Indonesian Journal of Electrical Engineering and Computer Science
Vol. 13, No. 2, February 2019, pp. 626~633
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v13.i2.pp626-633

Checking integrity of data and recovery in the cloud environment

Neha Narayan Kulkarni, Shitalkumar A. Jain
Department of Computer Engineering, MIT Academy of Engineering, Maharashtra, India

Article Info

Article history:

Received Jun 11, 2018
Revised Nov 22, 2018
Accepted Dec 3, 2018

Keywords:

Cloud server
Data integrity
Data recovery
Remote backup server

ABSTRACT

Cloud computing provides many services to access them dynamically over the internet as per the user's demand. The data is growing in tremendous amount and it should be managed correctly where storage service proves efficient. The Data stored online can be hacked by the third party so to secure this data verifying integrity is essential. Because of the human error or natural disaster, the data may get deleted from the cloud servers. Therefore, there is a requirement of improving the existing algorithm to recover data efficiently. Many algorithms are proposed, but they lack in efficiency, computational cost, and reliability. The proposed system in this paper is having the feature of verifying data integrity using Identity Based Remote Data Integrity Checking and recovery using the XOR operation. When the data is unavailable to the user, the proposed system provides flexibility for the user to regain data from the remote server.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Neha Narayan Kulkarni,
Department of Computer Engineering,
MIT Academy of Engineering,
Pune, 412105, Maharashtra, India.
Email: nehakulkarni281@gmail.com

1. INTRODUCTION

The Cloud computing is the next phase of the internet evolution which would provide computational power, storage service, infrastructure and platform for the user to maintain the data. The cloud technology is a technology which has evolved, widely used and moving ahead to provide the storage space online. Cloud computing model distributes over the virtualized pool of shared resources. The user can demand as well as release the resources as per the use and requirement [1]. Cloud computing brings out many of the benefits to the user. For example, 1) Cloud user can reduce the storage infrastructure cost. 2) The cloud user can request service and pay as per use. 3) The user can access data comfortably independent of place. There are many potential services that the cloud provides the user, and they could be managed virtually. The concept of the cloud evolved from the fundamentals of the network.

The services provided by the cloud plays important role because the massive amount of data is stored online and many applications are deployed on the cloud providing services to the user. Cloud computing service works with some service models as follow:

a) SaaS (Software as a Service): Cloud model that provides access to the software through the web whereas the user does not has control over the infrastructure. Example: Google Docs, Microsoft Office, DropBox.
b) PaaS (Platform as a Service): The Cloud Service Provider (CSP) provides the necessary hardware and platform for the developers [2]. Example: Flexiscale, Gizmox.
c) IaaS (Infrastructure as a Service): This model allows the user to access virtual and physical environment. The software, application, operating system is manageable to the user. Example: Amazon Web Service, Cisco cloud verse. The deployment models for cloud computing are: 1) Public cloud: Open access to network publicly. 2) Private cloud: Open to the authorized end users. 3) Community cloud: This cloud is

accessible to a category of the users who satisfies the policy for authorization. 4) Hybrid cloud: It integrates the features of public, private, community cloud [3, 4]. Cloud computing provides the pay-as-per-use service. The data stored online can be damaged by the natural disaster or human mistakes. There are existing solutions as well as the enhancements are in progress for the recovery in natural disasters, some are discussed in the paper [5]. Here, we have discussed the human perspective and solution for the recovery of information because of the human faults.

The essential parameters considered for checking the integrity of the data are validity, consistency, and accuracy. The integrity of the stored data can get affected because of malicious attacks or unconsciousness errors of the individual. The data integrity is verified when the data owner gets the proof that its outsourced data is intact accurately in the cloud. Cloud service provider does not provide the data integrity checking explicitly for the users [6]. They provide MD5 tag value, but this is insufficient as the tag change when the data is modified. The data owner, an organization needs to trust the third party service or technology to store data securely using various cloud services. However, the problem arises when the data owner is unable to control or audit data stored outside of the trusted zone of the data owner. This independent service creates the problem of data leakage or losing privacy of the owner. The data of the client is always susceptible and confidential which should not be disclosed to the unauthorized users. There are many methods proposed for securing sensitive data such as encryption of data before uploading but this alone is insufficient to guarantee the data security [7]. So, in the proposed system apart from the data owner, the cloud server we have involved the admin who acts as a third-party auditor to build trust between them. Still, there is a possibility of data leakage so we have used the model of challenge and response for the Admin. Also, we have added the security parameter to protect the data against the Admin and the cloud server.

The volume of data in the cloud is increasing exponentially in structured as well as in an unstructured way, and hence, there is a need for efficient data recovery from historical data. Data stored in the remote data centers is also increasing which can result in severe issues related to data such as breaching and data loss. However, there need the efficient and compatible method for the data recovery because the data may get deleted by the human errors or modified by the unauthorized user [8, 19]. In the financial world, if the server gets crashed or data corrupts then it directly impacts causing economic loss. There are many proposed algorithm for data recovery, but they fail because of inefficiency and incompatibility. Here, we have discussed an algorithm using XOR operation and backup server for recovery of the data.

1.1. Literature Survey

We have discussed various techniques for checking integrity, Proof of Retrievability, and recovery of the data in this section of literature survey. The client's data is sensitive so, we need to ensure the security of data. It is crucial to verify access control for managing the data. There is various access controls such as User-Based, Role-Based, and Attribute-Based. The system contains the record of the user with their allocated permission and authorization to manage data in the User-Based control. In Role-Based, the users are categorized using their assigned roles. In Attribute Based the access policy is defined and the user obeying the policy can manage the data.

The author of the paper [9] has discussed the RSA based hash and RSA based tag. RSA algorithm works by partitioning the data into the blocks and generating the proof. The author has also discussed the MD5 algorithm. The MD5 function produces the message digest when the compacted file is provided as input to the function. The produced output is coded and added to the original file. This appended file is used by MD5 to produce a hash of the file. The comparison of the available record and the generated hash is undertaken.

The encryption algorithm MAC gives message authentication to check integrity. MAC accepts various size blocks and a cryptographic key which helps in generating the authenticated code. The owner possesses the secret key and message which is used to verify the integrity.

In paper [10], the author has shown the contribution to ID-CDIC which addresses the key management issue and complex certificate management. The author proposed the protocol based on RSA signature and supported variable size file block and public auditing.

The paper [11] discusses the ID-DPDP protocol. The author has contributed to the ID-DPDP which verifies the data integrity without retrieving the whole file. This protocol eliminated the complex certificate management and supported multi-cloud scenario. It uses BLS signature scheme for security.

The author of the paper [12] provides detail study of the ID-PUIC scheme. ID-PUIC verifies whether the data is uploaded and intact correctly. It works with the hard code of Diffie Hellman and bilinear pairing.

In paper [13] the author has shown the implementation of the Proof of Storage. POS is the powerful cryptographic method for outsourced data. Existing POS work with single user environment.The author has introduced new features deduplication for the cross-user using Homomorphic Authentication Tree it solves private key generation problem.

The paper [14] provides the detailed study of the Remote Data Possession technique. This method verifies the privacy of the authenticator for the cloud. It supports block-less verifications.

In paper [15] the author has discussed the ECAL method. This method is a dynamic POR which stores encoded log, and these logs are garbled, to secure them from the server.

In paper [16] the author has shown the contribution to the Message Lock POR. It addresses the problem of data retrievability and the cross-user deduplication. The author has proposed some modification in the existing POR which lack the deduplication.

In paper [17] the author has provided the discussion of different methods related to the Proof of Retrievability. The Basic Static method encrypts few bits of data, so storage reduces which minimize the load on the client. This scheme reduces the proof size and network bandwidth utilization. It limits the capacity and computational overhead on the server as well as the client side. High Availability and Integrity Layer (HAIL) checks correctness and provides high availability of storage. It avoids the server redundancy. The verification of the file is dependent on the verifier who interacts with the server.

The paper [18], describes different algorithms developed for recovery of the data such as Cold back-up, Hot back-up, PCS, HSDRT, Linux Box, ERGOT. Comparing PCS with the other methods, we can observe that PCS is manageable, facile and reliable for recovery. PCS is executed depending on congruence service. HSDRT is useful for the mobile client. This method needs excessive execution cost. This scheme is not resistant to duplication of the data. Another method is the Efficient Routing Grounded on Taxonomy (ERGOT). The execution of ERGOT is based on a grammatical evaluation. ERGOT execute on the basis of service illustration and invocation for data retrieval [19]. Linux Box model provides the simple backup method and low-cost implementation. It provides the efficiency for migration of the client from one cloud to other. When the storage request to sync and backup it works for full storage rather than the single file this is the drawback of this method.

The execution of the cold backup strategy starts if the functioning of the system is ceased. If the system is operating correctly, this algorithm is not executed. The Hot backup algorithm works in an active network using a substitution policy and unconstrained procedure for a service configuration.

In paper [20], the author has improved the operational complexity using Efficient integrity verification scheme (EIVS) and Safety integrity verification scheme (SIVS). These schemes are built using the concept of Schnorr Signature. This proposed model reduces the cost and the complexity.

1.2. Contribution of our work

The summary of contributions in this paper is as follow:

a) We have proposed a system by integrating feature of verifying the integrity of the data using ID-RDIC and proposed the algorithm for recovery of the data using XOR operation on the cloud.
b) We prove the increased security for the outsourced data in the cloud, and the data can be retrieved in the original format.
c) We have also used Elliptical Curve Digital Signature Algorithm considering the security perspective towards data storage.
d) We show that this system is efficient, secure and practical to implement.

2. PROPOSED SYSTEM

The proposed system is implemented with the hardware specifications such as Corei3 Processor, 100GB Hard Disk, 4GB RAM and the software specification configuration required are Windows 7, JDK1.8, MySQL, and Apache Tomcat 8.

The approach of the existing model is considered for checking the integrity of the data and in the proposed system we have contributed by adding the security parameter for checking integrity also added the feature of recovery with the log maintenance in the system. The significant contribution related to this project is a recovery of data and verifying the integrity. The integrity is verified using the Remote Data Integrity Checking using Elliptical Curve Cryptography. The data recovery works with the central cloud server and remote backup server if some data is tampered. Recovery is executed using the XOR operation along with hashing for security function. The system consists of five different network entities known as Data Owner, User, Admin, Central Cloud Server, and Backup Cloud Server. Central Cloud server stores the outsourced data of the client. Backup server stores the replica of the file maintained on the server. Data Owner can operate using the function to upload, modify, delete, download a file and view the logs. The user can perform only the operation of downloading a file. Admin role is to act as auditor to communicate with the server. Data owner uploads the file on the Cloud at the same time file is stored on the backup server also. The uploaded data is encrypted using the XOR operation and stored as an array. Data owner request the Admin to verify the Data Integrity for the file. Admin then produces a challenge and send it to the server. The integrity proof is produced

by the server and sent to the Admin. When the Admin gets the acknowledgment, the Admin verifies the proof to acknowledge the data owner with the status of the data. The data owner receives a status whether the data integrity is verified or failed. When the data owner tries to modify the file the data owner receives a security question which is stored by the owner while outsourcing the file. If the answer matches, the owner can modify else data owner is not able to modify. When the user wants to download a file, the user has a search option where the user can search using the tags and request the data owner for the file access. When the data owner receives a request from the user, the owner has authority to accept or deny the access. If the owner affirms the request, then the user can download the file from the cloud server. If the file corrupts or get modifies then the data integrity fails, so there we have implemented the algorithm for the data recovery. At the initial stage, we have metadata stored in the file using the XOR operation so from the remote backup server we can fetch the file and cross XOR to retain the original file. For every action in the system, we maintain the logs. The data owner and the user have authority to view the logs. The Admin has the authority to view the logs. Even the Admin can view the registered user with their role in the system. Figure 1. explains the flow of proposed framework.

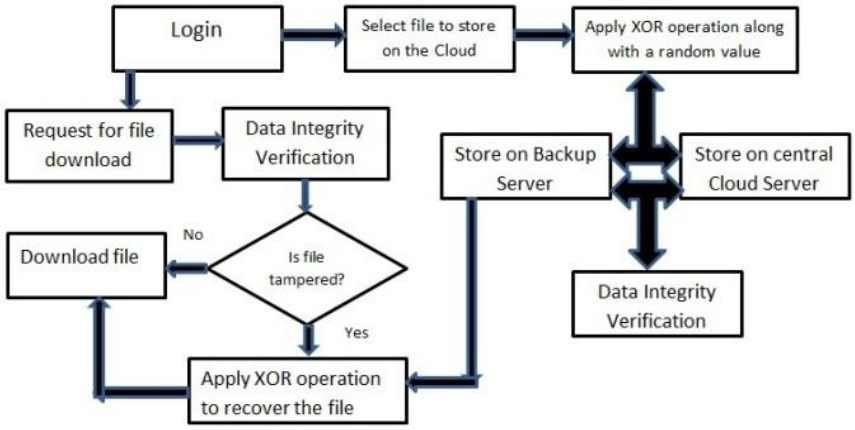

Figure1. Proposed Framework

3. RESEARCH METHOD

3.1. Identity Based Remote Data Integrity Checking

The ID-RDIC scheme enables the client to check that the data is correctly intact with the server. The existing algorithms have complicated key management procedure whereas ID-RDIC minimizes the complexity and the key maintenance cost by using the homomorphic key model. ID-RDIC introduced the concept of the trusted auditor to include the skillful object who can check the integrity. Once the data is outsourced to the server, using Setup algorithm generates the keys. Using the private key and the client id we extract the features of the file [21]. In tag generation algorithm, the pre-processing operations are executed using tokenization concept and calculate term frequency for the file and store it in the database. The Admin sends a challenge with the file name and the client identity if the data owner request for the integrity verification. As the cloud server receives a challenge from the TPA, it generates the proof using the client id, challenge, file and the tag. This proof gets forwarded to the TPA for verification. When TPA receives the proof, it uses proof, client id, challenge, file to verify the proof. If the proof verifies, then the TPA sends the response to the owner with the data integrity status.

Procedure for ID-RDIC:
Input: File "F" is uploaded to the Cloud server
Output: Data Integrity Status
Procedure:
Step 1: Generate key pair and Setup curve parameters
Step 2: Generate tags by reading the file
Step 3: Generate the proof for the file
Step 4: Verify the proof status
Step 5: Send the proof verification status to the data owner

3.2. Elliptical Curve Digital Signature Algorithm (ECDSA)

The security of the data remains unaffected by ECDSA, but it produces the digital signature and verifies the signature. ECDSA is different from AES and another algorithm as it does not encrypt the data but it provides the security, or it protects the data by making sure whether the data tamper. The digital signature verification in ECDSA is done using the public key, so it is safe and secure. The public key does not produce the private key. ECDSA uses 160 bits to make it more secure [22]. ECDSA uses SHA-1 cryptographic hash function to generate a hash for the stored file which is 20 bytes. ECDSA uses the hash value of the file to sign. SHA-1 is the random algorithm which has very less chance of hash collision. For ECDSA, NIST (National Institute of Standards and Technology) and SECG (Standards of Efficient Cryptographic Group) has stated the standard curve parameters which are more secure and efficient. The security provided by ECDSA for key size 160 bits is similar to the RSA with key size 1024 bits. Thus ECDSA is preferred. ECDSA security lies with the trap door principle as we can use only the point of addition and multiplication.

Procedure for ECDSA:
Input: File stored on the Cloud ‘F.'
Output: Verification of the digital signature and data integrity.
Procedure:
Step 1: Using the NIST and SECG defined equation the system draw the curve.
Step 2: The point lying on the curve is selected referred as "Point of Origin" ‘G.'
Step 3: Generate a random number referred as private key ‘A.'
Step 4: Using the point of origin and private key, the public key is generated ‘Q.
Step 5: Generate digital signature using hash value of the file ‘H' and the private key ‘A.'
Step 6: Verify the digital signature using ‘S' and the public key ‘Q.'
If the coordinates of P and R match then the digital signature is verified.

3.3. Proposed Algorithm

We have proposed an algorithm for data recovery. Many algorithms are developed but some of them lack the feature of accuracy, cost-benefit, and compatibility. In the proposed algorithm we have considered the accuracy and cost beneficial model for the user. The entities involved in the proposed algorithm are Cloud storage, Backup server, and the client. The proposed algorithm enhances the data privacy for the remotely stored data. Here, we have used XOR operation on the data and a random variable which gets stored in an encrypted array pattern.

Procedure for the proposed algorithm:
Input: File “F” is uploaded at Cloud storage “cs."
Output: If the servers crashes or file "F" is deleted then recover the file “F."
Procedure:
Step 1: Assign the client id “client_id" to the data owner
Step 2: Produce a value “s” for each user
s =m’ XOR client_id
Store this at remote server “rs."
Step 3: If the user modifies the file "F” Then we need to some metadata file “f1."
f1=h1 XOR si
Store this f1 at both servers in integer array pattern
Step 4: IF the file “F" deletes from "cs” Then for recovery of file “F” is as:
h1=f1 XOR si
Return this h1 to search the file F
Download the requested data to the client frame

4. RESULTS ANALYSIS

The following results are derived from the proposed system for various files for the result analysis. The results are derived by testing the proposed system for different files ranging from 7KB and further. Figure 2. shows the proposed system results of time required for tag generation and digital signature verification. Tag generation time and signature verification time increases according to the file size. Figure 3 shows the proposed system results of time required for data integrity verification and file recovery time. We can infer that time for data integrity verification is minimal compared to the file size.

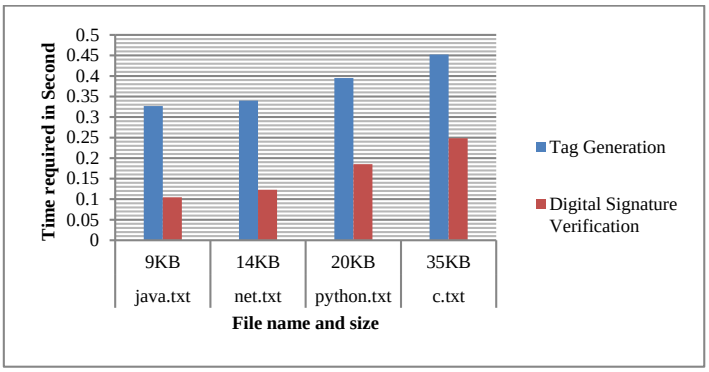

Figure 2. Time required for tag generation and signature verification

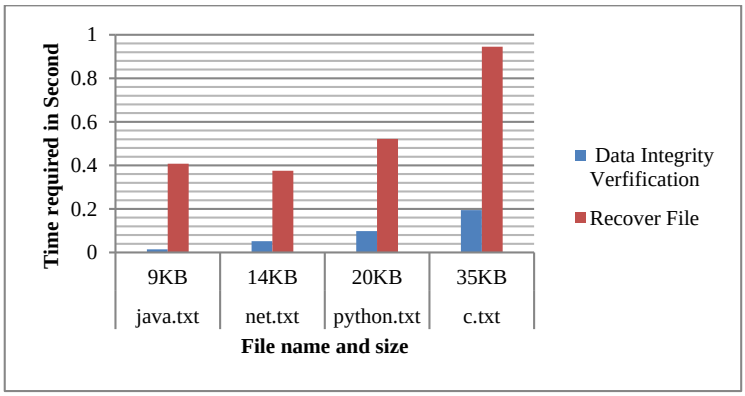

Figure 3. Time required for data integrity and file recovery

Figure 4 demonstrates the comparison of existing system and proposed system results. From all above results we can observe that the time required for the tag generation differ with the change in the text pattern content of the file and the average length of the word and the sentence. Time required for challenge parameter depends on the same parameters of the file. Proof Generation and Proof Verification executes by digital signature generation and verification with respect to the file. We can see the time cost required for processing over 1 MB file is reduced which contributes to increase the performance of the system.

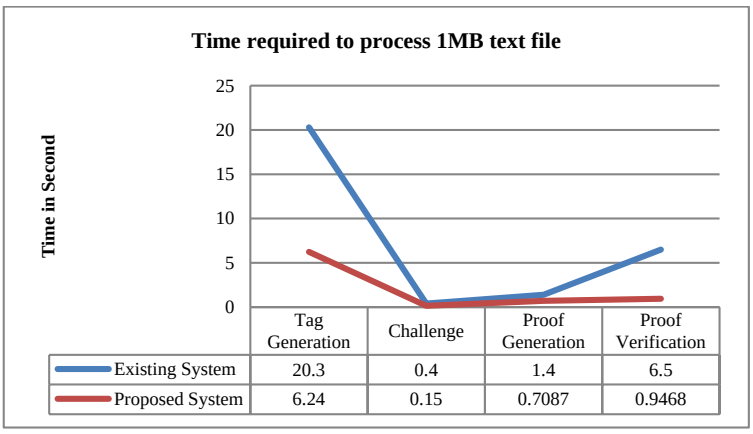

Figure 4. Comparison of Time required for 1MB text file in existing and proposed system

Table 1 shows the outcomes related to the security properties of the system. It could be observed that the proposed system have improved the feature of recovery of the data and the availability of the data in the existing system.

Table 1. Comparison of the security properties

Security properties	Existing System	Proposed System
Authentication	Yes	Yes
Confidentiality	Yes	Yes
Security	Yes	Yes
Integrity	Yes	Yes
Recovery	No	Yes
Availability	No	Yes

Table 2 shows the result of the proposed system for data recovery from the cloud storage. From the result it is demonstrated that the accuracy of the recovery of the text file is 100% without loss of the data after recovering the file. This makes system reliable, accurate and efficient for recovery of the data.

Table 2. Originality of the recovered file

File Name	Original File Size(KB)	Recovered File Size(KB)
education.txt	46	46
bigdata.txt	218	218
boat.txt	605	605
project.txt	1025	1025
recovery.txt	2513	2513

Analysing all the above results it is proved that the proposed system provides accuracy, increased performance, efficient and reduced cost for implementation with the objective of verifying data integrity and recovering data from the cloud. As the proposed system have backup server this provides high availability for data maintained on the cloud.

5. CONCLUSION

The proposed system has the feature of verifying the integrity and recovering data from secure cloud storage with the remote backup cloud server. We have integrated the data recovery feature using XOR operation along with the data integrity using Identity Based Remote Data Integrity Checking and Elliptical Curve Digital Signature Algorithm. We can observe that the time required for the operation is small whereas the time increases with the increase in the file size. The proposed system satisfies the properties of data privacy, confidentiality, and integrity. Considering the security model we have demonstrated that the above designed system is efficient, comfortable and practical to implement.

Though the proposed system integrates many features, a perspective regarding network issue can be considered separately with various possibilities. While designing this system, the use case considers the file always goes to a destination without loss. Hence, we can consider the network issues for further future work in this system. Also, the high availability module can be integrated with the system using the concept of clustering to provide the continuous service access to the clients.

REFERENCES

[1] A. Razaque and S. S. Rizvi. Privacy preserving model: a new scheme for auditing cloud stakeholders. *Journal of Cloud Computing*. 2017; 6(1):7.
[2] M. S. Kiraz. A comprehensive meta-analysis of cryptographic security mechanisms for cloud computing. *Journal of Ambient Intelligence and Humanized Computing*. 2016; 7(5):731–760.
[3] A. Kumar. World of cloud computing & security. *International Journal of Cloud Computing and Services Science*. 2012; 1(2):53.
[4] P. S. Challagidad, A. S. Dalawai, and M. N. Birje. Efficient and Reliable Data Recovery Technique in Cloud Computing. *Internet of Things and Cloud Computing. Special Issue: Advances in Cloud and Internet of Things*. 2017; 5(1):13-18.
[5]G. Li, Q. Zhang, W. Li, and Z. Feng. The design and verification of disaster recovery strategies in cloud disaster recovery center. *Indonesian Journal of Electrical Engineering and Computer Science*. 2013; 11(10):6179–6186.

[6] K. N. Sevis and E. Seker. Survey on data integrity in cloud, in Cyber Security and Cloud Computing (CSCloud). *2016 IEEE 3rd International Conference on. IEEE*. 2016; pp. 167–171.
[7] N. Garg and S. Bawa. Comparative analysis of cloud data integrity auditing protocols. *Journal of Network and Computer Applications*. 2016; 66:17–32.
[8] S. G. Worku, Z. Ting, and Q. Zhi-Guang. Survey on cloud data integrity proof technique, in Information Security (Asia JCIS). *2012 Seventh Asia Joint Conference on. IEEE*. 2012; pp. 85–91.
[9] C. V. Desai and G. B. Jethava. *Survey on data integrity checking techniques in cloud data storage. International Journal*. 2014; 4(12).
[10] Y. Yu, L. Xue, M. H. Au, W. Susilo, J. Ni, Y. Zhang, A. V. Vasilakos, and J. Shen. Cloud data integrity checking with an identity-based auditing mechanism from rsa. *Future Generation Computer Systems*. 2016; 62:85–91.
[11] H. Liu, Y. Mu, J. Zhao, C. Xu, H. Wang, L. Chen, and Y. Yu. Identity-based provable data possession revisited: Security analysis and generic construction. *Computer Standards & Interfaces*. 2017; 54:10–19.
[12] H. Wang, D. He, and S. Tang. Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud. *IEEE Transactions on Information Forensics and Security*. 2016; 11(6):1165–1176.
[13] K. He, J. Chen, R. Du, Q. Wu, G. Xue, and X. Zhang. Deypos: Deduplicatable dynamic proof of storage for multi-user environments. *IEEE Transactions on Computers*. 2016; 65(12):3631–3645.
[14] W. Shen, G. Yang, J. Yu, H. Zhang, F. Kong, and R. Hao. Remote data possession checking with privacy-preserving authenticators for cloud storage. *Future Generation Computer Systems*. 2017; 76:136–145.
[15] M. Mohammad Etemad and A. K¨up¸c¨u. Generic efficient dynamic proofs of Retrievability, *in Proceedings of the 2016 ACM on Cloud Computing Security Workshop. ACM*, 2016, pp. 85–96.
[16] D. Vasilopoulos, M. Onen, K. Elkhiyaoui, and R. Molva. Message-locked proofs of retrievability with secure deduplication, *in Proceedings of the 2016 ACM on Cloud Computing Security Workshop. ACM*, 2016, pp. 73–83.
[17] R. A. Hegde and M. Prakash. *A survey on proof of retrievability and its techniques.*
[18] K. Pophale, P. Patil, R. Shelake, and S. Sapkal. Seed block algorithm: Remote smart data-backup technique for cloud computing. *International Journal of Advanced Research in Computer and Communication Engineering*. 2015; 4(3).
[19] M. Raje and D. Mukhopadhyay. Algorithm for back-up and recovery of data stored on cloud along with authentication of the user, *in Information Technology (ICIT), 2015 International Conference on. IEEE*. 2015, pp. 175–180.
[20] H. Deng, S. Xiuli, and T. *Jingsong. A double efficient integrity verification scheme to cloud storage data. TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(9):7007-7013.
[21] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. *IEEE Transactions on Information Forensics and Security*. 2017; 12(4):767–778.
[22] A. Khalique, K. Singh, and S. Sood. Implementation of elliptic curve digital signature algorithm. *International journal of computer applications*. 2010; 2(2):21–27.