Journal of Convergence Information Technology
Volume 5, Number 7, September 2010

A Study on Secure Data Storage Strategy in Cloud Computing

[1]Danwei Chen, [2]Yanjun He
[1, First Author]*College of Computer Technology, Nanjing University of Posts and Telecommunications, chendw@njupt.edu.cn*
[*2,Corresponding Author] *College of Computer Technology, Nanjing University of Posts and Telecommunications,realmeh@gmail.com*
doi: 10.4156/jcit.vol5.issue7.23

Abstract

Based on fundamental theories of k equations in algebra, n congruence surplus principle in elementary number theory, and the Abhishek's online data storage algorithm, we propose a secure data storage strategy in cloud computing. The strategy splits data d into k sections using the data splitting algorithm, ensures high data security by simplifying k equation solutions, and at the same time, guarantees highly reliable data using the coefficients generated by the splitting algorithm.

Keywords: *Cloud computing, Data partitioning, Distributed storage, Security strategy*

1. Introduction

Cloud computing mainly provides three kinds of services: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) [1]. The major difference between service based on cloud computing and traditional service is that user data is stored not in the local server, but in the distributed storage system of the service supplier. In many cases, however, users (especially business users) have high demands regarding data security and reliability.

Generally, in traditional data protection methods, plaintext data is stored after encryption. In practical applications, symmetric encryption algorithms, such as DES and AES, are usually adopted because of their efficiency. Although data stored in the cloud server are encrypted, encryption algorithm provides relatively lower security. Therefore, encrypted data are very likely to be vulnerable to attacks [2] and business interests become compromised once the server is invaded.

In this paper, we propose a secure data storage strategy capable of addressing the shortcomings of traditional data protection methods and improving security and reliability in cloud computing.

2. Data security storage strategies

Secure data storage in cloud computing is realized on the basis of a distributed system. After reaching the cloud, data can be randomly stored in any one or more servers. According to characteristics of the storage mode, each server in the distributed system can be abstracted as a storage node.

Suppose there are *m* servers in the system, written as: $S=\{s_1, s_2,.., s_m\}$.

Suppose the plaintext data set is *d*. The k equations based on the splitting algorithm is applied to data set *d* to generate *k(k<m)* data, written as:$\{d_1, d_2,.., d_k\} = Partition(d)$ in which *Partition()* is the data splitting algorithm illustrated in detail in Section 3 of this paper.

The generated data blocks are then split, and k servers are randomly chosen out of m servers, which can be expressed as the following formula:$\{d_1, d_2,.., d_k\}=map(S)$, where $S=\{s_1, s_2,.., s_m\}$.

The data restoration process can be expressed as $d = d_1 \cdot d_2 \cdot \cdots \cdot d_k \mod p$, where *p* is a large prime number.

The core of the secure storage strategy is its data splitting algorithm, which is an extension of fundamental theories of k equations in algebra, n congruence surplus principle in elementary number theory[3], key sharing of Shamir[4] and online data storage algorithm of Abhishek[5,6], through which data splitting storage is realized. The safety of the strategy mainly depends on two aspects. First, is the difficulty of decoding the data splitting algorithm. The second, is that because storage servers are randomly chosen after data splitting, encrypted data cannot be completely obtained by attacking one or more servers, making decoding even more difficult.

In addition, the strategy has inherent advantages in its fault tolerance compared with traditional data protection methods. In cloud computing, no assumptions on the robustness of any node in the distributed system can be made. Various unexpected factors can all result in temporary inaccessibility of some nodes or permanent inaccessibility of data. In such a case, traditional data protection means are often powerless. The secure strategy proposed in this paper ensures that data can be restored even when some nodes fail, which considerably improves system reliability.

3. Core algorithm of the secure strategy

3.1 Data splitting algorithm

In cryptography, it is much more convenience for constructing an isomorphic quotient ring as a complex field than algebraic operation when with the same structure [7]. We construct an isomorphic quotient ring with the same structure as complex field Z_p (where *p* is a large prime number), and a k congruence equation expressed as:

$x^k + \sum_{i=1}^{k-1} a_{k-i} x^{k-i} + d \equiv 0 \bmod p$ (1) where d Z_p is the data to be split, *0≤*a_i *≤p-1,* and *0≤d≤p-1* (Note: d here can be -*d*). According to the fundamental principle of k equations in algebra, Equation (1) has *k* roots. These roots are expressed as: $x^k + \sum_{i=1}^{k-1} a_{k-i} x^{k-i} + d \equiv 0 \bmod p$ and$\{r_1, r_2, ..., r_k\} \subseteq C$ (C is a set of complex numbers),The Equation (1) can be rewritten as:

$\prod_{i=1}^{k}(x - r_i) \equiv 0 \bmod p$ (2) where *1≤*r_i *≤p-1*. These r_i are data blocks generated after the splitting. Equations (1) and (2) show that d is independent of variable x. Therefore, the following can be generated:

$\prod_{i=1}^{k} r_i \equiv d \bmod p$ (3)

3.2 High efficiency of the algorithm

High efficiency of the algorithm is illustrated through two aspects: data splitting and storage process and data restoration process. In the data splitting and storage process, splitting algorithm applied to data set *d* generates k blocks of data r_1, r_2, ..., r_k. Then, these data blocks are stored in a randomly chosen server. In addition, coefficient a_i is stored as backup information. The process mainly includes the following operations:

1. *k-1* numbers $r_1, r_2, ..., r_{k-1}$ are randomly chosen within the finite field Z_p.
2. $r_k = d \cdot (r_1 \cdot r_2 \cdot\cdots\cdot r_{k-1})^{-1} \bmod p$ is calculated.
3. Coefficients $a_1, ..., a_{k-1}$ is calculated by constructing polynomial *p(k)*, in which *p(k)* is shown in the following:

$$p(k) = (x - r_1)(x - r_2)...(x - r_k) \bmod p$$
$$= x^k + a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + ... + a_1 x + a_0 \bmod p$$

From the calculation process above, we can infer that *k* multiplications, one modular inversion, and the multiplication of p(k) polynomial to k times are needed for the algorithm to generate k blocks of data. Therefore, the time complexity is *O(k)*[8].

For data decoding and restoration, the user retrieves data of each block *R={*$r_1, r_2, ..., r_k$*}* from relevant servers according to the locally-stored data position index, and then obtains the plaintext data by calculating $d = r_1 \cdot r_2 \cdot\cdots\cdot r_k \bmod p$. Clearly, time complexity of the decoding process, the same as that of the encryption process, is *O(k)*.

Therefore, execution efficiency of the algorithm, whether in an encryption or decoding process, is rather high—much higher than in asymmetrical encryption algorithm.

3.3 Security of the algorithm

Theorem 1. If coefficient a_i *(1≤i≤k-1)* is randomly chosen and is zero when *k-1* coefficients are different, the probability of generating authentic data set *d* is less than *1/p*, even when the roots of the *k-1* equations are known.

Proof: Given data set *d*, the coefficient in Equation (1) is chosen with the following method to ensure that the equation has *k* roots: *k-1* roots $r_1, r_2 ,..., r_{k-1}$ are randomly chosen in finite field Z_p. The k-th root can be obtained with the following equation:

$$r_k = d \cdot \left(r_1 \cdot r_2 \cdot \cdots \cdot r_{k-1}\right)^{-1} \bmod p \quad (4)$$

As the *k* roots are randomly distributed in Z_p [9], the probability of obtaining r_k without knowing *d* is *1/p*. Conversely, the probability of obtaining *d* without knowing r_k is also *1/p*.

The following explanation is presented to discuss why the coefficients cannot be zero at the same time. Suppose $a_0 = a_1 = ... = a_{k-1}$ *=0*, Equation (1) is converted into $x^k + d \equiv 0 \bmod p$ (5).

Based on n congruence surplus principle in elementary number theory, we can infer that, for Equation (5) to have *k* roots, the necessary conditions are *GCD(p-1,k)=k; GCD(d,p)=1;* and $\exists b \quad Z_p$. Data set *d* is the k-th power of b. Usually, *p=(k×s+1)* is chosen, in which *s N*. In such a case, there are certain requirements on data set *d*, as well, which is unacceptable in practical application. Even if *d* satisfies relevant requirements, the attacker can easily calculate original data set *d*, by merely determining the data of one block generated by the splitting and the number of blocks. In this case, security is rather poor. Therefore, the algorithm requires that the chosen coefficients cannot be zero at the same time.

Theorem 2. If the attacker invades a storage node, steals data block r_i, and wants to restore data set *d* with aggressive methods based on r_i and decode coefficients of *k(k≥2)* polynomial in finite field Z_p, the required time complexity is $\Omega\left(\left\lceil \frac{p^{k-1}}{(k-1)!} \right\rceil\right)$.

Proof: Suppose the coefficient set of Equation (1) is *A={*$a_0, a_1 ,..., a_{k-1}$*}*, in which *0≤*a_i *≤p-1*. Each group of examples of a_i values are correspondent to the only series of solutions of its root set *R={*$r_1, r_2 ,..., r_k$*}*, and vice versa. To solve the coefficients of *k(k≥2)* polynomial, Equation (3) can be used. Suppose the attacker obtains r_i; he needs to randomly choose *k-1* numbers (Note: *k-1* numbers here can be repeated) from the set *S={0, 1, 2, ... ,p-1}*. Clearly, times of calculations that the attacker needs can be expressed as the following formula:

$$\left\langle {p \atop k-1} \right\rangle = \binom{p-1+k-1}{k-1}$$

$$= \frac{(p+k-2)!}{(p-1)!(k-1)!}$$

$$= \frac{(p+k-2)(p+k-3)...(p+k-k)(p-1)!}{(p-1)!(k-1)!} = \frac{(p+k-2)(p+k-3)...(p+k-k)}{(k-1)!}$$

$$\geq \left\lceil \frac{p^{k-1}}{(k-1)!} \right\rceil$$

In practical application, p>>k>>2. Such a calculation amount is far larger than the processing ability of mainstream computers and ensures that it cannot be decoded in current computing environments.

4. Reliability of the secure data storage strategy

One of the advantages distinguishing the proposed secure strategy from traditional data protection is that it provides highly reliable data protection. When splitting plaintext data into data blocks, we obtain *k* data blocks $r_1, r_2 ,..., r_k$ and *k-1* coefficients $a_1 ,..., a_{k-1}$ of *k* equations. These coefficients are stored in the server as backup information [10]. In an actual environment, one or more of the *k* nodes storing data blocks cannot be accessed because of the problems of the network or the server itself. Now data set *d* cannot be restored with Equation (5). In such case, visiting only one of the *k* nodes is needed. Suppose the data were retrieved from node r_i .coefficients a_1 ,..., a_{k-1} are then retrieved from the server of the backup coefficient. r_i and the coefficients are substituted into Equation (3), and plaintext data set *d* can be obtained. Therefore, the strategy provides a solid method for protecting the data stored in the distributed system.

5. A comparison between the secure storage strategy and traditional data protection methods

5.1 Security of the analysis strategy of the experiment

The security of data splitting algorithm is related to key length. Furthermore, it also increases exponentially with the increase in the number of data blocks. However, traditional data protection methods usually adopt symmetrical encryption, such as DES[11], the security of which merely depends on key length. For a computer capable of processing one million instructions within a second, with the same key length, the decoding time of the splitting encryption method significantly increases with the increase in the number of data blocks, whereas the decoding time of the symmetrical encryption algorithm remains almost the same (Figure 1). In practical analysis, the key length of the algorithm is usually determined as 129 bits. The number of data blocks is 16. Its security is 8 times higher than that of traditional methods, and its reliability is 50 times higher.

5.2 Reliability of the analysis strategy of the experiment

The reliability of the secure data storage strategy depends on the backup data coefficients. When one or more nodes cannot be accessed, the secure strategy can ensure that the data will be restored as long as one of the k nodes can be accessed. However, traditional data storage methods require all the data in the k nodes to be retrieved. Thus, the more blocks the data are split into, the poorer the reliability of traditional data storage. Figure 2 shows that the ratio of the reliability of the splitting storage strategy to that of traditional data protection methods increases exponentially, with the increase in the number of data splitting blocks. Therefore, the secure storage strategy has tremendous advantages in terms of reliability.

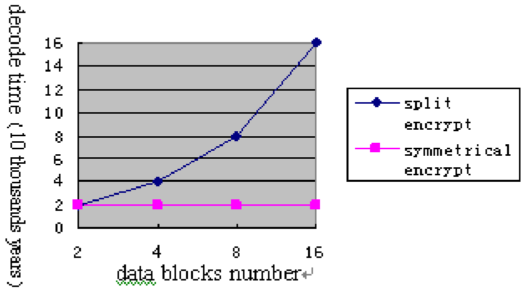

Figure 1. Comparison between decoding time of splitting encryption and that of encryption decoding.

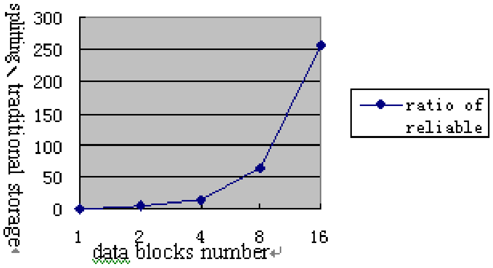

Figure 2. Analysis of the reliability of splitting storage.

6. Conclusion

Based on relevant principles in algebra and elementary number theory, key sharing of Shamir and online data storage algorithm of Abhishek, this paper proposes a secure data storage strategy applicable to the distributed system in cloud computing, which successfully solves various data security problems encountered by service modes based on cloud computing. In terms of data security, the strategy enhances the decoding difficulty tenfold with the increased number of data blocks. In addition, its fault tolerance is higher than that of the single-node storage method by hundreds of times. The secure strategy, however, also has its shortcomings, such as much data redundancy. These shortcomings can be taken as improvement directions of subsequent research.

7. Reference

[1] J.Rittinghouse,J.Ransome, Cloud Computing: Implementation, Management, and Security, 2009 .

[2] Prasanta GogoiB, Borah,D K Bhattacharyya, Anomaly Detection Analysis of Intrusion Data using Supervised & Unsupervised Approach, Journal of AICIT, AICIT, vol.5, no.1, pp.95-111, 2010.

[3] K.q. FENG Number Theory and Cryptography, Science Press, China, 2007.

[4] A.Shamir How to Share a Secret[J]. Communications of the ACM, vol.22,no.11,pp.612-613, 1979.

[5] M.H. Dehkordi, S. Mashhadi, New efficient and practical verifiable multi-secret sharing schemes, Information Sciences, vol.9 , no.2262–2274, 2008.

[6] A.Parakh , S. Kak. Online data storage using implicit security, Information Sciences, vol.179, no. 3323–3331, 2009.

[7] T. Moon, Error Correction Coding: Mathematical Methods and Algorithms, Wiley, USA, 2005.

[8] A. Aho,J. Hopcroft, J. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, USA ,1974.

[9] S. Kak, A cubic public-key transformation,Circuits, Systems and Signal Processing, vol.26, pp. 353–359, 2007.

[10] Anestis A. Toptsis, K-grid: A Structure for Storage and Retrieval of Affective Knowledge, Journal of AICIT, AICIT, vol. 4, no. 2, pp.16-30, 2009.

[11] Bruce Schneier, Applied Cryptography, John Wiley & Sons, USA, 1996.