International Journal of Current Research and Modern Education (IJCRME)
ISSN (Online): 2455 - 5428
(www.rdmodernresearch.com) Volume I, Issue I, 2016

PROVABLE CHECK SERVICE FOR UNRELIABLE STORAGE

S. Aarthi* & R. Selvakumar**

* PG Scholar, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

** Assistant Professor, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

Abstract:

A Verifiable audit service for integrity verification of dishonest and outsourced storages. Our audit service is constructed based on the techniques, fragment structure, random sampling, and index-hash table, supporting provable updates to outsourced data and timely anomaly detection. Constructed on interactive proof systems (IPS) with the zero knowledge property, our audit service can provide public verifying without downloading raw data and protect privacy of the data. We also propose an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services. Security and performance objectives such as Public audit ability, dynamic operations, timely detection, and effective forensic are the objectives should be addressed to achieve an efficient audit for outsourced storage in the clouds. This audit service is significantly important for digital forensics and credibility in clouds. To implements public audit ability, the notions of proof of irretrievability and provable data possession have been proposed by some researchers. A proof-of concept prototype is also implemented to evaluate the feasibility and viability of our proposed approaches. Our experimental results not only validate the effectiveness of our approaches, but also show that our system does not create any significant computation cost and require less extra storage for integrity verification.

Key Terms: Integrity Auditing, Database Encryption, Outsourcing Computation, Cloud Computing, Deduplication, Cloud Service Provider (CSP) & Outsourced Database

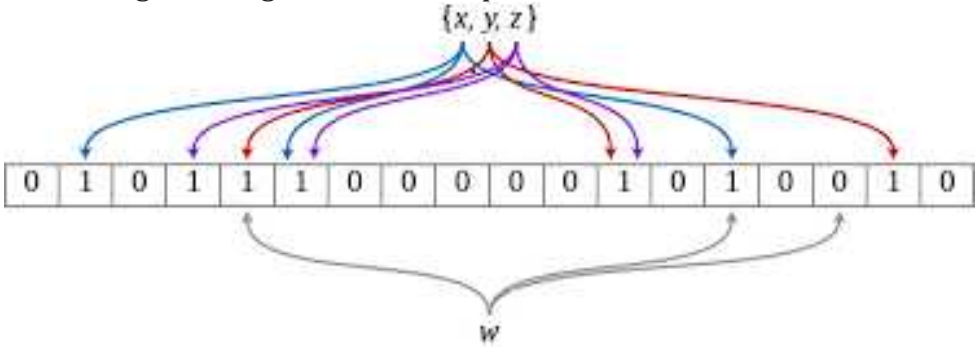
1. Introduction:

The cloud storage service (CSS) relieves the burden for storage management and maintenance. However, if such an important service is vulnerable to attacks or failures, it would bring irretrievable losses to the clients because their data or archives are stored in an uncertain storage pool outside the enterprises. These security risks come from the cloud infrastructures are much more powerful and more reliable than personal computing devices, but they are still susceptible to internal threats and external threats that can damage data integrity. For the benefits of possession, there exist various motivations for cloud service providers (CSP) to behave unfaithfully toward the cloud users; furthermore, disputes occasionally suffer from the lack of trust on CSP because the data change may not be timely known by the cloud users, even if these disputes may result from the users' own improper operation. Therefore, it is necessary for CSP to offer an efficient audit service to check the integrity and availability of stored data. Their security and performance objectives such as Public audit ability, dynamic operations, timely detection, and effective forensic are the objectives should be addressed to achieve an efficient audit for outsourced storage in the clouds. Public audit ability to allow a third party auditor (TPA) or clients with the help of TPA to verify the correctness of cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to cloud services. Dynamic operations to ensure there is no attack to compromise the security of verification protocol or cryptosystem

by using dynamic data operations. Timely detection to detect data errors or losses in outsourced storage, as well as anomalous behaviors of data operations in a timely manner. Effective forensic to allow TPA to exercise strict audit and supervision for outsourced data, and offer efficient evidences for anomalies.

2. Related Works:

The primitive of database outsourcing has been well studied in the past decades. The existing ODB schemes can be categorized into three types by different verification approaches: The first approach uses authenticated data structures (e.g., Merkle hash tree) to provide integrity of search results. The main idea is that an index is generated based on the Merkle hash tree for the whole database and the integrity auditing of search result can be achieved by re-computing the signature on the root of the MHT. The disadvantage of MHT-based approach is that it requires plenty of communication and computation overhead. That is, in order to achieve the verifiability of each single data tuple, a verification object including many tuples needs to be sent to the user, which is a set of login sibling nodes from a specific leaf to the root.



On the other hand, none of the MHT-based schemes can ensure the completeness of searching results. The second approach is a probabilistic integrity verification method. The main idea is that the data owner inserts some faked tuples to the database beforehand (the solution is very similar to the idea of "ringers"). There are two disadvantages for this solution: First, the faked tuples must be shared by all authorized users and thus the CSP could collude with a corrupted user. Second, this method requires the CSP to return all attributes of tuple and thus cannot support some common database operations such as projection. The last approach is based on the signature chaining technique which can save communication and computation overhead for query verification.

Organization: The rest of the paper is organized as follows. We present some preliminaries in Section 2. In Section 3, we present the system and threat model of our proposed scheme. The detailed description of our proposed scheme and its security Analysis are presented in Section 4. The performance evaluation is given in Sections 5. Finally, the conclusion is given in Section 6.

Preliminaries: In this section, we first present the basic definitions and properties of Bloom filter (BF) and Merkle hash tree. We then review the encryption scheme of outsourced database proposed by Evdokimov.

Bloom Filter: The Bloom filter is a kind of space-efficient data structure proposed by Bloom in 1970, which can be used to check whether an element belongs to a set. A bloom filter consists of a bit array of m bits and k independent hash functions Defined as follows In the initial phase, all the positions of the array are set to 0. To add an element to the set, feed it to each of the k hash functions to get k array positions. Set the bits value at all these position to 1. It is need to note that the operation does not work at

the same position as long as its value is 1.Given any element x, the steps for checking whether it is in the set, feed it to each of the k hash functions to get k array positions.

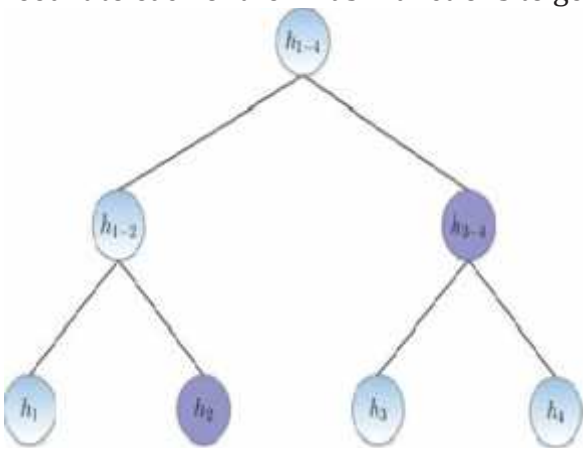


Figure 2: An example of Merkle Hash Tree

This is due to the existence of false positive, i.e., there is a possibility for some x =2 S, all bits at the corresponding positions of are 1. A toy instance of Bloom filter is shown in Fig. 1. It is easy to see that the element z does not belong to the set S. In the case of element w, though all the corresponding positions are 1, it does not belong to S. It is called as false positive. In, the authors analyze the relationship among the size of the Bloom filter m, the number of hash functions k, the number of elements n and the probability of false positive. Specifically, for given m and n, Pf is equal to Pf reaches the minimum value when k is ln2. In this case, Pf is approximately

Merkle Hash Tree: Merkle hash tree is a specific binary tree, which is used to authenticate data with lower communication overhead. In the MHT, each internal node stores a hash value of the concatenation of its child nodes and each leaf node stores Hash value of the authenticated data value, where Þ is a one-way collision-resistant hash function. The MHT can be used to authenticate any subset of D by using the verification object. A VO is a set of all sibling nodes on the path from the desire leaf node to the root. For example, in order to authenticate d1, the VO contains h2, h3_4 and the signature of h1_4. The verifier computes and checks whether h0 1_4 is equal to h1_4. If so, d1 valid; otherwise, it has been tampered with.

3. Proposed Work:

The protection of computer based resources that include hardware, software, data, procedures and people against unauthorized use or natural Disaster is known as System Security. System Security can be divided into four related issues:

- ✓ **System Security** refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.
- ✓ **Data Security** is the protection of data from loss, disclosure, modification and destruction.
- ✓ **System Integrity** refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.
- ✓ **Privacy defines** the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

- ✓ **Confidentiality** is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

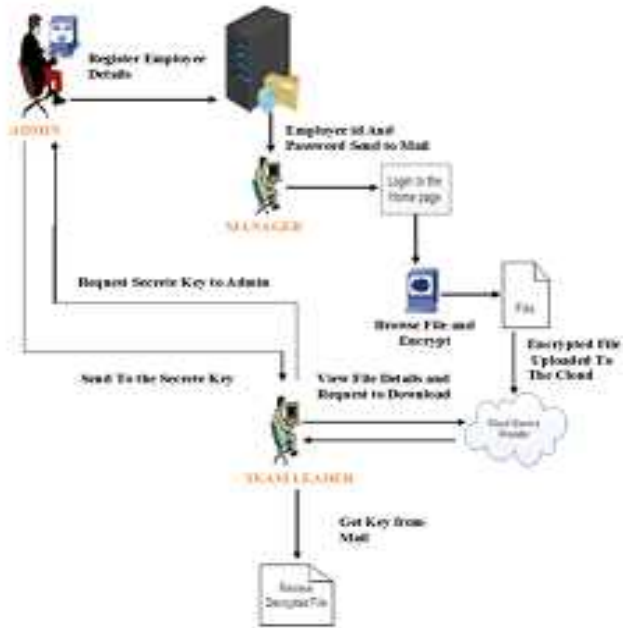
4. Experimental Analysis and Results:

System security refers to various validations on data in the form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employs two types of checks and controls.

Client Side Validation: Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks are imposed:

- ✓ VBScript in used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
- ✓ Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
- ✓ Tab-indexes are set according to the need and taking into account the ease of use while working with the system.

Server Side Validation: Some checks cannot be applied on the client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:



- ✓ A Server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results in a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
- ✓ The User is intimated through appropriate messages about the successful operations or exceptions occurring at server side.
- ✓ Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the

system and can have access according to their category. User- name, passwords and permissions are controlled on the server side.

- ✓ Using server side validation, constraints on several restricted operations are imposed.

Efficiency Analysis: In this section, we present efficiency analysis of the proposed scheme in theory. For the convenience of discussion, some marks are introduced. We denote by Han operation of hash, E an encryption (resp., by D a decryption), by S a signature, by P the paring operation, by Exp the modular exponentiation, by n the number of outsourcing data records, by k the hash number of bloom filter, by a the number of attributes of each data record, and by r the number of returned data records, Besides, Notice that we omit the ordinary file upload (download) operations for simplification. Table 1 presents the comparison between our scheme and two previous schemes.

Cost of System Setup: In the system setup phase, the main computation cost is dominated by hash function calculations, which are used to construct Bloom filter and Bloom filter tree. More precisely, the user conducts (l+1)kna hash and 2 signature operations. It needs to be pointed out that we cannot omit the time cost of hash operation when the number of operation is very large. We perform the experiments with data records ranging from 1,000 to 8,000. As shown in Figs. 6a and 6b, the time cost will linearly increase with data records. It can clearly be seen that the time overhead of constructing BFT is much more than BF because there are more hash operations for BFT construction. Actually, the time cost of constructing BFT (l-level) is approximately equal to l times of constructing BF with the same length. The results indicate that our construction can be acceptable without any computation-intensive operations.

Cost of Data Outsourcing and Retrieving: In order to ensure the security and privacy of outsourced database, the data owner could encrypt each attribute value tuple by tuple before storing it. Specifically, the total computation consists of n signature, an encryption (symmetric), and ð2a _ 1Þn hash operations. Consider that the signature of time is far greater than hash and symmetric encryption; we mainly concern the signature operation in our experiment. As shown in Fig. 7a, the time cost of data outsourcing is increasing with the number of tuples of database. For different number of attributes in each tuple, the time is almost the same because hash and symmetric encryption operations are negligible. Note that the phase of data outsourcing Fig. 7a is obviously slower because there are signature operations for each tuples. That is, the user signs the root of every attribute MHT with his private key, which is used as the proof of tuple in phase of verifying. Although the computation overhead is slightly high, the phase of outsourcing is one-time work and can be done by off-line way. On receiving a user's search request, the CSP checks the corresponding attributes tuple by tuple. For each attribute, the CSP conducts 1 decryption operation. The computation overhead is linearly increasing with data records. The time cost of CSP for different number of data records.

Cost of Verifying: To verify the integrity of search result, we consider two cases: (i) no result is retrieved. That is, the CSP claims that there is no match tuple exists in the outsourced database. Only the Bloom filter is returned as the proof, which is used to check whether the CSP is honest. The main computation is verifying of signature for it. As shown in Fig. 7c, it can be seen that the computation burden keeps constant with the vary of the data records. That is because signature of time is far more than k hash operation. (ii) Some results are retrieved. In this case, the user verifies the integrity of

search result by signature verification. To be specific, for a given tuple in the result set, the user first checks the integrity of attribute by hash reconstruction. Then the integrity of tuple can be checked by verifying signature of root of tuple-MHT. We can see from Figure that the time cost is linearly increasing with the number of tuple in result set. (141065) and Program for New Century Excellent Talents in Fujian University (JA14067), the Fundamental Research Funds for the Central Universities (Nos. BDY151402 and JB142001-14) and Australian Research Council (ARC) projects DP150103732, DP140103649, and LP140100816. Yang Xiang is the corresponding author. (141065) and Program for New Century Excellent Talents in Fujian University (JA14067), the Fundamental Research Funds for the Central Universities (Nos. BDY151402 and JB142001-14) and Australian Research Council (ARC) projects DP150103732, DP140103649, and LP140100816.Yang Xiang is the corresponding author.

5. Conclusion and Future Enhancement:

In this paper, we investigate the integrity auditing of outsourced database in cloud computing. Our main contribution is to propose a new verifiable auditing scheme of outsourced database which can achieve the verifiability of search result even if the result is an empty set. Besides, our scheme supports common database operations such as selection and projection. Furthermore, our construction can be easily extended to the scenario of dynamic database by incorporating the notion of verifiable database with updates. We also prove that our scheme can achieve the desired security goals and provide detailed simulation tests. Verifiable auditing scheme of outsourced database providing Reliable services to the user by protecting data from the dishonest cloud service provider (CSP), a innovative verifiable auditing scheme is proposed that allows the user to audit the outsourced database to check the correctness of the result produced by CSP even in the case of encrypted data. outsourcing In the outsourced database (ODB) model, the data owner delegate the database management to the CSP, in order to reduce the heavy database maintenance cost and the authentication is done for security purpose. The verifiable auditing scheme for outsourced database, which can ensure the integrity of search results even if the dishonest CSP purposely returns an empty set. Our main idea is that we use the above primitives of BFT and Tuple-MHT to check the query integrity.

6. References:

1. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, "Provable data possession at untrustedstores," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 598–609.
2. B. Andrei and M. Michael, "Network applications of bloom filters: A survey," Internet Math., vol. 1, no. 4, pp. 485–509, 2004.
3. M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. H. Spafford,"Secure outsourcing of scientific computations," Adv. Comput.,vol. 54, pp. 215–272, Jan. 2002.
4. B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," Commun. ACM, vol. 13, no. 7, pp. 422–426, Jul. 1970.
5. E. Bertino, B. Carminati, E. Ferrari, B. M. Thuraisingham, and A.Gupta, "Selective and authentic third-party distribution of XML documents," IEEE Trans. Knowl. Data Eng., vol. 16, no. 10, pp. 1263–1278, Oct. 2004.
6. P. Mell and T. Grance, "Draft nist working definition of cloud com-puting," Referenced on Jan. 23rd, 2010 online at http://csrc.nist.gov/ groups/SNS/cloud-computing/index.html, 2010.

7. Juels and B. S. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 584–597.
8. D. Ma, R. H. Deng, H. Pang, and J. Zhou, "Authenticating query results in data publishing," in Proc. 7th Int. Conf. Inf. Commun. Security, 2005, pp. 376–388.
9. R. C. Merkle, "Protocols for public key cryptosystems," in Proc. IEEE Symp. Security Privacy, 1980, pp. 122–134.
10. E. Mykletun, M. Narasimha, and G. Tsudik, "Signature bouquets: Immutability for aggregated/condensed signatures," in Proc. 9th Eur. Symp. Res. Comput. Security, 2004, pp. 160–176.
11. E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," ACM Trans. Storage, vol. 2, no. 2, pp. 107–138, May 2006.