Secrecy Archiving Public Ascertaining for Immune Cloud Storage

Vidya B. M
Dept. of CSE
SDM College of Engineering and Technology
Dharwad-580002

Nita Kakhandaki
Asst. Professor, Dept. of CSE
SDM College of Engineering and Technology
Dharwad-580002

***Abstract*— Without the burden of local data storage and maintenance, with the help of cloud storage, users can remotely store their data. They can also enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. The data integrity protection in Cloud Computing is a formidable task, since the users with constrained computing resources do not physical possession the outsourced data. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public ascertability for cloud storage is of critical importance so that users can resort to a third party ascertor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the ascertaining process should bring in no vulnerabilities towards user data secrecy, and introduce no additional online burden to the user. This paper proposes an immune cloud storage system supporting secrecy archiving public ascertaining. We further extend our result to enable the TPA to perform ascerts for multiple users simultaneously and efficiently.**

Keywords*— *Data storage; secrecy archiving; public ascertability; cryptographic protocols; Coud Computing

I. INTRODUCTION

Cloud Computing is the next generation information technology architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [1]. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [2]. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards user's outsourced data. Since 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. Firstly, although infrastructures under the cloud are much powerful and reliable than personal computing devices, they are still facing broad range of both internal and external threats for data integrity [3]-[7]. Secondly, there do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding the status of their outsourced data [8]-[10].

Sometimes the data stored in the cloud is so important that the clients must ensure that it is not lost or corrupted. It is easy to check data integrity after completely downloading the data to be checked. Downloading large amounts of data just for checking data integrity is a waste of communication bandwidth. Hence, a lot of works have been done on designing remote data integrity checking protocols, which allow data integrity to be checked without completely downloading the data. Remote data integrity checking was first introduced, which independently proposed RSA-based methods for solving this problem. Subsequently a remote storage ascertaining method based on pre-computed challenge-response pairs were proposed. These works focused on providing three advanced features for remote data integrity checking protocols: data dynamic, public verifiability and privacy against ascertors. Thus, enabling public ascertability for cloud storage is of critical importance, so that users can resort to a third party ascertor (TPA) to check the integrity of outsourced data and also to enable TPA to perform ascerts for multiple users simultaneously and efficiently.

II. PROBLEM DEFINITION

A. Existing system

In the existing system, the notion of public ascertability has been proposed in the context of ensuring remotely stored data integrity under different system and security models. Public ascertability allows an external party, in addition to the user itself, to verify the correctness of remotely stored data. However most of these schemes do not consider the secrecy protection of users against external ascertors. Indeed, they may potentially reveal user's data to ascertors. This severe drawback greatly affects the security of protocols in Could

Computing. From the perspective of protecting the data, users do not want this ascertaining process introducing new vulnerabilities namely unauthorized information leakage towards their own security [13].

Disadvantages of exiting system:

- Although the infrastructures under the cloud are much powerful and reliable than personal computing devices, they are still facing broad range of both internal and external threat
.
- There exist various motivations for CSP to behave unfaithfully towards the cloud users regarding their outsourced data status.

- Simply downloading all the data for its integrity verification is not a practical solution, due to the expensiveness in I/O and transmission cost across the network. It is often insufficient to detect the data corruption only when accessing the data, as it does not give user correctness assurance for those unaccessed data and might be too late to recover the data loss or damage.

- Encryption does not completely solve the problem of protecting data secrecy against third party, but reduces it to the complex key management domain. Unauthorized data leakage still remains possible due to the potential exposure of decryption keys [11].

B. Motivation

- Ascertaining of cloud systems is gaining importance because of
    1. A lot of sensitive and valuable processes, business functions and data moving to cloud.

    2. Ascertaining enables identification of threats or risks and speeds up time to response.

- Cloud users want security and secrecy protections for
heightened awareness of cloud failures due to system problems and hacking.

C. Proposed system

Public ascertaining system supports an external ascertor to ascertain the user's outsourced data without learning knowledge on the data content. The proposed system utilizes a technique of public key based homomorphic linear authenticator and random masking [8], [10], [12]. This system achieves batch ascertaining, where multiple delegated ascertaining tasks from different users can be performed simultaneously by the TPA. The system supports dynamic operations on data blocks i.e. data update, append and delete. This system also enables retrieval of modified data content and user's outsourced data content.

Cloud computing components are classified as:

- Cloud User (CU)
- Cloud service provider (CSP) and cloud server(CS)
- Third party ascertor (TPA)

Public ascertaining system runs with 2 phases:

- Setup phase
- Ascertain phase

D. Advantages of proposed system

- The proposed system motivates the public ascertaining system of data storage security in computing and provides a secrecy archiving ascertaining system. This scheme enables an external ascertor to ascert user's cloud data without learning the data content.

- This system is the first to support scalable and efficient secrecy archiving public ascertaining in cloud storage. System also achieves batch ascertaining where multiple delegated ascertaining tasks from different users can be performed simultaneously by the TPA in a secrecy-archiving manner.

E. Design goals of proposed system

- Public ascertability: Allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of whole data or introducing additional online burden to the users.

- Storage corrections: To ensure that there exists no cheating cloud server that can pass TPA's ascert without indeed storing user's data intact.

- Secrecy archiving: To ensure that the TPA cannot derive user's data content from information collected during ascertaining process.

- Batch ascertaining: To enable TPA with immune and efficient ascertaining capability to cope with multiple ascertaining delegations from possible large number of different users simultaneously.

- Light-weight: To allow TPA to perform ascertaining with minimum communication and computation overhead.

III. SYSTEM MODELING AND DESIGN

A. General Architecture

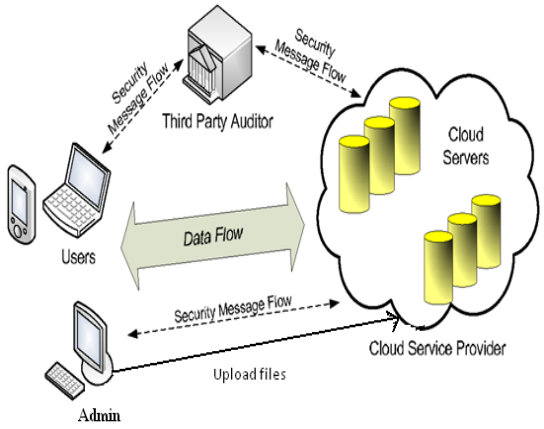

Fig 1: The architecture of cloud data storage service

- U: cloud user has a large amount of data to store in the cloud.
- Admin: Uploads, deletes, and corrects files to/from cloud server.
- CS: cloud server which is managed by the CSP and has significant data storage and computing power.
- TPA: third party ascertor has expertise and capabilities that U and CSP don't have. TPA is trusted to assess the CSP's storage security upon request from U.

Public ascertaining system runs with 2 Phases:

- Setup Phase
- Ascertain Phase

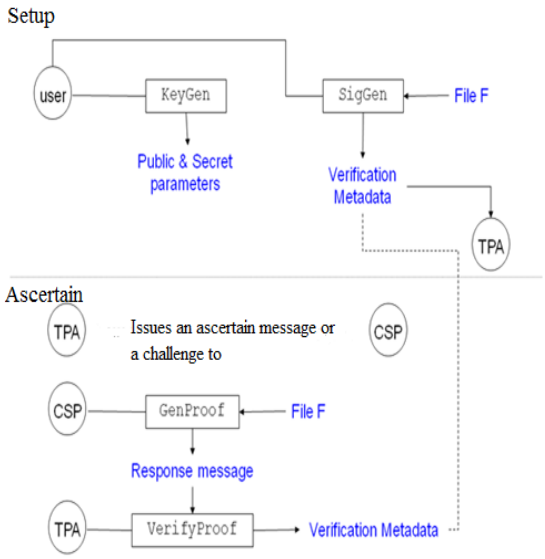

Fig 2: Setup and Ascertain Phase

B. Algorithms

Consists of four algorithms (KeyGen, SigGen, GenProof VerifyProof)

- KeyGen: key generation algorithm that is run by the user to setup the scheme.
- SigGen: used by the user to generate verification metadata, which may consist of MAC, signatures or other information used for ascertaining.
- Genproof: run by the cloud server to generate a proof of data storage correctness.
- VerifyProof: run by the TPA to ascert the proof from the cloud server.

IV. IMPLEMENTATION

MODULE DESCRIPTION

1. Metadata Generation
2. Data Dynamics
   - Insertion
   - Modification
   - Deletion
3. Batch ascertaining
4. Secrecy against Third Parity Verifiers

1. Metadata generation

The art of protecting information by transforming it (encrypting it) into an unreadable format is called cipher text. Only those who possess a secret key can decipher (or decrypt) the message into plain text. Encrypted messages can sometimes be broken by cryptanalysis, also called code breaking, although modern cryptography techniques are virtually unbreakable.

RSA ALGORITHM

RSA involves a public key and private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated in the following way:

1. Distinct prime numbers p and q are chosen.
   For security purposes, the integer's p and q should be chosen at random, and should be of similar bit-length.
2. Compute n = pq.
   Where 'n' is used as the modulus for both the public and private keys.
3. Compute φ (n) = (p – 1) (q – 1), where φ is Euler's totient function.
4. Select an integer e such that 1< e <φ (n) and greatest common divisor of (e, φ (n)) = 1; i.e., e and φ(n) are co-prime.
   'e' is released as the public key exponent.
5. Determine d as:

$$d \equiv e^{-1} \pmod{\varphi(n)}$$

i.e., d is the multiplicative inverse of e mod φ (n).

Encryption:

Encryption is the process of converting plain text into ciphertext

$$c=m^e \pmod n$$

Decryption:

Decryption is the process of converting ciphertext into plain text

$$m= c^d \pmod n$$

BLS SIGNATURE ALGORITHM

A signature scheme consists of three functions, generate, sign and verify.

Key generation:

The key generation algorithm selects a random integer χ in the interval $[0, r-1]$. The private key is χ. The holder of the private key publishes the public key, *y*.

Signing:

Given the private key χ and some message *m*, we compute the signature by hashing he
bitstring *m* as h=H(m), We output the signature $\sigma=h^{\chi}$.

Verification:

Given a signature σ and a public key g^x, we verify that,

$$e(\sigma, g) = e(H(m)\, g^x)$$

MD5 MESSAGEDIGEST ALGORITHM

Algorithm takes as input a message of arbitrary length and produces as output a 128 bit "fingerprint" or "message digest" of the input. Consider a b-bit message as input, and that we need to find its message digest.

1) Append padded bits: The message is padded so that its length is congruent to 448, modulo 512 i.e. extended to 64 bits shy of being of 512 bits long. A single "1" bit is appended to the message, and then "0" bits are appended so that length in bits equal 448 modulo 512.

2) Appended length: A 64 bit representation of b is appended to the result of previous step. The resulting message has a length that is an exact multiple of 512 bits.

3) Initialize MD buffer: A four-word buffer (A, B, C, D) is used to compute the message digest. Each of A, B, C, D is a 32-bit register.
The registers are initialized to the following hexadecimal values:

Word A: 01 23 45 67
Word B: 89 ab cd ef
Word C: fe dc ba 98
Word D: 76 54 32 10

4) Process message in 16-word blocks: Four auxiliary functions that take as input three 32-bit words and produce as output one 32-bit word.

F (X, Y, Z) = XY **V** not(X) Z
G (X, Y, Z) = XZ **V** Y not (Z)
H (X, Y, Z) = X xor Y xor Z
I (X, Y, Z) = Y xor (X **v** not (Z))

If the bits of X,Y and Z are unbiased, the each bit of F(X, Y, Z), G(X, Y, Z), H(X, Y, Z) and I(X, Y, Z) will be independent and unbiased.

5) Output: The message digest produced as output is A, B, C, D. i.e. Output begins with low-order byte of A, and end with the high-order byte of D.

2. Data dynamics

Data dynamics means after owner stores their data at the remote server, clients can dynamically update their data at later times. The main operations are insertion, modification and deletion

- Insertion: The client can insert anything on the file.
- Deletion: The client can delete anything on the file.
- Modification: The client can modify anything on the file.

3.

3. Batch ascertaining

With the establishment of secrecy archiving public ascertaining in cloud computing, TPA may concurrently handle multiple ascertaining delegations upon different user's requests. The individual ascertaining of these tasks for TPA can be tedious and very inefficient. Batch ascertaining not only allows TPA to perform the multiple ascertaining tasks simultaneously, but also greatly reduces the computation cost on the TPA side.

4. Secrecy against third party ascertor

Under the semi-honest model, a third party ascertor cannot get any information about the client's data from the system execution. Hence, the system is private against third party ascertor. If the server modifies any part of the client's data, the client should be able to detect it. Furthermore, any third party ascertor should also be able to detect it. In case a third party ascertor who verifies the integrity of the client's data, the data should be kept private against the third party ascertor too.

V. EXPERIMENTAL ANALYSIS

Whenever administrator uploads a file to the cloud, the time of uploading the file is recorded. After the file is inserted again the corresponding uploaded time is recorded. The difference between the uploaded time and uploading time is calculated and plotted on the graph.

Long time=System.currentTimeMills ()

The above formula is used to find the values for column time in the table below.

File Name	Time (ms)
Dbms.txt	9
ComputerGraphics.txt	10
SoftwareEnigeneering.txt	8
ImageProcessing.txt	11
ComputerNetworks.txt	12

Table 1: Time taken for uploading files to cloud

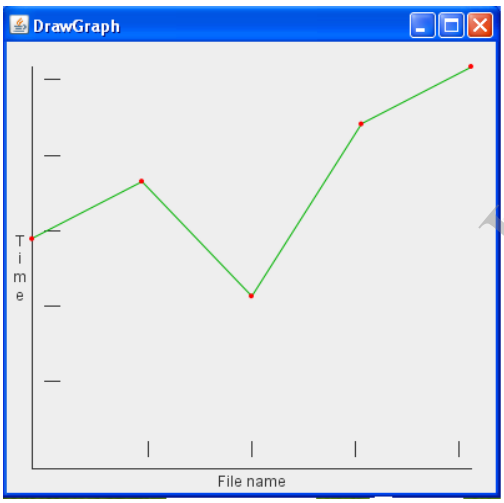

Fig 3: Graph of Filename vs file uploading time

VI. CONCLUSION AND FUTURE SCOPE

We have proposed a secrecy-archiving public ascertaining system for data storage security in Cloud Computing. By utilizing the homomorphic linear authenticator and random masking, we guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient ascertaining process, which not only eliminates the burden of cloud user from the tedious and possibly expensive ascertaining task, but also alleviates the users' fear of their outsourced data leakage. Also TPA concurrently handles multiple ascert sessions from different users for their outsourced data files, we further extend our secrecy-archiving public ascertaining system into a multi-user setting, where the TPA can perform multiple ascertaining tasks in a batch manner for better efficiency. Extensive analysis shows that our scheme is provably immune and highly efficient.

As future work we can construct a secrecy-archiving public ascertaining system, where multiple dynamic operations can be performed on same file and retrieval of all modifications done on that file. Also can extend the system to have an handshaking property between TPA and cloud server before ascertaining starts.

REFERENCES

[1]. P. Mell and T. Grance, "Draft NIST working definition of cloud computing," Referenced on June. 3rd, 2009 Online at http://csrc.nist.gov/groups/SNS/cloud-computing/index. html, 2009.

[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, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep.

[3]. 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, pp. 598–609.

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

[5] Amazon.com, "Amazon s3 availability event: July 20, 2008," Online at http://status.aws.amazon.com/s3-20080720.html, 2008.

[6] S. Wilson, "Appengine outage," Online at http://www. cio-weblog.com/50226711/appengine outage.php, June 2008.

[7] B. Krebs, "Payment Processor Breach May Be Largest Ever," Online at http://voices.washingtonpost.com/securityfix/2009/01/payment processor breach may b.html, Jan. 2009.

[8] 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, pp. 598–609.

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

[10] 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.

[11] 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.

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

[13] 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, pp.1–6