International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307 OPEN ACCESS

Available Online at :www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT2390521

Fortifying Data Availability and Security in Cloud Servers Via Verification Protocols

[1]Mr. N.Ananda Reddy, [2]Dr. Pratap Singh Patwal,

[1] Research Scholar, Computer Science and Engineering Department, GLOCAL University, Saharanpur, Uttar Pradesh, India

[2] Professor, Computer Science and Engineering Department, GLOCAL University, Saharanpur, Uttar Pradesh, India

ARTICLEINFO

Article History:

Accepted: 07 Sep 2023

Published: 25 Sep2023

Publication Issue

Volume 9, Issue 5

September-October-2023

Page Number

151-159

ABSTRACT

Distributed computing networks play a significant role in contemporary computer architecture, facilitating the processing, analysis, and storage of sensitive data. This issue has garnered significant interest in the realm of investigating various encryption techniques. There is a pressing need to establish a dependable and validated security approach that enhances the data-processing efficacy of Cloud servers, as current cryptographic investigations suggest the vulnerability of server-side data security to potential breaches. This paper elucidates the process of implementing a cloud-centric cloud storage architecture and presents a unique security schema that validates the evidence of encrypted secret data storage. The researchers in this paper propose a novel computational security technique to tackle the issue of internal nodes by using efficient security strategies, hence minimizing computational overhead. The experimental results reported in this paper provide evidence of the greater effectiveness of the suggested mechanism in comparison to other commonly used security strategies.

Keywords: Cloud Computing, Security, Distributed File System

I. INTRODUCTION

Over the course of the last century, advancements in computer technology have facilitated the use of automation across several industries. There is a widespread belief among individuals that the use of cloud technology and cloud computing has the potential to bring about a significant transformation in the field of computing in the next years. The term "cloud computing" was first used at the 2006 SES (Search Engine Strategies) international conference held in San Jose. Subsequently, the definition of NIST standards [1] has been elucidated. The National Institute of Standards and Technology (NIST). Cloud architecture, because to its inherent authenticity and adaptability, has constantly had a significant influence on the economic policies of corporations in terms of storage and service supply. Researchers from many

Copyright © 2023 The Author(s): This is an open-access article distributed under the terms of the Creative Commons Attribution **4.0 International License (CC BY-NC 4.0)** which permits unrestricted use, distribution, and reproduction in any medium for non-commercial use provided the original author and source are credited.

regions worldwide have been diligently striving to enhance the cost-effectiveness and efficiency of the underlying architectural model of the Cloud, particularly in relation to data storage and supply. Storage is often regarded as the most prevalent cloud service, despite the existence of other alternatives. The exponential growth in data volume and velocity in recent years can be attributed to the rapid advancement of networks and the introduction of key technologies like the Internet of Things (IoT) and Industrial Internet of Things (IIoT). These technologies play a crucial role in collecting diverse data from multiple sources, making them indispensable resources. This phenomenon arises from the circumstance that, in the event that the computation of such vast quantities of data proves to be arduous, the storage capacity accessible on local servers or workstations would be insufficient to satisfy the requirements of the user. Cloud computing has emerged as the predominant architecture for managing virtual servers and delivering storage capacity, with the aim of providing the highest possible level of storage capability. Cloud computing has shown its capacity to do computational tasks and oversee a substantial quantity of storage devices, enabling them to collaborate harmoniously via the application of orchestration principles. This is achieved through the use of the Distributed File System (DFS) and virtualization concepts. Several prominent corporations, like Google, Apple, and Microsoft, have established their own cloud storage platforms, such as Dropbox, Google Drive, and iCloud, respectively, in order to provide these services to its clientele. These firms are facilitating the development and use of cloud-based applications, which need significant storage capacity.

Multiple studies [2] have shown that the cloud storage service exhibits significant security weaknesses. Moreover, it is widely held in the technology realm that the Internet of Things (IoT) will seamlessly integrate into our daily routines via a diverse range of applications. The widespread use of many gadgets in the Internet of Things (IoT), such as smartphones and smartwatches, may be attributed to their high degree of mobility. These devices continuously collect data via various methods, which is contingent upon the cloud storage architecture allowed by different service providers. The need for cloud storage will see exponential growth in tandem with the increasing number of interconnected devices. The consequence of this will be an increase in network traffic and a decrease in response times, hence having a detrimental impact on the Quality of Experience (QoE) for end users. If the quality of experience (QoE) experiences a slow deterioration, it is likely that the performance of the Internet of Things (IoT) device will deteriorate, resulting in customer dissatisfaction. In 2012, Cisco introduced the concept of cloud computing as a means to address the many challenges and concerns associated with this computing paradigm. The primary operational idea of cloud computing is the virtualization and localization of remote cloud servers, with the aim of placing them in closer proximity to end users. This approach facilitates the transfer of these processes to a "edge" network. The use of cloud computing in the realm of data storage aims to address existing security concerns and improve the quality of experience (QoE). Current technological study suggests that cloud computing is widely recognized as a spatial-temporal network that prioritizes end-users, actively improving the quality of their experiences. Furthermore, cloud service providers on a worldwide scale continuously monitor this network.

Figure 1 is a schematic depicting the data transfers that occur inside a conventional cloud and cloud computing arrangement. Initially, customers will place their private data in the custody of cloud servers that are susceptible to security breaches and are geographically distant. The cloud service provider has the capability to access and examine a significant portion of data stored in the cloud. Meanwhile, malicious individuals sometimes referred to as hackers are compromising the security of data privacy. Several

studies have presented a diverse range of cryptographic methods [3] aimed at protecting confidential data saved in cloud environments. Prior studies have shown the efficacy of cryptographic methods in ensuring the confidentiality and integrity of data. However, it is obvious that their performance is impaired when it comes to detecting and preventing internal assaults. The use of Cloud nodes for the purpose of ensuring confidentiality has been extensively examined in several research papers [4].

However, it has been shown that despite the implementation of advanced algorithms such as the Reed Solomon code and the MD5 algorithms, instances of data loss still occur.

The primary contribution of this work is the development of a cloud-centric architecture that enhances the efficacy of data processing on cloud servers. This architecture addresses the significant issue of establishing data ownership on the cloud server. The primary objective of proven data possession is to ascertain if the data has been altered on the server by performing regular and efficient assessments of the accuracy of data maintenance subsequent to its delivery by the client.

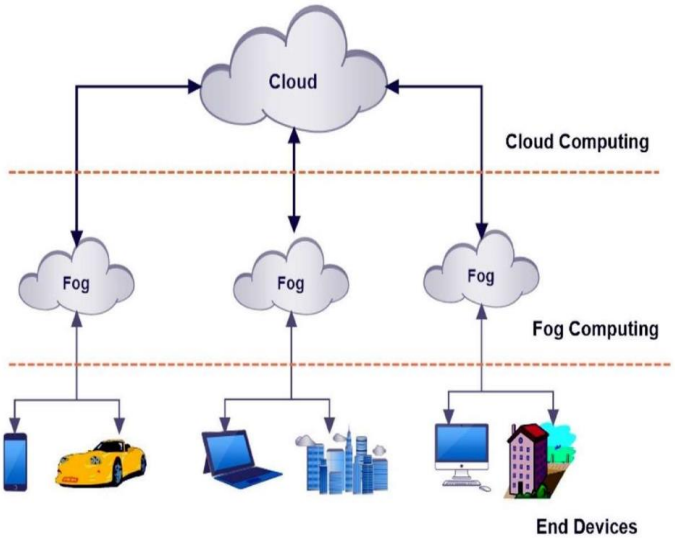


Figure 1. Generalized Cloud-Cloud Storage architecture

II. LITERATURE REVIEW

In recent years, there has been a notable increase in the attention given to the concept of distributed computing. Numerous persons have devoted their time and energy to improving its practicality and effectiveness, as seen by the scholarly works cited [7–11]. Mist processing is an emerging system innovation that aims to address the limitations of traditional distributed computing methodologies. This technique has garnered significant interest from both academic researchers and industry professionals. A limited number of studies address relevant security considerations and system architecture [12–15], which is expected given the novelty of the haze processing concept. Most of the current engineering literature focuses on elucidating the intricate framework and relevant performance metrics pertaining to Cisco's pioneering proposal for cloud computing architecture, given the nascent stage of research in the field of cloud computing [17]. This architecture has three distinct layers: a constructed cloud structure, an intermediary hazy layer situated between IoT devices and cloud servers, and a centralized network of IoT devices. In addition to discussing the technical intricacies of mist registration technology, the authors of References [13, 15] proposed an additional element for the distribution of remaining tasks that effectively manages power consumption, administrative idleness, CO2 emissions, and other relevant aspects. The challenges and security considerations associated with haze figuring have been thoroughly examined by the authors in References [7, 16, 12]. Alrawais et al. (2016) conducted a study that used fuzzy processing to conduct a contextual examination of the problem of confirmation disavowal in the setting of the Internet of Things (IoT).

Nevertheless, the existing research have not yet investigated the extent of the data review problem inside a loosely-defined distributed storage setting. To the best of our knowledge, the matter of information

review has not been explored in the context of Cloudcloud capacity, despite the inclusion of noteworthy results pertaining to distributed storage. Various methodologies for analyzing a distributed storage system have been investigated in previous studies (Refs. [9–16]) till recently. However, the existing review plans have several limitations, including a limitation on the total number of reviews and the possibility of data disclosure by a Third-Party Auditor (TPA). The article titled "Yang and 32" published in the Journal of Science and Technology at Tsinghua University, Volume 25, Issue 1, 2020, explores several aspects related to the topic. The article may be found on pages 28 to 43. Jia conducted an investigation into potential conspiracies pertaining to the capacity overhead and correspondence cost associated with the present study. The findings of their study revealed that the volume of data required to sustain the operations of the frameworks would become unmanageable over time, a recurring challenge often seen in projects of similar kind. Simultaneously, because to the need of face-to-face meetings between TPA and customers, these plans include significant costs associated with communication. The Merkle Hash Tree (MHT), which has gained significant popularity, has been used in following studies on auditing networked storage [9, 2, 4, 6]. These studies provide an improved review framework for distributed storage that offers lower metadata-related capacity overhead and correspondence overhead compared to previous approaches.

Liu et al. (2019, 2016) have proposed a multi-copy Merkle Hash Tree (MHT) approach for each data block, which includes a mechanism to alert the distributed storage administrator in the event of any replica failures. Consequently, the significance of confirmation communication is diminished, resulting in more flexibility of client information in the case of a capacity administration mistake. The Multi-Replica Dynamic Public Auditing (MuR-DPA) technique discussed in Reference [9] further amplifies the problem by necessitating the client to maintain and generate a distinct exceptional Merkle Hash Tree (MHT) for each replica, leading to substantial communication expenses. Nevertheless, the application of conventional review methodologies for distributed storage to haze distributed storage is challenging owing to the unique characteristics of haze distributed storage, such as its distinct operational protocols throughout consecutive working sessions. Due to the ability of a capacity administrator to reuse a processed hash value to verify the integrity of information, even if the content has been altered or deleted, all existing review plans based on Merkle Hash Trees (MHT) are vulnerable to replay attacks.

III.PROPOSED WORK

This section largely discusses an improved security approach for efficiently storing data at Cloud nodes, which are subject to frequent and rapid alterations. The primary purpose of this strategy is to effectively manage dynamically updated data at nodes that are centered on the Cloud, using established principles of data ownership. Multiple analyses have shown that the speed of demonstrating verifiable data ownership is associated with the quantity of disk input/output (I/O) operations, rather than the cryptographic analysis itself. The approach used in our study involves the utilization of B+ trees for data storage, specifically designed to accommodate composite keys. The suggested methodology offers enhanced value via the use of a composite key that amalgamates the index value and timestamp of the block. B+ trees are highly favored because to their self-balancing topologies, which enable efficient computation of results by effectively accepting dynamic changes originating from Cloud nodes. A crucial enhancement to the proposed methodology is the initial segmentation of the file intended for storage on the server into discrete blocks. These blocks may then be subjected to processing in order to generate metadata. The

metadata is then saved in the server throughout the indexing process and retained for a prolonged duration. The use of this methodology primarily capitalizes on the quantum channel.

3.1 Quantum channel for Cloud

The architecture of the Quantum Key Distribution (QKD) device involves two primary steps: the initialization of contact via the Quantum channel and the subsequent transmission of information through the conventional channel. The output of the gadget is analyzed based on the given parameters.

a. Secured Main Rate (Skr)

b. Error rate of Qubit (Qer)

The Protected Key Rate (Skr) is stated to be Skr = vBP..... (1)

Where 'vBP' is deemed from the source as pulses per second and 'BP' is the bit rate per pulse.

3.2 Analysis to calculate the Qubit error rate

The detections at Bob's end are used to estimate the mean observed signal per pulse and the bit rate per pulse with the help of protocol inherent efficiency Ni to first calculate the Qubit error rate.

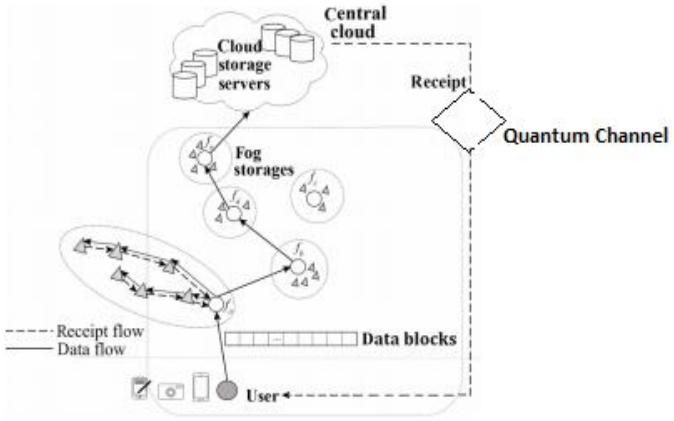


Figure 2: Proposed Quantum Cloud Storage Architecture

3.3.3 Computational Notations Utilized

Table 3.1: Notations used

Notation used	Attribute
F_s	Denotes the file to be stored at server in terms of blocks
O_d	Data Ownner
SOD	Outsourced data that is to be stored
T_s	Timestamp variable
CI_b	Counter variable to track Index over the file blocks
M_k, N_k	Pair of public key and private key
H(x)	Cryptographic hash fuction
P_{key}	Encryption mechanism used to encrypt the generated tags
P_{key}^{-1}	Decryption mechanism

The following elucidation presents the fundamental concept behind the operational mechanism. Prior to transmission to the server, the file Fs undergoes a process of fragmentation into many chunks. Once the preprocessing of these blocks is completed, metadata is generated and then sent to the server. In the preprocessing step, each block is assigned a randomly generated token, which serves the purpose of verifying the presence of the data on the server at a later stage. The encryption of the randomly produced tokens for each block is performed using the private key, denoted as Pkey. Subsequently, the encrypted tokens are sent and associated with the corresponding file. The file storage technique is implemented using a block-by-block approach, which is governed by the B+ tree storage mechanism. The block index variable serves as the fundamental basis for this strategy.

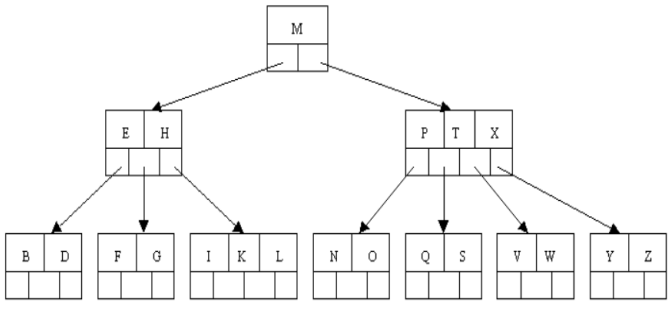


Figure 3. Block management using B+ Tree along

The terminal nodes of the B+ tree are the exclusive locations where data is stored. To locate the corresponding data block, the internal nodes of the

tree are linked to index values that are associated with a timestamp.

3.3.4 Initial setup

Initially the given file F_n is classified into 'n' blocks such as $F_n[1]$, $F_n[2]$, $F_n[3]$……. $F_n[n]$ with the time stamp of T_s for each file block where these blocks are stored in the Cloud server based on their index values.

Algorithm 3.1: Initial Setup

Begin

$CI_b = 0$

While (CI_b<= **n**)

Begin

Compute $L_i = H[T_{si}, F_n[i]]$

$L'_i = P_{key}(L_i)$

$CI_b = CI_b + 1$

Transmit to SOD ($F_n[i]$, T_{si}, L_i)

End

End

Utilizing the aforementioned procedure, a file undergoes first segmentation into distinct blocks, then stored on a Cloud server. The data is augmented with associated tags, which are systematically organized upon assignment.

3.3.5 Verification Module

During this phase, the data proprietor may verify the accuracy of the information stored on the Cloud server. This process involves the random selection of an index value, which is then sent to the Cloud server along with the corresponding time stamp value.

Algorithm 3.2: Verification Module

Begin

O_d transmits the data to SOD {j, T_{sj}}

SOD evaluates $A = H(T_{sj}, F_n[j])$ on locating the index values.

SOD reclaim L'_i and transmit back the verifier the testimony as (A, L_i)

Verifier validates the testimony as :

$L_i = P_{key}^{-1}(L'_i)$

If ($A = L_i$) then accept else reject

End

Upon reaching the Cloud server, the data undergoes a validation process against the specified tag. Subsequently, a newly generated tag, which is associated with the existing tag in the metadata, is sent. Through the implementation of dual-factor authentication, the process enables the verification of the authenticator.

4 Block Management at the Cloud layer

In this analysis, we will examine a B+ Tree of order n, where the tree has n key values and n+1 pointer keys. Our objective is to assess the efficiency of block operations inside this tree structure. The keys' values serve the purpose of identifying the pointer node Np that corresponds to a certain node inside the B+ tree. In the B+ tree data structure, each leaf node has a pointer that references the data page where all the leaf nodes are connected. This document presents a comprehensive analysis of the block operations, namely Insert, Delete, and Update.

4.1 Insertion

The process of inserting into the proposed system involves validating the entry each time the leaf node is accessed in the future. The fundamental concept behind the operation of this approach is the recursive insertion of a record by invoking the insertion algorithm on a designated child node. The process of recursion continues until it reaches the terminal node associated with its designated position, at which juncture it reverts back to the root node.

The proposed methodology involves designating the pointer node Z as the primary node, assigning a time stamp Ts, and introducing a new pointer as an initial null value until a split operation occurs. The variables Z and Ts denote the index and time stamp of a particular node, respectively.

Algorithm 3: Insertion of the record in to block (Pointer node (Z ,T_s), new)

Begin

Let N_p = Pointer to the node

If (pointer node is not a leaf node)

Locate s as if $Z_s = Z$ and ($Z<Z_{i+1}$)

Insert (Z ,T_s), new

Return

elseif (N_p include a space)

Put(new N_p)

New = null

Return

else

Split N_p in the process of creating new node

New = points the smallest value in the node

Create new node

4.2 Delete a record in the Cloud layer

The process of inserting into the proposed system involves validating the entry each time the leaf node is accessed in the future. The fundamental concept behind the operation of this approach is the recursive insertion of a record by invoking the insertion algorithm on a designated child node. The process of recursion continues until it reaches the terminal node associated with its designated position, at which juncture it reverts back to the root node.

The proposed methodology involves designating the pointer node Z as the primary node, assigning a time stamp Ts, and introducing a new pointer as an initial null value until a split operation occurs. The variables Z and Ts denote the index and time stamp of a particular node, respectively.

Algorithm 4: Delete a record in the block (Pointer node (Z ,T_s), old)

Begin
Let N_p = Pointer to the node
If (pointer node is not a leaf node)
Locate s as if $Z_s = Z$ and ($Z<Z_{i+1}$)
Delete (Z ,T_s), old
Return
If (old = = null)
Return
Else
Remove the old node from null
If(N_p has amd entry > n/2)
Set old to null
Endif
Endif
End

4.3 Handling dynamic updates

During the process of block updating, the tokens and the current block are simultaneously changed by the overwrite operation done on the current block. When a request for a specific block update is received by the Cloud server, it proceeds to search for the requested block. Once the server has located the block, it proceeds to update both the block itself and its associated token.

Algorithm 5: Update the block

Begin
Generate the request to SOD: update m,T_{sm}
At the SOD:
explore m,T_{sm}
transmit to O_d (F_n[m], L'$_m$, T_{sm})
At the O_d :
$F_n[m] = F_n[m]'$
$T_{sm} = T_{sm}'$
Evaluate a new tag $V_s = H(F_n[m], m , T_{sm})$
$V_s' = P_{key}(V_s)$

Transmit to SOD : (F_n[m], L'$_m$, T_{sm})

END

V. Performance analysis

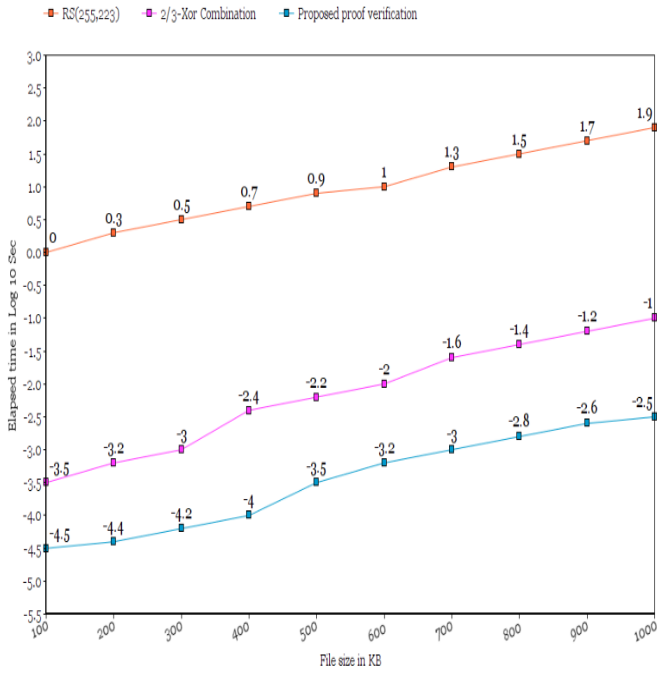
This section of the study involves a comparative analysis between the proposed approach and the previous research conducted by Wang et al. [13], using a series of test examinations. In order to establish a more intricate association, we conducted modifications to many conditional variables, such as block size and communication speed, among others. In the context of information security, the proposed approach involves the use of Xor-Combination, while Wang et al. employ Reed-Solomon coding as their chosen technique. In essence, the proposed approach and the method used by Wang et al. both utilize tailored iterations of the CRH (Customized Robust Hashing) and MMD (Maliciously Manipulated Detection) algorithms in order to discern hazardous modifications. Therefore, the suggested plot is contrasted and shown as follows. Table 1 displays the anticipated time complexity of the proposed setup and verification modules. This study compares the proposed mechanism with the RS coding technique and the Xor combination algorithm in terms of data processing, communication expenses, and update detection costs.

Table .1: Complexity analysis of the proposed algorithm

Phase	Space Complexity	Communication complexity	Time Complexity
Set-up Phase	O(n(k+m+p)	K+m+p bits	O(n)
Verification Phase	O(k+m)	M+q bits	O($\log_b$ n)

A. Data Processing

In the given context, a dataset consisting of blocks of varying sizes between 100KB and 1MB is systematically processed. The evaluation of a proof verification approach is conducted by using RS code and the 2/3-XoR combination method with parameters (255, 223). All of the algorithms maintain the integrity of the data partitions, ensuring that each iteration processes a single data block. Presented below is the comparative analysis of the execution times that has been generated.



V CONCLUSION

This paper presents a novel security protocol inside a cloud-centric cloud storage architecture, which serves the purpose of authenticating the storage of sensitive data. The present study introduces a complete computational security mechanism aimed at mitigating the issue of internal nodes. This mechanism achieves a reduction in computational cost via the use of efficient security algorithms. The proposed approach seeks to enhance the processing data efficiency of the Cloud server. Moreover, it effectively minimizes the use of network capacity while facilitating instantaneous updates of data. Moreover, the efficacy of this technique may be enhanced by taking into account the distinctive characteristics of the edge network.

VI References

[1] P. Mell and T. Grance, "The nist definition of cloud computing," 2010.

[2] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in Computer Science and Electronics Engineering (ICCSEE), International Conference on, vol. 1, pp. 647–651, IEEE, 2012.

[3] https://en.wikipedia.org/wiki/Clou_computing.

[4] L. A. Maghrabi, "The threats of data security over the cloud as perceived by experts and university students," in Computer Applications & Research (WSCAR), 2014 World Symposium on, pp. 1–6, IEEE, 2014.

[5] S. Debnath, S. Neog, S. C. Sahana, and B. Bhuyan, "Study on Secrecy and Privacy Preserving Approaches over Cloud Data Outsourcing," in International Conference on Computing and Communication System(I3CS), vol. II, pp. 135–142, 2015.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Infocom, 2010 proceedings ieee, pp. 1–9, IEEE, 2010.

[7] D. J. Abadi, "Data management in the cloud: Limitations and opportunities.," IEEE Data Eng. Bull., vol. 32, no. 1, pp. 3–12, 2009.

[8] A. Sharma et al., "Privacy and security issues in cloud computing," International Journal of Global Research in Computer Science (UGC Approved Journal), vol. 4, no. 9, pp. 15–17, 2013.

[9] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," IEEE transactions on parallel and distributed systems, vol. 25, no. 2, pp. 384–394, 2014.

[10] S. Debnath and B. Bhuyan, "An Expressive Access Control Mechanism with user Revocation for Cloud Data Outsourcing," in Proceedings of the International Conference on Electrical, Electronics,Computers, Communication, Machanical and Computing(EECCMC), pp. 250–256, IEEE, 2018.

[11] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," IEEE transactions on Services Computing, vol. 5, no. 2, pp. 220–232, 2012.

[12] Y. A. Younis, K. Kifayat, and M. Merabti, "An access control model for cloud computing," Journal of Information Security and Applications, vol. 19, no. 1, pp. 45–60, 2014.

[13] D. Servos and S. L. Osborn, "Current research and open problems in attributebased access control," ACM Computing Surveys (CSUR), vol. 49, no. 4, p. 65, 2017.

[14] D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role-based access control," Computer, vol. 43, no. 6, pp. 79–81, 2010.

[15] J. Hur, "Improving security and efficiency in attribute-based data sharing," IEEE transactions on knowledge and data engineering, vol. 25, no. 10, pp. 2271–2282, 2013.

[16] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," IEEE transactions on parallel and distributed systems, vol. 25, no. 7, pp. 1735–1744, 2014.

[17] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. A. Au, "Improving privacy and security in decentralized ciphertext-policy attribute-based encryption," IEEE Transactions on Information Forensics and Security, vol. 10, no. 3, pp. 665–678, 2015.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes," in Workshop on the theory and application of cryptographic techniques, pp. 47–53, Springer, 1984.

[19] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Annual international cryptology conference, pp. 213–229, Springer, 2001.

[20] A. Sahai, B. Waters, et al., "Fuzzy identity-based encryption.," in Eurocrypt, vol. 3494, pp. 457–473, Springer, 2005.

[21] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," Journal of Computer Security, vol. 18, no. 5, pp. 799–837, 2010.

[22] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization.," in Public Key Cryptography, vol. 6571, pp. 53–70, Springer, 2011.

[23] M. S. Kiraz and O. Uzunkol, "Still wrong use of pairings in cryptography," arXiv preprint arXiv:1603.02826, 2016.

[24] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Security and Privacy, 2007. SP'07. IEEE Symposium on, pp. 321–334, IEEE, 2007.

[25] K. E. Fu, Group sharing and random access in cryptographic storage file systems. PhD thesis, Massachusetts Institute of Technology, 1999.