International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 11, November - 2013

A Survey on Privacy Preserving Public Auditing for Secure Data Storage

Bhuvaneswari. M [1], Rohini. R [2], Preetha. B [3]

PG Scholar, Computer Science and Engineering, Vivekananda College of Engineering for Women, Tiruchengode, India [1]
Assistant Professor, Computer Science and Engineering, Vivekananda College of Engineering for Women, Tiruchengode, India [2]
Assistant Professor, Computer Science and Engineering, Vivekananda College of Engineering for Women, Tiruchengode, India [3]

Abstract: Cloud Computing is using hardware and software as computing resources to provide service through internet. We can access the data from anywhere, at any time on demand. The major problem in cloud data storage is security. So in order to provide high security, we proposed Privacy Preserving Auditing Protocol along with blow fish algorithm which enables an external auditor to audit user's outsourced data in cloud without reading the actual data content. We also improved efficiency by means of Batch Auditing.

Keywords: Batch Auditing, External auditor, Privacy Preserving Auditing, Security.

I. INTRODUCTION

The Cloud computing is a latest technology which provides several services through internet. The Cloud server allows user to store their data on a cloud without disturbing correctness and integrity of data. Cloud data storage has many benefits over local data storage. User can upload their data on cloud and can access those data anytime anywhere without any additional problem. User can upload their data on cloud without disturbing about storage and maintenance. But as data is stored at the remote place which in turn, users will get the confirmation about stored data. Hence Cloud data storage should have some mechanism which will agree storage correctness and integrity of data stored on a cloud. Cost is low or pay per usage basis. Hardware and software resources are easily offered without location independent.

The main advantage of storing data on a cloud is the relief of problem for storage management, universal data access with location independent and preventing capital expenditure on hardware, software and personal maintenance.

In cloud computing, cloud data storage have two entities such as cloud user and cloud service provider or cloud server. Cloud user is a person who stores huge amount of data on cloud server which is handled by the cloud service provider. A cloud service provider will offer services to cloud user. The major drawbacks in cloud data storage is to obtain correctness and integrity of data stored on the cloud. Cloud Service Provider (CSP) has to provide some form of mechanism through which user will get the confirmation that cloud data is secure or is stored as it is. No data loss or modification is done.

Privacy preserving is used to provide an trusted service sends does not reveal the key and the data that an trusted customer sends in response to an auditor that follows the protocol (honest, but curious) does not reveal the key. Security in cloud computing can be addressed in several ways as authentication, integrity, confidentiality. Data integrity or data correctness is another security drawbacks that needs to be considered. The scheme states that the data storage correctness can be exploit using SMDS (Secure Model for cloud Data Storage). The two kinds of hash function such as Secure Hash Algorithm (SHA1) for digital signature and Message Digest (MD5) is a cryptographic hash function which is used to check the data integrity.

The major goals of proposed schemes as

- The User needs to use best encryption method.
- Secure key management.
- Supple access right managements.
- Light weight integrity verification process for verifying the unauthorized change in the original data without need of local copy data.

The proposed scheme uses symmetric encryption which provides confidentiality, integrity, verification with low cost. It also provides enquiry for data owner and access control through which only authorized user can access the data. CSP may hide data loss or damage from users to maintain a reputation.

To achieve security, we can handover our data to a third outsource party who will identify the correctness and integrity of the cloud data. Hence Third party auditor (TPA) will check the data stored on the cloud based on the user's request.

We cannot achieve privacy; TPA can see the actual content stored on a cloud during the verifying phase. TPA itself may distribute the information stored in the cloud which violate security concept. To avoid the violation of security, Encryption technique is used where data is encrypted before storing it on the cloud. Hence using auditing with zero knowledge privacy technique where TPA will audit users data without seeing the contents. It uses existing public key based homomorphic linear authentication (HLA) [11], [12] that allows TPA to perform auditing without requesting for user data. It reduces communication and computation overhead.

II. LITERATURE SURVEY

A.PDP

In provable data possession (PDP) model, a third party stores a file. In this model, server will access small portions of the file in order to generate the proof. This is the first provably-secure scheme for remote data verifying. This techniques use homomorphic verifiable tags. Because of this property, tags calculated for multiple file blocks can be combined into a one value. The client computes tags for each block of a file early and then stores the file and its tags with a server. Afterwards the client can verify that the server possesses the file by producing a random challenge against a randomly selected set of file blocks. Using the queried blocks and their suitable tags, the server generates a proof of possession. The client is compromised of data possession, without actually having to access file blocks.

The database will be duplicated at multiple sites. Each and every site includes resource-sharing partners that interchange storage capacity to support reliability and scale. The location and physical proposal of these replicas are handled independently by each partner [13].

Our PDP schemes offer data format independence and put no limitation on the number of times the client can face the server to prove data possession.

PDP scheme uses symmetric-key encryption and MACs to check integrity of stored data. This method has low overhead. This scheme allows operations such as updating and deletions on the stored file.

Drawbacks:

- The computational requirements of remote data checking give problem to the remote storage sites.
- Performance is low.
- More time taken to verify.
- Even Partners may outsource storage to third-party storage server providers.

B. E-PDP Model

E-PDP verifies a 64MB file quickly. It generates proofs as soon as the disk produces data. Finally, E-PDP is 185 times faster than the previous secure protocol on 768 KB files. The model generates proofs by taking random sets of blocks from the server, which will reduce I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol will transmits only a small, constant amount of data in order to minimizes network communication. This method not using cryptographic computation. This scheme will give assurance by sampling the server's storage and provide a practical method to verify large data sets.

Drawbacks:

- Outsourcing data task will be done repeatedly.
- Computation cost for entire file is expensive.
- Due to Small latent errors or data corruptions, few bits or single blocks may be lost.
- Schemes increases computational burden to the server.

C. Proof of Retrievability" (POR) model

"Proof of Retrievability" (POR) model for ensuring possession and retreivability of files. Their scheme combines sentinels in a derivative of original file. The derivative is an encrypted version that has been redundantly coded for error correction. The redundancy prevents small errors and the auditing checks for a sentinel that catches large omissions in the file.

In this protocols, we reduce the burden of keeping these secret keys to a storage service. Since services are already in the business of preserving customers' data and privacy, the keys are safer with them. Keeping the data content private from the service is elective. A customer can keep the keys and encrypted data with the same service, thereby see-through the contents to that service and allowing it to provide additional features beyond storage like search. Otherwise, the customer can distinguish the keys and encrypted data onto non-colluding services to preserving complete privacy. The auditor is responsible for auditing and extracting both the encrypted data and the secret. Moreover, extraction in their scheme is a by-product of auditing keys. This protocol never see-through the secret key. Our schemes, reminiscent of Diffie-Hellman key exchange, rely on the discrete log assumption (DLA) for privacy.

This schemes divide the data into two parts, an encryption key and the encrypted data. This protocol allows an auditor to check both those pieces and extract those pieces without enlightening the underlying contents of either. The user needs not to maintain any long-term state (secret keys or hashes). The protocols for the encrypted data depend on cryptographic hashes and symmetric key encryption.

Drawbacks:

- Computing HMACs for the entire contents will create overhead.
- Schemes require the auditor to be honest and not collude with either party.
- Protocols for the encryption key assume that the computing the discrete log is difficult.
- Costly.

D. Compact Proof of Retrievability

This scheme using publicly verifiable homomorphic authenticators from BLS signatures [6] and provably secure in the random oracle model. Public retrievability is succeeded based on BLS construction and the proofs can be combined into a small authenticator value. The authors consider only static data files. This scheme extends PDP model [2] using rank-based authenticated skip lists in order to support provable updates to stored data files. This scheme eliminates the index information in the "tag" in block insertion to support updating operation computation in Ateniese's PDP model [2]. To achieve this, before the verification procedure, they use authenticated skip list data structure to authenticate the tag information of challenged or updated blocks first. Once a file block is inserted, the signatures of the file blocks should be recomputed with the new indexes. This limitation can be removed by the index information i in generating the signatures and use H(mi) as the tag for block mi instead of H(name||i) [1] or h(v||i) [3], so individual data operation on any file block will not affect the others. It supports both block verification and stateless verification. We extended the POR model [1] by using an elegant Merkle hash tree construction to support fully dynamic data operation.

Drawbacks:

- Data insertion is not supported.
- The scheme only supports limited number of integrity challenges and partially data updates.

E. Fine-Grained Data Access Control Model

In this scheme, we using the technique of hybrid encryption to preserve data files, i.e., we encrypt files using symmetric DEKs with KPABE. Using KP-ABE, we are able to suddenly enjoy fine-grained data access control and well-organized operations such as file creation/deletion and new user grant. To resolve the challenging issue of user revocation, we combine the technique of proxy re-encryption with KP-ABE and reduce most of the problem in Cloud Servers. We achieve this by keeping a partial copy of each user's secret key. When the data owner again specifies a certain set of attributes for the purpose of user revocation, he also produces corresponding proxy re-encryption keys and sends them to Cloud Servers. Cloud Servers, given these proxy re-encryption keys, can append user secret key components and re-encrypt data files accordingly without knowing the underlying data files. This enhancement releases the data owner from the possible huge computation overhead on user revocation. The data owner also does not need to always stay online. In order to save computation overhead of Cloud Servers on user revocation, we use lazy re-encryption technique and enable Cloud Servers to combine multiple successive secret key update or file re-encryption operations into one.

1) Key Policy Attribute-Based Encryption (KP-ABE)

KP KP-ABE [15] is a public key cryptography primitive for one-to-many communications. In KP-ABE, data are combined with attributes for each and public key component is defined. The encrypter combines the set of attributes to the message by encrypting it with the matching public key components. Each user is allocated with an access structure which is usually defined as an access tree over data attributes are threshold gates and leaf nodes are combined with attributes. User secret key is defined to echo the access structure so that the user is able to decrypt a cipher text if and only if the data attributes fulfil his access structure.

2) Proxy Re-Encryption (PRE)

Proxy Re-Encryption (PRE) is a cryptographic primitive in which a semi-trusted proxy is able to translate a cipher text encrypted under Alice's public key into another cipher text that can be opened by Bob's private key without seeing the original plaintext. More formally, a PRE scheme allows the proxy, given the proxy re-encryption key, to convert cipher texts under public key pka into cipher texts under public key pkb and vice versa. This scheme provides the advantages such as short time taken to start new services, Low maintenance costs, higher utilization through imagination and Easier disaster recovery.

Drawbacks:

- It does not provide adequate proof of data confidentiality.
- The complexity lies in file creation and user grant or revocation.

III. CONCLUSION

We finally proposed Blow Fish algorithm which will provide better privacy than the existing methods. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. In all, the Blowfish encryption algorithm will run 521 times to generate all the sub keys about 4KB of data is processed. The slow initialization of the cipher with each change of key, it is granted a natural protection against brute-force attacks, which doesn't really justify key sizes longer than 448 bits. It supports batch auditing where TPA will handle multiple users request at the same time which reduces communication and computation overhead. It allows TPA to audit user's data without knowing data Content. It provides security and increases performance of the system. Data dynamics support is achieved by replacing the index information i with mi in the computation of block authenticators and using Merkle hash tree. This scheme saves amount of auditing time. It describes the periodic verification for improving the performance of audit services. It achieves Audit-without-downloading,Verification-correctness, Privacy-preserving and High-performance.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "*Provable data possession at untrusted stores*," in Proc. of CCS'07, Alexandria, VA, October 2007.

[2] M. A. Shah, R. Swaminathan, and M. Baker, "*Privacy preserving audit and extraction of digital contents*," Cryptology ePrint Archive, Report 2008/186, 2008.

[3] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, *"Enabling public verifiability and data dynamics for storage security in cloud computing,*" in Proc. of ESORICS'09, volume 5789 of LNCS.Springer-Verlag, Sep. 2009, pp. 355–370.

[4] A. Juels and J. Burton S. Kaliski, "*Pors: Proofs of retrievability for large files*," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.

[5] Y. Dodis, S. P. Vadhan, and D. Wichs, *"Proofs of retrievability via hardness amplification*," in TCC, 2009, pp. 109–127.

[6] H. Shacham and B. Waters, "*Compact proofs of retrievability,"* in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.

[7] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan,"*Auditing to keep online storage services honest*," in Proc. Of HotOS'07. Berkeley, CA, USA: USENIX Association, 2007.

[8] 104th United States Congress, "*Health Insurance Portability and Accountability Act of 1996 (HIPPA),"*

[9] S. Yu, C. Wang, K. Ren, and W. Lou, "*Achieving secure,scalable, and fine-grained access control in cloud computing,"* in Proc. of IEEE INFOCOM'10, San Diego, CA, USA, March 2010.

[10] D. Boneh, B. Lynn, and H. hacham, "*Short signatures from the Weil pairing,*" J. Cryptology, vol. 17, no. 4, pp. 297–319,2004.

[11] A. L. Ferrara, M. Greeny, S. Hohenberger, and M. Pedersen,"*Practical short signature batch verification,*" in Proceedings of CT-RSA, volume 5473 of LNCS. Springer-Verlag, 2009, pp. 309–324.

[12] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik,"*Scalable and efficient provable data possession,*" in Proc. Of SecureComm'08, 2008, pp. 1–10.

[13] C.Wang, Q.Wang, K. Ren, andW. Lou, "*Ensuring data storage security in cloud computing*," in Proc. of IWQoS'09, July 2009, pp. 1–9.

[14] C. Erway, A. Kupcu, C. Papamanthou, and R.Tamassia, "*Dynamic provable data possession*," in Proc. of CCS'09, 2009,pp. 213–222.

[15] R. C.Merkle, "*Protocols for public key cryptosystems,*" in Proc. of IEEE Symposium on Security and Privacy, Los Alamitos, CA,USA, 1980.