A Cloud-Based Watermarking Method for Health Data Security

Zhiwei Yu[a], Clark Thomborson[b], Chaokun Wang[c], Jianmin Wang[c], and Rui Li[c]

[a]*Department of Computer Science and Technology, Tsinghua University, Beijing, China*
[b]*Department of Computer Science,The University of Auckland, New Zealand*
[c]*School of Software, Tsinghua University, Key Laboratory for Information System Security, Ministry of Education*
Tsinghua National Laboratory for Information Science and Technology(TNList), Beijing, China

POSTER PAPER

Abstract—**Private health information once confined to local medical institutions is migrating onto the Internet as an Electronic Health Record (EHR) that is accessed by cloud computing. No matter where it is hosted, health data is subject to security breaches, privacy abuses, and access control violations. However, novel technologies have new vulnerabilities, and allow new mitigations. In this paper, we propose a watermarking method in the architecture of cloud computing, to mitigate the risks of insider disclosures. Our design and preliminary implementation are accomplished by exploiting the MapReduce mechanism in the cloudlet we built. Our evaluation shows that our proposal addresses all of the requirements of the Cloud Oriented Architecture (COA) framework of the Jericho Forum.**

Keywords—**Cloud Computing; Health Data Security; Watermarking**

I. Introduction

Health information is necessary for patients to receive treatment, and for medical practitioners to analyze and alleviate disease propagation [1]. Personal health information refers to “information that concerns a person’s health, medical history or medical treatment in a form that enables the person to be identified by a person other than the treating clinician” [2]. The Electronic Health Record (EHR) is being widely accepted due to the following reasons. It can meet the the need of highly mobile patients and help the doctors, nurses, and administrators in hospital management systems, with simultaneous access to healthcare records. At the same time, it improves decision-support and clinical research [3]. Some healthcare organizations allow public access to their health statistics via the network, with the goal of promoting user involvements and researchers’ access to relevant datasets.

Cloud computing is a style of Internet-based computing which refers to both the applications delivered as services as well as to the hardware and systems software in the infrastructure that provide those services [4]. It has recently attracted great attention, both commercially and academically. Cloud computing has many perceived benefits such as reduced cost, increased storage, improved levels of automation and flexibility. These perceptions make a shift to cloud computing very attractive to individuals, the public sectors, and commercial organizations.

In 2008, Google offered a “Health” portal to individuals who wanted their EHR to be available to their health providers, and to themselves. Write-access to this cloud computing service was terminated in January 2012, due to insufficient demand, with read-access continuing until January 2013 [5]. Microsoft’s HealthVault service, launched in 2007, is continuing to provide cloud-based access to individual EHRs [6]. Microsoft is also providing infrastructure support to health service providers who want cloud-based access to the EHRs of their clients [7]. In 2010, IBM and Aetna jointly announced a novel use of IBM’s cloud computing platform designed to “... help physicians and other health care professionals to quickly access patient information such as medical records, claims, medication and lab data gathered from multiple sources to create a detailed patient record” [8].

The financial and functional advantages of cloud computing are accompanied by an increased vulnerability to insider attacks. The identity and location of intermediaries and the service-providers are obfuscated by the cloud. Encryption can effectively prevent confidentiality leaks by intermediaries but not at the end-points. Service providers must be able to decrypt the user’s data, and if this service-provider relies, in turn, on other cloud-services then the user’s data may be readable by many entities in the cloud. What is required to increase the end-user’s “trust in the cloud”, by this line of reasoning, is some mechanism that can reliably detect and then punish any breaches of confidentiality.

The end-user must trust the entity (perhaps Google Health, or Microsoft Health Vault) who manages their EHRs. To be trustworthy, the EHR manager must have some means to detect (and then to punish) any breaches of confidentiality by any enterprise or organization to whom they release the record. In this article, we explore the possibility that watermarks on leaked EHRs can be effectively detectable by a trusted organization “in the cloud”, without compromising the medical accuracy or functional efficiency of these EHRs.

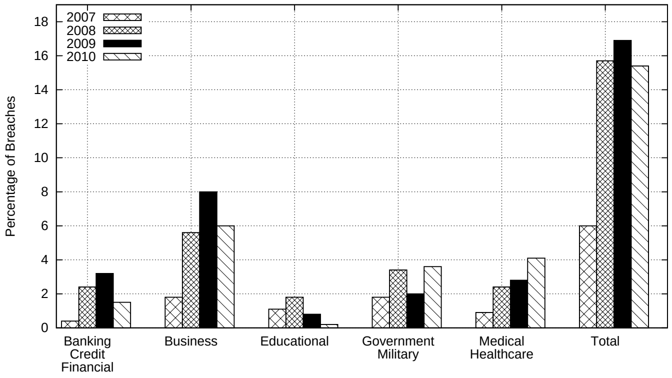

Figure 1. Insider theft proportion in data breach

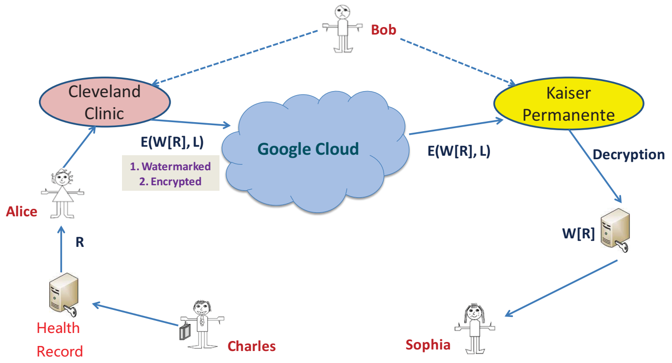

Figure 2. Insider threat model

The remaining sections are organized as follows: Section 2 proposes an insider threat model for the health data storage in the cloud, finding the security gaps. Section 3 discusses our cloud-based watermarking method including architecture and implementation, for enhancing cloud security. In Section 4, primary experimental results are presented to evaluate our method, according to the assessment criteria addressed by Jericho. The limitations of our proposal are discussed in Section 5. Finally, we summarize our contributions and discuss further research needed in Section 6.

II. Insider Threat to Confidentiality of Health Care Information

In this section, we propose an insider threat model. Medical personnel who can access the healthcare information might release it to outsiders for spite, revenge, or profit. It is believed that insiders rather than outsiders induce the main threats to privacy in computerised clinical record systems, and this may exacerbated by the data aggregation [2] [3]. For example, a teller in a bank may access to a customer's account; a cameraman can retain the models' pictures if he wish. Health systems are not likely to be different. Clinicians must have the ability to read, and to modify appropriate portions of, their patients' EHRs. If these data holders are untrustworthy, the consequences can be serious and irretrievable. Machado, a front desk clerk of Cleveland Clinic, sold the medical information of more than 1,100 patients to who used the stolen identities to file an estimated \$7.1 million in fraudulent claims [9].

In order to confirm the severe situation of insider theft, we collect and analyze the statistics about insider theft proportion in data breach form Identity Theft Resource Center (ITRC) which is a nonprofit, well respected organization dedicated to the understanding and prevention of identity theft. As we can see from Figure 1, insider thefts account for a high proportion of data breach (6%, 15.7%, 16.9% and 15.4% in 2007, 2008, 2009 and 2010 respectively [10]), which has more than doubled between 2007 and 2008. In particular, the insider theft proportion of category "Medical Healthcare" keeps increasing over years.

In order to illustrate the insider threats further in details, we take a specific medical scenario as an example. Before the description, specific formalizations and abbreviations should be introduced:

- *CC*: Cleveland Clinic, a healthcare clinic
- *KP*: Kaiser Permanente, a healthcare institution
- *R*: Health Record
- *E(R)*: Encrypted form of *R*
- *L*: Label of *R*, which indicates the authorized readers of *R*
- *E(R, L)*: Encrypted form of *R* and *L*
- *W[R]*: Watermarked form of *R*
- *E(W[R], L)*: Encrypted form of *W[R]* and *L*

As shown in Figure 2, Charles (a public figure, perhaps a movie star or a politician) is admitted to *CC* for medical treatment. He worries that his medical admission might be known to the general public, and what's worse, the details of medical condition to be released to journalists. Charles' doctor Alice decides that a consultation with a specialist Sophia at *KP* is medically important, so she releases Charles' medical record *R* to Sophia (via the Google-hosted cloud). *R* is encrypted, at all times while it is within the cloud. We write *E(R)* for the encrypted form of record *R*, where the encryption uses *KP's* public key. So long as *KP* maintains adequate security for its private key, then *E(R)* can only be decrypted on a workstation in *KP*.

In order for *KP* to be able to enforce the appropriate access control on *R*, *CC* and *KP* must agree on a meta-data label *L* (perhaps based on XACML) which indicates that only Sophia should be allowed to read *R*. Thus *CC's* system should append *L* to *R* before encrypting it with *KP's* public key: *E(L, R)* should be written to the cloud, rather than *E(R)*. Note that *KP* must be trusted by *CC* to enforce access controls, for *KP's* information systems are not under the control of *CC*.

Maintaining access control within *CC* and *KP* is important, but avoiding medical mistakes and improving medical efficiency is even more important. For this reason, fine-grained access control on medical records is typically controlled only by physical proximity. Anyone can read a medical record, if they are near a workstation that is authorized (under its current login) to display that medical record. Cryptographic security is used only for a coarse-grained access control, whereby only a restricted number of workstations and logins are authorized to display a medical record. Thus, although the label L on medical record R might indicate the name of the specialist at KP who is authorized to read R, others at KP (and at CC) will be able to read R.

The security threat in this scenario is that some unauthorized party (Bob in Figure 2) reads R and reveals its contents to a journalist. Formally, Bob has mainly three types of attacks [9]: privacy violations (sell the healthcare data), medical fraud (billing payers for treatment never rendered) and financial/medical identity theft (obtain medical services).

The cryptographic protection on R will make it very difficult for an outsider to read R, but many medical and administrative workers at KP or CC can surreptitiously read R. Audit logs on the workstations at CC and KP would reveal the timestamps of all accesses to R, however we would need records of all individuals in the vicinity of these workstations in order to construct a list of suspects – and the list of suspects would be dauntingly long, if R has been read multiple times. The suspects could be interviewed, but this would be a very expensive and time-consuming investigation. Video recordings, or other records of people within viewing range of a workstation would be helpful, but a careful analysis of these records would be required in order to develop a short list of primary suspects. Furthermore, such record keeping would be very expensive and intrusive. We conclude that tight control over the confidentiality of medical records will be very difficult and expensive to achieve. This raises the question of whether we can find an inexpensive, non-intrusive way to mitigate (but not eliminate) the risk of disclosure of R.

III. A Cloud-Based Watermarking Method

A. Architecture

Our watermarking architecture is shown in Figure 3. We insert a watermarking process into the Service Layer of a cloud [11]. This layer provides "software as a service" (SaaS) to end-users. We do not modify the lower layers of "platform as a service" (PaaS) and "infrastructure as a service" (IaaS). Using our watermarks, Cleveland Clinic (CC) can mark all of its medical records R with a cleartext label L (to specify the allowed readers of R) and a stegotext which contains a copy of L as well as the name of the providing clinic (CC, in this instance) and the name of the person (in this case a doctor) who released the watermarked R to the cloud.

Our proposal of putting the watermark apparatus in the EHR cloud is similar, technically, to the Watermark Clearance Center proposed by Kwok for mass-market digital rights management [12] but with some differences in the watermark content.

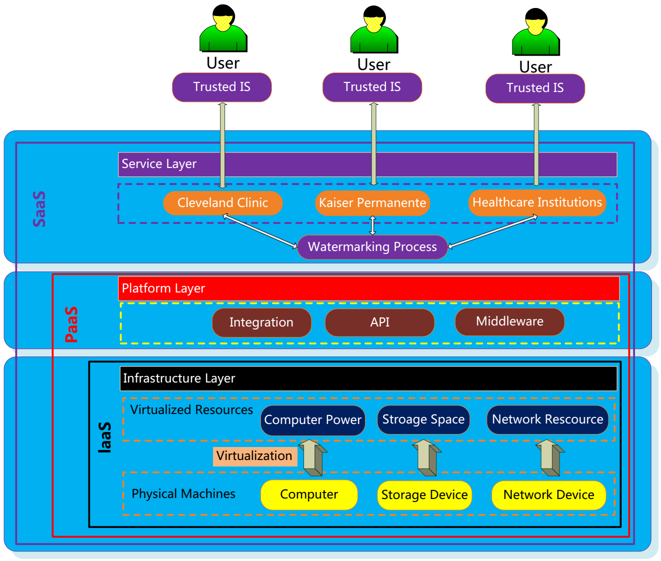

Figure 3. Watermarking enhanced cloud architecture

B. Watermarking Process

The watermark embedding stage in the cloud should be capable of handling a huge number of healthcare records, which sets high requirements for both the watermarking technique and the processing method. The watermarking algorithm should designed by someone who is expert in both watermarking and EHR, for the reasons discussed in the previous subsection. As to the processing method, we suggest the use of MapReduce, which is a programming model and an associated implementation for processing and generating large datasets in cloud computing. We conducted some preliminary experiments using Hadoop, which is an open source Java software framework that supports massive data processing across a cluster of servers. Hadoop uses the MapReduce paradigm to divide a large dataset into many small fragments and distributes them to each of the slave nodes. All slave nodes run the MapReduce executable on their subsets of the data. Hadoop then gather the results from all of the slave nodes, and make them into a finished output. Amazon [13] suggests image watermarking as a suitable application for their Elastic MapReduce service: "Typically, the processing involves performing relatively simple operations on very large amounts of data, for example, adding a watermark to $1,000,000$ digital images." Our experimentation is directed at evaluating the performance characteristics of watermark embedding and recognition, when these are conducted on a private cloud running Hadoop. Readers may wish to re-run our experiments on their own computing infrastructure, if they want to compare its cost-performance to that offered by commercial cloud providers such as Amazon. Our experimentation also indicates

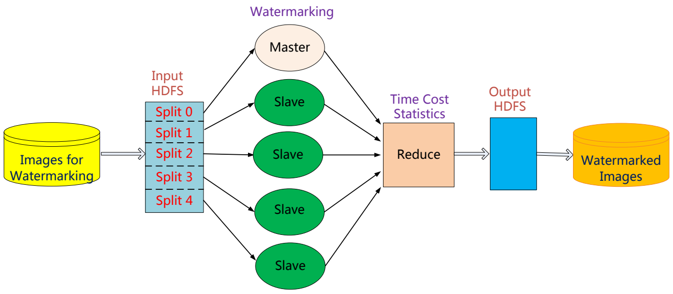

Figure 4. Watermarking process in MapReduce

how to evaluate the accuracy of a watermarking process.

As shown in Figure 4, images for our watermarking experiments are firstly put into the Hadoop Distributed File System (HDFS). As its name suggests, HDFS is a distributed file system that provides high throughput access to application data), and the large amount of data. The data must be subdivided in a preprocessing step on a master processor, to create many small tasks that will be performed independently by slave processors.

The pseudocode displayed in this section indicates how we used MapReduce to embed watermarks into the images. A Map task consists of watermarking a list of images. The images are watermarked [14] with different watermarks according to the labels, healthcare providers and doctors, and then put back to the HDFS. After all of the Map tasks have been completed, the Reduce task begins. In more complicated applications of the MapReduce paradigm, slaves perform some important computations during the Reduce task, but in our experimentation the Reduce step merely consists of recording the runtime of the computation.

```
Procedure Watermarking($iListName$)
Input   : $iList$ – A list of images to be watermarked;
          $K$ – Embed Key; $A$ – Watermarking Algorithm;
          $L$ – Label; $PRO$ – Healthcare Provider;
          $D$ – Doctor;
$iList$ = getImage(HDFS, $iListName$);
/* Get the images from HDFS */
for each image $I \in iList$ do
    $W \leftarrow (L, PRO, D)$;
    /* Generate a watermark */
    $I'$ = Embed.A($I$, $K$, $W$);
    /* Embed the watermark */
    putImage(HDFS, $I'$);
    /* Put the watermarked image to HDFS */
end for
EmbeddingTime = getTimecost(Embed)
```

We note that several components in the architecture must be trusted: the MapReduce provider, some IS infrastructure at every participating healthcare provider, and the commmunication channel linking the MapReduce with these IS infrastructures. With these assumptions, a medical record can be watermarked securely and differently (by the MapReduce service) every time it is released to a different healthcare provider – these watermarks would indicate the healthcare provider. Furthermore, if the MapReduce latency is low enough for this, and the computational costs are low enough, then the medical record can be watermarked differently every time it is released to a different workstation – these watermarks would indicate the name of the healthcare worker to whom the record is being released.

Our private cloudlet has the following characteristics: five computational nodes, with each node having a Pentium (R) Dual-Core CPU E6300 2.8GHz, 4G memory, 800G hard drive, and a 1 GHz ethernet connection to the other nodes. The five nodes provide services as slave nodes, one within which deployed as the primary node at the same time. All nodes run Ubuntu 9.10 with JDK version 1.6.0_12. In the Hadoop profile, we limited the size of each of the emulated processor's virtual memory to 2 GB. the largest sub-process virtual memory to 2G in Hadoop profile. We collected a dataset consists of 4000 BMP images with sizes ranging from 200KB to 2MB. Our experimentation is preliminary – it is intended to demonstrate how to conduct the relevant performance evaluation, rather than to estimate the performance of a fullscale implementation. To form an accurate estimate of performance, it would be important to collect a dataset of imagery (or other EHR documents) that is representative of what would actually be used in an application. It would also be important to perform scaling experiments, to determine how much the computational performance degrades as the number of nodes (and their geographic dispersal) is increased.

IV. Evaluation

We assert that an EHR watermarking system should be assessed on the following five attributes: usability, availability, performance, effectiveness and agility. These attributes are defined by the COA (Collaboration Oriented Architectures) which has been developed, by the Jericho Forum [15], to meet the information-system requirements of large corporate and governmental organizations. Below, we summarise our assessment of our proposed system, based on its structure and our preliminary experimentation.

Usability. We did not test usability in our experimentation. When developing the GUI and supporting procedures for a complete implementation, usability should be a top-of-mind consideration in all design decisions. It is never easily achieved; however we see no structural impediments in our design. Our proposed watermarking process will not be visible to end-users, and can be conducted transparently (with roughly a few extra CPU-seconds of overhead, per record) during the IaaS store-and-retrieve functions of any cloud-based EHR system. When alarms are issued by the watermark detector, these should be investigated for credibility, and a mitigating response must be launched if the alarm is found to be credible. Any false alarm will, we suspect, be reported as a usability fault, so we recommend that watermark detection

Image Number	10	100	400	800	1000	2000	4000
Time Cost in Local PC (s)	0.5	5	23	45	54	110	223
Time Cost in Cloudlet (s)	22	24	31	34	36	45	52

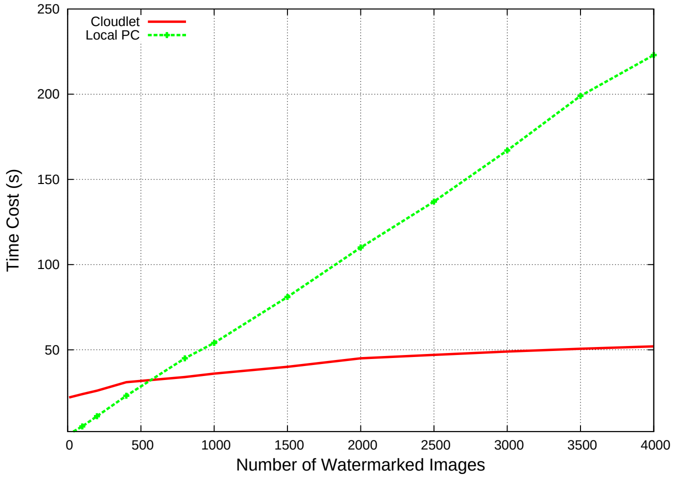

Figure 5. Comparison of time cost between local PC and cloudlet

thresholds be set to the highest level which still allows some true-positives to be reported. Our cloud-based watermarking method would be readily managed by Google's cloud platform, and will not disturb patients or medical practitioners because the watermarking process is transparent to users.

Availability. The Jericho Forum defines this goal as follows: "Information shared between collaborating organizations should not be rendered unavailable either by mistake or by an adversary" [15]. Cloud computing provides an on-demand services with very high availability. However, if a hospital's internet service is ever disrupted, its medical practice must be able to continue. We conclude that our system must have a fail-safe option in which it releases EHRs from a local store without passing them through the watermarking process. When operating in its fail-safe mode, our system would offer no protection against insider abuse but would not degrade the availability of medical services.

Performance. According to the Jericho Forum, "Security measures should not greatly affect the latency, bandwidth, or total cost of data retrieval, storage, or transmission" [15]. We measure this aspect experimentally, as described below. As shown in Figure 5, even when the number of watermarking images is extremely small, there is only a little disadvantage to running MapReduce in comparison to running a standalone watermarking job. As the number of images rises, the MapReduce overhead becomes negligible, and the cloudlet (running MapReduce) becomes much faster than the single processor. By looking a little more carefully at our data, we can extract the following performance parameters.

On a single PC, each image requires about 50 ms of processing time. On the cloudlet, the average time per image drops from about 220 ms to about 13 ms, as the number of images increases from 10 to 4000. We note that the crossover, where the latency of watermarking is almost equal on the single PC versus the cloudlet, is at approximately 600 watermarking jobs. The position of this crossover is, obviously, dependent on our experimental setup, however it should be clear that an externally-hosted and rapidly-scalable cloud will have a significant performance and cost advantage over an in-house computing system, if a wildly variable number of watermarking jobs are required each minute.

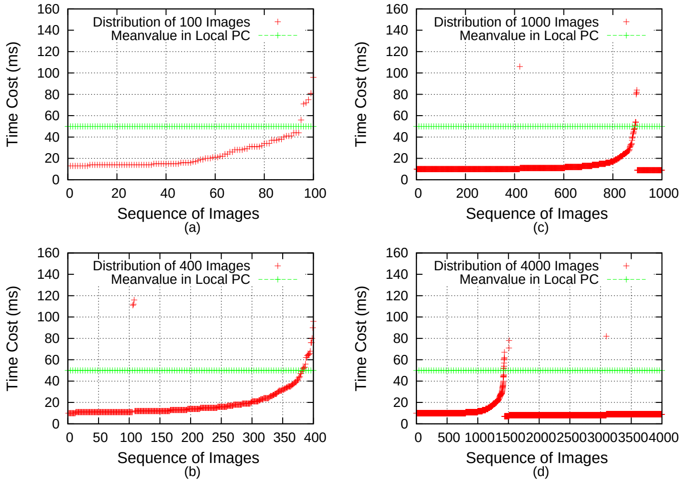

Figure 6. Distribution of time cost in cloudlet

In Figure 6, subgraphs (a)-(d) show the processing time for each image in image distributions of size 100, 400, 1000 and 4000. The per-image latency is apparently quite variable on the cloud, however images are typically processed in about 10 ms. The work queue of the cloud apparently becomes overfull occasionally, leading to latencies above 10 ms. As indicated in Figures 6(c) and 6(d), the work queue can subsequently be emptied so that images are once again processed in about 10 ms apiece. We speculate that the 10 ms typical processing time for an image on the cloudlet, as compared to the 50 ms processing time on the single PC, can be attributed to our cloud's high-performance file system.

Our watermarking scheme will thus be sufficiently efficient, if a single EHR can be watermarked on a standalone PC at an acceptable time cost. We note that clinicians who produce some new (unwatermarked) component of an EHR will not be affected by our scheme; the watermarking process will merely add a few CPU-seconds to the latency of updating the master EHR record in the cloud. Researchers are already starting to produce suitable watermarks for medical imagery [16] [17].

Effectiveness. COA-compliant architectures should provide an effective approach to organizing and controlling secure data transport and storage [15]. In a locally maintained system, a single person with administrator's rights would be able to access any medical records. In a cloud-hosted system, we must take some care when connecting to the service provider, but the provider itself does not have to give (and we assert that a trustworthy provider would *not* give) any single person unrestricted access to all user data. Therefore, the watermarking module integrated in the service layer in a connection to a trustworthy service provider will have a relatively higher effectiveness for mitigating the insider-disclosure vulnerability. Furthermore, SaaS providers such as Google would be able to provide more consistent training and supervision of its cloud-system operators than any but the largest corporations would be able to provide for their in-house IT staff. Such training should significantly reduce the risk of operational errors which would compromise security. No watermarking system can provide an absolute assurance of security, because of the possibility of false-positive and false-negative watermark detections, especially when confronted with an expert attacker who is able to analyse the technical vulnerabilities of the watermarking process. However our proposed system will deter the inexpert attacker. It will also raise the bar for the expert attacker, who will have to expend time and resources to remove all of the watermarks they are able to find – and any expenditure of time and resources may leave some forensic trail. Furthermore, even the most expert attacker can never be completely certain that they have removed all of the watermarks which have been embedded. Thus watermarking can provide an important second line of defense for cases when cryptographic key-control has been breached.

Agility. "COA must take into account the dimensions of timeliness and flexibility, so as to enable development of business-driven enterprise architectures that are appropriately flexible and adaptable to facilitate changes in business operations with optimal rapidity and ease, with minimal disruption" [15]. Our design is modular – the watermarking component can be updated whenever a more suitable watermark is discovered, or whenever a previously-used watermark has been found to be insecure. However we note that shifting to a new watermarking method will have very significant costs. A difficult choice must be made: to rewatermark all files with the new method, or to continue to detect the old watermarks while also detecting the new ones, or to have a traitor-tracing capability only for the files with the new watermarks. We conclude that our system is only marginally agile, and for this reason (among others) it will be important not to "oversell" the traitor-tracing ability of this system.

V. Limitations

The scope of this paper is a proposal, a preliminary evaluation, and a novel method for mitigating the risk of an insider disclosure. We are not able to assess the resilience of a watermark against a removal or modification attack by an attacker who has technical expertise in watermarking. Our method will not mitigate the risk of an insider attack on the confidentiality of the original, unwatermarked, healthcare record. Finally, our analysis is based on the assumption that a suitable watermarking method can be find, which appropriately balances traceability against medical accuracy.

VI. Conclusion

To enhance the cloud computing platform with the watermarking component, this paper presents a novel method for mitigating the security gaps exist in healthcare data protection. We claim two contributions in this paper. The first is that we identify that watermarking methods can be employed for mitigating the insider threats, which is out of the consideration of previous watermarking researchers. The other contribution is that we made a cloud-based watermarking method by use of the advantages of cloud computing. The evaluation showed that our proposal fits in with the demands required by Jericho.

Acknowledgment

This work is supported by the National Basic Research Program of China (No. 2009CB320700), and National Natural Science Foundation of China (No. 61073005).

References

[1] S. Gao, D. Mioc, X. Yi, F. Anton, E. Oldfield, and D. Coleman, "Towards Web-based representation and processing of health information," *International Journal of Health Geographics*, vol. 8, no. 1, p. 3, 2009.

[2] R. J. Anderson, *Security in clinical information systems.* Computer Laboratory University of Cambridge: BMA, Jan. 1996.

[3] T. C. Rindfleisch, "Privacy, information technology, and health care," *COMMUNICATIONS of The ACM*, vol. 40, no. 8, pp. 93–100, Aug. 1997.

[4] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep., Feb. 2009.

[5] A. Brown and B. Weihl, "An update on Google Health and Google PowerMeter," Jun. 2011. [Online]. Available: http://googleblog.blogspot.co.nz/2011/06/update-on-google-health-and-google.html

[6] N. Gohring, "Microsoft says HealthVault still going strong," *ComputerWorld*, Jun. 2011. [Online]. Available: http://www.computerworld.com/s/article/9217970/Microsoft_says_HealthVault_still_going_strong

[7] Microsoft, "Extending the value of electronic health records through interoperability," Feb. 2011. [Online]. Available: http://www.microsoft.com/presspass/features/2011/feb11/02-21himss.mspx

[8] M. Alazraki, "IBM and Aetna launch health care support service," *Daily Finance*, May 2010. [Online]. Available: http://www.dailyfinance.com/2010/08/05/ibm-and-aetna-launch-health-care-support-service/

[9] M. E. Johnson, "Data hemorrhages in the health-care sector," in *Financial Cryptography*, ser. Lecture Notes in Computer Science, R. Dingledine and P. Golle, Eds., vol. 5628. Springer, 2009, pp. 71–89.

[10] Identity Theft Resource Center, "2010 data breach insider theft category summary," Dec. 2010. [Online]. Available: http://www.idtheftcenter.org/artman2/publish/lib_survey/ITRC_2008_Breach_List.shtml#

[11] Cloud Security Alliance, "Security guidance for critical areas of focus," Dec. 2009. [Online]. Available: http://www.cloudsecurityalliance.org/csaguide.pdf

[12] S. H. Kwok, "Watermark-based copyright protection system security," *Commun. ACM*, vol. 46, no. 10, pp. 98–101, 2003.

[13] Amazon, "Introduction to Elastic MapReduce." [Online]. Available: http://docs.amazonwebservices.com/ElasticMapReduce/latest/GettingStartedGuide/

[14] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *ICIP (2)*, 1994, pp. 86–90.

[15] Jericho Forum, *Position Paper – COA Framework*, Nov. 2008.

[16] G. Xuan, Y. Q. Shi, P. Chai, X. Tong, J. Teng, and J. Li, "Reversible binary image data hiding by run-length histogram modification," in *ICPR*. IEEE, 2008, pp. 1–4.

[17] M. Li, S. Narayanan, and R. Poovendran, "Tracing medical images using multi-band watermarks," in *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, vol. 2, 2004, pp. 3233–3236.