Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 171 (2007) 107–120
www.elsevier.com/locate/entcs

Establishment of Ad-Hoc Communities through Policy-Based Cryptography

Walid Bagga[1]

Institut Eurecom
Corporate Communications
2229, route des Crêtes B.P. 193
06904 Sophia Antipolis (France)

Stefano Crosta , Pietro Michiardi , Refik Molva

Institut Eurecom
Corporate Communications
2229, route des Crêtes B.P. 193
06904 Sophia Antipolis (France)

Abstract

A policy-based encryption scheme allows to encrypt a message according to a credential-based policy formalized as monotone Boolean expression written in standard normal form. The encryption is so that only the users having access to a qualified set of credentials for the policy are able to decrypt the message. In this paper, we first revisit the formal definition of policy-based encryption and describe a policy-based encryption scheme from bilinear pairings. Our scheme improves the one proposed in [2] in terms of ciphertext size, while at the same time preserving the computational efficiency. Then, we describe an application of policy-based encryption in the context of ad-hoc networks. More precisely, we show how the policy-based encryption primitive can be used to achieve a privacy-enhanced secure establishment of ad-hoc communities.

Keywords: Bilinear Pairings, Credentials, Ad-Hoc Communities

1 Introduction

The concept of policy-based cryptography, first formalized in [2], provides a framework for performing cryptographic operations with respect to policies formalized as monotone Boolean expressions written in standard normal forms. In particular, a policy-based encryption scheme allows to encrypt a message with respect to a policy in such a way that only the users that are compliant with the policy are able to

[1] Email: walid.bagga@eurecom.fr

doi:10.1016/j.entcs.2006.11.013

decrypt the message. Basically, a policy consists of conjunctions (logical AND operation) and disjunctions (logical OR operation) of conditions, where each condition is fulfilled by a digital credential representing the signature of a specific credential issuer on a certain assertion. A user is thus compliant with a policy if and only if he has been issued a qualified set of credentials for the policy i.e. a set of credentials fulfilling the combination of conditions defined by the policy. More generally, policy-based encryption belongs to an emerging family of cryptographic schemes sharing the ability to integrate encryption with credential-based access structures. This ability allows for several interesting applications in different contexts including but not restricted to oblivious access control [2,7,18], trust negotiation [6,11,17], and cryptographic workflow [1].

In the first part of this paper (Section 2), we provide a further refinement of the formalization of the policy-based encryption primitive given [2]. Then, we describe a policy-based encryption scheme from bilinear pairings. Our scheme improves the one proposed in [2] in terms of ciphertext size, while at the same time preserving the computational efficiency. Furthermore, in contrast with the heuristic, rather intuitive security analysis provided in [2], our scheme is supported by formal security arguments. Due to space limitation, the details of our security analysis are given in the appendix.

In [14], an ad-hoc network is perceived as a community of interconnected autonomous devices providing services and resources to each other. Such devices often belong to users from different security domains that do not have pre-existing trust relationships. A security framework is therefore needed to ensure trustworthy interactions within such kind of communities. A comprehensive policy-based security framework supporting the establishment, evolution and management of ad-hoc networks is proposed in [14]. In the second part of this paper (Section 3), we leverage this framework, and show how the policy-based encryption primitive could be used to achieve a privacy-enhanced secure establishment of ad-hoc communities.

2 Policy-Based Encryption

In this section, we first set the context for the policy-based encryption primitive including the terminology, the notation and the policy model. Then, we provide a precise definition for policy-based encryption schemes and describe our pairing-based scheme. In the appendix, we provide a formal definition for message confidentiality in the context of policy-based encryption. Then, we prove the security of our scheme in the random oracle model.

2.1 Setting the Context

The concept of policy-based cryptography considers environments where interactions may occur between entities from different security domains without pre-existing knowledge of each other. Such interactions can involve the exchange of sensitive resources in which case they need to be carefully controlled through clear

and concise policies. In the considered environments, the identity of users is rarely of interest to determining whether a user could be trusted or authorized to conduct some sensitive transactions. Instead, statements about the user such as attributes, properties, capabilities and/or privileges are more relevant. The validity of such statements is checked and certified by trusted entities called credential issuers through a digital signature procedure.

Each user that wants to use the policy-based encryption primitive defines a trust infrastructure that consists of a set of credential issuers $\mathcal{I} = \{I_1, \ldots, I_N\}$, where the public key of I_κ, for $\kappa \in \{1, \ldots, N\}$, is denoted R_κ while the corresponding master key is denoted s_κ. Any credential issuer $I_\kappa \in \mathcal{I}$ may be asked by a user to issue a credential corresponding to a set of statements. The requested credential is basically the digital signature of the credential issuer on an assertion denoted A containing the set of statements as well as a set of additional information such as the validity period of the credential. Note that the term "trust infrastructure" means that the user trusts any credential issuer belonging to $\mathcal{I}$ for never issuing credentials corresponding to invalid assertions.

Upon receiving a request for generating a credential on assertion A, a credential issuer I_κ first checks the validity of the assertion. If it is valid, then I_κ executes a credential generation algorithm and returns a credential denoted $\varsigma(R_\kappa, A)$. Otherwise, I_κ returns an error message. Upon receiving the credential $\varsigma(R_\kappa, A)$, the requester may check its integrity using I_κ's public key R_κ. The process of checking the validity of a set of statements about a certain entity is out of the scope of this paper. Note that it is implicitly assumed that end users know a trustworthy value of the public key of each of the credential issuers included in $\mathcal{I}$. Besides, as the representation of assertions is out of the scope of this paper, note that they will simply be encoded as binary strings.

We consider credential-based policies formalized as monotone boolean expressions involving conjunctions (AND / $\wedge$) and disjunctions (OR / $\vee$) of credential-based conditions. A credential-based condition is defined through a pair $\langle I_\kappa, A\rangle$ specifying an assertion $A \in \{0,1\}^*$ and a credential issuer $I_\kappa \in \mathcal{I}$ that is trusted to check and certify the validity of A. A user fulfills the condition $\langle I_\kappa, A\rangle$ if and only if he has been issued the credential $\varsigma(R_\kappa, A)$.

We consider policies written in standard normal forms, i.e. written either in conjunctive normal form (CNF) or in disjunctive normal form (DNF). In order to address the two standard normal forms, we use the conjunctive-disjunctive normal form (CDNF) introduced in [18]. Thus, a policy denoted Pol is written as follows:

$$Pol = \wedge_{i=1}^{m}[\vee_{j=1}^{m_i}[\wedge_{k=1}^{m_{i,j}}\langle I_{\kappa_{i,j,k}}, A_{i,j,k}\rangle]], \text{ where } I_{\kappa_{i,j,k}} \in \mathcal{I} \text{ and } A_{i,j,k} \in \{0,1\}^*$$

Under the CDNF notation, policies written in CNF correspond to the case where $\{m_{i,j} = 1\}_{i,j}$, while policies written in DNF correspond to the case where $m = 1$.

Note. Writing policies in standard normal forms allows us to improve the performance of policy-based encryption, both in terms of both computational cost and bandwidth consumption, as will be shown in subsection 2.3.

Let $\varsigma_{j_1,\ldots,j_m}(Pol)$ denote the set of credentials $\{\{\varsigma(R_{\kappa_{i,j_i},k}, A_{i,j_i,k})\}_{k=1}^{m_{i,j_i}}\}_{i=1}^{m}$, for some $\{j_i \in \{1,\ldots,m_i\}\}_{i=1}^{m}$. Then, $\varsigma_{j_1,\ldots,j_m}(Pol)$ is a qualified set of credentials for Pol.

2.2 Formal Definition

A policy-based encryption scheme is specified by five algorithms: Setup, Issuer-Setup, CredGen, PolEnc and PolDec which we describe below.

Setup. On input of a security parameter k, this algorithm generates a set of public parameters $\mathcal{P}$ which specifies the different parameters, groups and public functions that will be referenced by subsequent algorithms. Furthermore, it specifies a message space $\mathcal{M}$ and a ciphertext space $\mathcal{C}$.

Issuer-Setup. This algorithm generates a random master key s_κ and the corresponding public key R_κ for credential issuer $I_\kappa \in \mathcal{I}$.

CredGen. On input of the public key R_κ of a credential issuer $I_\kappa \in \mathcal{I}$ and an assertion $A \in \{0,1\}^*$, this algorithm returns the credential $\varsigma(R_\kappa, A)$.

PolEnc. On input of a message $M \in \mathcal{M}$ and a policy Pol, this algorithm returns a ciphertext $C \in \mathcal{C}$ representing the encryption of M with respect to policy Pol.

PolDec. On input of a ciphertext $C \in \mathcal{C}$, a policy Pol and a qualified set of credentials $\varsigma_{j_1,\ldots,j_m}(Pol)$, this algorithm returns a message $M \in \mathcal{M}$ or $\perp$ (for 'error').

The algorithms described above have to satisfy the standard consistency constraint i.e.

$$C = \mathsf{PolEnc}(M, Pol) \;\Rightarrow\; \mathsf{PolDec}(C, Pol, \varsigma_{j_1,\ldots,j_m}(Pol)) = M, \text{ for some } \{j_i \in \{1,\ldots,m_i\}\}_{i=1}^{m}$$

In the following, we describe a concrete implementation of policy-based encryption using bilinear pairings over elliptic curves.

2.3 Our Policy-Based Encryption Scheme

Before describing our policy-based encryption scheme, we define algorithm *BDH-Setup* as follows:

BDH-Setup. On input of a security parameter k, generate a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, e, P)$ where the map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear pairing, $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, *)$ are two groups of the same order q, and P is a random generator of $\mathbb{G}_1$. The generated parameters are such that the Bilinear Diffie-Hellman Problem (denoted BDHP) is hard.

Note-1. We recall that a bilinear pairing satisfies the following three properties:

(i) Bilinear: for $Q, Q' \in \mathbb{G}_1$ and for $a, b \in \mathbb{Z}_q^*$, $e(a \cdot Q, b \cdot Q') = e(Q, Q')^{ab}$

(ii) Non-degenerate: $e(P, P) \neq 1$ and therefore it is a generator of $\mathbb{G}_2$

(iii) Computable: there exists an efficient algorithm to compute $e(Q, Q')$ for all $Q, Q' \in \mathbb{G}_1$.

Note-2. BDHP is defined as follows: on input of a tuple $(P, a \cdot P, b \cdot P, c \cdot P)$ for randomly chosen $a, b, c \in \mathbb{Z}_q^*$, compute the value $e(P,P)^{abc}$. The hardness of BDHP can be ensured by choosing groups on supersingular elliptic curves or hyperelliptic curves over finite fields and deriving the bilinear pairings from Weil or Tate pairings. The hardness of BDHP implies the hardness of the so called Computational Diffie-Hellman Problem (denoted CDHP) which is defined as follows: on input of a tuple $(P, a \cdot P, b \cdot P)$ for randomly chosen $a, b \in \mathbb{Z}_q^*$, compute the value $ab \cdot P$. As we merely apply these mathematical primitives in this paper, we refer for instance to [12,19] for further details.

Our policy-based encryption scheme consists of the algorithms described below.

Setup. On input of the security parameter k, do the following:

(i) Run algorithm BDH-Setup to obtain a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, e, P)$

(ii) Let $\mathcal{M} = \{0,1\}^{n-n_0}$ and $\mathcal{C} = \mathbb{G}_1 \times (\{0,1\}^n)^*$ (for some $n, n_0 \in \mathbb{N}^*$ such that $n_0 \ll n$)

(iii) Define three hash functions: $H_0 : \{0,1\}^* \to \mathbb{G}_1$, $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \to \{0,1\}^n$

(iv) Let $\mathcal{I} = (q, \mathbb{G}_1, \mathbb{G}_2, e, P, n, n_0, H_0, H_1, H_2)$

Issuer-Setup. Let $\mathcal{I} = \{I_1, \ldots, I_N\}$ be a set of credential issuers. Each credential issuer $I_\kappa \in \mathcal{I}$ picks at random a secret master key $s_\kappa \in \mathbb{Z}_q^*$ and publishes the corresponding public key $R_\kappa = s_\kappa \cdot P$.

CredGen. On input of issuer $I_\kappa \in \mathcal{I}$ and assertion $A \in \{0,1\}^*$, this algorithm outputs the credential $\varsigma(R_\kappa, A) = s_\kappa \cdot H_0(A)$.

PolEnc. On input of message $M \in \mathcal{M}$ and policy Pol, do the following:

(i) Pick at random $M_i \in \{0,1\}^{n-n_0}$ (for $i = 1, \ldots, m-1$), then compute $M_m = M \oplus (\oplus_{i=1}^{m-1} M_i)$

(ii) Pick at random $t_i \in \{0,1\}^{n_0}$ (for $i = 1, \ldots, m$)

(iii) Compute $r = H_1(M_1 \| \ldots \| M_m \| t_1 \| \ldots \| t_m)$, then compute $U = r \cdot P$

(iv) Compute $\pi_{i,j} = \prod_{k=1}^{m_{i,j}} e(R_{\kappa_{i,j,k}}, H_0(A_{i,j,k}))$ (for $j = 1, \ldots, m_i$ and $i = 1, \ldots, m$)

(v) Compute $\mu_{i,j} = H_2(\pi_{i,j}^r \| i \| j)$, then compute $v_{i,j} = (M_i \| t_i) \oplus \mu_{i,j}$ (for $j = 1, \ldots, m_i$ and $i = 1, \ldots, m$)

(vi) Return $C = (U, [[v_{i,j}]_{j=1}^{m_i}]_{i=1}^{m})$

The intuition behind the encryption algorithm PolEnc is as follows: each conjunction of conditions $\wedge_{k=1}^{m_{i,j}} \langle I_{\kappa_{i,j,k}}, A_{i,j,k} \rangle$ is first associated to a mask $\mu_{i,j}$ that depends on the different credentials related to the specified conditions. The encrypted message M is split into m random shares $[M_i]_{i=1}^m$, then for each index $i \in \{1, \ldots, m\}$, the value $M_i \| t_i$ is associated to the disjunction $\vee_{j=1}^{m_i} \wedge_{k=1}^{m_{i,j}} \langle I_{\kappa_{i,j,k}}, A_{i,j,k} \rangle$, where t_i is a randomly chosen intermediate key. Each value $M_i \| t_i$ is encrypted m_i times using each of the masks $\mu_{i,j}$. This way, it is sufficient to compute any one of the masks $\mu_{i,j}$ in order to be able to retrieve $M_i \| t_i$. In order to be able to retrieve the encrypted

message, an entity needs to retrieve all the shares as well as all the intermediate keys t_i using a set of qualified credentials for policy Pol.

PolDec. On input of ciphertext $C = (U, [[v_{i,j}]_{j=1}^{m_i}]_{i=1}^{m})$, policy Pol, and the qualified set of credentials $\varsigma_{j_1,\ldots,j_m}(Pol)$, do the following:

(i) Compute $\tilde{\pi}_{i,j_i} = e(U, \sum_{k=1}^{m_{i,j_i}} \varsigma(R_{\kappa_{i,j_i},k}, A_{i,j_i,k}))$ (for $i = 1, \ldots, m$)

(ii) Compute $\tilde{\mu}_{i,j_i} = H_2(\tilde{\pi}_{i,j_i} \| i \| j_i)$, then compute $(M_i \| t_i) = v_{i,j_i} \oplus \tilde{\mu}_{i,j_i}$

(iii) Compute $r = H_1(M_1 \| \ldots \| M_m \| t_1 \| \ldots \| t_m)$

(iv) If $U = r \cdot P$, then return the message $M = \oplus_{i=1}^{m} M_i$, otherwise return $\perp$

The algorithms described above satisfy the standard consistency constraint. In fact, thanks to the bilinearity property of bilinear pairings, the following holds

$$\tilde{\pi}_{i,j_i} = e(r \cdot P, \sum_{k=1}^{m_{i,j_i}} s_{\kappa_{i,j_i},k} \cdot H_0(A_{i,j_i,k})) = \prod_{k=1}^{m_{i,j_i}} e(s_{\kappa_{i,j_i},k} \cdot P, H_0(A_{i,j_i,k}))^r = \pi_{i,j_i}^r$$

The essential operation in pairing-based cryptography is pairing computations. In Table 1, we provide the computational costs of our encryption and decryption algorithms in terms of pairing computations as well as the size of the resulting ciphertext. Note that l_1 denotes the bit-length of the bilinear representation of an element of group $\mathbb{G}_1$.

	Encryption	Decryption	Ciphertext Size
Our scheme	$\sum_{i=1}^{m} \sum_{j=1}^{m_i} m_{i,j}$	m	$l_1 + (\sum_{i=1}^{m} m_i).n$
The scheme of [2]	$\sum_{i=1}^{m} \sum_{j=1}^{m_i} m_{i,j}$	m	$l_1 + (\sum_{i=1}^{m} m_i).n + n$
The scheme of [6]	$\sum_{i=1}^{m} \sum_{j=1}^{m_i} m_{i,j}$	$\sum_{i=1}^{m} m_{i,j_i}$	$l_1 + (\sum_{i=1}^{m} \sum_{j=1}^{m_i} m_{i,j}).n + n$

Table 1
Performance of our scheme compared with the schemes of [2] and [6]

In Table 1, we include the costs related to the encryption scheme described in [2] and the one described in [6]. In fact, the latter is a policy-based encryption scheme when applied to policies written in standard normal forms following the notation defined in Section 2. While the three encryption algorithms require the same amount of pairing computations, our decryption algorithm and the one described in [2] are more efficient than the one described in [6] because $m_{i,j_i} \geq 1$ for $i = 1, \ldots, m$. Besides, our scheme leads to ciphertexts more compact than the ones given by the scheme of [2]. Finally, observe that because $m_{i,j} \geq 1$ for $j = 1, \ldots, m_i$ and $i = 1, \ldots, m$, the size of the ciphertexts resulting from the scheme of [2] is at least as short as the one of the ciphertexts produced by the scheme of [6].

Note-1. As for standard asymmetric encryption schemes, policy-based encryption schemes are much less efficient than symmetric encryption schemes. In practice, they should be used to exchange the symmetric (session) keys that are used for bulk encryption.

Note-2. In some scenarios, a policy may be expressed in terms of a (k, n)-threshold structure (for $1 < k < n$). Our approach offers an advantage over threshold schemes in that our schemes address general access structures including those for which

there exists no corresponding (k, n)-threshold representation (such structures exist according to Theorem 1 of [4]). Although any threshold structure may be written using only $\wedge$ and $\vee$ operators and thus may match our formalism, we believe that dedicated threshold schemes (in particular some ID-based threshold decryption schemes) might handle such structures more efficiently and elegantly than our generic approach.

Note-3. The concept of policy-based cryptography might be compared with the concept of generalized threshold cryptosystems introduced in [16]. Generalized threshold cryptography is an extension of the original threshold cryptography to the case of general access structures. Our policy-based encryption scheme uses a technique similar but not exactly the same than the secret sharing method presented in [4]. In [9], it has been observed that using the previous general secret sharing method it is possible to construct an RSA-based generalized threshold decryption scheme. In contrast with the RSA-oriented approach, our policy-based encryption scheme supports the notion of cryptographic workflow discussed in [1]. In fact, using our approach, a message can be encrypted with respect to a specific access structure before the decryption keys (the credentials) are generated and given to the authorized users. As future work, we are planning to conduct a detailed comparison between policy-based and generalized threshold cryptosystems.

3 Establishment of Ad-Hoc Communities through Policy-Based Encryption

In [14], Keoh et al. propose a comprehensive policy-based security framework supporting the establishment, evolution and management of ad-hoc networks. In this section, we first provide an overview of their approach. Then, we leverage their policy-based trust establishment model and show, through the description of an application scenario, how the policy-based encryption primitive can be used to achieve a secure establishment of ad-hoc communities, while adhering to the privacy principle of data minimization (called the *data quality principle* in OECD guidelines [8]) according to which only strictly necessary information should be collected for a given purpose.

3.1 Policy-Based Establishment of Ad-Hoc Communities

In [14], an ad-hoc network is perceived as a community of interconnected autonomous devices providing services and resources to each other. More precisely, ad-hoc communities are defined as follows:

Definition. *An ad-hoc community interconnects a group of devices, maintains membership and ensures that only entities, i.e., users or computing services, which possess certain credentials, attribute information and characteristics can join the community (common characteristics). The members of the community rely upon each other to provide services and share resources (interactions). These interactions are regulated through a set of well-defined rules and policies (law) that govern the*

access to the services and resources in the community.

With regard to their definition of ah-hoc communities, Keoh et al. introduce a community specification, called *doctrine*. The latter specifies a set of roles that can be associated to the participants in the community, the characteristics that participants must exhibit in order to be eligible to play a specific role, as well as the authorization and obligation policies governing the behavior of the participants within the community depending on their roles. Based on the doctrine, a set of security protocols is proposed to bootstrap the community, manage the membership (joining and leaving the community), and govern the access to the services provided by the participants.

Note. As in [14], we assume throughout this section that there is an underlying routing infrastructure that supports the relay of data packets in an ad-hoc network.

The characteristics that a participant must fulfill in order to be eligible to play a specific role in a community are expressed in terms of a credential-based policy, called user-role policy, that is formalized as monotone Boolean expression written in a standard normal form. The policy is defined by the entity that initiates the bootstrapping of the community, and is broadcasted (flooded) to the other participants. The credentials considered in [14] are public-key certificates (X.509 certificates) issued by certification authorities and attribute certificates (SPKI/SDSI) issued by trusted attribute authorities. As argued in [14], the idea of the proposed approach is not to establish trusted authorities in mobile ad-hoc networks. On the contrary, it is assumed that the participants have been already issued various certificates during their past connections to the wired environment. Such assumption is admissible in a wide range of application scenarios. For example, consider the case where the laptops and PDAs of different persons interact in an ad-hoc business meeting. Typically, the interacting devices belong to individuals from multiple domains: employees of their institutions or companies, members of collaborative projects, *etc.* In each domain, the individuals obtain credentials certifying their attributes within the domain. A credential is basically the signature of the credential issuer on an assertion that binds an identifier (a public key or a pseudonym) of the credential owner to the set of statements/attributes whose validity is checked and certified by the credential issuer.

Note. First, note that the policy model in [14] is limited to policies written in the Disjunctive Normal Form (DNF) but can naturally be extended to support the Conjunctive Normal Form (CNF). Besides, the trust model relies on a security infrastructure that consists of well-established trusted authorities in the Internet. We can extend this model to support any entity, including the participants themselves, that is trusted to check and certify the validity of specific credentials. Finally, note that it is assumed that the entity that defines the user-role policy have access to trusted values of the public keys of the different credential issuers referenced in the policy.

The community bootstrapping and community joining protocols described in [14] necessitate the verification of the compliance of the participants that want to join

the community with the user-role policies associated to the roles they wish to play within the community. Such verification, as described in [14], involves the exchange of credentials, checking their validity as well as their compliance with the user-role policies. For an illustration, consider the scenario described below:

Scenario. *Alice is on a business trip for the collaborative project P. On the train there might be other colleagues from different companies working on the same project. Alice has some documents she is willing to share and possibly discuss only with the members of the project that are either from company X or from company Y.*

Following the approach proposed in [14], Alice defines a community with a role, say $r_p = partner$, that allows having access to the proposed documents as well as initiating a private discussion with Alice. The user-role policy (defined by Alice) associated to the role r_p is:

$$Pol_{r_p} = [\langle Company\ X\ ,\ Bob\ is\ Employee\rangle \vee \langle Company\ Y\ ,\ Bob\ is\ Employee\rangle] \wedge \langle Project\ P\ ,\ Bob\ is\ Member\rangle$$

Alice initiates the bootstrapping of the community by flooding her policy Pol_{r_p} as well as the privileges granted by role r_p. Assume that Bob, who is a member of the collaborative project P working for company X, is interested in reading the documents proposed by Alice and potentially having a private discussion with her. In this case, Bob needs to send a join request to Alice as well as the credentials proving his compliance with the policy Pol_{r_p} i.e. his employee credential delivered by company X and his membership credential delivered by the manager of project P. Upon receiving Bob's joining request, Alice first checks the validity of his credentials using the public keys of the credential issuers, then she checks that the received credentials fulfill the policy Pol_{r_p}. Once the admission conditions are validated, Bob is assigned a token granting the privileges corresponding to role r_p i.e. Bob can have access to the project's documents held by Alice and can initiate a private discussion with Alice.

3.2 Our Approach

In the scenario described above, the main concern of Alice is to ensure that the participants that are not compliant with the user-role policy Pol_{r_p} associated to role r_p cannot have the privileges corresponding to r_p i.e. they cannot read the documents proposed by Alice and cannot initiate a private discussion with her. In other words, Alice wants to be sure that her user-role policy is effectively enforced. Consider the two use cases described below:

(i) In the first case, Bob is interested in reading the documents proposed by Alice. However, he is not willing to have further interactions with her. According to the privacy principle of data minimization, the policy enforcement mechanism should not allow Alice to know whether Bob is compliant with her policy.

(ii) In the second case, after having read Alice's documents, Bob wants to have further interactions with her. Alice will know that Bob is compliant with her policy. However, according to the privacy principle of data minimization, the

policy enforcement mechanism should not allow Alice to know for which specific company Bob is working i.e. Alice should not know the fact that Bob is from company X.

In the two cases described above, the standard approach proposed in [14] for policy enforcement cannot respect the privacy principle of data minimization. In fact, because Bob must provide the credentials proving his compliance with Pol_{r_p}, Alice will know anyway whether his is compliant with her policy and from which company he comes from. More generally, as long as the policy enforcement mechanism involves the exchange of credentials, the privacy principle of data minimization cannot be satisfied.

The policy-based encryption primitive can be used to enforce the policy of Alice while respecting the privacy principle of data minimization. In the following, we describe a simple mechanism that illustrates our approach.

(i) Upon receiving the join request of Bob, Alice generates at random a symmetric key k_s which she uses to encrypt the different documents she is willing to share with Bob if he fulfills the admission policy Pol_{r_p}. Then, Alice generates at random a nonce n_{Bob} and encrypts the pair (k_s, n_{Bob}) with respect to the policy Pol_{r_p} using a policy-based encryption algorithm (PolEnc). Finally, Alice sends the resulting ciphertexts to Bob.

(ii) Upon receiving the two ciphertexts, Bob uses his credentials to decrypt the pair (k_s, n_{Bob}) using the policy-based decryption algorithm (PolDec) corresponding to the encryption scheme used by Alice. As Bob has access to a qualified set of credentials for policy Pol_{r_p}, he is able to get the symmetric key k_s which he uses to decrypt the different documents he wanted to read. In the case where there is no further interactions between Alice and Bob, there is no way for Alice to know that Bob fulfills her policy.

(iii) In the case where Bob is willing to start a private discussion with Alice, he can send his request with the nonce n_{Bob}. The fact that he knows such random nonce proves to Alice that Bob was able to decrypt her ciphertexts and therefore is compliant with her policy. However, as Alice does not know which specific credentials were used to decrypt the nonce, she cannot know which company Bob is working for.

As shown in the simple mechanism described above, the policy-based encryption primitive allows to enforce the communities' user-role policies, while adhering to the privacy principle of data minimization. This is enabled by the fact that, in contrast with the standard approach where the credentials need to be exchanged, the credentials are used as decryption keys in policy-based encryption. We are currently investigating the implementation of a comprehensive security framework for ad-hoc communities using policy-based encryption that optimizes the overall number of messages that need to be exchanged between the participants.

4 Conclusion

The concept of policy-based cryptography is a promising paradigm for trust establishment and authorization in open environments. In this paper, we focused on the policy-based encryption primitive. We first described a provably secure policy-based encryption scheme from bilinear pairings that improves the existing schemes in terms of ciphertext size. Then, we leveraged the security framework proposed in [14] and showed how policy-based encryption can be used for a privacy-enhanced enforcement of the user-role policies specified by ad-hoc communities.

References

[1] S.S. Al-Riyami, J. Malone-Lee, and N.P. Smart. Escrow-free encryption supporting cryptographic workflow. Cryptology ePrint Archive, Report 2004/258, 2004. http://eprint.iacr.org/.

[2] W. Bagga and R. Molva. Policy-based cryptography and applications. In *Proceedings of Financial Cryptography and Data Security (FC'05)*, volume 3570 of *LNCS*, pages 72–87. Springer-Verlag, 2005.

[3] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM Press, 1993.

[4] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *CRYPTO '88: Proceedings on Advances in cryptology*, pages 27–35, New York, NY, USA, 1990. Springer-Verlag New York, Inc.

[5] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229. Springer-Verlag, 2001.

[6] R. Bradshaw, J. Holt, and K. Seamons. Concealing complex policies with hidden credentials. Cryptology ePrint Archive, Report 2004/109, 2004. http://eprint.iacr.org/.

[7] L. Chen, K. Harrison, D. Soldera, and N. Smart. Applications of multiple trust authorities in pairing based cryptosystems. In *Proceedings of the International Conference on Infrastructure Security*, pages 260–275. Springer-Verlag, 2002.

[8] Organization for Economic Cooperation and Development (OECD). Recommendation of the council concerning guidelines governing the protection of privacy and transborder flows of personal data, 1980. http://www.oecd.org/home/.

[9] Y. Frankel and Y. Desmedt. Parallel reliable threshold multisignature. Technical Report, University of Wisconsin-Milwaukee, TR-92-04-02, April, 1992.

[10] D. Galindo. Boneh-franklin identity based encryption revisited. To appear in Proceedings of 32nd International Colloquium on Automata, Languages and Programming (ICALP 2005).

[11] J. Holt, R. Bradshaw, K. E. Seamons, and H. Orman. Hidden credentials. In *Proc. of the 2003 ACM Workshop on Privacy in the Electronic Society*. ACM Press, 2003.

[12] A. Joux. The weil and tate pairings as building blocks for public key cryptosystems. In *Proceedings of the 5th International Symposium on Algorithmic Number Theory*, pages 20–32. Springer-Verlag, 2002.

[13] J. Kahn. Entropy, independent sets and antichains: a new approach to dedekind's problem. In *Proc. Amer. Math. Soc. 130*, pages 371–378, 2002.

[14] S. L. Keoh, E. Lupu, and M. Sloman. Peace: A policy-based establishment of ad-hoc communities. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04)*, pages 386–395. IEEE Computer Society, 2004.

[15] D. Kleitman. On dedekind's problem: the number of monotone boolean functions. In *Proc. Amer. Math. Soc. 21*, pages 677–682, 1969.

[16] C. Laih and L. Harn. Generalized threshold cryptosystems. In *ASIACRYPT '91: Proceedings of the International Conference on the Theory and Applications of Cryptology*, pages 159–166, London, UK, 1993. Springer-Verlag.

[17] N. Li, W. Du, and D. Boneh. Oblivious signature-based envelope. In *Proceedings of the 22nd annual symposium on Principles of distributed computing*, pages 182–189. ACM Press, 2003.

[18] N. Smart. Access control using pairing based cryptography. In *Proceedings CT-RSA 2003*, pages 111–121. Springer-Verlag LNCS 2612, April 2003.

[19] Y. Yacobi. A note on the bilinear diffie-hellman assumption. Cryptology ePrint Archive, Report 2002/113, 2002. http://eprint.iacr.org/.

A Formal Security Analysis

In the following, we provide the details of the security analysis of the policy-based encryption scheme described in Section 2. We first describe a formal model for message confidentiality adapted to the context of policy-based cryptography. Then, we state the results related to our scheme and give the details of our reductionist security proof.

A.1 Formal Security Model

The motivation behind our model is as follows: the standard acceptable notion of security for public key encryption schemes is indistinguishability against chosen ciphertext attacks (denoted IND-CCA). Hence, it is natural to require that a policy-based encryption scheme also satisfies this strong notion of security. However, the definition of this security notion must be adapted to the policy-based setting. In [5], Boneh and Franklin extend the IND-CCA model to a stronger security model denoted IND-ID-CCA. The latter allows the adversary to obtain the private key corresponding to any identifier of his choice, other than the identifier being attacked. Furthermore, in the defined IND-ID-CCA model, the adversary is allowed to choose the identifier on which he wishes to be challenged, while in the standard IND-CCA model, the adversary is challenged on a randomly chosen public key. In a similar way, our security model should allow the adversary to obtain a set of credentials fulfilling any policy of his choice, other than the policy on which he is challenged. Furthermore, the adversary should be allowed to specify the challenge policy.

We define our security model in terms of an interactive game, played between a challenger and an adversary. The game consists of five stages: Setup, Phase-1, Challenge, Phase-2 and Guess which we describe below.

Setup. On input of a security parameter k, the challenger first runs algorithm Setup to obtain the system public parameters $\mathcal{P}$. Then, the challenger runs algorithm Issuer-Setup once or multiple times to obtain a set of credential issuers $\mathcal{I} = \{I_1, \ldots, I_N\}$. Finally, the challenger gives to the adversary the public parameters $\mathcal{P}$ as well as the public keys of the different trusted authorities included in $\mathcal{I}$.

Phase-1. The adversary performs a polynomial number of oracle queries adaptively i.e. each query may depend on the replies to the previously performed queries.

Challenge. Once the adversary decides that Phase-1 is over, it gives to the challenger two equal length messages M_0, M_1 and a policy Pol_{ch} on which it wishes to be

challenged. The challenger picks at random $b \in \{0,1\}$, then runs algorithm PolEnc on input of the tuple (M_b, Pol_{ch}), and finally returns the resulting ciphertext C_{ch} to the adversary.

Phase-2. The adversary performs again a polynomial number of adaptive oracle queries.

Guess. The adversary outputs a guess b', and wins the game if $b = b'$.

During Phase-1 and Phase-2, the adversary may perform queries to two oracles controlled by the challenger. On one hand, a credential generation oracle denoted **CredGen-O**. On the other hand, a policy-base decryption oracle denoted **PolDec-O**. While the oracles are executed by the challenger, their input is specified by the adversary. The two oracles are defined as follows:

- **CredGen-O**. On input of a credential $I_\kappa \in \mathcal{T}$ and an assertion $A \in \{0,1\}^*$, run algorithm CredGen on input of the tuple (I_κ, A) and return the resulting credential $\varsigma(R_\kappa, A)$.
- **PolDec-O**. On input of $C \in \mathcal{C}$, a policy Pol and a set of indices $\{j_1, \ldots, j_m\}$, first run algorithm CredGen multiple times to obtain the qualified set of credentials $\varsigma_{j_1,\ldots,j_m}(Pol)$, then run algorithm PolDec on input of the tuple $(C, Pol, \varsigma_{j_1,\ldots,j_m}(Pol))$ and return the resulting output.

The oracle queries made by the adversary during Phase-1 and Phase-2 are subject to two restrictions. On one hand, the adversary is not allowed to obtain a qualified set of credentials for the challenge policy Pol_{ch}. On the other hand, he is not allowed to perform a query to oracle PolDec-O on a tuple $(C, Pol, \{j_1, \ldots, j_m\})$ such that $\varphi_{j_1,\ldots,j_m}(C, Pol) = \varphi_{j_1,\ldots,j_m}(C_{\text{ch}}, Pol_{\text{ch}})$. In fact, in the policy-based setting, for an encrypted message with respect to a policy with disjunctions, there is more than one possible qualified set of credentials that can be used to perform the decryption. That is, forbidding the adversary from making decryption queries on the challenge tuple $(C_{\text{ch}}, Pol_{\text{ch}})$, as in the IND-ID-CCA model, is not sufficient anymore. Indeed, we may have tuples such that $(C, Pol) \neq (C_{\text{ch}}, Pol_{\text{ch}})$ while $\varphi_{j_1,\ldots,j_m}(C, Pol) = \varphi_{j_1,\ldots,j_m}(C_{\text{ch}}, Pol_{\text{ch}})$. Decryption queries on such tuples should then be forbidden as well.

The game described above is denoted IND-Pol-CCA. A formal definition of chosen ciphertext security for PBE schemes is given below. As usual, a real function g is said to be negligible if $g(k) \leq \frac{1}{f(k)}$ for any polynomial f.

Definition A.1 The advantage of an adversary $\mathcal{A}$ in the IND-Pol-CCA game is defined to be the quantity $\text{Adv}_{\mathcal{A}} = |Pr[b = b'] - \frac{1}{2}|$. A PBE scheme is IND-Pol-CCA secure if no probabilistic polynomial time adversary has a non-negligible advantage in the IND-Pol-CCA game.

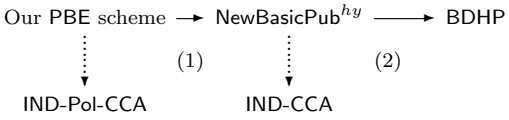
We define $\varphi_{j_1,\ldots,j_m}(C, Pol)$ to be the information from the ciphertext C and the policy Pol that is required to correctly perform the decryption of C with respect to Pol using the qualified set of credentials $\varsigma_{j_1,\ldots,j_m}(Pol)$. A concrete example is given when describing our PBE scheme.

A.2 Security Results for our Policy-Based Encryption Scheme

In the following, we show that our policy-based encryption scheme (denoted PBE) is IND-Pol-CCA secure in the random oracle model defined in [3].

Theorem A.2 *Our* PBE *scheme is* IND-Pol-CCA *secure in the random oracle model under the assumption that BDHP is hard.*

Proof. Theorem A.2 follows from a sequence of reduction arguments that are summarized in the following diagram:



(i) Lemma A.3 shows that an IND-Pol-CCA attack on our PBE scheme can be converted into an IND-CCA attack on the NewBasicPub^{hy} scheme described in Gal05.

(ii) In [10], algorithm NewBasicPub^{hy} is shown to be IND-CCA secure in the random oracle model under the assumption that BDHP is hard.

Lemma A.3 stated below uses a function $F(.)$ having a rather unaesthetic expression. Computing $F(q_c, q_d, q_0, N, m_{\vee\wedge}, m_{\vee}, m_{\wedge})$ relies on computing the quantity $\Upsilon(X, m_{\vee\wedge}, m_{\vee}, m_{\wedge})$, which is defined to be the total number of 'minimal' policies written in CDNF, given the upper-bounds $(m_{\vee\wedge}, m_{\vee}, m_{\wedge})$ and X possible credential-based conditions. Computing $\Upsilon(X, m_{\vee\wedge}, m_{\vee}, m_{\wedge})$ is similar, but not exactly the same as the problems of computing the number of monotone boolean functions of n variables (Dedekind's Problem [15]) and computing the number of antichains on a set $\{1, \ldots, n\}$ [13]. As opposed to these problems, the order of the terms must be taken into consideration when dealing with our policies. This is a typical, yet interesting, 'counting' problem. However, as we do not address exact security in this paper, we do not elaborate more on the details (improving the tightness of the reductions is left to future research work).

Lemma A.3 *Let* $\mathcal{A}^{\circ}$ *be an* IND-Pol-CCA *adversary with advantage* $Adv_{\mathcal{A}^{\circ}} \geq \epsilon$ *when attacking our* PBE *scheme. Assume that* $\mathcal{A}^{\circ}$ *has running time* $t_{\mathcal{A}^{\circ}}$ *and makes at most* q_c *queries to oracle* CredGen-O, q_d *queries to oracle* PolDec-O *as well as* q_0 *queries to oracle* H_0. *Then, there exists an* IND-CCA *adversary* $\mathcal{A}^{\bullet}$ *the advantage of which, when attacking the* NewBasicPub^{hy} *scheme, is such that* $Adv_{\mathcal{A}^{\bullet}} \geq F(q_c, q_d, q_0, N, m_{\vee\wedge}, m_{\vee}, m_{\wedge}).\epsilon$. *Its running time is* $t_{\mathcal{A}^{\bullet}} = O(t_{\mathcal{A}^{\circ}})$.

Due to space limitation, proof of Lemma A.3 is given in the full version of this paper.

Note. In the particular case where $N = m_{\vee\wedge} = m_{\vee} = m_{\wedge} = 1$, our PBE scheme is equivalent to the New-FullIdent scheme of [10]. In this case, our result match Result 5 of [10].

□