ISSN:0975-9646

Providing Data Utility on Cloud using Slicing approach and Dynamic Auditing Protocol using Third Party Auditor to maintain Integrity of Data

Madhuri R. Rokade[1], Siddaling B.Natikar[2]

Department of Computer Engineering,
Vishwabharati Academy's College of Engineering, Ahmednagar-414201

Abstract -The Cloud-computing releases the data owner's burden for storage management and maintenance by providing a compare ably low-cost, scalable, location-independent platform. However, the fact that clients no longer have physical possession of data indicates that they are facing a potentially formidable risk for missing or corrupted data. This new model of data hosting service commence a new security challenges, which requires an independent auditing service which audit the data integrity of cloud. There are different existing auditing services available in cloud which audit data integrity remotely in static motion but these are not applicable whenever data is dynamically updated in cloud. To avoid the security risks, secure and efficient dynamic auditing protocol is introduced and to ensure the integrity of file stored in cloud and To solve the data privacy problem, we present a novel technique called slicing, which partitions the data both horizontally and vertically. We show that slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data.

1. INTRODUCTION

All CLOUD storage is an important service of cloud computing, which allows data owners (owners) to move data from their local computing systems to the cloud. Owners would worry that the data could be lost in the cloud. This is because data loss could happen in any infrastructure, no matter what high degree of reliable measures cloud service providers would take [4], [5], [6], [7], [8]. Sometimes, cloud service providers might be dishonest. They could discard the data that have not been accessed or rarely accessed to save the storage space and claim that the data are still correctly stored in the cloud. Therefore, owners need to be convinced that the data are correctly stored in the cloud. Recently, several remote integrity checking protocols were proposed to allow the auditor to check the data integrity on the remote server, the authors proposed a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers, Again the authors extended their dynamic auditing scheme to be privacy preserving and support the batch auditing for multiple owners. [10], [11], [12], [13], [14], [15], [16], [17], [18]. In [15], Zhu et al. proposed a cooperative provable data possession scheme that can support the batch auditing for multiple clouds and also extend it to support the dynamic auditing in [26].However, their scheme cannot support the batch auditing for multiple owners. Furthermore, both Wang's schemes and Zhu's schemes incur heavy computation cost of the auditor, which makes the auditor a performance bottleneck.AS far as privacy preserving is concerned, recent work has shown that loses considerable amount of information, especially for high-dimensional data. Bucketization, on the other hand, does not prevent membership disclosure and does not apply for data that do not have a clear separation between quasi- identifying attributes and sensitive attributes.

In both generalization and bucketization, one first removes identifiers from the data and then partitions tuples into buckets. The two techniques differ in the next step. Generalization transforms the QI-values in each bucket into "less specific but semantically consistent" values so that tuples in the same bucket cannot be distinguished by their QI values.In bucketization, one separates the SAs from the QIs by randomly permuting the SA values in each bucket. The anonymized data consists of a set of buckets with permuted sensitive attribute values.

In this paper, we present a novel technique called slicing, which partitions the data both horizontally and vertically. We show that slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data. We show how slicing can be used for attribute disclosure protection. We design an auditing framework for cloud storage systems and propose a privacy-preserving and efficient storage auditing protocol. Our auditing protocol ensures the data privacy.Our auditing protocol incurs less communication cost between the auditor and the server. It also reduces the computing loads of the auditor by moving it to the server.We extend our auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model.

2. SYSTEM ARCHITECTURE

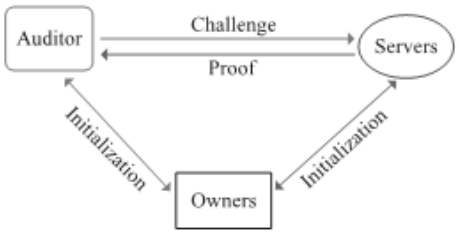

Fig. 1. System model of the data storage auditing

We consider an auditing system for cloud storage as shown in Fig.1, which involves data owners (owner), the cloud server (server), and the third-party auditor (auditor). The owners create the data and host their data in desired cloud.

The cloud server stocks the owners' data and provides the data access to users. The auditor is a trustworthy third-party that has expertise and capabilities to provide data storage auditing service for both the servers and owners. we introduce a novel data anonymization technique called slicing to improve the current state of the art. Slicing partitions the dataset both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets. Finally, within each bucket, values in each column are randomly permutated (or sorted) to break the linking between different columns.

Cloud Data Storage Model

The cloud storage model considering here is consists of three main components as illustrated in Fig. 2.

1) Cloud User: the user, who can be an individual or an organization originally storing their data in cloud and accessing the data.
2) Cloud Service Provider (CSP): the CSP, who manages cloud servers (CSs) and provides a paid storage space on its infrastructure to users as a service.
3) Third Party Auditor (TPA) or Verifier: the TPA or Verifier, who has expertise and capabilities that users may not have and verifies the integrity of outsourced data in cloud on behalf of users. Based on the audit result, the TPA could release an audit report to user.

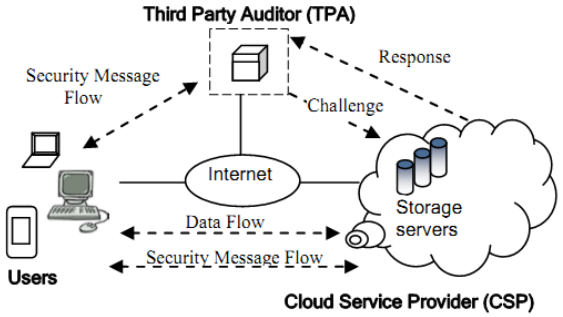

Fig. 2. Cloud Data Storage Model

3. SLICING ALGORITHM FOR BETTER DATA UTILITY:

Our algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning.

Attribute Partitioning algorithm

In this, we compute the correlations between pairs of attributes and then cluster attributes based on their correlations.

Measures of Correlation

Two widely-used measures of association are Pearson correlation coefficient and mean-square contingency coefficient. Pearson correlation coefficient is used for measuring correlations between two continuous attributes Although mean-square contingency coefficient is a chi-square measure of correlation between two categorical attributes. So, the mean-square contingency coefficient is used because most of our attributes are categorical. Given two attributes A 1 and A 2 with domain s v11, v12, ..., v1d1 and v21, v22, ..., v2d2 , respectively. Their domain sizes are thus d 1 an d2, respectively. The mean-square contingency coefficient between A 1 and A 2 is defined as: Here fi and fj are the fraction of occurrences of v1i and v2j in the data, correspondingly

$$\phi^2(A1, A2) = \frac{1}{\min\{d1, d2\}-1} \sum_{i=1}^{d1} \sum_{j=1}^{d2} \frac{(fij - fi.fj.)^2}{fi.fj.} \quad (1)$$

fij is the fraction of co-occurrences of v1i an d v2j in the data. Therefore,fi and f j are the marginal totals of

fij :fi . $= \sum_{j=1}^{d2}$ fij andf.j $= \sum_{i=1}^{d1}$ fij.

It can be shown that $0 \le \phi^2 = (A1, A2) \le 1$.

For continuous attribute, discretization can be applied to partition the domain of a continuous attribute into intervals and then treat the collection of interval values as a discrete domain. Discretization has been frequently used for decision tree summarization, classification, and frequent items set mining. We use equal-width discretization, that partitions an attribute domain into (some k) equal-sized intervals.

Attribute Clustering

Having computed the correlations for each pair of attributes, clustering to partition attributes into columns is used. In our algorithm, each attribute is a point in the clustering space. The distance between two attributes in the clustering space is defined as d(A1,A2) = 1 −φ2(A1,A2),which is in between of 0 and 1. Two attributes that are strongly-correlated will have a smaller distance between the corresponding data points in our clustering space. We choose the k-medoid method for the following reasons. Starting with first,, many present clustering algorithms (e.g., k-means) requires the calculation of the "centroids". But there is no notion of"centroids"in our setting where each attribute forms a data point in the clustering space. Second, k-medoid method is very robust to the existence of outliers (i.e., data points that are very far away from the rest of data).Third, the sequence in which the data points are examined does not affect the clusters computed from the k-medoid method. We use the well-known k-medoid algorithm PAM (Partition Around Medoids) [14]. PAM starts by an arbitrary selection of k data points as the initial medoids. In each subsequent step, PAM chooses one medoid point and one non-medoid point and swaps them as long as the cost of clustering reduces. Here, the clustering cost is calculated as the sum of the cost of each cluster, which is in turn measured as the sum of the distance from each data point in the cluster to the medoid point of the cluster. The time complexity of PAM is O(k(n −k)2). Thus, it is known that PAM suffers from high computational complexity for large datasets. However, the data points in our clustering space are attributes, rather than tuples in the micro data. Therefore, PAM will not have computational problems for clustering attributes.

Special Attribute Partitioning

In the above procedure, all attributes (including both QIs and SAs) are clustered into columns. The k-medoid method ensures that the attributes are clustered into k columns but does not have any guarantee on the size of the sensitive column Cc. In some cases, we may pre-determine the

number of attributes in the sensitive column to be α. The parameter α determines the size of the sensitive column Cc, i.e., |Cc| = α.

If α= 1, then |Cc| = 1, which means that Cc = {S}. And When c = 2, slicing in this case becomes equivalent to bucketization. If α>1, then |Cc| >1, the sensitive column also contains some QI attributes. We adapt the above algorithm to partition attributes into columns such that the sensitive column Cc contains αat-tributes. We first calculate correlations between the sensitive attribute S and each QI attribute. Then, we rank the QI attributes by the decreasing order of their correlations with S and select the top α−1 QI attributes. Now, the sensitive column Cc consists of S and the selected QI attributes. All other QI attributes form the other c −1 columns using the attribute clustering algorithm.

4. SECURE DYNAMIC AUDITING

Algorithm for Auditing Protocol

A storage auditing protocol contains of the following five algorithms:

1.KeyGen(λ) =(sKh,sKt, pKt)This key generation algorithm takes no input other than the implicit security parameter. It outputs a secret hash key sKh and a pair of secret-public tag key (sKt, pKt).

2. TagGen (M,skt,skh)=T. The tag generation algorithm takes as inputs an encrypted file M, the secret tag key skt, and the secret hash key skh. For each data block mi, it computes a data tag ti based on skh and skt. It outputs a set of data tags T={ti}iε[1,n]

3. Chall(Minfo)=C. The challenge algorithm take s as input the abstract information of the data Minfo (e.g., file identity, total number of blocks,etc.). It outputs a challenge C.

4. Prove (M,T,C)=P. The prove algorithm takes as inputs the file M, the tags T, and the challenge from the auditor C. It outputs a proof P.

5. Verify(C,P,skh,pkt, Minfo)=0/1. The verification algorithm takes as inputs P from the server, the secret hash key skh, the public tag key pkt, and the abstract information of the data Minfo. It outputs the auditing result as 0 or 1.

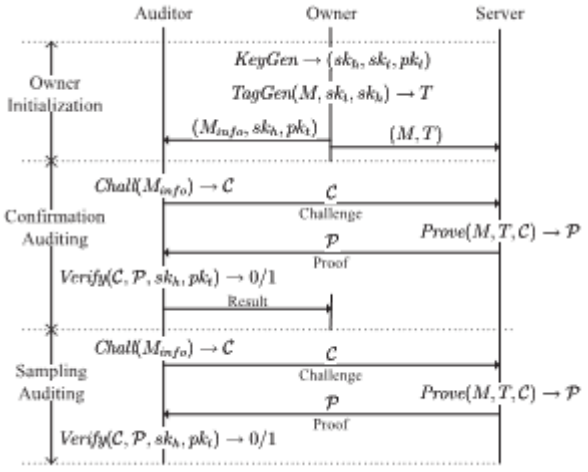

Fig. 3 Framework of privacy preserving Protocol

Suppose a file F has metadata components as F =(F1;... ;Fm). Each data component has its physical meanings and can be updated dynamically by the data owners. The data owner does not need to encrypt it for public data components, but for private data component, the data owner should to encrypt it with its corresponding key. Each data component Fk is divided into nk data blocks denoted as Fk =mk1;mk2 ;...;mknk. Due to the security purpose, the data block size should be restricted by the security parameter. Suppose the security level is set to be 160 bit (20 Byte), the data block size should be of size 20 Byte. A data component of 50-KByte will be divided into 2,500 data blocks and 2,500 data tags will be generated, which acquires 50-Kbyte storage overhead. By using the data fragment technique, next split each data block into sectors. The sector size is controlled by the security parameter. Then, generate one data tag for each data block that consists of sectors, such that less data tags are generated. In above, a 50-KByte data component only incurs50/s Kbyte storage overhead. In actual storage systems, the data block size can be various. That is, different data blocks could have different number of sectors. For example, if a data block mi will be frequently read, then si could be large, but for those frequently updated data blocks, may be comparatively small.

5. CONCLUSION

In this paper, we have studied the problem of data utility and integrity of data storage in cloud computing and proposed an efficient and secure dynamic auditing protocol. The proposed method presents a new approach called slicing to privacy-preserving data. Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats in cloud. we proposed an efficient and inherently secure dynamic auditing protocol which audits the data present in the cloud periodically and also whenever auditor wants to check it.Also dynamic data changes are also audited. . Furthermore, our auditing scheme incurs less communication cost and less computation cost of the auditor by moving the computing loads of auditing from the auditor to the server, which greatly improves the auditing performance and can be applied to large-scale cloud storage systems.

REFERENCES

[1] Kan Yang, Student Member , IEEE , and Xiaohua Jia, Fellow , IEEE,"An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing",IEEE TRANSAC TIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 9, SEPTEMBER 2013.

[2] Tiancheng Li, Ninghui Li, Jian Zhang, Ian Molloy Purdue University, West Lafayette, IN 47907, “Slicing: A New Approach to Privacy Preserving Data Publishing,” IEEE 2012 Transactions on Knowledge and Data Engineering, volume:24,Issue:3.

[3] T. Velte, A. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach, first ed., ch. 7. McGraw-Hill, 2010.

[4] J. Li, M.N. Krohn, D. Mazie`res, and D. Shasha, “Secure Untrusted Data Repository (SUNDR),” Proc. Sixth Conf. Symp. Operating Systems Design Implementation, pp. 121-136, 2004.

[5] G.R. Goodson, J.J. Wylie, G.R. Ganger, and M.K. Reiter, “Efficient Byzantine-Tolerant Erasure-Coded Storage,” Pro c. Int ’l Conf. Dependable Systems and Networks, pp. 135-144, 2004.

[6] V. Kher and Y. Kim, “Securing Distributed Storage: Challenges, Techniques, and Systems,” Proc. ACM Workshop Storage Security

and Survivability (StorageSS), V. Atluri, P. Samarati, W. Yurcik, L. Brumbaugh, and Y. Zhou, eds., pp. 9-25, 2005.

[7] L.N. Bairavasundaram, G.R. Goodson, S. Pasupathy, and J. Schindler, "An Analysis of Latent Sector Errors in Disk Drives," Proc. ACM SIGMETRICS Int'l Conf. Measurement and Modeling of Computer Systems, L. Golubchik, M.H. Ammar, and M. Harchol- Balter, eds., pp. 289-300, 2007.

[8] B. Schroeder and G.A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" Proc. USENIX Conf. File and Storage Technologies, pp. 1-16, 2007.

[9] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," Proc. USENIX Ann. Technical Conf., pp. 29-41, 2003.

[10] G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, "Provable Data Possession at Untrusted Stores," Proc. ACM Conf. Computer and Comm. Security, P. Ning, S.D.C. di Vimercati, and P.F. Syverson, eds., pp. 598-609, 2007.

[11] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Info rma tion Se curit y: Adva nces in Cr yptol ogy , J. Pieprzyk, ed., pp. 90-107, 2008.

[12] C.C. Erway, A. Ku¨ pc¸u¨ , C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. ACM Conf. Computer and Comm. Security, E. Al-Shaer, S. Jha, and A.D. Keromytis, eds., pp. 213-222, 2009.

[13] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[14] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.

[15] Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231-2244, Dec. 2012.

[16] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S. Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. ACM Symp. Applied Computing, W.C. Chu, W.E. Wong, M.J. Palakal, and C.-C. Hung, eds., pp. 1550-1557, 2011.

[17] K. Zeng, "Publicly Verifiable Remote Data Integrity," Proc. 10th Int'l Conf. Information and Comm. Security, L. Chen, M.D. Ryan, and G. Wang, eds., pp. 419-434, 2008.

[18] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, M. Matsui, ed., pp. 319-333, 2009.