Journal of Computing and Electronic Information Management

ISSN: 2413-1660

A Novel Scheme of Data Update and Integrity Verification in Cloud Storage

Xinrui Zhang[1, a], Ping Zhu[1, b], Hua Zhang[2, c]

[1]School of Science Beijing University of Posts and Telecommunications, Beijing, China

[2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

[a]zxr_0222@163.com, [b]pzhubupt@gmail.com, [c]zhanghua_288@bupt.edu.cn

Abstract: Using cloud storage, data owners can remotely store their data and enjoy the on-demand high quality cloud services without the burden of local data storage and maintenance. However, due to data outsourcing and untrusted cloud servers, the data integrity becomes a challenging issue in cloud storage systems. A major concern is how to ensure the integrity of the outsourcing data. Recently, lots of dynamic auditing protocol for cloud storage was proposed while the efficiency of verification is undesirable. In this paper, we propose an effective and secure data integrity verification scheme for multi-cloud storage. This scheme also can achieve updating data dynamically and preventing cheating. Through experiment and analysis of security, we show that our proposed verification protocol are secure and efficient, especially it reduces the computation of verification.

Keywords: integrity verification, cloud storage, updating, security analysis

1. Introduction

Cloud storage is an important service for cloud computing. It allows data owners to remotely store their data and access them via networks at any time and from anywhere. Despite a large number of benefits such as improved scalability and accessibility, data replication and backup of cloud storage, it also brings some new security problems. Owing to the data are outsourced, the data owners relinquish the control over the fate of their data. Several recent surveys [1] show that cloud servers may hide data loss accidents to maintain the reputation, or discard the data which have not been or are rarely accessed to save storage space [2] . Therefore, checking the integrity and availability of the cloud data is essential for data owners and users.

Several efficient data access control protocols [3] [4] have been proposed to ensure the integrity of the static data. However, static storages are far from sufficient for cloud applications. The data stored in the cloud may be frequently updated by data owners. To prevent the untrusted servers from accessing sensitive data, traditional methods[5] usually encrypt the data and the data access control becomes the matter of key distribution. Recently, Qian Wang et al,[6]proposed a scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append. These schemes and our scheme are all for integrity verification and checking in multi-cloud service, while we apply the secret-sharing scheme to the cloud service model to create a new scheme. When making verification for the data, if the outsourcing data is correct, it can be checked by our scheme at once, verification time can be saved at great degree.

Secret-sharing schemes have been widely used since it was proposed. Actually. Aiming to avoid fraud problem, many authors have made in-depth study [7], C. Padro et al [8] proposed a security scheme to check the cheaters in space secret-sharing. Based on Padro's scheme, our paper increases the verification of the data so that it is suitable for the cloud storage service and convenient in verification. To implement the scheme, our paper uses polynomial and equations to build a mathematical model, and applies it into cloud storage service.

In our scheme, an efficient access control protocol is proposed to support data dynamic updating and integrity verification. An active adversary is able to arbitrarily modify the data in the cloud. As a result, this adversary can fool the real users and the data owner to believe that the data are well maintained in the cloud. Then we will give a new scheme to solve the problem in an effective way.

Our proposed scheme has two main contributions:

1. Low cost of Verification: When the user achieves the secret file, he can verify whether the sub-secret saved in the cloud is tampered. In our scheme, user needn't verify the correctness of the sub-secrets provided by each cloud server one by one. So the cost of the verification time is saved in a large degree.

2. Public verifiability: Our paper provides a major variation to achieve public verification. When the data owner renews some sub-secret, he will make necessary parameters public. At the same time, our paper provides a secret detecting factor which is produced in the polynomial. When user achieves the secret file, he will not get any effective information about the scheme.

2. Scheme Models

In our paper, we consider an auditing system for cloud storage as shown in Fig.1, which involves data owner, user, and the cloud server. In our scheme, data owner and user both could verify the correctness of the data which is stored in the cloud.

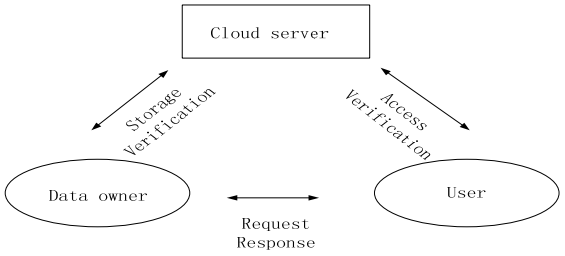

Figure 1. System model of the data storage verification

All the related network entities as follows:

- *Data owner (DO):* He uploads his own data which contains commercial value to the cloud secretly. He can be a large cloud service provider, an enterprise, an ordinary mobile phone user or a Pad user.
- *Cloud server (CS):* It is independent, who mainly provide the data storage service with the huge storage space and the powerful computational ability. At the same time, every cloud platform is transparent and open to public.
- *User:* the user gets the access to the data from DO by submitting the application, they can also validate the integrity of the data.

Before describing our protocol, we first define some notations as listed in Table 1.

TABLE I. NOTATIONS

Symbol	Physical Meaning
ID_U	user's ID
$\Delta f(z_l)$	the change of $f(z_l)$
F	data component
n	number of cloud servers
l	random number $(l < n)$
$\delta_l^{(i)}$	the change upon original sub-data block

The basic service model 1: DO first uploads his data to the data server secretly. The data server manipulates the data, generates some sub-secrets and distributes them to some different cloud servers. If some of the sub-secrets are tampered by the malicious

cloud server or external adversary (in our paper, we assume there is only one cheater in our model), *DO* can check out the checking servers and claim for compensation. When *DO* wants to modify his data on the cloud, he could do that at any time. The process is showed in Fig.2.

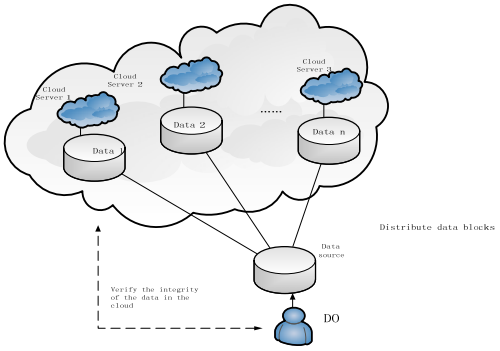

Figure 2. Service model without a user.

The basic service model 2: Based on the model 1, a user who can get the data with some authority is added into the model. If the user wants to apply the access to the data, he must get authority and the necessary verification factor. The process is showed in Fig.3.

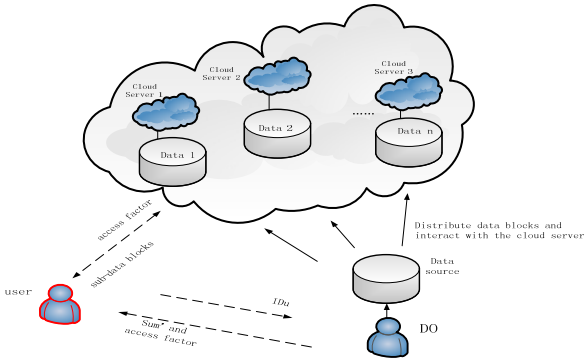

Figure 3. Service model with a user

3. Scheme Description

In our cloud storage scheme, we usually assume $S = \{server_1, server_2, \ldots, server_n\}$ as the n cloud servers which are used to storage the partitions. p, q are prime, $p = mq + 1$, m is as small as possible. $g \in Z_p^*$. DO computes the $sum = \sum v_i$, where $v_1, v_2, \ldots, v_{n-2} \in GF(q)$ are chosen randomly.

3.1 Specific Process in Model 1

1. Secret Distribution Phase

DO constructs the polynomial $f(z) = F + v_1 z + v_2 z^2 + \cdots + v_{2l-2} \cdot z^{2l-2} + sum \cdot z^{2l-1}$, where the secret file $F \in GF(q)$. Then he makes the $g^{v_1}, g^{v_2}, \ldots, g^{v_{2l-2}}$ to be the public parameters and $y_i = f(z_i) mod\ q, y_{n+i} = f(z_{n+i}) mod\ q\ (i = 1,2,\ldots,n)$. (z_i, z_{n+i}), (y_i, y_{n+i}) are the sub-secrets which are distributed to each $server_i\ (i = 1,2,\ldots,n)$.

2. Secret Renew Phase In Model 1

(I) preparation phase

At the moment $t = 1,2,3\ldots$, *DO* does some work as follows:

1) He chooses 2l-2 figures $\delta_1^{(t)}, \delta_2^{(t)}, \ldots, \delta_{2l-2}^{(t)}$ from Z_q^* and he defines $\delta_0^{(t)} = \Delta f$. He computes $u_i^{(t)} = \sum_{j=1}^{2l-2} \delta_j^{(t)} z_i^j + \Delta f_i$, $u_{n+i}^{(t)} = \sum_{j=1}^{2l-2} \delta_i^{(t)} z_{n+i}^j + \Delta f_{n+i}$, $\Delta f_i, (i = 1,2,3,\ldots,2l)$ is the updating secret.

At the same time, he computes $\varepsilon_j^{(t)} = g^{\delta_j^{(t)}} mod\ q, (j = 0,1,2,\ldots,2l-2)$, $e_i^{(t)} = ENC\left(u_i^{(t)}\right)$, $s_i^{(t)} = g^{x_i^{(t)}}$, where ENC is the public key encryption information from each cloud $server_i$.

2) publish the information

$MES(t) = (t, \varepsilon_0^{(t)}, \varepsilon_1^{(t)}, \ldots, \varepsilon_{2l-2}^{(t)}, e_1^{(t)}, e_2^{(t)}, \ldots, e_n^{(t)}, s_1^{(t)}, \ldots, s_n^{(t)})$

(II) update the sub-secret phase

1) The cloud servers decrypt $e^{(t)} = ENC\left(u_i^{(t)}\right)$ respectively, verify the updating sub-secret $u_i^{(t)}$. Then verifier will judge follow equation $g^{u_i^{(t)}} = \left(\varepsilon_1^{(t)}\right)\left(\varepsilon_2^{(t)}\right)^{z_i} \ldots \left(\varepsilon_{2l-2}^{(t)}\right)^{z_i^{2l-2}} mod\ q$

2) Updating sub-secret block $y_i^{(t)} \leftarrow y_i^{(t-1)} + u_i^{(t)}$, $y_{n+i}^{(t)} \leftarrow y_{n+i}^{(t-1)} + u_{n+i}^{(t)}$. Finally the cloud servers destroy $y_i^{(t-1)}, y_{n+i}^{(t-1)}$.

(III) authentication of the sub-secret phase

If L cloud servers reconstruct the secret *F*, each server provides $(y_1, y_{n+1}), \ldots, (y_l, y_{n+l})$ respectively. Then verifier (*DO*) composes equations (1) as follows.

$$\begin{aligned} y_1 &= F + v_1 z_1 + v_2 z_1^2 + \cdots + v_{2l-2} z_1^{2l-2} + v_{2l-1} z_1^{2l-1} \\ &\vdots \\ y_l &= F + v_1 z_l + v_2 z_l^2 + \cdots + v_{2l-2} z_l^{2l-2} + v_{2l-1} z_l^{2l-1} \\ y_{n+1} &= F + v_1 z_{n+1} + v_2 z_{n+1}^2 + \cdots + v_{2l-2} z_{n+1}^{2l-2} + v_{2l-1} z_{n+1}^{2l-1} \\ &\vdots \\ y_{n+l} &= F + v_1 z_{n+l} + v_2 z_{n+l}^2 + \cdots + v_{2l-2} z_{n+l}^{2l-2} + v_{2l-1} z_{n+l}^{2l-1} \end{aligned} \quad (1)$$

DO compares v_{2l-1} with the sum $\sum v_i$. If they are unequal, it illustrates there exists a cheater. We assume that the cheater provides Δy_1 instead of y_1. Δy_1 could be diffused into every coefficient and only when it is diffused into *F*, *DO* can't check out it.

3.2 Scheme in Model 2

The scheme in model 1 is effective for *DO* to check out the cheater, but it'll not be reasonable if there exists a user who wants to get the data. Because the user will recover the data and get the parameters $(v_1, v_2, \ldots, v_{2l-2})$ at the same time. If the user conspires with the servers, the secret file *F* and encryption scheme are exposed. Hence we improve our scheme to solve out the secret file *F* and v_{2l-1} simply by adding Δsum to sum , in model 2. By this way, user could check the data which he gets from cloud servers is right or not..

4. Scheme Evaluation

The improved scheme described above enjoys the desirable feature of privacy and supports dynamic auditing and updating. Regarding the security of scheme, we carry out some analysis. At the time t, *DO* updates the data block regularly and destroys the sub-block before. The information which was grabbed in the previous cycle by the external adversary or malicious cloud servers has been useless.

4.1 Correctness analysis

Recovery of the secret file F

Each $server_i$ sends its sub-secret $(y_i^{(t)}, y_{n+i}^{(t)})$ to the authorized user. Since the user also knows the public factor $(z_1, z_2, \ldots, z_l, z_{n+1}, z_{n+2}, \ldots, z_{n+l})$, he could construct the equations as equation (1)

The equations can be simplified as equation $Y = AX$, while X and Y are known, user can get $A = YX^{-1}$, then user gets the file *F* from the solution A.

4.2 Security analysis

Lemma 1: As assume there is a cloud server which provided a false sub-data block, we can't find the cheater, if and only if *F* is changed, the probability is $Pr1 = \frac{1}{C_{2l}^1 + C_{2l}^2 + \cdots + C_{2l}^{2l}} \cdot \frac{1}{2l}$.

Proof:

User constructs an equations with the sub-secret $(y_i, y_{n+i})(i = 1,2, \ldots, l)$ and the public parameters (z_i, z_{n+i}) $(i = 1,2, \ldots, l)$. He could get the data information by solving equations (1). If only sub-data block y_1 is false (in our paper we prove our solution with y_1 is false). y_1 could be diffused into every coefficient during we solve equations (1). There

are $C_{2l}^1 + C_{2l}^2 + \cdots + C_{2l}^{2l}$ kinds of cases. The probability that each coefficient was changed is equal and the value is $\frac{1}{2l}$. So we can get $Pr1 = \frac{1}{C_{2l}^1 + C_{2l}^2 + \cdots + C_{2l}^{2l}} \cdot \frac{1}{2l}$.

Lemma 2: The probability that equations (1) has solutions is $Pr2 = \frac{A_q^{2l-1}}{q^{2l-1}} \cdot \frac{q-(2l-1)}{q}$

Proof:

The matrix mentioned above is $2l \cdot 2l$. The probability of that $y_1, \ldots, y_l, y_{l+1}, \cdots, y_{n+l}$ are different is $\frac{A_q^{2l-1}}{q^{2l-1}}$, and the probability of that y_1 differs from the others is $\frac{q-(2l-1)}{q}$. So we can get the probability of the equations has solutions is $\frac{A_q^{2l-1}}{q^{2l-1}} \cdot \frac{q-(2l-1)}{q}$

Theorem: Assume one or several servers cheat successfully but the cheating can't be realized, the probability is $\frac{A_q^{2l-1}}{q^{2l-1}} \cdot \frac{q-(2l-1)}{q} \cdot \left(1 - \frac{1}{q^{2l-1}}\right)$

Proof:

We compute the propabilities of all the conditions. The cheating cloud server must be checked; other conditions are computed as follows. The sum of all the probabilities is $P_1 + P_2 \ldots + P_{2l-2} = \frac{(q-1)^{2l-2}}{q^{2l-2}} = 1 - \frac{1}{q^{2l-1}}$, so we get $Pr3 = \frac{A_q^{2l-1}}{q^{2l-1}} \cdot \frac{q-(2l-1)}{q} \cdot \left(1 - \frac{1}{q^{2l-1}}\right)$. The probability of checking servers guess the polynomial coefficients expect the situation above is $Pr4 = \frac{1}{q^{2l-1}}$.

By the above analysis, we can see that the probability of checking the integrity of data is very ideal. No matter *DO* or the user is very easy to find whether the data he get is right or not. At the same time, in our scheme, when the user recovers the data block, he couldn't get the secret coefficients, avoiding conspiracy between the data user and cloud servers.

4.3 Efficiency analysis

In this section, we show the data storage efficiency of the scheme. The experiment is conducted by Matlab to simulate data segmentation storing in multi-cloud servers. We get the values of sub-secretsby by solving polynomia and distribute them from the data owner to different cloud servers.

1. To analysis the effect of the size of data block, we perform experimentation of data storing. From Fig.4 we can see the time cost tends to stability increase along with the increased number of shares, while the time cost of no partition is the biggest. That means the more numbers of cloud servers to storage sub-secret will not bring a larger time cost and it will improve the security.

2. Then we will show the relationship between the data size and cloud storage in the situation of different number of cloud servers in Fig.5 When we store the data F in the server, the number of bytes will increase with the generated partitions. And we get $\frac{1}{2n}$ as the transmission efficiency ratio.

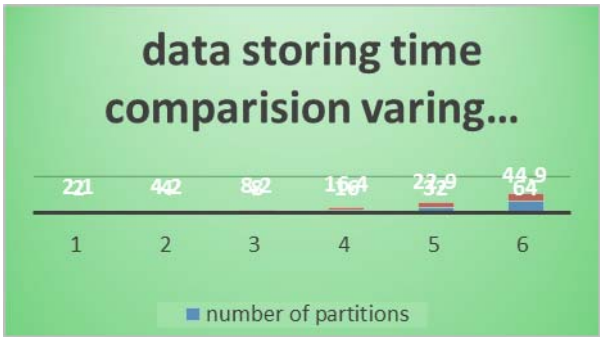

Figure 4. Relationship between storing time and shares

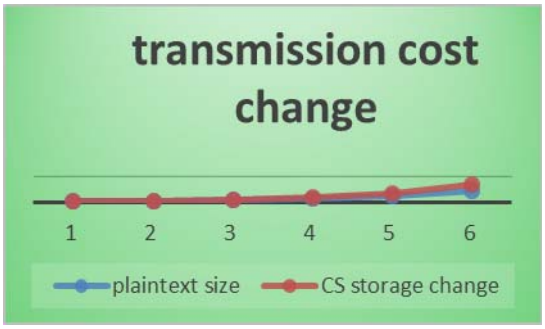

Figure 5. Transmission cost change.

3. At the same time, we will show the relationship between requested size of user and the actual downloads in Fig.6. which is similar with the Fig.5. If the number of authorized subset servers is l, we get $\frac{1}{2l}$ as the communication ratio.

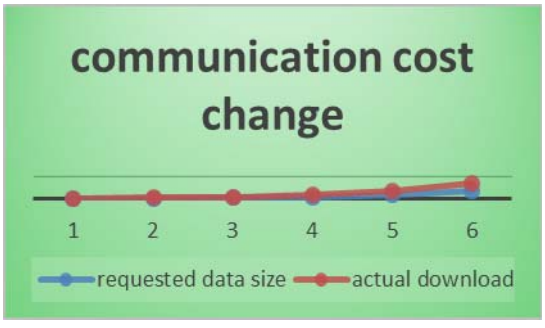

Figure 6. Communication cost change

5. Conclusion

In this paper, we investigate the problem of data integrity checking in cloud data storage. We propose a secure and transparent cloud data storage scheme, which provides secure data access control and reduces the amount of computation. Our scheme also supports dynamic outsourcing of data and integrity verification. It saves lots of time to check out whether there are cheaters. Though our scheme makes contributions on time saving of checking the cheaters, it still has some shortcomings in storage cost, which will be considered in the further paper.

Acknowledgments

This work is supported by NSFC (Grant Nos. 61300181, 61202434), the Fundamental Research Funds for the Central Universities (Grant No. 2015RC23).

References

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

[2] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, “Enabling public auditability and data dynamics for storage security in cloud computing”. Parallel and Distributed Systems, IEEE Transactions on, 2011, vol. 22, no. 5: pp. 847-859.

[3] Q. Wang, K. Ren, W. Lou, and Y. Zhang, “Dependable and secure sensor data storage with dynamic integrity assurance,” in Proc. Of IEEE INFOCOM’09, Rio de Janeiro, Brazil, April 2009.

[4] H. Shacham and B. Waters, “Compact proofs of retrievability”. Journal of Cryptology, 2013, vol. 26, no. 3: pp. 442-483

[5] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, “Patient controlled encryption: Ensuring privacy of electronic medical records”. In Proc.CCSW’09, 2009, pp. 103–114, ACM.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring data storage security in cloud computing,” in Proc. of IWQoS’09, July 2009, pp. 1–9.

[7] Cappentieri M., “A perfect threshold secret sharing scheme to identify cheaters, Designs, Codes and Cryptography”. 1995, vol.5: 183-187.

[8] Padro C., “Detection of threshold secret sharing scheme to identity cheaters, Designs, Codes and Cryptography”. 1999, 16:75-8.