2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)
Dec 20-22, 2013, Shenyang, China

A Distribution Model for Data Leakage Prevention

Yin Fan
School of Computer, Wuhan University
College of Computer Science, South-Central University for Nationalities
Wuhan, China

Wang Lina
School of Computer, Wuhan University
Wuhan, China

Yu Rongwei
School of Computer, Wuhan University
Wuhan, China

Ma Xiaoyan
School of Computer, Wuhan University
Wuhan, China

***Abstract*—In this paper we present a file distribution model aiming at data leakage prevention. Taking into consideration the guilt probability which is the probability of the user having leaked files, this model can select a file allocation plan with the least overlap between obtained file sets of users, as the consequence the model can find out leakage sources with high probability. So that measures can be taken for information security further. The simulation experiments reveal that the model can distinguish malicious users and detect the leak source effectively.**

Keywords—Distribution Model; Cloud computing; Data Leakage Prevention; Information Security

I. INTRODUCTION

With the popularity of cloud computing, sensitive information leakage become more and more critical. Current study of cloud computing has made considerable progress, but development of cloud computing still faces serious security issues [1]. According to Gartner's survey in 2009 showed that more than 70% of companies would not adopt cloud computing in the near future for the reason of security and data privacy concerns. Meanwhile, cloud computing security accidents broke out in several global giants such as Google, Amazon and Microsoft. In 2009 Google's cloud computing platform fails, resulting in a large number of user data leakage. Therefore, in the course of accessing and distributing files that contain sensitive data in a company or a department, to ensure the privacy of sensitive files and to control properly user's usage right over them are now key scientific problems in the field of information security, drawing close attention from institutions at home and abroad.

There a lot of works studying on data leakage prevention. Digital watermarks can be used to verify the authenticity or integrity of the carrier signal or to show the identity of its owners. so digital watermarks may prevent illegal copying in and trace the leak source[2-4]. But a watermark will modify the item being watermarked. If the object to be watermarked cannot be modified then a watermark cannot be used. Data provenance[5] technology, that can determine the data the original owners mainly used in the database, may detect the agents leaking data. In Ref.[6], a trust management paradigm is presented to protect both intra- and inter-organizational information flows against the threat of information disclosure. Data allocation strategies featured with fake objects injection are proposed to improve the probability of identifying leakages[7], but they do not take users' guilt probability into account of distribution strategy. To address this weakness, we improve the distribution algorithm.

In this paper, we study the application where there is a distributor, as a trusted party, managing and distributing files that contain sensitive information to authorized users when they require. In Ref.[8],we have presented a file distribution model based on trustworthiness that aims to prevent data leakage. To assess the risk of distributing a file, two elements are taken into consideration comprehensively, the subjective one—guilt probability *G*, which is calculated based on the user's trustworthiness and his obtained file set overlapping leaked file set; and the objective one—platform vulnerability, which may cause data leakage unintentionally. On this basis, in this paper we present a more integrated file distribution model which selects the best allocation plan with consideration of the user's guilt probability *G* which describes the probability that the user has leaked files as a factor. Considering that the user's accessed files and file leakage, the

Foundation item: Supported by the National Nature Science Foundation of China(61373169, 61103219), the Doctoral Fund of the Ministry of Education priority areas of development projects(20110141130006), and the Fundamental Research Funds for the Central Universities(CTZ13023)

978-1-4799-2565-0/13/$31.00 ©2013 IEEE

allocation plan makes the minimum intersection between file sets got by the involved users. Therefore, comparing with the distribution model of Ref. [7,8] , file leakage source can be confirmed with higher probability in case of files leak.

II. Distribution Strategies

We have improved the distribution model of Ref. [7] in the following two aspects:(1) Realize multiple rounds distribution, that's to say, before each round of distribution the user's accessed files should be considered; (2) Consider the impact of the user's guilt probability[8] G which describe the probability of the users having leaked files. The introduction of guilt probability can contribute to distinguish the leakage source effectively.

Let T denote the whole set of files, D_i denote the set of files of a user u_i , G_i denote the guilt probability of u_i, $hist_i$ denote the number of obtained files of a user u_i, m_i denote the number of requiring files of u_i in this round distribution , and Vt_k denote the set of users who have received the file t_k and , and $a[k]$ denote number of users who have received the file t_k. Improved Distribution strategy is shown in Eq. 1 and Eq. 2.

$$Sval = \sum_{i=1}^{n} \frac{1}{|D_i| \times (1 - G_i)} \left(\sum_{\substack{j=1 \\ j \neq i}}^{n} |D_i \cap D_j| \right) = \sum_{k=1}^{|T|} c[k] \quad (1)$$

$$c[k] = (a[k] - 1) \sum_{U_i \in Vtk} \frac{1}{(hist_i + m_i) \times (1 - G_i)} \quad (2)$$

In our distribution strategy is to choose the allocation with the lowest *Sval*. Because the allocation plan is best if its *Sval* is lowest, that is to say, the minimum overlap of users' file set means the greatest difference among the users' obtained files. Thus, when tracing and auditing the leak source of a file, we can lock leak source as small as possible, so as to prevent the more information leakage.

III. Algorithm Description

A. Overall Algorithmic Process

We suppose that the file distribution system collects all users' requirements in a round cycle and decides how to allocate files at the end of this round. Each round of the distribution process is as follows.

1 Generate the initial allocation plan *AP* of this round. If this is the first round of distribution, Algorithm FirstDis is called to generate the initial allocation plan, Otherwise, Algorithm NextDis is called;

2 Call Algorithm DistOneDis to generate new allocation plans *AP'* based on *AP*;

3 Call Algorithm MinimumSum to calculate *Sval* of each allocation plan *AP'* according to Eq. 1, and choose the best allocation plan AP'_b among all the allocation plan, that is, the *Sval* of AP'_b is lowest;

4 If AP'_b is better than *AP*, that is, the *Sval* of AP'_b is lower than *AP*, AP'_b is replaced *AP*, then go to step 2; else *AP* is the best allocation plan of this round distribution.

B. Algorithm FirstDis and Algorithm NextDis

If this is the first round of distribution, Algorithm FirstDis is called to generate the initial allocation plan. The input of the algorithm is the requiring file number m_i of each user u_i . Let *M* denote the total number of files that all users requiring. The output is an array *a* as the allocation plan *AP*. The algorithm will distribute the file t_k to *a[k]* different users, And *a[k]* is

$$a[k] = \begin{cases} M/|T| + 1, & k \leq M\%|T| \\ M/|T|, & for\ the\ rest \end{cases} \quad (3)$$

If this is not the first round of distribution, Algorithm NextDis is called to generate the initial allocation plan. Its basic idea, as same as Algorithm FirstDis, is to distribute the file shared by least users ,but it should take the case of the received files of users into account.

C. Algorithm DistOneDis

This algorithm produces different new allocation plans by adjusting the values of *a[k]* in the initial plan caused by Algorithm FirstDis or Algorithm NextDis. The algorithm procedure is as follows.

1 All files in the initial allocation plan *AP* are sorted in descending order by each *a[k]*.

2 Generate new allocation plan *AP'*: if $k_i > k_j$, then adjust the number of users who will received the file t_{ki} and t_{kj} as followed.

$$a[k_i] = a[k_i]_{old} + 1 \,\&\&\, a[k_j] = a[k_j]_{old} - 1 \,; \quad (4)$$

Figure 1 shows an example how to produce new allocation plans based the initial plan. There are 6 files to be distributed. Assuming files' *a[k]* is {5,5,4,4,3,3} in the initial allocation plan, then six new plans, such as {6,5,4,4,3,2}, {5,5,5,4,3,2}, {5,5,4,4,4,2}, {6,5,4,3,3,3}, {5,5,5,3,3,3} and {6,4,4,3,3,3}, can be produced as shown in Figure 1.

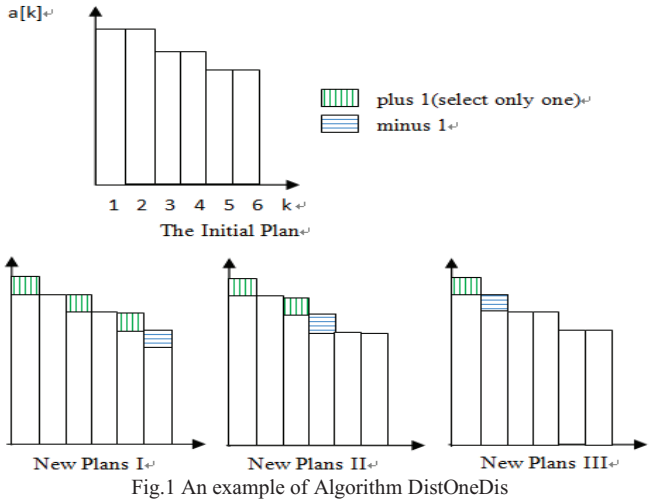

Fig.1 An example of Algorithm DistOneDis

D. Algorithm MinimumSum

This algorithm is calculated the minimum value of each allocation plan according to Eq. 1. Algorithm input includes ①Vector m, $m[i]$ denotes the number of requiring files of u_i in this round distribution;② An allocation plan AP.

Algorithm steps are as follows:

1 For each file t_k in T

1.1 Calculate the user set N which must obtain t_k. Distribute t_k to each user of set N.

1.2 Calculate the users set E which may obtain t_k. If $|E|<a[k]$, the allocation plan is infeasible, failed to exit.

1.3 Sort the elements of the E-N in ascending order by $(m_i+hist_i)*(1-G_i)$, and take the top $a[k]-|N|$ elements form a set O. Distribute t_k to each user of set O.

1.4 Calculate $c[k]$ of each file t_k according to Eq. 2.

2 Calculate *Sval* according to Eq. 1

IV. Experiment Results

In order to analyze and validate that the user's guilt probability G introduced to distribution strategy can help mark leakage source, we conduct simulation experiments. In the experiments, users are divided into three types: type Ⅰ are honest and would never leak files; type Ⅱ are malicious and would leak files quite often; type Ⅲ are common users and might leak a few files they got occasionally. We assume there are 100 users, 20% are type Ⅰ, 20% are type Ⅱ and 60% are type Ⅲ. Besides, there are 600 files. Each user can apply for an average of 30 files and a file cannot be distributed to the same user multiple times.

To verify effects of the guilt probability on improving the probability of marking leakage source, in this paper we select two simulation distributing files experiments, one taking the guilt probability into account and the other not, then we analyze and compare two experiment results. These two experiments are set the same scene, that means in the course of distributing files all the users act same - the specific user requires, receives or leaks a specific file at the specific moment.

We present the average guilt probabilities of the three types users whether taking account of the guilt probabilities in distribution strategy Figure 2 and Figure 3 respectively. We suppose that the distribution lasts 40 rounds and a round is made up of 75 time steps, within every one of which there is a user applying for a file. The file distribution system collects all users' requirements of this round and decides whether allocate files at the end of the round.

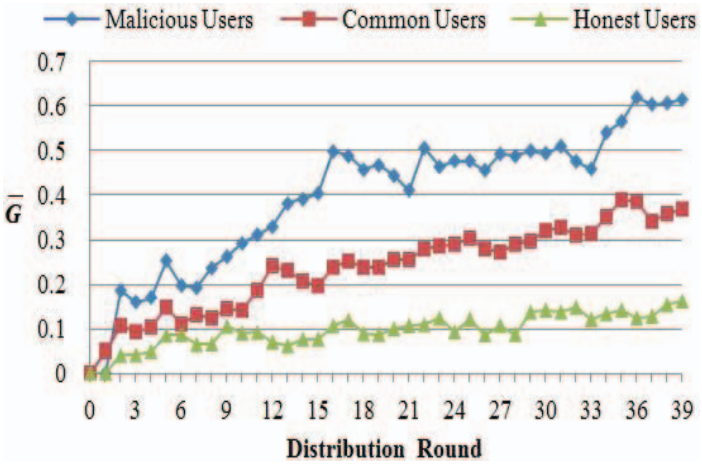

Fig.2 $\overline{G}$ in the distribution strategy without guilt probability

We can see from both Figure 2 and Figure 3 that the average guilt probability for the malicious users is almost highest, the average guilt probability for the honest users is lowest and the average guilt probability for the common users is between the other types. In Figure 3 the average guilt probability for the malicious users is peaked at about 0.8 and the difference of the average guilt probability between malicious and honest users is up to 0.7. By contrast in Figure 2 the average guilt probability for the malicious users is peaked at about 0.7 and the difference of the average guilt probability between malicious and honest users is only up to 0.4. We can draw a conclusion that we can more easily recognize malicious users if taking the guilt probability into account of distribution strategy.

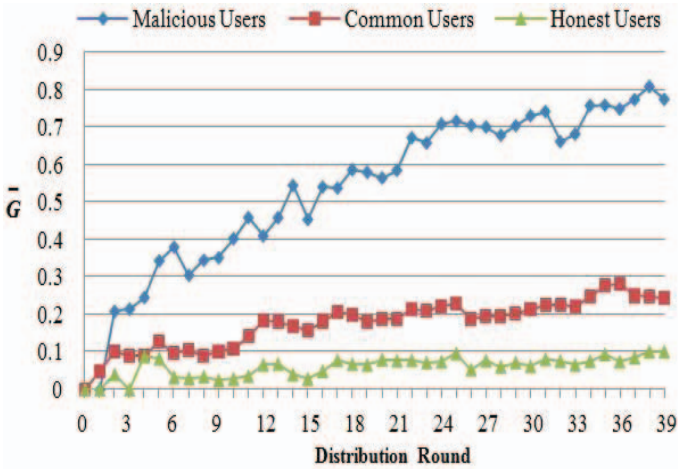

Fig.3 $\overline{G}$ in the distribution strategy with guilt probability

Because it is difficult to directly detect the leakage behavior of users in practical applications, the guilt probability can be used to infer the leaker when the file is found leakage. If the user's guilt probability is very high, then we can conclude that the user may be a leaker. Based on the above analysis, the distribution strategy taking account of the guilt probability can increase the gap between users of different types, helping more accurately determine a malicious user and then take measures for data leakage prevention.

V. CONCLUSION

As sensitive information leakage become more and more critical in the cloud computing, this paper provide a novel distribution model aiming at data leakage prevention. With the account of the users' guilt probability, the model can select an allocation plan with the least overlap between obtained file set of involved users. Hence the consequence the model can find out leakage sources with high probability, and measures could be taken to prevent further leakage. The simulation experiments indicate that the model can distinguish malicious users, that is, the guilt probability of malicious is highest with contrast of honest or common users. Thereby the distribution model can effectively detect the leak source so as to protect information security.

REFERENCES

[1] Feng Dengguo, Zhang Min, Zhang Yan. Study on cloud computing security [J]. Journal of Software, 2011, 22(1): 71-83.

[2] J.J.K.O.Ruanaidh, W.J.Dowling, F.M.Boland, Watermarking digital images for copyright protection. I.E.E. Proceedings on Vision, Signal and Image Processing, 1996,143(4):250–256.

[3] F. Hartung and B. Girod. Watermarking of uncompressed and compressed video[J], Signal Processing, 1998, 66(3):283–301.

[4] IK Yeo, HJ Kim, Modified patchwork algorithm: A novel audio watermarking scheme[J]. IEEE Transactions on Speech and Audio Processing, 2003, 11(4): 381–386.

[5] P Buneman, WC Tan. Provenance in databases[C]. In SIGMOD '07: Proceedings of the 2007 ACM, SIGMOD international conference on Management of data, New York, NY, USA, ACM, 2007: 1171–1173.

[6] M. Srivatsa, S. Balfe, K. G. Paterson and P. Rohatgi. Trust management for secure information flows. In P. Ning, P. F. Syverson, and S. Jha, editiors, ACM Conference on Computer and Communications Security, pages 175-188. ACM, 2008

[7] Panagiotis Papadimitriou, Hector Garcia-Molina: A Model for Data Leakage Detection. ICDE 2009: 1307-1310

[8] Yin Fan, Wang Yu, Wang Lina, et al. A trustworthiness-based distribution model for data leakage prevention[J]. Wuhan University Journal of Natural Sciences,2010,15(3):205-209.