Policy-Based Cryptography and Applications ⋆

Walid Bagga , Refik Molva

Institut Eurécom
Corporate Communications
2229, route des Crêtes B.P. 193
06904 Sophia Antipolis (France)
{bagga,molva}@eurecom.fr

Abstract. In this paper, we formulate the concept of *policy-based cryptography* which makes it possible to perform policy enforcement in large-scale open environments like the Internet, with respect to the data minimization principle according to which only strictly necessary information should be collected for a given purpose. We use existing cryptographic primitives based on bilinear pairings over elliptic curves to develop concrete policy-based encryption and signature schemes which allow performing relatively efficient encryption and signature operations with respect to policies formalized as monotonic logical formulae. we illustrate the properties of our policy-based cryptographic schemes through the description of three application scenarios.

Keywords: Policy, Authorization, Credentials, Privacy, ID-based Cryptography

1 Introduction

In open computing environments like the Internet, many interactions may occur between entities from different security domains without pre-existing trust relationships. Such interactions may require the exchange of sensitive resources which need to be carefully protected through clear and concise policies. A policy specifies the constraints under which a specific action can be performed on a certain sensitive resource. An increasingly popular approach for authorization in distributed systems consists in defining conditions which are fulfilled by digital credentials. A digital credential is basically a digitally signed assertion by a trusted authority (credential issuer) about a specific user (credential owner). It describes one or multiple properties of the user that are validated by the trusted authority. It is generated using the trusted authority's private key and can be verified using its public key.

Consider the following scenario: a user named Bob controls a sensitive resource denoted 'res', and for a specific action denoted 'act' he defines a policy denoted 'pol' which specifies the conditions under which 'act' may be performed on 'res'. Policy 'pol' is fulfilled by a set of credentials generated by one or multiple trusted authorities. In order for a user named Alice to be authorized to perform 'act' on 'res', she has to prove her compliance to Bob's policy i.e. she has to prove that she possesses a minimal

⋆ The work reported in this paper is supported by the IST PRIME project and by Institut Eurécom; however, it represents the view of the authors only.

set of credentials that is required by 'pol' to permit action 'act' on 'res'. In standard credentials systems like $X.509$, Alice needs first to request the credentials from the appropriate trusted authorities. Then, Alice has to show her credentials to Bob who verifies their validity using the public keys of the issuing trusted authorities. Bob authorizes Alice to perform 'act' on 'res' if and only if he receives a set of valid credentials satisfying 'pol'. Such scenario does not meet the *data minimization* requirement (called the *data quality principle* in OECD guidelines [8]) according to which only strictly necessary information should be collected for a given purpose. In fact, the standard approach allows Bob, on one hand, to enforce his policy i.e. to get a proof that Alice is compliant to his policy before authorizing her to perform the requested action on the specified sensitive resource. On the other hand, it allows him to collect additional 'out-of-purpose' information on Alice's specific credentials.

In this paper, we formulate the concept of *policy-based cryptography* which allows to perform policy enforcement while respecting the data minimization principle. Such 'privacy-aware' policy enforcement is enabled by two cryptographic primitives: *policy-based encryption* and *policy-based signature*. Intuitively, policy-based encryption allows to encrypt data according to a policy so that only entities fulfilling the policy are able to successfully perform the decryption and retrieve the plaintext data, whereas policy-based signature allows to generate a digital signature on data with respect to a policy so that only entities satisfying the policy are able to generate a valid signature.

Our cryptography-based policy enforcement mechanisms manipulate policies that are formalized as monotonic logical expressions involving complex disjunctions and conjunctions of conditions. Each condition is fulfilled by a specific credential issued by a certain trusted authority. Such policy model allows multiple trusted authorities to participate to the authorization process which makes it, on one hand, more realistic because each authority should be responsible for a specific, autonomous and limited administrative domain, and on the other hand, more trustworthy compared with models relying on a centralized trusted authority (which could be seen as a single point of failure) to issue the required credentials. Furthermore, in contrast to the traditional approach where credentials are revealed during policy compliance proofs, our credentials have to be kept secret by their owners. They are used to perform policy-based decryption and policy-based signature operations. We note that the idea of using secret credentials as decryption keys has already been used or at least mentioned in the literature, especially in the contexts of access control and trust negotiation systems [3, 7, 15, 12, 9].

We use existing cryptographic primitives from bilinear pairings on elliptic curves to construct concrete policy-based cryptographic schemes. In fact, our credentials system is based on the short signature scheme defined in [4], our policy-based encryption scheme extends the ID-based encryption scheme described in [3] and our policy-based signature scheme extends the ID-based ring signatures given in [13, 18]. Our algorithms offer a more elegant and efficient way to handle complex authorization structures than the widely used naive approach based on onion-like encryptions to deal with conjunctions (ANDs) and multiple encryptions to deal with disjunctions (ORs). Apart from performance considerations, our policy-based cryptographic primitives have many interesting applications in different critical contexts in today's Internet such as access control, sticky privacy policies, trust establishment, and automated trust negotiation.

The sequel of the paper is organized as follows: we provide in Section 2 a formal model for policy-based cryptography. Moreover, we give formal definitions for policy-based encryption and signature schemes. In Section 3, we describe our concrete policy-based encryption and signature schemes. We briefly discuss their efficiency in Section 4 and analyze their security properties in Section 5. In Section 6, we illustrate the privacy properties of our policy-based primitives. In Section 7, we discuss related work before concluding in Section 8.

2 Model

In this section, we formulate the concept of policy-based cryptography. We first describe the policy-based cryptosystem setup procedure. We then describe the policy model and define the related terminology. We finally provide formal definitions for policy-based encryption and policy-based signature.

2.1 System Setup

A policy-based cryptosystem setup procedure is specified by two randomized algorithms PBC-Setup and TA-Setup which we describe below.

PBC-Setup. On input of a security parameter k, this algorithm generates a set of public parameters, denoted $\mathcal{P}$, which specifies the different groups and public functions that will be used by the system procedures and participants. Furthermore, it includes a description of a message space denoted $\mathcal{M}$, a ciphertext space denoted $\mathcal{C}$, and a signature space denoted $\mathcal{S}$. We assume that the set of parameters $\mathcal{P}$ is publicly known so that we do not need to explicitly provide it as input to subsequent policy-based procedures.

TA-Setup. Each trusted authority TA uses this algorithm to generate a secret master-key s and a corresponding public key R. We assume that a set of trusted authorities denoted $\mathcal{T}$ is publicly known and thus can be referenced by all the system participants i.e. a trustworthy value of the public key of each trusted authority included in $\mathcal{T}$ is known by the system participants. At any time, a new trusted authority may be added to $\mathcal{T}$.

2.2 Policy Model

In the context of this paper, we define an assertion to be a declaration about a subject, where a subject is an entity (either human or computer) that has an identifier in some security domain. An assertion can convey information about the subject's attributes, properties, capabilities, etc. The representation of assertions being out of the scope of this paper, they will be simply encoded as binary strings. We define a credential to be an assertion which validity is certified by a trusted authority through a signature procedure. A trusted authority is basically 'trusted' for not issuing credentials corresponding to invalid assertions. Whenever a trusted authority $TA \in \mathcal{T}$ is asked to sign an assertion $A \in \{0,1\}^*$, it first checks the validity of A. If A is valid, then TA executes algorithm CredGen defined below and returns the output back to the credential requester. Otherwise, TA returns an error message.

CredGen. On input of assertion A and TA's master-key s, this algorithm outputs a credential denoted $\varsigma(R,A)$ where R denotes TA's public key. For every pair (TA,A), the credential $\varsigma(R,A)$ can be generated only by the trusted authority TA using its secret master-key s, while its validity can be checked using its public key R.

We define a policy to be a monotonic logical expression involving conjunctions ($\wedge$) and disjunctions ($\vee$) of 'atomic' conditions. Each condition is defined through a pair $\langle TA,A\rangle$ which specifies an assertion A and indicates the authority TA that is trusted to check and certify A's validity. Let the expression 'user $\leftharpoonup \varsigma(R,A)$' denote the fact that 'user' has been issued credential $\varsigma(R,A)$ and let the expression 'user $\rightleftharpoons \langle TA,A\rangle$' denote the fact that 'user' fulfills condition $\langle TA,A\rangle$.Then, we state the following property

$$\text{user} \rightleftharpoons \langle TA,A\rangle \;\Leftrightarrow\; \text{user} \leftharpoonup \varsigma(R,A) \tag{1}$$

As every statement in logic consisting of a combination of multiple $\wedge$ and $\vee$, a policy can be written in either conjunctive normal form (CNF) or in disjunctive normal form (DNF). In order to address these two normal forms, a policy denoted 'pol' will be written in conjunctive-disjunctive normal form (CDNF) (defined in [15])

$$\text{pol} = \wedge_{i=1}^{m}[\vee_{j=1}^{m_i}[\wedge_{k=1}^{m_{i,j}}\langle TA_{i,j,k},A_{i,j,k}\rangle]]$$

Thus, policies expressed in CNF form are such that $m_{i,j}=1$ for all i,j, while policies expressed in DNF form are such that $m=1$.

Given $j_i \in \{1,\ldots,m_i\}$ for all $i \in \{1,\ldots,m\}$, we define $\varsigma_{j_1,\ldots,j_m}(\text{pol})$ to be the set of credentials $\{\{\varsigma(R_{i,j_i,k},A_{i,j_i,k})\}_{1\le k\le m_{i,j_i}}\}_{1\le i\le m}$. Let the expression 'user $\leftharpoonup \varsigma_{j_1,\ldots,j_m}(\text{pol})$' denote the fact that 'user' has been issued all the credentials included in $\varsigma_{j_1,\ldots,j_m}(\text{pol})$ i.e.

$$\forall\, i \in \{1,\ldots,m\}, \forall\, k \in \{1,\ldots,m_{i,j_i}\}, \text{user} \leftharpoonup \varsigma(R_{i,j_i,k},A_{i,j_i,k})$$

Let the expression 'user $\rightleftharpoons$ pol', for $\text{pol} = \wedge_{i=1}^{m}[\vee_{j=1}^{m_i}[\wedge_{k=1}^{m_{i,j}}\langle TA_{i,j,k},A_{i,j,k}\rangle]]$, denote the fact that 'user' fulfills (satisfies) policy 'pol'. Property (1) leads to the following

$$\text{user} \rightleftharpoons \text{pol} \;\Leftrightarrow\; \forall\, i \in \{1,\ldots,m\}, \exists\; j_i \in \{1,\ldots,m_i\} : \text{user} \leftharpoonup \varsigma_{j_1,\ldots,j_m}(\text{pol}) \tag{2}$$

Informally, we may say that the set of credentials $\varsigma_{j_1,\ldots,j_m}(\text{pol})$ fulfills policy 'pol'.

2.3 Policy-Based Encryption

A policy-based encryption scheme (denoted PBE) consists of two randomized algorithms: PolEnc and PolDec which we describe below.

PolEnc. On input of message m and policy pol_A, this algorithm returns a ciphertext c which represents the message m encrypted according to policy pol_A.

PolDec. On input of ciphertext c, policy pol_A and a set of credentials $\varsigma_{j_1,\ldots,j_a}(\text{pol}_\text{A})$, this algorithm returns a message m.

Algorithms PolEnc and PolDec have to satisfy the standard consistency constraint i.e.

$$c = \text{PolEnc}(m,\text{pol}_\text{A}) \;\Rightarrow\; \text{PolDec}(c,\text{pol}_\text{A},\varsigma_{j_1,\ldots,j_a}(\text{pol}_\text{A})) = m$$

2.4 Policy-Based Signature

A policy-based signature scheme (denoted PBS) consists of two randomized algorithms: PolSig and PolVrf which we describe below.

PolSig. On input of message m, policy pol_B and a set of credentials $\varsigma_{j_1,\ldots,j_b}(\text{pol}_\text{B})$, this algorithm returns a signature σ which represents the signature on message m according to policy pol_B.

PolVrf. On input of message m, policy pol_B and signature σ, this algorithm returns $\top$ (for 'true') if σ is a valid signature on m according to policy pol_B. Otherwise, it returns $\perp$ (for 'false').

Algorithms PolSig and PolVrf have to satisfy the standard consistency constraint i.e.

$$\sigma = \mathsf{PolSig}(m, \text{pol}_\text{B}, \varsigma_{j_1,\ldots,j_b}(\text{pol}_\text{B})) \;\Rightarrow\; \mathsf{PolVrf}(m, \text{pol}_\text{B}, \sigma) = \top$$

3 Policy-Based Cryptography from Bilinear Pairings

In this section, we describe concrete policy-based encryption and signature schemes based on bilinear pairings over elliptic curves.

3.1 System Setup

We define algorithm BDH-Setup to be a bilinear Diffie-Hellman parameter generator satisfying the BDH assumption as this has been formally defined in [3]. Thus, on input of a security parameter k, algorithm BDH-Setup generates a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, e)$ where the map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear pairing, $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, *)$ are two groups of the same order q, where q is determined by the security parameter k. We recall that a bilinear pairing satisfies the following three properties:

1. Bilinear: for $Q, Q' \in \mathbb{G}_1$ and for $a, b \in \mathbb{Z}_q^*$, $e(a \cdot Q, b \cdot Q') = e(Q, Q')^{ab}$
2. Non-degenerate: $e(P, P) \neq 1$ and therefore it is a generator of $\mathbb{G}_2$
3. Computable: there exists an efficient algorithm to compute $e(Q, Q')$ for all $Q, Q' \in \mathbb{G}_1$

The tuple $(q, \mathbb{G}_1, \mathbb{G}_2, e)$ is such that the mathematical problems defined below are such that there is no polynomial time algorithms to solve them with non-negligible probability.

- Discrete Logarithm Problem (DLP). Given $Q, Q' \in \mathbb{G}_1$ such that $Q' = x \cdot Q$ for some $x \in \mathbb{Z}_q^*$: find x
- Bilinear Pairing Inversion Problem (BPIP). Given $Q \in \mathbb{G}_1$ and $e(Q, Q')$ for some $Q' \in \mathbb{G}_1$: find Q'
- Bilinear Diffie-Hellman Problem (BDHP). Given $(P, a \cdot P, b \cdot P, c \cdot P)$ for $a, b, c \in \mathbb{Z}_q^*$: compute $e(P, P)^{abc}$

The hardness of the problems defined above 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 [10]. As we merely apply these mathematical primitives in this paper, we refer to [17] for further details.

Our PBC-Setup, TA-Setup and CredGen algorithms are described below.

PBC-Setup. Given a security parameter k, do the following:

1. Run algorithm BDH-Setup on input k to generate output $(q, \mathbb{G}_1, \mathbb{G}_2, e)$
2. Pick at random a generator $P \in \mathbb{G}_1$
3. For some chosen $n \in \mathbb{N}^*$, let $\mathcal{M} = \{0,1\}^n$
4. Let $\mathcal{C} = \mathbb{G}_1 \times (\{0,1\}^n)^* \times \mathcal{M}$ and $\mathcal{S} = (\mathbb{G}_2)^* \times \mathbb{G}_1$
5. Define five hash functions: $H_0 : \{0,1\}^* \to \mathbb{G}_1$, $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \to \{0,1\}^n$, $H_3 : \{0,1\}^n \to \{0,1\}^n$ and $H_4 : \{0,1\}^* \to \mathbb{Z}_q^*$
6. Set the system public parameters to be $\mathcal{P} = (q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, H_0, H_1, H_2, H_3, H_4)$

TA-Setup. Each trusted authority TA picks at random a master-key $s \in \mathbb{Z}_q^*$ and keeps it secret while publishing the corresponding public key $R = s \cdot P$.

CredGen. Given a valid assertion A and TA's master-key s, this algorithm outputs the credential $\varsigma(R,A) = s \cdot H_0(A)$.

3.2 Policy-Based Encryption

Our policy-based encryption scheme can be seen as a kind of extension or generalization of the Boneh-Franklin ID-based encryption scheme given in [3]. Let pol_A denote a policy of the form $\wedge_{i=1}^{a}[\vee_{j=1}^{a_i}[\wedge_{k=1}^{a_{i,j}}\langle TA_{i,j,k}, A_{i,j,k}\rangle]]$, we describe our PolEnc algorithm below.

PolEnc. Given message m and policy pol_A, do the following:

1. Pick randomly $t_i \in \{0,1\}^n$ for $i = 1, \ldots, a$
2. Compute $t = \oplus_{i=1}^{a} t_i$, then compute $r = H_1(m \| t \| \text{pol}_\text{A})$ and $U = r \cdot P$
3. For $i = 1, \ldots, a$, for $j = 1, \ldots, a_i$,
   (a) Compute $g_{i,j} = \prod_{k=1}^{a_{i,j}} e(R_{i,j,k}, H_0(A_{i,j,k}))$
   (b) Compute $v_{i,j} = t_i \oplus H_2(g_{i,j}^r \| i \| j)$
4. Compute $w = m \oplus H_3(t)$
5. Set the ciphertext to be $c = (U, [v_{i,1}, v_{i,2}, \ldots, v_{i,a_i}]_{1 \le i \le a}, w)$

The intuition behind the encryption procedure described above is as follows: each conjunction of conditions $\wedge_{i,j} = \wedge_{k=1}^{a_{i,j}}\langle TA_{i,j,k}, A_{i,j,k}\rangle$ is associated to a kind of mask we denote $\mu_{i,j} = H_2(g_{i,j}^r \| i \| j)$. For each index i, a randomly chosen key t_i is associated to the disjunction $\vee_i = \vee_{j=1}^{a_i} \wedge_{i,j}$. Each t_i is encrypted a_i times using each of the masks $\mu_{i,j}$. Thus, it is sufficient to compute any one of the masks $\mu_{i,j}$ in order to be able to retrieve the key t_i. In order to be able to perform the decryption procedure successfully, an entity needs to retrieve all the keys t_i. Our PolDec algorithm is described below.

PolDec. Given the ciphertext $c = (U, [v_{i,1}, v_{i,2}, \ldots, v_{i,a_i}]_{1 \le i \le a}, w)$, policy pol_A and the set of credentials $\varsigma_{j_1,\ldots,j_a}(\text{pol}_\text{A})$, do the following:

1. For $i = 1, \ldots, a$,
   (a) Compute $\tilde{g}_{i,j_i} = e(U, \sum_{k=1}^{a_{i,j_i}} \varsigma(R_{i,j_i,k}, A_{i,j_i,k}))$
   (b) Compute $\tilde{t}_i = v_{i,j_i} \oplus H_2(\tilde{g}_{i,j_i} \| i \| j_i)$
2. Compute $\tilde{m} = w \oplus H_3(\oplus_{i=1}^{a} \tilde{t}_i)$
3. Compute $\tilde{U} = H_1(\tilde{m} \| \oplus_{i=1}^{a} \tilde{t}_i \| \text{pol}_\text{A}) \cdot P$
4. If $\tilde{U} = U$, then return message $\tilde{m}$, otherwise return $\perp$ (for 'error')

Our algorithms PolEnc and PolDec satisfy the standard consistency constraint. In fact, thanks to the properties of bilinear pairings, it is easy to check that for every index i, $\tilde{g}_{i,j_i} = g_{i,j_i}^r$.

3.3 Policy-Based Signature

Our policy-based signature scheme is a kind of extension of the ID-based ring signature schemes given in [18, 13]. In an ID-based ring signature, the signer sets up a finite set of identities including his identity. The set of identities represents the set of all possible signers i.e. ring members. A valid signature will convince the verifier that the signature is generated by one of the ring members, without revealing any information about which member has actually generated the signature. Let $\mathrm{pol_B}$ denote a policy of the form $\wedge_{i=1}^{b}[\vee_{j=1}^{b_i}[\wedge_{k=1}^{b_{i,j}}\langle TA_{i,j,k}, A_{i,j,k}\rangle]]$, we describe our PolSig algorithm below.

PolSig. Given message m, policy $\mathrm{pol_B}$ and the set of credentials $\varsigma_{j_1,\ldots,j_b}(\mathrm{pol_B})$, do the following:

1. For $i = 1,\ldots,b$,
   (a) Pick randomly $Y_i \in \mathbb{G}_1$, then compute $x_{i,j_i+1} = e(P, Y_i)$
   (b) For $l = j_i+1,\ldots,b_i,1,\ldots,j_i-1 \bmod (b_i+1)$,
      i. Compute $\tau_{i,l} = \prod_{k=1}^{b_{i,l}} e(R_{i,l,k}, H_0(A_{i,l,k}))$
      ii. Pick randomly $Y_{i,l} \in \mathbb{G}_1$, then compute $x_{i,l+1} = e(P, Y_{i,l}) * \tau_{i,l}^{H_4(m\|x_{i,l}\|\mathrm{pol_B})}$
   (c) Compute $Y_{i,j_i} = Y_i - H_4(m\|x_{i,j_i}\|\mathrm{pol_B}) \cdot (\sum_{k=1}^{b_{i,j_i}} \varsigma(R_{i,j_i,k}, A_{i,j_i,k}))$
2. Compute $Y = \sum_{i=1}^{b}\sum_{j=1}^{b_i} Y_{i,j}$
3. Set the signature to be $\sigma = ([x_{i,1}, x_{i,2},\ldots,x_{i,b_i}]_{1\le i\le b}, Y)$

The intuition behind the signature procedure described above is as follows: each conjunction of conditions $\wedge_{i,j} = \wedge_{k=1}^{b_{i,j}}\langle TA_{i,j,k}, A_{i,j,k}\rangle$ is associated to a tag $\tau_{i,j}$. For each index i, the set of tags $\{\tau_{i,j}\}_j$ corresponds to a set of ring members. The signature key associated to the tag $\tau_{i,j}$ corresponds to the set of credentials $\{\varsigma(R_{i,j,k}, A_{i,j,k})\}_{1\le k\le b_{i,j}}$. Our PolVrf algorithm is described below.

PolVrf. Given message m, policy $\mathrm{pol_B}$ and the signature $\sigma = ([x_{i,1}, x_{i,2},\ldots,x_{i,b_i}]_{1\le i\le b}, Y)$, do the following:

1. Compute $z_1 = \prod_{i=1}^{b}[\prod_{j=1}^{b_i} x_{i,j}]$
2. For $i = 1,\ldots,b$ and for $j = 1,\ldots,b_i$, compute $\tau_{i,j} = \prod_{k=1}^{b_{i,j}} e(R_{i,j,k}, H_0(A_{i,j,k}))$
3. Compute $z_2 = e(P,Y) * \prod_{i=1}^{b}[\prod_{j=1}^{b_i} \tau_{i,j}^{H_4(m\|x_{i,j}\|\mathrm{pol_B})}]$
4. If $z_1 = z_2$, then return $\top$, otherwise return $\bot$

Our algorithms PolSig and PolVrf satisfy the standard consistency constraint. In fact, it is easy to check that for $i = 1,\ldots,b$ and $j = 1,\ldots,b_i$, the following holds

$$\tau_{i,j}^{H_4(m\|x_{i,j}\|\mathrm{pol_B})} = x_{i,j+1} * e(P, Y_{i,j})^{-1} \text{ (where } x_{i,b_i+1} = x_{i,1})$$

Let $\lambda = e(P,Y)$, then the following holds

$$z_2 = \lambda * \prod_{i=1}^{b}[\prod_{j=1}^{b_i} \tau_{i,j}^{H_4(m\|x_{i,j}\|\mathrm{pol_B})}] = \lambda * \prod_{i=1}^{b}[\prod_{j=1}^{b_i-1} x_{i,j+1} * e(P, Y_{i,j})^{-1} * x_{i,1} * e(P, Y_{i,b_i})^{-1}]$$

$$= \lambda * \prod_{i=1}^{b}[\prod_{j=1}^{b_i} x_{i,j} * \prod_{j=1}^{b_i} e(P, Y_{i,j})^{-1}] = \lambda * [\prod_{i=1}^{b}\prod_{j=1}^{b_i} x_{i,j}] * [e(P, \sum_{i=1}^{n}\sum_{j=1}^{b_i} Y_{i,j})]^{-1} = \lambda * z_1 * \lambda^{-1}$$

4 Efficiency

The essential operation in pairings-based cryptography is pairing computation. Although such operation can be optimized as explained in [1], it still have to be minimized. Table 1 summarizes the computational costs of our policy-based encryption and signature schemes in terms of pairing computations.

PolEnc	PolDec	PolSig	PolVrf
$\sum_{i=1}^{a}\sum_{j=1}^{a_i} a_{i,j}$	a	$\sum_{i=1}^{b} b_i + \sum_{i=1}^{b}\sum_{j\neq j_i} b_{i,j}$	$1+\sum_{i=1}^{b}\sum_{j=1}^{b_i} b_{i,j}$

Table 1. Computational costs in terms of pairing computations

Notice that for all i, j, k, the pairing $e(R_{i,j,k}, H_0(A_{i,j,k}))$ involved in algorithms PolSig, PolEnc and PolVrf does not depend on the message m. Thus, it can be pre-computed, cached and used in subsequent signatures, encryptions and verifications involving the condition $\langle TA_{i,j,k}, A_{i,j,k}\rangle$.

Let l_i be the bit-length of the bilinear representation of an element of group $\mathbb{G}_i$ for $i = 1, 2$. Then, the bit-length of a ciphertext produced by our encryption algorithm is equal to $l_1 + (1+\sum_{i=1}^{a} a_i).n$, and the bit-length of a signature produced by our signature algorithm is equal to $(\sum_{i=1}^{b} b_i).l_2 + l_1$.

The sizes of the ciphertexts and the signatures generated by our policy-based encryption and signature algorithms respectively is highly dependent on the values $\sum_{i=1}^{a} a_i$ and $\sum_{i=1}^{b} b_i$, which then need to be minimized. For this reason, we require that the representation of a policy $\wedge_{i=1}^{m}[\vee_{j=1}^{m_i}[\wedge_{k=1}^{m_{i,j}}\langle TA_{i,j,k}, A_{i,j,k}\rangle]]$ minimizes the sum $\sum_{i=1}^{m} m_i$.

5 Security

In this section, we focus on the security properties of our policy-based cryptographic schemes. Informally, a policy-based encryption scheme must satisfy the semantic security property i.e. an adversary who does not fulfill the encryption policy learns nothing about the encrypted message from the corresponding ciphertext. While a policy-based signature scheme must satisfy, on one hand, the existential unforgeability property i.e. an adversary cannot generate a valid signature without having access to a set of credentials fulfilling the signature policy, and, on the other hand, the credentials ambiguity property i.e. while the verifier is able to check the validity of the signature, there is no way for him to know which set of credentials has been used to generate it. A formal analysis of these security properties requires, in addition to the specification of attacks' goals, the establishment of adequate attack models i.e. chosen ciphertext attacks for policy-based encryption and chosen message attacks for policy-based signature. Because of the lack of space, we only point out, in this paper, the security properties of our schemes and provide intuitive and rather heuristic proofs of our claimed security properties. Our security analysis relies on the random oracle model as defined and discussed in [2].

5.1 Policy-Based Encryption

Claim. Our policy-based encryption scheme is semantically secure in the random oracle model under the assumption that BDHP is hard.

Given a policy $\text{pol}_\text{A} = \wedge_{i=1}^{a}[\vee_{j=1}^{a_i}[\wedge_{k=1}^{a_{i,j}}\langle TA_{i,j,k}, A_{i,j,k}\rangle]]$, we provide in the following a proof sketch of our claim through a step-by-step approach going from simple cases to more general ones.

Case 1. Assume that $a = 1$, $a_1 = 1$ and $a_{1,1} = 1$ i.e. $\text{pol}_\text{A} = \langle TA_{1,1,1}, A_{1,1,1}\rangle$. Here, our policy-based encryption algorithm is reduced to an ID-based encryption algorithm similar to algorithm FullIdent defined in [3]. Thus, we can define a game between a challenger and an adversary and run a corresponding simulation proving that our algorithm is secure as long as BDHP is hard. The game we may define is similar to the one described in Section 2 of [3]. The only difference is in the definition of extraction queries. In [3], an extraction query allows the adversary to get the credential corresponding to any specified identity ID_i, with the natural restriction that he does not get the credential corresponding to the identity ID_i^* on which he is challenged. As we deal with multiple trusted authorities, an extraction query in our game should allow the adversary to get the credential corresponding to any pair $(TA_{i,j,k}, A_{i,j,k})$ he specifies, with the natural restriction that he does not get the credential corresponding to the pair $(TA_{i,j,k}^*, A_{i,j,k}^*)$ on which he is challenged. Notice that the adversary learns nothing about the challenge pair from queries on pairs $(TA_{i,j,k}^*, A_{i,j,k})$ and $(TA_{i,j,k}, A_{i,j,k}^*)$ because the trusted authorities generate their master-keys randomly and independently. Thus, we may conclude that our policy-based encryption algorithm is as secure as FullIdent. The latter is, in fact, proven to be semantically secure against chosen ciphertext attacks in the random oracle model.

Case 2. Assume that $a = 1$, $a_1 = 1$ and $a_{1,1} > 1$ i.e. $\text{pol}_\text{A} = \wedge_{k=1}^{a_{1,1}}\langle TA_{1,1,k}, A_{1,1,k}\rangle$. As for the previous case, we can define a game and run a corresponding simulation proving that our algorithm is secure as long as BDHP is hard. Here, each extraction query should allow the adversary to ask the challenger each time for the credentials corresponding to $a_{1,1}$ pairs of the form $(TA_{i,j,k}, A_{i,j,k})$, instead of a single pair as for the previous case. The only restriction is that the adversary does not get all the credentials corresponding to the set of pairs $\{(TA_{i,j,k}^*, A_{i,j,k}^*)_1, \ldots, (TA_{i,j,k}^*, A_{i,j,k}^*)_{a_{1,1}}\}$ on which he is challenged. The fact that the game defined for the previous simple case allows the adversary to perform an unlimited number of extraction queries, leads to the conclusion that our encryption algorithm remains semantically secure when $a = 1$, $a_1 = 1$ and $a_{1,1} > 1$.

Case 3. Assume that $a = 1$ and $a_1 > 1$ i.e. $\text{pol}_\text{A} = \vee_{j=1}^{a_1}[\wedge_{k=1}^{a_{1,j}}\langle TA_{1,j,k}, A_{1,j,k}\rangle]$. Here, the difference with the previous case is that the ciphertext contains a_1 encryptions of the randomly generated ephemeral key t_1, instead of a single one as for the previous case. The fact that H_2 is a random oracle allows to generate a different uniformly distributed pad for each of the input entries $(g_{1,j}^r, 1, j)$. The semantic security of the Vernam one-time pad leads to the conclusion that our encryption algorithm remains semantically secure when $a = 1$ and $a_1 > 1$.

Case 4. Assume that $a > 1$ (this corresponds to the general case). First of all, notice that for all i, encrypting a_i times the ephemeral key t_i does not weaken its security because the random oracle hash function H_2 outputs different uniformly-distributed

pads for the different input entries $(g_{i,j}^r, i, j)$ so that no pad is used more than one time. Now, we give an intuitive recursive proof of the semantic security of our policy-based encryption scheme. Assume that the encryption is semantically secure if $a = A$ for some A, and consider the case where $a = A+1$. For a given message m, let $c = (U, [v_{i,1}, v_{i,2}, \ldots, v_{i,a_i}]_{1 \leq i \leq p+1}, w = m \oplus H_3(\oplus_{i=1}^{A+1} t_i)$ be the ciphertext generated by our policy-based encryption algorithm. Let $c_{\mathrm{A}} = (U, [v_{i,1}, v_{i,2}, \ldots, v_{i,a_i}]_{1 \leq i \leq A}, w_{\mathrm{A}} = m \oplus H_3(\oplus_{i=1}^{A} t_i))$ and $c_{A+1} = (U, [v_{A+1,1}, v_{A+1,2}, \ldots, v_{A+1,a_{A+1}}], w_{\mathrm{A}} \oplus H_3(t_{A+1}))$. We know that the adversary learns nothing about m from c_{A}. Moreover, that the adversary learns nothing neither about m nor about w_{A} from c_{A+1} thanks to the random oracle assumption. This leads to the fact that the adversary gets no useful information about m from c_{A} and c_{A+1}. As the different ephemeral keys t_i are generated randomly, it is highly improbable that $\oplus_{i=1}^{A} t_i = t_{A+1}$. Because $m \oplus H_3(\oplus_{i=1}^{A+1} t_i)$ is at least as secure as $m \oplus H_3(\oplus_{i=1}^{A} t_i) \oplus H_3(t_{A+1})$, we may conclude that our policy-based encryption algorithm achieves the semantic security property.

5.2 Policy-Based Signature

Claim. Our policy-based signature scheme achieves signature unforgeability in the random oracle model under the assumption that DLP and BPIP are hard.

Given policy $\mathrm{pol}_{\mathrm{B}} = \wedge_{i=1}^{b} [\vee_{j=1}^{b_i} [\wedge_{k=1}^{b_{i,j}} \langle TA_{i,j,k}, A_{i,j,k} \rangle]]$, we give an intuitive proof of our claim similarly to the proof given in [13]: an adversary who does not possess a set of credentials fulfilling $\mathrm{pol}_{\mathrm{B}}$ may try to generate a signature $\sigma = ([x_{i,1}, x_{i,2}, \ldots, x_{i,b_i}]_{1 \leq i \leq b}, Y)$ on a message m according to $\mathrm{pol}_{\mathrm{B}}$ through two possible attacks. On one hand, the adversary chooses the values $x_{i,j}$ for all $1 \leq i \leq b$ and all $1 \leq j \leq b_i$, then tries to compute Y such that σ is valid i.e. the adversary computes Y from the equation

$$e(P,Y) = [\prod_{i=1}^{b} [\prod_{j=1}^{b_i} x_{i,j}]] * [\prod_{i=1}^{b} [\prod_{j=1}^{b_i} \tau_{i,j}^{H_4(m \| x_{i,j} \| \mathrm{pol}_{\mathrm{B}})}]]^{-1}$$

Such attack is equivalent to solving PBIP which is assumed to be hard. On the other hand, the adversary chooses Y and all the values $x_{i,j}$ for $1 \leq i \leq b$ and $1 \leq j \leq b_i$ but the value x_{i_0,j_0} for certain $1 \leq i_0 \leq b$ and $1 \leq j_0 \leq b_{i_0}$, then tries to compute x_{i_0,j_0} such that σ is valid i.e. the adversary solves the equation

$$x_{i_0,j_0} = \xi * \tau_{i_0,j_0}^{H_4(m \| x_{i_0,j_0} \| \mathrm{pol}_{\mathrm{B}})}$$

where $\xi = [\prod_{i \neq i_0} [\prod_{j \neq j_0} x_{i,j}]]^{-1} * e(P,Y) * [\prod_{i \neq i_0} [\prod_{j \neq j_0} \tau_{i,j}^{H_4(m \| x_{i,j} \| \mathrm{pol}_{\mathrm{B}})}]]$. Because H_4 is assumed to be a random oracle, there's no way for the adversary to solve such equation apart from a brute force approach which consists in trying all the elements of $\mathbb{G}_2$. Hence, the probability of forging a signature through this attack is less than $1/q$ which is considered to be negligible.

Claim. Our policy-based signature scheme achieves credentials ambiguity in the random oracle model.

We give an intuitive proof of our claim similarly to the proof given in [13]: for all indices i, Y_i is chosen randomly in $\mathbb{G}_1$ which means that x_{i,j_i} is uniformly distributed

in $\mathbb{G}_2$. Similarly, for all indices i and l, $Y_{i,l}$ is chosen randomly in $\mathbb{G}_1$ which leads to the fact that all $x_{i,l}$ are uniformly distributed in $\mathbb{G}_2$. Thus, given a message m and the signature $\sigma = ([x_{i,1}, x_{i,2}, \ldots, x_{i,b_i}]_{1 \leq i \leq b}, Y)$ on m according to pol_B, σ does not reveal which credentials have been used to generate it.

6 Application Scenarios

Assume that Bob (service provider) controls a sensitive resource 'res', and that for a specific action 'act' on 'res', he defines a policy 'pol' which specifies the conditions under which 'act' may be performed on 'res'. Assume that Alice (service requester) wants to perform action 'act' on 'res'. As a simple example, we assume that Bob's policy is

$$\text{pol}_\text{B} = \langle \text{IFCA}, \text{alice:member} \rangle \wedge [\langle \text{X}, \text{alice:employee} \rangle \vee \langle \text{Y}, \text{alice:employee} \rangle]$$

Here 'IFCA' stands for the International Financial Cryptography Association, while 'X' and 'Y' are two partners of Bob. Bob's policy states that in order for Alice to be authorized to perform action 'act' on 'res', Alice must be a member of IFCA as well as an employee of either partner 'X' or partner 'Y'. We assume, for instance, that Alice is a member of 'IFCA' and works for 'X' i.e. Alice possesses the secret credentials $\varsigma_{\text{IFCA}} = \varsigma(R_{\text{IFCA}}, \text{alice:member})$ and $\varsigma_\text{X} = \varsigma(R_\text{X}, \text{alice:employee})$. In the following, we describe three different policy enforcement scenarios and show how our approach allows performing privacy-aware policy enforcement (with respect to the data minimization principle).

Scenario 1. Assume that 'res' is a PDF file containing a confidential report and assume that Alice wants to have a read access to the report. Here, the only concern of Bob is to ensure that Alice does not read the file if she is not compliant to pol_B. He needs to know neither whether Alice fulfills his policy or not, nor whether she is an employee of X or Y. The standard approach allows Bob to get such 'out-of-purpose' information because Alice has to show her credentials in order to prove her compliance to pol_B, whilst our policy-based cryptographic approach allows to avoid this privacy flaw as follows:

1. First, Bob encrypts the protected file according to policy pol_B i.e. Bob computes $c = \mathsf{PolEnc}(\text{res}, \text{pol}_\text{B})$. Then, he sends c to Alice. Note that practically, Bob does not encrypt res but the session key which encrypts res.
2. Upon receiving c, Alice decrypts it using her secret credentials i.e. Alice computes $\text{res} = \mathsf{PolDec}(c, \text{pol}_\text{B}, \{\varsigma_{\text{IFCA}}, \varsigma_\text{X}\})$

Scenario 1 may be applied to solve the cyclic policy interdependency problem as described in [12, 9]. An additional interesting application of policy-based encryption is the sticky privacy policy paradigm, first defined in [11], according to which the policy that is specified and consented by data subjects at collection, and which governs data usage, holds true throughout the data's lifetime, even when the data is disclosed by one organization to another. Thus, a data subject may encrypt his private data according to a policy reflecting his privacy preferences. The exchange of encrypted privacy-sensitive data ensures that only principals fulfilling the privacy requirements are able to perform

the decryption operation successfully and retrieve the privacy-sensitive data. As an illustrative example, a user Alice may require that a company is a member of either the Better Business Bureau (BBB) or the International Chamber of Commerce (ICC) in order to be able to have access to her professional e-mail address (*alice@X.net*). Thus, Alice may encrypt *alice@X.net* according to her policy

$$\mathrm{pol_A} = \langle \mathrm{BBB}, \text{member:current-year} \rangle \vee \langle \mathrm{ICC}, \text{member:current-year} \rangle$$

Scenario 2. Assume that 'res' is a CD-ROM containing a confidential piece of software and that Alice asks Bob to ship it to her home address. The only useful information for Bob is to know whether Alice is compliant to $\mathrm{pol_B}$ or not. He does not need to know for which company Alice works. While the standard approach obliges Alice to show her employee credential in order to prove her compliance to $\mathrm{pol_B}$, our policy-based cryptographic approach allows to avoid this privacy flaw as follows:

1. First, Bob picks a random challenge nonce n_{ch} and encrypts it according to $\mathrm{pol_B}$ i.e. Bob computes $c = \mathsf{PolEnc}(n_{\mathrm{ch}}, \mathrm{pol_B})$. Then, he sends c to Alice as a 'policy compliance' challenge
2. Upon receiving c, Alice decrypts it using her secret credentials i.e. Alice computes $n_{\mathrm{resp}} = \mathsf{PolDec}(c, \mathrm{pol_B}, \{\varsigma_{\mathrm{IFCA}}, \varsigma_{\mathrm{X}}\})$. Then Alice sends n_{resp} as a response for Bob's challenge
3. Upon receiving n_{resp}, Bob checks whether $n_{\mathrm{resp}} = n_{\mathrm{ch}}$ in which case he authorizes the shipping of the requested CD-ROM to Alice's home address. If Alice does not send her response or if the response is not equal to the challenge nonce, Bob infers that she is not compliant to $\mathrm{pol_B}$ and thus does not authorize the shipping of the requested CD-ROM

Scenario 2 applies either when the action 'act' on the sensitive resource 'res' is different from 'read' or when the communication partners wish to conduct mutli-round transactions during which a party needs to know whether the other is compliant to his policy or not.

Scenario 3. Consider the previous scenario while assuming now that Bob wishes to keep a non-forgeable and/or non-repudiable proof that Alice is compliant to $\mathrm{pol_B}$. In the standard approach, Bob gets all the credentials of Alice allowing her to prove her compliance to $\mathrm{pol_B}$. In this case, the set of received credentials may be seen as a policy compliance proof. In addition to the required proof, Bob knows for which company Alice works. The collection of such 'out-of-purpose' information represents a privacy flaw which could be avoided using our policy-based cryptographic approach as follows:

1. First, Bob picks a random challenge nonce n_{ch} and sends it to Alice
2. Upon receiving the challenge, Alice signs it according to $\mathrm{pol_B}$ using her secret credentials i.e. Alice computes $\sigma = \mathsf{PolSig}(n_{\mathrm{ch}}, \mathrm{pol_B}, \{\varsigma_{\mathrm{IFCA}}, \varsigma_{\mathrm{X}}\})$. Then Alice sends σ to Bob as a response for his challenge
3. Upon receiving σ, Bob checks whether it is a valid signature with respect to $\mathrm{pol_B}$ i.e. Bob checks whether $\mathsf{PolVrf}(n_{\mathrm{ch}}, \mathrm{pol_B}, \sigma) = \top$, in which case Bob authorizes the requested action to be performed (CD-ROM shipping)

Scenario 3 allows a number of interesting value-added services such as accountability i.e. Alice cannot deny being compliant to Bob's policy at certain period in time, service customization i.e. Bob may make a special offers or discounts to customers respecting pol_B at a certain period in time, policy-based single sign-on i.e. based on Alice's poof of compliance to policy pol_B, Alice may get multiple services from Bob's partners (within a federation) without re-proving her compliance to pol_B, etc. Note that the non-repudiation property is due to the fact that the credentials are attached to Alice's name (identifier).

7 Related Work

Many cryptography-based policy enforcement mechanisms have been presented over the years, especially in the context of access control. In [16], Wilkinson et al. show how to achieve trustworthy access control with untrustworthy web servers through standard symmetric and asymmetric cryptographic mechanisms. Their approach allows removing access control responsibilities from web server software which are subject to failure, while delegating access control functionalities to encryption and decryption proxies. Their access control 'expressions' (policies) are described through conjunctions and disjunctions of groups each containing a number of users. They describe how they perform encryption operations and generate decryption keys according to these policies. Their approach remains naive in the sense that they use onion-like encryptions to deal with conjunctions and multiple encryptions to deal with disjunctions. Moreover, they use standard public key cryptography which main drawback consists in dealing with public key certificates. This weakness could be avoided by using identity-based cryptography as formulated by Shamir in [14].

In [7], Chen et al. investigate a number of issues related to the use of multiple authorities in ID-based encryption from bilinear pairings. They present a number of interesting applications of the addition of keys, and show how to perform encryptions according to disjunctions and conjunctions of keys. However, their solution remains restricted to limited disjunctions of keys. In [15], Smart continues the ideas discussed in [7]. He presents an elegant and efficient mechanism to perform encryption according to arbitrary combinations of keys, yet generated by a single trusted authority. Our work could be seen as an extension of [15] in the sense that we use the same policy model while allowing multiple trusted authorities and defining the policy-based signature primitive.

Apart from access control systems, the exchange of digital credentials is an increasingly popular approach for trust establishment in open distributed systems where communications may occur between strangers. In such conditions, the possession of certain credentials may be considered as security or privacy sensitive information. Automated trust negotiation (ATN) allows regulating the flow of sensitive credentials during trust establishment through the definition of disclosure policies. One of the major problems in ATN is called the cyclic policy interdependency which occurs when a communication party is obliged to be the first to reveal a sensitive credential to the other. In [12], Li et al. model the cyclic policy interdependency problem as a 2-party secure function evaluation (SFE) and propose oblivious signature-based envelopes (OSBE) for efficiently solving the FSE problem. Among other schemes, they describe an OSBE

scheme based on ID-based cryptography which is almost similar to our policy-based encryption scheme in the particular case where the considered policy is satisfied by a single credential. Thus, our encryption scheme could be seen as a generalization of the identity-based OSBE scheme.

In [9], Holt et al. introduce the notion of hidden credentials which are similar to our policy-based encryption scheme in that the ability to read a sensitive resource is contingent on having been issued the required credentials. In contrast with OSBE, hidden credentials deal with complex policies expressed as monotonic Boolean expressions. They use onion-like encryptions and multiple encryptions to deal with conjunctions and disjunctions respectively. Their approach remains inefficient in terms of both computational costs and bandwidth consumption (ciphertext size) especially when authorization structures become very complex. While our policy-based encryption and signature schemes are based on publicly known policies, hidden credentials consider the policies as sensitive so that they should never be revealed. Thus, decryptions are performed in a blind way in the sense that the decrypting entity has not only to possess a set of credentials satisfying the encryption policy but also to find the correct combination of credentials corresponding to the policy structure. Very recently, Bradshaw et al. proposed a solution to improve decryption efficiency as well as policy concealment when implementing hidden credentials with sensitive policies [5].

In [6], Brands introduced practical techniques and protocols for designing, issuing and disclosing private credentials. He describes in chapter 3 of [6] a set of showing protocols enabling the credentials owner to selectively disclose properties about them. Brands' approach is data subject-centric, while our approach for privacy focuses on the quality of data exchange during privacy-sensitive transactions. Besides, Brands' credentials are based on standard public key cryptography, whilst our policy-based cryptographic schemes are based on identity-based cryptography from bilinear pairings.

8 Conclusion

In this paper, we formulated the concept of policy-based cryptography which allows performing privacy-aware policy enforcement in open distributed systems like the Internet. We mainly focused on the compliance to the data minimization principle which has been advocated by several privacy protection guidelines and legislations. We defined the policy-based encryption and signature primitives, and we proposed concrete schemes from bilinear pairings. Our algorithms allow handling complex policies in an elegant and relatively efficient manner. Moreover, their properties allow using them in a wide range of applications, from the traditional access control systems to the more sophisticated privacy protection and trust establishment systems. Future research may focus on improving the efficiency of the proposed policy-based schemes and on developing additional policy-based cryptographic primitives. We are currently investigating the real deployment of our policy-based approach in the context of sticky privacy policies. Besides, we are developing formal security models and proofs for policy-based cryptographic schemes.

References

1. P. Barreto, H. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, pages 354–368. Springer-Verlag, 2002.
2. 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.
3. 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.
4. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532. Springer-Verlag, 2001.
5. R. Bradshaw, J. Holt, and K. Seamons. Concealing complex policies with hidden credentials. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 146–157. ACM Press, 2004.
6. S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000.
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. 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.
10. 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.
11. G. Karjoth, M. Schunter, , and M. Waidner. The platform for enterprise privacy practices–privacy-enabled management of customer data. In *2nd Workshop on Privacy Enhancing Technologies (PET 2002)*, volume 2482 of *LNCS*, pages 69–84. Springer-Verlag, April 2002.
12. 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.
13. C. Lin and T. Wu. An identity-based ring signature scheme from bilinear pairings. In *Proceedings of the 18th International Conference on Advanced Information Networking and Applications*. IEEE Computer Society, 2004.
14. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53. Springer-Verlag New York, Inc., 1985.
15. N. Smart. Access control using pairing based cryptography. In *Proceedings CT-RSA 2003*, pages 111–121. Springer-Verlag LNCS 2612, April 2003.
16. T. Wilkinson, D. Hearn, and S. Wiseman. Trustworthy access control with untrustworthy web servers. In *Proceedings of the 15th Annual Computer Security Applications Conference*, page 12. IEEE Computer Society, 1999.
17. Y. Yacobi. A note on the bilinear diffie-hellman assumption. Cryptology ePrint Archive, Report 2002/113, 2002. http://eprint.iacr.org/.
18. F. Zhang and K. Kim. Id-based blind signature and ring signature from pairings. In *ASIACRYPT*, pages 533–547. Springer-Verlag LNCS 2501, 2002.