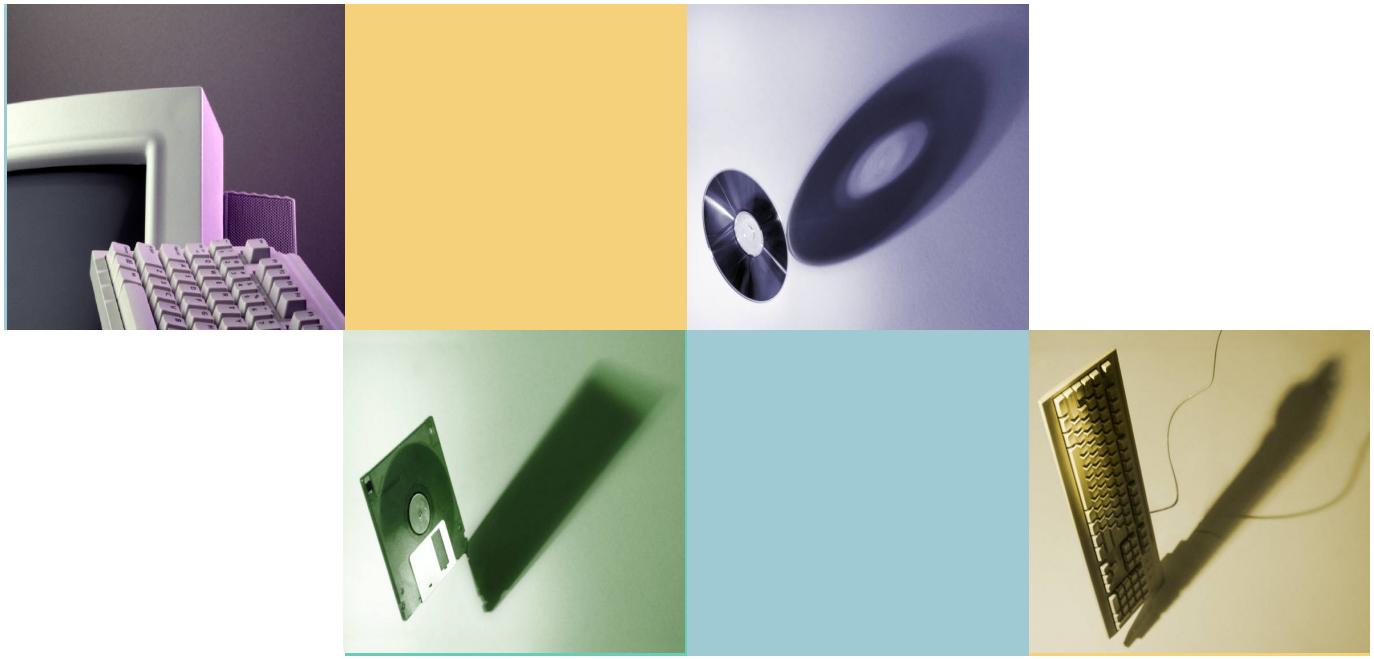
HCBE: Achieving Fine-Grained Access Control in Cloud-based PHR Systems

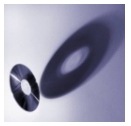


Xuhui Liu[1], Qin Liu[1], Tao Peng[2] , and Jie Wu[3]

[1] Hunan University, China
[2] Central South University, China
[3] Temple University, USA

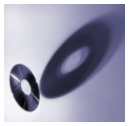
Outline

1. Abstract & Introduction
2. Preliminaries
3. Scheme description
4. Experimental Results
5. Conclusion


Abstract & Introduction

- Personal Health Record(PHR)
- Comparison-based Encryption(CBE)
- Contributions
- Application scenario

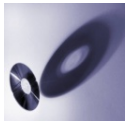

Abstract & Introduction

What is Personal Health Record ?

Personal health record (PHR) is a patient-centric model of health information exchange. It has become popular with more and more users due to its convenience to access a patient's centralized profile. PHR allows medical practitioners to online access a complete and accurate summary of a patient's medical history.

What is Comparison-based encryption ?

Comparison-based encryption (CBE) was the first to realize time comparison in attribute-based access policy by means of the forward/backward derivation functions.


Abstract & Introduction

For example, suppose that the access policy of the ciphertext is *At* $\wedge$ *[tx, ty], and the authorization time of* the user with attribute *At is [ta, tb]. Then, the user can decrypt the ciphertext* only when the current time (*tc* $\in$ *[tx, ty])* $\wedge$ *(tc* $\in$ *[ta, tb]).*

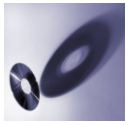
The drawback of CBE

The main drawback of CBE is that the encryption cost grows linearly with the number of attributes in the access policy. For a system of a large number of attributes, the cost for encryption may be extensive.

Abstract & Introduction

Solutions

To efficiently realize a fine-grained access control for PHRs in clouds, we propose a hierarchical comparison-based encryption (HCBE)scheme by incorporating an attribute hierarchy into CBE. The main idea of the HCBE scheme is building a hierarchical structure for attributes, where the attribute at a higher level is a generalization of the attributes at lower levels.

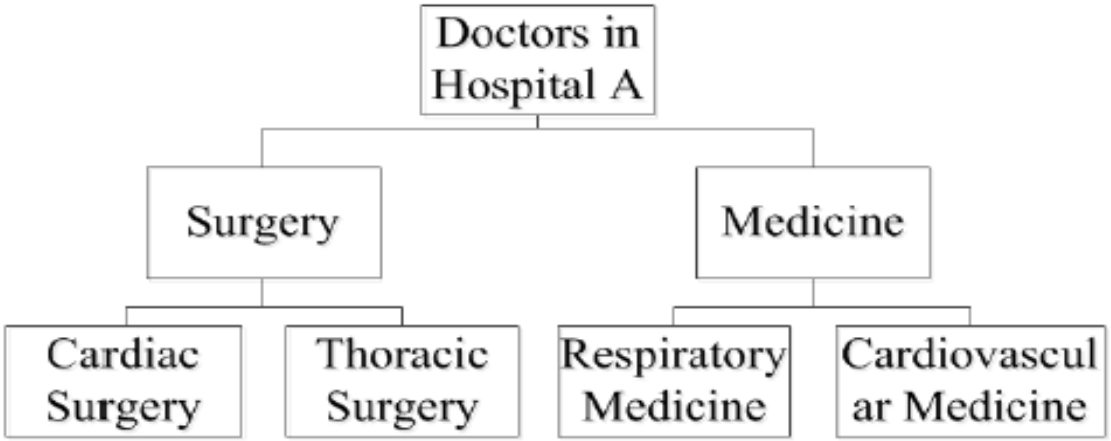




Abstract & Introduction

Sample Application scenario

(*Cardiac Surgery* OR *Thoracic Surgery*) OR
{(*Respiratory Medicine* OR *Cardiovascular Medicine*)
AND (*2015/4/1* ⩽ *Period-of-Validity* ⩽ *2015/4/30*)}

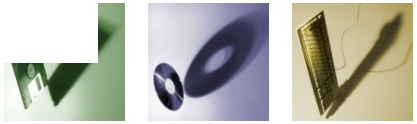
(a) Access policy of Alice's PHR.



(b) Attribute Hierarchy.

(*Surgery*) *OR*
{ (*Medicine*)AND(*2015/4/1* ⩽*Period-of-Validity* ⩽ *2015/4/30*) }

(c) Simplified access policy.



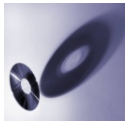
Abstract & Introduction

Contributions

- 1. We proposed a hierarchical comparison-based encryption (HCBE) scheme,by incorporating attribute hierarchy into CBE, so as to efficiently achieve a fine-grained access control in cloud-based PHR systems.
- 2. We constructed an attribute hierarchy tree, and encode each attribute node with the PNDF coding. By applying the backward derivation function, the users with the specific attributes can decrypt the ciphertext encrypted with the generalized attributes.
- 3. We analyze the security of the proposed scheme, and conduct experiments to validate its effectiveness and efficiency.

Preliminaries

- System model
- BDF & FDF in CBE scheme


System model

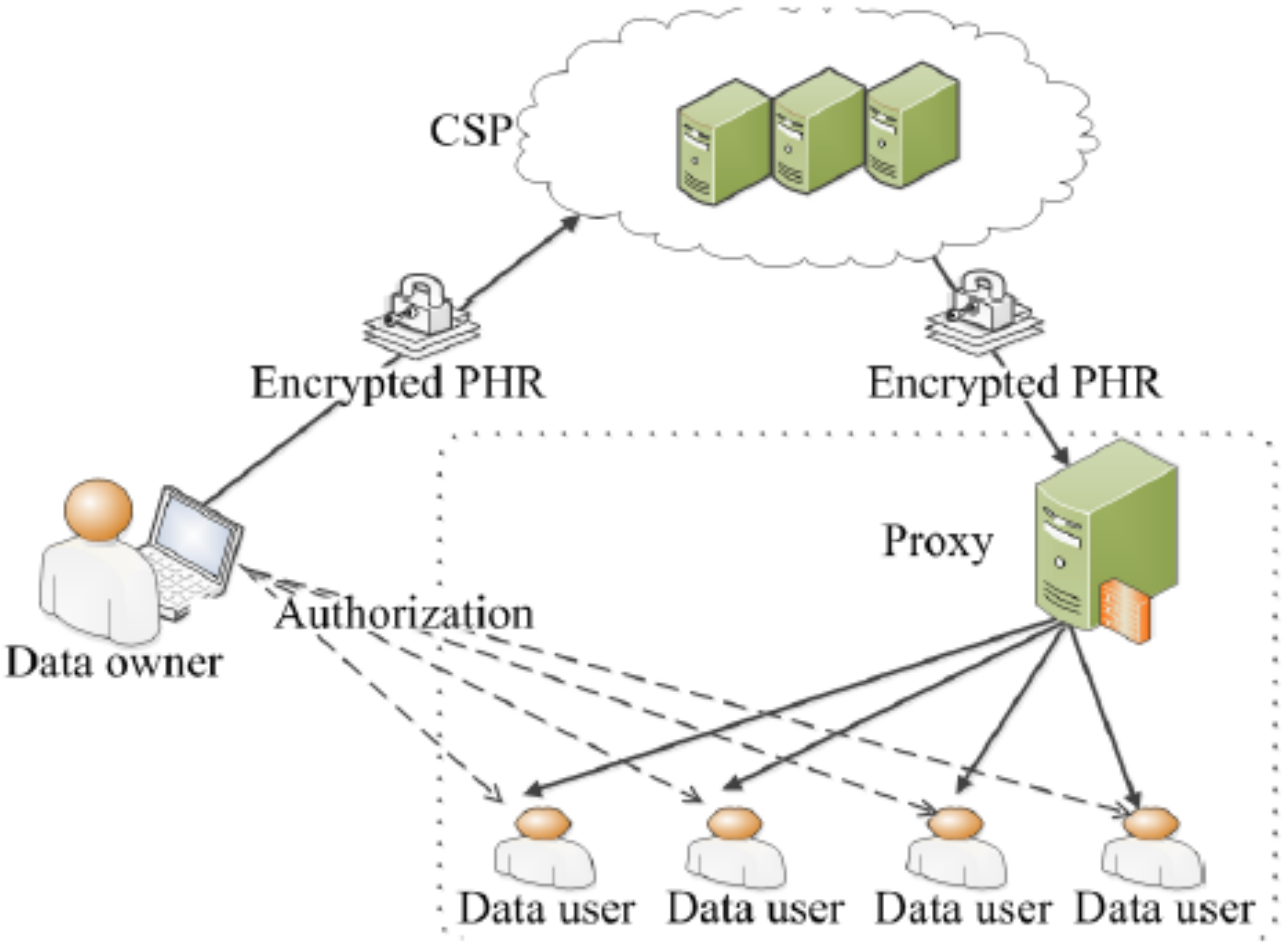


Fig. 2. System model.

System model

The system consists of the following parties: the cloud service provider (CSP),the data owner, and the data users. The CSP operates the cloud-based PHR system, which locates on a large number of interconnected cloud servers with abundant hardware resources. The data owner is the individual patient who employs the cloud-based PHR system to manage her PHR. The data users are the entities who is authorized by the data owner to access the cloud-based PHR system.

FDF & BDF in CBE scheme

CBE scheme utilizes "one-way" property to represent the total ordering relation of integers. This means that given the integer relation $t_i \leq t_j$ and two corresponding values v_{ti}, v_{tj}, there exists an efficient algorithm to obtain v_{ti} from v_{tj}, however, it is hard to compute v_{tj} from v_{ti}.

Firstly, we define two mapping functions $(\psi(\cdot), \bar{\psi}(\cdot))$ from an integer set $U = \{t_1, \cdots, t_T\}$ into $V = \{v_{t1}, \cdots, v_{tT}\}$ and $\bar{V} = \{v_{t1}, \cdots, v_{tT}\}$ as follows:

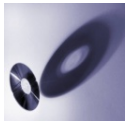
$$v_{t_i} \leftarrow \psi(t_i) = \varphi^{\lambda^{t_i}} \in \mathbb{G}_{n'},$$
$$\bar{v}_{t_i} \leftarrow \bar{\psi}(t_i) = \bar{\varphi}^{\mu^{Z-t_i}} \in \mathbb{G}_{n'}.$$

FDF & BDF in CBE scheme

Next, according to the definition of $\psi(\cdot)$ and $\overline{\psi}(\cdot)$, it is easy to define the FDF $f(\cdot)$ and BDF $\overline{f}(\cdot)$ as:

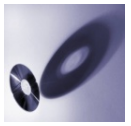
$$v_{t_j} \leftarrow f_{t_i \le t_j}(v_{t_i}) = (v_{t_i})^{\lambda^{t_j - t_i}} \in \mathbb{G}_{n'},$$

$$\bar{v}_{t_j} \leftarrow \bar{f}_{t_i \ge t_j}(\bar{v}_{t_i}) = (\bar{v}_{t_i})^{\mu^{t_i - t_j}} \in \mathbb{G}_{n'}.$$

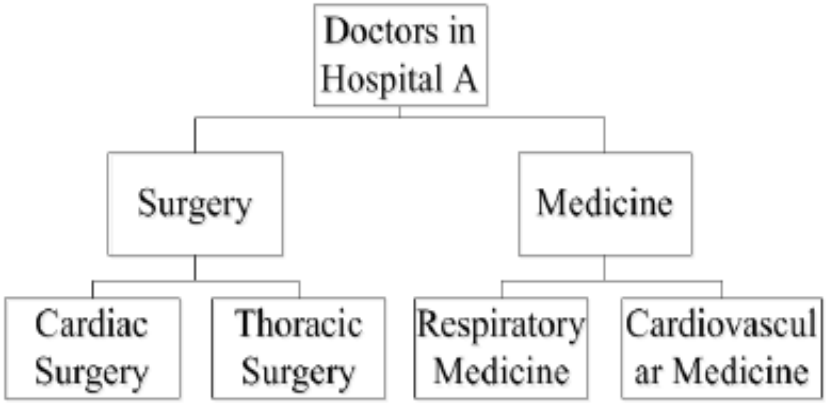



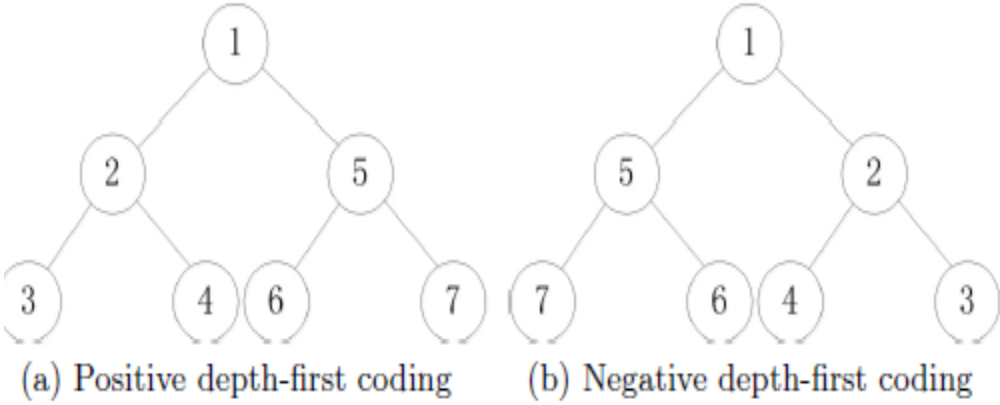

Scheme description

- Positive-Negative Depth-First Coding
- Basic scheme
- Our Construction


Positive-Negative Depth-First Coding





Take the attribute tree is shown in left as an example. The PNDF coding is shown in right. Let *Pcodei and* Ncodei *denote the Pcode* and Ncode of node i, respectively. The PNDF coding has the property that Pcodei > Pcodej and Ncodei > Ncodej , if i is the descendant node of j.

Basic scheme

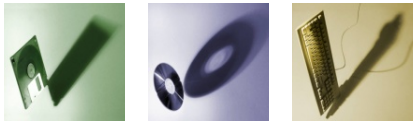
We apply the BDF to accomplish the attribute hierarchy. The denition of the HCBE scheme is as follows:

Suppose that the number of nodes in the attribute hierarchy is m. In HCBE, the hierarchical codes are denoted as a set of discrete values $U_m = \{(Pcode_1, Ncode_1), (Pcode_2, Ncode_2), \ldots, (Pcode_k, Ncode_k), \ldots, (Pcode_m, Ncode_m)\}$, with total ordering $0 \leq Pcode_1 \leq Pcode_2 \leq \ldots, \leq Pcode_m \leq Z_m$ and $0 \leq Ncode_1 \leq Ncode_2 \leq \ldots, \leq Ncode_m \leq Z_m$, where Z_m is the maximal integer.

Next, we define mapping functions *ψ1(.), ψ2(.) is as follows:*

$$v_{Pcode_k} = \varphi_1^{\theta_1^{Z_m - Pcode_k}}$$

$$v_{Ncode_k} = \varphi_2^{\theta_2^{Z_m - Ncode_k}}$$

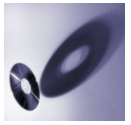


Basic scheme

According to the definitions of *ψ1(.), ψ2(.), it is easy to define BDFs f1(.), f2(.)* as follows:

$$v_{Pcode_j} \leftarrow f_1(v_{Pcode_k}) = (v_{Pcode_k})^{\theta_1{}^{Pcode_k - Pcode_j}}, \quad Pcode_k \geq Pcode_j$$

$$v_{Ncode_j} \leftarrow f_2(v_{Ncode_k}) = (v_{Ncode_k})^{\theta_2{}^{Ncode_k - Ncode_j}}, \quad Ncode_k \geq Ncode_j$$


Our construction

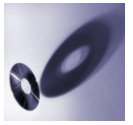
The definition of the HCBE scheme consists of the following algorithms:

- $Setup(1^{\kappa}, \widehat{\mathcal{A}}) \rightarrow (MK, PK_{\widehat{\mathcal{A}}})$: The data owner takes a security parameter κ and the attribute hierarchy $\widehat{\mathcal{A}}$ as inputs, and outputs the master key MK and the system public key $PK_{\widehat{\mathcal{A}}}$;
- $GenKey(MK, uk, \widehat{\mathcal{L}}) \rightarrow SK_{\widehat{\mathcal{L}}}$: The data owner utilizes her master key MK to generate a private key $SK_{\widehat{\mathcal{L}}}$ on an access privilege $\widehat{\mathcal{L}}$ for user uk, where each attribute $A_k \in \widehat{\mathcal{L}}$, denoted as $A_k(t_a, t_b, Pcode_k, Ncode_k)$, is associated with the authorization time $[t_a, t_b]$ and hierarchy codes $\{Pcode_k, Ncode_k\}$.
- $Encrypt(PK_{\widehat{\mathcal{A}}}, \widehat{AP}) \rightarrow (\widehat{\mathcal{H}_{\mathcal{P}}}, ek)$: The data owner takes the public key $PK_{\widehat{\mathcal{A}}}$ and an access policy $\widehat{AP}$ as inputs to generate a session key ek and a ciphertext header $\widehat{\mathcal{H}_{\mathcal{P}}}$, where each attribute $A_l \in \widehat{AP}$, denoted as $A_l(t_i, t_j, Pcode_l, Ncode_l)$, is associated with the time condition $[t_i, t_j]$ and hierarchy codes $\{Pcode_l, Ncode_l\}$.
- $Delegate(SK_{\widehat{\mathcal{L}}}, \widehat{\mathcal{L}'}) \rightarrow SK_{\widehat{\mathcal{L}'}}$: The data user takes the private key $SK_{\widehat{\mathcal{L}}}$ and an access privilege $\widehat{\mathcal{L}'}$ as inputs to generate a derived private key $SK_{\widehat{\mathcal{L}'}}$ for the proxy server if $\widehat{\mathcal{L}'} \preceq \widehat{\mathcal{L}}$ [2].
- $Decrypt1(SK_{\widehat{\mathcal{L}'}}, \widehat{\mathcal{H}_{\mathcal{P}}}) \rightarrow \widehat{\mathcal{H}'_{\mathcal{P}}}$: The proxy server takes the derived private key $SK_{\widehat{\mathcal{L}'}}$ and a ciphertext header $\widehat{\mathcal{H}_{\mathcal{P}}}$ as inputs, and outputs a new ciphertext header $\widehat{\mathcal{H}'_{\mathcal{P}}}$ if $\widehat{\mathcal{L}'}$ satisfies $\widehat{AP}$.
- $Decrypt2(SK_{\widehat{\mathcal{L}}}, \widehat{\mathcal{H}'_{\mathcal{P}}}) \rightarrow ek$: The data user takes the private key $SK_{\widehat{\mathcal{L}}}$ and the new ciphertext header $\widehat{\mathcal{H}'_{\mathcal{P}}}$ as inputs, and outputs a session key ek, which can be used to decrypt the stored data.

Experimental Results

In this section, we will compare our proposed scheme with the CBE scheme interms of computation cost. Our experiments are conducted with Java programming language. We implement our scheme on an stand-alone mode, on a PC with Intel Core i3 CPU running at 2.3GHz and 2G memory.

The parameter settings in the experiments are as follows : NA is the number of specific attributes in the access policy, m indicates the number of attribute nodes in an attribute hierarchy tree. Here, we take m = 50,NA = 10 and m = 100, respectively.


Experimental Results

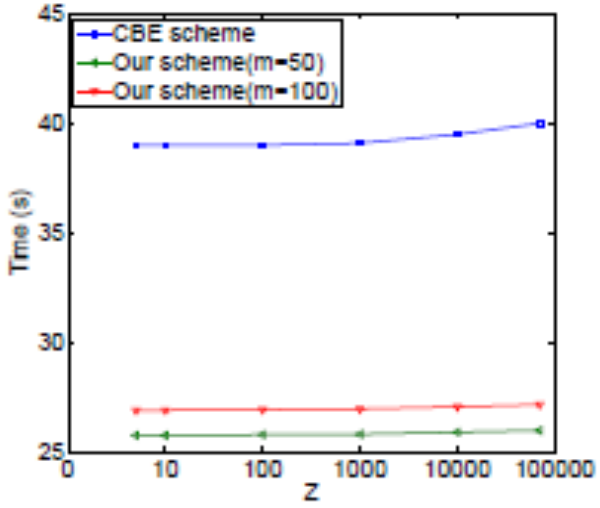
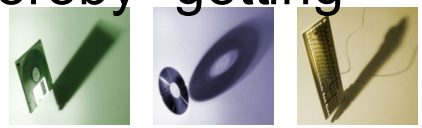


Fig. 5. Computation cost of *Encrypt*.

As shown in Fig. 5, the encryption time of our scheme is much smaller than that of the CBE scheme. Furthermore, with the decrease of the number of attribute, m, in access policy, our scheme has better performance. Therefore, in our scheme, the data owner's time overhead will be reduced, thereby getting better service experience.



Conclusion

In this paper, we proposed a HCBE scheme for achieving a fine-grained access control in cloud-based PHR systems. Our scheme supports time comparison in attribute-based encryption in an efficient way, by incorporating attribute hierarchy into CBE.

Conclusion

Thank you!

Q&A