Privacy Enhanced Access Control by Means of Policy Blinding

Saeed Sedghi, Pieter Hartel, Willem Jonker, and Svetla Nikova

University of Twente, The Netherlands

Abstract. Traditional techniques of enforcing an access control policy rely on an *honest* reference monitor to enforce the policy. However, for applications where the resources are sensitive, the access control policy might also be sensitive. As a result, an *honest-but-curious* reference monitor would glean some interesting information from the requests that it processes. For example if a requestor in a role psychiatrist is granted access to a document, the patient associated with that document probably has a psychiatric problem. The patient would consider this sensitive information, and she might prefer the honest-but-curious reference monitor to remain oblivious of her mental problem.

We present a high level framework for querying and enforcing a role based access control policy that identifies where sensitive information might be disclosed. We then propose a construction which enforces a role based access control policy cryptographically, in such a way that the reference monitor learns as little as possible about the policy. (The reference monitor only learns something from repeated queries). We prove the security of our scheme showing that it works in theory, but that it has a practical drawback. However, the practical drawback is common to all cryptographically enforced access policy schemes. We identify several approaches to mitigate the drawback and conclude by arguing that there is an underlying fundamental problem that cannot be solved. We also show why attribute based encryption techniques do not not solve the problem of enforcing policy by an honest but curious reference monitor.

1 Introduction

The standard implementation of an access control system has a reference monitor to enforce the policy. The reference monitor has to satisfy two strong assumptions. Firstly, the reference monitor is assumed to be honest in the sense that it faithfully enforces the policy. Secondly, the reference monitor is assumed *not* to be curious, in the sense that it does not leak information on the policy. We believe that the second requirement is unnecessarily strong.

Normally, the reference monitor would be considered part of the trusted computing base, so that even if it does learn interesting facts about users, this should not be a concern. However, in a distributed system, the reference monitor is not a simple component, so we should be reluctant to trust it fully. This is merely an instance of the principle of the least privilege, which applied here states that

F. Bao and J. Weng (Eds.): ISPEC 2011, LNCS 6672, pp. 108–122, 2011.

the reference monitor should do its job with as little privileges as necessary. To achieve this, we propose a method to enforce access control policies based on a weaker assumption than normal with the *honest-but-curious* reference monitor.

Consider the following Role Based Access Control (RBAC) scenario to illustrate the urgency of the problem. Imagine that users (e.g. Alice) store their personal health records (PHR) on a PHR server that is believed to be honest but curious. Examples of such PHR servers include Google Health, and Microsoft Health Vault. To protect her privacy, Alice stores her PHR in encrypted form, such that using a searchable encryption scheme the encrypted records can be retrieved selectively with the minimum leakage of information [13,8,9,27,25,14,1,6,10,16]. The honesty of the PHR system ensures that Alice will receive an honest answer to her queries. Now, assume that Alice is undergoing treatment for a mental problem, and that she wants to store the medical records of her mental problem on the PHR server in a way that only a psychiatrist is permitted to access. Then, an access control policy is required which restricts the access of users who do not have the role psychiatrist. However, as the access control policy will mention the role psychiatrist, the reference monitor of the PHR server learns that Alice might have a mental problem. The problem is then that Alice must assume that the curiosity of the PHR system will cause it to leak that she may have a mental problem.

Contribution. First we propose a unifying framework for searching *and* enforcing policy by an honest-but-curious server. Second, we propose a scheme that permits the reference monitor on an honest-but-curious server to enforce an access control policy without learning much. We will make more precise later what "much" actually means. We then describe practical issues for the security of cryptographically enforced access control policy schemes.

The rest of the paper is organized as follows. Section 2 presents a summary of related work showing why prominent cryptographic schemes such as attribute based encryption (ABE) techniques do not provide a suitable solution for our problem. Section 3 describes the problem and the proposed solution from a high level point of view. Section 4 shows how existing schemes for searching in encrypted data fit into the general framework. Section 5 describes the details of a simple scheme to enforce a blinded policy under the assumption that each user only has one role. We prove the security of the scheme. The last section concludes.

2 Related Work

Our work is built on searchable encryption and attribute based encryption techniques. We discuss the most important related work in each of these fields.

Searchable encryption was first proposed by Song, Wagner, and Perrig [26]. The SWP scheme allows a client to store her documents on a database in encrypted form while the client can query her encrypted documents. Searchable encryption in symmetric settings, where the encryption and construction of the query is performed by the same key, was developed further by several authors [13,8,9,27,25,14]. Boneh et. al. [3], were the first who proposed searchable

encryption in the public key setting (PEKS), where a client constructs a query by her private key to search on documents encrypted by her public key. Abdalla et. al. [1] discuss the PEKS scheme further. The idea of searchable encryption in public key settings was extended to a multi-user setting, where a group of users can store documents on the server in encrypted form while other users of the group can search the entire database for a query [9]. The drawback of the latter work is that users must all share the same key. To alleviate this drawback Dong et al. [10], Bao et al. [28], and Ho et al. [16] propose schemes which allow each user of the group to encrypt documents by a unique key. Although searchable encryption allows for secure storage and retrieval of data, access control is not provided. For this other means are necessary, in particular Attribute Based Encryption seems expedient.

Hidden credentials (HC) and Attribute based encryption (ABE) techniques allow a party to specify a policy upon encrypting a message, such that a requestor will be able to decrypt the message if and only if his attributes satisfy the policy associated with the message. HC schemes [12,7,20], which are proposed before introducing ABE, do not resist against collusion attacks [2]. A scheme is resistant against collusion attacks if it does not allow multiple users who do not have the credential of a role to gain access the role by colluding. Sahai and Waters were the first to propose ABE [22], which in turn is based on identity based encryption (IBE) [4]. In ABE actually the identity of the receiver is viewed as a set of descriptive attributes such that users possessing the specified attributes can decrypt a message. Goyal et al. develop key policy ABE (KP-ABE) [15], where the policy is associated with the decryption key and the message is decrypted if the ciphertext contains sufficient attributes. Bethencourt et al. propose ciphertext policy ABE(CP-ABE) [2] where instead of associating the policy with the decryption key, the former is associated with the ciphertext and the attributes of the user are associated with the decryption key.

Predicate encryption (PE) is a class of attribute based encryption techniques in where the ciphertext hides the policy or the attributes. Anonymous IBE is the simplest predicate encryption, where the attributes are the IDs of users and the policy is equality between the ID of the ciphertext and the ID of the decryption key. Katz et al. proposed the most expressive predicate encryption scheme [19]. In this scheme the message is decrypted if the inner product of the attributes of the decryption key and the ciphertext is zero. PE schemes which allow for the decryption key to contain don't care or wildcard values are called hidden vector encryption (HVE). HVE was first proposed by Boneh and Waters [6]. This technique which is also known as hidden vector encryption was developed [17,5,21,24].

Why do ABE and PE not solve the problem of enforcing policy by an honest but curious reference monitor? To leverage the idea of ABE or PE for enforcing an access policy cryptographically, each user should have a decryption key which is generated according to his roles. Documents are also encrypted with their associated roles using a PE scheme. the reason why ABE is not appropriate for this situation is that ABE schemes do not hide the policy and therefore reveal the

associated roles upon the storage. For a user to access a document, first a query is sent to the server which searches the encrypted documents for ones that satisfy the query (documents whose associated keywords contain the queried keyword). The server then sends those encrypted documents to the user who checks for each document whether it is decryptable by his decryption key. This approach has two drawbacks: firstly, extra communication overhead and computational complexity at the user side is required since users retrieve documents that are not decryptable. Secondly, after receiving the encrypted documents from the server, the user learns that all those documents contain the query even if some of them are not decryptable by her decryption key. This shows that PE schemes generate overhead and a leakage on documents. Therefore the policy should be enforced by the server. To enforce the policy by the server, the decryption key of the user should be sent to the server. Although PE schemes hide the roles, the latter remains hidden as long as the server does not have the decryption key that matches with the policy of the ciphertext. Therefore, PE schemes cannot be used to enforce the policy by the server. This is the purpose of our paper to propose a scheme which supports enforcing the policy by the server such that the former does not reveal any information about the documents. To the best of our knowledge, there is no related work on the closely related problem of enforcing access control using an honest but curious reference monitor.

3 Blinded Server

A blinded data base (1) supports data base queries and (2) enforces access control policies on (3) an honest-but-curious server. The query and the enforcement should leak as little information as possible to the server, which we propose to achieve by making the server blind to all sensitive information. In this section we provide a high level specification of such a blinded server, which will be refined in subsequent sections. The high level specification of the blinded server provides a framework in which all seminal related work on searching in encrypted can be fit. The specification also shows that there is uncharted terrain in which blinded policy enforcement would fit. We have not been able to find relevant related work, and therefore propose a first scheme on blinded policy enforcement that fits into this uncharted terrain.

Consider a finite set of roles R, documents D, document identifiers I, and keywords W. If the server, which consists of a database and a reference monitor, is honest and *not* curious, then no blinding is necessary. In the next subsection we will show as a base line how the data base can be queried and how access control can be enforced without blinding.

3.1 Unblinded Database and Reference Monitor

The following set of functions determines the interface to an unblinded search and access control scheme:

- The unblinded data base consists of a bijection id and a function iw, and the policy consists of a function ir:
  - $\mathsf{id} : I \to D$ is a bijection that looks up the document for a given document identifier.
  - $\mathsf{iw} : I \to \mathcal{P}(W)$ associates a set of keywords with each document identifier, that can be used to query the data base.
  - $\mathsf{ir} : I \to \mathcal{P}(R)$ represents the unblinded access control policy. This function associates each document identifier with a set of authorized roles.
- When a user in a particular role $r \in R$ queries the unblinded data base for a certain keyword $w \in W$, the query function rw returns all relevant documents to which the user has access as determined by the unblinded policy.

$$\mathsf{rw} : (R \times W) \to (I \times \mathcal{P}(D))$$
$$\mathsf{rw}(r, w) = \{(i, d) \mid i \in I \wedge r \in \mathsf{ir}(i) \wedge w \in \mathsf{iw}(i) \wedge d = \mathsf{id}(i)\}$$

  The term $r \in \mathsf{ir}(i)$ enforces the unblinded RBAC policy.

3.2 Blinded Database and Reference Monitor

To search a curious database we extend the query with a trapdoor, which is basically some extra data that allows the query to be executed, while at the same time leaking as little as possible about the keyword to be queried. The literature provides a wide selection of trapdoor constructions. In the same vein, to enforce access control with a curious reference monitor, we use some extra data in the form of a blinded credential, which can be used to check access rights of a blinded role. To the best of our knowledge, using blinded credentials for this purpose is new.

Consider a finite set of secret keys K, trapdoors T, blinded documents $\overline{D}$, blinded keywords $\overline{W}$, blinded roles $\overline{R}$, credentials C, and blinded credentials $\overline{C}$. The following set of functions determines the interface for a blind query and access control scheme:

- The user is assumed to be able to blind and unblind a document using a pair of functions as follows, where the keys K are appropriately chosen and kept secret by the user:
  - $\mathsf{kd} : (K \times D) \to \overline{D}$
  - $\mathsf{kd}^{-1} : (K \times \overline{D}) \to D$
- The user is assumed to be able to blind a keyword and to generate a trapdoor that can be used to search for the keyword using a function:
  - $\mathsf{kw} : (K \times W) \to (T \times \overline{W})$
- The user is assumed to be able to blind a role and to generate a blinded credential that can be used to enforce the RBAC policy using a function:
  - $\mathsf{kr} : (K \times R) \to (\overline{C} \times \overline{R})$
- The blinded data base consists of a bijection $\overline{\mathsf{id}}$ and a function $\overline{\mathsf{iw}}$, and the blinded policy consists of a function $\overline{\mathsf{ir}}$ as follows:

- $\overline{\mathsf{id}} : I \to \overline{D}$ looks up the blinded document for a given document identifier. It must not be possible for the document identifier to leak information on the document.
- $\overline{\mathsf{iw}} : I \to \mathcal{P}(\overline{W})$ associates a set of blinded keywords with each document identifier.
- $\overline{\mathsf{ir}} : I \to \mathcal{P}(\overline{R})$ represents the blinded access control policy. This function associates each document identifier with a set of blinded roles.

– If a user in a particular role $r \in R$, and with a particular key $k \in K$ queries the blinded data base for a certain keyword $w \in W$, the query function $\overline{\mathsf{krw}}$ returns all relevant documents to which the user has access as determined by the blinded policy:

$$\overline{\mathsf{krw}} : (K \times R \times W) \to \mathcal{P}(I \times D)$$

$$\overline{\mathsf{krw}}(k, r, w) = \{(i, \mathsf{kd}^{-1}(k, \overline{d})) \mid \boxed{i \in I \wedge \overline{r} \in_{\overline{c}} \overline{\mathsf{ir}}(i) \wedge \overline{w} \in_{t} \overline{\mathsf{iw}}(i) \wedge \overline{d} = \overline{\mathsf{id}}(i)}\}$$

$$\text{where } (\overline{c}, \overline{r}) = \mathsf{kr}(k, r) \text{ and } (t, \overline{w}) = \mathsf{kw}(k, w)$$

The two terms after the where clause calculate the blinded credential $\overline{c}$ for the role and the trapdoor t for the blinded keyword. The set membership operations are adorned with a subscript indicating which credential/trapdoor to use when comparing keywords or roles. The term in the box is calculated on the server, the remaining calculations are performed on the client, thus showing that the server never sees unblinded roles, keywords or documents, nor any keys.

The set of functions $\overline{\mathsf{id}}$, $\overline{\mathsf{iw}}$, and $\overline{\mathsf{ir}}$ will be used throughout the paper to represent the data base and the policy.

4 Blinding the Database

Section 3.2 describes how a blinded server consists of a blinded database and a blinded reference monitor. In this section, we review the state of the art of searchable encryption.

Since the refinement focuses on the management of keys we need two new sets to represent details about keys. Firstly, since the strength of the keys is usually determined by a security parameter, we introduce a set S to draw the security parameter from. Secondly, it will be useful to be able to blind a key, for which we introduce a set $\overline{K}$.

A multi-user searchable encryption scheme consists of the following functions:

– $\mathsf{UserKey} : S \to K$, which when given a security parameter $s \in S$, generates a secret key $k \in K$.
– Blinding consists of the following sub-functions:
  - $\mathsf{DocKeyGen} : S \to (K \times K)$, Which when given a security parameter $s \in S$ outputs a pair of encryption keys $(k_d, k_w) \in (K \times K)$.
  - $\mathsf{DocBlinding} : (K \times D) \to \overline{D}$, which when given a key $k_d \in K$ and a document $d \in D$ delivers a blinded document $\overline{d} \in \overline{D}$.

Table 1. Relation between the high level functions and their refinements for blinding a database

High level	kd	kd^{-1}	kw	$\overline{\mathsf{krw}}$
Refinement	DocKeyGen	UserKeyGen	KeywordBlinding	
Blinded Database	DocBlinding	KeyUnblinding	DocKeyGen	Search
		DocUnblinding	Trapdoor	

  - KeywordBlinding : $(K \times W) \rightarrow \overline{W}$, which when given a key $k_w \in K$ and a keyword $w \in W$ delivers a blinded keyword $\overline{w} \in \overline{W}$.
  - KeyBlinding : $(K \times K) \rightarrow \overline{K}$, which when given a key pair $(k_d, k_u) \in (K \times K)$ transforms k_d (normally associated with a document) to a blinded form $\overline{k_d}$ using k_u (normally associated with a user), such that k_d can be retrieved by holders of k_u.
- Trapdoor : $(K \times W) \rightarrow T$, which when given a key $k \in K$ and a keyword $w \in W$ delivers a trapdoor $t_w \in T$.
- Search : $(\overline{W} \times T) \rightarrow \mathcal{P}(I \times D)$, which when given a blinded keyword $\overline{w} \in \overline{W}$ and a corresponding trapdoor $t_w \in T$ searches the data base for a match.
- Retrieval consists of two sub-functions:
  - KeyUnblinding : $(K \times \overline{K}) \rightarrow K$, which when given a key $k \in K$ and the corresponding blinded key $\overline{k_d} \in \overline{K}$ recovers the unblinded key $k_d \in K$.
  - DocUnblinding : $(K \times \overline{D}) \rightarrow D$, which when given a document $\overline{d} \in D$ and the corresponding blinding key $k \in K$ yields an unblinded document $d \in D$.

We now indicate the relation of the refinements explained above to the high level functions explained in 3.2 to blind the database. The relation is illustrated in table. 1. We describe the refinements of the functions which blind the reference monitor in section 5.

5 Construction

We now present a refinement of the general idea of policy blinding as provided in Section 3.2, which shows how to blind the roles and credentials, and how to perform access decisions by using blinded credential. This time there is no related work to draw on and the refinement will therefore consist of our contribution to blinded policy enforcement.

The refinement consists of introducing a master key and functions to generate a blinded credential for each user, to generate a secret key for each user, to transform a role to blinded form, and to make a decision about a request. However, before we present the scheme we describe its security requirement.

Decisional Bilinear Diffie-Hellman assumption. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative groups of order q. Let g be the generator of group $\mathbb{G}_1$. The DBDH problem states that given the tuple (g^a, g^b, g^c, Z) for random $a, b, c \in \mathbb{Z}_q$ it is hard to decide whether $Z = e(g, g)^{abc}$ or Z is a random picked from $\mathbb{G}_2$, where

$e(.,.)$ is a bilinear map. Let s be a security parameter that determines the size of the group $\mathbb{G}_1$ (i.e. q is a prime s bit string).

The DBDH advantage $Adv_{\mathcal{A}}^{DBDH}(\mathbf{s})$ of a probabilistic polynomial-time algorithm $\mathcal{A}$ is defined as follows:

$$Adv_{\mathcal{A}}^{DBDH}(s) = |Pr[\mathcal{A}(g^a, g^b, g^c, e(g,g)^{abc}) = 1] - Pr[\mathcal{A}(g^a, g^b, g^c, Z \in_R \mathbb{G}_2) = 1]|$$

Security Definition. Informally the security of the blinding policy scheme states that blind roles do not reveal any non-trivial information to an adversary who obtains the blind credential of other roles. A blinding policy scheme is semantically secure if for all probabilistic polynomial time adversaries $\mathcal{A}$, $|Pr[EXP_{\mathcal{A}} = 1] - \frac{1}{2}| < \varepsilon(s)$, for some negligible $\varepsilon(s)$ function, where $EXP_{\mathcal{A}}$ is the following experiment between a challenger and adversary $\mathcal{A}$:

- **Setup:** The challenger sends the public parameters of the role blinding scheme to $\mathcal{A}$.
- **Query phase I:** In this phase $\mathcal{A}$ adaptively queries for the blinded credential $\overline{C}_r$ of a role r.
- **Challenge:** Once $\mathcal{A}$ decides that the query phase is over, $\mathcal{A}$ picks two challenge roles (r_0, r_1) which have not been used in the query phase. Given the challenge roles, the challenger flips a coin $b \in \{0, 1\}$ and transforms role r_b to a blind role $\overline{r}_b$. The challenger then sends $\overline{r}_b$ to $\mathcal{A}$.
- **Query phase II:**This phase is the same as **Query phase I.** The adversary is still not allowed to query for the blind credentials of challenge roles (r_0, r_1).
- **Response:** Finally, adversary $\mathcal{A}$ outputs its guess b' for the bit b and sends it to the challenger.

The adversary in this security model can be any user (including the reference monitor) who does not have access the blind credential of the challenge role. Since this experiment allows the adversary to collect the blind credential of any role he wants (except for the challenge role), security in this model guaranties resisting against collusion attacks.

5.1 Scheme

Here, we explain the scheme under the assumption that different users in the same role are indistinguishable. This assumption makes the scheme easier to understand, but less realistic. Therefore, in Section 5.2 we present an extension of the scheme which ensures that different users in the same role *can* be distinguished.

We assume that there is a Centralized Authority (CA) in the system that controls the roles and the assignment of users to roles, in the sense that:

1. the CA decides which roles there are
2. the CA decides which user has which role
3. the CA is the only authorized entity to generate master keys, user secret keys and blinded credentials.

For now we assume that at any one time a user has only one role, and relies on the CA to enable users to assume different roles (but see Section 5.2).

The CA is assumed to be honest and not curious, which we believe is justified by the following argument. After generating a blind credential and a user key for each user, the CA can be kept off-line. Hence, although requiring a CA that is not curious seems at odds with using a curious server, since the CA is kept off-line most of the time, it would have no chance of benefitting from curiosity.

For the construction we make use of a bilinear map. Assume that $\mathbb{G}_1$ and $\mathbb{G}_2$ are two multiplicative groups of order q, where q is represented as an s bit number (s is thus a security parameter). A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ between these two groups satisfies the following properties:

1. Computable: given $g, h \in \mathbb{G}_1$ there exists a polynomial time algorithm to compute $e(g, h) \in \mathbb{G}_2$.
2. Bilinear: for any two integers $x, y \in [1, q]$ we have $e(g^x, g^y) = e(g, g)^{xy}$.
3. Non-degenerate: if g is a generator of $\mathbb{G}_1$ then $e(g, g)$ is a generator of $\mathbb{G}_2$.

We also need a random oracle $H : \{0, 1\}^* \rightarrow \mathbb{G}_1$.

A blinded reference monitor consists of the following functions:

- KeyGen : $S \rightarrow K_{prv}$, where $K_{prv} = \mathbb{Z}_q$, which when given a security parameter $s \in S$, outputs a master secret key $k \in K_{prv}$.
- UserKeyGen : $K_{prv} \times U \rightarrow K_{usr}$, where $K_{usr} = \mathbb{G}_1$, which when given the master secret key $k \in K_{prv}$ and the pseudonym of a user $usr \in U$, first picks random $a_{usr} \in \mathbb{Z}_q$ and then computes $b_{usr} = \frac{k}{a_{usr}}$. The user private key is $k_{usr} = g^{a_{usr}} \in K_{usr}$. The CA then sends $g^{a_{usr}}$ to the user and (b_{usr}, usr) to the server. This function is used exclusively by the CA. Here, the pseudonym usr is a random value generated uniquely for the user such that it does not reveal any information about the identity of the user. The pseudonym is generated by the CA.
- CredBlinding : $(K_{prv} \times R) \rightarrow \overline{C}$, where $\overline{C} = \mathbb{G}_1$, which when given the master secret key $k \in K_{prv}$, and a role $r \in R$ outputs a blinded credential $\overline{c_r} = H(r)^k \in \overline{C}$. This function is used exclusively by the CA. Since the identity of the user represented by k_{usr} is not used here, all users in the same role have the same blinded credential. In Section 5.2 we show how users can be distinguished.
- RoleBlinding : $(K_{usr} \times R) \rightarrow \overline{R}$, where $\overline{R} = (\mathbb{G}_1, \mathbb{G}_2)$ which when given the user private key $k_{usr} = g^{a_{usr}} \in K_{usr}$, and a role $r \in R$, generates a blinded role $\overline{r}$ which is executed by the user and the server as follows:
  - **User:** computes $e(g^{a_{usr}}, H(r)^\gamma)$ and sends $(g^\gamma, e(g, H(r))^{a_{usr}\gamma}, usr)$, to the server, where $\gamma \in \mathbb{Z}_q$ is a random value generated by the user.
  - **Server:** Given $(g^\gamma, e(g, H(r))^{a_{usr}\gamma}, usr)$, the server first searches the database for (usr, b_{usr}). After the server finds the tuple (usr, b_{usr}), picks b_{usr} and computes $(e(g, H(r))^{a_{usr}\gamma})^{b_{usr}} = e(g, H(r))^{\gamma k}$. The server then stores $(\overline{r} = (g^\gamma, e(g, H(r))^{\gamma k})$ on the database.
- AccessDecision :$(\overline{C} \times \overline{R}) \times I \rightarrow B$. Let $\overline{r} = (x, y)$. Given a blinded credential $\overline{c} \in \overline{C}$, a blinded role $\overline{r} \in \overline{R}$, and a document identifier $i \in I$, outputs $True \in B$ if $e(\overline{c}, x) = y$ and $\overline{ir}(i) = \overline{r}$, otherwise outputs $False \in B$. This function is used by the reference monitor.

Table 2. Relation between the high level functions and their refinements for blinding a server

High level	kd	kd^{-1}	kw	kr	$\overline{\mathsf{krw}}$
Refinement Blinded Database	DocKeyGen DocBlinding	UserKeyGen KeyUnblinding DocUnblinding	KeywordBlinding DocKeyGen Trapdoor		Search
Refinement Blinded Reference Monitor				KeyGen UserKeyGen RoleBlinding CredBlinding	AccessDecision

Here, we described the construction of our scheme for one role only. The scheme can be extended to support any expressive policy by blinding each role and then specifying the structure of the policy to the blind roles. However, in this case we blind the roles only and not the structure of the policy.

Having described the refinements of blinding database functions, we informally relate those functions to the abstract functions of Section 3.2. Table. 2 shows that functions described in Section 4 which blind the database, and the functions described in Section 5 which blind the reference monitor, are the refinement of which functions explained in Section 3.2 to blind a server.

5.2 Extension

The construction presented in section 5.1 issues a blinded credential for each user based on the role of the user. This construction imposes the following limitations: i) to revoke a user from the group, the blinded credential of all users with the same role of the revoked user should be changed, ii) the risk of the disclosure of blinded credentials is high because all users in the same role share the blinded credential.

Here, we propose a simple extension of the construction which issues a unique blinded, identity based credential for each user. In this extension the KeyGen, the UserKeyGen and the RoleBlinding functions remain unaltered. However, the CredBlinding and the AccessDecision functions are modified. The idea is to divide the blinded user agnostic role credential by the factor b_{usr} representing the user identity, and then during the access decision to multiply the user specific credential by b_{usr} again. The refinements of the functions CredBlinding and AccessDesicion are as follows:

- CredBlinding(k_{prv}, r, usr) : Given the master private key k_{prv}, a role r, and the pseudonym of the user usr, the CA picks a random $t_{usr} \in \mathbb{Z}_q$ and computes a blind identity-role credential $\overline{c_{usr,r}} = (H(r)^k)^{t_{usr}b_{usr}}$. The CA sends $\overline{c_{usr,r}} = (H(r)^k)^{t_{usr}b_{usr}}$ to the user usr and $\overline{c_{usr}} = (t_{usr}b_{usr}, usr)$ to the server for storage.
- AccessDesicion: Given the tuple consisting of role-identity credential and a user pseudonym $((H(r)^k)^{t_{usr}b_{usr}}, usr)$, the server first searches the identity

credential $\overline{c_{usr}}$ for the $t_{usr}b_{usr}$ associated with the name usr. Having found $t_{usr}b_{usr}$ the server then computes

$$((\overline{c_{usr,r}})^{\frac{1}{t_{usr}b_{usr}}}) = \overline{c_r} = H(r)^k$$

to obtain the blind role credential $\overline{c_r} = H(r)^k$. Let $\overline{\mathsf{ir}}(i) = \overline{r}$ and let $\overline{r} = (x, y)$. The reference monitor grants access to $\overline{d}$ iff $e(H(r)^k, x) = y$, and $\overline{\mathsf{id}}(i) = \overline{d}$.

Theorem 1. *The blinded policy scheme is semantically secure against a chosen plain-text attack in the random oracle model assuming DBDH is intractable.*

Proof: see the extended version.

5.3 Efficiency

We discuss the cost of our construction. Since there is no related work that addresses enforcing blind roles, we compare the efficiency of our construction to the base line provided by the standard reference monitor. Table 3 shows a break down of the costs into four categories. The first two categories describe the cost incurred by the CA per user per role. The standard reference monitor only has to pick a credential per user per role, where the blinded reference monitor performs a number of computations as indicated. The third category shows that the user and the reference monitor have to perform number of computations per role, which do not have a counterpart for the standard reference monitor. Finally, the fourth category shows that for each access the standard reference monitor does one comparison against a pairing computation by the blinded reference monitor.

6 Discussion of Practical Issues

The blinded role based access control scheme that we have presented in this paper allows a data base server to enforce the policy in such a manner that the server does not learn which user has which role. While this scheme offers better privacy than any other RBAC scheme, this enhanced privacy comes at a cost: as soon as one user reveals her private key to the server, the latter is able to discover all users who have the same role as the user who leaked her key. While in point to point encryption schemes, a user revealing her private key leads to compromising the revealing user's information only, in our scheme, since the database is shared among whole users, a user revealing her key will compromise other user's information also. To explain this security drawback in more detail, assume that a user in role u reveals her private key to the server. Since the user can blind any role using her private key, the server can now blind any role too. Hence, upon receiving a blind credential, say $\overline{c}_u$, the server takes a role s from the universe of roles and transforms it to a blind form $\overline{c}_s$ using the users private key. The server then checks if $\mathsf{AccessDecision}(\overline{c}_u, \overline{s}) = 1$. If the server learns that $\overline{c}_u$ is the blind credential of role s, the server has discovered that s maps onto $\overline{s}$, otherwise the server keeps trying other roles from the universe of roles until the above equation holds.

Table 3. Comparing the cost of the blind reference monitor with the standard reference monitor. In this table R.O. stands for the random oracle.

	Blind reference monitor			**Standard reference monitor**
Cost	**CA**	**User**	**Reference monitor**	**Reference monitor**
Cost of credential per user	Pick a value + 1 exponentiation per role			Pick a value per role
Cost of user key per user	1 exponentiation + 1 R.O. computation per role			0
Cost of role blinding		2 exponentiations + 1 R.O. computation+ 1 pairing computation per role	1 exponentiation per role	0
Cost of enforcement per access			1 pairing computation per role	1 comparison per role

Therefore, if a user reveals her private key to the server, a blind role remains secure only until the blind credential of the same role is queried. Revealing the private key of a user to the server thus impacts the security of the other users in the same role.

We now argue that the security drawback we described above is a common problem with all cryptographically enforced access control policy schemes (including attribute based encryption). The only countermeasure against the problem explained above is to prevent an adversary, who is the server in our case, from blinding roles. This can be achieved only if all users protect their private key, which in practice is not achievable.

Having to trust all users is a severe practical limitation. However, we can mitigate the risk of a user revealing her user private key in a number of ways:

False positive it would be possible to change the AccessDecision function such that it generates false positives: Let AccessDecision with a certain probability outputs 1 for blind roles that do not match with a blind credential. This causes documents to be returned to the user that she will not be able to decrypt. Increasing the false positive rate increases the uncertainty for the server of deciding the role of each blind role. However, the performance of the system will decrease due to an increase in communication overhead and an increase in the complexity on the user side, where the false positives have to be discarded. Hence, introducing false positives creates a security performance trade-off.

CA involvement for blinding. The task of the CA in our scheme is to issue a blind credential and a private key for each user when the user first enters the system. However, the CA could also be involved in blinding roles, for example by splitting the user private key in two parts. To blind a role, the

user first blinds the role using her part of the key. The CA then completes the blinding using the other part the key. In this case both a user and the CA will have to collude for the server to learn which role belongs to which user. Involving the CA in every access control decision carries a heavy cost, which represents a sever drawback of this mitigating approach.

Restriction on blinding roles. In many practical scenarios, each user should be able to blind only a subset of the available roles. If this is the case, a limited number of roles is revealed to the server when a user reveals her private key to the server. However, restricting the roles each user can blind requires an access control policy by the CA. Therefore, we are actually bootstrapping the access control policy of the server by the access control policy of the CA, which is at least inelegant.

Forward security. To mitigate the effect of key leakage the scheme could be made forward secure in the sense that given a blind credential only the blind roles stored on the server beforehand can be enforced. More technically, each role is blinded with a user key and the time when the blind role is constructed. The user also re-blinds her blind credential such that the AccessDecision function outputs 1 if (i) the blind credential and the blind roles match and (ii) the time of the storage is before the time of the query. However, this approach requires that the user, who requests a document, constructs the forward secrecy part of the blind credential with the right time.

The ideas presented above just mitigate the risk to other users of one user revealing her private key. However, we believe that reducing the risk to zero is impossible and this is an inherent limitation of using purely cryptographic means to enhance the privacy of access control [23]. As we mentioned earlier this limitation comes from the fact that the security of cryptographic schemes relies on the ability of users to protect their private keys. For applications where a private key is used for a shared database, revealing a private key has a large impact on the security of the entire database. Therefore, reducing the risk to zero might be possible by applying techniques beside cryptography. As an instance of the same problem consider digital rights management (DRM) [11,18] where to protect digital contents, the latter is symmetrically encrypted. The decryption key is sent to the right users who have legally purchased the content. Therefore all users who acquire the decryption key must be trusted since once a user reveals her private key, the encrypted content can be copied and used illegally. Here key management has been one of the cryptographic restrictions of DRM, and to cope with such limitations, digital watermarking techniques have been proposed to hide the encrypted message in a multimedia signal [18].

7 Conclusion

We proposed a scheme which enhances the privacy of role based access control systems by deploying cryptographic techniques. Our scheme reduces the communication overhead and user's computational complexity in comparison with

attribute based encryption techniques where the policy is enforced cryptographically by the users. Our scheme prevents the reference monitor of a server to learn any information about the policy even after performing the access decision. Our scheme also resists against attacks from colluding users. However, our scheme remains secure as long as no user reveals her private key to the server. After a user reveals hey private key to the server, the latter will be able to learn the role of the blind roles that matches with a blind credential. We have discussed this drawback and argued that this is an inherent limitation of cryptographic tools on such privacy enhancing access policy schemes.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. J. Cryptol. 21(3), 350–391 (2008)
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
3. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004), http://dx.doi.org/10.1007/b97182
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. SIAM J. Comput. 32(3), 586–615 (2003)
5. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
6. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (Without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
7. Bradshaw, R.W., Holt, J.E., Seamons, K.E.: Concealing complex policies with hidden credentials. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, pp. 146–157. ACM, New York (2004), http://doi.acm.org/10.1145/1030083.1030104
8. Chang, Y., Mitzenmacher, M.: Privacy preserving keyword searches on remote encrypted data. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 442–455. Springer, Heidelberg (2005)
9. Curtmola, R., Garay, J., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. In: CCS 2006: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 79–88. ACM, New York (2006)
10. Dong, C., Russello, G., Dulay, N.: Shared and searchable encrypted data for untrusted servers. In: DBSec, pp. 127–143 (2008)
11. Feigenbaum, J., Freedman, M.J., Sander, T., Shostack, A.: Privacy engineering for digital rights management systems. In: Sander, T. (ed.) DRM 2001. LNCS, vol. 2320, pp. 76–105. Springer, Heidelberg (2002)
12. Frikken, K., Atallah, M., Li, J.: Attribute-based access control with hidden policies and hidden credentials. IEEE Trans. Comput. 55, 1259–1270 (2006), http://portal.acm.org/citation.cfm?id=1159156.1159225

13. Goh, E.: Secure indexes. Cryptology ePrint Archive, Report 2003/216 (2003), http://eprint.iacr.org/2003/216/
14. Golle, P., Staddon, J., Waters, B.: Secure conjunctive keyword search over encrypted data. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 31–45. Springer, Heidelberg (2004)
15. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) ACM Conference on Computer and Communications Security, pp. 89–98. ACM, New York (2006)
16. Hwang, Y., Lee, P.: Public key encryption with conjunctive keyword search and its extension to a multi-user system. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 2–22. Springer, Heidelberg (2007)
17. Iovino, V., Persiano, G.: Hidden-vector encryption with groups of prime order. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 75–88. Springer, Heidelberg (2008)
18. Jonker, W., Linnartz, J.P.: Digital rights management in consumer electronics products. IEEE Signal Processing Magazine 21(2), 82–91 (2004), http://doc.utwente.nl/55653/
19. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
20. Li, J., Li, N.: Oacerts: Oblivious attribute certificates. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 301–317. Springer, Heidelberg (2005)
21. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 111–129. Springer, Heidelberg (2008)
22. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
23. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 457–473. Springer, Heidelberg (2009)
24. Shi, E., Waters, B.: Delegating capabilities in predicate encryption systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 560–578. Springer, Heidelberg (2008)
25. Sion, R., Carbunar, B.: Conjunctive keyword search on encrypted data with completeness and computational privacy (2005)
26. Song, D., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: SP 2000: Proceedings of the 2000 IEEE Symposium on Security and Privacy, p. 44. IEEE Computer Society, Washington, DC, USA (2000)
27. Waters, B.R., Balfanz, D., Durfee, G., Smetters, D.K.: Building an encrypted and searchable audit log. In: Proceedings of Network and Distributed System Security Symposium 2004 (NDSS 2004), San Diego, CA (February 2004)
28. Yang, Y., Bao, F., Ding, X., Deng, R.: Multiuser private queries over encrypted databases. IJACT 1(4), 309–319 (2009)