International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 3 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195346

# Enhanced Integrity Checking for Preserve Data Owner and User Level Privacy Using Dual Cryptography Approach

**Poovizhi. M[1], Raja. G[2]**
[1]ME-Computer Science and Engineering, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamil Nadu, India
[2]Assistant Professor, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamil Nadu, India

## ABSTRACT

Using Cloud Storage, users can tenuously store their data and enjoy the on-demand great quality applications and facilities from a shared pool of configurable computing resources, without the problem of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained dividing resources. From users' perspective, including both individuals and IT systems, storing data remotely into the cloud in a flexible on-demand manner brings tempting benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. To securely introduce an effective Sanitizer and third party auditor (TPA), the following two fundamental requirements have to be met: 1) TPA should be able to capably audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2) The third party auditing process should take in no new vulnerabilities towards user data privacy. In this project, utilize and uniquely combine the public auditing protocols with double encryption approach to achieve the privacy-preserving public cloud data auditing system, which meets all integrity checking without any leakage of data. To support efficient handling of multiple auditing tasks, we further explore the technique of online signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. We can implement double encryption algorithm to encrypt the data twice and stored cloud server in Electronic Health Record applications.

**Keywords :** Cloud Framework, Public Auditing, Data Integrity Protection, Double Encryption, Multi User Setting

## I. INTRODUCTION

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. It is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment. The idea of cloud computing is based on a very fundamental principles of reusability of IT capabilities. The difference that cloud computing brings compared to traditional concepts of "grid computing", "distributed computing", "utility computing", or "autonomic computing" is to broaden

CSEIT195346 | Received : 01 May 2019 | Accepted : 17 May 2019 | May-June -2019 [ 5 (3) : 138-146 ]

horizons across organizational boundaries. Forrester defines cloud computing as: "A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end customer applications and billed by consumption". It is a technology that uses the internet and central remote servers to maintain data and applications and allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing data storage, processing and bandwidth. Cloud computing examples are Yahoo email, Gmail, or Hotmail. While the storage of corporate data on remote servers is not a new development, current expansion of cloud computing justifies a more careful look at its actual consequences involving privacy and confidentiality issues. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those un-accessed data and might be too late to recover the data loss or damage. To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and affordable way for the users to ensure their storage correctness in the cloud. In a word, enabling public auditing services will play an important role for this nascent cloud economy to become fully established; where users will need ways to assess risk and gain trust in the cloud. The basic cloud is shown in fig 1.

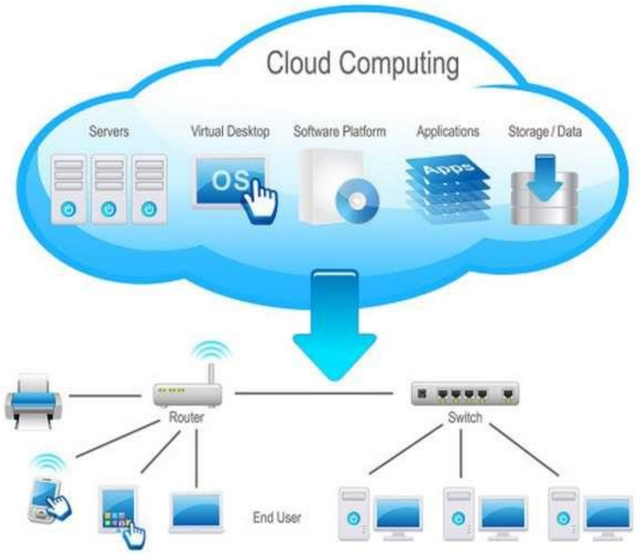

**Fig 1 :** Cloud deployment model

## II. RELATED WORK

G. Ateniese, et.al,…[1] provide a framework for constructing leakage-resilient ID protocols in the BRM from publicly verifiable proofs of storage (PoS) that are computationally zero-knowledge (ZK). PoS are interactive protocols allowing a client to verify that a server faithfully stores its file. A PoS is publicly verifiable if anyone with access to the client's public-key can verify the server's storage and it is computationally ZK if, roughly speaking, its verification phase leaks no useful information about the file to a bounded adversary. We show how to construct such a scheme based on the RSA assumption. The secret key of the identification protocol is the encoding of a randomly-generated file and the public key is the state information generated by encoding the file together with the public key for the PoS. To identify itself, the prover executes the verification phase of the PoS with the verifier to prove that it indeed holds the file. We showed that zero-knowledge proof-of-storage schemes can be used to build leakage-resilient identification protocols in the bounded retrieval model (BRM). Our framework provides new insights into the BRM and unfolds new ways to build leakage-resilient identification protocols in this model.

Z. Fu, K. Ren, et.al,…[2] provided a popular way to search over encrypted data is searchable encryption and many constructive schemes have been put forward under different applications. However, these searchable encryption schemes based on keyword no longer fully satisfy the new challenge and users' increasing needs, specifically manifested in the following two aspects. One is that most of existing schemes follow the model of "one size fits all" and ignore individual users' experience due to their different hobbies, interests or cultural backgrounds. In those schemes, the cloud will return all files that match the user's query, which may cause a huge consumption of network bandwidth. Moreover, it will cost user much time and many resources to filter his real interesting ones among a large quantity of returned files. In the practical application, different users may find different things relevant because of different importance or priorities of query terms, indicating the necessity of personalized search, which takes personal keyword preference or keyword priority into account. So how to design an efficient search scheme that can really understand the user's search intention is a pressing problem. However, these schemes cannot be directly applied in searchable encryption schemes due to the lack of consideration of privacy and security. And proposed a preferred keyword search scheme over encrypted data, but the artificial manner of measuring keyword preference has great randomness and fails to consider different users' search histories. The other one is that most of these schemes support only exact keyword search. That means the returned result is only related to the user's input. When the user queries some uncommon terms, it is possible that just a few matched results are returned and the user may be not satisfied with the returned results.

Z. Hao, S. Zhong, et.al,… [3] analyzed the system which an increasing number of clients store their important data in remote servers in the cloud, without leaving a copy in their local computers. Sometimes the data stored in the cloud is so important that the clients must ensure it is not lost or corrupted. While it is easy to check data integrity after completely downloading the data to be checked, downloading large amounts of data just for checking data integrity is a waste of communication bandwidth. Hence, a lot of works have been done on designing remote data integrity checking protocols, which allow data integrity to be checked without completely downloading the data. In remote data integrity checking protocols, the client can challenge the server about the integrity of a certain data file, and the server generates responses proving that it has access to the complete and uncorrupted data. The basic requirements are that the client does not need to access the complete original data file when performing the verification of data integrity, and that the client should be able to verify integrity for an unlimited number of times. Furthermore, the protocol needs to be secure against a malicious server that tries to pass the data integrity verification without access to the complete and uncorrupted data. In a realistic application, these advanced features may be needed at the same time. For example, consider an online document system, in which the client can create and modify her documents. The client can also cooperate on a document with her partners. When the client or her partners modify the document, the document and the tags need to be updated.

H. Liu, L. Chen, et.al,… [4] implemented the Cloud storage is becoming increasingly popular because of a laundry list of advantages of this kind of novel storage model. Currently, many cloud storage services such as Amazon S3, Google Cloud, and Microsoft Skydrive have attracted millions of users all over the world, including individuals and organizations. The flexibility and on demand manner of cloud storage brings a lot of appealing benefits over traditional storage approach, say, relief of the burden of storage management, avoiding capital expenditure on hardware, software and personnel maintenance, access to data with independent geographical locations. In this paper, we show the construction is not secure in their security model or in a correct security model. To be specific, with the aid of signature queries, a malicious cloud server could generate a valid response to a challenge from a third party auditor (TPA) even the server has deleted all the files of a user or has corrupted the file. Cloud servers are not necessarily fully trusted and consequently, malicious servers might discard the data that have not been or are rarely accessed for monetary reasons. As a result, strong evidence that their data accommodated on cloud keeps unchanged and is not being tampered with or partially deleted is highly essential for cloud users. Regarding the data privacy, what the scheme can achieve is that an adversary cannot recover the entire file from the auditing process, which is similar to the one wayness of encryption. In fact, the security model is unrealistic in the sense there is no a scheme that can be proven secure in this model.

J. K. Liu, et.al,… [5] analyzed end users access cloud-based applications through a web browser, thin client or mobile app while the business software and user's data are stored on servers at a remote location. The benefits of web-based cloud computing services are huge, which include the ease of accessibility, reduced costs and capital expenditures, increased operational efficiencies, scalability, flexibility and immediate time to market. Though the new paradigm of cloud computing provides great advantages, there are meanwhile also concerns about security and privacy especially for web-based cloud services. As sensitive data may be stored in the cloud for sharing purpose or convenient access; and eligible users may also access the cloud system for various applications and services, user authentication has become a critical component for any cloud system. A user is required to login before using the cloud services or accessing the sensitive data stored in the cloud. There are two problems for the traditional account/password based system. First, the traditional account/password-based authentication is not privacy-preserving. Second, it is common to share a computer among different people. It may-be easy for hackers to install some spyware to learn the login password from the web-browser.

## III. EXISTING METHODOLOGIES

While cloud computing makes various advantages, it can be mentioned in chapter 1 and challenging security threats toward users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. CSP might reclaim storage for monetary reasons by discarding data that have not been or are rarely accessed, or even hide data loss incidents to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users [2] correctness assurance for those un-accessed data and might be too late to recover the data loss or damage.

### 3.1. WATERMARKING SCHEME

And implement the system to provide water marking process, to store the data or images in the cloud server by assigning the public key, and this key and watermarking images are sent to third party and third party have complete authority to check the key and sent it to the server, and there Third Party Auditor must have a public key whenever the data to be retrieved. In the watermarking process, the security level is very high so the data or images cannot be identified by the attackers in the cloud and also use Compression technique for watermark image to reduce communication overhead. The main elements in watermarking process: an embedded, a communication channel and a detector. Watermark information is embedded into original image itself, and it is performed in the encryption process for making security on original information. Embedded is similar to encryption process which is used to change content into another format with the help of the secret key. Detector process is also similar to decryption process which is used to perform reverse process of encryption. The watermark information is embedded within the original image before the watermarked image is transmitted over the communication channel, so that the watermark image can be detected at the receiving end.

### 3.2 ONE RING TO RULE THEM ALL (ORUTA) scheme:

Then implemented ORUTA that include a privacy preserving public auditing mechanism for shared data in an untrusted cloud. In Oruta, utilize ring signatures to construct homomorphic authenticators, so that the third party auditor is able to verify the integrity of shared data for a group of users without retrieving the entire data — while the identity of the signer on each block in shared data is kept private from the TPA. In addition, extend the mechanism to support batch auditing, which can audit multiple shared data simultaneously in a single auditing task. Meanwhile, Oruta continues to use random masking to support data privacy during public auditing, and leverage index hash tables to support fully dynamic operations on shared data. A dynamic operation indicates an insert, delete or update operation on a single block in shared data. In this paper, we only consider how to audit the integrity of shared data in the cloud with static groups. It means the group is pre-defined before shared data is created in the cloud and the membership of users in the group is not changed during data sharing. The original user is responsible for deciding who is able to share her data before outsourcing data to the cloud. Another interesting problem is how to audit the integrity of shared data in the cloud with dynamic groups — a new user can be added into the group and an existing group member can be revoked during data sharing — while still preserving identity privacy.

## IV. PROPOSED METHODS

The system model in this project involves three parties: the cloud server, a group of users and a public verifier. There are two types of users in a group: the original user and a number of group users. The original user initially creates shared data in the cloud, and shares it with group users. Both the original user and group users are members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata (i.e. signatures) are both stored in the cloud server. A public verifier, such as a third-party auditor (TPA) providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server. When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. After receiving the auditing challenge, the cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof. Essentially, the process of public auditing is a challenge and-response protocol between a public verifier and the cloud server.

**Public Auditing** A public verifier is able to publicly verify the integrity of shared data without retrieving the entire data from the cloud.

**Correctness** A public verifier is able to correctly verify shared data integrity.

**Unforgetability** Only a user in the group can generate valid verification metadata (i.e., signatures) on shared data.

**Identity Privacy** A public verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.

With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes standard feature in most cloud storage offerings, including Dropbox, iCloud and Google Drive. The integrity of data in cloud storage, however, is subject to skepticism and scrutiny, as data stored in the cloud can easily be lost or corrupted due to the inevitable hardware/software failures and human errors. The traditional approach for checking data correctness is to retrieve the entire data from the cloud, and then verify data integrity by checking the correctness of signatures or hash values of the entire data. Certainly, this conventional approach able to successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is in doubt. The main reason is that the size of cloud data is large in general. Downloading the entire cloud data to verify data integrity will cost or even waste user's amounts of computation and communication resources, especially when data have been corrupted in the cloud. Recently, many mechanisms have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing. In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking. A public verifier could be a data user (e.g. researcher) who would like to utilize the owner's data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking service. In this proposed system we can implement Merkle Hash Tree to spilt the files into various parts and to provide double encryption concept to encrypt the data first at owner side and again encrypt the data based on TPA provided keys. Finally provide batch auditing schemes to perform multiple tasks at a time and user level privacy can be implemented to share the data without any leakages. The proposed framework is shown in fig 2.

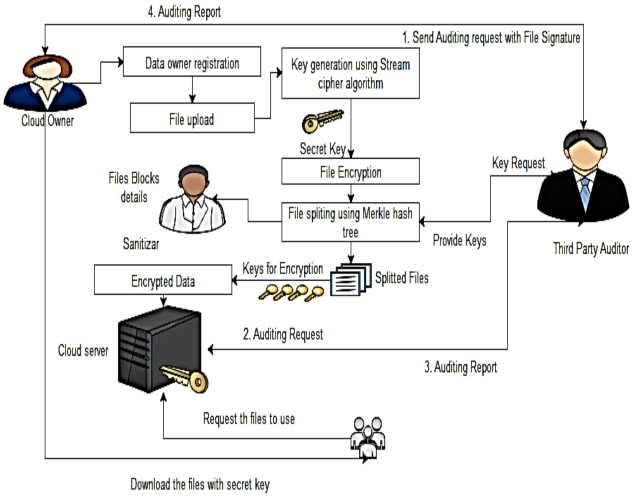

**Fig 2.** Proposed Framework

## SYMMETRIC KEY ALGORITHM

A stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (key-stream). In a stream cipher, each plaintext digit is encrypted one at a time with the corresponding digit of the key-stream, to give a digit of the cipher-text stream. Since encryption of each digit is dependent on the current state of the cipher, it is also known as state cipher. In practice, a digit is typically a bit and the combining operation an exclusive-or (XOR). The steps are

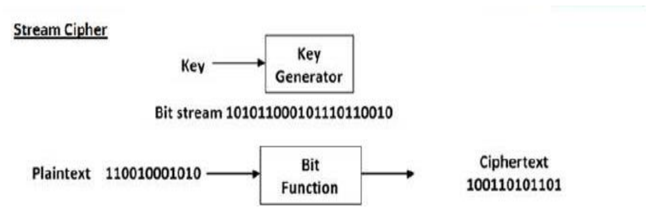

Fig 3: Symmetric Key algorithm

## MERKLE HASH TREE (MHT)

To achieve privacy-preserving public auditing, propose to uniquely integrate the linear authenticator with binary tree technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server. With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. On the other hand, the correctness validation of the block-authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. Our design makes use of a public key-based MHT, to equip the auditing protocol with public audit ability. A MHT Encryption scheme is comprised of a tuple of algorithms (Gen, E,D, Eval), and is defined with respect to a circuit C with t inputs. Though a MHT scheme can be either a public-key or symmetric-key system, we will define it as a public-key system here. The key generation algorithm Gen takes the security parameter $1^k$ as input, and outputs the public key and private key for the system (Notation: (pk, sk) ← Gen($1^k$)).

Assume that messages $M \in \{0, 1\}^{1\,(k)}$.

The encryption algorithm E takes a public key and a message as input, and outputs a ciphertext C, (Notation: C ← E(pk,M) for $M \in \{0, 1\}^{l(k)}$).

The decryption algorithm D takes a secret key and a ciphertext, and returns a message, (Notation: M ← D(sk,C) and $M \in \{0, 1\}^{l}$).

Finally, the evaluation algorithm Eval takes as input a public key, a description of a t-input circuit C, and t ciphertexts $C_1, \ldots, C_t$ such that Ci← E(pk,Mi), and produces as output $C^*$, (Notation: $C^*$ ← Eval(pk,C, $C_1, \ldots, C_t$)).

We add a new correctness property to the standard correctness requirement for an encryption scheme as follows. We say that an encryption scheme is homomorphic with respect to a t-input circuit C if ∀k, ∀M1, . . . ,Mt, Pr[(pk, sk) ← Gen(1k); C1, . . . Ct ← E(pk,M1), . . . , E(pk,Mt); $C^*$ ← Eval(pk,C, $C_1, \ldots, C_t$) : D(sk,$C^*$) = C($M_1, \ldots, M_t$)] = 1.

Similarly, a scheme with respect to a family of circuits $\{C_i\}$ if the correctness property holds for any circuit $C \in \{C_i\}$. Note that so far, our definition makes no requirement that the output $C^*$ of Eval should look like a standard ciphertext. Indeed, without some additional restriction on $C^*$, every standard encryption scheme (Gen, E,D) can be trivially modified to yield a homomorphic encryption scheme (Gen', E',D', Eval') with respect to all circuits as follows.

Gen' runs as Gen.
E' runs as E.

The Eval' is constructed to take a public key, a circuit description, and up to t ciphertexts, and then output the circuit description concatenated with each of the ciphertexts, as C* ← Eval"(pk,C, C1, . . . ,Ct) = C|C1| . . . |Ct, with | used to denote concatenation.
On special cipher texts C* containing a circuit description, D' parses its input into C, $C_1$, . . . ,$C_t$, runs the original decryption algorithm D on the ciphertexts to obtain messages Mi ← D(sk,Ci), and runs the circuit C on these messages, to obtain D'(sk,C*) = C($M_1$, . . . ,$M_t$), satisfying the homomorphic correctness property. On ciphertexts without circuit descriptions, D'(sk,C) simply returns D(sk,C).

## BATCH AUDITING

With the establishment of privacy-preserving public auditing, the TPA may concurrently handle multiple auditing upon different users' delegation. The individual auditing of these tasks for the TPA can be tedious and very inefficient. Given K auditing delegations on K distinct data files from K different users, it is more advantageous for the TPA to batch these multiple tasks together and audit at one time. Keeping this natural demand in mind, we slightly modify the protocol in a single user case, and achieve the aggregation of K verification equations (for K auditing tasks) into a single one. As a result, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

1. Verify file tag tk for each user k, and quit if fail
For each user k (1≤ k ≤ K)
2. Generate a random challenge
3. Compute μk, σk, Rkas single user case;
   Chal = {(I, Vi)} i∈ I
4. Compute R=R1, R2,….Rk
   L = vk1||vk2||…..||$vk_k$
5. Compute μk = rk + γkμ′kmodp
6. Compute γk = h(R||Vk||L) for each user k and do batch auditing

## V. CONCLUSION

Cloud computing securities are discussed and analyzed in previous study. In this project, some of the privacy threats are addressed and the techniques to overcome them are surveyed. While some approaches utilized traditional cryptographic methods to achieve privacy, some other approaches kept them away and focused on alternate methodologies in achieving privacy. Also, approaches to preserve privacy at the time of public auditing are also discussed. Thus, to conclude it is necessary that every cloud user must be guaranteed that his data is stored, processed, accessed and audited in a secured manner at any time. Data freshness is essential to protect against misconfiguration errors or rollbacks caused intentionally and can develop an authenticated file system that supports the migration of an enterprise-class distributed file system into the cloud efficiently, transparently and in a scalable manner. It's authenticated in the sense that enables an enterprise tenant to verify the freshness of retrieved data while performing the file system operations. The user must be given complete access control over the published data. Also, powerful security mechanisms must always supplement every cloud application. Attaining all these would end up in achieving the long dreamt vision of secured Cloud Computing in the nearest future.

## VI. REFERENCES

[1]. G. Ateniese, A. Faonio, and S. Kamara, "Leakage-resilient identification schemes from zero-knowledge proofs of storage," in IMA Inte. Conf. Cryptography and Coding, 2015, pp. 311–328.

[2]. Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," IEEE Trans. Parallel Distrib. Syst., doi. 10.1109/TPDS. 2015.2506573.

[3]. Z. Hao, S. Zhong, and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," IEEE Trans. Knowl. Data Eng., vol.23, no.9, pp.1432-1437, 2011.

[4]. H. Liu, L. Chen, Z. Davar, and M. Pour, "Insecurity of an efficient privacy-preserving public auditing scheme for cloud data storage," J. Universal Comput. Sci., vol. 21, no. 3, pp. 473–482, 2015.

[5]. J. K. Liu, M. H. Au, X. Huang, R. Lu, and J. Li, "Fine-grained twofactor access control for web-Based cloud computing services," IEEE Trans. Inf. Forens. Security, vol. 11, no. 3, pp. 484–497, 2016.

[6]. F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J. J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. Knowl. Data Eng., vol. 20, no.8, pp. 1034–1038, 2008

[7]. C. Wang, Q. Wang, S. C, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, 2013.

[8]. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM), 2010, pp. 1–9.

[9]. Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 2, pp. 340-352, 2015.

[10]. L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and Z. Dong, "Round-efficient and sender-unrestricted dynamic group key agreement protocol for secure group communications," IEEE Trans. Inf. Forens. Security, vol. 10, no. 11, pp. 2352-2364, 2015.

**Cite this article as :**

Poovizhi. M, Raja. G, "Enhanced Integrity Checking for Preserve Data Owner and User Level Privacy Using Dual Cryptography Approach", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 3, pp. 138-146, May-June 2019. Available at doi : https://doi.org/10.32628/CSEIT195346
Journal URL : http://ijsrcseit.com/CSEIT195346