IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 7, July 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907008 |

A Hybrid Cloud Approach for Secure Authorized Deduplication with Attribute-Based Encryption

Dumbre Anita S[1], Prof. Monika D. Rokade. [2]

PG Student, Department of Computer SPCOE, Dumbarwadi (Otur) Pune, India

Assistant Professor (ME Co-coordinator), Department of Computer SPCOE, Dumbarwadi (Otur) Pune, India

ABSTRACT: Nearly all organizations, the storage space contains duplicate copies of several blocks of data. Deduplication removes redundant copies by saving just one copy of the data and replacing the other copies with pointers pointing to the original copy. Companies frequently use deduplication in backup and disaster recovery applications, but it can be used to free up space in primary storage as well. Deduplication will save the bandwidth while uploading a file in cloud storage. To protect the confidentiality of the sensitive data while supporting deduplication, data is encrypted by Attribute-Based Encryption. If file uploader gives *n* attributes while encrypting the file, then while decryption any *k* out of *n* correct attributes should form a decryption key runtime and the user will be able to decrypt data using that key.To demonstrate the proposed system, an application to securely store text files is planned for. Data Owner is the one who owns the file and uploads his encrypted file to the public cloud. When the data owner uploads the file, it should not be mandatory for him to know in advance who will be using his file in the public cloud. So when data owner uploads the file, with some attributes he will encrypt the file and uploads it to the cloud. When some other data user wants to access the same file, he will specify a few of the attributes. If the attributes are correct the key will be generated runtime and the user will be able to access the data.

I.INTRODUCTION

Today's cloud service suppliers offer both hugely accessible storage and vastly parallel computing resources at comparatively lesser costs. As cloud computing becomes predominant, an increasing quantity of data is being stored in the cloud and shared by users using particular privileges, which illustrate the access rights of the stored information. One serious challenge of cloud storage services is the management of the increasing volume of data.

Data deduplication is a special type of data compression technique to eliminate duplicate copies of iterating data in storage. Deduplication improvesstorage utilization and also reduce the number of bytes that have to be sent to the cloud. Instead of storing multiple data copies with the same content, deduplication eliminates duplicate data by keeping only one physical copy and referring other duplicate data to that copy.

Privacy and security concerns arise because of users' sensitive data are susceptible to both insider and outsider attacks. Traditional encryption technique requires different users to encrypt their data with his/her own keys. Therefore, indistinguishable data copies of different users form different ciphertexts, making deduplication almost impossible.

Most existing public key encryption methods allow a party to encrypt data to a particular user. Considering real life scenario, it is possible that the user doesn't know in advance who will be using her file in future. So a system is required where for the user it is not mandatory to know in advance who will use a file. Instead, an owner should give the attributes which will be used to encrypt the file. A legitimate user who has the correct attributes can able to decrypt the data. This system also achieves one to many encryption.

II.RELATED WORK

Dropbox[1], Google and many more service providers perform deduplication which saves space by storing only one copy of each file uploaded. In traditional encryption, a key used for encryption is different per user. This results in completely

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907008 |

different encrypted data though the contents are same. DeDu[22] is efficient deduplication system but fails to handle encrypted data. The solution for this problem is provided in Message-Locked Encryption (MLE) [2] The MLE is Convergent Encryption (CE), introduced by Douceur et al. [3] and others [4], [5], [6]. Many commercial and research storage service systems use CE. CE working: Let *M* be a file's data. Cryptographic hash function*H* is applied to the data *M*.The result of this function is an encryption key *K: K←H*. With this key ciphertext is computed *C← E(K, M)* via a deterministic symmetric encryption scheme. When another user comes with the same file the resultant cryptographic hash will be the same which produces the same encryption key. Hence making deduplication possible. However, CE has limitation to offline brute-force dictionary attacks [7], [8]. Knowing that the target data *M* underlying the target ciphertext*C* is drawn from a dictionary *S={M1.Mn}* of size *n*, an attacker can recover *M* in the time for *n=|S|* off-line encryptions: for each *i = 1..., n*, it simply CE encrypts M_i to get a ciphertext denoted as $C_i$and returns M_i such that $C = C_i$. This works because CE is deterministic and keyless. The security of CE is only possible when the target data is drawn from a space too large to exhaust.

Another problem with CE is to access the file user must have the file contents.
A token is hash of the privilege key of user and data to be uploaded to the cloud. The scheme presented in [9] combines user privilege to obtain a file token with token unforgeability. If a user is not a legitimate user but if he has the hash of the file then attack of manipulation of data identifier is possible. To avoid this Meye et al. proposed to adopt two servers for intra-user deduplication and interdeduplication[10]. The user key is used to encrypt the ciphertext C of CE and then transferred to the servers ClouDedup [11] tries to cope with the security exposures of CEy using data deletion. But data deletion has its certain issues and ClouDedup [11] cannot address these issues. Like a data holder who removed the data from the cloud can still access the data because the encryption key is known. If data is present in the cloud then with encryption key he can decrypt it.

Data Ownership Verification and Others
In the fig-1 L1, L2, L3, L4 are data blocks. On next level hash of L1, a hash of L2, a hash of L3 and hash of L4 is taken. On the next level hash of 2 blocks are concatenated. And then root stores concatenation of all bottom hash. So in Merkle tree, every leaf node is labeled with a hash of data block. Every non-leaf node is labeled with the cryptographic hash of the labels of its child nodes.

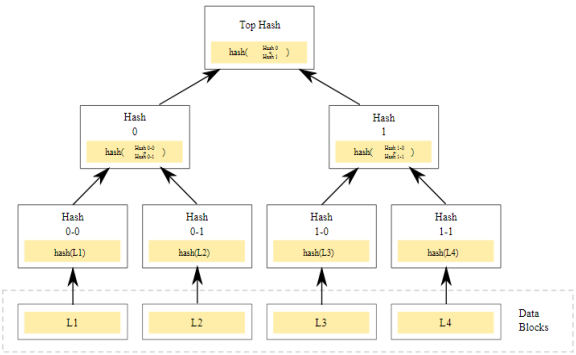

Fig-1 : Role of ownership

Proofs of Ownership (PoW) practical implementation is first introduced by Halevi et al. [12]. This uses Merkle tree for deduplication, which realized client-side deduplication. The proposed system applies a hash function over original file and then use Merkel tree on pre-processed data and generate information for verification. When the authenticity of data user needs to be proved a verifier randomly chooses several leaves of the tree. It traces down the sibling-paths of all these leaves. If the checksum matches that prove the authenticity of the user. Only when all paths are valid, will the verifier accept the proof. But this scheme has a drawback that server for data storage could be aware of file contents. Ng et al. [13] implemented PoW to provide security to deduplication of encrypted data. The system proposed also use Merkle trees to generates verification information for deduplication. Based on several data blocks each leaf value is generated. Each interactive proof protocol can only challenge one leaf of the Merkle tree. To achieve higher security the system needs to be executed several times causing additional overhead.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907008 |

Hybrid data deduplication is proposed by Fan et al. This system provides a practical solution with partial semanticsecurity [14]. Deduplication on ciphertext and plaintext is presented. But this system lacks encrypted data deduplication. The CSP has data encryption key. But data holders or owners don't trust CSP. This problem is overcome in A Hybrid Cloud Approach for Secure Authorized Deduplication. The paper introduces new entity called as the Private cloud which acts as an execution interface between Data owner and CSP. The token generation, privilege key generation is done by Private cloud.

Monika Rokade and YogeshPatil [21] proposed a system deep learning classification using nomaly detection from network dataset. The Recurrent Neural Network (RNN) has classification algorithm has used for detection and classifying the abnormal activities. The major benefit of system it can works on structured as well as unstructured imbalance dataset.

The MLIDS A Machine Learning Approach for Intrusion Detection for Real Time Network Dataset has proposed by Monika Rokade and Dr. YogeshPatil in [22]. The numerous soft computing and machine learning classification algorithms have been used for detection the malicious activity from network dataset. The system depicts around 95% accuracy ok KDDCUP and NSLKDD dataset.

Monika D. Rokade and Yogesh Kumar Sharma [23] proposed a system to identification of Malicious Activity for Network Packet using Deep Learning. 6 standard dataset has sued for detection of malicious attacks with minimum three machine learning algorithms.

Sunil S. Khatal and Yogeshkumar Sharma [24] proposed a system Health Care Patient Monitoring using IoT and Machine Learning for detection of heart and chronic diseases of human body. The IoT environment has used for collection of real data while machine learning technique has used for classification those data, as it normal or abnormal.

Data Hiding In Audio-Video Using Anti Forensics Technique For Authentication has proposed by Sunil S.Khatal and Yogeshkumar Sharma [25]. This is a secure data hiding approach for hide the text data into video as well as image. Once sender hide data into specific objects while receivers does same operation for authentication. The major benefit of this system can eliminate zero day attacks in untrusted environments.

Sunil S.Khatal and Yogesh Kumar Sharma [26] proposed a system to analyzing the role of Heart Disease Prediction System using IoT and Machine Learning. This is the analytical based system to detection and prediction of heart disease from IoT dataset. This system can able to detect the disease and predict accordingly

III.PROPOSED SYSTEM

The proposed system allows the user to check the file duplication before uploading to the cloud. For the security of the file, it is encrypted using Attribute-Based Encryption. To demonstrate proposed system File Management is planned for. The user can upload a file by entering attributes associated with file, download file if he has the correct attributes. Also, check file duplication before file upload

There are 4 entities present.
Data owner- Who outsource data on the public cloud, he can access the data later. Only unique data is uploaded on CSP. In this system, each user has given a set of privileges in the system setup. Each file is protected with Attribute-based encryption.
Data User- Who reads the file from the cloud.
S-CSP- This is public cloud which provides service of data storage. To reduce the cost of storage deduplication is used while uploading the data.
Private cloud- As S-SCP is not fully trusted private cloud is used. Private cloud acts as an interface between a user and public cloud. Private cloud generates privilege keys which are used while generating token.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907008 |

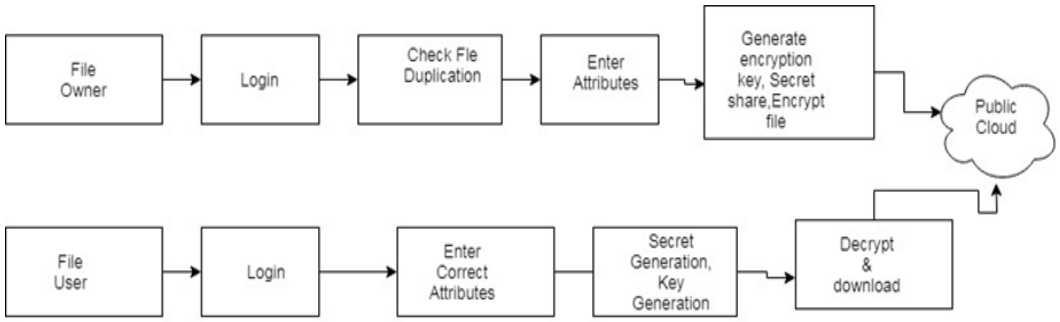

Fig-2 Process flow

Step1- Data owner/user registers themselves to private cloud.
Step 2-Then Data owner/user asks Private cloud to access rights for him.
Step 3- Private cloud generates privilege key for access right per user at the back end. Once the data owner user is assigned the privileges he can now upload download file.
Step 4-When data owner tries to upload the file, file duplication is checked first.

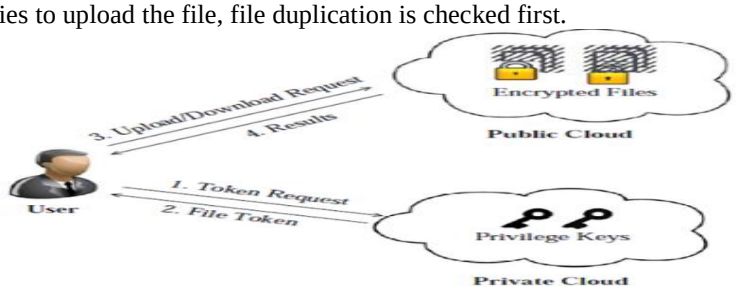

Fig-3 Data Storage view

Step 4.1- Data owner sends a hash of file contents to private cloud.
Step 4.2-Private cloud generates token ϕF,p = H(F, Kp) and send back to the Data Owner.
Step 4.3- Data Owner sends token to S-CSP.
Step 4.4-S-CSP verifies token and send duplication status back to Data Owner.
Step 4.5- If a file is the duplicate file is not uploaded to the cloud.
Step 4.6- If a file is not duplicate Attribute-based encryption is applied.
Step 5: Attributes related to the file are entered while uploading a file.
Step 6. A file is encrypted with AES & send to the S-CSP.
Step 7. When data user wants to download the file, he is asked to enter the attributes, if attributes are correct then a file is decrypted and downloaded securely. If there are total N attributes then at least K attributes should be correct to be able to download the file.

IV.IMPLEMENTATION

Java realization for a model of Hybrid Cloud Approach for Deduplication with Attribute-Based Encryption algorithm system consists of following modules:
1. Data Owner/User Registration:
2. Duplication check & File upload module
3. File Download
There are 3 entities present. Private cloud is taken as local machine. Amazon EC2 cloud for System for Uploading/downloading file.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907008 |

1. Registration Module:
This module is used for user registration of the proposed system. When a user wants to access the proposed system, first step is to register in the system. In this module, user details are taken for registration. At the backend, for every user, two methods are called. The first method is generateNextID(), which gives a unique id to a user. Each time a new user registers in a system, auto-increment of role user id is done each time. This module is executed by the private cloud.
2. Data Owner/User request private cloud to assign privileges:

Owner/User side:Owner/User registers himself and asks private cloud to set his privileges. For that Owner/User sends the username, email to the private cloud.
The request is sent to private cloud by Amazon Simple Notification Service. Amazon SNS is a web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients. The message is send using syntax:
SNS.send(String arg0,String arg1,String arg2,String arg3,String arg4)
Private cloud side:Id, username, Token, finalkey, emailed, request values are the fields present at Private cloud side.
Privilege key is the key representing read/write access for data owner user.
The final key is the privilege key generated by the private cloud.
Request Value=0 represents data owner/user file read/write privileges are not set by the private cloud.
Request Value=1 represents data owner/user file read/write privileges are set by the private cloud.
A token is SHA1(File, Privilege key).
SNS.Receive() method is written in the private cloud to receive an Email and username from the data owner/user. When the data owner registers himself he requests private cloud to assign him privileges for a file. Email id, username, Ip address are passed to the private cloud through SNS. After receiving these parameters Private cloud grants access permission to data owner/user. Private cloud generates the privilege key of reading permission or privilege key of write permission or key for both read-write permission together depending on the privileges assigned to data owner/user.
When data owner/user privileges are not set by private cloud the Request value is set to 0. Once the privilege key is generated by Private cloud then Id, username, finalkey, emailed, request values are updated accordingly. The Token is set 0 before file upload.
File uploader side:
Duplication Check:

The private keys for privileges will not be issued to users directly, which will be kept and managed by the private cloud server instead.
To get a file token, the user needs to send a request to the private cloud server. To perform the duplicate check for some file, the user needs to get the file token from the private cloud server. The private cloud server will also check the user's identity before issuing the corresponding file token to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file. Based on the results of the duplicate check, the user uploads this file.
Suppose that a data owner wants to upload and share a file F with users whose privilege belongs to the privilege set. The data owner needs to interact with the private cloud before performing a duplicate check with the public cloud. More precisely, the data owner performs an identification to prove its identity with a private key. The user computes and sends the file hash to the private cloud server, who will return token=SHA256(file, privilege key) back to the user then, the user will interact and send the file token Public cloud with SNS. SNS.receive() method is written in the public cloud to receive the data by a user. Public cloud checks the token is available or not and send back the duplication status to file uploader.

FileTag(File)—It computes a SHA-1 hash of the File as File Tag;
TokenReq(Tag, UserID)—It requests the Private Server for File Token generation with the File Tag and User ID;
DupCheckReq(Token)—It requests the Storage Server for Duplicate Check of the File by sending the file token received from the private server;
Taking input as attributes.

If a file is not duplicate the uploader enters attributes related to file and send to the private cloud.
3. Generating Shamir secret. Generating Shamir shares corresponding to each attribute.
ABE algorithm uses Shamir Secret sharing scheme.
Shamir Secret Sharing is a form of secret sharing, where a secret is divided into parts, giving each participant its own unique part, where some of the parts or all of them are needed in order to reconstruct the secret. This property is used in ABE.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907008 |

When the file owner sets attributes, Shamir secret share is generated corresponding to every attribute. The encryption key is generated based on attributes entered while uploading the file. The uploaded file is encrypted and sent to the public cloud.*n* is the number of secret shares to produce, *k* the threshold i.e. out of *n* parts minimum *k* parts should be produced to form a secret. An implementation of Shamir's Secret Sharing to securely split secrets into *n* parts, of which any *k* can be joined to recover the original secret is as below.

Polynomial initialization: We wish to divide the secret into *n* parts where any subset of *k* parts is sufficient to reconstruct the secret. At random we obtain two *k-1* numbers. The polynomial will be of degree *k-1*. The coefficient of polynomial *r* should be greater than 0 and less than prime number generated randomly.
Shamir secret shares generation:
Input: BigInteger secret, threshold *k*, total share created *n*, BigInteger prime, random number
$f(x)=a_0 + a_1x + a_2x^2 + a_3x^3 + \ . \ . + a_{k-1}x^{k-1}$
A secret is assigned at a_0. Secret shares are generated from the above equation.
For each attribute entered while uploading file secret, a share is calculated and assigned to the attribute. The encrypted file and the secret of the file are sent to the public cloud for storage. Key generation and file encryption are mentioned below.
For key generation by the data user.

Step 1: User enters the n attribute with the search query to access the respective file
Step 2: for each attribute

$$\boldsymbol{P[Byte]} = \sum_{\boldsymbol{k=0}}^{\boldsymbol{n}} \binom{\boldsymbol{n}}{\boldsymbol{S[k]}}$$

Step 3: convert PK as ObjectList to 64 bit String decoder text
Encrypt file with encoder and send to the public cloud.

At data user side-
A user will choose the file, enter the attributes related to that file. Vector Cosine Similarity(VCS) of the entered attributes are checked with attributes entered while encrypting the file. The algorithm is described below. If the VCS is above a threshold the Shamir secret the is calculated from the attributes. If the secret formed is correct then a user is able to download the file else not.

The secret calculation is done by Lagrange's interpolation.

$$\ell_0 = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2}$$

$$\ell_1 = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2}$$

$$\ell_2 = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1}$$

$$f(x) = \sum_{j=0}^{2} y_j \cdot \ell_j(x)$$

V.EVALUATION

We conduct an evaluation of our proposed system. Our evaluation focuses on comparing the following:
File upload time for CE & ABE is compared below.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907008 |

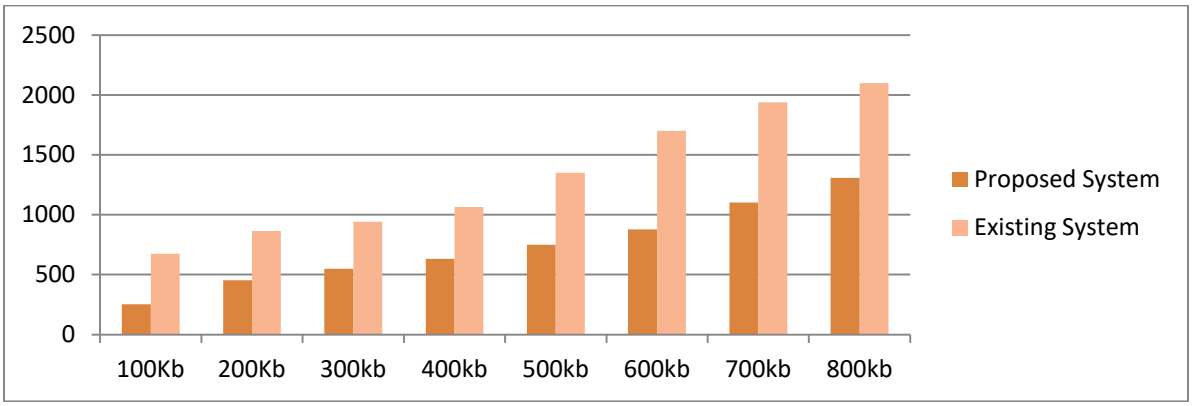

Figure 4: File download time for CE & ABE are compared below

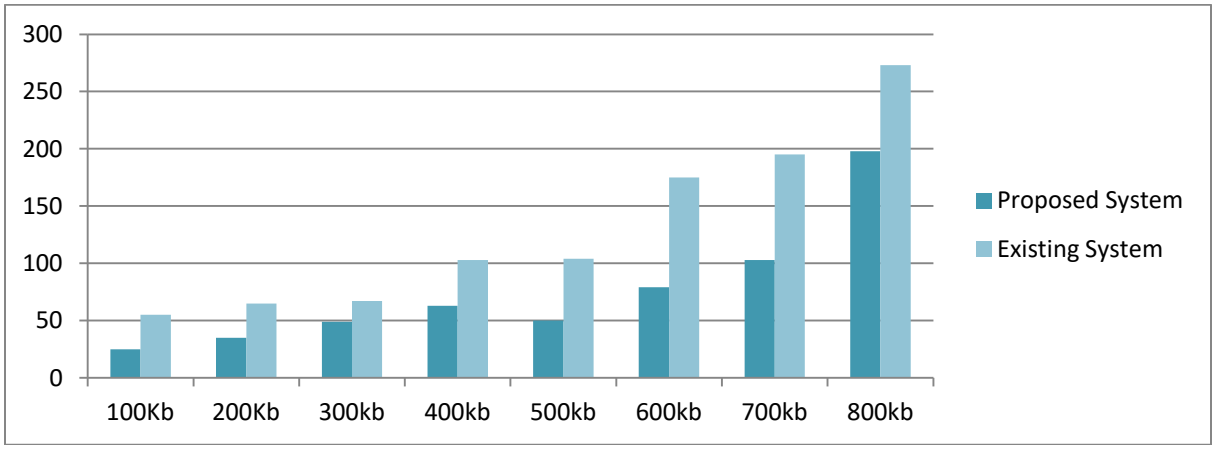

Figure 5 : Proposed vs existing

VI.CONCLUSION

To secure user privacy, numerous privacy-preserving category methods have been suggested over the past several years. The current methods are not appropriate to contracted database surroundings where the information exists in secured form on a third-party server. This paper suggested a novel method for storing data on cloud and checking the duplicate data.

REFERENCES

[1] Dropbox, A file-storage and sharing service. (2016). [Online].Available: http://www.dropbox.com
[2] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer,"Reclaiming space from duplicate files in a serverless distributedfile system," in Proc. IEEE Int. Conf. Distrib. Comput.Syst., 2002,pp. 617–624, doi:10.1109/ICDCS.2002.1022312.
[3] G. Wallace, et al., "Characteristics of backup workloads in productionsystems," in Proc. USENIX Conf. File Storage Technol., 2012,pp. 1–16.
[4] Z. O. Wilcox, "Convergent encryption reconsidered," 2011.[Online]. Available: http://www.mailarchive.com/cryptography@metzdowd.com/msg08949.html
[5] J. Pettitt, "Hash of plaintext as key?" (2016). [Online]. Available:http://cypherpunks.venona.com/date/1996/02/msg02013.html
[6] The Freenet Project, Freenet. (2016). [Online]. Available: https://freenetproject.org/
[7] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-lockedencryption and secure deduplication," in Proc. Cryptology—EUROCRYPT,2013, pp. 296–312, doi:10.1007/978-3-642-38348-9_18.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907008 |

[8] D. Perttula, B. Warner, and Z. Wilcox-O'Hearn, "Attacks on convergentencryption." (2016). [Online]. Available: http://bit.ly/yQxyvl
[9] J. Li, Y. K. Li, X. F. Chen, P. P. C. Lee, and W. J. Lou, "A hybridcloud approach for secure authorized deduplication," IEEE Trans.Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206–1216, May 2015,doi:10.1109/TPDS.2014.2318320.
[10] P. Meye, P. Raipin, F. Tronel, and E. Anceaume, "A secure twophasedata deduplication scheme," in Proc. HPCC/CSS/ICESS,2014, pp. 802–809, doi:10.1109/HPCC.2014.134.
[11] P. Puzio, R. Molva, M. Onen, and S. Loureiro, "ClouDedup:Secure deduplication with encrypted data for cloud storage," inProc. IEEE Int. Cof. Cloud Comput. Technol. Sci., 2013, pp. 363–370,doi:10.1109/CloudCom.2013.54.
[12] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs ofownership in remote storage systems," in Proc. 18th ACM Conf.Comput. Commun.Secur., 2011, pp. 491–500, doi:10.1145/2046707.2046765.
[13] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocolsin cloud storage," in Proc 27th Annu. ACM Symp. Appl. Comput.,2012, pp. 441–446
[14] C. Fan, S. Y. Huang, and W. C. Hsu, "Hybrid data deduplicationin cloud environment," in Proc. Int. Conf. Inf. Secur.Intell.Control,2012, pp. 174–177, doi:10.1109/ISIC.2012.6449734.
[15] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P.C. Lee, and Wenjing Lou "A Hybrid Cloud Approach for Secure Authorized Deduplication" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 26, NO. 5, MAY 2015. pp-1206-1216
[16] J. Bethencourt, A. Sahai, and B. Waters " Ciphertext-Policy Attribute-Based Encryption" In Proceedings of the 2007 IEEE Symposium on Security and Privacy SP '07,Page no: 321-334, Washington, DC, USA, 2007. IEEE Computer Society.
[17] John R. Douceur,AtulAdya,William J. Bolosky,DanSimon,MarvinTheimer
"Reclaiming Space from Duplicate Files in a Serverless Distributed File System" July 2002 Technical Report MSR-TR-2002-30
[18] Ravi S. Sandhu Edward J. Coynek, Hal L. Feinsteink and Charles E. Youmank
" Role-Based Access Control Models_yz"
Revised October 26, 1995 IEEE Computer, Volume 29, Number 2, February 1996, pages 38-47.
[19]ShaiHalevi, Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg
"Proofs of Ownership in Remote Storage Systems " 2011/8/11
[20] MihirBellar, SriramKeelveedhi, Thomas Ristenpart "Message-Locked Encryption and Secure Deduplication"
A preliminary version of this paper appears in the proceedings of Eurocrypt 2013. This is the full version. March 2013
[21] Monika D.Rokade ,Dr.YogeshkumarSharma,"Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic."IOSR Journal of Engineering (IOSR JEN),ISSN (e): 2250-3021, ISSN (p): 2278-8719
[22] Monika D.Rokade ,Dr.YogeshkumarSharma"MLIDS: A Machine Learning Approach for Intrusion Detection for Real Time Network Dataset", 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE
[23]Monika D.Rokade, Dr. Yogesh Kumar Sharma. (2020). Identification of Malicious Activity for Network Packet using Deep Learning. *International Journal of Advanced Science and Technology*, *29*(9s), 2324 - 2331.
[24] Sunil S.Khatal ,Dr.Yogeshkumar Sharma, "Health Care Patient Monitoring using IoT and Machine Learning.", IOSR Journal of Engineering (IOSR JEN), ISSN (e): 2250-3021, ISSN (p): 2278-8719
[25]Sunil S.Khatal ,Dr.Yogeshkumar Sharma, "Data Hiding In Audio-Video Using Anti Forensics Technique ForAuthentication ", IJSRDV4I50349, Volume : 4, Issue : 5
[26]Sunil S.Khatal Dr. Yogesh Kumar Sharma. (2020). Analyzing the role of Heart Disease Prediction System using IoT and Machine Learning. *International Journal of Advanced Science and Technology*, *29*(9s), 2340 - 2346.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 7.542

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details