Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 13 Issue 4 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Survey on Efficient Audit Service to Ensure Data Integrity in Cloud Environment

By Jaspreet Kaur & Jasmeet Singh
Punjab Techincal University

Abstract - Cloud computing is an internet based computing which provides different users an opportunity to store their data in the cloud. While data outsourcing relieves the owner of the burden of the local data storage and maintenance but as they have no longer physical possession of outsourced data makes data integrity protection a very challenging task. This paper explores the secure cryptographic hash function along with some other techniques that can be used by TPA to ensure the integrity of data stored in the cloud at regular intervals or on user request.

Keywords : *cloud, data integrity, secure hash algorithm, station-to-station protocol, third party auditor.*

GJCST-C Classification: *H.2.7*

Strictly as per the compliance and regulations of:

Survey on Efficient Audit Service to Ensure Data Integrity in Cloud Environment

Jaspreet Kaur [α] & Jasmeet Singh [σ]

Abstract - **Cloud computing is an internet based computing which provides different users an opportunity to store their data in the cloud. While data outsourcing relieves the owner of the burden of the local data storage and maintenance but as they have no longer physical possession of outsourced data makes data integrity protection a very challenging task. This paper explores the secure cryptographic hash function along with some other techniques that can be used by TPA to ensure the integrity of data stored in the cloud at regular intervals or on user request.**

Keywords : cloud, data integrity, secure hash algorithm, station-to-station protocol, third party auditor.

I. Introduction

Cloud computing is a general term for anything that involves delivering hosted services over the internet. These services are broadly divided into three categories:

a) SaaS (Software as a Service)

SaaS is a model of software deployment where consumer use the provider's application running on a cloud infrastructure through a thin client interface.

b) PaaS (Platform as a Service)

The platforms used to develop, build and test applications are provided to the consumer by the cloud.

c) IaaS (Infrastructure as a Service)

This is a model in which service provider owns the equipments used to support operations, including storage, hardware, servers and networking components. The client typically pays on a per-use basis. [3]

Cloud computing is becoming more and more popular now a days, where data is outsourced into the cloud. Its pros include relief of the burden of the storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software and personnel maintenance. However, outsourcing data introduces new security issues. [4]

The first issue is data integrity. In computer security, data integrity can be defined as "the state that exists when computerized data is the same as that in the source document and has not been exposed to accidental or malicious alterations or destruction". Integrity of data stored at the entrusted cloud server is not guaranteed. [1, 4]

The second issue is unfaithful cloud server providers (CSP). There are many reasons why CSPs are not always trustworthy like, for saving money and storage space, CSPs may discard the data that has not been accessed for long time (which belongs to ordinary client) or sometimes even hide data losses or corruptions to maintain a reputation.

As data owners outsourced their data to the cloud and do not maintain the local copy, so simple cryptographic measures cannot be used directly to monitor the integrity of data. Also simply downloading the data for monitoring integrity is not a viable solution as it incurs high cost of input/output and transmission across the network.

To check the integrity of data only while accessing is not sufficient as the un-accessed data left unchecked from the verification process and it might get too late to recover any loss or damage to the unchecked data.

In addition, from the system usability point of view, data owners should be able to just use cloud storage as if it is local, without worrying about the need to verify the correctness of data. Therefore, an external third party auditor (TPA) is required. [4]

The TPA is an independent authority that has expertise and capabilities to monitor the integrity of cloud data outsourced by the client and informs him about data corruption or loss, if any.[6] To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met:

1. TPA should be able to audit the cloud data storage efficiently without asking for the local copy of data thereby reducing the on-line burden of cloud users.
2. The third party auditing process should not affect user data privacy.[7]

Figure 1 represents the cloud storage architecture which consists of three different entities: Users, Cloud Storage Server and Third Party Auditor (TPA).

Author α : M.TECH (Computer Science and Engineering) RIMT-IET, Mandi Gobindgarh, Punjab (India). E-mail : kkhushi12@yahoo.co.in
Author σ : Asst. Professor (CSE Department) RIMT-IET, Mandi Gobindgarh, Punjab (India). E-mail : jasmeetgurm@gmail.com

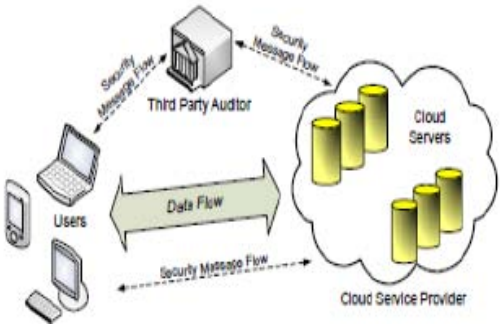

Figure 1 : Architecture of cloud data storage service

1. *User*

Users are the data owners who have the large amount of data to be stored in the cloud and access them when needed. Users depend on cloud storage server for data computation.

2. *Cloud Storage Server*

A cloud storage server (CSS) is an entity that is managed by cloud service provider (CSP). It provides space to the user for data storage and computation.

3. *Third Party Auditor (TPA)*

An TPA is an entity who has capabilities to verify the integrity of cloud data on behalf of the user's request. [1]

II. Related Work

In [9] author implemented mechanism in which integrity is checked at 2 sides-by cloud server (for inside attack) and by TPA (for outside attack) using digital signature with MD5.But we analyze that as cloud server is not trustworthy so data should not be exposed to the cloud. Instead user can rely on TPA who has expertise in verification process and moreover user can authenticate TPA to check for its credibility.

In [5] author proposed auditing scheme in which TPA checks the integrity of data outsourced by user in the cloud. For monitoring integrity auditor takes cipher text from the cloud and generates original data from it using XOR. TPA calculates the hash value using SHA-1 algorithm and compares it with the hash value taken from user. If the value matches it is ensured that data is safe, otherwise tampered.

This approach relieves the user from worry of downloading data and verifying its correctness. This also preserves user resources that could be consumed otherwise. But there is lack of entity authentication between user and TPA which is necessary to trust third party. So we analyze that it could be better to use modified version of Diffie-Hellman i.e. STS (Station-To-Station) protocol in place of Diffie-Hellman which provides entity authentication along with key generation. Because of the security flaws (Collision Attacks) found in SHA-1 it is preferable to use SHA-2 which is more secure and strong.

III. Techniques

a) Station-to-Station protocol(STS)

In public-key cryptography, the Station-to-Station (STS) protocol is a cryptographic key agreement scheme consists of Diffie-Hellman key establishment followed by an exchange of authentication signatures. In this protocol, we assume that the parameters used for the key establishment are fixed and known to all users.

i. *STS Setup*

The following data must be generated before initiating the protocol:-

- An asymmetric signature key pair for each party-Required for authentication. The public portion of this key pair may be shared prior to session establishment.
- Key establishment parameter-The specification of a particular cyclic group and the corresponding primitive element α. These parameters may be public.

Sharing this data prior to the beginning of the session lessens the complexity of the protocol.

ii. *Basic STS*

Supposing all setup data has been shared, the STS protocol proceeds as follows: (All exponentials are in the group specified by p)

- Alice generates a random number x and computes the exponential α^x and send it to Bob.
- Bob generates a random number y and computes the exponential α^y.
- Bob computes the shared secret key $K = (\alpha^x)^y$.
- Bob concatenates the exponentials (α^y, α^x) (order is important), signs them using his asymmetric key B, and then encrypts them with K. He sends the cipher text along with his own exponential α^y to Alice.
- Alice computes the shared secret key $K = (\alpha^y)^x$.
- Alice decrypts and verifies Bob's signature.
- Alice concatenates the exponentials (α^x, α^y) (order is important), signs them using her asymmetric key A, and then encrypts them with K. She sends the cipher text to Bob.
- Bob decrypts and verifies Alice's signature.

Alice and Bob are now mutually authenticated and have a shared secret. This secret, K, can then be used to encrypt further communication. The basic form of the protocol is formalized in the following three steps as shown in figure 2. [11]

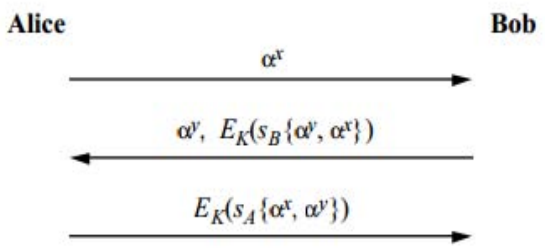

Figure 2 : Station to Station Protocol Steps

b) Exclusive or Operation

XOR is an operation in which same bits produce a resultant bit as 0 whereas different bits produce a resultant 1. In this, a XOR operation is done at original text along with secret value which gives cipher text. The original text can be obtained by performing an XOR operation between the secret key and resultant cipher text.

c) SHA-2

SHA stands for Secure Hash Algorithm. SHA-2 is the collective name of one-way hash functions developed by the NIST. SHA-256, SHA-384, and SHA-512 pertain to hashes whose outputs are 256 bits, 384 bits and 512 bits, respectively.

A hash function is an algorithm that transforms (hashes) an arbitrary set of data elements into a single fixed length value (the hash). The computed hash value may then be used to check the integrity of copies of the original data without providing any means to derive the source (irreversibly). A hash value therefore may be freely distributed or stored as it is only used for comparative purposes. SHA-2 features a higher level of security than its predecessor, SHA-1.

The comparisons between the SHA parameters are shown in table 1.

	sha-1	sha-256	sha-384	sha-512
message digest size	160	256	384	512
message size	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
block size	512	512	1024	1024
word size	32	32	64	64
number of steps	80	64	80	80
security	80	128	192	256
notes : 1.all sizes are measured in bits 2. security refers to the fact that a birthday attack on a message digest of size n produces a collision with a work factor of approximately $2^{n/2}$				

Table 1 : Comparison of SHA Parameters

The security provided by a hashing algorithm is entirely dependent upon its ability to produce a unique value for any specific set of data. When a hash function produces the same hash value for two different sets of data then a collision is said to occur. Collision raises the possibility that an attacker may be able to computationally craft sets of data which provide access to information secured by the hashed values of pass codes or to alter computer data files in a fashion that would not change the resulting hash value and would thereby escape detection. A strong hash function (e.g. SHA-2) is one that is resistant to such computational attacks. [10]

An overview of SHA-256 is given here, and then the differences between SHA-256 and the other members of the SHA-2 family are outlined. The SHA-256 algorithm essentially consists of 3 stages: (1) message padding and parsing; (2) expansion; and (3) compression.

i. *Message Padding and Parsing*

The binary message to be processed is appended with a '1' and padded with zeros until its length 448 mod 512. The original message length is then appended as a 64-bit binary number. The resultant padded message is parsed into N 512-bit blocks, denoted $M^{(1)}$, $M^{(2)}$, ……. $M^{(N)}$. These $M^{(i)}$ message blocks are passed individually to the message expander.

ii. *Message Expansion*

The functions in the SHA-256 algorithm operate on 32-bit words, so each 512-bit $M^{(i)}$ block from the padding stage is viewed as 16 32-bit blocks denoted $M_t^{(i)}$, $0 \leq t \leq 15$.The message expander (also called the message scheduler) takes each $M^{(i)}$ and expands it into 64 32-bit W_t blocks.

iii. *Message Compression*

The W_t words from the message expansion stage are then passed to the SHA compression function, or the 'SHA core'. The core utilizes 8 32-bit working variables labeled A, B, . . . , H, which are initialized to predefined values $H_0^{(0)}$–$H_7^{(0)}$ at the start of each call to the hash function. Sixty-four iterations of the compression function are then performed and intermediate hash value $H^{(i)}$ is calculated:

$$H_0^{(i)} = A + H_0^{(i-1)},\ H_1^{(i)} = B + H_1^{(i-1)},\ \ldots,\ H_7^{(i)}$$

The SHA-256 compression algorithm then repeats and begins processing another 512-bit block from the message padder. After all N data blocks have been processed, the final 256-bit output, $H^{(N)}$, is formed by concatenating the final hash values:

$$H^{(N)} = H_0^{(N)} \,\&\, H_1^{(N)} \,\&\, H_2^{(N)} \,\&\, \ldots \,\&\, H_7^{(N)}$$

iv. *SHA-2 Algorithm Differences*

The SHA-512 algorithm has a similar structure to the SHA-256 algorithm, where: (i) it processes messages in blocks of 1024 bits rather than 512 bits; (ii) it uses 64-bit operations instead of 32-bit operations; (iii) it iterates its compression function 80 times rather than 64 times, but are otherwise similar in structure.[8]

IV. Conclusions

Cloud Computing today is the beginning of network based computing over internet in force. So monitoring integrity of cloud data storage is of critical importance. For this, third party auditor who has expertise in verification process can monitor integrity on behalf of user. The techniques mentioned above can be used to achieve the auditing task efficiently. We believe that monitoring integrity of cloud storage data is very much needed as data in cloud is not secure.

References Références Referencias

1. Reenu Sara Georeg, Sabitha S," Survey on Data Integrity in Cloud Computing", International Journal

of Advanced Research in Computer Engineering & Technology (IJARCET) Vol. 2, Issue 1, January 2013.

2. K. Govinda, V. Gurunathprasad, H. Sathishkumar, "Third Party Auditing for Secure Data Storage in Cloud through Digital Signature using RSA", International Journal of Advanced Scientific and Technical Research", Vol. 4, Issue 2, August 2012.
3. Akkala Saibabu, T. Satyanarayana Murthy, "Security Provision in Publicly Auditable Secure Cloud Data Storage Services using SHA-1 Algorithm", International Journal of Computer Science and Information Technologies (IJCSIT) Vol. 3(3), 2012.
4. Changsheng Wan, Juan Zhang, Zhongyuan Qin, "A XOR based Public Auditing Scheme for Proof-of-Storage".
5. K. Govinda, E. Sathiyamoorthy, "Data Auditing in Cloud Environment using Message Authentication Code", International Conference on Emerging Trends on Advanced Engineering Research (ICETT), 2012.
6. Muralikrishnan Ramane, Bharath Elangovan, "A MetaData Verification Scheme for Data Auditing in Cloud Environment", International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.4, August 2012.
7. Lingaraj Dhabale, Priti Pavale, "Providing Secured Data Storage by Privacy and Third Party Auditing in Cloud", International Conference on Computing and Control Engineering (ICCCE 2012), 12 & 13 April, 2012.
8. Robert P. McEvoy, Francis M. Crowe, Colin C. Murphy and William P. Marnane, "Optimisation of the SHA-2 Family of Hash Functions on FPGAs".
9. Dalia Attas, Omar Batrafi," Efficient integrity checking technique for securing client data in cloud computing", International Journal of Electrical & Computer Sciences (IJECS-IJENS), Vol: 11, No: 05.
10. http://en.wikipedia.org/wiki/SHA-2
11. http://en.wikipedia.org/wiki/Station-to-Station_protocol