RBTBAC: Secure Access and Management of EHR Data

Rui Zhang, Jiqiang Liu, Zhen Han
School of Computer and Information Technology
Beijing Jiaotong University
Beijing, China 100044
Email: zhangrui,jqliu,zhan@bjtu.edu.cn

Ling Liu
College of Computing
Georgia Institute of Technology
Atlanta, GA, USA 30332-0765
Email: lingliu@cc.gatech.edu

***Abstract*—Security and privacy are widely recognized as important and personalized requirements for access and management of Electric Health Record (EHR) data. Different patients may have different privacy and security policies for their EHR data in different context. In this paper we argue that EHR data needs to be managed with customizable access control in both spatial and temporal dimension. We present a role-based and time-bound access control model (RBTBAC) that provides more flexibility of both roles (spatial capability) and temporal capability to control the access of sensitive data from time dimension. Through algorithmic combination of role-based access control and time-bound key management, RBTBAC model has three salient features. First, we have developed a privacy-aware and dynamic key structure for role-based privacy aware access and management of EHR data, focusing on the consistency of access authorization (including data and time interval) with the activated role of user. In addition to role-based access, a path-invisible EHR structure is build for preserving privacy of patients. Second, we have employed a time tree method for generating time granule values, offering fine granularity of time-bound access authorization and control. Our experimental results show that tree-like time structure can improve the performance of the key management scheme significantly and RBTBAC model is more suitable than existing solutions for EHR data management since it offers high-efficiency and better security and privacy for patients.**

Keywords-EHR system; privacy preserving; role-based access control; time-bound key management; time tree

I. Introduction

Electric Health Record (EHR) is a shared digital record across different healthcare settings, by network-connected enterprise-wide information systems called EHR systems. On one hand, EHR systems hold the promise to provide fast and on-demand access to medical documents and help reduce medical errors and enhance healthcare quality by providing healthcare workers with decision support. On the other hand, this openness, while being an essential feature of EHR, also exposes patients to the risks of privacy disclosure by improper authorization, misuse and abuse. Security and privacy are widely recognized as important non-functional requirements for access and management of EHR data.

We identify three main requirements for making EHR data access secure and privacy preserving.

First, when a patient is offered medical treatment, he expects that his medical records can only be accessed by authorized doctors, and other unauthorized doctors should not be able to read any part of his medical record.

Second, in the healthcare setting, a physician is allowed to access the medical data of a specific patient only during the time period of offering healthcare treatment. For example, emergency room doctor should not be able to access any medical document of a patient, once he or she has completed the emergency treatment of the patient and the patient has left the emergency room. In addition, given a patient, different doctors involved should have different time intervals in terms of accessibility to this patient's medical data. Furthermore, a doctor may need to access the medical data of different patients at the same time period. Thus, we need to support time-bound access and to be able to manage the accessibility of EHR data from time dimension in healthcare domain.

Finally, it is to the best interest of some patients that their doctor only know the medical data that are relevant to the diseases currently under treatment by the doctor (such as dental disease) but should not be able to access other medical data (such as psychotherapy data). Therefore, in addition to EHR data, the role-based structure of EHR data should not be released to the doctors or any other users who do not hold the authorization to access them, since the contents of internal nodes in such structure may contain sensitive information that can be used by healthcare professionals to infer other diseases of the patient. Moreover, the structure of EHR data may change frequently due to the involvement of different sets of healthcare providers for different medical conditions and treatments. Thus, the third challenge in securing EHR access is to make the structure of EHR data and access path invisible to users who are unauthorized or only authorized to access a partial component of the EHR data.

The main challenge is to find an efficient and scalable approach to integrate role-based access control and time bound key management under a coherent access control framework such that we can deliver both security and scalability required for role-based and time-bound access to EHR data. However, most existing time bound key management schemes [1]–[5] use linear hash chain to generate time granule value, which is known to be inefficient and suffers from poor scalability when

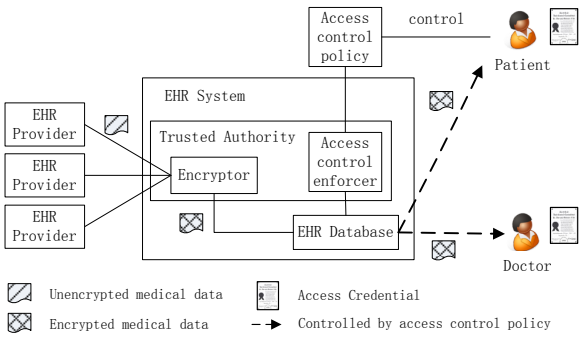


Figure 1. Security model for EHR system

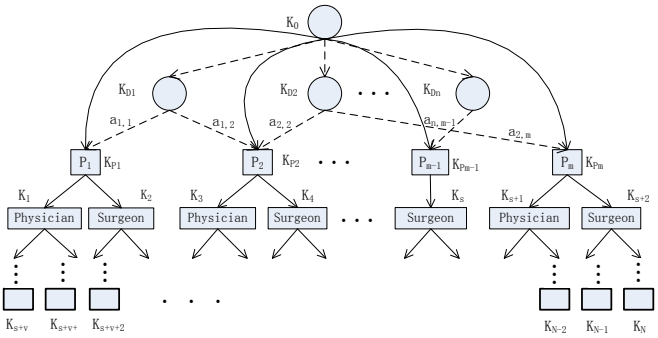


Figure 2. Hierarchical key structure

the size of the datasets to be protected for time-bound access is huge.

In this paper, we propose a general purpose *role-based and time-bound access control (RBTBAC)* model for EHR systems. The development of RBTBAC model is based on algorithmic combination of *role-based* access control and *time-bound* hierarchical key management such that a legitimate user of EHR system is authorized a time interval to access EHR data based on his/her role. The RBTBAC model offers more flexibility of both roles (spatial capability) and temporal capability to control the access of sensitive data from time dimension. Concretely, we have developed a privacy-aware and dynamic key structure for role-based privacy aware access and management of EHR data, focusing on the consistency of access authorization (including data and time interval) with the activated role of user. In addition we have employed a time tree method for generating time granule values, offering fine granularity of time-bound access authorization and control. Through extensive experimentation, we demonstrate RBTBAC model not only offers better security and privacy for access EHR data, but also provides high efficiency and customizability.

II. RBTBAC Model for EHR System

The reference security model mainly has three entities cooperatively to manage and control the usage of EHR data in security as shown in Fig.1. The trusted authority (TA) contains two parts, one is encryptor who is responsible for encrypting various EHR data from different EHR providers into a uniform ciphertext format, the other is access control enforcer who is used to enforce predefined access control policies. The remote EHR database is a necessary component to store the encoded composite EHR data. Additionally, an access control policy engine is the collector and developer for access control policies under patient's control.

The original medical data of a patient are collected periodically from repositories of different EHR providers. Then they are encrypted and uploaded to the remote EHR database by the encryptor. To request a set of access medical data of a patient, user is required two types of credentials: *system identity credential* and *access credential*. System credential is exclusive to each user and only used to log into EHR system, but can not access any medical data. While an access credential associates with an access for a set of EHR data. The scope of access depends on the access control policy, the role of user (spatial constraint), and temporal constraint of the access. Consequently, with both valid system identity credential and access credential, users can legally obtain corresponding encrypted data from EHR databases and decrypt them in the legitimate time interval (time bound). We argue that EHR data should be managed with customizable access control along both spatial and temporal dimension.

III. Enforcement of RBTBAC Model

The RBTB key management scheme is the foundation of RBTBAC model. An access key is mainly composed of three parameters: a long-term class key K_i, a time granule value $V_{B(t)}$ and a set of class identities $\{CID_k\}$. This section focuses on computation of the first two parameters. We first discuss the role-based key structures of RBTB key management scheme, and then put forward an efficient method - time tree for computing time granule values.

A. Role-Based Key Structure

A static encryption key structure is firstly constructed for encrypting EHR data (see solid lines in Fig.2). Then a decryption key structure is dynamically established. It aims to minimize the number of keys hold by each user, while users still can produce multiple access keys by themselves effectively.

Suppose that there are totally n doctors and m patients in an EHR system. Consider key tree as Fig.2, the partially ordered set $\langle C, \preceq \rangle$, where the vertices are $C = \{C_0, C_{P_1}, \cdots, C_{P_m}, C_1, C_2, \cdots\}$. In the hierarchical tree, each node denotes a security class. If $C_i \preceq C_j$, we say security class C_i is subordinate to security class C_j. $C_i \prec C_j$ means that $C_i \preceq C_j$ and $C_i \neq C_j$. If $C_i \prec C_j$ and there is no C_k such that $C_i \prec C_k \prec C_j$, we say C_i is immediate subordinate to C_j, which can be denoted by $C_i \prec_d C_j$.

In the encryption key tree, C_0 is the root of the hierarchical tree. Correspondingly, K_0 is the *system root key* used to encrypt the data in class C_0. The nodes in the second level are called *patient nodes*, each of them is a token of patient. All medical data of the patient are nested under the corresponding patient node. Therefore the corresponding key $K_{P_j} (j = 1, 2, \cdots, m)$ is called *patient master key*. The keys in following levels are encryption keys used to encrypt data in homologous classes in hierarchical EHR structure (role-based structure of EHR data [6]). Note that, only the keys in leaf

nodes are used to encrypt real medical data. In summary, the keys in lower level class C_i can be derived from the key in higher level class C_j, whenever there is partial order $C_i \prec C_j$.

Then we dynamically add *doctor node* between root node and patient nodes (see dashed lines in Fig.2), the key $K_{D_i}(i = 1, 2, \cdots, n)$ related to the doctor node is called *doctor master key*, which will be assigned to each doctor. Doctors can produce multiple keys for access different medical records with their master key.

h_1 and h_2 are two one-way hash functions. For encryption key, encryptor computes $K_{p_j} = h_1\left(K_0 \| r_{0,P_j}\right)$, where $r_{0,P_j} = h_2\left(CID_0 \| P_j\right)$, whenever $C_{P_j} \prec_d C_0\,(1 \le j \le m)$. Then encryptor calculates $K_i = h_1\left(K_{P_j} \| r_{P_j,i}\right)$, where $r_{P_j,i} = h_2\left(P_j \| CID_i\right)$, whenever $C_i \prec C_{P_j}$. For privacy of patient, we only publish the relationship of internal nodes with corresponding patient node. As a result, the encrypt key of each node is only in connection with patient node, but has nothing to do with the structure of EHR. In addition, if the structure of EHR is changed, the value $r_{P_j,i}$ is invariable.

For generating decryption key, encryptor first computes doctor master key though equation $K_{D_i} = h_1\left(K_0 \| r_{0,D_i}\right)$, where $r_{0,D_i} = h_2\left(CID_0 \| D_i\right)$. Next, encryptor establishes the connection between doctor nodes and patient nodes. A weight value $a_{i,j}$ is assigned on the edge between doctor node and patient node. Finally encryptor computes relation value $r_{D_i,P_j} = h_2\left(D_i \| P_j\right)$ and a weight value $a_{i,j} = h_1(K_0 \| r_{0,P_j}) / h_1(K_{D_i} \| r_{D_i,P_j})$. So user can correctly derive the key of patient node though formula $K_{P_j} = a_{i,j} h_1(K_{D_i} \| r_{D_i,P_j})$.

B. Time Tree

First of all, several concepts of different time periods involved in RBTBAC scheme need to be distinctly stated. The whole time period is called *timeline*, which is the longest time unit. It is divided into uniform small pieces, each piece slot is called *time granule*, which is the smallest unit of time period. Consecutive time granules is called *time interval* which will be authorized to a user for access a set of classes of medical data.

Timeline is divided into small granules, which are numbered as $0, 1, \cdots, z$, and these time granules are mapped onto a *Complete Binary Tree (CBT)*. Take Fig.3 as an example, the timeline is divided into eight small time granules, which are expressed as binary numbers $000, \cdots, 111$. For simplicity, let $B(t)$ denote the binary expression of time granule and $V_{B(t)}$ indicate the value of time granule t. The values with star in subscript mean internal nodes in the time tree. Obviously, the values of smallest time granule are labeled by leaf nodes in the CBT. Besides, the value of each node in the CBT can be calculated by the path from root node to itself. So we have following expressions, where $\|$ denotes string concatenation.

$V_{0*} = H(H(w) \| 0), V_{1*} = H(H(w) \| 1), \cdots$

$V_{000} = H(H(H(H(w) \| 0) \| 0) \| 0), \cdots$

We observe that all time interval $[t_b, t_e] \in [0, z]$ can be composed of a number of *Full Binary Subtrees (FBSs)*, that

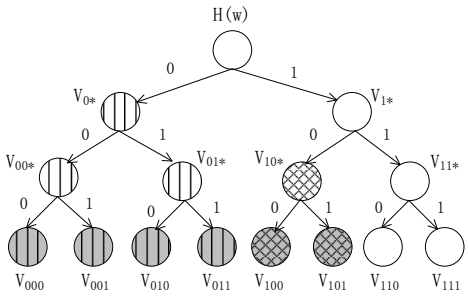


Figure 3. Time binary tree

is $[V_{B(t_b)}, V_{B(t_e)}] = FBS[V^1] \cup \cdots \cup FBS[V^u]$, where V^u denodes the root node of a FBS. For example, time interval $[t_b, t_e] = [000, 101]$ labeled in grey in Fig.3, contains two FBSs, one is rooted by node V_{0*}, the other is rooted by node V_{10*}, so that it can be expressed by equation $[V_{000}, V_{101}] = FBS[V_{0*}] \cup FBS[V_{10*}]$. Therefore, any leaf node of a FBS can be computed by the value of corresponding root node, that is time granules in time interval $[t_b, t_e] = [000, 101]$ can be computed by given values V_{0*} and V_{10*}.

For the efficiency of computing time granule values, time tree is much more efficient than linear hash chain method which is used in the most existing time-bound hierarchical key management schemes.

IV. Implement of RBTBAC Protocol

The access of medical data is an interactive process among user, EHR system and EHR database. As mentioned in Section II, to successfully access the encrypted medical data, the user should be granted both system identity credential and access credential. A RBTBAC protocol is composed of four sub-protocols: *initialization*, *encryption*, *user registration* and *decryption*.

A. Initialization

In this phase, system parameters for access medical records are initialized, patient master keys $K_{P_j}(1 \le j \le m)$ and class keys K_i are generated, and the hierarchical encryption key tree is established as Fig.2.

1) Encryptor chooses a random integer w, a keyed HMAC H_K, where K is the system access master key, and two one-way hash functions h_1 and h_2.
2) Encryptor selects a random value K_0 as system root key which is only known to encryptor, TA and DB, and then computes K_{p_j} and K_i using the method in Section III-A.
3) Encryptor publishes $r_{P_j,k}$ and h_1, and keeps w and h_2 secret, whenever k is a leaf node.

B. Encryption

Encryptor generates encryption key $K_{i,t}$ for $C_i(i = 1, 2, \cdots, N)$ at time granule $t \in [0, z]$ and encrypts the data of class C_i with corresponding key $K_{i,t}$. $K_{i,t}$ is computed by Formula (1), where $V_{B(t)}$ is computed by the method in Section III-B.

$$K_{i,t} = H_K\left(K_i \| V_{B(t)} \| CID_i\right) \tag{1}$$

C. User Registration

When a doctor D_i requests to access the EHR system, TA computes doctor master key K_{D_i} for the eligible doctor D_i and issues a system identity credential containing information $Enc_{PK_{D_i}}(K_{D_i}, H_K)$ to doctor D_i as the identity proof to access the EHR system.

Once the doctor D_i has the system identity credential, he can request to access EHR data in C_k (C_k denotes a class of leaf node) of patient j in consecutive time interval $[t_b, t_e] \in [0, z]$. TA first verifies the doctor's system identity credential. Then the relationship value r_{D_i,P_j} and a weight value $a_{i,j}$ are computed. Next, TA searches the access control policies, if the request matches any access control policy, TA retrieves the set of class identities of leaf nodes and finds the right internal nodes of time tree. At last, TA issues the doctor an access credential, which stores information $Enc_{PK_{D_i}}\left(t_b, t_e, \{V^u\}, \{CID_k\}, r_{D_i,P_j}, a_{i,j}\right)$ and $Enc_{PK_{DB}}(K_{D_i}, t_b, t_e, \{CID_k\})$, with TA's signature, where $\{V^u\}$ is a set of values of FBSs' root nodes, $\{CID_k\}$ is a set of class identities of leaf nodes, PK_{D_i} is the public key of D_i, and PK_{DB} is the public key of DB.

D. Decryption

Suppose a doctor D_i has received an access credential from TA and has granted access time interval $[t_b, t_e]$, doctor D_i can read EHR data in C_k at any time granule $t \in [t_b, t_e]$, if there is partial order $C_k \prec C_{D_i}$.

1) Doctor D_i retrieves r_{D_i,P_j} and $a_{i,j}$ from the access credential with his private key and computes secret key K_{P_j} of patient node C_{P_j}, whenever $C_{P_j} \prec_d C_{D_i}$, where $1 \leq j \leq m$.
2) Doctor D_i retrieves values $r_{P_j,k}$ from public board and computes $K_k = h_1\left(K_{P_j} \| r_{P_j,k}\right)$, whenever $C_k \prec C_{P_j}$, where C_k is a class of leaf node.
3) Doctor D_i requests EHR data with system identity credential and access credential.
4) DB checks authenticity of information $Enc_{PK_{DB}}(K_{D_i}, t_b, t_e, \{CID_k\})$ by verifying TA's signature and decrypts it. Then DB verifies equation $K_{D_i} = h_1\left(K_0 \| h_2\left(CID_0 \| D_i\right)\right)$. If the equation holds, DB sends the encrypted data of class C_k to doctor D_i.
5) In current time granule $t \in [t_b, t_e]$, doctor D_i retrieves values $V^0, \cdots, V^u$ and figures out $V_{B(t)}$. Then doctor D_i computes decryption key $K_{k,t} = H_K\left(K_k \| V_{B(t)} \| CID_k\right)$.
6) At time granule t, the protected data of class C_k can be decrypted by doctor D_i with key $K_{k,t}$.

V. Enforcement Verification

A. Security Analysis

Now we provide the security analysis of our RBTBAC protocol focusing on three types of possible attacks - attacks from the outside, attack on unauthorized class and attacks in unauthorized time interval.

1) Attack from The Outside: An adversary from outside the EHR system will attempt to gain the temporal decryption key. Since the adversary does not have system identity credential, he may try to request encrypted data from DB using a fake access credential with forged values $Enc_{PK_{D_i}}\left(t_b', t_e', \{V^{u'}\}, \{CID_k'\}, r_{D_i,P_j}, a_{i,j}\right)$ and $Enc_{PK_{DB}} = \left(K_{D_i}', t_b', t_e', \{CID_k'\}\right)$. But the adversary will fail to pass the verification, when DB verifies the signature of access credential with TA's public key. In addition, the forged master key will be detected by checking the equation $K_{D_i}' = h_1\left(K_0 \| h_2\left(CID_0 \| D_i'\right)\right)$, since the adversary does not know the system root key K_0, value CID_0 and hash function h_2.

2) Attack on Unauthorized Class: Suppose a doctor wants to derive a decryption key $K_{k',t}$ of class k' which is not authorized to him in consecutive time interval $[t_b, t_e]$. Assume that doctor D_i obtains access credential containing information $Enc_{PK_{D_i}}\left(t_b, t_e, \{V^u\}, \{CID_k\}, r_{D_i,P_j}, a_{i,j}\right)$ and $Enc_{PK_{DB}} = (K_{D_i}, t_b, t_e, \{CID_k\})$, where $C_k \prec C_{P_j}$, $C_{k'} \prec C_{P_j}$ and $k \neq k'$. To request the medical data of class k', doctor D_i must know $CID_{k'}$ and send $Enc_{PK_{DB}} = (K_{D_i}, t_b, t_e, \{CID_{k'}\})$ to DB. The adversary can obtain $CID_{k'}$ by colluding with other doctors or from previous access credential. However, the information $Enc_{PK_{DB}} = (K_{D_i}, t_b, t_e, \{CID_{k'}\})$ must have TA's signature, otherwise, DB will reject the request. Therefore, a doctor cannot forge an access credential to access any data in an unauthorized class even collusion with other doctors.

3) Attack in Unauthorized Time Interval: A doctor D_i may want to derive a decryption key $K_{k,t}$ of class k in unauthorized time interval $[t_b', t_e']$. We assume that in time interval $[t_b, t_e]$ doctor D_i has right to access the data of class k, that is he has access credential containing information $Enc_{PK_{D_i}}\left(t_b, t_e, \{V^u\}, \{CID_k\}, r_{D_i,P_j}, a_{i,j}\right)$ and $Enc_{PK_{DB}} = (K_{D_i}, t_b, t_e, \{CID_k\})$. In time interval $[t_b', t_e']$, where $t_b \leq t_e \leq t_b' \leq t_e'$, doctor D_i tries to access data of class k. To successfully obtain the data from DB, he needs an access credential with information $Enc_{PK_{D_i}}\left(t_b', t_e', \{V^{u'}\}, \{CID_k\}, r_{D_i,P_j}, a_{i,j}\right)$ and $Enc_{PK_{DB}} = (K_{D_i}, t_b', t_e', \{CID_k\})$. The doctor can get t_b', t_e' and $\{V^{u'}\}$ by colluding with other doctor. Again, the access credential sent to DB should with TA's signature to guarantee the authenticity; otherwise DB will reject the request. Thus doctor D_i cannot forge an access credential to access data in any unauthorized time interval $[t_b', t_e']$.

B. Privacy Analysis

Our RBTBAC model provides better privacy for patients, which is reflected in following three aspects.

First, we make use of RBTB key management to control the access time interval of medical records based on the role of doctor in EHR system. Accordingly, the privacy of patient can be protected both from unauthorized data and unauthorized time interval.

Second, encryptor only publishes the relationship values between patient nodes and leaf nodes, so that doctor cannot

TABLE I. EXPERIMENTAL SETUP

Hardware/Software	Components
Processor	Intel(R) Core(TM)2 Duo E7500 2.93GHz
Memory	2.0GB
Operating system	Windows 7 Ultimate
Programming language/Library	C++/Crypt++
IDE	Code::Blocks

know any internal node and path from patient node to leaf node. This can prevent privacy of the patient from revealing to some extent. For example, a doctor wants to access all blood test results of one patient which are nested under different disease category nodes. The doctor is only given the class identities of blood test records, but knows nothing about other class nodes, so that he cannot infer any other disease information from authorized data.

Finally, we suggest to use patient-controlled access policy to better protect patient privacy. Patient must participate in the formulation of access control policy to determine the sharing of his own sensitive information with other users.

VI. SPACE AND TIME COMPLEXITY

The performance analysis of RBTBAC model contains two folds: space and time complexity. RBTB key managemen-t scheme is the most important part of RBTBAC model. Therefore, performance analysis of RBTBAC model can be translated to the analysis on RBTB key management scheme. Here we focus our discussion on the RBTB key management scheme from two aspects: server's perspective and user's perspective.

A. Space Complexity

The space for public information is a main parameter to measure the space complexity of a key management scheme. In a RBTB key management scheme, the space for public information only depends on the number of leaf nodes. Assume that the structure of EHR data has totally N nodes and N_{leaf} leaf nodes. Thus, the space complexity of RBTB key management scheme is $O(N_{leaf})$. Compared with the space complexity of basic time-bound key management scheme $O(N^2 - N)$, our scheme is more space-saving.

B. Time Complexity

Table I contains a summary of hardware and software used in our experiments. The experiments used in this paper were run in Code Blocks. Each experiment was run 20 times and averaged.

Encryption of all data in each time granule is the main time consumer in such time-bound key management scheme. The speed of encryption depends on both the used encryption algorithm and server performance, which areS beyond scope of this paper. In this section, we only discuss the time complexity of computing secret keys. The analysis is respectively given in two folds: computation of encryption key and decryption key.

1) Computation of Encryption Key: In a time-bound key management scheme, generating all encryption keys in each time granule is also time-consuming. However, for server side, the time for generating an encryption key within a time granule mainly depends on the time for computing of time granule values, since the class key is unchanged. The main operation in generation of time granule values is one-way hash evaluation. Thus the efficiency for producing encryption key is principally determined by the times of hash evaluation. From above analysis, the time complexity of hash evaluations in each time granule is less than $\lceil \log_2 (z+1) \rceil$ times with binary tree structure, oppositely with traditional linear hash chain [2] [4] [5], it needs to do $z+1$ times hash evaluations.

2) Computation of Decryption Key: Before issuing an access credential, encryptor has to compute granted time parameters for user. In a linear chain scheme, the given time parameters are $H^{t_b}(a)$ and $H^{z-t_e}(b)$. Therefore, the time complexity of hash evaluation is $O(z-(t_e-t_b))$. While, in our scheme, the time for enryptor generating time parameters includes two parts: the time for locating internal root nodes of FBSs and the time for computing hash values of these root nodes.

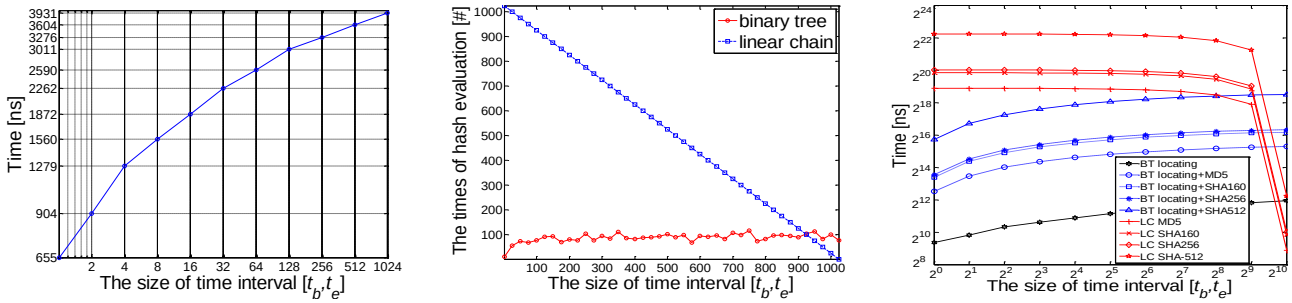
The efficiency of locating root nodes only depends on the scale of time interval $[t_b, t_e]$. The performance is showed in Fig.4(a). For the sake of convenience, we discuss the efficiency based on the condition that the scale of time interval $[t_b, t_e]$ (namely $t_e - t_b + 1$) is $[1, 2^x]$. The average results are displayed in the graph. When the scale of $[t_b, t_e]$ reaches to 2^{10}, it has to spend almost 4000 ns, which is much faster than hash evaluation (see Fig.5(a)).

Given the fixed scale of whole timeline $[0, 1023]$, we illustrate the comparison of the two methods in Fig.4(b) and Fig.4(c). Fig.4(b) shows the comparison of hash evaluations between binary tree method and linear chain method. In most cases, binary tree structure is extremely efficient than linear chain structure. Fig.4(c) demonstrates the total time consumption for encryptor generating time granule parameters. For saving space, binary tree method is abbreviated as BT and linear chain method is abbreviated as LC in the graph. The results are actually the sum of the time for locating root nodes (see Fig.4(a)) and the time for hash operations (see Fig.5(a)). From Fig.4(c) we have conclusion that our scheme is much more efficient than existing time-bound key management schemes.

In user side, the time cost for producing decryption key is also consistent of two parts. The first part is the time cost for generating long-term class key with user's master key. The second part is the time cost for computing time granule value.

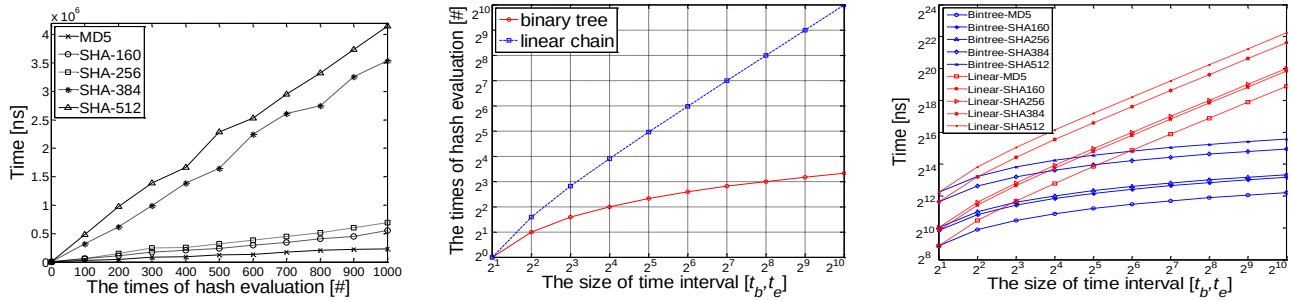
For computing class key, the user only needs to compute hash twice in our scheme. While it needs one or more times hash evaluations with linear chain method.

For calculating time granule value, the time complexity of hash evaluation with linear chain method is $O(t_e - t_b)$. On the contrary, it is only $O(\lfloor \log_2(t_e - t_b + 1) \rfloor)$ with binary tree method. Distinctly, the binary tree structure is efficient than linear train structure as shown in Fig.5(b). The experiments



(a) Time for locating root nodes (b) The number of hash evaluation times (c) The total time

Figure 4. Time for encryptor generating time granule parameters



(a) Time for hash functions (b) The number of hash evaluation times (c) The total time

Figure 5. Time for user computing time granule parameters

results are shown in Fig.5(c). The graphic shapes of the set of binary tree methods are quite flat, while the graphic shapes of the set of linear chain method are steep. Therefore, our scheme is time-saving in the user side.

In summary, our RBTB key management scheme is space-saving, more secure and more efficient than most existing schemes.

VII. Conclusion

We have presented a practical Role Based and Time Bound Access Control (RBTBAC) model for EHR system. Differ from the conventional RBAC model, this access control model integrates the time dimension into the role based access control of sensitive EHR data, enabling patients and healthcare professionals to customize the access control of the EHR data at different spatial and temporal granularities of preference. Technically, we have developed a role based and time bound (RBTB) hierarchical key management scheme with two salient features. First, we have developed a privacy-aware and dynamic key structure for flexible role-based access control. Second, we have employed a time tree method for generating time granule values and for integrating time-bound access control of EHR data with role-based access control using personalized and customizable time windows. We evaluate the RBTBAC model analytically and experimentally. Our results show that RBTBAC offers high-efficiency and flexibility for managing sensitive EHR data and is more customizable for security and privacy protection of EHR data.

Acknowledgment

The first author obtained the initial results of this research as a visiting PhD student at the Distributed Data Intensive Systems (DiSL) lab in the school of Computer Science, Georgia Institute of Technology under the China Scholarship Council. The first three authors would like to thank the partial support of this work by the National Key Basic Research Program (No.2007CB307101), China NSF(No.K09A300150,60973112), China PCSIRT(No.IRT0707), the Discipline Construction and Postgraduate Education Project of Beijing Municipal Education Commission. The last author would like to thank the partial support of this work by the medium size grant from NSF NetSE program and NSF CyberTrust program.

References

[1] W.-G. Tzeng, "A time-bound cryptographic key assignment scheme for access control in a hierarchy," *IEEE Trans. on Knowl. and Data Eng.*, vol. 14, no. 1, pp. 182–188, 2002.

[2] H.-Y. Chien, "Efficient time-bound hierarchical key assignment scheme," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, no. 10, pp. 1301–1304, 2004.

[3] S.-Y. Wang and C.-S. Laih, "Merging: An efficient solution for a time-bound hierarchical key assignment scheme," *IEEE Trans. Dependable Secur. Comput.*, vol. 3, no. 1, p. 91, 2006.

[4] E. Bertino, N. Shang, and S. S. Wagstaff Jr., "An efficient time-bound hierarchical key management scheme for secure broadcasting," *IEEE Trans. Dependable Secur. Comput.*, vol. 5, no. 2, pp. 65–70, 2008.

[5] Y. Sui, F. Maino, Y. Guo, K. Wang, and X. Zou, "An efficient time-bound access control scheme for dynamic access hierarchy," in *Proceedings of the 2009 Fifth International Conference on Mobile Ad-hoc and Sensor Networks*, ser. MSN '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 279–286.

[6] R. Zhang and L. Liu, "Security models and requirements for healthcare application clouds," in *CLOUD '10: Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 268–275.