A Comparative Analysis of Chain-Based Access Control and Role-Based Access Control in the Healthcare Domain

Esraa Omran, Gulf University for Science & Technology, Kuwait City, Kuwait

Tyrone Grandison, Proficiency Labs, Ashland, OR, USA

David Nelson, Faculty of Applied Sciences, University of Sunderland, Sunderland, UK

Albert Bokma, Avedas Information Management, Karlsruhe, Germany

ABSTRACT

The importance of electronic healthcare has caused numerous changes in both substantive and procedural aspects of healthcare processes. These changes have produced new challenges for patient privacy and information secrecy. Traditional privacy policies cannot respond to rapidly increased privacy needs of patients in electronic healthcare. Technically enforceable privacy policies are needed in order to protect patient privacy in modern healthcare with its cross-organizational information sharing and decision making. This paper proposes a personal information flow model that proposes a limited number of acts on this type of information. Ontology-classified chains of these acts can be used instead of the "intended business purposes" in the context of privacy access control. This enables the seamless integration of security and privacy into existing healthcare applications and their supporting infrastructures. In this paper, the authors present their idea of a Chain-Based Access Control (ChBAC) mechanism and provide a comparative analysis of it to Role-Based Access Control (RBAC). The evaluation is grounded in the healthcare domain and examines a range of typical access scenarios and approaches.

Keywords: *Access Control, Chains, Healthcare Information Systems, Information Protection, Personal Information Management, Role Based Access Control*

INTRODUCTION

Healthcare environments are a complex web of medical professionals and systems (both electronic and non-electronic). As the data being used, stored and transmitted in these environments are valuable and may have several negative outcomes attached to them, the privacy and security of this data is of utmost importance to patients, healthcare staff and the supporting Information Technology (IT) specialists. In this paper, we focus on the fundamental privacy and security mechanisms that are the foundation of healthcare IT systems; with an emphasis on comparing their use in real situations.

DOI: 10.4018/jisp.2013070103

In this paper, we deal with a complex systems scenario from the healthcare domain. Our work is based on work performed at the International Clinic in Kuwait (2011), which is distributed over several locations and serves a patient community in excess of 100,000. Consequently, there are a large number of professionals who are involved in a patient's care and who need access to patient records. The case is complex for a number of different reasons. There is an evolving set of patients and their records. There are a large number of different types of healthcare professionals, ancillary staff and management staff who deal with patients on a daily basis, and need appropriate access to records to perform their job. Finally there is a need to selectively share information with other healthcare organisations, third party service providers and insurance companies. Health records are particularly complex due to the sensitivity of the records and the need to provide maximum protection (Dick et al., 1997), while allowing access to that data by a large number of users who may require access to specific aspects of the records for varying specific purposes. This field is also heavily regulated; due to the sensitivity of the records and privacy implications. Many countries (University of Alberta, 2005; Webster, 1990) have healthcare-specific data and privacy protection legislation that prescribes the need for record keeping and restricting access to these records for only legitimate purposes.

We contend that this complexity causes several problems, which include:

1. It is difficult for database administrators to correctly define access privileges; giving rise to errors;
2. It is more difficult for another database administrator to subsequently maintain access restrictions;
3. There is an increased opportunity for unintended side-effects when complex privileges are interpreted by the system;
4. Solutions take more steps to compute and cause runtime inefficiencies when privileges have to be checked for a given request.

It would be desirable to have a simpler solution that is easier to configure, maintain and reliably execute. Our proposition is that simple controls and solutions scale and perform better as systems become more complex. This has proven true in other domains, e.g. massively-parallel processing with HADOOP (Borthakur, 2007), and it is hoped that it will be true for security and privacy mechanisms. Currently, the most widely adopted approach to access management, which is supported by the major database systems providers, is Role Based Access Control (RBAC). We purport that RBAC can be complicated to apply in healthcare scenarios and that a simpler approach is needed.

Based on an original concept presented by Al-Fedaghi (2007) we have operationalised the concept of the Chain-Based Access Control (ChBAC) and carried out an evaluation. To this end, we conduct experiments in a complex healthcare environment in order to compare ChBAC with RBAC.

Generally, the application of an effective approach has two phases that must be supported, namely the design phase where the system programmer needs to implement access policies and the runtime phase, where attempts to access data need to be assessed and either be granted or denied. Consequently, any useful method needs to be both easy to apply correctly during the design phase and efficient during runtime. Our evaluation in this paper concentrates on the design phase and we intend to report on the runtime performance in a future publication. Before proceeding with the experiments we will firstly discuss related work, and then outline the Chain-Based Access Control model.

RELATED WORK

We recognize that there are a variety of techniques that have been proposed including: RBAC; Enterprise Privacy Authorization Language (EPAL) (EPAL, 2009); the Platform for Privacy Preferences (P3P) (W3C, 2009); Hippocratic Database (HDB) (Agrawal et al., 2002); and Platform Privacy Preference (P3P) (W3C, 2009). We classify the attempts to preserve

privacy into three groups: the first group being the privacy laws, specifications and languages, such as EPAL. The second group tries to preserve privacy in the application level, such as RBAC (Sandhu, 1998) and Task Based Access Control (TBAC) (Thomas and Sandhu, 1993). The third group is trying to save the privacy in the data level, such as Hippocratic database technology (Agrawal et al., 2002).

P3P (W3C, 2009) is a machine-readable vocabulary and syntax for expressing a website's personal data and information management policy. P3P policies present a snapshot summary of how the site collects, handles and uses personal information about its visitors. P3P-enabled web browsers and other P3P applications will read and understand this snapshot information automatically, compare it to the web user's set of privacy preferences, and alert the user when these preferences do not match the practices of the website. However, setting rules does not guarantee their enforcement. IBM proposes the Enterprise Privacy Authorization Language (EPAL) (EPAL, 2003) to support organisations in keeping their privacy promises. EPAL provides enterprises with a way to formalize the exact privacy policy that shall be enforced within the enterprise. An EPAL policy consists of a vocabulary and a rule set. The vocabulary defines the scope of the policy. Rules are statements that specify which actions a user can or cannot perform on a certain object and for which purpose. When data are requested, the privacy management enforcement monitors ensure that only data accesses complying with the privacy policy are allowed.

In RBAC, access decisions are based on the roles that individual users have as part of an organization. Users are given assigned roles, such as doctor, nurse, teller, or manager. Access rights are grouped by role title, and the use of resources is restricted to individuals who are authorised to assume the associated role. For example, within a healthcare system, the role of doctor can include functions to perform a diagnosis, prescribe medication, and order x-ray tests; whereas the role of researcher can be limited to gathering anonymous clinical information for research purposes. There are a few areas in which the RBAC model may be improved. First, differentiating roles in different contexts often proves to be difficult. This can result in large quantities of role definitions, in some cases producing more roles than users. Second, RBAC remains somewhat coarse-grained while modern requirements are increasingly fine-grained. Finally, while the initial RBAC model was based on permissions only, the need to explicitly specify denial of access has become important. These factors have resulted in multiple variations of the RBAC model, including Task Based Access Control (TBAC) (Thomas & Sandhu, 1993). TBAC is well suited for distributed computing and information processing activities with multiple points of access, control, and decision making such as those found in workflow and distributed process and transaction management systems. TBAC varies from traditional access controls and security models in many respects (Thomas & Sandhu, 1993). Instead of having a system-centric view of security, TBAC approaches security modelling and enforcement at the application and enterprise level, which makes it more desirable in real world enterprises.

Agrawal et al. (2002) argue that future database systems must include responsibility for the privacy of data they manage as a founding tenet – a Hippocratic Database (HDB). The prominent advantage of such an architecture is that it uses privacy metadata consisting of privacy policies and privacy authorizations, stored in privacy-policy tables and privacy-authorization tables, to perform seamless enforcement of rules at the data level. According to Grandison et al (2008), determining purpose information is difficult and there is work to be done with regards to the retention and safety components of the technology suite.

CHAIN-BASED ACCESS CONTROL

Chain-Based Access Control (ChBAC) is based on the notion of a *chain of acts* (Fedaghi, 2007). Fedaghi (2007) presented the idea by changing the principle of data access control from

purposes to *chains of limited acts*. He purported that the management of attributes and users' purposes is a complex issue. To simplify the mapping process, users are assigned to roles, and access purpose permissions are granted to roles associated with tasks or functionalities, not directly to individual users.

Unlike RBAC, ChBAC doesn't need to have long, complicated policies for each group of roles (Fedaghi, 2007). Instead, a set of seven limited acts: Creating, Processing, Disclosing, Storing, Collecting, Using, and Mining (as shown in Figure 2), are distributed amongst the different groups of roles. These acts define the policy and purpose for which a particular group of roles is accessing the database and at the same time it includes the actions that the user can apply on the database.

As shown in Figure 1, data usage can be divided into four phases, namely: creation; collection; processing; and disclosure of personal information. Each phase can be associated with a number of allowed acts. Personal information can be created by proprietors (i.e. the data subject), by non-proprietors (i.e. any data recipient different from the data subject), or can be deduced from existing information (e.g. using data mining). Created information can either be used (e.g. for decision making), stored, or disclosed. In addition, information can enter into the processing and disclosing phases. The processing of personal information involves storing, using, and mining personal information. The disclosure phase involves releasing personal information to other actors.

Fedaghi (2007) argues that each role can be translated into a chain of acts on personal information, such as in Figure 1. He further proposes that any piece of personal information only requires a limited set of acts that can be operated on it. He claims that those limited acts could be used to design a more robust data access control mechanism that could safeguard personal information privacy. So, instead of a huge policy tables, there is instead a small, manageable set of limited acts.

Figure 2 represents the personal information flow model of a typical healthcare scenario. Here, the proprietor of personal information is the patient, whereas the non-proprietors are doctors, nurses, receptionists and insurance companies. Every actor involved in data processing is represented along with the acts that he or she can perform. For instance, nurses can collect, store, process and disclose patient information. The arrows between acts represent the allowed chains of acts. For instance, the information disclosed by the patient can be collected by the nurse, who in turn can either store it or process it. If the nurse stores them, she can either collect new information or process it. In Omran et al (2010) we have drawn the basic lines of the specifications of the Chain method construction.

Figure 3 shows that the result of the transformation process would be of the form:

<User ID>…. <User ID>
<Chain ID>…..<Chain ID>
…….

We need to specify the user and his associated chain in order to know which chain of acts is to be assigned to which user in order to connect them both to the artifacts. In this case, artifacts refer to data in the database. In addition, specifying the chains to the role will also clarify the actions/functions that this user can perform.

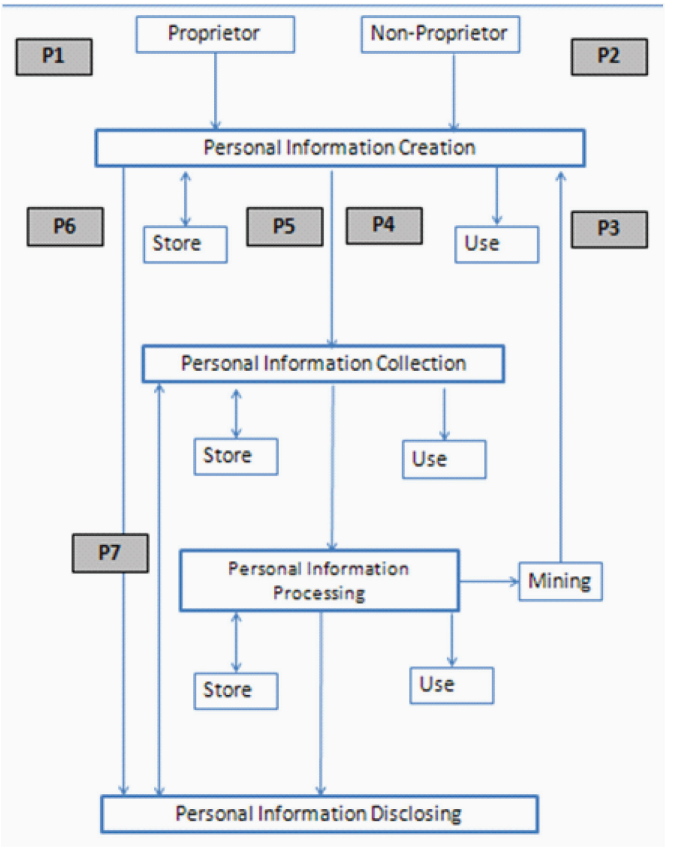
Compared to the same construction for the RBAC

The resulting policy statement is of the form presented in Box 1.

In Figure 4, we need to specify each user with his role and then specify the compound constraints that need to be given to each role. There are two types of constraint. The first is the user to which access to tables in the database is to be specified. The second is the type of actions to be applied on this table according to each role.

Given this background, we now discuss the scenario that we based our experiments on.

Figure 1. Personal Information flow model. Source: Fedaghi (2007). The flow model shows the main acts of the Chain method: Creating, mining, storing, using, creating and collecting.



SCENARIOS

As previously stated, healthcare provision in a hospital environment demonstrates a considerable amount of complexity in terms of processes and actors associated with treating individual patients, during which healthcare information needs to be accessed and shared. Patients are seen by consultants, need to be assessed and examined, and then a plan for treatment needs to be devised. This can involve several separate professionals before the patient is treated, discharged and eventually billed.

This section introduces typical scenarios for healthcare provision in a hospital environment. Table 1 shows a list of scenarios that routinely occur in a typical hospital and have been carefully abstracted from processes at the International Clinic Kuwait and broadly include:

- New patient registration and appointment bookings (scenario 1.1 in Table 1) ;
- Routine consultations with doctors and nurses (scenario 2.1 in Table 1);
- Emergency admissions (scenario 1.4 in Table 1);
- Billing for services (scenario 1.5 in Table 1);
- Managing patients (scenario 1.6 in Table 1);
- Outgoing and incoming referrals (scenarios 2.2 and 2.3 in Table 1);
- Issuing and dispensing prescriptions (scenario 2.4 in Table 1);

Figure 2. Architecture of information flow. Source: Fedaghi (2007). The Architecture shows an example for the different acts that are given to the different users in the healthcare and the way that they interact with each other.

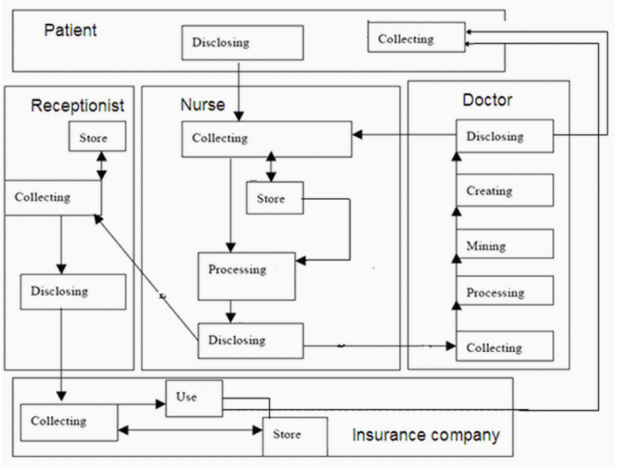
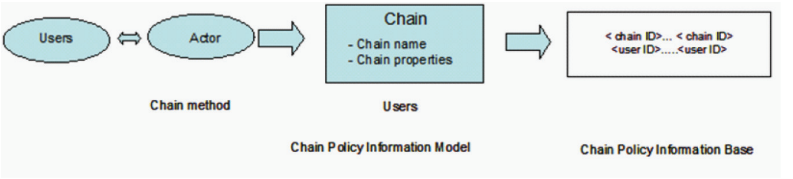


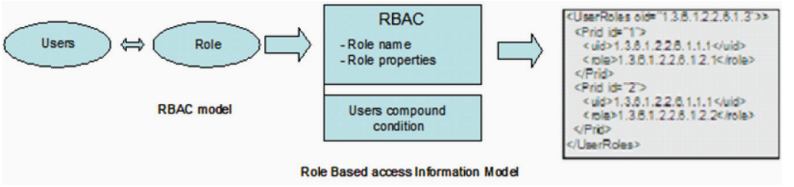
Figure 3. ChBAC system specifications. The illustration shows that ChBAC has two elements: Chain ID and User ID. Which makes the method application easy in semantic languages such as OWL and RDF.



Box 1.

```
. .
<RoleName>rdfs:subclassOfrbac:Role.
<ActiveRoleName>rdfs:subclassOfrbac:ActiveRole;rdfs:subclassOf<RoleName>.
<RoleName>rbac:activeForm<ActiveRoleName>
```

Figure 4. RBAC system specifications. The illustration shows that RBAC has elements: Role name, Role ID, Role active name, username and user ID. Which makes the method application complicated in semantic languages such as OWL and RDF.



- Radiology and laboratory referrals (scenario 6.1 in Table 1).

Some of these processes from our study include:

- **Patient Registration:** This is represented by scenario (1.1) in Table 1. In this scenario, one is registering patients for the first time and taking basic details takes place. The database entity is the patient record.

When a patient visits the hospital for the first time, he needs to provide basic details to the receptionist and fill in a form. In addition, other information may be required. For instance, if the patient has health insurance, he has to present a valid insurance card upon registration. When the necessary documentation has been presented, the receptionist creates a record for the patient in the hospital database system. Information about the patient (e.g. name, age, gender, disability, civil ID number, phone number and address) is entered into his file.

To apply this in ChBAC: the receptionist collects (act1) the patient's personal information (e.g. name, birth date, etc.). Then, the receptionist stores (act2) his personal information. This results in a requirement for two SQL statements (create table and insert data), one table requirement (i.e. the table contains the chain of acts, user and data entities) and two constraints (as the database administrators assumes that the acts are replacements for the policies in the RBAC) as shown in Table 3.

Whereas, to apply this in RBAC: four SQL statements are needed (i.e. create table for role – Table 1, create table of patients – Table 2, insert data in Table 1, insert data in Table 2). Two tables are required (creating patient table and creating role table). Three constraints statements are required (two for describing table privileges and at least one for describing the action to be done and given for this role and how to be applied on the required data). Note that there is no standard way in writing this in RBAC. This is one of the limitations of RBAC as it depends on database administrator skills. These results are shown in Table 2.

- **Routine Specialist Consultation:** This is represented by scenario (2.1) in Table 1. During the patient referral, the physician needs to review the notes he (or another physician) has made on the patient's health condition. He also needs to write his new notes as well as any required prescriptions. In addition, he may need to order x-ray images and analyze images sent by the lab. He can also access information from the registration file or from the nursing database (e.g. temperature, weight, etc.).

The full set of scenarios is presented in Table 1 and forms the basis of our experimental results.

EXPERIMENTAL RESULTS

In the previous section, we presented a number of scenarios that represent a typical range for healthcare professionals (users). These scenarios are based on our case study of the International Clinic in Kuwait. The objective of the evaluation was to compare the complexity of the process of configuring access permissions using the ChBAC method versus using RBAC.

The tests were carried out by three database administrators from the American University of Kuwait and one database administrator from the Kuwaiti International Clinic. The respondents were asked to implement the required restrictions using the ChBAC and the RBAC methods and to record the number of tables they had to create, as well as the number of SQL statements required and the number of constraints. The reason behind choosing these measures, which may be overlapping to some degree, is to gain some insight into the typical complexity of implementing them from the point of view of the database administrator, as well as the results produced in the database that will affect the complexity with assessing access requests when users try to access the database.

Table 1. List of scenarios in healthcare

Number	Possible scenarios for group of users	Name of scenario	Description	database entities	Data access privileges
1.1	**Receptionist/ Administrator**	New patient registration	registering patients for first time, taking basic details	patient record	Creation of patient record but read-edit only for demographic part
1.2		Booking appointments	patient booking for his next appointment	Patient record Appointments	read-edit access to demographic part and no access to other parts read-edit-creation and deletion access to appointment records
1.3		Visit for appointment	patient arriving to see doctor with existing appointment	Patient record Appointments	read-write access to demographic part and no access to other parts read-write-creation and deletion access to appointment records
1.4		Emergency case	patient arriving in emergency case	Patient record A&E Waiting List	read-write access to demographic part and no access to other parts read-write-creation access to appointment records
1.5		Billing	Preparing and managing bills with insurance company	Patient record Billing records	read-access to demographic part and no access to other parts read-edit-creation and deletion access to billing records
1.6		Managing Patients	Managing patients for actions required by healthcare professionals	Patient record Referral records Appointments All Waiting Lists	Read access to demographic part and pharmacy records, but no access to other parts read access to referral records/doctors' letters read-write-creation and deletion access to appointment records read-write-creation and deletion access to waiting lists records
2.1	**Doctors/ Consultants**	Routine Patient Consultation	Seeing patients who have appointment or are on lists to be seen	Patient record Referral records Appointments	Read-write access to full patient record read access to referral records/doctors' letters read-write access to appointment records
2.2		Outgoing referral	Patient to be referred to consultant/nurse or radiology	Patient record Referral records Appointments All Waiting Lists	Read-write access to full patient record Read-write access to referral records/doctors' letters Read-write access to appointment records Read-write access to waiting lists records
2.3		Ìncoming referral	Patient has been referred to doctor by other healthcare professionals	Patient record Referral records Appointments All Waiting Lists	Read access to demographic and prescriptions part of patient record read access to referral records/doctors' letters read access to appointment records read access to waiting lists records
2.4		Issuing Prescriptions	Prescription to be issued to patient	Patient record Prescription Record	Read-write access to full patient record Read-write access to prescriptions
2.5		Emergency case	Patient coming in emergency case without appointment	Patient record A&E Waiting List	Read-write access to full patient record Read-write access to waiting list

continued on following page

Table 1. Continued

Number	Possible scenarios for group of users	Name of scenario	Description	database entities	Data access privileges
2.6		Waiting list consultation	Patient without appointment but not in emergency case	Patient record All Waiting Lists	Read-write access to full patient record read-write access to waiting lists records
3.1	**Nurse**	Nurse Consultation	Patient initiated service requests with appointment	Patient record Appointments	read-edit access to demographic and nursing part and no access to other parts read-edit-creation and deletion access to appointment records
3.2		Incoming Referral	Deal with patient according to the doctor's instructions	Patient record Referral records Appointments All Waiting Lists	read-edit access to demographic and nursing part and no access to other parts Read access to referral records/doctors' letters read-edit-creation and deletion access to appointment records read access to waiting lists records
3.3		Emergency Assessment	Patient coming in emergency case without appointment	Patient record A&E Waiting List	Read-write access to full patient record without clinical record read-write access to waiting lists records
4.1	**Manger and Senior Administrator**	Compliance auditing	A patient is complaining about sensitive information being disclosed. And he asked the hospital to know who is behind this disclosing.	Patient record Referral records Appointments All Waiting Lists Data Access Logs	read-edit access to demographic part and no access to other parts Read-edit access to referral records read-edit-creation and deletion access to appointment records read-write access to waiting lists records Read access to data access logs (where available)
4.2		Managing Healthcare	Managing patients and healthcare provision	Patient record Referral records Appointments All Waiting Lists User Accounts	Read access to demographic part but no access to other parts read access to referral records but not doctors' letters read-write-creation and deletion access to appointment records read-write-creation and deletion access to waiting lists records read-write-creation and deletion access to user accounts
5.1	**Insurance company**	Billing	Receiving bills for treatment of patient	Billing records	read access to billing records
6.1	**Radiology lab**	Radiology referral	Patient being referred to radiology by doctor	Referral records Appointments All Waiting Lists	Read-write access to referral records/doctors' letters Read-write access to appointment records Read-write access to waiting lists records
7.1	**Pharmacist**	Dispensing Prescriptions	Dispensing Medication according to Doctor's instructions	Patient record Prescription Record	Read access to basic details of name address and DOB Read-write access to prescriptions
8.1	**Laboratory**	Laboratory referral	Patient being referred to laboratory for blood analysis	Patient record Referral records	Read access to basic information (name, address, DOB) Read access to referral records/doctors' letters

Table 2. Sample of detailed scenario for RBAC

Scenario	Name of scenario	Number of steps-Number of SQL commands	Number of tables	Number of Constraints	Constraints
1.1	New patient registration	4 1-Create table patient; 2-Create table Role-Privilege for administrators; 3-Insert data; 4-Insert data;	2 1-Patient, 2-Role-Privilege for administrators;	3	For each table constraints for: privilege and describing action on that table
1.2	Booking appointments	6 1-Create table patient; 2-Create table Role-Privilege for administrators; 3-Create table appointments; 4-Insert data; 5-Insert data; 6-Insert data;	3 1-Patient, 2-Role-Privilege for-administrators, 3-Appointments	7 At least	For each table constraints for: privilege and describing action on that table
1.3	Visit for appointment	6 1-Create table patient; 2-Create table Role-Privilege for administrators; 3-Create table appointments; 4-Insert data; 5-Insert data; 6-Insert data;	3 1-Patient, 2-Role-Privilege for-administrators, 3-Appointments	6 At least	For each table constraints for: privilege and describing action on that table
1.4	Emergency case	6 1-Create table patient; 2-Create table Role-Privilege for administrators; 3-Create table A&E; 4-Insert data; 5-Insert data; 6-Insert data;	3 1-Patient, 2-Role-Privilege for-administrators, 3-A&E	6 At least	For each table constraints for: privilege and describing action on that table

The respondents set up two complete and working designs: one for the RBAC method and the other for the ChBAC method. A sample of this work is represented in Tables 2 and 3. The four database administrators produced their results in a laboratory in the American University of Kuwait, in the IT department. The experimental setup consisted of four computer

Table 3. Sample of detailed scenario for ChBAC

Scenario	Name of scenario	Number of steps-Number of SQL commands	Number of tables	Number of Constraints	Constraints
1.1	New patient registration	2 1-Create table patient; 2-Insert data;	1 Patient	2 As in the case of the chain the constraints are the same as the chain	Create, Store (As the privilege and action on data are specified by the act of the chain)
1.2	Booking appointments	4 1-Create table patient; 2-Create table appointments; 3-Insert data; 4-Insert data;	2 1-Patient, 2-Appointments	4 For each table two constraints	Create, Store
1.3	Visit for appointment	4 1-Create table patient; 2-Create table appointments; 3-Insert data; 4-Insert data;	2 1-Patient, 2-Appointments	4 For each table two constraints	Create, Store
1.4	Emergency case	4 1-Create table patient; 2-Create table A&E; 3-Insert data; 4-Insert data;	2 1-Patient, 2-A&E	4 For each table two constraints	Create, Store

units, each with 1066MHz Quad-core processors and 16 GB of RAM. They used SAN Storage with 500 Gb database storage (mirrored with RAID5) with ORACLE 10g DBMS. This software platform was the clients' preferred back-end. The database administrators were asked to use the basic SQL statements in order to level-set the group and not create too much divergence in specification.

In Figures 5, 6 and 7, the averaged results for the four respondents are shown for the scenarios in Table 1 and outlined in Figure 2. Considering the number of SQL statements, Figure 5 shows that in almost all cases the ChBAC method required fewer statements, tables and constraints to set up. The results are mirrored for the three measures (Number of SQL statements, Number of Tables and Number of Constraints) with the only exception being the managerial access scenarios where the results are the same. With regard to the number of SQL statements needed to set up the restrictions (Figure 5), we observe in all but two cases an economy of at least two statements with ChBAC as opposed to RBAC. It shows that the number of SQL statements required for the Chain method is reduced by factor of 50% for scenarios 1.1 and 5.1. While the percentage is 60% for scenarios 1.2, 1.4, 1.5, 2.3, 2.4, 2.5, 2.6, 3.1, 3.3, 7.1 and 8.1, the percentage becomes 80% for scenarios1.6, 2.2 and 3.2.

The results presented in Figure 6 refer to the number of tables that had to be created to accommodate the restrictions. In all but two scenarios there was a reduction in the number of tables required for ChBAC as compared

Figure 5. Comparison by total number of SQL statements. It shows that ChBAC needs number of SQL statements less than RBAC by factors of (50%, 60% and 80%).

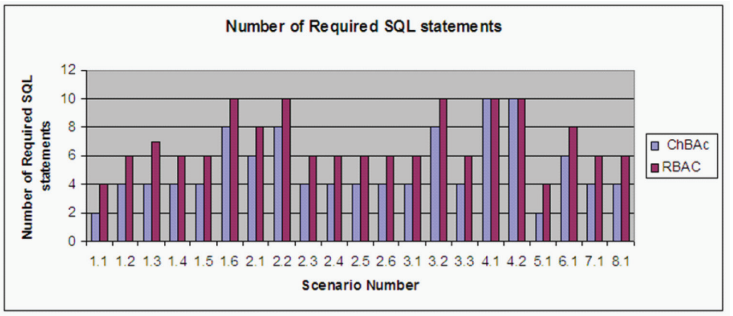


Figure 6. Comparison by total number of tables. It shows that ChBAC needs number of tables less than RBAC by factors of (50%, 60% and 80%).

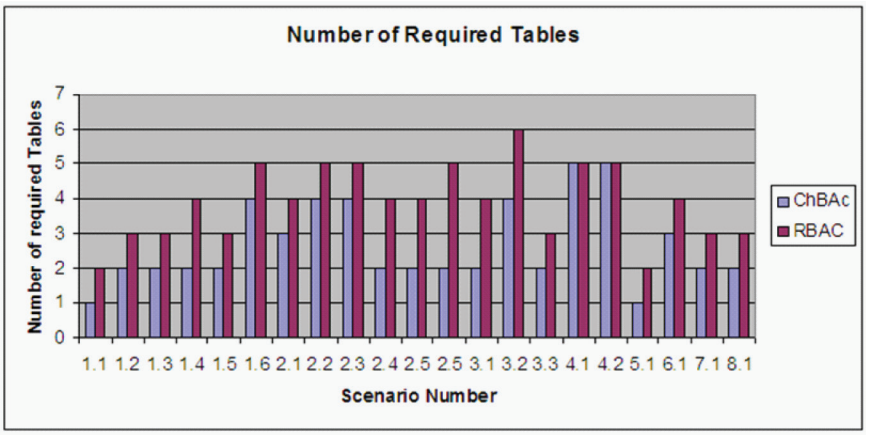
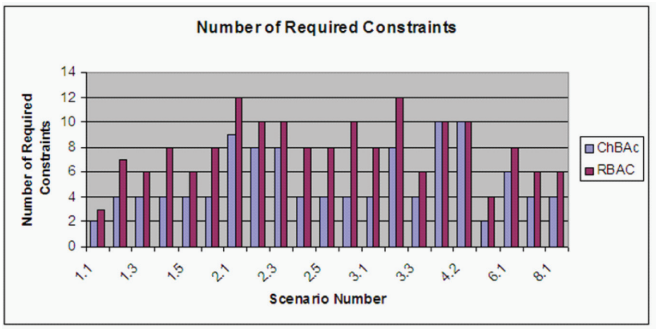


Figure 7. Comparison by total number of constraints. It shows that ChBAC needs number of constraints less than RBAC by factors of (50%, 60% and 80%).



to RBAC. Normally the economy was one table per scenario however in some cases the reduction in tables was higher. Again there was no difference for the management-related scenarios. It shows that the number of tables required for the Chain method is reduced by factor of 50% for scenarios 1.1, 1.4, 2.4, 2.5, 3.1 and 5.1. While the percentage is 60% for scenarios 1.2, 1.3, 1.5, 2.3, 2.4, 2.5, 2.6, 3.1, 3.3, 7.1 and 8.1, the percentage becomes 80% for scenarios 1.6, 2.2, 2.3 and 3.2.

The results shown in Figure 7, for the number of constraints broadly mirror the results shown in Figure 6 and while the number of constraints is generally larger than the number of tables, the same percentage change can be seen for the ChBAC results as compared to RBAC. It shows that number of constraints required for the Chain method is reduced by factor of 50% for scenarios 2.4, 2.5, 3.1 and 6.1. While the percentage is 66% for scenarios 1.3, 1.5, 2.3, 2.4, 2.5, 2.6, 3.1, 3.3, 7.1 and 8.1, the percentage becomes 80% for scenarios 1.6 2.3 and 3.2.

It should be noted however that these results are connected in that the SQL statements are used both for the setting up of the tables as well as for specifying constraints. In order to allow us to appreciate the overall picture and the overall result, we further analysed these results by calculating the average across the scenarios as well as the range of values and standard deviation. The results are presented in Figures 8, Figure 9 and Figure 10.

Figures 8, 9 and 10 show the average results for the two methods as the line between the grey and green blocks, while the extension of the blocks shows the standard deviation of results and the end of the lines emanating from the blocks show the range of values. The results on the left refer to the ChBAC Method while those on the right are concerned with the RBAC method.

The results for the number of SQL statements required, depicted in Figure 8, show an approximately 60% efficiency gain of the ChBAC method when compared with RBAC. While the whole range of values is more or less the same (given the average result and standard deviation), it can be seen that the ChBAC method consistently outperforms the RBAC method. While the upward variance of the standard deviation result is higher than in the case of RBAC, it nevertheless stays below the same result for RBAC overall, while at the lower end the standard deviation for RBAC only reaches the average result of the ChBAC method more or less.

Figure 9 shows the overall result for the number of tables. While at the lower end the range of values is the same as with RBAC at the top end the ChBAC method overall requires less tables. This is also reflected in the average results and the standard deviation that is somewhat more flexible on the top end but on average affords approximately 50% savings for the ChBAC method over RBAC.

With regard to the results on the number of constraints, shown on Figure 10, it can be observed that the performance of the ChBAC method shows approximately a 50% improvement in the average case and with less variance than the RBAC method. RBAC at the lower end performs very similar to the ChBAC Method, but for slightly more complex cases there is a definite advantage in using the ChBAC method.

The results for the different measures are overlapping to a certain extent and are not to be interpreted entirely separately but the results do show on average a 50% efficiency gain of the ChBAC method over the RBAC method.

We have not been able to confirm such results for other domains and more complex cases though there appears to be a definite advantage of using ChBAC over RBAC from a configuration perspective. We expect that if suitably implemented this reduced complexity could also speed up the assessment of privileges as users access the database to retrieve records.

DISCUSSION

It should be noted that the results presented in the previous section are not independent in that the SQL statements are used both for the setting up of tables and specifying constraints.

Figure 8. Aggregated results for SQL statements. It shows the average of required number of the SQL statements of the ChBAC method less by a factor of 50% than the RBAC method.

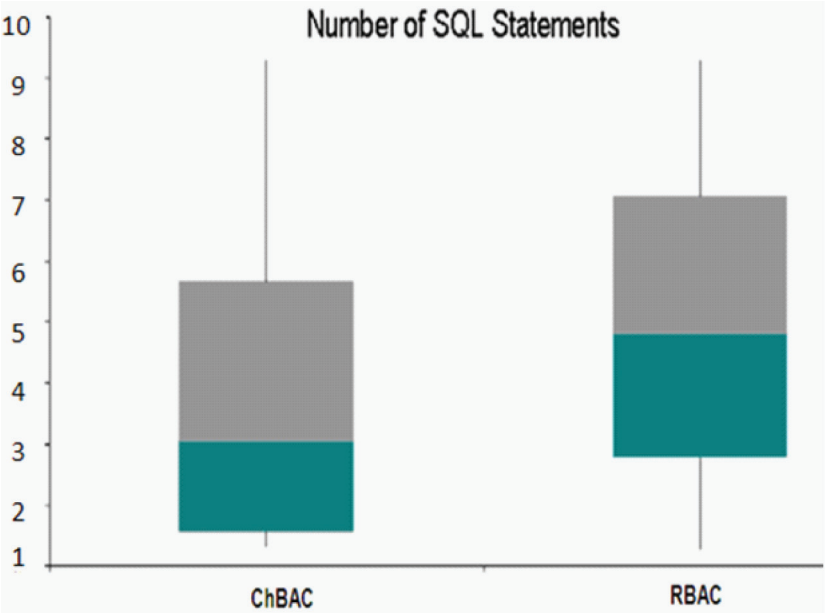


Figure 9. Aggregated results for number of tables. It shows the average of required number of the tables of the ChBAC method are less by a factor of 50% than the RBAC method.

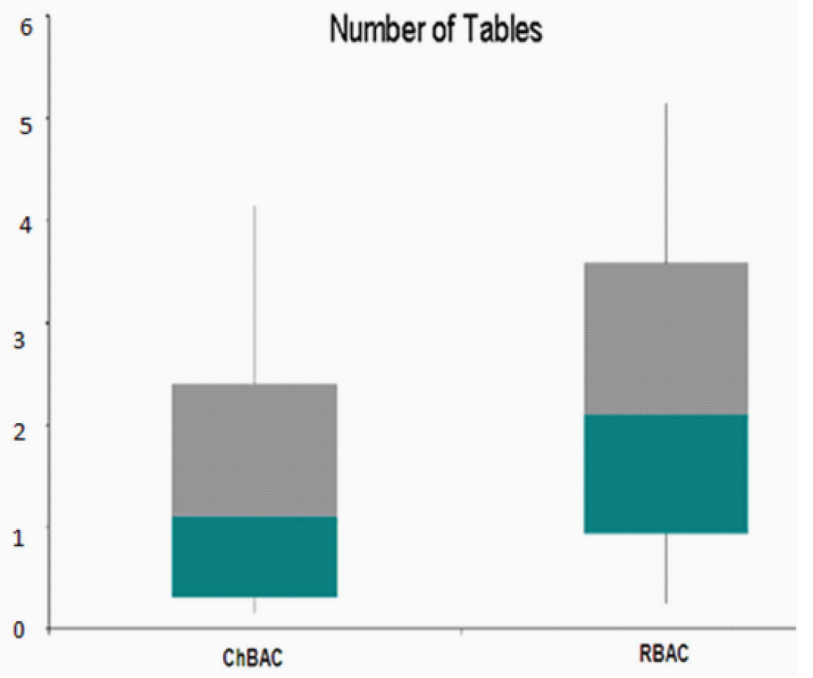
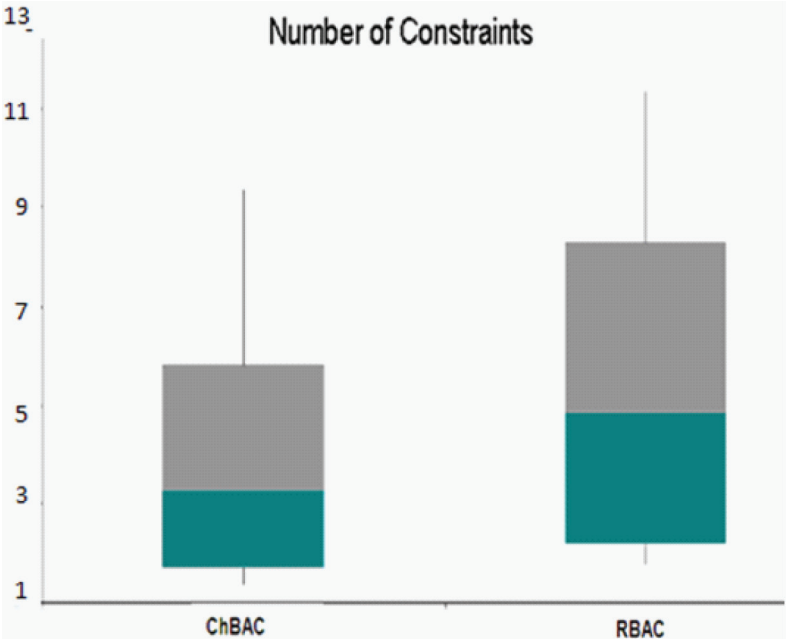


Figure 10. Aggregated results for number of constraints. It shows the average of required number of constraints of the ChBAC method are less by a factor of 50% than the RBAC method.



There appears to be a definite advantage of using ChBACs over RBAC from a configuration perspective and the authors expect that if suitably implemented this reduced complexity could also speed up the assessment of privileges as users access the database to retrieve records, though this remains to be demonstrated.

The benefits in terms of the measures presented are also reflected by the participants' responses to an exit poll. This was conducted by running experiments with database administrators. In the experiments, differing numbers of records, for differing scenarios, were used across the two methods. Despite being seasoned implementers of RBAC access restrictions the database administrators preferred the ChBAC method and felt that it was less complex and easier to implement. All five of our respondents, when questioned about their views on these two methods following the test implementations, agreed on the potential of the ChBAC method for their work in the database administration of the hospital. They were considering applying the ChBAC method to the new branches of the hospital. They felt that the limited acts would help reduce the time to complete the database design. They were impressed by the fact that setting up the required restriction took them half the time using ChBAC as compared to RBAC as have been shown in the previous section.

CONCLUSION

To the researchers' knowledge, the classical chain method that has been suggested by Al-Fedaghi (2007) has never been implemented nor tested in any hypothetical nor real enterprise. In addition, it has never been designed to solve any particular problem such as the problem of managing access to personal information in healthcare without loss of privacy.

In this paper, we have presented the ChBAC Method, which we purport allows easier specification of policy during the design phase and more sophisticated control during runtime than the RBAC. The paper showed a comparison

between ChBAC and RBAC based on three main criteria: number of SQL commands required to apply the access method, number of tables and number of constraints to apply each method. The comparison results showed that the ChBAC overcomes the RBAC for all criteria as it needs less SQL commands, tables and constraints. As a conclusion, this paper recommends ChBAC to be used as a reliable method in data access management for real applications.

We would at this point also like to express our gratitude to the International Clinic Kuwait and their database administrators for their kind assistance, without which we would not have been able to carry out this evaluation.

REFERENCES

W3C. (2009). *The platform for privacy preferences 1.0 specification*. World Wide Web Consortium. Retrieved October 15, 2009 from http://www.w3.org/TR/P3P/

Agrawal, R., Kiernan, J., Srikant, R., & Xu, Y. (2002). Hippocratic databases. In *Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, China (pp. 143-154).

Al-Fedaghi, S. (2007). Beyond purpose-based privacy access control. In *Proceedings of the 18th Australasian Database Conference*, Ballarat, Australia.

Borthakur, D. (2007). *The Hadoop distributed file system: Architecture and design.* Retrieved October 15, 2009 from http://hadoop.apache.org/core/docs/current/hdfs design.pdf

Byun, J. W., Bertino, E., & Li, N. (2005). Purpose based access control of complex data for privacy protection. IN *Proceedings of the 10thACM Symposium on Access Control Models and Technologies*, Stockholm, Sweden.

Dick, R., Steen, S., Elaine, B., & Detmer, D. E. (1997). The computer-based patient record: An essential technology for health care. In National Academy Press-Book. ISBN 0309055326. Washington, D.C.

EPAL. (2003). *Enterprise privacy authorization language (EPAL 1.2).* Retrieved October 15, 2009, from http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/

Ferraiolo, D. F., & Kuhn, R. (1992). Role-based access control. In *Proceedings of the 15th NIST-NSA National Computer Security Conference*, 554-563.

Grandison, T., Johnson, C., & Kiernan, J. (2008). Hippocratic databases: Current capabilities and future trends. In M. Gertz, & S. Jajodia (Eds.), *Handbook of database security: Applications and trends*. New York, NY: Springer. doi:10.1007/978-0-387-48533-1_17

Health Level Seven Inc. (2009). *HL7 standard*. Retrieved October 15, 2009 from http://www.hl7.org/

International Clinic. (2011). Retrieved from http://www.international-clinic.com/

Noffsinger, R., & Chin, S. (2000). Improving the delivery of care and reducing healthcare costs with the digitization of information. *Journal of Healthcare Information Management*, *14*(2), 23–30. PMID:11066646

Omran, O., Grandison, T., & Abu Almaati, S. (2010). Healthcare chains - Enabling application and data privacy controls for healthcare information systems. In *Proceedings of the 13th World Congress on Medical and Health Informatics (MEDINFO) 2009*, Cape Town, South Africa.

Sandhu, R. S. (1998). Role-based access control. *Advances in Computers*, *46*, 237–286. doi:10.1016/S0065-2458(08)60206-5

Tektonidis, D., Bokma, A., Oatley, G., & Salampasis, M. (2005). ONAR: An ontologies-based service oriented application integration framework. In *Proceedings of the First International Conference on Interoperability of Enterprise Software and Applications, Lecture Notes in Computer Science (Interoperability of Enterprise Software and Applications)*, ISBN: 1-84628-151-2, Geneva, Switzerland.

Thomas, R. K., & Sandhu, R. S. (1993). Task-based authorization controls (TBAC), A family of models for active and enterprise-oriented authorization management. In *Proceedings of the IFIP WG11.3 Workshop on Database Security*, Lake Tahoe, CA.

Thomson Reuters. (2009). *100 top hospitals: 2009.* Retrieved 15, 2009 from http://www.modernhealthcare.com/section/lists?djoPage=product_details&djoPid=10537&djoTry=1249923457

University of Alberta (2005). *Electronic health records and the personal information protection and electronic documents act,* Health Law Institute, University of Victoria, School of Health Information Science. Report prepared with generous funding support from the Office of the Privacy Commissioner of Canada.

Webster, C. (1990). Conflict and consensus: Explaining the British health service. *Twentieth Century British History*, *1*(2), 115–151. doi:10.1093/tcbh/1.2.115 PMID:11622411

Esraa Omran is an Assistant Professor of Computer Science at Gulf University for Science & Technology (GUST). Omran received his B.Sc. and M.Sc. degrees from the department of Electrical and computer Engineering from Kuwait University, State of Kuwait, in 2001 and 2006, respectively. She completed her Ph.D. in the Department of Computer Science at University of Sunderland, United Kingdom, in 2013. Omran joined the GUST faculty in 2013.Omran is a member of the IEEE-Computer society. She had a patent and study agreement proposals with IBM Company. Omran's current research interest is in the area of Security, Data access management, Networking, Ontology and Semantic web.

Tyrone W A Grandison is the CEO of Proficiency Labs International, which specializes in supporting healthcare entities design, build and evaluate privacy and security solutions for their systems. He received a B.S. degree in Computer Studies (Computer Science and Economics) from the University of the West Indies in 1997, a M.S. degree in Software Engineering in 1998 and a Ph.D. degree in Computer Science from the Imperial College of Science, Technology & Medicine in London (2003). Dr. Grandison is an IBM Master Inventor, a Distinguished Engineer of the Association of Computing Machinery (ACM), a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), a Fellow of the British Computer Society (BCS), has been recognized by the National Society of Black Engineers (as Pioneer of the Year in 2009), by the Black Engineer of the Year Award Board (as Modern Day Technology Leader in 2009, and Minority in Science Trailblazer in 2010, Science Spectrum Trailblazer in 2012) and has received the IEEE Technical Achievement Award in 2010 for "pioneering contributions to Secure and Private Data Management". He has authored over 90 technical papers and co-invented over 30 patents.

David Nelson is a Senior Lecturer in the Department of Computing, Engineering and Technology, at the University of Sunderland. His PhD and subsequent research has been in the area of databases, with focused interests in data models, data integration and database pedagogy. He also participates in reach-out with external companies, having worked on Knowledge Transfer projects with a number of companies in the North East of England. He is currently a member of the program and steering committees for both the British National Conference on Databases and the HE Academy Workshop on Teaching, Learning and Assessment of Databases.

Albert Bokma currently works as consultant at Avedas AG on research information systems since 2012. He previously was a Research Fellow at the University of Sunderland where he taught in Information Systems and supervised several doctoral students. He has been principal investigator as well as project manager on a series of European collaborative RTD projects. He holds a PhD and MSc in Computer science from the University of Durham, UK, and graduated from University College London with a BA (Hons) in Philosophy.