Research on the Security of Massive Data Storage

Zhai Guanghui
Xuchang university School of Computer
Xuchang, Chnia 461000
xczgh@163.com

Li Juan
Xuchang university School of Computer
Xuchang, Chnia 461000
zimulj@163.com

***Abstract*—Storage of massive data is receiving more and more attention in recent years and it has been widely used in many fields, meanwhile, its security is facing a great challenge, too. In this paper, we propose a distributed authentication protocol to ensure the security of massive data storage, this method utilizes Reed-Solomon codes to ensure the availability and reliability of data. In addition, it makes use of the Sobol sequences token to pre-calculate and verify the integrity of data. The proposed method can not only verify the correctness of the storage, but also recognize the server which executes wrong operation.**

Index Terms-Massive data, security of storage, Reed-Solomon codes, Sobol sequences

I. HIDDEN SAFETY CONCERNS OF MASSIVE DATA STORAGE

More and more fields require operations on massive data, generally the volume of data is very large, and it contains various kinds of spatial data, statistical data, texts, sounds, images, hypertexts etc.. Security of massive data storage is facing a great challenge, just consider what is your response when the password of your bank account is stolen. Thus in this paper we propose a distributed authentication protocol to ensure the security of massive data storage, this method utilizes Reed-Solomon codes to ensure the availability and reliability of data. In addition, it makes use of the Sobol sequences token to pre-calculate and verify the integrity of data. The proposed method can not only verify the correctness of the storage, but also recognize the server which executes wrong operation.

II. FRAMEWORK AND RELATED ALGORITHMS OF THE SYSTEM

In coding theory, Reed-Solomon (RS) codes are non-binary cyclic error-correcting codes invented by Irving S. Reed and Gustave Solomon. They described a systematic way of building codes that could detect and correct multiple random symbol errors. By adding t check symbols to the data, a RS code can detect any combination of up to t erroneous symbols, and correct up to $\lfloor t/2 \rfloor$ symbols. Reed-Solomon codes have since found important applications from deep-space communication to consumer electronics. They are prominently used in consumer electronics such as CDs, DVDs, Blu-ray Discs, in data transmission technologies such as DSL and WiMAX, in broadcast systems such as DVB and ATSC, and in computer applications such as RAID 6 systems.

Sobol sequences are an example of quasi-random low-discrepancy sequences. They were first introduced by the Russian mathematician I. M. Sobol in 1967. These sequences use a base of two to form successively finer uniform partitions of the unit interval, and then reorder the coordinates in each dimension.

The framework of massive data storage can be divided into three parts, that is, users, service provider and a third-party verification, the framework is shown in Fig. 1.

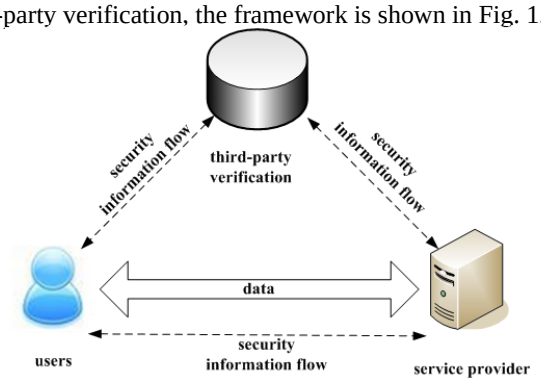

Fig. 1 Framework of massive data storage

Users store data in the servers which are provided by service provider, typically include desk-top computers, lap-top computers, mobile phones etc..

Service provider has a great deal of resources and professional support for data storage. Service provider provide service for data storage, thus users can access service via the Internet.

The third-party verification has professional knowledge and abilities, it can detect the risk of data storage.

In order to solve the security of massive data storage, we propose a distributed authentication protocol which uses Sobel sequences. The method allows users to verify the correctness of data which are stored by service provider, and recognize servers which execute wrong operation without local copy of original data. When users do not have time to verify the correctness of data, they can submit the task to the third-party verification.

III. FRAMEWORK OF SAFE STORAGE

So as to ensure the security of massive data, in this paper we propose a distributed verification model, which contains three stages: document distribution, token pre-calculation and challenge response protocol. Following are some variables used in the paper.

F denotes data files to be stored, let F denotes m data matrices which have the same size, each group has five

Published by Atlantis Press, Paris, France.

blocks, these data blocks are part of $GF(2^p)$, and p is 8, 16 or 32.

A denotes the distributed matrix of Reed-Solomon codes.

G is code file matrix, it consists of k vectors (k=m+n) , each vector has L blocks.

We use $f_{key}(.)$ as Sobol random function, its definition is: $f:\{0,1\}^* * key - GF(2^p)$.

$\pi_{key}(.)$ denotes Sobol random combination, it is defined as $\prod:\{0,1\}^{\log 2(l)} * key - \{0,1\}^{\log 2(l)}$

We utilize *ver* to record how many times the matrix is modified, and the default value is 0.

s_{ij}^{ver} is the seed of SRF, it depends on filename, block index i, location of server j and version number of *ver*.

IV. IMPLEMENTATION OF THE MODEL

A. Document Distribution

Erasure codes are used to ensure the error tolerance and improve performance for distributed storage system. In massive data storage, the proposed method uses this technology to allocate the entire data file, and ensure the availability and reliability of data through k (k=m+n) servers. In *A(m+n,n)*, Reed-Solomon erasure codes can produce n parity-check blocks from m data blocks. Thus the original data can be reconstructed from any m+n blocks, so the original file can be reconstructed from m+n servers, and it do not result in loss of data, furthermore, the space consumed is quite small.

Let $l \le 2^p - 1$, $F = \{F_1, F_2, ...F_m\}$ and $F_i = (F_{1i}, F_{2i}, ...F_{li})^T, (i \in \{1...m\})$, the distribution of data is odd-even symmetrical, matrix A is:

$$\begin{pmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_n \\ x_1^2 & \cdots & x_n^2 \\ \vdots & \ddots & \vdots \\ x_1^{n-1} & \cdots & x_n^{n-1} \end{pmatrix}$$

where $x_j (j \in \{1....k\})$ are different elements randomly chosen from $GF(2^p)$.

After a series of row elementary transformations, the new matrix is:

$$A = (I / P) = \begin{pmatrix} 1 & 0 & \cdots & 0 & p11 & p12 & \cdots & p1n \\ 0 & 1 & \cdots & 0 & p21 & p22 & \cdots & p2n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & pm1 & pm2 & \cdots & pmn \end{pmatrix}$$

where I is an identity matrix of size m by m, P is a matrix of size m by m which is produced by parity-check, one should note that A is Vandermonde matrix, and it is invertible. In order to obtain the code file, we should use the following formula:

$$G = F \bullet A[13] = (G^{(1)}, G^{(2)}, \ldots G^{(m)}, G^{(m+1)}, \ldots G^{(k)}) = (F_1, F_2, \ldots F_m, F_{m+1}, \cdots F_k)$$

where $G^{(j)} = (g_1^{(j)}, g_2^{(j)}, \ldots g_k^{(j)})^T$ (j∈ 1,…,k) .

The code file contains original data vector F and parity-check G(m+1),…,G(k) produced by F.

B. Token Pre-calculation

Files after coding, in order to ensure correctness of massive data storage and recognize the server which executes wrong operation. The proposed method totally depends on Sobol sequences to pre-calculate token. The main idea is before document distribution, users pre-calculate a certain quantity of short verification token via personal data vector *G(j)*. Each token has a group of random data blocks. Then after the token is pre-calculated, users store all the tokens in local devices. When users want to verify the correctness of data, they can verify via the random number produced by data blocks. When service provider receives the request from users, each server needs to calculate a short signature through corresponding random model and return it to users. As for maintaining the integrity of data, the return value $R_i^{(j)}$ should match corresponding token $v_i^{(j)}$ which is pre-calculated. Meanwhile, all the other servers work on the same group of random data block.

C. Challenge Response Protocol

Once the data is stored, we challenge the corresponding protocol to verify the integrity of erasure codes and recognize the server which executes wrong operation. However, response of servers is not enough to detect the integrity of data, but it is necessary to recognize the server which executes wrong operation. The procedure of the proposed algorithm is as follows:

(1) Start.

(2) Users utilize Sobol sequences to produce $x = fkSRF(i)$ and $y = fkSRP(i)$.

(3) Users send {x,y} to each server.

(4) The server calculates $\{R_i^{(j)} = \sum \mathrm{rq} = q * G(j)[\pi_y(q)], 1 \le j \le \mathrm{k}\}$.

(5) The server returns $R_i^{(j)}$ to users.

(6) for $\mathrm{j} \leftarrow \mathrm{m+1,k}$ do

(7) $R_i^{(j)} \leftarrow R_i^{(j)} - \sum rq = 1$.

(8) end for

(9) if $(R_i^{(1)}, \ldots, R_i^{(m)}) \bullet p == (R_i^{(m+1)}, \ldots, R_i^{(k)})$

(10) proceed with the next challenge response protocol.

(11) else

(12) for $j \leftarrow 1,k$ do

(13) if $(R_i^{(j)} != v_i^{(j)})$ then

(14) return the server is in wrong operation state.
(15) end.

V. CONCLUSIONS

In this paper, we propose a more effective and flexible method for distributed verification to solve the security of massive data storage. We utilize Reed-Solomon erasure codes to distribute files and ensure the availability and reliability of data. What's more, we use Sobol sequences to verify the integrity of erasure codes. However, the design of model is relatively complex, so in future, we will investigate how to improve the efficiency of the model.

REFERENCES

[1] J. S. Plank and Y. Ding, “Note: Correction to the 1997 Tutorial on Reed-Solomon Coding,” University of Tennessee, Tech. Rep. CS-03-504, 2003.

[2] Brately P and Fox B L (1988) Algorithm 659: Implementing Sobol’s Quasi-random Sequence Generator ACM Trans. Math. Software 14 (1) 88–100.

[3] A. Khetrapal and V. Ganesh, “HBase and Hypertable for large scale distributed storage systems,” Dept. of Computer Science, Purdue University.

[4] H. Meer, N Arvanitis, and M. Slaviero. Clobbering the cloud. Black Hat USA 2009.

[5] Urs Hoelzle, Luiz Andre Barroso, “The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines,” Morgan and Claypool Publishers, 2009.

Published by Atlantis Press, Paris, France.