IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 5, Oct-Nov, 2013
ISSN: 2320 - 8791
www.ijreat.org

An Efficient Controlled Information Sharing Using Break-Glass Policy in Healthcare Systems

Subetha.T[1] and Rajasekar.M[2]

[1]**ME CSE, Vel Tech Multi Tech Engineering College, Avadi, Chennai.**

[2]**Asst Professor, Department of Computer Science and Engineering, Vel Tech Multi Tech Engineering College, Avadi, Chennai.**

Abstract

During natural disasters or emergency situations, an essential requirement for an effective emergency management is the information sharing. In this paper, we present an access control model to enforce controlled information sharing during emergency situations. We also move this environment to cloud computing environment .Moving the data into cloud raises security issues like replay attacks, modification of the data which needs to be handled. So we use an encryption scheme namely attribute based encryption .Attribute based encryption encrypt the data based on the attributes of the users. The decryption of a ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext. A crucial security feature of Attribute-Based Encryption is collision-resistance. We extend our prototype to support break-glass policies which would allow the system to trace the violations of temporary access control policy.

Index Terms*—Access controls, privacy, security, data sharing, Attribute based encryption.*

1. INTRODUCTION

In the last years, natural catastrophic events, e.g., floods, earthquakes, hurricanes, and man-made disasters,e.g., airplane crashes, terrorist attacks, nuclear accidents, highlight the need for a more efficient emergency management. The lack of effective information sharing results in the need for a more efficient, timely and flexible information sharing during emergency management. The main area that draws an attention in information sharing during emergency management is healthcare systems. Therefore our main focus on this project is to propose an access control model for health care systems to enforce controlled information sharing during emergency situation. Emergency policies allow the instantiation of temporary access control policies (tacps) that override regular policies during emergency situations. In general, in emergency management scenarios the response plans are defined by experts on the field based on regulations and laws and based on reports resulting by the emergency preparedness phase, during which emergency managers conduct a risk assessment study[3].

Additionally we propose an information sharing among multiple organizations exploiting new cloud computing techniques. Cloud computing is suitable for the purpose of information sharing because it provides a common storage space where organization can share their data. Though moving the data into cloud is flexible it also raises several security issues such as replay attacks, modification of data etc.

A feasible and promising approach would be to encrypt the data before outsourcing. Basically the health record owner herself should decide how to encrypt her files and to allow which set of users to obtain access to each file. A health record should only be available to the users who are given the corresponding decryption key, while remain confidential to the users.

Furthermore they have retain the right not only to grant but also revoke access privileges when they feel it necessary[5].In order to protect the health data stored on a semi trusted server, we adopt attribute based encryption(ABE) as the main encryption primitive. Using ABE , access policies are expressed based on the attributes of users or data ,which enable a patient to selectively share the data among a set of users by encrypting the file under a set of attributes, without the need to know the complete list of users. This makes ABE especially attractive in dynamic environments such as grid computing, disaster management or the health care area.ABE creates particular challenges for providing one important feature break-glass, i.e., the traceable ,ad-hoc override of access control policies. Traditionally, break-glass access control systems are implemented in systems using user-based access control policies and centralized policy decisions points. In this paper, we present an integration of fine-grained break-glass concepts into a system for secure information sharing based on ABE.

The remainder of this paper is organized as follows. Section 2 presents an overview of the model. Section 3 presents about ABE. Section 4 presents ABE with break-glass. Section 5 presents prototype implementation .Section 7concludes the paper.

2. EMERGENCY INFORMATION SHARING

To enforce flexible information sharing during emergencies, it is usually necessary to grant users access to resources not normally authorized. Moreover ,it is often the case that specific actions should be performed to manage the emergency.To fulfil both these requirements the model presented in[7] supports tacps to be activated during emergencies and obligations that have to be fulfilled when an emergency is detected. The connection of an emergency with the corresponding tacps and obligations is modeled by emergency policies. The key characteristic of the model in[7] is that emergencies are specified through events on top of Complex Event Processing (CEP) systems [2]. A language, called Core Event Specification Language (CESL), is used to define events describing the beginning/ending of an emergency. The formal syntax of CESL operators is reported in [7].

Definition 2.1 (Emergency description). An emergency emg is a tuple (init, end, time-out, identifier), where init and end are emergency events specified in CESL, such that init denotes the event triggering the emergency and end is the optional event that turns off the emergency, time-out is the time within the emergency expires even though end has not occurred. Identifier is an attribute belonging to both the schemes of the event type corresponding to init and end eventsNote that the identifier plays a key role in that it ensures the connection between init and end events (see [7]for further details) as shown in the following example, which also illustrates the reference scenario used throughout the paper. This has been chosen to show how our model works in a challenging domain, where the number of emergencies and related emergency policies is large and the level of policy granularity is high. Even if we are aware that this is not a typical domain for emergency management (e.g., disaster management), we decide to select it because it gives us the opportunity to provide more complex examples of emergency descriptions and policies.

Example 2.1(Reference scenario) Patients are hospitalized at home, in a special clinic or in a hospital. Each of these structures provides patient treatments through specialized equipment able to ensure a real-time monitoring of patient vital signs. Data gathered by the monitoring equipment are collected by the CEP to automatically detect emergency situations. More precisely, we suppose that, every 30 seconds, each sensor sends the systolic pressure of patients to the monitoring system in theVitalSigns stream of tuples (pressure…patienti).A hypertension emergency can be defined as follows:

HypertensionE
mergencyinit:
VS1 v_1*;*
VS1 σ(pressure >
140)(VitalSigns);
end: VS2 v_2*;*
VS2 σ(pressure _
120)(VitalSigns);
timeout: ∞;
identifier:
patient_id;
end;

The emergency starts when the diastolic pressure of a patient is higher than 140 mmHg, and it ends when the pressure of the same patient (i.e., with the same patient_id) returns to less than or equal to 120 mmHg. When the Hypertension Emergency is detected for patient 1,the following emergency instance is created.

HypertensionEmergencyI
nstance1
emg:
HypertensionEmergency;
identifier:1;

The HypertensionEmergencyInstance1 is deleted when HypertensionEmergency ends for patient 1. Our model enforces controlled information sharing during emergencies through tacps. More precisely, because different instances of the same emergency might require different tacps, we associate with an emergency a tacp template, that will be properly instantiated when an emergency is detected.

Definition 2.2 (tacp template). A tacp template is a tuple (sbj, obj, priv, ctx, obl), with the following semantics: When the Boolean expression ctx defined on context is true, users identified by the subject specification sbj are authorized to exercise the privilege priv on the resource identified by object specification obj. In case the obligation obl is not null, it denotes a set of actions that must be fulfilled every time an authorized user exercises priv on the objects denoted by obj.

In [7], we intentionally gave a high-level definition of subject/object specification and context condition, whereas in this proposal, we adopt a model similar to attribute-centric RBAC-A [9]. The model in [9] is a combination of role-based access control and attribute-based access control (ABAC). We choose this model because we need to identify users by their roles (e.g., doctor, patient, and so on) as well as specify attribute-based conditions. Therefore, in our proposal, a subject specification sbj is a pair (roles, cond), where the first is a set of authorized roles and the second is a condition related to the user profile attributes. An object specification obj is a pair (object, cond), where object denotes a target object and cond is a condition related to the object attributes. The context is modeled as a set C of pairs (att, val), where att is a context attribute (e.g., time, location, session information, and so on) and val is the corresponding value.

Example 2.2. Consider the hypertension emergency pre-sented in Example 2.1.Suppose that, during this emergency, access to the electronic medical record (EMR) of a patient (object condition) should be extended to the subjects taking care of him/her (subject condition—e.g., paramedics). Moreover, when a subject not normally authorized accesses the EMR, the corresponding patient should be warned with an email (obligation). To enforce these requirements, the following tacp template can be defined:

```
HypertensionPolicy
  sbj: (paramedic, param_id =
  call.param_id);
  obj: (EMR patient_id = emg.patient_id);
  priv: read;
  ctx: -;
  obl:
mailto(patient_mail);
end
```

1. Cond is a Boolean combination of predicates in the form α, β, θ_, where α is an attribute belonging to the user profile (object, respectively),θ is a matching operator in (≥,≤,) whereas β is a constant value or an attribute att.

3. ABE FOR HEALTH CARE DATA

The main goal of our framework is to provide secure Health care data access and efficient key management at the same time the key idea to divide the system into multiple security domains (namely, public domains and personal domains) according to the different users‟ data access requirements. The PUDs(Public Domain) consist of users who will access based on their professional roles, such as doctors, nurses, and medical researchers. In practice, a PUD can be mapped to an independent sector in the society, like health care. For each PSD,the corresponding users are personally associated with a data owner (such as family members or friends),and they make accesses to health data based on access rights assigned by the owner. In both types of security domains, we utilize ABE to realize cryptographically enforced, health care data access. Especially , in a PUD multi authority ABE is used, in which there are multiple "attribute authorities "(AAs), each governing a disjoint subset of attributes. Role attributes are defined for all PUDs, representing the professional role or obligations of a PUD user. Users in PUDs obtain their attribute-based secret keys from AAs, without directly interacting with the owners. To control access from PUD users, owners are free to specify role-based fine-grained access policies for her health care data files while do not need to know the list of authorized users when doing encryption. Since the PUDs contain the majority of users, it greatly reduces the key management overhead for both the owners and users . Each data owner (e.g. , patient) is a trusted authority of her own PSD(public and personal domains), who uses a KP-ABE system to manage the secret keys and access rights of users in her PSD. Since the users are personally known by the owner, to realize patient-centric access, the owner is at the best position to grant user access privileges on a case-by-case basis. For PSD, data attributes are defined which refer to the intrinsic properties of the health care data, such as the category of a health care file . For the purpose of PSD access, each health file is labeled with its data attributes, while the key size is only linear with the number of file categories a user can access. Since the number of users in a PSD is often small, it reduces the burden for the owner. When encrypting the data for PSD, all that the owner needs to know is the intrinsic data properties The multi - domain approach best models different user types and access requirements in health care system. The use of ABE makes the encrypted file self-protective, i.e, they can be accessed by only authorized users even when storing on a semi trusted server, and when the owner is not online . In addition, an efficient and on demand user revocation is made possible via our ABE enhancements.

4. DEAL WITH BREAK-GLASS ACCESS

For certain parts of the health care data , medical staffs need to have temporary access when an emergency happens to a patient ,who may become unconscious and is unable to change her policy .The medical staffs will need some temporary authorization (e.g emergency key) to decrypt those data. Under our framework, this can be normally achieved by letting each patient delegate her emergency key to an emergency department. Specifically in the beginning, each owner defines an emergency attribute and builds it into the PSD part of the cipher text of the healthcare document that she allows break-glass access. She then generates an emergency key sk_{EM} using the single node key-policy "emergency" and delegates it to the ED who keeps it in a database of patient directory. Upon emergency a medical staff authenticates herself to the ED, request and obtains the corresponding patient‟ssk_{EM}, and then decrypts the documents using sk_{EM}..After the patient recovers from emergency, she can revoke the break-glass by computing a rekey: rk_{EM}, submit it to the ED and the server to update her sk_{EM} and CT to their newest versions, respectively.

5. PROTOTYPE IMPLEMENTATION

In our framework, there are multiple SDs, multiple owners, multiple AAs, and multiple users. In addition, two ABE systems are involved: for each PSD the YWRL‟s revocable KP-ABE scheme [6] is adopted; for each PUD, our proposed revocable MA-ABE scheme is used. The framework is illustrated in Fig. 1. We term the users having read and write access as data readers and contributors, respectively.

System setup and key distribution: The system first defines a common universe of data attributes shared by every PSD, such as "basic profile," "medical history," "allergies," and "prescriptions." An emergency attribute is also defined for break-glass access. Each owner‟s client application generates its corresponding public/master keys. The public keys can be published via user‟s profile in an online healthcare social-network (HSN) (which could be part of the health care service; e.g., the Indivo system [20]). There are two ways for distributing secret keys. First, when first using the PHR service, a owner can specify the access privilege of a data reader in her PSD, and let her application generate and distribute corresponding key to the latter, in a way resembling invitations in GoogleDoc. Second, a reader in PSD could obtain the secret key by sending a request (indicating which types of files she wants to access) to the owner via HSN, and the owner will grant her a subset of requested data types. Based on that, the policy engine of the application automatically derives an access structure, and runs keygen of KP-ABE to generate the user secret key that embeds her access structure. When the user is granted all the file types under a category, her access privilege will be represented by that category instead.

For the PUDs, the system defines role attributes, and a reader in a PUD obtains secret key from AAs, which binds the user to her claimed attributes/roles. For example, a physician in it would receive "hospital A, physician, M.D., internal medicine" as her attributes from the AAs. In practice, there exist multiple AAs each governing a different subset of role attributes. For instance, hospital staffs shall have a different AA from pharmacy specialists. This is reflected by(1) in Fig. 1. MA-ABE is used to encrypt the data In addition, the AAs distribute write keys that permit contributors in their PUD to write to some patients‟ PHR (patients healthrecord) (2).

PHR encryption and access: The owners upload ABE-encrypted PHR files to the server (3). Each owners PHR file is encrypted both under a certain fine- grained and role-based access policy for users from the PUD to access, and under a selected set of data attributes that allows access from users in the PSD. Only authorized users can decrypt the PHR files, excluding the server. The data readers download PHRfiles from the server, and they can decrypt the files only if they have suitable attribute-based keys (5). The data contributors will be granted write access to someone‟s PHR, if they present proper write keys (4).

User revocation: Here, we consider revocation of a data reader or her attributes/access privileges. There are several possible cases:

- revocation of one or more role attributes of a public domain user;
- revocation of a public domain user which is equivalent to revoking all of that user‟s attributes. These operations are done by the AA that the user belongs to, where the actual computations can be delegated to the server to improve efficiency(8).
- revocation of a personal domain user‟s access privileges;
- revocation of a personal domain user. These can be initiated through the PHR owner‟s client application in a similar way.

Policy updates. A PHR owner can update her sharing policy for an existing PHR document by updating the attributes (or access policy) in the cipher text. The supported operations include add/delete/modify, which can be done by the server on behalf of the user.

Break-glass. When an emergency happens, the regular access policies may no longer be applicable.To handle this situation, break-glass access is needed to access the victim‟s PHR. In our framework, each owner‟s PHR‟s access right is also delegated to an emergency department (ED, (6)). To prevent from abuse of break-glass option, the emergency staff needs to contact the ED to verify her identity and the emergency situation, and obtain temporary read keys (7). After the emergency is over, the patient can revoke the emergent access via ED.

An example. Here, we demonstrate how our framework works using a concrete example. Suppose PHR owner Alice is a patient associated with hospital A. After she creates a PHR file F_1 .she first encrypts it according to both F_1‟s data labels (under the YWRL KP-ABE), and a role-based file access policy P_1 (under our revocable MA-ABE). This policy can be decided based on recommended settings by the system, or Alice‟s own Preference. It may looks like P_1 := „„„(profession = physician)^(specialty = internal Medicine)^(organization = hospital A)‟‟‟‟:

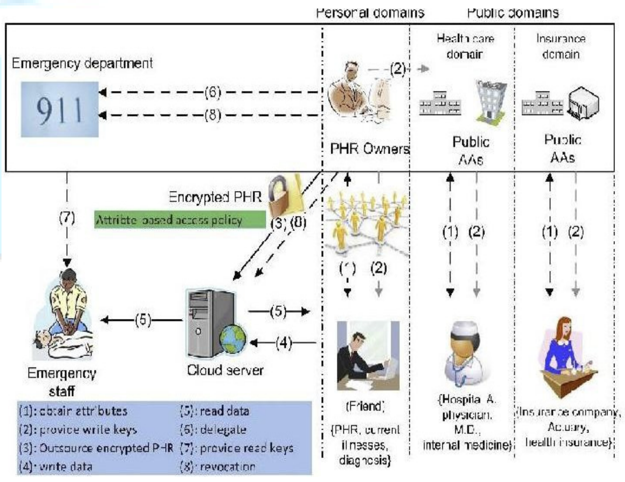

Fig. 1.The proposed framework for patient-centric, secure and scalable PHR sharing on semitrusted storage under multiowner settings

She also sends the break-glass key to the ED. In addition, Alice determines the access rights of users in her PSD, which can be done either online or offline. For example, she may approve her friend Bob‟s request to access files with labels {personal info} or {medical history}.

Her client application will distribute a secret key with the access structure {personal info v medical history} to Bob. When Bob wants to access another file F_2 with labels "PHR—me-dical history—medications," he is able to decrypt F_2 due to the "medical history" attribute. For another user Charlie who is a physician specializing in internal medicine in hospital B in the PUD, he obtains his secret key from multiple AAs such as the American Medical Association (AMA), the American Board of Medical Specialties (ABMS), and the American Hospital Association (AHA). But he cannot decrypt F_1, because his role attributes do not satisfy P_1. Finally, an emergency room staff, Dorothy who temporarily obtains the break-glass key from ED, can gain access to F_1 due to the emergency attribute in that key.

6. RELATED WORK

Our model enforces fine-grained access control with attribute-level granularity. Many models have been pro-posed in the literature, in support of fine-grained access control, for instance models derived from ABAC [or the XACML standard [4]. A remarkable model supporting fine-grained access control in a healthcare domain is C-TMAC presented in [8]. This approach allows team-based access control by also integrating contextual information. In [7], we intentionally gave a high-level definition ofthe access control model, whereas in this paper, we adoptRBAC-A. We believe the above-mentioned models can be adopted in our system instead of RBAC-A. However, none of them support emergency detection through CEP technology, which is a total novelty in access control systems.

6.1. ABE for fine – Grained data access Control

A number of works used ABE to realize fine-grained access control for outsourced data [11], [12], [6], [13]. Especially, there has been an increasing interest in applying ABE to secure electronic healthcare records (EHRs). Recently, Narayan et al. proposed an attribute-based infrastructure for EHR systems, where each patient‟s EHR files are encrypted using a broadcast variant of CP-ABE [14] that allows direct revocation. However, the cipher text length grows linearly with the number of unrevoked users.

6.2 Revocable ABE

It is a well-known challenging problem to revoke users/ attributes efficiently and on-demand in ABE. Traditionally, this is often done by the authority broadcasting periodic key updates to unrevoked users frequently [11], [15], which does not achieve complete backward/forward security and is less efficient. Recently, [16] and [17] proposed two CP-ABE schemes with immediate attribute revocation capability, instead of periodical revocation. However, they were not designed for MA-ABE.

7. CONCLUSIONS

In this paper, we proposed an extension of the emergency access control model presented in [11] with the possibility of defining administration policies, i.e., which subjects are enabled to define emergency policies and over which scope. we also carried out the work in cloud computing and provides security by using ABE. In addition, we would like to extend our prototype so as to support the break glass policies. This would allow the system to trace violations of tacps. We utilize ABE to encrypt the health care data, so that patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications, and affiliations. We plan to extend this work along different directions. First of all, we intend to carry out several experiments to evaluate the time needed for emergency administration policies enforcement. Moreover, we believe that tools to assist security administrator in emergencies and emergency policies definitions can be defined. More precisely, because a large number of risk assessment tools have been developed in the last years [22], we plan to analyze them so as to automatically extract emergency descriptions, policies, and obligations from emergency scenarios and response plans.

REFERENCES

[1] "The 9/11 Commission Report," technical report, Nat‟lCommis-sion on Terrorist Attacks upon theUnited States, July 2004.

[2] A.Margara and C.Cugola"processing flows of information:From data stream to Complex Event Processing",Proc,fifth ACM Int‟l Conference Distributed event based systems pp 359-360,2011

[3] T. F. E. M. A. (FEMA), "Emergency Response Plan Implementation @ONLINE," Sept. 2012.

[4] H.L. Bill Parducci, "eXtensible Access Control Markup Language (XACML) Specification 3.0," Aug. 2010.

[5] K.D. Mandl, P. Szolovits, and I.S. Kohane, "Public Standards and Patients‟ Control: How to Keep Electronic Medical Records Accessible but Private," BMJ, vol. 322, no. 7281, pp. 283-287, Feb. 2001.

[6] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM ‟10, 2010.

[7] B. Carminati, E. Ferrari, and M. Guglielmi, "Secure Information Sharing on Support of Emergency Management," Proc. IEEE Third Int‟l Conf. Privacy, Security, Risk and Trust (PASSAT), and IEEE Third Int‟l Conf. Social Computing (SocialCom), pp. 988-995, Oct. 2011.

[8] C.K. Georgiadis, I. Mavridis, G. Pangalos, and R.K. Thomas, "Flexible Team-Based Access Control Using Contexts," Proc. Sixth ACM Symp.Access Control Models and Technologies (SACMAT "01), pp. 21-27, 2001.

[9] D.R. Kuhn, E.J. Coyne, and T.R. Weil, "Adding Attributes to Role-Based Access Control," Computer, vol. 43, no. 6, pp. 79-81, June 2010.

[10]European network and information securityagency,"Inventory of risk management/risk assessment methods",sept 2012

[11]A. Boldyreva, V. Goyal, and V. Kumar, "Identity-Based Encryp-tion with Efficient Revocation," Proc. 15th ACM Conf. Computer and Comm. Security (CCS), pp. 417-426, 2008.

[12]L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Ciphertext-Policy Attribute-Based Threshold Decryption with Flexible Delegation and Revocation of User Attributes," 2009.

[13]S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. Fifth ACM Symp.Information, Computer and Comm. Security (ASIACCS "10), 2010.

[14]S. Narayan, M. Gagne´, and R. Safavi-Naini, "Privacy Preserving EHR System Using Attribute-Based Infrastructure," Proc. ACM Cloud Computing Security Workshop (CCSW "10), pp. 47-52, 2010.

[15]X. Liang, R. Lu, X. Lin, and X.S. Shen, "Ciphertext Policy Attribute Based Encryption with Efficient Revocation," technical report, Univ. of Waterloo, 2010.