ISO27005:2011.

2. Data attributes described as metadata can be represented as AMD, which forms Eq. (3).

$$AMD = \begin{cases} 0, 0 \le ILR \le 1 \\ 1, ILR > 1 \\ 5, \ ILR > 5 \end{cases} \tag{3}$$

If the value of AMD is 1, and then it is true (i.e., data must be confidential). As previously discussed, the impact level of data risk becomes low to high (1–5). Next, if the value of AMD is 0, then it is false (i.e., data

can be publicly viewed). In this situation, the impact level of data risk becomes very low (0–1). If the value of AMD is greater than 5, then it is ranked as special or highly true.

During data classification, the value of AMD is evaluated and inserted in the metadata file of the concerned data. In our proposed system, a classification algorithm is applied to data files that do not have AMD values. Files are categorized as follows:

Private file: This file may have more private sensitive data and can be accessed by authorized users only. This file contains sensitive data related to specific business organizations, personal data of customers, or financial data. This data leakage causes great loss to business organizations, or some intruders may misuse the personal data, thus producing negative feedback for the organization. The metadata of private files do not have sensitive information but provide extra information about the data files related to modification permission. These metadata help administrators have additional information in directories. Private sensitive metadata attributes are used to prevent potential security risks on sensitive files.

Public files: These data are considered simple data. Public files normally contain general information that can be viewed by any user without restrictions on files. These data are not processed in the algorithm for protection.

Special files: These data are more sensitive and can only be sent to a particular user. Metadata have only file type as sensitive. Only users who have permission can access this file.

File format may be txt, doc, image, audio, video, pdf, sql, or xml. Regardless of file type, HDFS classifies them into various partitions. Our proposed system classifies files as text file after performing a classification process. HDFS in the cloud further splits the text files based on the HDFS formatter. This formatter helps convert input data into logical data splits. Each file can come in a size of 128 MB to ensure that it does not exceed the available memory. All split files are assigned to a mapper function ( ). The file security value in each partition is used to identify files as private, public, or special. Some files, such as those in pdf, image, audio, or video formats, cannot be converted to text files by the classification algorithm. These file types are classified on the basis of the file information inside. Then, AMD value is set for the contents in the file and classified as any one of the three methods. If files do not have metadata content, their classification method must be decided the level and insert AMD value for such files.

Data classification in the cloud works as follows:

1. File is processed in HDFS.
2. If the AMD value of the file is true, then the file is classified as private; if the AMD value is highly true, then the file is considered special; if the AMD value is false, then the file is classified as a public file. These files do not need a security processing algorithm.
3. If the AMD file value is not processed, then the files are considered to be audio, video, pdf, or image files, not text files. For such files, the classification level must be set, and an AMD value must be inserted.
4. If AMD not known to doc, txt, sql files, then the HDFS formatter processes the input file and converts it into text.
5. When big data in the cloud are converted as a text file, they are further split into files with a size of 128 MB. A customized HDFS format (CHF) is used to split these files.
6. Partition files are assigned to the HDFS map function ( ).
7. The map function reads the partitioned file content and classifies it into any of the three methods based on the predefined rules.
8. Every partition file has an individual map function to perform.

9. The output of the map function is fed as input to reduce function in cloud HDFS. The reducer function combines the results of map classification and provides single-level classification as output, i.e., 0 is public, 1 is private, and neither 0 nor 1 is special.

4.2 DMoS Working in the Cloud

The classification level of the files is used to provide security procedures to data during mobility in various cloud nodes. If the security level is identified as public, then no security is required for the data. Otherwise, security techniques must be processed for other levels as follows:

1. Metadata attributes are transmitted from user to its sender. In between the sender and receiver cloud, encryption process is performed by sharable keys between users.
2. Random key access is used to transfer the node details of encrypted data and IDs of data files from sender to receiver.
3. The access key of the encrypted block is created by the receiver, and the key is shared by its nodes.
4. Data stored at the sender node is moved or copied by the receiver nodes in the cloud by using the access key of the encrypted block.
5. The sender node receives the request and decrypts the data for security authentication.
6. Once the authentication is processed, data packets are transmitted from sender to receiver. If the receiver confirms that it had received packets by the delivery message, then the sender stops sending packets. Otherwise, trails that failed by exceeding the sending limits are reported to the data administrator for authentication checking.
7. The receiver nodes check the hash value for received data packets.
8. Once the data packet hash value is confirmed by the receiver, encryption acknowledgement is transmitted to sender nodes. If the acknowledgement is missed, then the sender sends repeat copies of the data packet to the receiver. Future repeated packets are deleted.
9. After the sender node receives acknowledgement, it requests the receiver to delete the transmitted data or store the data.

The cloud security process is described in Fig. 2, where the working principle of the security algorithm is described. The symmetric key is used for login by the users, sender, and receiver. For authentication purposes, the level of authentication is analyzed. On the basis of the level, an access key is generated for processing in the nodes.

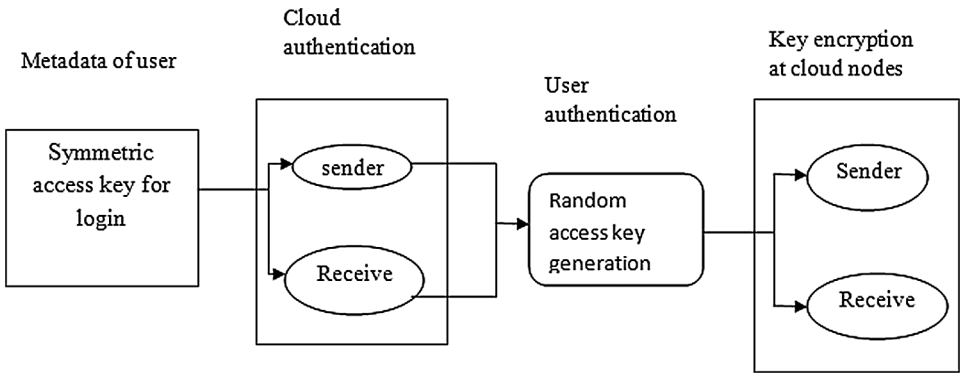

Figure 2: DMoS security algorithm

Analysis of Cloud Security Using the DMoS Algorithm

In each case, the security process and possibilities of data threats are carefully monitored. The invader may identify the metadata transferred between the sender and receiver. Invaders can only receive encrypted metadata. Furthermore, they cannot decrypt the files. Hence, the intruder's attacks are destroyed by our proposed approach. Moreover, if the intruders present themselves as a sender or receiver, they fail in attacking the data because senders and receivers use access keys and hash values in an encrypted manner while encrypted mode in on during data transfer.

A possible way the intruders can attack the data is by breaking the data packets during transmission. In this scenario, data availability is checked by ensuring the delivery of file packets and hash values. Here, we achieve data integrity.

4.3 Algorithms

The two algorithms are processed in our proposed system. The first algorithm is used for cloud data classification, whereas the second (DMoS) is used for securing data mobility. Tab. 1 describes the notations used in the two algorithms.

Table 1: Algorithm notations

HDFS	Hadoop distributed file system
AMD	Attributes of metadata
TXT	Text file
SN	Sender name node
SD	Sender data node
RN	Receiver name node
RD	Receiver data node
ACK	Acknowledgement
SN_{rt}	Number of retransmission to SD
RN_{ds}	Number of copies duplicated in RD
TD	Transferrable data
VTO	Value of timeout
E_k	Encryption key
MRT	Maximum retransmission numbers
t_{sn}	Sender node time
TBA	Token for block access
MDU	Metadata of user

4.3.1 Classification Algorithm 1

Input: AMD value of data files

Output: classification: public, private, special

1. Command line process
   {hadoop fi –get[-C]- name i-j[-en] <path>}
   Obtain file AMD value.
2. **If** AMD value is present for file, then
   Go to algorithm 2.
3. **Else**
   If files are converted to text file, **then**
4. Execute 6–9.
   Else
5. **Execute**
   AMD file is defined.
   Process command line to set AMD: {hadoop fi –setfile- x name [-y value]s-n name <path>}
   Go to DMoS algorithm.
6. Files are converted into small partitions with CHF.
7. Mapper is assigned to all partitioned small files.
8. Classify files using HDFS mapper as private, public, special.
9. Classification results are collected.
10. Reducer function is used to reduce the results of the mapper classification into a single result.
11. **If** one or more output is private,

then

reducer result is private.

12. **Else**, result is considered public
13. **If** the classification result does not produce any value, then the result is considered special.
14. **If** Result = private
    then, set as AMD private file
15. **Execute** DMoS algorithm
16. **Else**
17. Result = public
18. **Do** step 15.

4.3.2 DMoS Algorithm

Input: MDU, TBA, t_{sn}, VTO, MRT, SN_{rt}, RN_{ds}

Output: Secured data

1. If the file is private in AMD

perform 5–22

else, go to step 22

2. SN transfers {MDU} E_k to RN.
3. RN shares TBA with RN.
4. RD transfers {TBA} E_k to SD and call transfer data.
5. SD performs decryption on TBA, and authentication request is checked.
6. SD transmits {TD}$E_{k,}$ hash{(TD) E_k} to RD, begin $t_{sn.}$
7. RD obtains {TD}$E_{k,}$ hash{(TD) E_k}, checks hash.
8. RD transmits (ACK) E_k to SD.
9. **If** t_{sn} < VTO, **then**
10. SD waits until ACK is received.
11. **Else**
12. **If** SN_{rt} < MRT
13. Go to step 6.
14. **Else**
15. SD prompted by the system administrator
16. **If** RN_{ds} > MRT, **then**
17. RD prompted by the system administrator
18. After ACK is received by SD, it deletes TD in it.
19. Stop.

5 Results and Performance Evaluation

Files have different types and may contain structured, unstructured, or semistructured data. When data are processed in the cloud, some information about the files must be kept from the public. In our previous work, maps reduce designed to for security purposes. The same framework is designed here for performance validation. The achievement of the proposed algorithms is described in this section. The proposed framework is designed for data security during data mobility in cloud environments by using knowledge extraction.

Real-time applications in the cloud are increased. Big data in enterprises must be processed with high security. Providing security to sensitive enterprise data is a difficult task. Decision tree based parallel distributed method; Hadoop map reduce technique is accomplished here as a challenging process. Data splitting is considered an important measure function to select attributes for security purposes. Input

HDFS formatter used to divide huge data into multiple partitions of data. Every single partition is checked; if all partitions have the same security order, then the formatter stops splitting. Lastly, the decision tree is computed for the partition. Otherwise, the formatting operation continues splitting data until it meets the security condition of the same class. After the process is completed, a decision tree is generated. Classification steps for security are processed with new data.

5.1 Experimental Setup

The cloud data classification and security algorithms are tested with various data file types with a size of 1–16 GB. Hadoop 2.6.0 Apache is installed with R720 edge servers, two sockets with processor of 6 Intel core of E5-2630 Xeon, RAM of 64 GB, Linux OS, Java 1.0.7 Openjdk, and PuTTY 0.70 security system with an ethernet connection of 1 GB.

5.2 Results

Cloud data are processed with classification and securing algorithms for performance evaluation in calculation with time of data classification, data response, data delay time, and data throughput.

5.2.1 Cloud Data Classification Time

In Tab. 2, classification time is generated and summarized for five files (i.e., xls, csv, sql, and log) as public and secured.

Table 2: Time consumed for file classification

Data type	2 GB	4 GB	16 GB
Sql	195.4	470.6	1590.5
Xls	190.6	420.4	1502.7
Xml	185.4	370.5	1478.4
Doc	160.3	302.4	1402.6

5.2.2 Data Transmission Time

Data transfer between cloud servers are permitted after user security is assured. In the cloud, the sender and the receiver are authenticated by the user by combining the user's private key with the public encryption key. Our experiment is compared with existing nonsecured Hadoop and security algorithms. In HDFS designed to process large nodes. When the size of the data is large, additional challenges are allocated to HDFS. The challenges are mainly about providing security, data availability, and data consistency. The encryption on intracloud data is not supported by the present HDFS; such nonsupport is a major challenge in providing security for cloud data mobility. Kerberos and access control are some features built in HDFS; however, they cannot help in accessing sensitive files. Therefore, other security algorithms are needed to achieve cloud security.

Security techniques are used when data are transferred between cloud devices in the network. Cloud devices are processed by the HDFS method. On the basis of user demand, cloud security is initiated from source to destination cloud devices. User security is checked at the sender and receiver sides of cloud devices. A SSL is initiated between the sender and receiver in cloud nodes. The destination node generates the key temporarily. Tickets and random digits are encrypted using a session key for sender node communication purposes.

A hybrid method for data encryption is suggested. Such methods protect the data with an HDFS session key. Here, file encryption and decryption in cloud nodes use the symmetric encryption method. In addition, the asymmetric method for encryption protects symmetric session keys. This hybrid method also prevents attackers from fetching data from nodes and ensures that clients are light weighted. Above all, our proposed method produces high performance with the proposed architecture.

In Tab. 3, the average time for response during data transmission and security process is tabulated for five various files with the HDFS method at a speed of 64 Mb/s. Data mobility operations need a receiver to know sender metadata. Our proposed method achieves by creating a centralized key for the user to protect the sender and the receiver in the cloud. This common centralized key is used to provide authentication for the sender and receiver nodes in the cloud, enabling the verification of data tokens by not holding the transferred data in the clouds. Unwanted data bandwidths are decreased by our proposed algorithm and architecture. The only drawback with the existing system is the overhead in data processing because of extra data encryption, decryption, and transfer between the sender and receiver nodes of the cloud. Our framework helps reduce the response time of total transmission, delay in total processing time, and data throughput. The proposed architecture also has an improved performance for HDFS.

Table 3: Response time (sec) for cloud data transmission

Method	2 GB	4 GB	16 GB
Baseline [5]	257.55	565.74	2375.65
SSL security [24]	330.53	743.21	3560.22
Hybrid security [25]	395.32	804.98	3870.39
Proposed DMoS security	263.74	602.34	2543.81

The above tables reveal that our proposed framework is better than existing baseline HDFS hybrid methods, proving its superiority over other methods. The performance of data mobility in the cloud environment is improved using the combined algorithms in the proposed system. Meanwhile, data loading times are the same in all methods. In Fig. 3, the results of average time of response are achieved in security techniques of HDFS.

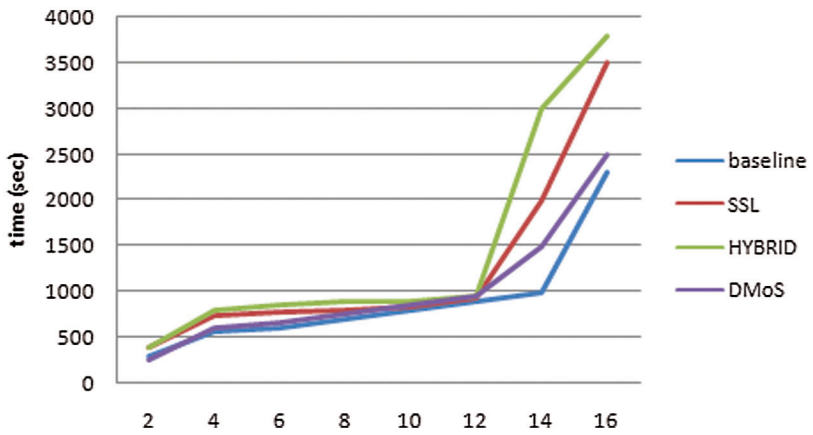

Figure 3: Response time of data

5.2.3 Transmission Delay Time for Data Security

The transmission delay time for data security (TDTDS) is defined as the difference between secured response time in data transmission (SRTDT) and baseline nonsecured response time in data transmission (BRTDT). TDTDS is computed as

$$\text{TDTDS} = \text{SRTDT} - \text{BRTDT}. \tag{4}$$

In Fig. 4, the delay time due to HDFS in cloud csv files is shown. Secured data transmission is delayed, which is less than that of other existing methods. In addition, cloud throughput decreases due to data security during mobility in cloud network.

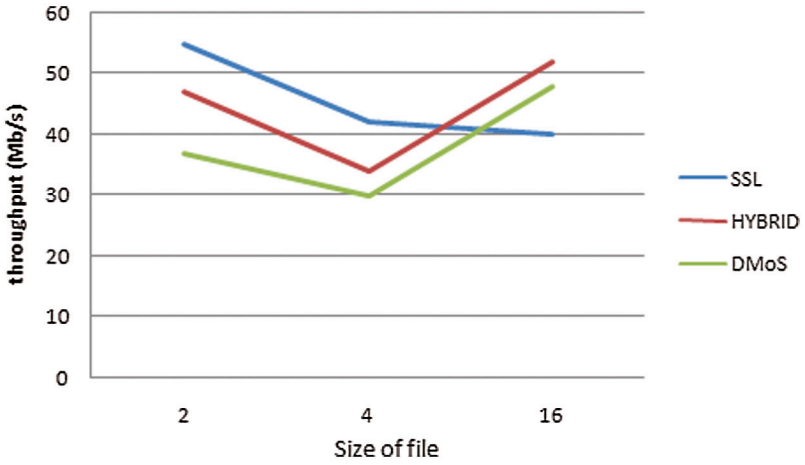

Figure 4: Throughput estimation graph

Throughput of data security: The HDFS security methods produce little side effects during data transfer. Throughput for secured data transfer (TSDT) in the cloud is defined as data transfer amount (DTA). DTA is size of file is represented as bits from sender to receiver. TSDT is computed as

$$\text{TSDT} = \text{DTA}/\ \text{SRTDT}. \tag{5}$$

Throughput (in Mb/s) using the HDFS method of data transmission is presented in Tab. 4.

Table 4: Throughput (Mb/s) for secure data transmission

Method	2 GB	4 GB	16 GB
Baseline [5]	62.43	58.61	51.65
SSL security [24]	55.21	42.10	37.23
Hybrid security [25]	47.84	34.78	30.42
Proposed DMoS security	56.42	52.23	48.63

As shown in the figure, when the size of the file increases, the throughput of data transmission in the cloud network decreases. Therefore, HDFS security methods cause shortages in transmission throughput. The proposed experiment split the file into smaller sizes; thus, HDFS performed best in our method. As shown in Tabs. 2, 3, and 4, the proposed algorithms outperform the existing methods.

5.3 Performance Evaluation

The security method in introduces an SSL to protect the data transmission in the network. SSL provides connection between the sender and receiver nodes of the cloud environment. The receiver produces a temporary session key with a hash value, and tickets are returned to the sender. This method produces an extra security process; thus, system performance is reduced. When we process big data using SSL, the performance level decreases.

The security method in ignores receiver acknowledgement. This ignorance between sender and receiver allows intruders to fix extra packets, leading to serious threats in data transfer. After the sender sends the file to the destination without acknowledgement, the sender might not know whether data are received correctly. This situation leads intruders to play a threat role between the cloud nodes. Consider attacker send the data, packets from sender side and delete the file but receiver does not know attacker file. Then, the threat is high here. In addition, data packet retransmission leads to an increase in bandwidth over the network.

Existing methods waste bandwidths due to extra data transmission, extra data encryption, and timing delay. Furthermore, throughput decreases, leading to low performance for data mobility in the cloud network. In our proposed system, keys are safely transmitted with data encryption, and only the data owner knows it. Public cloud data are easy to handle. This advantage makes data mobility friendly and easy in the cloud network.

Attacks in the reply mode are prevented by sending data with hash values after decrypting access tokens. Senders can only encrypt hash data because user information files are encrypted and stored in the cloud. Thus, in our proposed approach, data integrity and confidentiality are ensured. User privacy and data confidentiality are also provided. The data communication between user and sender/receiver is ensured by shared keys and tokens in the cloud. The acknowledgement method is used by the system to avoid data loss. Metadata encryption in our protocol protects the data from high attackers. In the middle, attacks on data packets by changing or deleting are prevented by data integrity.

6 Conclusions

The cloud data classification algorithm with DMoS provides superior data protection in the cloud network. It avoids threats and risks from intruders through improved data mobility in the cloud network. The proposed method easily identifies which data need to be secured by using a classification algorithm. It also reduces the extra cost for processing public files. Files that are identified using classification algorithms are kept confidential and secured during data transmission in the network. Redundant data encryption and data decryption are avoided to increase performance. Evaluation results prove the usefulness of the proposed integrated architecture in terms of cloud data mobility. By using a software tool, files that are secured and distributed in several cloud nodes are enhanced effectively with minimal time. The tool is also used to describe metadata for security confirmation. Experimental analysis is performed in real-world cloud applications to achieve excellent performance by combined methodology.

In future work, security constraints in big data classification are considered with high classification techniques. Furthermore, audio, video, and image classification files will be considered. These data types need special algorithms in nature. New technologies and algorithms are needed to handle big data in the future.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare no conflict of interest on the publication of this paper.

References

[1] P. Zhao, W. Yu, S. Yang, X. Yang and J. Lin, "On minimizing energy cost in internet-scale systems with dynamic data," *IEEE Access*, vol. 5, pp. 20068–20082, 2017.

[2] K. Y. Teng, S. A. Thekdi and J. H. Lambert, "Risk and safety program performance evaluation and business process modeling," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 6, pp. 1504–1513, 2012.

[3] M. Paryasto, A. Alamsyah and B. Rahardjo, "Big-data security management issues," in *Proc. ICoICT*, Bandung, Indonesia, pp. 59–63, 2014.

[4] Hadoop Resources, 2021, [Online]. Available: http://mirrors.sonic.net/apache/hadoop/common/hadoop2.6.0/.

[5] A. K. Tiwari, H. Chaudhary and S. Yadav, "A review on big data and its security," in *Proc. ICIIECS*, Coimbatore, India, pp. 1–5, 2015.

[6] J. V. Gautam, H. B. Prajapati, V. K. Dabhi and S. Chaudhary, "A survey on job scheduling algorithms in big data processing," in *Proc. ICECCT*, Tamil Nadu, India, pp. 1–11, 2015.

[7] A. Holmes, *Hadoop in practice*. New York: Manning, 2012.

[8] A. Sinha and P. K. Jana, "A hybrid mapreduce-based k-means clustering using genetic algorithm for distributed datasets," *The Journal of Supercomputing*, vol. 74, no. 4, pp. 1562–1579, 2018.

[9] A. Nasridinov and Y. H. Park, "Visual analytics for big data using R," in *Proc. CGC*, Karlsruhe, Germany, pp. 564–565, 2013.

[10] S. H. Kim, J. H. Eom and T. M. Chung, "Big data security hardening methodology using attributes relationship," in *Proc. ICISA*, Pattaya, Thailand, pp. 1–2, 2013.

[11] N. Chaudhari and S. Srivastava, "Big data security issues and challenges," in *Proc. ICCCA*, Greater Noida, India, pp. 60–64, 2016.

[12] A. Katal, M. Wazid and R. H. Goudar, "Big data: Issues, challenges, tools and good practices," in *Proc. IC3*, Noida, India, pp. 404–409, 2013.

[13] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Proc. CTS*, San Diego, CA, USA, pp. 42–47, 2013.

[14] T. Mahmood and U. Afzal, "Security analytics: Big data analytics for cybersecurity—A review of trends, techniques and tools," in *Proc. NCIA*, Rawalpindi, Pakistan, pp. 129–134, 2013.

[15] R. Lu, H. Zhu, X. Liu, J. K. Liu and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *IEEE Network*, vol. 28, no. 4, pp. 46–50, 2014.

[16] M. Shanmugam, A. Ramasamy, S. Paramasivam and P. Prabhakaran, "Monitoring the turmeric finger disease and growth characteristics using sensor based embedded system," *Scientific Research*, vol. 7, no. 8, pp. 1280–1296, 2016.

[17] S. Maheswaran, M. Ramya, P. Priyadharshini and P. Sivaranjani, "A real time image processing based system to scaring the birds from the agricultural field," *Indian Journal of Science and Technology*, vol. 9, no. 30, 2016.

[18] A. Khalid and M. Shahbaz, "Adaptive deadline-aware scheme (ADAS) for data migration between cloud and fog Layers," *KSII Transactions on Internet & Information Systems*, vol. 12, no. 3, 2018.

[19] T. Payton and T. Claypoole, *Privacy in the age of big data: Recognizing threats, defending your rights, and protecting your family*. Rowman & Littlefield, 2014. [Online]. Available: https://www.amazon.com/Privacy-Age-Big-Data-Recognizing/dp/1442225459.

[20] K. Davis, *Ethics of big data*. Balancing Risk and Innovation, O'Reilly Media, Inc., 2012. [Online]. https://www.oreilly.com/library/view/ethics-of-big/9781449314873/.

[21] Y. Gahi, M. Guennoun and H. T. Mouftah, "Big data analytics: Security and privacy challenges," in *Proc. ISCC*, Messina, Italy, pp. 952–957, 2016.

[22] R. Alguliyev and Y. Imamverdiyev, "Big data: Big promises for information security," in *Proc. AICT*, Astana, Kazakhstan, pp. 1–4, 2014.

[23] S. Marchal, X. Jiang, R. State and T. Engel, "A big data architecture for large scale security monitoring," in *Proc. Big Data*, Anchorage, AK, USA, pp. 56–63, 2014.

[24] E. Bertino and E. Ferrari, "Big data security and privacy," in *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer, pp. 425–439, 2018.

[25] T. Zaki, M. S. Uddin, M. M. Hasan and M. N. Islam, "Security threats for big data: A study on enron e-mail dataset," in *Proc. ICRIIS*, Langkawi, Malaysia, pp. 1–6, 2017.