Constant Size Ciphertexts in Threshold Attribute-Based Encryption

Javier Herranz[1], Fabien Laguillaumie[2], and Carla Ràfols[1]

[1] Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya,
C. Jordi Girona 1-3, Mòdul C3, 08034, Barcelona, Spain.
e-mail: {jherranz,crafols}@ma4.upc.edu
[2] GREYC - Université de Caen Basse-Normandie,
Boulevard du Maréchal Juin, BP 5186, 14032 Caen Cedex, France.
e-mail: fabien.laguillaumie@unicaen.fr

Abstract. Attribute-based cryptography has emerged in the last years as a promising primitive for digital security. For instance, it provides good solutions to the problem of anonymous access control. In a ciphertext-policy attribute-based encryption scheme, the secret keys of the users depend on their attributes. When encrypting a message, the sender chooses which subset of attributes must be held by a receiver in order to be able to decrypt.
All current attribute-based encryption schemes that admit reasonably expressive decryption policies produce ciphertexts whose size depends at least linearly on the number of attributes involved in the policy. In this paper we propose the first scheme whose ciphertexts have constant size. Our scheme works for the threshold case: users authorized to decrypt are those who hold at least t attributes among a certain universe of attributes, for some threshold t chosen by the sender. An extension to the case of weighted threshold decryption policies is possible. The security of the scheme against selective chosen plaintext attacks can be proven in the standard model by reduction to the augmented multi-sequence of exponents decisional Diffie-Hellman (aMSE-DDH) problem.

Keywords: attribute-based encryption, provable security, pairings.

1 Introduction

Encryption is the cryptographic primitive which provides confidentiality to digital communications. In a traditional public key encryption scheme, a message is encrypted with the public key of the intended receiver, who is the only person able to decrypt. This level of confidentiality is enough for many real-life applications, including e-mail and key escrow. However, new situations requiring different cryptographic functionalities appear constantly.

Let us consider for example the case of *anonymous access control*: a system must be accessible only to those who have received the appropriate rights, which are defined by the system administrator. Let us imagine how such a process could be implemented with a standard public key encryption scheme. First, a

user A claims that he is actually user A. Second, the system sends to this user a challenge: a ciphertext computed with the public key of A (obtained from a certification authority, maybe), for some random plaintext. Third, A decrypts and sends back the plaintext. Fourth, if the plaintext is correct, the system checks if user A must have access to the system, and if so, A is accepted. This solution has some weaknesses, the main one being the lack of anonymity, as user A must reveal his identity to the system. Furthermore, each time the system wants to change its access control policy, it has to update the database containing all the users that have the right to access the system.

A more desirable solution, employing encryption, would be as follows. First, in a (possibly interactive, physical) registration process, every potential user receives a secret key that depends on his age, his job, his company, his expertise, etc., in short, on his *attributes*. Later, the system defines his policy for access control as a (monotonic) family of subsets of attributes: attributes in one of such subsets must be held by a user in order to have the right to access the system; in particular, in an extreme case, this policy can contain a unique subset with the unique attribute 'right to access system X'. When a user tries to access the system, he receives as a challenge a ciphertext computed by the system, on a random message, using the current access policy. If the policy changes, the system administrator just has to take into account the new policy for generating the future challenges. A user is able to decrypt the challenge only if his attributes satisfy the considered policy. In this way, if a user answers such a challenge correctly, he does not leak who he is, only the fact that his attributes satisfy the access control policy.

Ciphertext-policy *attribute-based encryption* (ABE for short, from now on) is the cryptographic primitive which precisely realizes the functionality described in the previous paragraph. This primitive can be traced back to identity-based encryption [Sha84] (which can be seen as the particular case of ABE where the policy contains a single subset with a single attribute) and to fuzzy identity-based encryption [SW05] (the particular case of ABE where the policy is always defined by a predetermined threshold t: only users holding at least t attributes can decrypt).

Related work. The first paper dealing explicitly with ABE was [GPSW06]. Two different and complementary notions of ABE were defined there: key-policy ABE, where a ciphertext is associated to a list of attributes, and a secret key is associated to a policy for decryption; and ciphertext-policy ABE, where secret keys are associated to a list of attributes (i.e. credentials of that user) and ciphertexts are associated to policies for decryption. It seems that ciphertext-policy ABE can be more useful for practical applications than key-policy ABE. Another related notion is that of fuzzy identity-based encryption [SW05], which can be seen as a particular case of both key-policy and ciphertext-policy ABE.

A construction of a key-policy ABE scheme was provided in [GPSW06], while the first ciphertext-policy ABE scheme was proposed in [BSW07], but its security was proved in the generic group model. Later, a generic construction to transform a key-policy ABE scheme into a ciphertext-policy ABE scheme was given in

[GJPS08], with the drawback that the size of the ciphertexts is $\mathcal{O}(s^3)$, if s is the number of attributes involved in the decryption policy.

The most efficient ciphertext-policy ABE schemes in terms of ciphertext size can be found in [Wat08,DHMR08], the size of a ciphertext depending linearly on the number of attributes involved in the specific policy for that ciphertext. For example, in the case of (t, s)-threshold decryption policies, where there are s involved attributes and a user can decrypt only if he holds t or more attributes, the size of the ciphertexts in one of the schemes in [Wat08] is $s + \mathcal{O}(1)$, whereas the size of the ciphertexts in the scheme in [DHMR08] is $2(s - t) + \mathcal{O}(1)$. Both schemes admit however general policies (general monotonic access structures) and make use of secret sharing techniques.

All the constructions mentioned so far only achieve security under selective attacks, a model in which the attacker specifies the challenge access structure before the setup phase. The first CP-ABE scheme with full security has appeared very recently [LO+10]. The size of the ciphertexts in this scheme is $2s + \mathcal{O}(1)$.

A concept which is more generic than attribute-based encryption is that of predicate encryption [KSW08]: the decryption policy, chosen by the sender of the message, is hidden in the ciphertext, in such a way that even the receiver gets no information on this policy, other than the fact that his attributes satisfy it or not. Because of this additional strong privacy requirement, current proposals for predicate encryption consider quite simple (not very expressive) policies.

We stress that all the existing proposals for ABE schemes produce ciphertexts whose size depends (at least) linearly on the number of attributes involved in the policy for that ciphertext. An exception is the scheme in [EM+09], where ciphertexts have constant size; but this scheme admits only (s, s)-threshold decryption policies. Note that for this particular threshold case where $t = s$, the scheme in [DHMR08] already achieved constant-size ciphertexts. For more expressive or general decryption policies, no existing scheme has short ciphertexts. This fact can limit the applications of ABE in real life, if we consider for example the case of anonymous access control, with a low bandwidth available for the communication between the user and the system administrator.

An essential feature of ABE schemes is their collusion resistance property, which guarantees that a ciphertext can leak no information about the plaintext to users whose attributes do not satisfy the considered policy, even if the union of the attributes of these colluding users satisfies the policy. This property is essential to guarantee a reasonable level of security in many of the applications of ABE schemes, like anonymous access control or access to encrypted data.

A notion similar to ciphertext-policy ABE but without this collusion resistance property has been considered under different names: policy-based encryption [BM05], cryptographic work flow [AMS06], etc. This notion is actually equivalent to the primitive of dynamic distributed identity-based encryption [CCZ06,DHMR07,DP08,DHMR08]: the sender chooses ad-hoc a set of identities and a monotonic access structure defined on this set; the ciphertext can be decrypted only if users associated to the identities of some subset in the access structure cooperate.

Our contribution. In this paper we propose the first collusion-resistant ABE scheme which produces constant size ciphertexts and which admits reasonably expressive decryption policies. Our scheme is inspired by the dynamic threshold (identity-based) encryption scheme from [DP08], in which the ciphertext's size was constant as well. As we have just said, this scheme directly leads to a weak ABE scheme, without the collusion resistance property. The challenge was to modify this scheme in order to achieve collusion resistance without losing the other security and efficiency properties, in particular that of constant size ciphertexts. The resulting scheme works for threshold policies: the sender chooses ad-hoc a set S of attributes and a threshold t, and only users who hold at least t of the attributes in S can decrypt. An extension is possible in order to support also weighted threshold policies.

Our new scheme achieves security against selective chosen plaintext attacks (sCPA), in the standard model, under the assumption that the augmented multi-sequence of exponents decisional Diffie-Hellman (aMSE-DDH) problem is hard to solve. This is essentially the same level of security that was proved for the scheme in [DP08]. Using well-known techniques, it is possible to obtain security against chosen ciphertext attacks (CCA), in the random oracle model.

Organization of the paper. We define the syntactics of attribute-based encryption and the required security properties in Section 2, where we also describe the aMSE-DDH problem, on which the security of our scheme will be based. Section 3 contains the description of our scheme, the details on its correctness and consistency checking, and finally the formal proof of its security. In Section 4 we discuss how to extend our threshold scheme to the case of weighted threshold decryption policies, and the (im)possibility to achieve CCA security from CPA security in the standard model using a generic conversion due to [Wat08]. The work is concluded in Section 5.

2 Preliminaries

In this section we describe the algorithms that form an attribute-based encryption scheme which supports threshold decryption policies, as well as the basic security requirements for such schemes. We also introduce the computational problem called aMSE-DDH problem, to which we will relate the security of our scheme.

2.1 Attribute-Based Encryption

In a ciphertext-policy attribute-based encryption (ABE, for short) system, each user receives from a master entity a secret key which depends on the attributes that he satisfies (to soften the natural limitation of the *unique* trusted authority, the possibility to distribute the key extraction among several authorities has been investigated in [Cha07]). A sender can encrypt a message so that it can be decrypted only by users whose attributes satisfy some policy of his choice,

and which may depend of the message. Since the basic scheme that we propose in Section 3 works for *threshold* decryption policies, we describe the protocols and security model with respect to these threshold policies: the sender chooses a subset S of attributes and a threshold t such that $1 \leq t \leq |S|$, and encrypts a message m for the pair (S, t). A particular user will be able to decrypt the ciphertext only if he holds t or more attributes in S. The protocols and security model for ABE schemes supporting more general decryption policies can be described in a very similar way.

Syntactic Definition. A *ciphertext-policy attribute-based encryption scheme* $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ supporting threshold decryption policies consists of four probabilistic polynomial-time algorithms:

- The randomized *setup* algorithm Setup takes a security parameter λ and a universe of attributes $\mathcal{P} = \{\mathsf{at}_1, \ldots, \mathsf{at}_m\}$ as inputs and outputs some public parameters params, containing in particular the set $\mathcal{P}$, which will be common to all the users of the system, along with a secret key msk for the master entity. The public parameters will be an input of all the following algorithms. We write $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$ to denote an execution of this algorithm.
- The *key extraction* algorithm Ext is an interaction between a user and the master entity. The user proves to the master entity that he enjoys a subset $A \subset \mathcal{P}$ of attributes. After verifying that this is actually the case, the master entity uses his master secret key msk to generate a secret key sk_A (which depends on the subset A of attributes), and gives it to the user. We refer to an execution of this protocol as $\mathsf{sk}_A \leftarrow \mathsf{ABE.Ext}(\mathsf{params}, A, \mathsf{msk})$.
- The *encryption* algorithm Enc takes a subset of attributes $S \subset \mathcal{P}$, a threshold t such that $1 \leq t \leq |S|$, and a message M as inputs. The output is a ciphertext C. We denote an execution of the encryption algorithm as $C \leftarrow \mathsf{ABE.Enc}(\mathsf{params}, S, t, M)$.
- The *decryption* algorithm Dec takes a ciphertext C for the pair (S, t) and a secret key sk_A corresponding to some subset A of attributes as inputs. The output is a message $\tilde{M}$. We write $\tilde{M} \leftarrow \mathsf{ABE.Dec}(\mathsf{params}, C, (S, t), \mathsf{sk}_A)$ to refer to an execution of this protocol.

For correctness, it is required that

$$\mathsf{ABE.Dec}(\mathsf{params}, \mathsf{ABE.Enc}(\mathsf{params}, S, t, M), (S, t), \mathsf{sk}_A) \;=\; M,$$

whenever $|A \cap S| \geq t$ and the values $\mathsf{params}, \mathsf{msk}, \mathsf{sk}_A$ have been obtained by properly executing the protocols $\mathsf{ABE.Setup}$ and $\mathsf{ABE.Ext}$.

Security Model for ABE Schemes. Most previous schemes (all but the one in [LO+10]) consider only security under selective chosen plaintext attacks. This is also the security level that will be provably achieved by our scheme. *Indistinguishability under selective chosen plaintext attacks* ($\mathsf{IND\text{-}sCPA}$ security,

for short) for an attribute-based encryption scheme ABE supporting threshold decryption policies and for a security parameter $\lambda \in \mathbb{N}$ is defined by considering the following game that an attacker $\mathcal{A}$ plays against a challenger:

1. The challenger specifies a universe of attributes $\mathcal{P}$ of size m and gives it to the attacker $\mathcal{A}$.
2. $\mathcal{A}$ selects a subset $S \subset \mathcal{P}$ of s attributes and a threshold t such that $1 \leq t \leq s$.
3. The challenger runs $(\mathsf{params}, \mathsf{msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$ and gives params to $\mathcal{A}$.
4. [Secret key queries:] $\mathcal{A}$ adaptively sends subsets of attributes $B \subset \mathcal{P}$, with the restriction $|B \cap S| < t$, and must receive $\mathsf{sk}_B \leftarrow \mathsf{ABE.Ext}(\mathsf{params}, B, \mathsf{msk})$ as the answer.
5. $\mathcal{A}$ outputs two messages M_0, M_1 of the same length.
6. [Challenge:] The challenger picks a random bit $b^\star \in \{0,1\}$, computes $C^\star \leftarrow \mathsf{ABE.Enc}(\mathsf{params}, S, t, M_{b^\star})$ and gives $C^\star$ to $\mathcal{A}$.
7. Step 4 is repeated.
8. $\mathcal{A}$ outputs a bit b.

The *advantage* of such an adversary $\mathcal{A}$ in breaking the $\mathsf{IND\text{-}sCPA}$ security of the ABE scheme is defined as

$$\mathsf{Adv}^{\mathsf{IND\text{-}sCPA}}_{\mathcal{A},\mathsf{ABE}}(\lambda) = |2\Pr[b = b^\star] - 1|\,.$$

An attribute-based encryption scheme ABE is said to be $\mathsf{IND\text{-}sCPA}$ secure if $\mathsf{Adv}^{\mathsf{IND\text{-}sCPA}}_{\mathcal{A},\mathsf{ABE}}(\lambda)$ is negligible with respect to the security parameter λ, for any polynomial time adversary $\mathcal{A}$.

Note also that collusion resistance follows from the fact that the adversary can make multiple adaptive secret key queries both before and after the challenge phase.

This is not the strongest security notion that one can consider for ABE schemes. On the one hand, the attacker $\mathcal{A}$ can be allowed to make decryption queries, for ciphertexts C' of his choice (corresponding to pairs (S', t')), with the restriction that the challenge ciphertext C^* is never queried for the challenge pair (S, t). On the other hand, $\mathcal{A}$ can be allowed to choose the challenge pair (S, t) not at the beginning of the game, but at the same time when he chooses the two messages M_0, M_1. In this case, we say that $\mathcal{A}$ is a chosen ciphertext attacker, and that his goal is to break the CCA security of the ABE scheme.

2.2 The Augmented Multi-Sequence of Exponents Diffie-Hellman Problem

Our scheme uses an admissible bilinear map (or pairing) as an ingredient and its security relies on the hardness of a problem that we call the *augmented multi-sequence of exponents decisional Diffie-Hellman problem*, which is a slight modification of the multi-sequence of exponents decisional Diffie-Hellman problem considered in [DP08]. The generic complexity of these two problems is covered

by the analysis in [BBG05], because the problems fit their *general Diffie-Hellman exponent problem* framework.

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be three groups of the same prime order p (this is called a *bilinear group triple* in the sequel), and let $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ be a non-degenerate and efficiently computable bilinear map. Let g_0 be a generator of $\mathbb{G}_1$ and let h_0 be a generator of $\mathbb{G}_2$. In practice, the bilinear map e can be implemented on any pairing-friendly (hyper-)elliptic curve [FST10]; no more assumptions are made on the groups $\mathbb{G}_1$ and $\mathbb{G}_2$, or on the hypothetical existence of an efficient isomorphism from the one to the other.

Let $\tilde{\ell}, \tilde{m}, \tilde{t}$ be three integers. The $(\tilde{\ell}, \tilde{m}, \tilde{t})$*-augmented multi-sequence of exponents decisional Diffie-Hellman problem* ($(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-DDH) related to the group triplet $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ is as follows:

Input: the vector $\overrightarrow{x}_{\tilde{\ell}+\tilde{m}} = (x_1, \ldots, x_{\tilde{\ell}+\tilde{m}})$ whose components are pairwise distinct elements of $(\mathbb{Z}/p\mathbb{Z})^\star$ which define the polynomials

$$f(X) = \prod_{i=1}^{\tilde{\ell}} (X + x_i) \text{ and } g(X) = \prod_{i=\tilde{\ell}+1}^{\tilde{\ell}+\tilde{m}} (X + x_i),$$

the values

$$\begin{cases} g_0, g_0^{\gamma}, \ldots, g_0^{\gamma^{\tilde{\ell}+\tilde{t}-2}}, & g_0^{\kappa\cdot\gamma\cdot f(\gamma)}, & (1.1) \\ g_0^{\omega\gamma}, \ldots, g_0^{\omega\gamma^{\tilde{\ell}+\tilde{t}-2}}, & & (1.2) \\ g_0^{\alpha}, g_0^{\alpha\gamma}, \ldots, g_0^{\alpha\gamma^{\tilde{\ell}+\tilde{t}}}, & & (1.3) \\ h_0, h_0^{\gamma}, \ldots, h_0^{\gamma^{\tilde{m}-2}}, & h_0^{\kappa\cdot g(\gamma)} & (1.4) \\ h_0^{\omega}, h_0^{\omega\gamma}, \ldots, h_0^{\omega\gamma^{\tilde{m}-1}}, & & (1.5) \\ h_0^{\alpha}, h_0^{\alpha\gamma}, \ldots, h_0^{\alpha\gamma^{2(\tilde{m}-\tilde{t})+3}} & & (1.6) \end{cases}$$

where $\kappa, \alpha, \gamma, \omega$ are unknown random elements of $(\mathbb{Z}/p\mathbb{Z})^\star$, and finally an element $T \in \mathbb{G}_T$.

Output: a bit b.

The problem is correctly solved if the output is $b = 1$ when $T = e(g_0, h_0)^{\kappa\cdot f(\gamma)}$ or if the output is $b = 0$ when T is a random value from $\mathbb{G}_T$. In other words, the goal is to distinguish if T is a random value or if it is equal to $e(g_0, h_0)^{\kappa\cdot f(\gamma)}$.

More formally, let us denote by real the event that T is indeed equal to $T = e(g_0, h_0)^{\kappa\cdot f(\gamma)}$, by random the event that T is a random element from $\mathbb{G}_T$ and by $\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \omega, T)$ the input of the problem. Then, we define the *advantage* of an algorithm $\mathcal{B}$ in solving the $(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-DDH problem as

$$\begin{aligned} \mathsf{Adv}_{\mathcal{B}}^{(\tilde{\ell},\tilde{m},\tilde{t})-\text{aMSE-DDH}}(\lambda) = \Big| & \Pr\left[\mathcal{B}(\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \omega, T)) = 1 \middle| \mathsf{real}\right] \\ & - \Pr\left[\mathcal{B}(\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \omega, T)) = 1 \middle| \mathsf{random}\right] \Big| \end{aligned}$$

where the probability is taken over all random choices and over the random coins of $\mathcal{B}$.

The only difference with the multi-sequence of exponents decisional Diffie-Hellman problem from [DP08] is the presence in the input of two additional lines (l.2) and (l.5). The generic hardness of this problem is a consequence of Theorem A.2 from [BBG05]. It is stated in the next proposition whose proof follows (almost exactly) that of Corollary 3 in [DP08].

Proposition 1. *For any probabilistic algorithm $\mathcal{B}$ making at most q_G queries to the the oracle that computes the group operations (in groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order p) and the bilinear pairing $e(\cdot,\cdot)$, its advantage in solving the aMSE-DDH problem satisfies*

$$\mathsf{Adv}_{\mathcal{B}}^{(\tilde{\ell},\tilde{m},\tilde{t})-\mathsf{aMSE\text{-}DDH}}(\lambda) \leq \frac{(q_G + 2s + 2)^2 \cdot d}{2p}$$

where $s = 4\tilde{m} + 3\tilde{\ell} + \tilde{t} + 3$ and $d = \max\{2(\tilde{\ell}+2), 2(\tilde{m}+2), 4(\tilde{m}-\tilde{t})+10\}$.

3 The New ABE Scheme

This section is dedicated to the presentation of our ciphertext-policy attribute-based encryption scheme.

In the decryption process, we will use the algorithm Aggregate of [DP08]. Given a list of values $\{g^{\frac{r}{\gamma+x_i}}, x_i\}_{1\leq i\leq n}$, where $r, \gamma \in (\mathbb{Z}/p\mathbb{Z})^\star$ are unknown and $x_i \neq x_j$ if $i \neq j$, the algorithm computes the value

$$\mathsf{Aggregate}(\{g^{\frac{r}{\gamma+x_i}}, x_i\}_{1\leq i\leq n}) = g^{\frac{r}{\prod_{i=1}^{n}(\gamma+x_i)}}.$$

using $O(n^2)$ exponentiations.

Although the algorithm Aggregate of [DP08] is given for elements in $\mathbb{G}_T$, it is immediate to see that it works in any group of prime order. Running Aggregate for elements in $\mathbb{G}_1$ results in our case in a more efficient decryption algorithm.

3.1 Description of the Scheme

Setup, $\mathsf{ABE.Setup}(1^\lambda, \mathcal{P})$.
The master entity chooses a suitable encoding τ sending each of the m attributes $\mathsf{at} \in \mathcal{P}$ onto a (different) element $\tau(\mathsf{at}) = x \in (\mathbb{Z}/p\mathbb{Z})^\star$. He also chooses a bilinear group triple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of prime order p (such that p is λ bits long) and a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. He selects a generator g of $\mathbb{G}_1$ and a generator h of $\mathbb{G}_2$.

After that, he chooses a set $\mathcal{D} = \{d_1, \ldots, d_{m-1}\}$ consisting of $m-1$ pairwise different elements of $(\mathbb{Z}/p\mathbb{Z})^\star$, which must also be different to the values $x = \tau(\mathsf{at})$, for all $\mathsf{at} \in \mathcal{P}$. For any integer i lower or equal to $m-1$, we denote as $\mathcal{D}_i$ the set $\{d_1, \ldots, d_i\}$. Next, the master entity picks at random $\alpha, \gamma \in (\mathbb{Z}/p\mathbb{Z})^\star$ and sets $u = g^{\alpha\gamma}$ and $v = e(g^\alpha, h)$. The master secret key is then $\mathsf{msk} = (g, \alpha, \gamma)$ and the public parameters are

$$\mathsf{params} = \left\{\mathcal{P}, m, u, v, \left\{h^{\alpha\gamma^i}\right\}_{i=0,\ldots,2m-1}, \mathcal{D}, \tau\right\}.$$

Key Extraction, $\mathsf{ABE.Ext}(\mathsf{params}, A, \mathsf{msk})$.
Given any subset $A \subset \mathcal{P}$ of attributes, the master entity picks $r \in (\mathbb{Z}/p\mathbb{Z})^\star$ at random and computes $\mathsf{sk}_A = \left\{ \left\{ g^{\frac{r}{\gamma+\tau(\mathsf{at})}} \right\}_{\mathsf{at}\in A}, \left\{ h^{r\gamma^i} \right\}_{i=0,\ldots,m-2}, h^{\frac{r-1}{\gamma}} \right\}$.

Encryption, $\mathsf{ABE.Enc}(\mathsf{params}, S, t, M)$.
Given a subset $S \subset \mathcal{P}$ with $s = |S|$ attributes, a threshold t satisfying $1 \leq t \leq s$, and a message $M \in \mathbb{G}_T$, the sender picks at random $\kappa \in (\mathbb{Z}/p\mathbb{Z})^\star$ and computes

$$\begin{cases} C_1 = u^{-\kappa}, \\ C_2 = h^{\kappa\cdot\alpha\cdot \prod\limits_{\mathsf{at}\in S}(\gamma+\tau(\mathsf{at}))\prod\limits_{d\in\mathcal{D}_{m+t-1-s}}(\gamma+d)}, \\ K = v^{\kappa}. \end{cases}$$

The value C_2 is computed from the set $\{h^{\alpha\gamma^i}\}_{i=0,\ldots,2m-1}$ that can be found in the public parameters. The ciphertext is then (C_1, C_2, C_3), where $C_3 = K \cdot M$.

Decryption, $\mathsf{ABE.Dec}(\mathsf{params}, (C_1, C_2, C_3), (S, t), \mathsf{sk}_A)$.
Any user with a set of attributes A such that $|A \cap S| \geq t$ can use the secret key sk_A to decrypt the ciphertext, as follows. Let A_S be any subset of $A \cap S$ with $|A_S| = t$. The user computes, from all $\mathsf{at} \in A_S$, the value

$$\mathsf{Aggregate}(\{g^{\frac{r}{\gamma+\tau(\mathsf{at})}}, \tau(\mathsf{at})\}_{\mathsf{at}\in A_S}) = g^{\frac{r}{\prod_{\mathsf{at}\in A_S}(\gamma+\tau(\mathsf{at}))}}.$$

With the output of the algorithm $\mathsf{Aggregate}$ the user computes

$$L = e(g^{\frac{r}{\prod_{\mathsf{at}\in A_S}(\gamma+\tau(\mathsf{at}))}}, C_2) = e(g,h)^{r\cdot\kappa\cdot\alpha\cdot \prod\limits_{\mathsf{at}\in S\setminus A_S}(\gamma+\tau(\mathsf{at}))\prod\limits_{d\in\mathcal{D}_{m+t-1-s}}(\gamma+d)}.$$

For simplicity we define $\tau(d) = d$ for all $d \in \mathcal{D}$ and given a set $A_S \subset S$, $P_{(A_S,S)}(\gamma)$ is

$$P_{(A_S,S)}(\gamma) = \frac{1}{\gamma}\left(\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{m+t-1-s})\setminus A_S} (\gamma+\tau(\mathsf{at})) - \prod_{\mathsf{at}\in(S\cup\mathcal{D}_{m+t-1-s})\setminus A_S} \tau(\mathsf{at}) \right).$$

The crucial point is that, since $|A_S| \geq t$, the degree of the polynomial $P_{(A_S,S)}(X)$ is lower or equal to $m-2$. Therefore, from the values included in sk_A, the user can compute $h^{rP_{(A_S,S)}(\gamma)}$.

After that, the user calculates

$$e(C_1, h^{rP_{(A_S,S)}(\gamma)}) \cdot L = e(g,h)^{\kappa\cdot r\cdot\alpha\cdot\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{m+t-1-s})\setminus A_S}\tau(\mathsf{at})} \tag{1}$$

and

$$e(C_1, h^{\frac{r-1}{\gamma}}) = e(g,h)^{-\kappa\cdot\alpha\cdot r} \cdot e(g,h)^{\kappa\cdot\alpha} \tag{2}$$

From Equation (1) the user can obtain

$$e(g,h)^{\kappa\cdot r\cdot\alpha} = \left(e(C_1, h^{rP_{(A_S,S)}(\gamma)}) \cdot L \right)^{1/\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{m+t-1-s})\setminus A_S}\tau(\mathsf{at})}$$

and multiply this value in Equation (2). The result of this multiplication leads to $K = e(g,h)^{\kappa\cdot\alpha}$. Finally, the user recovers the message by computing $M = C_3/K$.

3.2 Consistency Checking and Efficiency Considerations

It is not hard to prove that the new ABE scheme satisfy the correctness property: if all the protocols are correctly executed, and if $|A \cap S| \geq t$, then sk_A allows to recover plaintexts that have been encrypted for the pair (S, t).

It is worth noting that, by adding g^α to the public parameters (this modification does not affect the security proof that we present in the next section), the users can check the consistency of the secret key they receive from the master entity. To do so, they must verify that, for all their attributes $\mathsf{at} \in A$,

$$e\left(g^{\frac{r}{\gamma+\tau(\mathsf{at})}}, h^{\alpha\gamma} \cdot (h^\alpha)^{\tau(\mathsf{at})}\right) = e\left(g^\alpha, h^r\right)$$

and then that, for $i = 1, \ldots, m-2$,

$$e\left(g^\alpha, h^{r\gamma^i}\right) = e\left(u, h^{r\gamma^{i-1}}\right)$$

Finally, they have to check that $e(u, h^{\frac{r-1}{\gamma}}) = e\left(g^\alpha, h^r\right)/v$.

In terms of efficiency, the main contribution of this new scheme is the constant size of the ciphertext, which consists of one element of each group $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$. The encryption requires no pairing computations, but $m + t + 1$ exponentiations. The decryption process requires 3 pairing evaluations and $O(t^2 + m)$ exponentiations. The size of the secret key is linear in the number of attributes, as in all existing ABE schemes.

3.3 Security Analysis

We are going to prove that our scheme is IND-sCPA secure, assuming that the aMSE-DDH problem is hard to solve.

Theorem 1. *Let λ be an integer. For any adversary $\mathcal{A}$ against the IND-sCPA security of our attribute-based encryption scheme, for a universe of m attributes $\mathcal{P}$, and a challenge pair (S, t) with $s = |S|$, there exists a solver $\mathcal{B}$ of the $(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-DDH problem, for $\tilde{\ell} = m - s$, $\tilde{m} = m + t - 1$ and $\tilde{t} = t + 1$, such that*

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{aMSE\text{-}DDH}}(\lambda) \;\geq\; \frac{1}{2} \cdot \mathsf{Adv}_{\mathcal{A}}^{\mathsf{IND\text{-}sCPA}}(\lambda).$$

Proof. We are going to construct an algorithm $\mathcal{B}$ that uses the adversary $\mathcal{A}$ as a black-box and that solves the $(m-s, m+t-1, t+1)$-augmented multi-sequence of exponents decisional Diffie-Hellman problem. The main trick in the proof will be to use the input of the aMSE-DDH problem to compute evaluations of some polynomials in γ "in the exponent".

Let $\mathcal{I}(\overrightarrow{x}_{2m+t-1-s}, \kappa, \alpha, \gamma, \omega, T)$ be the input of the algorithm $\mathcal{B}$. First, $\mathcal{B}$ specifies a universe of attributes, $\mathcal{P} = \{\mathsf{at}_1, \ldots, \mathsf{at}_m\}$. Next, the adversary $\mathcal{A}$ chooses a set $S \subset \mathcal{P}$ of cardinal s that he wants to attack, and a threshold t such that $1 \leq t \leq s$. Without loss of generality, we assume $S = \{\mathsf{at}_{m-s+1}, \ldots, \mathsf{at}_m\} \subset$

$\mathcal{P}$. From now on, we will denote by A_S the subset $A \cap S$, for any subset of attributes A.

Simulation of the setup. The algorithm $\mathcal{B}$ defines the encoding of the attributes as $\tau(\mathsf{at}_i) = x_i$ for $i = 1, \ldots, m$. Observe that the encodings of the first $m - s$ elements are the opposite of the roots of $f(X)$, and the encodings of the attributes in S are the opposite of some roots of $g(X)$.

The values corresponding to the "dummy" attributes $\mathcal{D} = \{d_1, \ldots, d_{m-1}\}$ are defined as $d_j = x_{m+j}$ if $j = 1 \ldots m + t - 1 - s$. For $j = m + t - s, \ldots, m - 1$, the d_j's are picked uniformly at random in $(\mathbb{Z}/p\mathbb{Z})^\star$ until they are distinct from $\{x_1, \ldots, x_{2m+t-1-s}, d_{m+t-s}, \ldots, d_{j-1}\}$.

The algorithm $\mathcal{B}$ defines $g := g_0^{f(\gamma)}$. Note that $\mathcal{B}$ can compute g with the elements of line (l.1) of its input, since f is a polynomial of degree $\tilde{\ell}$. To complete the setup phase, $\mathcal{B}$ sets $h = h_0$ and computes

- $u = g^{\alpha\gamma} = g_0^{\alpha \cdot \gamma \cdot f(\gamma)}$ with line (l.3) of its input, which is possible since $Xf(X)$ is a polynomial of degree $\tilde{\ell} + 1$. Indeed, $\alpha \cdot \gamma \cdot f(\gamma)$ is a linear combination of $\{\alpha\gamma, \ldots, \alpha\gamma^{\tilde{\ell}+1}\}$ and the coefficients of this linear combination are known to $\mathcal{B}$, so the value u can be computed from line (l.3).
- $v = e(g, h)^\alpha = e(g_0^{f(\gamma)\alpha}, h_0)$ with line (l.3) for $g_0^{f(\gamma)\alpha}$. Note that the value g^α could be computed by $\mathcal{B}$ and added to the public parameters, in case the verification of the consistency of the secret keys is desired for the scheme.

The algorithm $\mathcal{B}$ can compute the values $\{h^{\alpha\gamma^i}\}_{i=0,\ldots,2m-1}$ from line (l.6) of its input. Eventually, $\mathcal{B}$ gives to $\mathcal{A}$ the resulting

$$\mathsf{params} = \{\mathcal{P}, m, u, v, \{h^{\alpha\gamma^i}\}_{i=0,\ldots,2m-1}, \mathcal{D}, \tau\}.$$

Simulation of key extraction queries. Whenever the adversary $\mathcal{A}$ makes a key extraction query for a subset of attributes $A = \{\mathsf{at}_{i_1}, \ldots, \mathsf{at}_{i_n}\} \subset \mathcal{P}$ satisfying that $0 \leq |A_S| \leq t - 1$, the algorithm $\mathcal{B}$ must produce a tuple of the form

$$\mathsf{sk}_A = \left\{ \left\{ g^{\frac{r}{\gamma + \tau(\mathsf{at})}} \right\}_{\mathsf{at} \in A}, \left\{ h^{r\gamma^i} \right\}_{i=0,\ldots,m-2}, h^{\frac{r-1}{\gamma}} \right\},$$

for some random value $r \in (\mathbb{Z}/p\mathbb{Z})^\star$. To do so, $\mathcal{B}$ implicitly defines $r = (\omega y_A \gamma + 1)Q_A(\gamma)$, where y_A is randomly picked in $(\mathbb{Z}/p\mathbb{Z})^\star$, and the polynomial $Q_A(X)$ is defined as $Q_A(\gamma) = 1$ when $|A_S| = 0$, or $Q_A(X) = \lambda_A \cdot \prod_{\mathsf{at} \in A_S} (X + \tau(\mathsf{at}))$ otherwise, in which case $\lambda_A = (\prod_{\mathsf{at} \in A_S} \tau(\mathsf{at}))^{-1}$.

The elements which form sk_A are then computed as follows:

- For any $\mathsf{at} \in A_S$, $\mathcal{B}$ defines

$$Q_{\mathsf{at}}(\gamma) = Q_A(\gamma)/(\gamma + \tau(\mathsf{at})) = \lambda_A \cdot \prod_{\tilde{\mathsf{at}} \in A_S,\ \tilde{\mathsf{at}} \neq \mathsf{at}} (\gamma + \tau(\tilde{\mathsf{at}})).$$

Then $g^{\frac{r}{\gamma+\tau(\mathsf{at})}} = g_0^{f(\gamma)\omega y_A \gamma Q_{\mathsf{at}}(\gamma)} \cdot g_0^{f(\gamma)Q_{\mathsf{at}}(\gamma)}$. The first factor of the product (whose exponent is a polynomial in γ of degree at most $(m-s)+1+t-2$) can be computed from line (l.2), whereas the second factor (whose exponent is a polynomial in γ of degree at most $(m-s)+t-2$) can be computed from line (l.1).

- For any $\mathsf{at} \in A \setminus A_S$, $\mathcal{B}$ defines the polynomial $f_{\mathsf{at}}(X) = f(X)/(X+\tau(\mathsf{at}))$. Then $g^{\frac{r}{\gamma+\tau(\mathsf{at})}} = g_0^{f_{\mathsf{at}}(\gamma)\omega y_A \gamma Q_A(\gamma)} \cdot g_0^{f_{\mathsf{at}}(\gamma)Q_A(\gamma)}$. Again, the first factor of this product can be computed from line (l.2), and the second factor can be computed from line (l.1).
- The values $\left\{h^{r\gamma^i}\right\}_{i=0,\ldots,m-2}$ can be computed from line (l.4) and (l.5), since $h^{r\gamma^i} = h^{Q_A(\gamma)\omega y_A \gamma^{i+1}} \cdot h^{Q_A(\gamma)\gamma^i}$.
- Finally, $\mathcal{B}$ has to compute $h^{\frac{r-1}{\gamma}} = h^{Q_A(\gamma)\omega y_A} \cdot h^{\frac{Q_A(\gamma)-1}{\gamma}}$. The first factor of the product can be computed from line (l.5) and the second factor can be computed from line (l.4), since by definition of λ_A, $Q_A(X)$ is a polynomial with independent term equal to 1 and thus $\frac{Q_A(\gamma)-1}{\gamma}$ is a linear combination of $\{1, \gamma, \ldots, \gamma^{t-2}\}$.

Note that $Q_A(\gamma) \neq 0$ (otherwise $\gamma = \tau(at)$ for some $\mathsf{at} \in A_S$ and γ is public), in which case it is not hard to see that r is uniformly distributed in $\mathbb{Z}/p\mathbb{Z}$. If the choice of y_A leads to $r = 0$ (which occurs only with negligible probability anyhow), it suffices to pick a different value for y_A. That is, in the simulation r is uniformly distributed in $(\mathbb{Z}/p\mathbb{Z})^\star$.

Simulation of the challenge. Once $\mathcal{A}$ sends to $\mathcal{B}$ the two messages M_0 and M_1, $\mathcal{B}$ flips a coin $b \in \{0,1\}$, and sets $C_3^\star = T \cdot M_b$. To simulate the rest of the challenge ciphertext, $\mathcal{B}$ implicitly defines the randomness for the encryption as $\kappa' = \kappa/\alpha$, and sets $C_2^\star = h_0^{\kappa \cdot g(\gamma)}$ (given in line (l.4) of the $\mathsf{aMSE\text{-}DDH}$ input). To complete the ciphertext, $\mathcal{B}$ computes $C_1^\star = \left(g_0^{\kappa \cdot \gamma f(\gamma)}\right)^{-1}$ from line (l.1) of the input, which is equal to $u^{-\kappa'}$.

After the challenge step $\mathcal{A}$ may make other key extraction queries, which are answered as before.

Guess. Finally, $\mathcal{A}$ outputs a bit b'. If $b' = b$, $\mathcal{B}$ answers 1 as the solution to the given instance of the $\mathsf{aMSE\text{-}DDH}$ problem, meaning that $T = e(g_0, h_0)^{\kappa \cdot f(\gamma)}$. Otherwise, $\mathcal{B}$ answers 0, meaning that T is a random element.

We now have to analyze the advantage of the algorithm $\mathcal{B}$:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{B}}^{\mathsf{aMSE\text{-}DDH}}(\lambda) &= \Big| \Pr\left[\mathcal{B}(\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \omega, T)) = 1 \middle| \mathsf{real}\right] - \\ &\qquad \Pr\left[\mathcal{B}(\mathcal{I}(\overrightarrow{x}_{\tilde{\ell}+\tilde{m}}, \kappa, \alpha, \gamma, \omega, T)) = 1 \middle| \mathsf{random}\right] \Big| \\ &= \Big| \Pr\left[b = b' \middle| \mathsf{real}\right] - \Pr\left[b = b' \middle| \mathsf{random}\right] \Big|.\end{aligned}$$

When the event real occurs, then $\mathcal{A}$ is playing a real attack and therefore $|\Pr\left[b = b' \middle| \mathsf{real}\right] - 1/2| = \frac{1}{2}\mathsf{Adv}_{\mathcal{A},\Pi}^{\mathsf{IND\text{-}sCPA}}(\lambda)$. During the random event, the view of

$\mathcal{A}$ is completely independent of the bit b; in this case, the probability $\Pr[b = b']$ is equal to $1/2$. Summing up, we obtain

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{aMSE\text{-}DDH}}(\lambda) \geq \frac{1}{2}\mathsf{Adv}_{\mathcal{A},\Pi}^{\mathsf{IND\text{-}sCPA}}(\lambda).$$

□

4 Extensions

In this section we discuss two possible extensions of the basic scheme that we have described and analyzed in the previous section. First, we study the possibility of supporting more general decryption policies, not only threshold ones. After that, we discuss the options to obtain security against chosen ciphertext attacks.

4.1 More General Decryption Policies

Although we have considered in this paper the special case of threshold decryption policies, attribute-based encryption schemes can be defined for general decryption policies. Such a policy is determined by a monotone increasing family $\Gamma \subset 2^{\mathcal{P}}$ of subsets of attributes, in $\mathcal{P} = \{\mathsf{at}_1, \ldots, \mathsf{at}_n\}$. This family (or *access structure*) is chosen by the sender at the time of encryption, in such a way that only users whose subset of attributes A belong to Γ can decrypt. Even if many users collude, each of them having a subset of attributes out of Γ, the encryption scheme must remain secure.

The threshold ABE scheme that we have described and analyzed in this paper is inspired on the dynamic threshold identity-based encryption scheme of [DP08]. It is claimed in [DP08] that the threshold scheme there can be extended to admit "all the classical cases" of more general access structures. However, this is not completely true, because their extension only applies to a sub family of access structures, *weighted threshold* ones. A family $\Gamma \subset 2^{\mathcal{P}}$ is a weighted threshold access structure if there exist a threshold t and an assignment of weights $\omega : \mathcal{P} \to \mathbb{Z}^+$ such that $A \in \Gamma \iff \sum_{\mathsf{at} \in A} \omega(\mathsf{at}) \geq t$. Of course, there are many access structures which are not weighted threshold, for example $\Gamma = \{\{\mathsf{at}_1, \mathsf{at}_2\}, \{\mathsf{at}_2, \mathsf{at}_3\}, \{\mathsf{at}_3, \mathsf{at}_4\}\}$ in the set $\mathcal{P} = \{\mathsf{at}_1, \mathsf{at}_2, \mathsf{at}_3, \mathsf{at}_4\}$.

The same extension proposed in [DP08] works for our threshold ABE scheme. Let K be an upper bound for $\omega(\mathsf{at})$, for all $\mathsf{at} \in \mathcal{P}$ and for all possible assignments of weights that realize weighted threshold decryption policies. During the setup of the ABE scheme, the new universe of attributes will be $\mathcal{P}' = \{\mathsf{at}_1||1, \mathsf{at}_1||2, \ldots, \mathsf{at}_1||K, \ldots, \mathsf{at}_n||1, \ldots, \mathsf{at}_n||K\}$. During the secret key request phase, if an attribute at belongs to the requested subset $A \subset \mathcal{P}$, the secret key sk_A will contain the elements $g^{\frac{r}{\gamma+\tau(\mathsf{at}^{(j)})}}$ corresponding to $\mathsf{at}^{(j)} = \mathsf{at}||j$, for all $j = 1, \ldots, K$.

Later, suppose a sender wants to encrypt a message for a weighted threshold decryption policy Γ, defined on a subset of attributes $S = \{\mathsf{at}_1, \ldots, \mathsf{at}_s\}$ (without

loss of generality). Let t and $\omega : S \to \mathbb{Z}^+$ be the threshold and assignment of weights that realize Γ. The sender can use the threshold ABE encryption routine described in Section 3.1, with threshold t, but applied to the set of attributes $S' = \{\mathsf{at}_1||1, \ldots, \mathsf{at}_1||\omega(\mathsf{at}_1), \ldots, \mathsf{at}_s||1, \ldots, \mathsf{at}_s||\omega(\mathsf{at}_s)\}$. In this way, if a user holds a subset of attributes $A \in \Gamma$, he will have $\omega(\mathsf{at})$ valid elements in his secret key, for each attribute $\mathsf{at} \in A$. In total, he will have $\sum_{\mathsf{at} \in A} \omega(\mathsf{at}) \geq t$ valid elements, so he will be able to run the decryption routine of the threshold ABE scheme and decrypt the ciphertext.

The security analysis can be extended to this more general case, as well. Therefore, we can conclude that our ABE scheme with constant size ciphertexts also admits weighted threshold decryption policies.

4.2 Security under Chosen Ciphertext Attacks

Some ABE schemes proposed in the literature [BSW07,CN07,Wat08] achieve security under selective chosen ciphertext attacks (sCCA security). This is done in two steps. Firstly sCPA security is proved, and secondly the scheme is shown to admit delegation of secret keys: it is possible to compute a valid secret key $\mathsf{sk}_{A'}$ from a valid secret key sk_A, for any $A' \subset A$. If this is the case, the basic ABE scheme can be viewed as a hierarchical ABE scheme, where the hierarchy is the classical one: a user holding attributes A is over a user holding attributes A', if $A' \subset A$. Finally, the techniques developed in [CHK04] can be applied to this sCPA secure hierarchical ABE scheme, which results in a sCCA secure ABE scheme, in the standard model.

Unfortunately our scheme does not seem to admit delegation of secret keys. Therefore, it is still an open problem to come up with an ABE scheme with constant size ciphertexts, achieving sCCA security in the standard model. In contrast, if one requires security in the random oracle model only, such a result is easily obtained by applying to our scheme (a variant of) some classical CPA to CCA transformation, such as the Fujisaki-Okamoto one [FuOk99].

5 Conclusion

We have proposed in this paper the first (reasonably expressive) attribute-based encryption scheme with constant size ciphertexts. The design of the scheme is inspired by the dynamic threshold encryption scheme in [DP08]. Our ABE scheme works for threshold policies: the sender chooses, at the time of encryption, the involved set of attributes and a threshold, in such a way that only those users holding (at least) this threshold of the involved attributes can decrypt. However, the scheme can be easily extended to admit weighted threshold decryption policies, as well.

Although finding attribute-based encryption schemes with short ciphertexts supporting even more expressive decryption policies is an important open problem, weighted threshold decryption policies are quite expressive and can cover a

wide range of applications. Therefore, we think that our proposal achieves a fair trade-off between expressiveness and efficiency.

Our scheme employs bilinear pairings, and its security is based on the assumption that a newly introduced problem, the augmented Multi-Sequence of Exponents Decisional Diffie-Hellman (aMSE-DDH) problem, is hard. It remains an open problem to obtain a scheme with constant ciphertext's length whose security is based on a more standard algorithmic problem and which achieves full security (i.e. not only selective security).

Acknowledgments

This work was partially done while Javier Herranz and Carla Ràfols were visiting Université de Caen Basse-Normandie.
The work of Javier Herranz is supported by a *Ramón y Cajal* grant, partially funded by the European Social Fund (ESF) of the Spanish MICINN Ministry. Carla Ràfols holds an *FPI* grant of the Spanish MICINN Ministry. The work of both these authors is partially supported by the Spanish MICINN Ministry under project MTM2009-07694. The work of Fabien Laguillaumie is supported by the French ANR-07-TCOM-013-04 PACE Project.

References

[AMS06] S. Al-Riyami, J. Malone-Lee and N.P. Smart. Escrow-free encryption supporting cryptographic workflow. *International Journal of Information Security*, vol. **5** (4), pp. 217–229 (2006)

[BM05] W. Bagga and R. Molva. Policy-based cryptography and applications. *Proceedings of Financial Cryptography'05*, LNCS **3570**, Springer-Verlag, pp. 72–87 (2005)

[BSW07] J. Bethencourt, A. Sahai and B. Waters. Ciphertext-policy attribute-based encryption. *Proceedings of IEEE Symposium on Security and Privacy*, IEEE Society Press, pp. 321–334 (2007)

[BBG05] D. Boneh, X. Boyen and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. *Proceedings of Eurocrypt'05*, LNCS **3494**, Springer-Verlag, pp. 440–456 (2005)

[CG99] R. Canetti and S. Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. *Proceedings of Eurocrypt'99*, LNCS **1592**, Springer-Verlag, pp. 90–106 (1999)

[CHK04] R. Canetti, S. Halevi and J. Katz. Chosen-ciphertext security from identity-based encryption. *Proceedings of Eurocrypt'04*, LNCS **3027**, Springer-Verlag, pp. 207–222 (2004)

[CCZ06] Z. Chai, Z. Cao and Y. Zhou. Efficient ID-based broadcast threshold decryption in ad hoc network. *Proceedings of IMSCCS'06*, Volume 2, IEEE Computer Society, pp. 148–154 (2006)

[CN07] L. Cheung and C. C. Newport. Provably secure ciphertext policy ABE. *Proceedings of Computer and Communications Security, CCS'07*, ACM, pp. 456–465 (2007)

[Cha07] M. Chase. Multi-authority attribute based encryption. *Proceedings of TCC'07*, LNCS **4392**, Springer-Verlag, pp. 515–534 (2007)

[DHMR07] V. Daza, J. Herranz, P. Morillo and C. Ràfols. CCA2-secure threshold broadcast encryption with shorter ciphertexts. *Proceedings of ProvSec'07*, LNCS **4784**, Springer-Verlag, pp. 35–50 (2007)

[DHMR08] V. Daza, J. Herranz, P. Morillo and C. Ràfols. Extended access structures and their cryptographic applications. To appear in *Applicable Algebra in Engineering, Communication and Computing*. Available at `http://eprint.iacr.org/2008/502` (2008)

[DP08] C. Delerablée and D. Pointcheval. Dynamic threshold public-key encryption. *Proceedings of Crypto'08*, LNCS **5157**, Springer-Verlag, pp. 317–334 (2008)

[EM+09] K. Emura, A. Miyaji, A. Nomura, K. Omote and M. Sosh. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. *Proceedings of ISPEC'09*, LNCS **5451**, Springer-Verlag, pp. 13–23 (2009)

[FST10] D. Freeman, M. Scott and E. Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, vol. **23** (2), pp. 224–280 (2010)

[FuOk99] E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. *Proceedings of PKC'99*, LNCS **1560**, Springer-Verlag, pp. 53–68 (1999)

[GJPS08] V. Goyal, A. Jain, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute-based encryption. *Proceedings of ICALP'08*, LNCS **5126**, Springer-Verlag, pp. 579–591 (2008)

[GPSW06] V. Goyal, O. Pandey, A. Sahai and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. *Proceedings of Computer and Communications Security, CCS'06*, ACM, pp. 89–98 (2006)

[KSW08] J. Katz, A. Sahai and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. *Proceedings of Eurocrypt'08*, LNCS **4965**, Springer-Verlag, pp. 146–162 (2008)

[LO+10] A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. To appear in *Proceedings of Eurocrypt'10*. Available at `http://eprint.iacr.org/2010/110` (2010)

[SW05] A. Sahai and B. Waters. Fuzzy identity-based encryption. *Proceedings of Eurocrypt'05*, LNCS **3494**, Springer-Verlag, pp. 457–473 (2005)

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, vol. **22**, pp. 612–613 (1979)

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. *Proceedings of Crypto'84*, LNCS **196**, Springer-Verlag, pp. 47–53 (1984)

[Wat08] B. Waters. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. Manuscript available at `http://eprint.iacr.org/2008/290` (2008)