International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 5, May 2020 ||

Dynamic Identity-Based Deterministic Multi- Copy Information Custody in Multi-Cloud Repository

Dr. Ganesh Kumar .K[1], Poovizhi .V.K[2], Sakunthala.Pl[3], Sugashini.K[4]

Assistant Professor, Department of Information Technology, Velalar College of Engineering and Technology, Erode, Tamil Nadu, India[1*]

B. Tech Final Year Students, Department of Information Technology, Velalar College of Engineering and Technology, Erode, Tamil Nadu, India[2.3.4]

ABSTRACT: Cloud data storage seems to be commonplace for data to be stored in the cloud and shared across multiple users. Unfortunately, the integrity of cloud data is to be doubted due to the existence of failures related to hardware or software and errors caused by people. Several methods are designed to allow both data owners and public verifiers to efficiently audit the integrity of cloud data without getting the entire data from the cloud server. Public auditing on the integrity of shared data with these existing methods will reveal confidential information. Cloud Computing is being considered as a popular scenario for storing the data in the online repository which will continue in the future that leads the application software and databases to be stored in large data centers which are centralized, Due to this feature, the management of data and services or not fully trusted. This paper studies the various issue of ensuring the integrity of data storage in Cloud Computing. This study considers the method of allowing a threshold proxy re-encryption, in the name of the cloud client, the integrity of the dynamic data stored in the cloud can be verified. This study overcomes the remote data integrity that often lacks the support of public Audit ability or dynamic data operations.

KEYWORDS: Cloud Computing, Dataintegrity, Homomorphic Encryption, Proxyre-encryption,Security.

I.INTRODUCTION

Distributed computing has been considered as the creation of information development plan for endeavors, due to its broad summary of liking in the IT history. Which provides self-advantage for better performance, reliability, scalability and inherent distribution[1]. As a great development with huge consequences, distributed computing is altering the specific method of how associations use the information for advancement. Which works on scalability, high availability and multiple architectures[2]. Distributed memory systems use multiple computers to solve a common problem, with distributed computation among nodes and passing messages to communicate between the nodes. While distributed computing makes these compensations more attractive than some other time in the current period, it also creates new testing security risks[3,7] to customers due to outsourced data. As organization providers (CSP) also use administrative components partly, data outsourcing is truly surrendering customer's control more than the data's fate[14-16]. As an issue of first importance, Despite the way that the underneath structures are altogether which makes it more powerful and trustworthy[13] than individual devices, they are still having both inside and outside risks for data respectability.

II.RELATED WORKS

The system to increase the availability and durability of the outsourced data, many customers store multiple copies on multiple cloud servers. To guarantee the integrity of multi-copies, some protocols such as provable data possession (PDP) for multi-copy are presented. However, most of the previous PDP protocols which consider all copies to be stored on only one cloud storage server. Furthermore, many PDP protocols depend on the technique called public key infrastructure (PKI), which suffers many types of security vulnerabilities and also brings heavy communicational and computational cost. It provides advantages to increase security and efficiency, it also

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 5, May 2020 ||

provides a PDP scheme which is a novel identity based on multi-copy on multiple cloud storage servers. In this scheme, all the multi-copies are delivered to different cloud storage servers, which work cooperatively to store the customer's data. By the homomorphic verifiable tags, the integrity of all copies can be checked simultaneously. The system model and security model of this scheme are provided in the paper. The security for this scheme is proved based on the Computation Diffie-Hellman (CDH) hard problem. Analysis and experimental evaluation show that the scheme is efficient and practical. The proposed scheme is proved by securing the CDH assumption. This scheme is designed to verify the integrity of multiple copies on multiple cloud servers within one challenge-response interaction. Moreover, the structural advantage of identity-based cryptography makes this scheme more efficient and secure due to removing the certificate management problem in PKI. Also, this scheme supports the feature of public verification. The main disadvantage is batch verification is not implemented and it does not support transactional data.

III.MATERIALS AND METHODS

In this paper, a proposal has been made for a distributed scheme that is effective and flexible with explicit dynamic data support to ensure the correctness of user data in the cloud. This proposal relies on MD5 correcting code in the file distribution preparation to provide integrity and guarantee the data dependability.MD5 is a cryptographic hash function whose main purpose is to verify that a file has been unaltered. Instead of confirming that two sets of data are similar by comparing the raw data, MD5 does this by producing a checksum on both sets and then comparing the checksums to verify that they're the same. It is perfectly acceptable to use it for standard file verification The Above method reduces the communication and storage overhead hugely when compared to replication-based file distribution techniques. By utilizing the homomorphism token with distributed verification of MD5-coded data, this scheme achieves the data error localization as well as storage correctness insurance.

It is used to convert the data into ciphertext that can be analyzed and worked with as if it were still in its original form. It allows the user to store encrypted data in the public cloud and takes advantage of the cloud provider's analytic service. This also uses a checksum algorithm whose value is used to verify the integrity of a file that checks the validity of the data. It compares the two sets of data and makes sure they are the same. Using the homomorphism token with verification of MD5-coded data that is distributed, this scheme achieves the storage correctness and localization of data error. To allow not only a data owner itself but also a public verifier to perform integrity checking effectively without downloading the entire data from the cloud, which is referred to as public auditing with ring signatures, a verifier is convinced only if that a signature is computed using one of group member's private keys, but the verifier is not able to determine which one. The public verifier knows each block in shared data is either signed by Alice or Bob, because it needs both user's public keys to verify the correctness of the entire shared data.

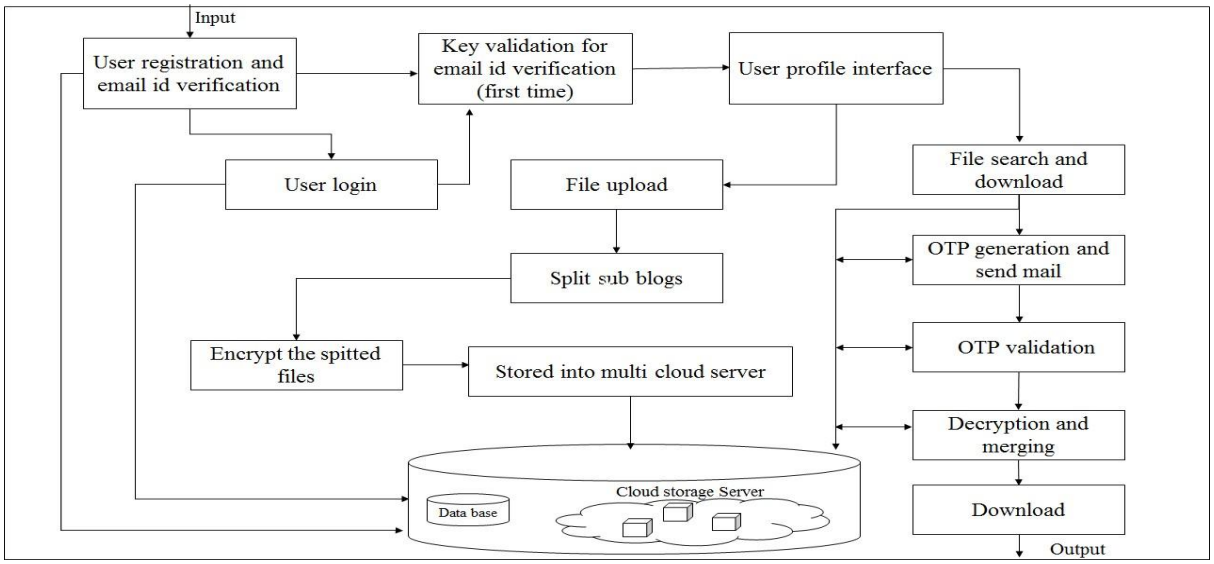


Figure 1. System Architecture

Initially users need to register with the verified email id. After registration user has to login with the registered

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 5, May 2020 ||

credentials. Now user can upload the file which is split into sub-blogs and encrypted and gets stored in the file server. Another user can search the file and download it. Before downloading OTP will be generated and will be sent in a mail. Then after entering the OTP it will be validated. If validation is successful then it will proceed to decryption and merging after which file download will happen. As per this proposal the file uploaded to a cloud uses different algorithms that split a file into 3 or more parts and each part is stored in 3 or more servers in the public cloud along with their keywords causing the data to be secured. As users other than the owner of the file cannot see or edit the data as it is split in multiple servers. Keywords will be generated for different parts of a file which will be used as an initial authentication factor for the file. The homomorphic algorithm which is used in this proposal will be used which will check the keywords and decrypt the data after entering the OTP generated and sent to the Owner of the file has been
authenticated. Once authenticated the different parts will be decrypted and Merged together and then the file will be sent for download. It can also share files present in the cloud with multiple people. But even when the shared person edits the file, its contents will not be directly edited rather contents will be updated virtually and the owner of the file can see the updated contents. Only his approval file will show the changes. Making it secure. The owner of the file has full control over his data in the cloud and without his approval data or contents cannot be changed as explained above which was not present as part of the existing system.

Algorithm

A. MD5

MD5 hashing is a one-way cryptographic function or algorithm that accepts a message of any length as input and returns a fixed- length digest value as output which will be used for authenticating the original message. The algorithm takes a message of arbitrary length as input and produces a 128-bit 'fingerprint' or 'message digest' of the input as the output. Producing two messages having the same message digest, or to produce any message having a given pre-specified target message digest is highly not possible. The MD5 algorithm is created and used by digital signature applications, where a large file will be 'compressed' in a secure manner before being encrypted with a private (secret) key under cryptosystem that is public-key such as RSA.

B. Homomorphic Encryption

The type of encryption that allows computation on ciphertexts is known as Homomorphic Encryption. This also allows generating an encrypted result which, which when decrypted, matches the result of the operations that had been performed on the plain text. Homomorphic encryption can be used for computation and privacy-preserving outsourced storage[1-3]. This allows data to be encrypted and out-sourced to the commercial cloud environments for processing, all while remaining encrypted. Homomorphic encryption is a type of encryption which allows data to remain encrypted while it's being processed and manipulated. This also enables you or a third party (such as a cloud provider) to apply some functions on encrypted data without needing to reveal the values of the data.

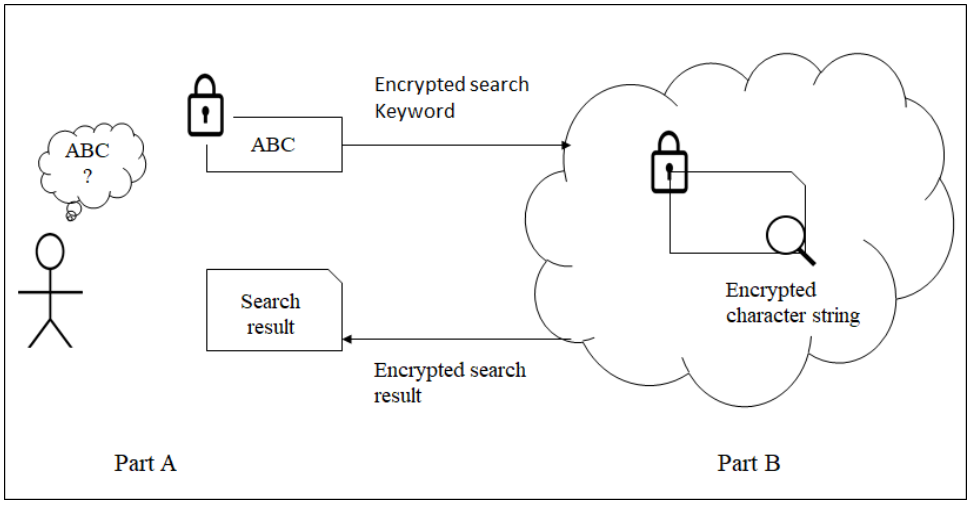


Figure 2. Homomorphic Encryption

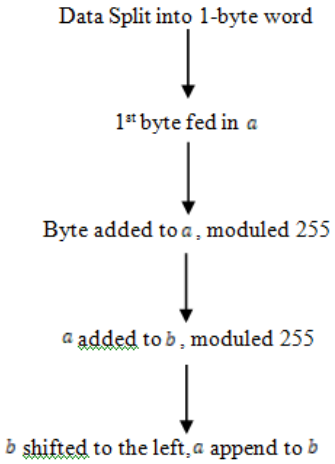
| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 5, May 2020 ||

C. Checksum

A checksum is a data in small size derived from a block of digital data to detect errors that may have raised during the transmission or storage. Checksums are often used to verify data integrity but are not relied entirely upon to verify data authenticity. The procedure which generates this checksum is called a checksum function or a checksum algorithm. Depending on its design plan, a good checksum algorithm will usually output a significantly different value, even for small changes made to the input. This is also true for cryptographic hash functions, which may be used to detect many data corruption errors and verify the overall data integrity[1-5].

Cycles:



MODULE DESCRIPTION:

User Plugin:

In this proposed System there is a user interface that is very user friendly as it is very easy to interact with this System. This system also provides a very high level of security making it highly secure as compared to the previously existing systems. Every user plays can act as a data owner as well as act as a data consumer. During the upload process of a file if they have full control over the file they are the owner of such files and if they can search other users' files then they act as the consumer of such files. Users can create the account by themselves, for this creation there are a set of separate pages where the account creation can be done. It will get all the details necessary for the creation of the account from the user and there is also a high-security authentication mechanism involved for users that are only related to the files or verified users can only create a user account.

In this System, it is easy for the users to search the file and they don't need to remember all the details of uploaded files such as full exact name, instead of the full name it uses keywords for each file present in the cloud. While uploading the files these keywords will help the users to search the files available in the cloud easily.

Uploading File:

Storing data over storage servers is one way to provide data robustness. Which replicates a message such that each storage server stores a message or another way is to encode a message of k symbols into a codeword of n symbols by MD5 coding. To store a message, each of its codeword symbols is stored in multiple storage servers. A storage server corresponds to an MD5 error of the codeword symbol. As long as the number of servers is tolerated then from the threshold of the MD5 code, the message can be recovered from the codeword symbols stored in the available storage servers by the decoding process.

In this module where uploading of a file takes place for each file uploaded there will be many options related to sharing, keywords related to the files will be there. These options will be asked whenever trying to upload a file into the cloud. While uploading a new file to the cloud. The first option will be whether you want to be able to share the file with other users or not you can share files to multiple users. If the user selects YES as the option, then

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 5, May 2020 ||

the file will be enabled for sharing and if you select the option as NO file will not be enabled for sharing. The second option if you have selected YES for the sharing, then you can set keywords for the file to be shared which will be linked to the file that you are uploading. Also users can use the keywords later to take the file easily. The third option will be to mention to whom and all files can be shared. User ids of people to whom the file will be shared can be mentioned here. Next you can upload the file by selecting the file using the browse menu.

Secret Key Formation:

After the user uploads a file that he wants to share along with the keyword and user id details as part of the Uploading File Module. Then Secret key formation module will happen which will cause a secret key to be generated which will be unique for each file during the uploading process itself. Firstly, the secret key will be generated as the initial step while uploading the file, every file which is uploaded will have a unique secret key. This key will be taken as the identification of every file.

The secret key which will be generated is a three-digit number that will be used for both uploading and downloading. If the user wants to download some file and if the user gives the download request then the secret key of that file will be sent to the file owner so, he can share it. This secret key which is generated during the uploading of a file is one important part of providing security to the file in the cloud. Using these secret keys only both the uploading and downloading can be done. The owner of the file will be sent with these secret keys which he will use to share the file with other users.

File Allocation Process:

The file allocation process can share a file to a registered user by providing basic credentials, with the sharing option as it is necessary to provide authority to the shared user if to view or edit the file. A user can view the shared file within the application without downloading it and the same is possible with the edit option where the user can edit the shared file within the application without downloading it. The source file cannot be edited without the knowledge of the file owner.

The file allocation process is the process where a registered user has authenticated himself and has the file shared to him by the owner of the file. So after which file will be allocated such that the shared user can view the shared file within this application without downloading it and also can make changes to the file which will not get reflected in the original file without the approval from the owner of the file.

File Analyzing Process:

Auditing is the process of checking the file whether the original contents of the file are changed. This module provides the file owner auditing, this achieves by generating tokens. By utilizing the homomorphism token with distributed verification of MD5-coded data, the tokens are generated with the ASCII values of the characters in the file and these characters are stored in the DataBase while uploading the file. If a shared user edits the file and saves it, again a new token will be generated and stored in the Data Base. If the initial token and the current token aren't the same then a notification will be sent to the file owner.

For every edit made tokens will be generated and if the tokens generated are different from the original file tokens before the change then after comparison of these tokens owner of the file will be notified about this change by the shared user. After this owner of the file can approve the edits or cannot allow the changes to be reflected in the file.

File Loading Process:

The process to download the file is to get the corresponding secret key to the corresponding file to the user mail id and then decrypt the file data. The process of downloading a file's re-encryption key to storage servers such that storage servers perform the re-encryption Operation. The forwarded message length and the computation of re-encryption are taken care of by storage servers. Re-encryption of Proxy Schemes significantly reduces the overhead of the data Forwarding function in a secure storage system.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 5, May 2020 ||

Alert Mail:

The uploading and downloading process of the user is to get the secret key in the corresponding user email id and then apply the secret key to encrypted data to send the server storage and decrypts it with the help of a secret key to download the corresponding data file in the server storage systems. Conversion of the secret key uses the Share KeyGen (SKA, t, m). Secret key SKA of a user is shared by the algorithm to a set of key servers.

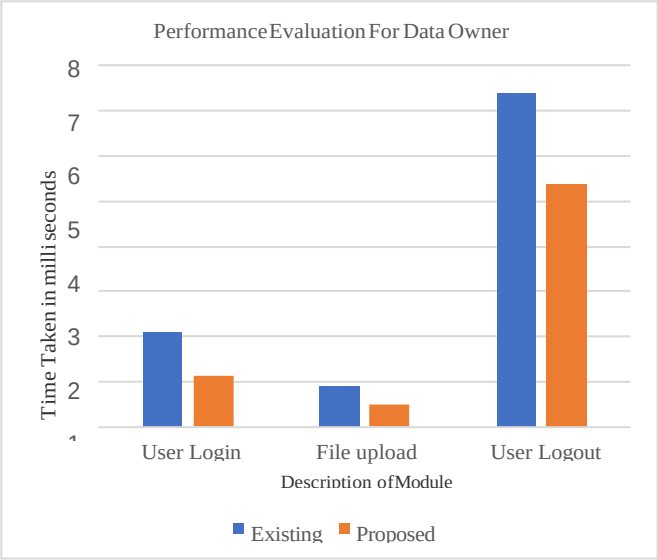
Advantages:

- By utilizing the homomorphic token with distributed verification of MD5-coded data, this scheme achieves the storage correctness insurance as well as data error localization.
- This proposes an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of user's data in the cloud.

- This relies on MD5 correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability.
- This construction drastically reduces the communication and storage overhead as compared to the traditional replication- based file distribution techniques.

IV.RESULTS AND DISCUSSIONS

Performance Evaluation



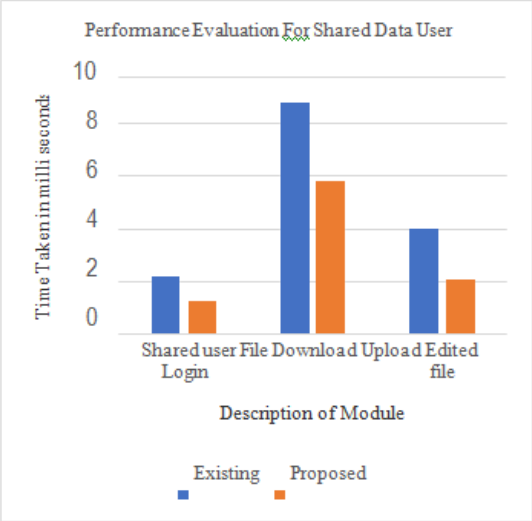
Modules	Existing	Proposed
User Login	2.1ms	1.13ms
File upload	0.9ms	0.5ms
User Logout	7.4ms	5.4ms

Tab:1: Time Taken for different modules

From the table, the user login took 2.1ms in the existing system and it is 1.13ms in the proposed system, the file upload took 0.9ms in the existing system and 0.5ms in the proposed system and user logout took 7.4ms in the existing system and 5.4 ms in the proposed system. So User login, File upload, and User logout all took less time in the proposed system compared to the existing system.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 5, May 2020 ||



Modules	Existing	Proposed
Shared user Login	2.2ms	1.27ms
File Download	8.8ms	5.8ms
Upload Edited file	4ms	2.06ms

Tab:2: Time Taken for the shared data user

From the table, the user login took 2.1ms in the existing system and it is 1.13ms in the proposed system, the file upload took 0.9ms in the existing system and 0.5ms in the proposed system and user logout took 7.4ms in the existing system and 5.4 ms in the proposed system. So User login, File upload, and User logout all took less time in the proposed system compared to the existing system The use of checksum and MD5 technique has made the system with these enhancements. MD5, Checksum and Homomorphic encryption techniques and algorithms have increased the time of all the modules as compared to the Existing System where only the AES Encryption technique is used.

V.CONCLUSION

A proposal has been made for a privacy-preserving public auditing system for data storage security in cloud computing. Homomorphic linear authenticator and random masking are used to guarantee that the TPA will not be able to know about the data stored on the cloud server during the process of auditing, which eliminates the burden of cloud user from the tedious and possibly expensive auditing task and alleviates the users' fear of their data being leaked. TPA may concurrently handle multiple audit sessions from different users for their data files that are outsourced, Further extension is done for privacy-preserving public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks as a batch for better efficiency. In Conclusion, this proposal aims to make the security and integrity of the data in the cloud to be high. As the owner of the file has full control over his data in the cloud and without his approval data or contents cannot be changed as explained above which was not present as part of the existing system.

VI.FUTURE SCOPE

- In the future there is a possibility to expand the protection safeguarding open evaluating convention into a multi-client setting, where the TPA can play out different examining errands in a cluster way for better productivity.
- In the future it will enhance the execution.
- In this framework it has only content records, in future this will incorporate the picture, sound, video documents.
- In our framework the OTP sent to proprietor mail id, in future the customer will get the OTP using a portable method by utilizing the versatile number.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 5, May 2020 ||

ACKNOWLEDEMENTS

We express our deepest sense of gratitude towards our respected and noble guide Dr.K.Ganesh Kumar Sir for spending his valuable time on several occasions to impart us the gains of his knowledge. It was our privilege to have worked under his guidance. we shall always remain indebted to him. We wish to express our earnest thanks to the Industrial adepts for providing us the opportunity to explore the feasible options that were available to us.

REFERENCES

[1] Jiguo Li, Hao Yan, and YichenZang,(2019). Constant Size –Ciphertext Distributed CP-ABE Scheme With Privacy Protection and Fully Hiding Access Structure. IEEE Transaction on Dependable and Secure Computing.

[2] Han Qiu, Hassan Noura, MeikangaQiu, Zhong Ming and Gerard Memmi,(2019). A User-Centric Data Protection Method for Cloud Storage Based On Invertible DWT. IEEE Transaction on Cloud Computing.

[3] Hazem M. Eldeeb, Hisham M. Dahshan, Alaa El-DiR. Shehata, (2018). A new Cryptographic Pairing-based Security Solution for Cloud Storage. International Japan-Africa Conference on Electronics,Communications and Computations (JAC-ECC)

[4] Z. Ren, L. Wang, R. Deng and R. Yu, (2018). Improved fair and dynamic provable data possession supporting public verification,'' Wuhan University Journal of Natural Sciences, vol. 18(4): 248-354.

[5] ArunaGuruvayaMogarala, Dr. Mohan K.G, (2018).Security and privacy designs based data encryption in cloud storage and challenges: A review. 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT).

[6] YongYu, Man Ho Au, Giuseppe Ateniese , Xinyi Huang, Willy Susilo , Yuanshun Dai, Geyong Min,(2017).Identity-based Remote Data Integrity Checking with Perfect Data Privacy Preserving for Cloud Storage. IEEE Transactions on Information Forensics and Security.Vol. 12(4): 767-778.

[7] Anmin Fu, Shui Yu, Yuqing Zhang, Huaqun Wang, ChanyingHuang,(2017).NPP: A New Privacy-Aware Public Auditing Scheme for Cloud Data Sharing with Group Users. IEEE Transactions on Big Data.

[8] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg and I. Brandic, (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, Vol. 25(6): 599-616.

[9] J. Li, W. Yao, Y. Zhang, H. Qian and J. Han, (2017). Flexible and fine-grained attribute-based data storage in cloud computing," IEEE Trans. Service Compute., Vol.10(5): 785-796.

[10] J. Li, X. Lin, Y. Zhang, and J. Han, (2017). KSF- OABE: outsourced attribute-based encryption with keyword search function for cloud storage. IEEE Trans. Service Compute., Vol. 10(5): 715-725.

[11] J. Li, Q. Yu and Y. Zhang, (2019). Key-policy attribute-based encryption against continual auxiliary input leakage.
Information Sciences, Vol. 470: 175–188

[12] H. Qian, J. Li, Y. Zhang and J. Han, (2015). Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation. Int. J. Inf. Secure., vol. 14(6): 487-497

[13] J. Li, Q. Yu and Y. Zhang, (2019). Hierarchical attribute-based encryption with continuous leakage-resilience. Information Sciences, Vol. 484: 113–134.

[14] Y. Zhang, J. Ni, X. Tao, Y. Wang and Y. Yu, (2016). Provable multiple replication data possession with full dynamics for secure cloud storage. Concurrency and Computation: Practice and Experience, Vol. 28(4): 1161-1173.

[15] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu and X. Zhang, (2017). Efficient public verification of data integrity for cloud storage systems from distinguish ability Obfuscation. IEEE Trans. Inf.Foren. and Sec., Vol. 12(3): 676-688.

[16] M. Ali, S.U. Khan and A.V. Vasilakos, (2015). Security in cloud computing: Opportunities and Challenges. Inf. Sci. vol. 305(1):357–383.