Journal of Digital Imaging

A HIPAA-Compliant Architecture for Securing Clinical Images

Brent J. Liu, Zheng Zhou, and H. K. Huang

The Health Insurance Portability and Accountability Act (HIPAA, instituted April 2003) Security Standards mandate health institutions to protect health information against unauthorized use or disclosure. One approach to addressing this mandate is by utilizing user access control and generating audit trails of the various authorized as well as unauthorized user access of health data. Although most current clinical image systems [e.g., picture archiving and communication system (PACS)] have components that generate log files for application debugging purposes, there is a lack of methodology to obtain and synthesize the pertinent data from the large volumes of log data generated by these multiple components within a PACS. We have designed a HIPAA-compliant architecture specifically for tracking and auditing the image workflow of clinical imaging systems such as PACS. As an initial first step, we developed HIPAA-compliant auditing system (H-CAS) based on parts of this HIPAA-compliant architecture. H-CAS was implemented within a test-bed PACS simulator located in the Image Processing and Informatics lab at the University of Southern California. Evaluation scenarios were developed where different user types performed legal and illegal access of PACS image data within each of the different components in the PACS simulator. Results were based on whether the scenarios of unauthorized access were correctly identified and documented as well as on normal operational activity. Integration and implementation pitfalls were also noted and included.

KEY WORDS: HIPAA, security, auditing, monitoring

INTRODUCTION

Health Insurance Portability and Accountability Act (HIPAA)[1,2] of 1996, Public Law 104-191, was officially instituted on April 14, 2003 to enforce healthcare providers to be compliant by April 2005 deadline. The major goal and focus of HIPAA is to set and enforce broad standards in the attempt to protect the privacy and security of health data throughout the patient care environment. To date, there are four types of standards in HIPAA:

1) Transaction and code set standards
2) Identifier standards
3) Privacy standards
4) Security standards

In this paper, we focus on the fourth standard type—security. HIPAA Security Standards[3] are aimed at the protection of confidentiality, integrity, and public availability of electronic health information against unauthorized use or disclosure. This is accomplished by utilizing administrative, physical, and technical safeguards. In particular, the technical safeguards consist of technical methods to assure security of the health data. One such technical method proposed by HIPAA is the on-demand generation of an audit trail that can record and examine information system activities such as data access of a specific patient. Specifically, HIPAA-compliant audit trails require the following information for the health data access:[4]

- Identification of the person who accessed the data
- Identification of the accessed data
- Where the data were accessed
- Timestamp of when the data were accessed

From the Image Processing & Informatics Laboratory, Department of Radiology, Keck School of Medicine, University of Southern California, Los Angeles, CA, USA.

Correspondence to: Brent J. Liu, 4676 Admiralty Way, Suite 601, Marina del Rey, CA 90292, USA; tel: +1-310-4489120; fax: +1-310-4489441; e-mail: brentliu@usc.edu

Online publication 15 December 2005
doi: 10.1007/s10278-005-9248-5

- Types of access (e.g., create, read, write, modify, delete)
- Status of access (e.g., success or failure)

Because health data and information is such a broad area containing vast amounts of data types, the major focus of this research is on clinical imaging data that are generated and distributed through picture archiving and communication system (PACS).

Some efforts have been achieved by developing HIPAA-compliant auditing tools for general health information systems.[5–7] These auditing tools generate audit trails by recording the health data transactions or changes in logs and extracting the pertinent auditing information from these logs on demand. This method is applicable for health information systems that have all the data transactions or data flow controlled by a centralized server, such as radiology information system.[8] However, the data flow is much different in integrated medical imaging systems, such as PACS. There is no single component that controls and records the data flow of all the multiple components within PACS. This makes it very difficult for these auditing tools to record all the data transactions and changes in PACS. For example, the PACS archive server, even within client–server architecture, has no control of the workflow of the CT modality and vice versa. Additionally, there are various other components within a PACS that require a system-wide architecture instead of a single component-based approach. Most current clinical imaging systems have no such ability to generate HIPAA-compliant audit trails, although they generate activity logs. Furthermore, although pertinent auditing information can be extracted from these logs to create audit trails, it requires tedious if not manual methods to produce the requested audit information and analysis. There is a lack of a formal methodology to interpret the potential large volumes of these log data and generate these HIPAA-compliant audit trails. Therefore, a HIPAA-compliant auditing architecture for integrated medical imaging systems needs to be tailored to the complex workflow.

In this research, we present the design and development of a HIPAA-compliant architecture specifically for tracking and auditing the image workflow of clinical imaging systems such as PACS. The architecture is designed to facilitate the generation of HIPAA-compliant audit trails of image data access for a specific patient so that various types of audit queries can be performed on demand. It also provides the mechanisms to automatically monitor the data flow of PACS and to facilitate the detection of unauthorized image access and other abnormal activities. As an initial first step, HIPAA-compliant auditing system (H-CAS) was developed based on parts of this HIPAA-compliant architecture. This initial H-CAS was implemented and evaluated within a test-bed PACS simulator located in the Image Processing and Informatics (IPI) lab at the University of Southern California.[9] Evaluation scenarios were developed, and results were based on whether the scenarios of unauthorized access were correctly identified and documented as well as on normal operational activity.

METHODS AND MATERIALS

Design Criteria

To apply the HIPAA-compliant architecture for auditing and tracking clinical images to various PACS generating different format log files, it must be independent from any individual PACS architecture or manufacturer. For this reason, we define the necessary architecture criteria as follows:

(1) HIPAA compliant. The ability to facilitate generation of the HIPAA-compliant auditing trail report in terms of who accesses it, when, where, what are accessed, access status, and access types.
(2) Open and extensible. Provide interfaces for integration of new auditing or monitoring techniques and the ability to support current HIPAA auditing requirements and accommodate new HIPAA additions in the future without affecting already existed components.
(3) Portable. Not tied down to any individual PACS or PACS architecture.
(4) No interruption of clinical PACS workflow. Any interruption on the workflow of PACS is avoided.

HIPAA-Compliant Architecture Design

Based on the criteria described above, the HIPAA-compliant architecture was designed as a four-layer system shown in Figure 1. The first layer (the lowest layer) is the record layer, consisting of various logs within PACS components. By logically separating PACS logs from other logs and layers, independence from PACS and portability can be achieved. The second layer is the audit layer, which includes a centralized auditing database and other audit data analysis and interpretation tools. HIPAA-compliant audit trails can be generated based on the auditing database. This layer also enables us to automatically monitor the data flow of PACS, which greatly assists PACS management. The third layer is the notification layer, which has a notification component sending warning or alert messages of abnormal events to end users, such as PACS administrators. Finally, in the fourth layer, end users can decide to take certain actions against these abnormal events. These layers will be described in more detail in the following paragraphs.

Record Layer

This first layer is the data resource layer, including but not limited to the various types of log data shown in Figure 1. PACS application logs are event logs generated by the individual PACS applications. For example, an image query/retrieve event in PACS archive server may include such information as time, local host name, Digital Imaging and Communications in Medicine (DICOM) application entity title, patient information, and query/retrieve status. In addition, PACS "user logs" record any login events of users for each individual PACS component. Other computer system logs generated in PACS components, such as application access logs, can also provide supplement information.

Because of the flexibility of this architecture, new logs can also be added to this layer. For example, an image integrity log can be added to record image data integrity verification events. Data integrity, as one requirement of HIPAA Security Standards, refers to protecting image data from being altered or destroyed by unauthorized users. A lossless digital signature embedding (LDSE) method has been developed to ensure the data integrity of medical images at IPI laboratory.[10] By recording signature verification time, local machine, and signature verification status in the integrity log, the LDSE method can provide logs to generate HIPAA-compliant audit trails on the data integrity of image.

To extract and interpret the pertinent information from thousands of log events requires proper methodology, which will be addressed in the second layer, the audit layer.

Audit Layer

As shown in Figure 1, the audit layer is the heart of the architecture. It collects the audit data from distributed PACS components and stores the data in a centralized auditing database. The database is then used for audit analysis and automatic monitoring. Currently, there are seven components in this layer: audit log collector, syslog server, log data norlmalizer, auditing database, audit analysis tool, role-based policy, and monitor tool.

Audit Log Collector. Because the audit data are scattered within large volume of logs, a collector was designed to extract the pertinent data from logs and send the data to the centralized auditing database. PACS logs may be stored with different

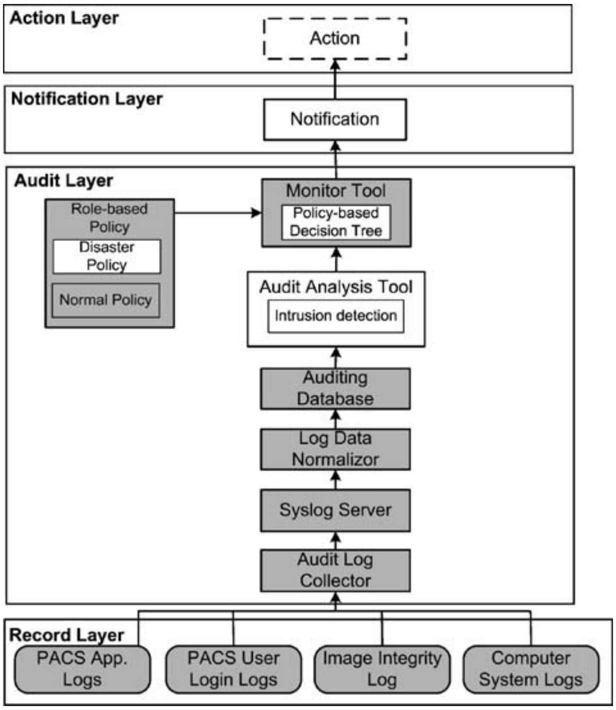


Fig. 1. The four-layer HIPAA-compliant architecture for auditing of medical images in PACS showing various components for each layer: (1) record layer, (2) audit layer, (3) notification layer, and (4) action layer. Shaded components represent already developed components of H-CAS.

formats, such as database tables or textual files. The collector should be designed to support the various types of logs.

System Log (Syslog) Server. The pertinent data extracted from PACS logs are distributed in different PACS components connected by digital networks. To store them in the centralized auditing database, a transmission mechanism is needed.

Currently, syslog[11] is a de facto standard for transport and storage of event notification messages in UNIX and Windows-based systems, network devices, and network applications. Syslog is a client–server mechanism. The clients can be configured to locally store event messages or directly send event messages to the server without local storage. Syslog uses user datagram protocol to transfer event messages. This feature can be utilized to reduce the overhead added to the image transmission in PACS caused by event message communication because PACS uses DICOM protocol and transmission control protocol. For this advantage, syslog technology was adopted as the architecture standard to transfer pertinent log data. The data are converted to syslog format by the syslog client in each PACS component. The client then sends the data to syslog server, which will forward the data to the log data normalizer.

Log Data Normalizer. The pertinent data extracted from PACS components might have different terminologies for the same object. For example, the name "film librarian" in the CT modality might be named as "clerk" in an MR modality. For this reason, a log data normalizer was designed to normalize the data into common terms and then add them to the auditing database. The current dictionary contains some of these object classifications and is designed to be scalable to support additional objects in the future.

Auditing Database. To generate HIPAA-compliant audit trails in a short time, a centralized database was designed to preserve all the obtained auditing data. The structure of database was designed based on the requirement of HIPAA-compliant audit trails, including who, when, where, what, how, and status. Patient information, such as name and id, and other relevant information are also included in the database. The advantages to use database technology to preserve the log data are as follows:

- No loss of historical logs. Because all the logs generated in PACS components are obtained and stored in the database everyday, there is no loss of log data when these logs are updated by PACS components.
- Centralized management of data access information. The image data access events for an individual patient usually happen in multiple PACS components. For example, an event that a CT image is generated in a CT modality and another event that the same CT image is retrieved to viewing workstation for clinical review are related to the same patient. But these two events were recorded in two different logs at two separate PACS components. This pertinent information will need to be extracted from these two components every time HIPAA-compliant audit trails of image access for this patient are desired. Therefore, a centralized database design enables us to quickly generate audit trails in one centralized location where data are stored.

The auditing database design contains three tables: event, patient, and study. The relationship among them is shown in Figure 2.

- The event table contains seven columns: Event_No, Event_Type, Event_Location, Event_Time, Event_Status, Event_User, and Patient_No. The Patient_No column is a link that connects to the Patient_No column in the second table, the patient table.
- The patient table contains five columns: Patient_No, Patient_Name, Patient_ID, Patient_Sex, and Patient_Age.
- The study table contains five columns: Study_No, Study_InstanceUID, Accession_Number, Modality, and Patient_No, which is a link between the study table and the patient table.

The relationship between the patient table and the event table is 0 or 1 to 0 or more because a single patient can be associated with multiple events that access the data of this patient. The same relationship is between the patient table and the study table because a single patient can contain multiple studies. Whenever an event occurs where a study is accessed in PACS, the auditing database records the information to these three tables accordingly.

Audit Analysis Tool. Most current PACS lack a mechanism to dynamically monitor the data flow, which results in PACS management mostly relying on the experience of PACS administrators. A monitoring tool that can automatically analyze the data to find abnormal patterns and make decisions

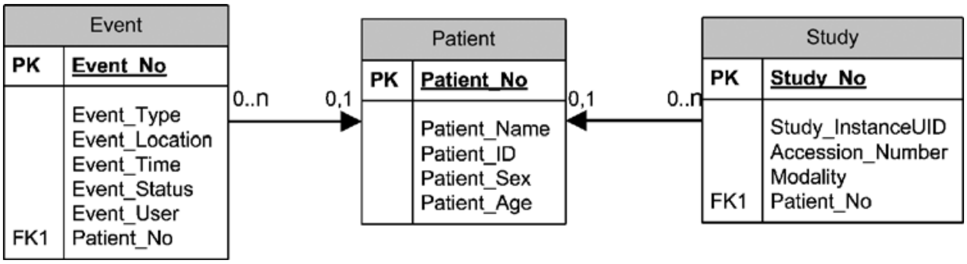


Fig. 2. E–R model of the auditing database in the audit layer of the HIPAA-compliant architecture. Three tables and their relationships are shown: (1) event table; (2) patient table; and (3) study table. PK: primary key. FK: foreign key.

on the patterns would make PACS management much easier. To develop such a tool, the information of data flow of PACS needs to be collected and analyzed in real time. With audit data collected in the auditing database, the HIPAA-compliant architecture can provide this ability using some data analysis techniques, such as intrusion detection technology.[12] Audit analysis tool is the component to perform such data analysis functions.

Monitor Tool. After the audit analysis tool finds abnormal patterns in the data flow of PACS, a monitor tool was designed to monitor the pattern and make decisions whether it is an unauthorized data access for the abnormal pattern based on the role-based policy. Any pattern that violates the policy would automatically cause a warning or alert result. For example, audit analysis tool discovers an abnormal pattern of image query/retrieve by a PACS user "A", belonging to the role of "Clerk", which was defined to have no image query/retrieve right in the policy. The monitor tool automatically makes a decision that this is an unauthorized image query/retrieve and gives a warning message.

Role-Based Policy. The role-based policy defines the roles for PACS users based on the roles they performed in the clinical environment, such as clerk, PACS manager, and radiologists, and the image access rights for each role. Two types of policies, normal policy and disaster policy, are defined for two different conditions. Normal policy is for daily operation, whereas disaster policy is defined for the emergency situations, such as earthquake, when normal policy can be bypassed.

Notification Layer

Notification layer consists of a notification component, which receives the warning or alert messages from the audit layer and notifies PACS end users of the unauthorized image data access and other abnormal activities.

Action Layer

Action layer is designed for PACS end users to take actions, such as access control, against the unauthorized image access and other abnormal activities. This four-layer architecture enables PACS to generate HIPAA-compliant audit trails of image data access for a specific patient on demand. Meanwhile, it can automatically monitor the data flow of PACS facilitating PACS management. With an open and extensible design, the architecture can also easily incorporate new data analysis and monitoring techniques and be extended to support future HIPAA requirements.

PRELIMINARY RESULTS AND DISCUSSION

A H-CAS has been developed for automatic monitoring of the data flow of PACS based on partial components of the audit layer in the architecture. The H-CAS and its graphic user interface (GUI) were installed in a LINIX machine. H-CAS currently includes such components as audit log collector, syslog server, auditing database, monitor tool, and role-based policy (normal policy). These components are represented by the shaded boxes in Figure 1. H-CAS can monitor the dynamic data flow of PACS. First, it collects pertinent auditing data from PACS application logs, PACS user login logs, and other computer system logs. It then stores the log data in the auditing database. Next, a comparison is made of the user name in every record in the auditing database and the user name in the policy table. If a match occurs, further comparison is made with the application name in the database record and the application name in the policy table. If any comparison fails, the H-CAS gives out a warning message of unauthorized image data access in its GUI. Otherwise, a normal message is given out. In addition, H-CAS can generate HIPAA-compliant audit trails of image data access for a specific patient.

Integration with PACS Simulator in a Laboratory Environment

To evaluate the impact of H-CAS in PACS, a laboratory-based PACS simulator[13,14] was implemented with the toolkit to simulate the data flow of clinical PACS. The simulator can simulate the complete data workflow of clinical PACS from patient registration to exam ordering and to image generation, image archive, and display. The clinical images used for simulation are replenished continuously through a clinical PACS connection but with the patient information in the DICOM header of the image removed. Figure 3 shows the integration of H-CAS with the PACS simulator for evaluation.

The log collector clients are installed in every component to receive event messages of each image data access activity generated by these components. Log messages generated are automatically collected and inputted into the H-CAS via the syslog server. The log data include what, when, and where images are accessed. In addition, user login logs and computer system logs are collected. All pertinent log information are collected and stored in the centralized auditing database. As with any PACS, log data can be

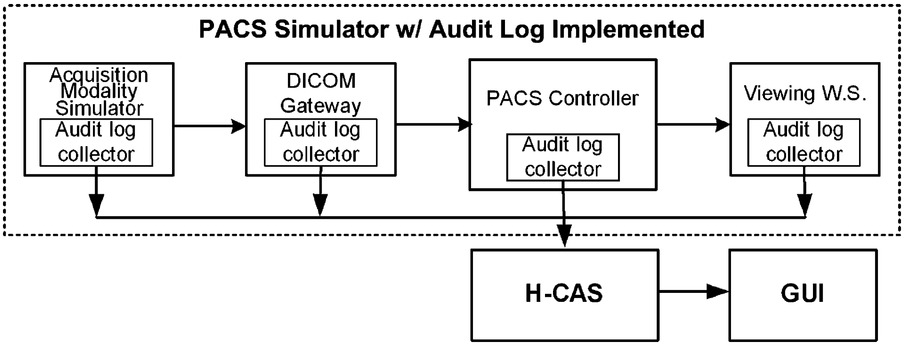


Fig. 3. H-CAS implemented with the PACS simulator in the laboratory environment showing log collector clients at each of the PACS simulator components.

stored in different formats. For example, the log data of the acquisition modality simulator (AMS) are stored in the Microsoft access database, whereas the log data of the PACS controller are stored in the Oracle database. The user login information and application access information from the computer system logs of the DICOM gateway and viewing workstations were also collected and can be from different operating systems such as Windows and UNIX. An interface is developed to extract each of the different log data and integrate it within the H-CAS SQL auditing database. All data logs are sent to the centralized H-CAS where analysis is performed. A user-friendly GUI application was developed in conjunction with H-CAS to understand the HIPAA audit trails concept by searching H-CAS for data access events on a specific patient or user. In addition, it also provides a set of features allowing users to configure the role-based policy and dynamically monitor the image data flow of the PACS simulator based on the policy.

Evaluation Methodology

Test Scenario Design

Test scenarios for laboratory evaluation of the H-CAS are anchored around creating a clinical simulation of the PACS workflow from exam generation to the retrieval of exams for clinical review. According to the perspective of clinical end users, two categories of test scenarios were designed for the evaluation.

- Category 1: Background scenarios
- Category 2: On-demand scenarios

Category 1 background scenarios are basically automatic storage functions, such as DICOM gateway sending images to PACS archive server, whereas category 2 on-demand scenarios are requests issued by end users, such as image query/retrieve at viewing workstations. To simulate clinical 24/7 image automatic storage, a loop process was designed to repeatedly send various types of modality images, such as CT, MR, CR, and ultrasound images, to DICOM gateway, which automatically forwards the images to PACS controller. One example of a category 2 test scenario is performing an on-demand query/retrieve from the viewing workstations.

Experiment Description

During the laboratory experiments, H-CAS was tested with five CT exams, five MR exams, and five CR exams. Table 1 tabulates the test data. Three scenarios were performed in the experi-

Table 1. Exams tested in the laboratory experiments

	CT exams	MR exams	CR exams	Total
No. of exams	5	5	5	15
Total images	186	288	11	485
Average no. of images per exam	37.2	57.6	2.2	
Data size (MB)	93.9	53.0	83.3	230.2

ments: (1) AMS simulates generation of an exam; (2) DICOM gateway automatically forwards the exams to the PACS controller; and (3) the viewing workstations query/retrieve the exams for clinical review. Scenarios 1 and 3 are in the active category 2, whereas scenario 2 is a passive category-1-type scenario. Most of the experiments were performed with all three scenarios in the order of 1, 2, and 3. Some experiments only included one or two scenarios performed in random order. The experiments were conducted over a 6-month period.

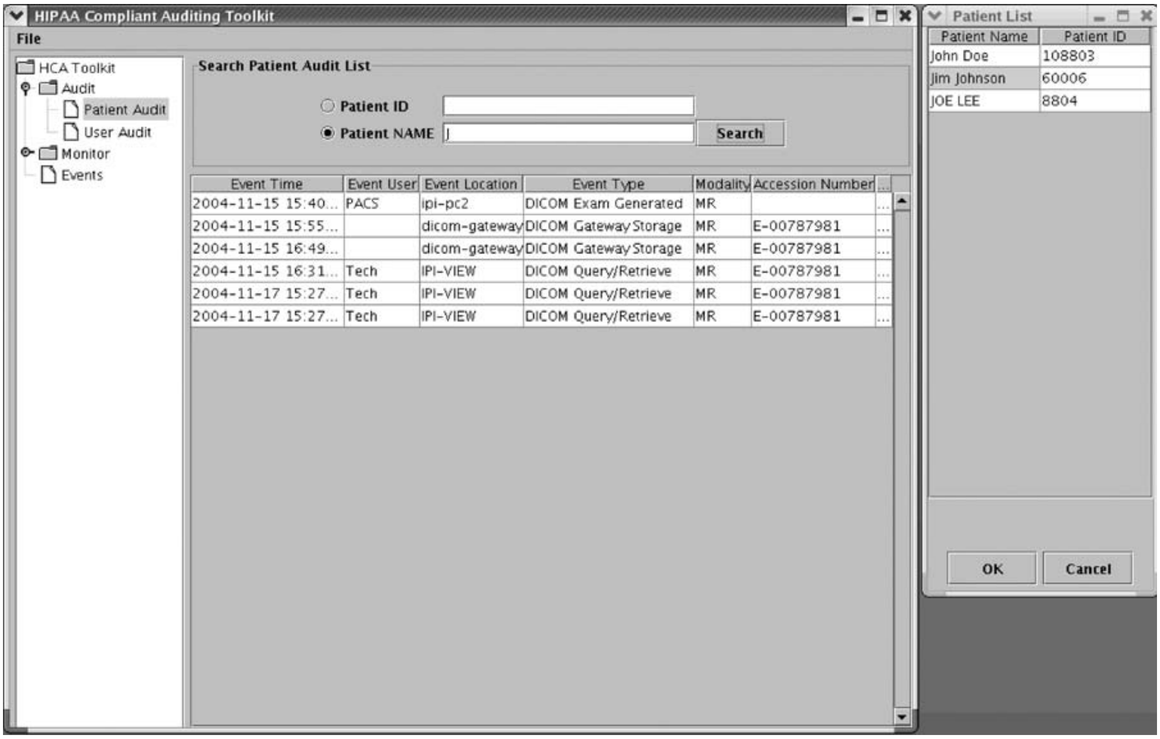
Examples of Results

Two examples of results are shown in Figures 4 and 5.

The first example shows H-CAS searching the auditing database to generate a HIPAA-compliant audit trail based on a given patient name "Jim Johnson." Two operations need to be performed to generate the audit trail for a patient. First, search the patient in the database with "Patient Name" or "Patient ID". In this case, "Patient Name" was chosen. The GUI supports wildcard searching; therefore, a "J" was typed in the search field to search the patient with the first name starting with "J". A list of patients matching the search criteria was returned in a new popup window. Next, the patient name "Jim Johnson" was selected. A HIPAA-compliant audit trail is then generated as seen in Figure 4. As shown in Figure 4, line 1 shows an example of scenario 1, where the user "PACS" performed a MR exam generation at the modality simulator "ipi-pc2" on "2004-11-15 15:40." Line 2 shows an example of scenario 2, where the DICOM gateway stored the exam in the PACS controller. Line 4 shows an example of scenario 3, where the user "Tech" performed a DICOM query/retrieve of the exam at the viewing workstation "IPI-VIEW."

Figure 5 shows a second example of results of dynamic monitoring of the image data flow of the PACS simulator. Two types of results, "Normal" or "Warning," were given for each data access event based on the role-based policy.

The role-based policy is defined and configured prior to using the monitoring function and is based



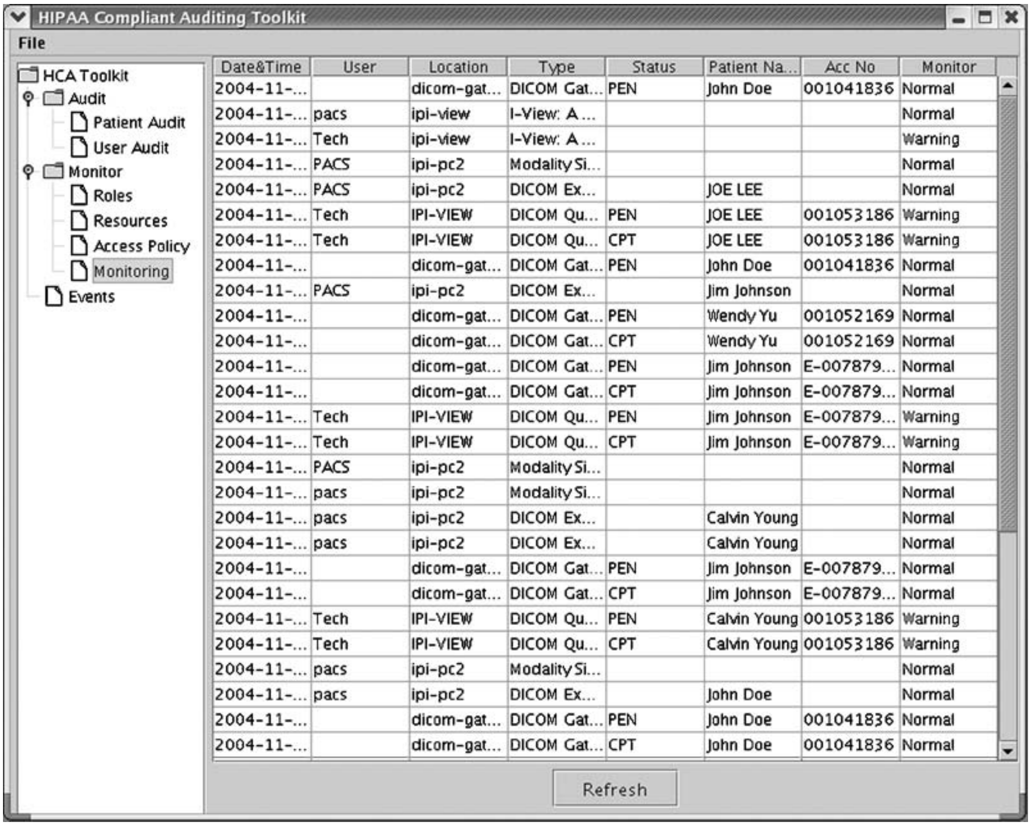
○ Patient ID

● Patient NAME J

Event Time	Event User	Event Location	Event Type	Modality	Accession Number	...
2004-11-15 15:40...	PACS	ipi-pc2	DICOM Exam Generated	MR		...
2004-11-15 15:55...		dicom-gateway	DICOM Gateway Storage	MR	E-00787981	...
2004-11-15 16:49...		dicom-gateway	DICOM Gateway Storage	MR	E-00787981	...
2004-11-15 16:31...	Tech	IPI-VIEW	DICOM Query/Retrieve	MR	E-00787981	...
2004-11-17 15:27...	Tech	IPI-VIEW	DICOM Query/Retrieve	MR	E-00787981	...
2004-11-17 15:27...	Tech	IPI-VIEW	DICOM Query/Retrieve	MR	E-00787981	...

Patient Name	Patient ID
John Doe	108803
Jim Johnson	60006
JOE LEE	8804

Fig. 4. Example of generating the HIPAA-compliant audit trails for a patient "Jim Johnson", who is shaded in the list on the right.



Date&Time	User	Location	Type	Status	Patient Na...	Acc No	Monitor
2004-11-...		dicom-gat...	DICOM Gat...	PEN	John Doe	001041836	Normal
2004-11-...	pacs	ipi-view	I-View: A ...				Normal
2004-11-...	Tech	ipi-view	I-View: A ...				Warning
2004-11-...	PACS	ipi-pc2	Modality Si...				Normal
2004-11-...	PACS	ipi-pc2	DICOM Ex...		JOE LEE		Normal
2004-11-...	Tech	IPI-VIEW	DICOM Qu...	PEN	JOE LEE	001053186	Warning
2004-11-...	Tech	IPI-VIEW	DICOM Qu...	CPT	JOE LEE	001053186	Warning
2004-11-...		dicom-gat...	DICOM Gat...	PEN	John Doe	001041836	Normal
2004-11-...	PACS	ipi-pc2	DICOM Ex...		Jim Johnson		Normal
2004-11-...		dicom-gat...	DICOM Gat...	PEN	Wendy Yu	001052169	Normal
2004-11-...		dicom-gat...	DICOM Gat...	CPT	Wendy Yu	001052169	Normal
2004-11-...		dicom-gat...	DICOM Gat...	PEN	Jim Johnson	E-007879...	Normal
2004-11-...		dicom-gat...	DICOM Gat...	CPT	Jim Johnson	E-007879...	Normal
2004-11-...	Tech	IPI-VIEW	DICOM Qu...	PEN	Jim Johnson	E-007879...	Warning
2004-11-...	Tech	IPI-VIEW	DICOM Qu...	CPT	Jim Johnson	E-007879...	Warning
2004-11-...	PACS	ipi-pc2	Modality Si...				Normal
2004-11-...	pacs	ipi-pc2	Modality Si...				Normal
2004-11-...	pacs	ipi-pc2	DICOM Ex...		Calvin Young		Normal
2004-11-...	pacs	ipi-pc2	DICOM Ex...		Calvin Young		Normal
2004-11-...		dicom-gat...	DICOM Gat...	PEN	Jim Johnson	E-007879...	Normal
2004-11-...		dicom-gat...	DICOM Gat...	CPT	Jim Johnson	E-007879...	Normal
2004-11-...	Tech	IPI-VIEW	DICOM Qu...	PEN	Calvin Young	001053186	Warning
2004-11-...	Tech	IPI-VIEW	DICOM Qu...	CPT	Calvin Young	001053186	Warning
2004-11-...	pacs	ipi-pc2	Modality Si...				Normal
2004-11-...	pacs	ipi-pc2	DICOM Ex...		John Doe		Normal
2004-11-...		dicom-gat...	DICOM Gat...	PEN	John Doe	001041836	Normal
2004-11-...		dicom-gat...	DICOM Gat...	CPT	John Doe	001041836	Normal

Fig. 5. Example of the dynamic monitoring of the image data flow of the PACS simulator based on the role-based policy.

on each institution's own access policy. The GUI of the toolkit provides users an interface called "Access Policy" to add, modify, or delete the role-based policy. Table 2 lists an example policy used in the experiments. In this case, user "Pacs" was defined as "PACS administrator" with the right to perform all the applications in the PACS, whereas user "Tech" was defined as "Technician" with the right to perform the exam generation in the modality simulator. For this particular experiment, it would be a violation of the policy if user "Tech" performed the exam query/retrieve at the viewing workstation designated only for radiologists. The "Warning" in the sixth row in Figure 5 indicates that a violation occurred when user "Tech" retrieved an exam with patient name "JOE LEE." Because it is sometimes necessary at a particular institution for a user "Tech" to perform query/retrieve at the radiologist's workstation, the role-based policy is flexible to accommodate each institution's own particular access policy to allow such a function for the user "Tech."

Table 2. Example of a role-based policy table where "all" indicates access to all components within the PACS simulator

User name	Role name	Resources
Pacs	PACS administrator	All
Clerk	Film clerk	Viewing workstation
Rad	Radiologist	All
Tech	Technician	AMS

The events shown in Figure 5 are all examples of the three scenarios developed. For example, line 1 is an example of scenario 2, line 2 is an example of scenario 3, and line 5 is an example of scenario 1.

Discussion

Three testing scenarios have been developed for the laboratory evaluation of the H-CAS integrated with the PACS simulator. For the evaluation of the function of generating HIPAA-compliant audit trails, the results from active category 2, scenarios 1 and 3, are more important than the ones from passive category 1, scenario 2, because of the increased likelihood of HIPAA security violation when humans are involved. On the other hand, all three scenarios are almost equally important for the dynamic monitoring because any unreported event may indicate a failure in the image data flow. The three scenarios developed are the most typical data flow in PACS and can represent most of the processes in PACS. However, one important process, image storage or archive, was not included in the testing. The image storage is very important to both evaluated functions of H-CAS because the stored image data could be compromised without detection. By integrating image integrity logs at the record layer, this image storage issue can be solved.[10]

To give an approximate estimate of the number of records that can be collected by the H-CAS, we assume that a community-sized hospital generates about 150,000 imaging studies per year. In addition, assuming each exam is accessed an average of five times (once at the modality, once at the PACS server, once in the diagnostic reading workstation, once in the reviewing workstation for the referring physician, and once as a historical review), there will be a total of 750,000 records in the event table, 150,000 records in the study table, and, assuming an average of two exams per patient, 75,000 records in the patient table yearly.

The current H-CAS functionality of dynamic monitoring is still limited. A more sophisticated monitoring system can be developed using intrusion detection technology and policy-based decision tree approaches. The role-based policy can also be improved to a multiple-factor-based policy instead of a single role-based one. In addition, robust and well-formatted logs generated by PACS and imaging modalities would greatly enhance H-CAS functions.

CONCLUSION

The advent of HIPAA greatly impacts medical imaging systems, such as PACS, and even the entire health information systems. To be HIPAA-compliant, every medical imaging system must satisfy the HIPAA requirement of audit trails.

In this research, we presented a HIPAA-compliant architecture for auditing medical images in PACS. The architecture enables PACS to generate HIPAA-compliant audit trails of image data access for a specific patient on demand. It also enables PACS to automatically monitor the image flow in the system, including detection of unauthorized usages of image data and other abnormal activities. As an initial first step, a H-CAS has been developed partially based on the audit and record layers of this architecture. H-CAS can automatically monitor the image flow in PACS and the ability to generate HIPAA-compliant audit trails. A PACS simulator was integrated with the H-CAS for laboratory evaluation. An evaluation methodology was developed, and test scenarios were designed accordingly to perform the evaluation. Evaluation is currently ongoing with promising initial data results.

ACKNOWLEDGMENT

This research is partially supported by NIH Grant no. R01-LM06270.

REFERENCES

1. HIPAA, http://www.cms.hhs.gov/hipaa/hipaa2/general/background/pl104191.asp

2. HIPAA, http://www.rx2000.org/KnowledgeCenter/hipaa/hipfaq.htm

3. HIPAA Security Standard, 2003, http://www.cms.hhs.gov/hipaa/hipaa2/regulations/security/03-3877.pdf

4. Cao F, Huang HK, Zhou XQ: Medical image security in a HIPAA mandated PACS environment. Comput Med Imaging Graph 27(2–3):185–196, 2003.

5. HIPAA audit, http://www.datamirror.com/products/liveaudit/

6. HIPAA audit tool, http://www.axolotl.com/press/20021113/

7. HIPAA audit tool, http://www.peacefulpackers.com/it_solutions/hs2.htm

8. Huang HK: PACS: Basic Principles and Applications. Wiley-Liss, p. 291, 1999

9. Zhou Z: Lossless Digital Signature Embedding for Medical Image Integrity Assurance, Ph.D. Dissertation, Chap. 5–6, July 2005

10. Zhou Z, Huang HK, Liu B: Digital signature embedding for medical image integrity in a data grid off-site backup archive. SPIE Med Imaging 6:306–317, 2005

11. Syslog, http://www.loriotpro.com/Products/SyslogCollector/SyslogDataSheet_ENv3.php

12. White GB, Fisch EA, Pooch UW: Computer System and Network Security. Boca Raton, FL: CRC Press, 1996

13. Law MYY, Zhou Z: New direction in PACS education and training. Comput Med Imaging Graph 27:147–156, 2003

14. Zhou Z, Huang HK, Cao F, Liu BJ, Zhang J, Mogel GT: Educational RIS/PACS simulator. SPIE Med Imaging 4:139–147, 2003