Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 2, February 2015, pg.284 – 292

RESEARCH ARTICLE

Secure Erasure Code-Based Privacy-Preserving Public Auditing System

S.Ramya (ramyaseshachalam5194@gmail.com), K.B.Suganya (suganyababu@rocketmail.com), C.R.Sathya (sathyacr6894@gmail.com), Ms.V.Sathiya M.E. (deviviji2000@yahoo.co.in), Department of Computer Science and Engineering, Panimalar Engineering College, Chennai

ABSTRACT— Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centres, where the management of the data and services may not be fully trustworthy. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. *Several mechanisms have been used for privacy preserving of shared data in the cloud but these may fail to end the direct server attacks and other similar attacks. In these mechanisms data is stored only in one server. In this project, we ensure the integrity of the remote data in the cloud. We divide the data into blocks and thus we exploit the shared key generation algorithm which create key for each block and these blocks are encrypted and stored in multiple servers. We also consider the task of allowing a threshold proxy re-encryption, on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. Thus our mechanism improves the storage and security related problems. In addition, our mechanism also improves the security problem from unauthorized user by creating one time password with key generated to the data owner and this also provides dynamic data operations to data owner and privileged user.*

Index Terms— Public auditing, privacy-preserving, shared data, ***proxy re-encryption,*** *secure erasure code, cloud computing*

I. INTRODUCTION

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.

Cloud service providers offer users efficient and scalable data storage services with a much lower marginal cost than traditional approaches [2]. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes a standard feature in most cloud storage offerings, including Drop box, 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 [3], [4]. To make this matter even worse, cloud service providers may be reluctant to inform users about these data errors in order to maintain the reputation of their services and avoid losing profits [5]. Therefore, the integrity of cloud data should be verified before any data utilization, such as search or computation over cloud data [6].

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 (e.g., RSA [7]) or hash values (e.g., MD5 [8]) of the entire data. Certainly, this conventional approach is able to successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is in doubt [9].

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 users amounts of computation and communication resources, especially when data have been corrupted in the cloud. Besides, many uses of cloud data (e.g., data mining and machine learning) do not necessarily need users to download the entire cloud data to local devices [2]. It is because cloud providers, such as Amazon, can offer users computation services directly on large-scale data that already existed in the cloud.

Recently, many mechanisms [9], [10], [11], [12], [13] 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 [5]. 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 [9]. 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 services.

II. EXISTING WORK

A cloud computing environment, failure is the norm, and nodes may be Upgraded, replaced, and added in the system.

Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing.

Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data.

Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging.

Secondly, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure **storage correctness** under dynamic data update is hence of paramount importance.

In the existing system privacy preserving mechanism that supports public auditing on shared data stored in the cloud.

It 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 this system, data is divided into blocks and key is assigned and these blocks are stored in single server. It use ring signatures to compute verification metadata needed to audit the correctness of shared data.

With this the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. This mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one.

Disadvantage: It does not support storage related problems since the entire data is stored in the single server. It lacks the security related problems since it is easy for hacker to get the data from one server itself.

III. PROPOSED WORK

We propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud.

We rely on erasure 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.

By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization. In our mechanism we divide the data into blocks and thus we exploit the shared key generation algorithm which create key for each block and these blocks are encrypted and stored in multiple servers. Thus our mechanism improves the storage and security related problems. Our mechanism also improves the security problem from unauthorized user by creating one time password with key generated to the data owner and this also provides dynamic data operations to data owner and privileged user.

Advantage: It supports dynamic data operations for both data owner and privileged user. It avoids direct server attack and similar attacks. It improves the storage space. Each time the data owner gets the notification messages if any change is made.

IV. SYSTEM MODEL

The system model in this paper 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 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.

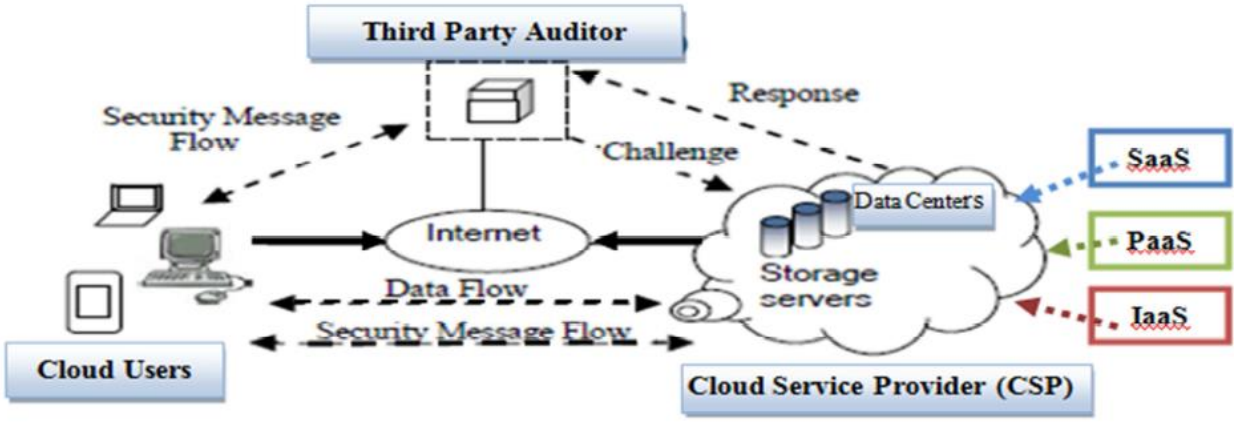


Figure1: Cloud Data Storage Model

ARCHITECTURE DIAGRAM

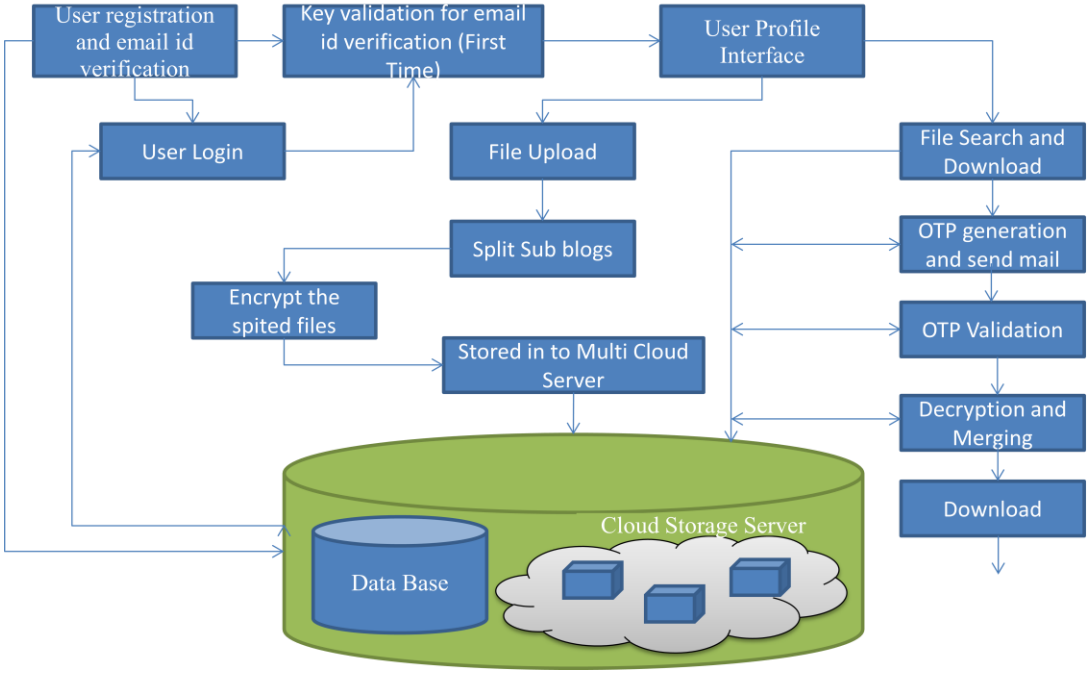


Figure 2

V. Secure Erasure Algorithm

Given a signal of m blocks, recode to n, where n > m. Optimal: reconstruct signal given any m unique blocks.

Suboptimal: Reconstruct signal using (1+e) m unique blocks. Rate r=m/n, and storage overhead is 1/r.

Optimal erasure codes have the property that any k out of the n code word symbols are sufficient to recover the original message (i.e., they have optimal reception efficiency). Optimal erasure codes are maximum distance separable codes (MDS codes).

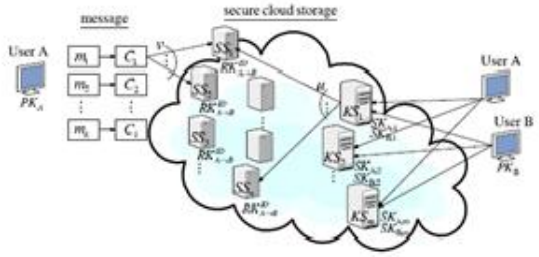


fig.3 A general system model of our work.

A cloud storage system, consisting of a collection of storage servers, provides long-term storage services over the Internet. Storing data in a third party's cloud system causes serious concern over data confidentiality.

General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data.

Parity check: It is the special case where n = k + 1. From a set of k values, a checksum is computed and appended to the k source values: The set of k + 1 values is now consistent with regard to the checksum. If one of these values, , is erased, it can be easily recovered by summing the remaining variables.

$$v_e = -\sum_{i=1, i\neq e}^{k+1} v_i .$$

AES or DES:

It is a web tool to encrypt and decrypt text using *AES encryption* algorithm. You can chose 128, 192 or 256-bit long key size for encryption and decryption. The result of the process is downloadable in a text file.

AES Algorithm

Key Expansions—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

In Initial Round, AddRoundKey—each byte of the state is combined with a block of the round key using bitwise xor.

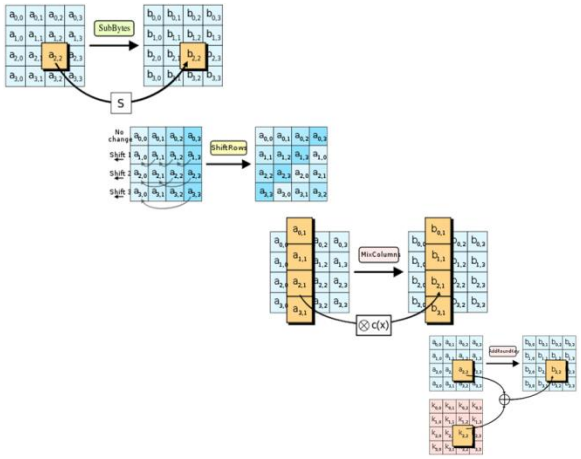
The other Rounds are SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

Shift Rows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

AddRoundKey—Final Round: (no MixColumns) SubBytes, ShiftRows, AddRoundKey.

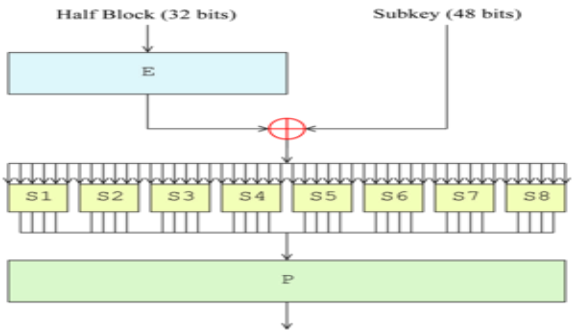
fig.4



Encryption has become a part and parcel of our lives and we have accepted the fact that data is going to encrypted and decrypted at various stages.

However, there is not a single encryption algorithm followed everywhere. There are a number of algorithms existing, and I feel there is a need to understand how they work.

So this text explains a number of popular encryption algorithms and makes you look at them as mathematical formulas.



If you want to encrypt a text put it in the white textarea above, set the key of the encryption then push the *Encrypt* button.

The result of the *encryption* will appear in base64 encoded to prevent character encoding problems. If you want to decrypt a text be sure it is in base64 encoded and is encrypted with *AES* algorithm.

Put the encrypted text in the white text area, set the key and push the *Decrypt* button.

When you want to encrypt a confidential text into a decryptable format, for example when you need to send sensitive data in e-mail.

The decryption of the encrypted text it is possible only if you know the right password.

AES (acronym of Advanced Encryption Standard) is a symmetric encryption algorithm.

The algorithm was developed by two Belgian cryptographer Joan Daemen and Vincent Rijmen. AES was designed to be efficient in both hardware and software, and supports a block length of 128 bits and key lengths of 128, 192, and 256 bits.

AES encryption is used by U.S. for securing sensitive but unclassified material, so we can say it is enough secure.

VI. CONCLUSION

In this paper, we propose a privacy-preserving public auditing system for data storage security in cloud computing. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving

public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

REFERENCES

[1] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf. Cloud Computing, pp. 295-302, 2012.
[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.
[3] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.
[4] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.
[5] 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.
[6] B. Wang, M. Li, S.S. Chow, and H. Li, "Computing Encrypted Cloud Data Efficiently under Multiple Keys," Proc. IEEE Conf. Comm. and Network Security (CNS '13), pp. 90-99, 2013.
[7] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.
[8] The MD5 Message-Digest Algorithm (RFC1321). https://tools.ietf.org/html/rfc1321, 2014.
[9] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-610, 2007.
[10] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.
[11] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS'09), pp. 213-222, 2009.
[12] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.
[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS'09), pp.1-9,2009.

