MEDINFO 2013
C.U. Lehmann et al. (Eds.)

doi:10.3233/978-1-61499-289-9-171

Simulating Cloud Environment for HIS Backup using Secret Sharing

Tomohiro Kuroda[a], Eizen Kimura[b], Yasushi Matsumura[c], Yoshinori Yamashita[d], Haruhiko Hiramatsu[e], Naoto Kume[f]

[a] *Division of Medical Information Technology and Administration Planning, Kyoto University Hospital, Japan*
[b] *Department of Medical Informatics, Ehime University Hospital, Japan*
[c] *Department of Medical Informatics, Osaka University Medical Hospital, Japan*
[d] *Department of Medical Informatics, University of Fukui Hospital, Japan*
[e] *Department of Medical Informatics, Hyogo College of Medicine, Japan*
[f] *Graduate School of Informatics, Kyoto University, Japan*

Abstract

In the face of a disaster hospitals are expected to be able to continue providing efficient and high-quality care to patients. It is therefore crucial for hospitals to develop business continuity plans (BCPs) that identify their vulnerabilities, and prepare procedures to overcome them. A key aspect of most hospitals' BCPs is creating the backup of the hospital information system (HIS) data at multiple remote sites. However, the need to keep the data confidential dramatically increases the costs of making such backups. Secret sharing is a method to split an original secret message so that individual pieces are meaningless, but putting sufficient number of pieces together reveals the original message. It allows creation of pseudo-redundant arrays of independent disks for privacy-sensitive data over the Internet. We developed a secret sharing environment for StarBED, a large-scale network experiment environment, and evaluated its potential and performance during disaster recovery. Simulation results showed that the entire main HIS database of Kyoto University Hospital could be retrieved within three days even if one of the distributed storage systems crashed during a disaster.

Keywords:

Business Continuity Plan, Secret Sharing, Hospital Information System, Disaster Medicine.

Introduction

A business continuity plan (BCP) enables hospitals to continue their function during disasters. It is important for a country, however, to maximize the cost-efficiency of its social healthcare system.

Introduction of the information communication technologies into hospitals has made the hospital information systems (HIS) more important than ever. An HIS contains all health records, clinical guidelines, available laboratory reports, medicines, and other important information for clinical activities. An HIS also supports clinical activities through advanced functions such as scheduling system and auto ID barcode-enabled medication administration system (ABMA). Therefore, the loss of HIS could drastically decrease the performance of a hospital in times of a disaster. Therefore it is crucial to develop ways to secure and recover HIS data and include these procedures in BCPs.

Patient records contain privacy-sensitive data. Laws in most countries aim to achieve maximum security for health records. Although the Ministry of Healthcare, Labor, and Welfare (MHLW) of Japan has allowed hospitals to store electronic patient records (EPRs) at remote data centers since 2010 [1], the guidelines of the Ministry of Internal Affairs and Communication (MIC) of Japan require that these data centers keep all applications, platforms, servers, and storage devices under the control of Japanese law [2]. This requirement not only causes additional costs for providers, but also makes it easier for hackers to break through systems and access data, as they need only attack a limited number of data centers.

Another solution is a mutual exchange of HIS data backup between hospitals. After the 2011 Tohoku Disaster, Ishinomaki City Hospital, which lost all of its storage servers during the tsunami, was able to retrieve its EPRs from a backup stored at Yamagata City Hospital, under a mutual backup exchange agreement. Although such exchanges are effective, they are also expensive, especially when all data must be kept confidential.

The recent advancements in information security technologies provide secured communication and storage, e.g., secure socket layer (SSL) and the secure storage such as self encryption drives (SED) using advanced encryption standard (AES). Although several commercial remote backup solutions provide good security using these technologies, they increase the cost. Here we propose and evaluate a data backup method that is flexible, secure, and cost-effective.

Materials and Methods

Secret Sharing

Secret Sharing, also called secret splitting, is a method to distributing a secret among multiple participants by splitting original secret message into multiple fragments and distribute them among the participants. Each fragment by itself is meaningless, but sufficient number of the pieces can be put together to reveal the original data. This method was originally proposed as an electric tally, which distributes the risk of losing and abusing key among multiple administrators, and was proposed by Blackley [3] and Sharman [4] independently in 1979. The

concept is now widely applied for data encryption methods such as pretty good privacy (PGP).

The (*k, n*) threshold scheme is a way to achieve secret sharing, by splitting the original secret message into *n* pieces so as to reveal the original message from *k* pieces among them (k-out-of-n secrecy). Figure 1 shows the basic idea of (2, 3) threshold scheme. Secret message *M* is encrypted by equation (1) and denoted by three points (*A, B*, and *C*) on the line generated by equation (1). The three points are stored separately.

$$y = nx + M \tag{1}$$

As two points define a straight line, sets of two out of three pieces of data shown in equation (2) reveal original secret message *S*. Data set represented by equation (2) is called a qualified set. On the other hand, a set of single piece of data represented by equation (3) is called forbidden set, and by itself will not disclose the original secret message.

$$S_1 = \{x, y | x, y \in \{A, B, C\}, x \neq y\} \tag{2}$$

$$S_2 = \{x | x \in \{A, B, C\}\} \tag{3}$$

Unlike conventional computationally secure encryption methods, the security of (*k, n*) threshold scheme is based on information theory. Even a processer with unlimited processing power and memory cannot disclose the original message from the forbidden set.

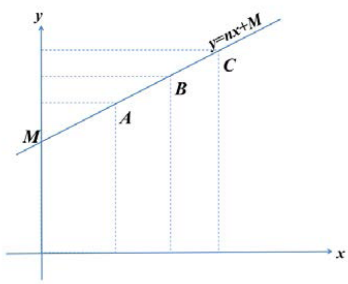

Figure 1 – (2, 3) threshold scheme secret sharing

In this scheme, the size of each piece cannot be smaller than the size of the original message. In order to decrease total size of data, Yamamoto [5] proposed (*k. L, n*) threshold scheme. Although sets of more than *k* - *L* and less than *k* fragments, called ramp set, reveal part of original secret message, this scheme makes the size of each fragment into *1/L* of original message. Therefore, total size becomes *n/L*.

Secret sharing using (*k, L, n*) scheme enables hospitals to backup the HIS data and removes privacy-sensitive nature from its distributed fragments. In addition, the method provides *n* - *k* redundancy. In this way, if more than *n* hospitals start mutual exchange of backups individual hospitals need not worry about leakage of entrusted privacy-sensitive data. The method also enables hospitals to utilize low-cost cloud storage services for backups, consequently securing each fragment of data by the redundancy of cloud storage services themselves.

The secret sharing is quite suitable for HIS backup, however, there are no commercial or academic trials known to authors that use this technology for HIS backup. In this paper, we evaluate feasibility of the method through simulation.

Simulating Mutual Backup Exchanging Environment

We set up simulation environment as shown in figure 2. Five hospitals, fully connected to each other via a gigabit network, participated in a mutual HIS backup exchange. Each hospital distributes 40 GB of HIS data using (3, 2, 4) threshold scheme, which is most appropriate for balancing various parameters, such as required storage size, bandwidth, calculation, and acceptable down time of data centers [6].

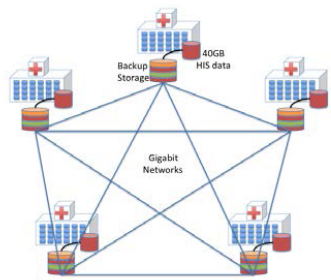

Figure 2 – Simulation setup

The authors have deployed SecureCube / Secret Share of NRI Secure Technologies Ltd. [7] into StarBED3 [8,9], the large scale network experiment environment of National Institute of Information and Communications Technology (NICT). Among various secret-sharing methods available on the SecureCube/Secret Share, we selected Matsumoto's [6] algorithm for taking advantage of computational speed.

Equations (4) shows Matsumoto's [6] encryption algorithm. The algorithm splits original secret message *s* into fragments *(a, b, c, d)* with auxiliary data *r*. Here, *u* is mediation variable. The auxiliary data and the fragments consist of half number of elements of original message, where the length of each element is identical. As the algorithm is based on simple bit-wise XOR operation, it does not require much computational power for encryption and decryption.

The randomness of the generated fragments is dependent on the randomness of the auxiliary data *r*. To reveal the original secret message, a hacker needs to collect qualified set of fragments from multiple sites, and to know proper decryption algorithm and auxiliary data.

$$s = (s_0, s_1, s_2, s_3, s_4, s_5) \quad r = (r_0, r_1, r_2)$$

$$a = (a_0, a_1, a_2) \quad b = (b_0, b_1, b_2) \quad c = (c_0, c_1, c_2) \quad d = (d_0, d_1, d_2)$$

$$u = (u_0, u_1, u_2, u_3, u_4, u_5)$$

$$\begin{array}{lll} u_0 = s_3 \oplus s_2 & u_1 = s_0 \oplus s_1 & u_2 = s_4 \oplus s_2 \quad u_3 = s_4 \oplus s_5 \\ u_4 = r_2 \oplus s_3 \oplus s_1 & u_5 = u_1 \oplus u_2 & u_6 = u_3 \oplus r_1 \oplus s_0 \\ a_0 = u_1 \oplus r_0 & a_1 = u_0 \oplus r_1 & a_0 = u_3 \oplus r_2 \\ b_0 = u_2 \oplus r_0 & b_1 = u_6 \oplus s_2 & b_2 = u_4 \oplus s_5 \\ c_0 = u_o \oplus r_0 \oplus s_5 & c_1 = u_o \oplus r_1 \oplus s_4 \oplus s_1 & c_2 = u_5 \oplus r_2 \oplus s_5 \\ d_0 = u_5 \oplus r_0 & d_1 = u_6 \oplus s_3 & d_2 = u_4 \oplus s_4 \end{array} \tag{4}$$

The simulation performed under the scenario is shown in figure 3. In the first step ("before disaster"), five hospitals distribute their respective HIS data to each other. In the second step ("under disaster"), a disaster hits hospital A, which loses its own HIS and its data center for mutual backup exchange. In this condition, another hospital (hospital B) may retrieve the HIS data of hospital A to treat its patients. At the same time,

all the distributed data must be reorganized to recover redundancy. After a while, the hospital A recovers from the damage of the disaster and regains normal operations. At this stage ("after disaster"), all the data needs to be reorganized to rebalance whole system.

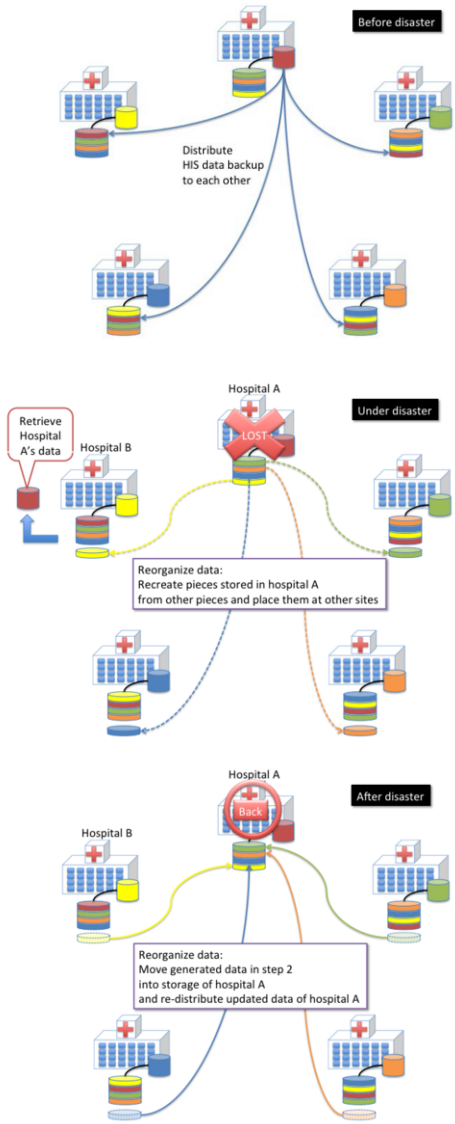

Figure 3 – Simulation scenario

The authors measured the time to perform backup, retrieval, and reorganization of the data. The measurements were scaled to match the size of the data set at the Kyoto University Hospital (KUHP) to evaluate the feasibility of the service.

KUHP has been developing its HIS since 1971. It introduced fully functioning clinical physician order entries (CPOEs) in 1990, picture archiving and communicating system (PACS) in 1993, and electronic patient records (EPRs) in 2005. The total size of the data stored in HIS at the end of 2012 was approximately 50 TB including about 40 TB of data in PACS. The main components of the system are backed up using the flash copy feature of DB2, and 1 TB of data are moved to backup storage each day. Hence, we scaled up our simulation to estimate the required time to backup, retrieve and reorganize 1 TB of data.

Result and Discussion

Performance

Table 1 – Simulation Result

	Before	Under disaster		After
	Backup	**Retrieve**	**Reorganize**	**Reorganize**
40GB	03:05:35	02:20:03	03:29:47	03:07:43
1TB	77:19:35	58:21:15	87:24:35	78:12:55

(hour:minute:second)

Table 1 shows the simulation results. It took three to four days to store, retrieve, and reorganize 1 TB of data. Although it is impossible to store daily backups to the platform, the hospitals may perform full backup once a week and store differentials daily. As the Matsumoto's [6] algorithm is suitable for processing multiple small files simultaneously, combination of weekly full backup and daily differential ones may increase the performance of the platform. These results indicate that secret sharing is a realistic method for backing up an HIS.

In the real world, hospitals are not fully connected to each other via a gigabit network, and damage caused by a disaster may degrade network performance. Therefore, future studies should perform more simulations under various conditions.

On the other hand, the simulation results clarified that the performance bottleneck was not the bandwidth but the computation. As the Matsumoto's method [6] is based on simple XOR operations, parallel processing may increase the computational performance. The computational performance is also affected by various parameters of an original data set as well as the computational platform. A detailed analysis of performance using real HIS data is therefore necessary.

To decrease the required time for data retrieval in case of disaster, the data should be reorganized before secret sharing. Clinicians need access to the basic profiles of patients including information on allergies, recent prescriptions, and laboratory tests conducted in the recent past. The experiences of the Kobe Disaster in 1995 and the Tohoku Disaster in 2011 revealed that the detailed clinical information is required after the initial triage stage. Hence, three days seems acceptable timeframe for retrieval of the necessary data for the second stage of disaster response.

Storage

The exchange platform needs to provide double buffer architecture, like video cards of computer, in order to secure at least one generation of backup. Additionally, a temporary storage to retrieve data in case of disaster should be larger than the largest backup data.

When m hospitals participate the mutual exchange platform using the (k, L, n) threshold scheme, m must be bigger than n to maintain redundancy, even if one hospital is lost due to a disaster. Assume a total size of backup is X_{total} and the size of largest backup is X_{max}. Then, each participating hospital needs to contribute storage defined by equation (4).

$$S = 2nX_{total}/mL + X_{max} \quad (m-1 \geq n) \qquad (4)$$

Using our same simulation setup, *S* becomes 200 GB. If all of the hospitals have 1 TB of data, *S* becomes 5 TB. However, if each backup consists of a single data set and the sizes of backups are different equation (4) will not provide sufficient storage. To get around this, the data can be compressed before secret sharing. The compression ratio depends on the features of the original data set. Therefore, evaluation using real HIS data with various compression methods is also necessary.

Legal issue

There are several conflicts between MHLW and MIC guidelines regarding the treatment of data encrypted by computationally secure encryption methods. However, many lawyers agree that each fragment generated by secret sharing can be treated as non-private information. Thus, secret sharing may overcome such conflicts. In addition, all hospitals participating in a mutual backup exchange platform would be able to backup their systems without taking extra security precautions or signing complex contracts regarding the treatment of privacy-sensitive data. This aspect of secret sharing decreases the social and economic barriers that can prevent such platforms from being used for healthcare data (see Fig. 4).

On the other hand, we have to keep in mind that this solution will create new challenges. If a hospital is lost, the medical professionals at another hospital may need to retrieve the EPRs of the lost hospital to provide medical support. Hence, agreements among hospitals must be carefully designed to allow medical professionals of other hospitals to retrieve the data of partner hospitals under certain circumstances.

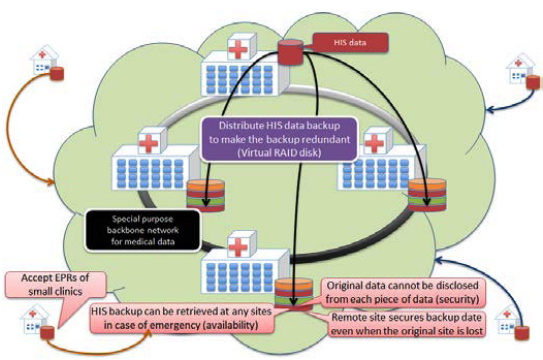

Figure 4 – A sketch of healthcare data cloud social platform

Conclusions

We evaluated the use of a secret sharing method for backing up HIS data. Simulations showed that secret sharing is a feasible method for backing up and securing data among collaborating hospitals. However, further analyses using real HIS data are required to address the remaining questions. To that end, we are already developing a mutual HIS backup exchange platform among five university hospitals in Japan for further evaluation. The results of future work may provide sufficient information to create a real working social platform for HIS backups.

Acknowledgements

This research is partly funded by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications (MIC) of Japan. The authors are grateful for the support from the Hokuriku StarBED Technology Center, National Institute of Information and Communications Technology (NICT), NRI Secure Technologies Ltd., and IBM Japan.

References

[1] The Ministry of Healthcare, Labour and Welfare of Japan, The Guideline for the Security Management of Medical Information Systems version 4.1, 2009.

[2] The Ministry of Internal Affairs and Communications of Japan. The Guideline Regarding Safety Control when ASP/SaaS Business Workers Handle Medical Information version 1.1 2009.

[3] Blackley GR. Safeguarding Cryptographic Keys. Proc. the National Computer Conf. 1979: 313.

[4] Shamir A. How to Share a Secret. Comm. ACM 1979: 612-3.

[5] Yamamoto H. Secret Sharing System Using (*k, L, n*) Threshold Scheme. Electronics and Communications in Japan 1986; 69(9): 46-54.

[6] Matsumoto T, Seito T, Kamoshida A, Shingai T, Sato A. High-speed Secret Sharing System for Secure Data Storage Service. Proc. Symp. Cryptography and Information Security 2012; 1E2-4. (Japanese)

[7] NRI Secure Technologies. SecureCube / Secret Share. http://www.nri-secure.co.jp/service/cube/secretshare.html.

[8] Miyachi T, Nakagawa T, Chinen K, Miwa S, Shinoda Y. StarBED and SpringOS Architectures and Their Performance. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012; 90(1): 43-58.

[9] StarBED Project. http://www.starbed.org/.

Address for correspondence

Tomohiro Kuroda
Division of Medical Information Technology and Administrative Planning
Kyoto University Hospital
Shogo-in Kawahara-cho 54, Sakyo-ku, 606-8507, Kyoto, Japan
Email: tomo@kuhp.kyoto-u.ac.jp