Extended Access Structures and Their Cryptographic Applications

Vanesa Daza[1], Javier Herranz [2], Paz Morillo[3] and Carla Ràfols[3]

[1] Dept. Tecnologies de la Informació i les Comunicacions,
Universitat Pompeu Fabra, Pg. Circumval·lació 8, Barcelona, Spain
vanesa.daza@upf.edu
[2] IIIA-CSIC,
Campus UAB, s/n, Bellaterra, Spain
jherranz@iiia.csic.es
[3] Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya,
C. Jordi Girona, 1-3, Mòdul C3, Barcelona, Spain
{paz,crafols}@ma4.upc.edu

November 28, 2008

Abstract

In secret sharing schemes a secret is distributed among a set of users $\mathcal{P}$ in such a way that only some sets, the authorized sets, can recover it. The family Γ of authorized sets is called access structure. Given such a monotone family $\Gamma \subset 2^{\mathcal{P}}$, we introduce the concept of *extended access structures*, defined over a larger set $\mathcal{P}' = \mathcal{P} \cup \tilde{\mathcal{P}}$, satisfying these two properties:

- the set $\mathcal{P}$ is a minimal subset of Γ', i.e. $\mathcal{P} - \{R_i\} \notin \Gamma'$ for every $R_i \in \mathcal{P}$,
- a subset $A \subset \mathcal{P}$ is in Γ if and only if the subset $A \cup \tilde{\mathcal{P}}$ is in Γ'.

As our first contribution, we give an explicit construction of an extended access structure Γ' starting from a vector space access structure Γ, and we prove that Γ' is also vector space. Our second contribution is to show that the concept of extended access structure can be used to design encryption schemes which involve access structures that are chosen ad-hoc at the time of encryption. Specifically, we design and analyze a dynamic distributed encryption scheme and a ciphertext-policy attribute-based encryption scheme. In some cases, the new schemes enjoy better properties than the existing ones.

Keywords: vector space access structure, secret sharing, dynamic distributed encryption, attribute-based encryption.

1 Introduction

Secret sharing schemes [25] allow a secret to be distributed among several parties. Roughly speaking, secret sharing schemes - SSS from now on- can force parties to

cooperate to perform a certain sensitive task, instead of trusting a single party. For example, a bank requiring that at least two of its employees cooperate to open the vault can *share* the vault's lock combination in such a way that any two employees can recover it, but one cannot. Similar challenges can arise in other sensitive areas where a secret should be recovered only if certain users, the *authorized sets*, get together. Given a set of users $\mathcal{P} = \{R_1, \ldots, R_n\}$, the family $\Gamma \subset 2^{\mathcal{P}}$ of authorized sets is the *access structure* of the secret sharing scheme.

A classical example of secret sharing schemes is Shamir's (t, n)-threshold scheme [25], in which every set of at least t users - out of a total of n - can recover the secret. Such a scheme can be easily implemented by means of polynomial interpolation. Indeed, it suffices to publicly associate to each user R_i a different element $\alpha_i \neq 0$ in a finite field $\mathcal{K}$ and to choose a random polynomial $Q \in \mathcal{K}[X]$ of degree $t-1$. If each user is given the share $Q(\alpha_i)$, the secret $Q(0)$ has been shared in such a way that at least t users have to get together to recover the secret. Such t is referred to as the *threshold* of the scheme.

Secret sharing schemes are widely used as primitives in many cryptographic protocols, for instance in distributed encryption schemes, where decryption can be done only by authorized groups of users. Typically, to allow decryption to be distributed according to a certain access structure Γ, the secret key required for decryption is shared among the users according to a certain SSS realizing Γ. The sender encrypts using the public key, then each user uses his share of the secret key to compute a partial decryption, and finally the partial decryptions of an authorized set can be combined to obtain the decrypted message. However, this solution may not be flexible enough to accommodate certain additional functionalities. For instance, in the case of *dynamic distributed encryption*, where the sender of the message chooses the receivers and the subsets authorized to decrypt at the time of encryption - so that different messages can be decrypted by different sets of users-, this solution will not do.

In this case a trivial alternative is to share the message m by means of a SSS realizing Γ and then to encrypt each share m_i using the public key of user R_i. This trivial solution has ciphertext length $n + O(1)$, if n is the total number of receivers. If we restrict ourselves to the threshold case, where the sender chooses the threshold t at the time of encryption, the ciphertext length can be reduced to $n - t + O(1)$, as shown in [16]. Indeed, given a set of n users with key pairs (pk_i, sk_i), the idea is to use all their public keys to generate a global public key PK - which implicitly defines a secret key SK- in such a way that sk_i can be seen as the share of user R_i of the secret SK. That is, SK is shared with an (n, n)-threshold secret sharing scheme. If the sender wants the threshold to be t, all it has to do is to include in the ciphertext $n - t$ partial decryptions, corresponding to $n - t$ *dummy*, not real, users. Then, any set of at least t of the real users can jointly decrypt.

Non-threshold access structures have received less attention in the literature of cryptographic protocols, partly because SSS realizing threshold access structures have a very simple description and intuitive realization, as we have seen above. However, more general access structures make a lot of sense in some scenarios like dynamic distributed encryption, or *attribute-based encryption*, where the sender

chooses a *decryption policy* for each ciphertext, such that a user can decrypt a message if he satisfies some requirements (attributes). This decryption policy can be seen as an access structure Γ over the set of all attributes. In this context, since different attributes may have different significance, it is not reasonable to restrict the sender to the threshold case and, in fact, most works dealing with the concept of attribute-based encryption consider general access structures (see [31], for example).

Not all the cryptographic threshold protocols can be easily extended to allow other access structures. The initial goal of this work was to adapt to the scenario of general access structures the (threshold) ideas of [16]: to reduce the ciphertext length by encrypting with some global public key and then adding some partial decryptions corresponding to some dummy players. To this purpose, we introduce the concept of extended access structures.

1.1 Our results

Given $\Gamma \subset 2^P$, an *extended access structure* Γ', defined over a larger set $\mathcal{P}' = \mathcal{P} \cup \tilde{\mathcal{P}}$, is an access structure satisfying these two properties:

- the set $\mathcal{P}$ is a minimal subset of Γ', i.e. $\mathcal{P} - \{R_i\} \notin \Gamma'$ for every $R_i \in \mathcal{P}$,
- a subset $\mathcal{A} \subset \mathcal{P}$ is in Γ if and only if the subset $A \cup \tilde{\mathcal{P}}$ is in Γ'.

These additional users $\tilde{\mathcal{P}}$ correspond to the aforementioned *dummy* users.

In Section 3 we prove, using linear algebra tools, that if Γ is a vector space access structure, then it is possible to explicitly construct an extended access structure Γ' which is also a vector space one. Γ is a vector space access structure if there exists some assignment of vectors (one for each user and one for a special external user D) such that Γ contains exactly those subsets of users whose vectors linearly generate the vector of D. It is easy to see that threshold access structures are a particular case of vector space ones.

We use this result to design in Section 4 the first dynamic distributed encryption scheme which works for the non-threshold case with ciphertexts containing less than n elements, where n is the number of receivers. We then show that the concept of extended access structures may be of independent interest, because it can be employed to design other distributed cryptographic protocols, not only dynamic distributed encryption schemes. As an example, we construct in Section 5 an attribute-based encryption scheme which works for any vector space access structure (for the subsets of attributes needed to decrypt) and which has in some cases shorter ciphertexts than the rest of attribute-based schemes in the literature. For completeness, the security analysis of these two cryptographic schemes is included in Appendices A and B. We give some concluding remarks in Section 6.

2 Preliminaries

In this section we recall some basics on the primitives of secret sharing, dynamic distributed encryption and attribute-based encryption, which will appear later in the rest of the paper.

2.1 Secret Sharing Schemes

The idea of *secret sharing schemes* was independently introduced by Shamir [25] and Blakley [5]. Let $\mathcal{P} = \{R_1, \ldots, R_n\}$ be a set of n players. In this set of players, a family of authorized or qualified subsets $\Gamma \subset 2^{\mathcal{P}}$ is defined. This family is called the *access structure* of the scheme, and it must be monotone increasing; that is, if $A_1 \in \Gamma$ and $A_1 \subset A_2 \subset \mathcal{P}$, then $A_2 \in \Gamma$. Because of this property, an access structure is fully determined by its basis $\Gamma_0 = \{A \in \Gamma \mid A - \{R_i\} \notin \Gamma, \text{ for all } R_i \in A\}$.

Given a monotone increasing access structure Γ and a secret to be shared, the idea behind a secret sharing scheme is that each player of the set $\mathcal{P}$ receives from a trusted and external authority (the *dealer*, usually denoted by D) a share of the secret. On the one hand, from the shares of any authorized subset, in Γ, the secret can be efficiently recovered. On the other hand, from the shares of a non-authorized subset, out of Γ, no information about the secret should be obtained.

Shamir proposed in [25] a *threshold* scheme, where subsets that can recover the secret are those with at least t members (t is the threshold); in other words, the access structure is $\Gamma = \{A \subset \mathcal{P} : |A| \geq t\}$. The scheme is based on polynomial interpolation.

A more general family of secret sharing schemes are *vector space* ones, introduced by Brickell in [12]. An access structure Γ is realizable by such a scheme, over a finite field $\mathcal{K}$, if there exist a positive integer d and a map $\psi : \mathcal{P} \cup \{D\} \longrightarrow (\mathcal{K})^d$, such that $A \in \Gamma$ if and only if $\psi(D) \in \langle \psi(R_i) \rangle_{R_i \in A}$. In this case, we say that Γ is a *vector space access structure*. If a dealer wants to distribute a secret value $s \in \mathcal{K}$ according to such an access structure, he takes a random vector $\omega \in (\mathcal{K})^d$, such that $\omega \cdot \psi(D) = s$. The share of a player $R_i \in \mathcal{P}$ is $s_i = \omega \cdot \psi(R_i) \in \mathcal{K}$. Let A be an authorized subset, $A \in \Gamma$; then, by definition, $\psi(D) = \sum_{R_i \in A} \lambda_i^A \psi(R_i)$, for some values $\lambda_i^A \in \mathcal{K}$. In order to recover the secret from their shares, players in A compute

$$\sum_{R_i \in A} \lambda_i^A s_i = \sum_{R_i \in A} \lambda_i^A \left(\omega \cdot \psi(R_i)\right) = \omega \cdot \sum_{R_i \in A} \lambda_i^A \psi(R_i) = \omega \cdot \psi(D) = s.$$

Shamir's threshold secret sharing scheme can be seen as a particular case of vector space ones, by defining $\psi(D) = (1, 0, \ldots, 0) \in (\mathbb{Z}_q)^t$ and $\psi(R_i) = (1, i, i^2, \ldots, i^{t-1}) \in (\mathcal{K})^t$ for every player $R_i \in \mathcal{P}$.

2.2 Dynamic Distributed Encryption

Roughly speaking, an encryption scheme with dynamic distributed decryption (DDE scheme, for short) works as follows. Each potential receiver generates his own pair of secret and public keys. The sender of a message chooses (ad-hoc) a set of receivers $\mathcal{P}$ and an access structure $\Gamma \subset 2^{\mathcal{P}}$ of authorized receivers, and then encrypts this message by using the public keys of these receivers. Given the resulting ciphertext, the original message can be recovered by any subset in Γ: they use their secret keys to compute partial decryptions which are then combined to obtain the message. This kind of schemes is strongly related to standard distributed encryption schemes [13, 8], but in these latter schemes the set of receivers and the access structures are

defined in the setup phase of the system, not chosen by the sender of each message. Furthermore, the receivers do not generate their key pairs independently: there is a distributed key generation phase, where a global public key is defined for the whole set of receivers, and their secret keys are shares of the corresponding global secret key. We emphasize that these differences are quite strong and call for specific solutions to construct DDE schemes.

More formally, a DDE scheme DDE= (DDE.Setup, DDE.KG, DDE.Enc, DDE.PartDec, DDE.Dec) consists of five algorithms:

- The randomized setup algorithm DDE.Setup takes as input a security parameter k and outputs some public parameters `params`, which will be common to all the users of the system. We write $\texttt{params} \leftarrow \text{DDE.Setup}(1^k)$.

- The randomized key generation algorithm DDE.KG is run by each user R_i. It takes as input some public parameters `params` and returns a pair (pk_i, sk_i) consisting of a public key and a matching secret key; we denote an execution of this protocol as $(pk_i, sk_i) \leftarrow \text{DDE.KG}(\texttt{params})$.

- The randomized encryption algorithm DDE.Enc takes as input a set of public keys $\{pk_i\}_{R_i \in \mathcal{P}}$ corresponding to a set $\mathcal{P}$ of n receivers, a monotone increasing family $\Gamma \subset 2^{\mathcal{P}}$ (the access structure), and a message m. The output is a ciphertext C, which contains the description of $\mathcal{P}$ and Γ; we write $C \leftarrow \text{DDE.Enc}(\mathcal{P}, \{pk_i\}_{R_i \in \mathcal{P}}, \Gamma, m)$.

- The (possibly randomized) partial decryption algorithm DDE.PartDec takes as input a ciphertext C for the pair $(\mathcal{P}, \Gamma)$ and a secret key sk_i of a receiver $R_i \in \mathcal{P}$. The output is a partial decryption value κ_i or a special symbol $\perp$. We denote with $\kappa_i \leftarrow \text{DDE.PartDec}(C, sk_i)$ an execution of this protocol.

- The deterministic final decryption algorithm DDE.Dec takes as input a ciphertext C for the pair $(\mathcal{P}, \Gamma)$ and partial decryptions $\{\kappa_i\}_{R_i \in A}$ corresponding to receivers in some authorized subset $A \in \Gamma$. The output is a message m. We write $m \leftarrow \text{DDE.Dec}(C, \{\kappa_i\}_{R_i \in A}, A)$.

When formalizing security of a DDE scheme, one considers an attacker that tries to break the security of the scheme. This attacker can corrupt different users, obtaining their secret keys. The final goal of the attacker is to obtain some information about a message which has been encrypted for a pair $(\mathcal{P}^*, \Gamma^*)$ such that the subset $\mathcal{U}'$ of corrupted players is not in Γ^*. Depending on whether the attacker has access to a decryption oracle, one can consider chosen plaintext attacks (CPA) or chosen ciphertext attacks (CCA). The resulting levels of security are known as *indistinguishability under CPA* (or IND-CPA security) and *indistinguishability under CCA* (or IND-CCA security). See Appendix A for a more formal definition of the IND-CPA security of this kind of schemes.

To the best of our knowledge, only a few works [19, 21, 15, 16] have dealt with DDE schemes, and all of them consider only the threshold case, where the authorized subsets of receivers are those in the threshold access structure $\Gamma = \{A \subset \mathcal{P} : |A| \geq t\}$,

for some value of the threshold t. The best of these results [16] achieves ciphertext length of size roughly $n - t$, where n is an upper bound for the size of the set of receivers. A slightly different variant of (threshold) DDE is considered in [17], where a master entity is in charge of generating the secret and public keys of every user; in this scenario, a scheme with constant-size ciphertexts but $\mathcal{O}(n)$ long public parameters is proposed.

The DDE scheme that we will describe in Section 4 is based on ElGamal's cryptosystem, which works as follows. The key generation protocol takes as input a security parameter k and generates two prime numbers p and q such that q is k bit long and $q|p-1$. Then a cyclic subgroup $\mathbb{G} = \langle g \rangle$ of $\mathbb{Z}_p$ is chosen, with order q. All these values are made public. The secret key sk of a user is chosen at random in $\mathbb{Z}_q^*$, whereas the matching public key is $pk = g^{sk} \bmod p$. To encrypt a message $m \in \mathbb{G}$ for the user with public key pk, a random value $a \in \mathbb{Z}_q^*$ is chosen, and the ciphertext $C = (r, s)$ is defined as $r = g^a \bmod p$ and $s = m \cdot pk^a \bmod p$. Finally, the owner of the secret key sk who receives a ciphertext $C = (r, s)$ can decrypt and obtain the original message, as $s/r^{sk} \bmod p = m$.

2.3 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 enjoys; examples of attributes can be at_1 ='student', at_2 ='professor', at_3 ='member of MIT', at_4 ='director of a department', etc. 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. For example, a ciphertext could be decrypted by users who are members of MIT 'and' are furthermore either professors 'or' directors of a department. Note that, if we define as $\mathcal{P} = \{at_1, \ldots, at_n\}$ the set of all possible attributes in such a system, a decryption policy for a determined ciphertext can always be defined as a monotone increasing family (or access structure) of subsets of $\mathcal{P}$. In the example above, with $n = 4$, the policy can be expressed by the access structure $\Gamma = \{\{at_2, at_3\}, \{at_3, at_4\}\}$.

An ABE scheme ABE= (ABE.Setup, ABE.Ext, ABE.Enc, ABE.Dec) consists of four probabilistic polynomial-time algorithms:

- The randomized setup algorithm ABE.Setup takes as input a security parameter k and outputs some public parameters `params` (containing the set $\mathcal{P}$ of possible attributes), which will be common to all the users of the system, along with a secret key msk for the master entity. We write $(\texttt{params}, msk) \leftarrow \text{ABE.Setup}(1^k)$.

- The key extraction algorithm ABE.Ext is an interaction between a user and the master entity. Let $\mathcal{P} = \{at_1, \ldots, at_n\}$ be the set of all possible attributes for users in the system. 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 denote an execution of this protocol as $sk_A \leftarrow \text{ABE.Ext}(\texttt{params}, A, msk)$.

- The encryption algorithm ABE.Enc takes as input a monotone increasing family $\Gamma \subset 2^{\mathcal{P}}$, i.e. the access structure that determines the policy for decryption, and a message m. The output is a ciphertext C, which must contain the description of Γ; we write $C \leftarrow \text{ABE.Enc}(\texttt{params}, \Gamma, m)$.

- The decryption algorithm ABE.Dec takes as input a ciphertext C for the policy Γ and a secret key sk_A corresponding to some subset A of attributes. The output is a message $\tilde{m}$. We write $\tilde{m} \leftarrow \text{ABE.Dec}(\texttt{params}, C, sk_A)$.

For correctness, it is required that $\text{ABE.Dec}(\texttt{params}, \text{ABE.Enc}(\texttt{params}, \Gamma, m), sk_A) = m$, whenever $A \in \Gamma$ and the values $\texttt{params}, msk, sk_A$ have been obtained by properly executing the protocols ABE.Setup and ABE.Ext.

For security, again quite informally, an ABE scheme must resist the action of an attacker that can query for secret keys of subsets $A_1, \ldots, A_\ell$ of attributes of his choice, and later tries to obtain some information about a message that is encrypted by using a policy Γ such that $A_i \notin \Gamma$, for all $i = 1, \ldots, \ell$. Note that if a scheme is secure in front of this kind of attacks, it resists collusions of users who try to decrypt a message encrypted under a policy Γ that they do not individually satisfy, even if the union of all the attributes of these users would give an authorized subset of attributes (for example, the whole set $\mathcal{P}$). See Appendix B for a more formal definition of the (selective) security of this kind of schemes.

The notion of attribute-based encryption appeared implicitly in [24]. In 2006 the first paper dealing explicitly with ciphertext-policy ABE [20] was published, while in [4] other models for ABE were defined. In this paper we only consider ciphertext-policy ABE. A recent work [31] gives some constructions of ciphertext-policy ABE schemes, using also tools from secret sharing, as we will do in our construction in Section 5.

In our ABE scheme, bilinear pairings will be an essential ingredient. Given an additive group $\mathbb{G}_1 = \langle P \rangle$ and a multiplicative group $\mathbb{G}_2$, both with prime order q, we say that they admit a *bilinear pairing* if there exists a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ satisfying the following properties:

1. it is bilinear: $e(aP, bP) = e(P, P)^{ab} = e(bP, aP)$, for all $a, b \in \mathbb{Z}_q$;
2. it can be efficiently computed for any possible input pair;
3. it is non-degenerate, which means that $e(P, P) \neq 1$.

Bilinear pairings can be constructed over groups defined on elliptic curves. In the last years, bilinear pairings have been widely used in cryptography, for example in the design of identity-based cryptographic protocols.

3 Extended Access Structures

Let $\mathcal{P} = \{R_1, \ldots, R_n\}$ be a set of players and let $\Gamma \subset 2^{\mathcal{P}}$ be a vector space access structure. Our goal is to create an *extended* access structure Γ', defined on an extended set $\mathcal{P}' = \mathcal{P} \cup \tilde{\mathcal{P}}$, where $\tilde{\mathcal{P}}$ denotes a set of *dummy* players, in such a way that:

1. $\mathcal{P} \in (\Gamma')_0$, that is, the set of real receivers is a minimal authorized subset of the extended access structure; and
2. $A \in \Gamma \iff A' = A \cup \tilde{\mathcal{P}} \in \Gamma'$, that is, an extended subset is authorized in the extended access structure if and only if the *real* members of this subset form an authorized subset in the original access structure.

In the following sections of the paper we will see how useful these properties are to design some encryption schemes. This relation between two access structures Γ and Γ' is not unusual at all. The opposite transformation, from Γ' to Γ, has been studied in secret sharing or matroid theory (see [22], for example). It is well known that if Γ' is a vector space access structure, then Γ (which in this case is denoted as a *minor* of Γ') is vector space, too.

We are going to show the same property for the opposite transformation. That is, given a vector space access structure Γ, we prove that an extension from Γ to Γ' is always possible, and that the vector space property of Γ is preserved in Γ'.

Proposition 1. *Let $\Gamma \subset 2^{\mathcal{P}}$ be a vector space access structure defined on a set $\mathcal{P}$ of n players. Then there exist a set of dummy players $\tilde{\mathcal{P}}$, satisfying $\tilde{\mathcal{P}} \cap \mathcal{P} = \emptyset$, and a vector space access structure $\Gamma' \subset 2^{\mathcal{P}'}$, where $\mathcal{P}' = \mathcal{P} \cup \tilde{\mathcal{P}}$, such that:*

1. *$\mathcal{P} \in (\Gamma')_0$, and*
2. *$A \in \Gamma \iff A' = A \cup \tilde{\mathcal{P}} \in \Gamma'$.*

Proof. Let $\psi : \mathcal{P} \cup \{D\} \longrightarrow (\mathbb{Z}_q)^d$ be the map which realizes Γ as a vector space access structure. If we denote as M the matrix whose n rows are the vectors $\psi(R_i)$, for $i = 1, \ldots, n$, then we can assume that the rank of M is d, because otherwise we can remove useless columns of M. This implies in particular that $d \leq n$.

We are going to construct a map $\psi' : \mathcal{P}' \cup \{D\} \longrightarrow (\mathbb{Z}_q)^n$ realizing a vector space access structure Γ' over the set $\mathcal{P}' = \mathcal{P} \cup \tilde{\mathcal{P}}$, where $\tilde{\mathcal{P}} \cap \mathcal{P} = \emptyset$ and $\tilde{\mathcal{P}}$ contains $n - d$ dummy players, such that Γ' satisfies the desired conditions.

First of all, since the d columns of M are linearly independent over $(\mathbb{Z}_q)^n$, we can extend them to a basis of $(\mathbb{Z}_q)^n$, via Steinitz, by adding $n - d$ columns. The resulting matrix M' has n linearly independent columns. The new rows are of the form $(\psi(R_i)|v_i)$, for $i = 1, \ldots, n$, for some vectors $v_i \in (\mathbb{Z}_q)^{n-d}$. These extended rows will be precisely the new vectors assigned to the real receivers. That is,

$$\psi'(R_i) = (\psi(R_i)|v_i) \in (\mathbb{Z}_q)^n, \quad \text{for } i = 1, \ldots, n.$$

Note that these vectors are a basis of $(\mathbb{Z}_q)^n$.

For the dummy players, that we denote $\tilde{\mathcal{P}} = \{R_{n+1}, \ldots, R_{2n-d}\}$, we consider a basis $\{w_j\}_{j=1,\ldots,n-d}$ of $(\mathbb{Z}_q)^{n-d}$, and we define

$$\psi'(R_{n+j}) = (\mathbf{0}|w_j) \in (\mathbb{Z}_q)^n, \quad \text{for } j = 1, \ldots, n-d.$$

Here $\mathbf{0}$ denotes a vector with d zeros. Now we have to define the vector $\psi'(D)$, in such a way that:

(i) $\psi'(D)$ is a linear combination, with all coefficients different from zero, of all the vectors in $\{\psi'(R_i)\}_{R_i \in \mathcal{P}}$. This would ensure that $\mathcal{P} \in (\Gamma')_0$.

(ii) $\psi'(D) = (a\ \psi(D)|w)$, for some $a \in \mathbb{Z}_q$ and some vector $w \in (\mathbb{Z}_q)^{n-d}$. This would ensure the second desired condition for Γ'. Indeed, if $A \in \Gamma$, we have $a\ \psi(D) = \sum_{R_i \in A} \lambda_i \psi(R_i)$, for some coefficients $\lambda_i \in \mathbb{Z}_q$. On the other hand, we have $w - \sum_{R_i \in A} \lambda_i v_i = \sum_{R_{n+j} \in \tilde{\mathcal{P}}} \mu_j w_j$, for some coefficients $\mu_j \in \mathbb{Z}_q$, because $\{w_j\}_{j=1,\ldots,n-d}$ is a basis of $(\mathbb{Z}_q)^{n-d}$. Summing up, we would have

$$\psi'(D) = (a\ \psi(D)|w) = \sum_{R_i \in A} \lambda_i \psi'(R_i) \ + \sum_{R_{n+j} \in \tilde{\mathcal{P}}} \mu_j \psi'(R_{n+j}).$$

And so $A \cup \tilde{\mathcal{P}} \in \Gamma'$. Reciprocally, if $A' \in \Gamma'$, then we would have $\psi'(D) = (a\ \psi(D)|w) = \sum_{R_i \in A'} \lambda_i \psi'(R_i)$. Since the vectors $\psi'(R_{n+j})$ of the dummy players have the first d components equal to 0, the vectors in $\{\psi(R_i)\}_{R_i \in A' \cap \mathcal{P}}$ would generate $a\ \psi(D)$ (and also $\psi(D)$) and so $A = A' \cap \mathcal{P} \in \Gamma$.

Let us show how such a vector $\psi'(D)$ can be constructed. We consider a minimal cover of $\mathcal{P}$ consisting of minimal authorized subsets $A_1, \ldots, A_r$, ordered in some arbitrary way. Note that such a cover must always exist, because otherwise there would be useless receivers in $\mathcal{P}$. We have $\psi(D) = \sum_{R_i \in A_\ell} \lambda_i^\ell \psi(R_i)$, for all $\ell = 1, \ldots, r$ and some coefficients $\lambda_i^\ell \in \mathbb{Z}_q$. The idea now is to multiply each of these equalities by a value $\alpha_\ell \in \mathbb{Z}_q$, and then to sum them all. If we define $a = \sum_{\ell=1}^r \alpha_\ell$, we obtain

$$a\ \psi(D) = \left(\sum_{\ell=1}^r \alpha_\ell\right) \psi(D) = \sum_{\ell=1}^r \sum_{R_i \in A_\ell} \alpha_\ell \lambda_i^\ell \psi(R_i) = \sum_{R_i \in \mathcal{P}} \left(\sum_{A_\ell \mid R_i \in A_\ell} \alpha_\ell \lambda_i^\ell \right) \psi(R_i).$$

We just have to ensure that all these coefficients $\rho_i = \sum_{A_\ell \mid R_i \in A_\ell} \alpha_\ell \lambda_i^\ell$ are different from zero, for $i = 1, \ldots, n$. If this is the case, then we will have that $\psi'(D) = \sum_{R_i \in \mathcal{P}} \rho_i \psi'(R_i)$ satisfies conditions (i) and (ii). To ensure $\rho_i \neq 0$ for all $i = 1, \ldots, n$, we consider a matrix $B = (b_{i\ell})$ with n rows, one for each $R_i \in \mathcal{P}$, and r columns, one for each authorized subset A_ℓ. We define $b_{i\ell} = \lambda_i^\ell$, if $R_i \in A_\ell$, and $b_{i\ell} = 0$ otherwise.

Now, we will define α_ℓ from $\ell = 1$ to $\ell = r$. For each column ℓ, we consider the players R_i such that A_ℓ is the last subset of the cover which contains R_i; in other

words, the rest of the i-th row of B, on the right of the ℓ-th column, contains only zeros. Note that for each column ℓ there will be at least one player R_i satisfying this condition; otherwise, the subset A_ℓ could be removed (but we are assuming that these subsets form a minimal covering). For these players R_i, since the values $\alpha_1, \ldots, \alpha_{\ell-1}$ are already defined, and the next values $\alpha_{\ell+1}, \ldots, \alpha_r$ do not affect these ρ_i, we choose a value for α_ℓ such that all the corresponding ρ_i are different from zero. More precisely, for each of these players R_i, there exists a unique value for α_ℓ which leads to $\rho_i = 0$. Therefore, we have at most n forbidden values. If we assume that $q \geq n$ (which will be the case in our encryption schemes, because q is a very large prime number), then there will exist a non-forbidden value for α_ℓ. Proceeding iteratively, we define all these values and obtain that $\rho_i \neq 0$, for all $R_i \in \mathcal{P}$, as desired. This completes the proof. □

The method described in this proof always works to realize an extended access structure Γ' from Γ, with the desired properties. For some particular cases of access structures Γ, however, there are more efficient and simple methods to construct an appropriate Γ', as we can see in the following section.

3.1 Particular Cases

In the following sections we will show how to use the concept of extended access structures in order to design both dynamic distributed encryption (DDE) and attribute-based encryption (ABE) schemes. An important parameter to measure the efficiency of such schemes is the length of the ciphertexts C. From this point of view, our schemes will not very efficient in general, because the ciphertexts must include the description of Γ', in particular all the vectors $\psi'(D), \{\psi'(R_i)\}_{R_i \in \mathcal{P} \cup \tilde{\mathcal{P}}}$. Note however that similar inefficiency problems will always appear as long as we want to consider general access structures Γ in DDE or ABE schemes, because the description of Γ and the secret sharing scheme which realizes it are always necessary.

However, for some particular cases of access structures Γ, it is possible to find an appropriate Γ' such that the description of the map ψ' can be made very short.

For example, let us consider the threshold case, where $\Gamma_{(t,n)} = \{A \subset \mathcal{P} : |A| \geq t\}$, for some threshold t such that $1 \leq t \leq n$. In this case, if $\mathcal{P} = \{R_1, \ldots, R_n\}$, we can define $\tilde{\mathcal{P}} = \{R_{n+1}, \ldots, R_{2n-t}\}$ and then the extended threshold access structure $\Gamma'_{(n,2n-t)} = \{A' \subset \mathcal{P} \cup \tilde{\mathcal{P}} : |A'| \geq n\}$ satisfies the desired conditions, stated in Proposition 1. This access structure can be realized by Shamir's secret sharing scheme, taking $\psi'(D) = (1, 0, 0, \ldots, 0) \in (\mathbb{Z}_q)^n$ and $\psi'(R_i) = (1, i, i^2, \ldots, i^{n-1}) \in (\mathbb{Z}_q)^n$, for all $R_i \in \mathcal{P} \cup \tilde{\mathcal{P}}$.

Note that, in general, in Shamir's threshold secret sharing scheme, each player R_i is associated with a different element $\alpha_i \in \mathbb{Z}_q$, and then the vector is defined as $\psi'(R_i) = (1, \alpha_i, \alpha_i^2, \ldots, \alpha_i^{n-1})$. This can be done by defining $\alpha_i = g(R_i)$ for some public and collision-resistant hash function $g : \{0,1\}^* \to \mathbb{Z}_q$. In this case, given the set $\mathcal{P}$ of n real players, finding an appropriate set $\tilde{\mathcal{P}}$ of $n-t$ dummy users such that $\mathcal{P} \cap \tilde{\mathcal{P}} = \emptyset$ and such that the description of $\tilde{\mathcal{P}}$ is short can be done in the following way: the sender looks for an interval of $n-t$ integers $J = \{j_0, j_0+1, \ldots, j_0+n-t-1\}$

(modulo q) such that $\alpha_i \notin J$ for all $R_i \in \mathcal{P}$, and defines the set $\tilde{\mathcal{P}}$ simply as the $n-t$ dummy users R_j whose associated values are $\alpha_j \in J$. Such an interval J exists as long as $n(n-t) < q-1$, which is very likely since q is a very large number. Note that the value j_0 is enough to describe the set $\tilde{\mathcal{P}}$, if the ciphertext already contains $\mathcal{P}$ (and so n) and the threshold t for the decryption.

Finally, there are other families of access structures Γ for which an appropriate Γ' can be found directly, without using the generic construction described in the proof of Proposition 1. In these cases, as it happens in the threshold case, Γ' is of the same kind as Γ, and so the description of Γ' in the ciphertext can be made very short, just by including the general parameters which define Γ and Γ', and the specific (usually well-known) secret sharing schemes that realize them. Some examples of such families of access structures are bipartite ones [23], hierarchical threshold ones [12, 29], weighted threshold ones [1], or compartmented ones [12, 30].

A very illustrative case is that of weighted threshold access structures. If Γ is such a structure, then there exist an assignment of positive integers $\omega : \mathcal{P} \to \mathbb{Z}_+$ and a threshold β such that $A \in \Gamma \Leftrightarrow \sum_{R_i \in A} \omega(R_i) \geq \beta$. In this case, we can easily obtain a suitable extended access structure Γ' by adding a single dummy user, i.e. $\tilde{\mathcal{P}} = \{R_{n+1}\}$, by defining the extended threshold as $\beta' = \sum_{R_i \in \mathcal{P}} \omega(R_i)$ and the new weights as $\omega'(R_i) = \omega(R_i)$ for the real users, and $\omega'(R_{n+1}) = \beta' - \beta$ for the dummy user.

4 First Application: Dynamic Distributed Encryption

In this section we propose the first construction of encryption schemes with dynamic distributed decryption which admits general access structures (not only threshold ones) for the subsets of receivers authorized to decrypt a message, and whose ciphertexts contain less than n elements (where n is the number of receivers).

The basic idea is to think of a standard distributed encryption scheme, with a global public key PK and a global secret key SK which is shared among the receivers, according to some access structure and secret sharing scheme. In our setting, however, the set of receivers and the global public key will not be always the same, but generated ad-hoc by the sender of each message. Furthermore, each potential receiver R_i has his own key pair (sk_i, pk_i), individually generated at the beginning of the life of the system.

Now suppose a sender wants to encrypt a message for a set of receivers $\mathcal{P}$ and a vector space access structure $\Gamma \subset 2^{\mathcal{P}}$. The idea for the encryption process is the following: the sender computes the *global* public key corresponding to Γ' (i.e. the public key whose implicit matching secret key can be obtained only from the secret keys of a subset in Γ') from the individual public keys of the real receivers, because $\mathcal{P} \in (\Gamma')_0$. Then, he encrypts the message under this global public key and adds to the ciphertext the partial decryption values of the dummy players $\tilde{\mathcal{P}}$. If members of an authorized set $A \in \Gamma$ of real receivers want to decrypt, they can combine their partial decryption values with the dummy ones, in the ciphertext, to form an authorized subset $A \cup \tilde{\mathcal{P}}$ for Γ', and then recover the plaintext.

Let us now describe the scheme in detail. It is based on ElGamal's cryptosystem. The five algorithms of our DDE scheme (DDE.Setup, DDE.KG, DDE.Enc, DDE.PartDec, DDE.Dec) work as follows.

Setup, DDE.Setup. Given a security parameter k, two prime numbers p and q are generated at random, such that q is k bits long and $q|p-1$. Then a cyclic subgroup $\mathbb{G} = \langle g \rangle$ of $\mathbb{Z}_p$ is chosen, with order q. Therefore, the output of the protocol is $\texttt{params} = (p, q, \mathbb{G}, g, h)$.

Key generation, DDE.KG. Each player R_i chooses at random his secret key $sk_i \in \mathbb{Z}_q^*$. The matching public key is $pk_i = g^{sk_i} \bmod p$. (We will sometimes omit the explicit $\bmod p$.)

Encryption, DDE.Enc. The goal is to encrypt a message $m \in \mathbb{G}$ addressed to some set $\mathcal{P} = \{R_1, \ldots, R_n\}$ of n receivers, with access structure $\Gamma \subset 2^{\mathcal{P}}$ for the decryption. We assume that Γ is a vector space access structure realized by some map $\psi : \mathcal{P} \cup \{D\} \longrightarrow (\mathbb{Z}_q)^d$. The sender finds an appropriate subset $\tilde{\mathcal{P}} = \{R_{n+1}, \ldots, R_{2n-d}\}$, access structure $\Gamma' \subset 2^{\mathcal{P}'}$, where $\mathcal{P}' = \mathcal{P} \cup \tilde{\mathcal{P}}$, and map $\psi' : \mathcal{P}' \cup \{D\} \longrightarrow (\mathbb{Z}_q)^n$ realizing Γ', by following the method explained in the proof of Proposition 1. Recall that the vectors in $\{\psi'(R_i)\}_{R_i \in \mathcal{P}}$ form a basis of $(\mathbb{Z}_q)^n$, so there exist coefficients $\lambda_{i0}^{\mathcal{P}}$, $\lambda_{ij}^{\mathcal{P}}$ such that

$$\psi'(D) = \sum_{R_i \in \mathcal{P}} \lambda_{i0}^{\mathcal{P}} \psi'(R_i) \quad \text{and } \psi'(R_j) = \sum_{R_i \in \mathcal{P}} \lambda_{ij}^{\mathcal{P}} \psi'(R_i),$$

for all the dummy players R_j, with $j = n+1, \ldots, 2n-d$.

The sender acts then as follows.

1. Define $PK = \prod_{R_i \in \mathcal{P}} pk_i^{\lambda_{i0}^{\mathcal{P}}} \bmod p$. Note that, if we write $SK = \sum_{R_i \in \mathcal{P}} \lambda_{i0}^{\mathcal{P}} sk_i$, we have that $PK = g^{SK}$. In other words, there is an implicit secret sharing in the exponent, according to Γ' (recall that $\mathcal{P} \in (\Gamma')_0$), where the secret is SK and the share of each real receiver R_i is his secret key sk_i.

2. For each dummy receiver $R_j \in \tilde{\mathcal{P}}$, define $pk_j = \prod_{R_i \in \mathcal{P}} pk_i^{\lambda_{ij}^{\mathcal{P}}} \bmod p$. Following the argument above, we could write $pk_j = g^{sk_j}$ for some element $sk_j \in \mathbb{Z}_q$; this element is the (implicit) secret share of the dummy user R_j in the secret sharing process which happens in the exponent of the public keys, with access structure Γ' and map ψ'.

3. Choose at random $a \in \mathbb{Z}_q^*$ and compute $r = g^a \bmod p$.

4. Compute $s = m \cdot PK^a \bmod p$.

5. For each $R_j \in \tilde{\mathcal{P}}$, compute the partial decryption $\kappa_j = pk_j^a \bmod p$, which is equal to r^{sk_j}.

6. Define the final ciphertext as $C = (\mathcal{P}, \Gamma, \tilde{\mathcal{P}}, \psi', r, s, \{\kappa_j\}_{R_j \in \tilde{\mathcal{P}}})$.

Note that the values PK and $\{pk_j\}_{R_j \in \tilde{\mathcal{P}}}$ are uniquely determined from the public keys of the real receivers and from ψ', so they can be re-used every time a message is encrypted for this set $\mathcal{P}$ and this access structure Γ.

Partial decryption, DDE.PartDec. Given a ciphertext $C = (\mathcal{P}, \Gamma, \tilde{\mathcal{P}}, \psi', r, s, \{\kappa_j\}_{R_j \in \tilde{\mathcal{P}}})$, any real receiver $R_i \in \mathcal{P}$ can compute his partial decryption $\kappa_i = r^{sk_i} \bmod p$.

Final decryption, DDE.Dec. Given a ciphertext $C = (\mathcal{P}, \Gamma, \tilde{\mathcal{P}}, \psi', r, s, \{\kappa_j\}_{R_j \in \tilde{\mathcal{P}}})$ and partial decryptions $\{\kappa_i\}_{R_i \in A}$ corresponding to receivers in some authorized subset $A \in \Gamma$, a combiner algorithm considers the whole set of partial decryptions in $A' = A \cup \tilde{\mathcal{P}}$. Due to the conditions on Γ', we have that $A' \in \Gamma'$, so there exist coefficients $\{\lambda_{i0}^{A'}\}_{R_i \in A'}$ such that $\psi'(D) = \sum_{R_i \in A'} \lambda_{i0}^{A'} \psi'(R_i)$. Translating this fact to the secret sharing which is implicitly performed in the exponents of the public keys, we have that $PK = g^{SK} = g^{\sum_{R_i \in A'} \lambda_{i0}^{A'} sk_i} = \prod_{R_i \in A'} pk_i^{\lambda_{i0}^{A'}}$.

The combiner therefore computes

$$\kappa = \prod_{R_i \in A'} \kappa_i^{\lambda_{i0}^{A'}} \bmod p = g^{a \sum_{R_i \in A'} \lambda_{i0}^{A'} sk_i} =$$

$$= g^{aSK} = PK^a.$$

Then the plaintext m is recovered as $m = s/\kappa \bmod p$.

4.1 Analysis: Security and Efficiency

Since this scheme is based on ElGamal (for example, considering $\mathcal{P} = \Gamma = \{R_j\}$, for a single receiver R_j, leads to ElGamal's standard cryptosystem), the achieved security can be at most the same as the security of ElGamal's cryptosystem. In Appendix A we formally prove that this DDE scheme is IND-CPA secure, assuming that the Decisional Diffie-Hellman problem is intractable.

It is possible to use our ideas of extended access structures, combined with the schemes and ideas in [6, 14] (as done in [16] for the threshold case), in order to obtain a DDE scheme for general access structures which is IND-CCA secure, in the standard model.

Regarding efficiency, and excluding the scheme in [17] which considers a different model for DDE, the new scheme is more efficient than all the previous proposals, in terms of ciphertexts' length. Specifically, in our scheme the length of a ciphertext is $n - d + \mathcal{O}(1)$, whereas all the proposed schemes (except the one in [16], which is the particular threshold case of our new scheme) have ciphertexts whose length is at least $n + \mathcal{O}(1)$, being n the number of receivers.

The improvement provided by our scheme, i.e. the value of d, depends on the degree of restriction of the family Γ. On the one hand, if the family is very restrictive, meaning that few subsets (with many members) can decrypt, then the value d will be high, and the length of the ciphertexts in our scheme will be smaller. On the other hand, if the decryption policy is permissive, then d will be smaller, and the length of our ciphertexts will be more or less the same as in other proposed DDE schemes.

5 Second Application: Attribute-Based Encryption

In this section we describe a ciphertext-policy ABE scheme which admits general access structures (or policies) for the subsets of attributes whose owners are authorized to decrypt a ciphertext. The essential ingredients for the design of our scheme are the identity-based encryption scheme of Boneh-Franklin [10] and the concept of extended access structures that we have introduced in this paper. Now the role of dummy players will correspond to dummy attributes, out of the set $\mathcal{P}$ of attributes admitted in the system.

The algorithms of our ABE scheme (ABE.Setup, ABE.Ext, ABE.Enc, ABE.Dec) work as follows.

Setup, ABE.Setup. Given a security parameter k, it generates a prime number q with k bits, an additive groups $\mathbb{G}_1 = \langle P \rangle$ and a multiplicative group $\mathbb{G}_2$, both with order q, which admit a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. A hash function $H : \{0,1\}^* \to \mathbb{G}_1$ is chosen. The secret key of the master entity consists of three random elements $msk = (\gamma, u, v) \in (\mathbb{Z}_q^*)^3$. The values $P_1 = \gamma P$, $V = \frac{1}{v}P$ and $U = \frac{u}{v}P$ will be part of the public parameters. Finally, the whole set $\mathcal{P}$ of possible attributes is chosen. The output of the protocol is the master secret key $msk = (\gamma, u, v)$ and the public parameters $\texttt{params} = (\mathcal{P}, q, \mathbb{G}_1, \mathbb{G}_2, P, e, H, P_1, V, U)$.

Key Extraction, ABE.Ext. A user proves to the master entity that he holds a subset of attributes $A \subset \mathcal{P}$. Once the master entity verifies the correctness of this proof, she chooses a fresh random value $t \in \mathbb{Z}_q^*$, computes $T = tP$, $T_u = utP$ and, for each $at_i \in A$, computes the value $Q_i = H(at_i)$ and then the pair $D_{i,u} = utQ_i + \gamma Q_i$ and $D_{i,v} = vtQ_i$. The resulting secret key is $sk_A = (T, T_u, \{(D_{i,u}, D_{i,v})\}_{at_i \in A})$.

Encryption, ABE.Enc. The goal is to encrypt a message $m \in \mathbb{G}_2$ addressed to some vector space access structure $\Gamma \subset 2^{\mathcal{P}}$ defined on the set of attributes $\mathcal{P} = \{at_1, \ldots, at_n\}$. Let $\psi : \mathcal{P} \cup \{D\} \longrightarrow (\mathbb{Z}_q)^d$ be the map that realizes Γ. The sender finds an appropriate subset $\tilde{\mathcal{P}} = \{at_{n+1}, \ldots, at_{2n-d}\}$, an extended access structure $\Gamma' \subset 2^{\mathcal{P}'}$, where $\mathcal{P}' = \mathcal{P} \cup \tilde{\mathcal{P}}$, and a map $\psi' : \mathcal{P}' \cup \{D\} \longrightarrow (\mathbb{Z}_q)^n$ realizing Γ', by following the method explained in the proof of Proposition 1. Recall that the vectors in $\{\psi'(at_i)\}_{at_i \in \mathcal{P}}$ form a basis of $(\mathbb{Z}_q)^n$, so there exist coefficients $\lambda_{i0}^{\mathcal{P}}$, $\lambda_{ij}^{\mathcal{P}}$ such that

$$\psi'(D) = \sum_{at_i \in \mathcal{P}} \lambda_{i0}^{\mathcal{P}} \psi'(at_i) \quad \text{and } \psi'(at_j) = \sum_{at_i \in \mathcal{P}} \lambda_{ij}^{\mathcal{P}} \psi'(at_i),$$

for all the dummy attributes at_j, with $j = n+1, \ldots, 2n-d$.

The sender acts then as follows.

1. Define $Q = \sum_{at_i \in \mathcal{P}} \lambda_{i0}^{\mathcal{P}} Q_i$, where $Q_i = H(at_i)$.
2. For each $at_j \in \tilde{\mathcal{P}}$, define $Q_j = \sum_{at_i \in \mathcal{P}} \lambda_{ij}^{\mathcal{P}} Q_i$.
3. Choose at random $r_1, x \in \mathbb{Z}_q^*$, and (implicitly) define r_2 such that $ur_1 + vr_2 = x \bmod q$.
4. Compute $C_1 = r_1 P$ and $C_2 = r_2 P$, as $C_2 = xV - r_1 U$.
5. Compute $C_3 = m \cdot e(P_1, Q)^{r_1}$.
6. Compute $X = xQ$.
7. For each $at_j \in \tilde{\mathcal{P}}$, compute the pair of values $\kappa_{j,1} = r_1 Q_j$ and $\kappa_{j,x} = xQ_j$.
8. Define the final ciphertext as $C = (\mathcal{P}, \Gamma, \tilde{\mathcal{P}}, \psi', C_1, C_2, C_3, X, \{(\kappa_{j,1}, \kappa_{j,x})\}_{at_j \in \tilde{\mathcal{P}}})$.

Decryption, ABE.Dec. Given a ciphertext $C = (\mathcal{P}, \Gamma, \tilde{\mathcal{P}}, \psi', C_1, C_2, C_3, X, \{(\kappa_{j,1}, \kappa_{j,x})\}_{at_j \in \tilde{\mathcal{P}}})$, a user with secret key $sk_A = (T, T_u, \{(D_{i,u}, D_{i,v})\}_{at_i \in A})$ for a subset of attributes $A \in \Gamma$ can recover the encrypted message, as follows. Recall that $A' = A \cup \tilde{\mathcal{P}} \in \Gamma'$ and, therefore, there exist coefficients $\{\lambda_{i0}^{A'}\}_{at_i \in A'}$ such that $\psi'(D) = \sum_{at_i \in A'} \lambda_{i0}^{A'} \psi'(at_i)$ and so $Q = \sum_{at_i \in A'} \lambda_{i0}^{A'} Q_i$.

The user computes $\kappa =$

$$\frac{e\left(T_u + P_1\ ,\ \sum_{at_j \in \tilde{\mathcal{P}}} \lambda_{j0}^{A'} \kappa_{j,1}\right) \cdot e\left(T\ ,\ \sum_{at_j \in \tilde{\mathcal{P}}} \lambda_{j0}^{A'} \kappa_{j,x}\right) \cdot e\left(C_1, \sum_{at_i \in A} \lambda_{i0}^{A'} D_{i,u}\right) \cdot e\left(C_2, \sum_{at_i \in A} \lambda_{i0}^{A'} D_{i,v}\right)}{e\left(T_u\ ,\ \sum_{at_j \in \tilde{\mathcal{P}}} \lambda_{j0}^{A'} \kappa_{j,1}\right) \cdot e(T, X)} =$$

$$= \ldots = \frac{e(utP, r_1 Q) \cdot e(vtP, r_2 Q) \cdot e(P_1, r_1 Q)}{e(tP, xQ)} = e(P_1, Q)^{r_1}.$$

The plaintext m is recovered by computing $m = C_3 / \kappa$.

5.1 Analysis: Security and Efficiency

We will only analyze the performance of our scheme in comparison with the scheme of Waters [31], which seems to be the most competitive ABE scheme up to date, in terms of efficiency, security and flexibility.

As it happens in the scheme of Waters, our scheme admits general (not necessarily threshold) families Γ of authorized subsets of attributes that must be held by a receiver in order to correctly decrypt. Regarding security, our scheme achieves the

same level of security as Waters' scheme: it enjoys selective CPA-security under the assumption that the decisional ℓ-Bilinear Diffie-Hellman Exponent problem is hard. The formal proof of this result can be found in Appendix B.

Finally, with respect to efficiency, the two schemes are very similar, for example in terms of the computational cost for encryption and decryption. The main difference between the two schemes is the length of the ciphertexts. In the scheme of Waters, this length (without considering the description of $\mathcal{P}$ and Γ) is $n + \mathcal{O}(1)$, being n the total number of attributes. In the case of our scheme, the length of a ciphertext is $2(n - d) + \mathcal{O}(1)$. Roughly speaking, our scheme is more efficient than Waters' scheme, in terms of ciphertext length, if and only if $d \geq n/2$. Note that the larger the value d is, the more restrictive the family Γ of authorized subsets of attributes is. In other words, our scheme can be more suitable for situations where the decryption ability is restricted to few persons, holding many attributes. For example, let us think of the threshold case in a system which considers $n = 10$ attributes in total. If a sender wants to encrypt a very confidential message, in such a way that only those people holding at least $t = 8$ attributes will be able to decrypt, then the length of the ciphertext in our scheme will be $4+\mathcal{O}(1)$, whereas it will be $10+\mathcal{O}(1)$ in Waters' scheme. Summing up, it turns out that the ABE scheme that we have constructed by using the concept of extended access structures is, essentially, as efficient as the best existing schemes of this kind.

6 Conclusion

We have introduced the concept of extended access structure, where a set of dummy players is added to an existing set of real players. We have used linear algebra tools to prove that any vector space access structure Γ admits an extended access structure Γ' which is also a vector space one.

We believe that extended access structure can be a useful tool to design distributed cryptographic protocols. To support this claim, we have given two practical applications of this concept: a dynamic distributed encryption scheme and an attribute-based encryption scheme. Both constructions improve over the existing schemes of these kinds, specially regarding the size of the ciphertexts. We believe these results bring a nice application of linear algebra to the construction of cryptographic protocols.

References

[1] A. Beimel, T. Tassa and E. Weinreb. (2005) Characterizing ideal weighted threshold secret sharing. *Proceedings of TCC'05*, LNCS **3378**, Springer-Verlag, pp. 600–619.

[2] M. Bellare, A. Boldyreva and S. Micali. (2000) Public-key encryption in a multi-user setting: security proofs and improvements. *Proceedings of Eurocrypt'00*, LNCS **1807**, Springer-Verlag, pp. 259–274.

[3] M. Bellare and P. Rogaway. (1993) Random oracles are practical: a paradigm for designing efficient protocols. *Proceedings of Computer and Communications Security, CCS'93*, ACM, pp. 62–73.

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

[5] G.R. Blakley. Safeguarding cryptographic keys. *Proceedings of the National Computer Conference, American Federation of Information*, Processing Societies Proceedings **48**, pp. 313–317 (1979).

[6] D. Boneh and X. Boyen. (2004) Efficient selective-ID secure identity-based encryption without random oracles. *Proceedings of Eurocrypt'04*, LNCS **3027**, Springer-Verlag, pp. 223–238.

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

[8] D. Boneh, X. Boyen and S. Halevi. (2006) Chosen ciphertext secure public key threshold encryption without random oracles. *Proceedings of CT-RSA'06*, LNCS **3860**, Springer-Verlag, pp. 226–243.

[9] D. Boneh, R. Canetti, J. Katz and S. Halevi. (2007) Chosen-ciphertext security from identity-based encryption, *SIAM Journal on Computing*, vol. **36** (5), pp. 1301–1328.

[10] D. Boneh and M.K. Franklin. (2003) Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, vol. **32** (3), pp. 586–615.

[11] D. Boneh, C. Gentry and B. Waters. (2005) Collusion resistant broadcast encryption with short ciphertexts and private keys. *Proceedings of Crypto'05*, LNCS **3621**, Springer-Verlag, pp. 258–275.

[12] E.F. Brickell.(1989) Some ideal secret sharing schemes. *Journal of Combinatorial Mathematics and Combinatorial Computing*, **9**, pp. 105–113 .

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

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

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

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

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

[18] A. Fiat and M. Naor. (1994) Broadcast encryption. *Proceedings of Crypto'93*, LNCS **773**, Springer-Verlag, pp. 480–491.

[19] H. Ghodosi, J. Pieprzyk and R. Safavi-Naini. (1996) Dynamic threshold cryptosystems: a new scheme in group oriented cryptography. *Proceedings of Pragocrypt'96*, CTU Publishing house, pp. 370-379.

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

[21] C.H. Lim and P.J. Lee. (1997) Directed signatures and application to threshold cryptosystems. *Proceedings of Security Protocols Workshop'96*, LNCS **1189**, Springer-Verlag, pp. 131–138.

[22] J. Martí-Farré and C. Padró. (2007) On secret sharing schemes, matroids and polymatroids. *Proceedings of TCC'07*, LNCS **4392**, Springer-Verlag, pp. 273–290.

[23] C. Padró and G. Sáez. (2000) Secret sharing schemes with bipartite access structure. *IEEE Transactions on Information Theory*, **46** (7), pp. 2596–2604.

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

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

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

[27] V. Shoup and R. Gennaro. (2002) Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, vol. **15** (2), Springer-Verlag, pp. 75–96.

[28] G.J.Simmons, W. Jackson and K. Martin.(1991) The geometry of secret sharing schemes. *Bulletin of the ICA*, **1**, pp. 71–88 .

[29] T. Tassa. (2004) Hierarchical threshold secret sharing. *Proceedings of TCC'04*, LNCS **2951**, Springer-Verlag, pp. 473–490.

[30] T. Tassa and N. Dyn. (2006) Multipartite secret sharing by bivariate interpolation. *Proceedings of ICALP'06*, LNCS **4052**, Springer-Verlag, pp. 288–299.

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

A Security Analysis of the DDE Scheme

Indistinguishability under CPA, for DDE schemes, is formally defined by considering the following game that an attacker $\mathcal{A}$ plays against a challenger:

$\mathcal{U} = \emptyset$.
$\texttt{params} \leftarrow \text{DDE.Setup}(1^k)$.
Each time $\mathcal{A}$ requires the creation of a new user R_i, $(pk_i, sk_i) \leftarrow \text{DDE.KG}(\texttt{params})$ is executed and R_i is added to $\mathcal{U}$.
$(St, \mathcal{P}, \Gamma, m_0, m_1) \leftarrow \mathcal{A}^{Cor}(\text{find}, \{pk_i\}_{R_i \in \mathcal{U}})$.
$b \leftarrow \{0, 1\}$ at random.
$C^* \leftarrow \text{DDE.Enc}(\mathcal{P}, \{pk_i\}_{R_i \in \mathcal{P}}, \Gamma, m_b)$.
$b' \leftarrow \mathcal{A}^{Cor}(\text{guess}, C^*, St)$.

In both phases of the attack, $\mathcal{A}$ has access to a corruption oracle Cor: $\mathcal{A}$ submits to the oracle a user $R_i \in \mathcal{U}$, and must receive as answer his secret key sk_i. We denote as q_c the total number of such corruptions. Let $\mathcal{U}' \subset \mathcal{U}$ be the subset of users that $\mathcal{A}$ has corrupted during the attack. In order to consider meaningful and successful such an attack, we require $\mathcal{U}' \notin \Gamma$. Otherwise, $\mathcal{A}$ knows the secret key of an authorized subset of $\mathcal{P}$ and can decrypt C^* by himself, obtaining m_b.

The advantage of such an adversary $\mathcal{A}$ in breaking the CPA-security of the DDE scheme is defined as

$$\varepsilon = \left| \Pr[b' = b] - \frac{1}{2} \right|$$

A DDE scheme is said to be ε-indistinguishable under CPA if the advantage of any polynomial time attacker $\mathcal{A}$ is at most ε.

We are now ready to prove, by a reduction argument, that the DDE scheme that we have proposed in Section 4 enjoys CPA-security, if we assume that the decisional Diffie-Hellman problem is hard to solve.

Definition 1. *We define the decisional Diffie-Hellman (DDH, for short) problem in the group $\mathbb{G} = \langle g \rangle$ as follows. Let $x, y \in \mathbb{Z}_p$ be chosen at random; the input of the problem is the tuple (g, g^x, g^y). A challenger randomly chooses a bit $z \in \{0, 1\}$, defining $T = g^{xy}$ if $z = 1$ and T as a random element in $\mathbb{G}$ if $z = 0$. The goal for a solver of the problem is to find the correct bit z.*

A solver of the DDH problem in $\mathbb{G}$, which outputs a bit z', has advantage ε if

$$\left| \Pr[z' = z] - \frac{1}{2} \right| \geq \varepsilon$$

Through the following theorem we prove that a hypothetical attacker $\mathcal{A}$ against the CPA-security of our DDE scheme, with advantage ε, could be used to construct

a solver of the DDH problem with advantage $\frac{\varepsilon}{6(q_c+1)}$, where q_c is the number of corruption queries made by $\mathcal{A}$. Since the DDH problem is assumed to be hard, we conclude that ε must be negligible, which means that our DDE scheme enjoys CPA-security.

Theorem 1. *The DDE scheme in Section 4 has CPA-security, assuming that the DDH problem is hard in* $\mathbb{G}$.

Proof. Let us assume the existence of an attacker $\mathcal{A}$ against the CPA-security of our DDE scheme, and let us construct a solver of the DDH problem, for an instance (g, g^x, g^y, T).

We prepare the initialization of the attacker $\mathcal{A}$. Namely, every time that $\mathcal{A}$ requires the creation of a new user R_i, we choose at random $\gamma_i \in \mathbb{Z}_q^*$. Let $\mu \in (0,1)$ be a real number to be determined later. With probability μ, the value $c_i = 0$ is chosen, and then we define $pk_i = g^{\gamma_i}$ (in this case, $sk_i = \gamma_i$ is known to us). On the other hand, with probability $1-\mu$, the value $c_i = 1$ is chosen, and we define $pk_i = (g^x)^{\gamma_i}$ (in this case, we do not know the value of sk_i). The public keys pk_i are sent to $\mathcal{A}$. The values (c_i, γ_i) are stored in a table. We denote as $\mathcal{U}$ the total set of users created by $\mathcal{A}$.

$\mathcal{A}$ is allowed to corrupt some users. If $\mathcal{A}$ sends a corruption query for user R_i, we look for c_i in the table. If $c_i = 0$, then the value $sk_i = \gamma_i$ is sent back to $\mathcal{A}$. Otherwise, if $c_i = 1$, we abort and output a random bit $z' \in \{0,1\}$ as our answer to the DDH problem. If the number of corruption queries from $\mathcal{A}$ is q_c, then the probability that we do not abort in this phase is μ^{q_c}. Let $\mathcal{U}' \subset \mathcal{U}$ denote the subset of users that $\mathcal{A}$ corrupts during the attack.

At some point, $\mathcal{A}$ broadcasts a set $\mathcal{P} = \{R_1, \ldots, R_n\}$, an access structure $\Gamma \subset 2^{\mathcal{P}}$ for decryption, and two messages $m_0, m_1 \in \mathbb{G}_2$, such that the corrupted users do not form an authorized subset for Γ, i.e. $\mathcal{U} \cap \mathcal{P} \notin \Gamma$. This means that at least one user $R_u \in \mathcal{P}$ has not been corrupted by $\mathcal{A}$ (otherwise, Γ would be empty). With probability $1-\mu$, we have $c_u = 1$ and so $pk_u = (g^x)^{\gamma_u}$. In general, we define $\mathcal{P}_0 = \{R_i \in \mathcal{P} : c_i = 0\}$ and $\mathcal{P}_1 = \{R_\ell \in \mathcal{P} : c_\ell = 1\}$. As we have just said, $\mathcal{P}_1$ is not empty with probability at least $1-\mu$. If this is not the case, we abort and output a random bit $z' \in \{0,1\}$.

We run the method explained in the proof of Proposition 1 to obtain a set $\tilde{\mathcal{P}}$ of $n-d$ dummy players such that $\mathcal{P} \cap \tilde{\mathcal{P}} = \emptyset$, and an appropriate access structure Γ' with associated map ψ'.

For the challenge ciphertext C^* to be given to $\mathcal{A}$, we choose at random a bit $b \in \{0,1\}$. We first define $r = g^y$, and then we have to simulate the value $s = m_b \cdot PK^y \bmod p$. Remember that

$$PK = \prod_{R_i \in \mathcal{P}} pk_i^{\lambda_{i0}^{\mathcal{P}}} = \prod_{R_i \in \mathcal{P}_0} (g^{\gamma_i})^{\lambda_{i0}^{\mathcal{P}}} \prod_{R_\ell \in \mathcal{P}_1} ((g^x)^{\gamma_\ell})^{\lambda_{\ell 0}^{\mathcal{P}}}.$$

Therefore, we can define PK^y as

$$\prod_{R_i \in \mathcal{P}_0} ((g^y)^{\gamma_i})^{\lambda_{i0}^{\mathcal{P}}} \prod_{R_\ell \in \mathcal{P}_1} (T^{\gamma_\ell})^{\lambda_{\ell 0}^{\mathcal{P}}},$$

which will be a consistent definition if and only if $T = g^{xy}$. If T is a random element, then this value will also be completely random, and so the resulting $s = m_b \cdot PK^y$ will be completely independent on the bit b.

Finally, for each dummy user $R_j \in \tilde{\mathcal{P}}$, we must simulate the partial decryption $\kappa_j = pk_j^y \bmod p$, where

$$pk_j = \prod_{R_i \in \mathcal{P}} pk_i^{\lambda_{ij}^{\mathcal{P}}} = \prod_{R_i \in \mathcal{P}_0} (g^{\gamma_i})^{\lambda_{ij}^{\mathcal{P}}} \prod_{R_\ell \in \mathcal{P}_1} ((g^x)^{\gamma_\ell})^{\lambda_{\ell j}^{\mathcal{P}}}.$$

Using an analogous argument, we define

$$\kappa_j = \prod_{R_i \in \mathcal{P}_0} ((g^y)^{\gamma_i})^{\lambda_{ij}^{\mathcal{P}}} \prod_{R_\ell \in \mathcal{P}_1} (T^{\gamma_\ell})^{\lambda_{\ell j}^{\mathcal{P}}},$$

which is consistent, again, if and only if $T = g^{xy}$.

The challenge ciphertext is defined as $C^* = (\mathcal{P}, \Gamma, \tilde{\mathcal{P}}, \psi', r, s, \{\kappa_j\}_{R_j \in \tilde{\mathcal{P}}})$.

The attacker $\mathcal{A}$ eventually outputs a bit b'. If $b' = b$, then we output $z' = 1$ as our answer to the given instance of the DDH problem. If $b' \neq b$, then we output $z' = 0$.

Let us compute our success probability of solving the DDH problem. With probability $1/2$, we have $T = g^{xy}$ and so the challenge ciphertext is consistent and, by hypothesis, $\mathcal{A}$ guesses the correct bit b with probability $1/2 + \varepsilon$. On the other hand, with probability $1/2$, the value T is completely random, and in this case the view of $\mathcal{A}$ is independent of the bit b, and so $\mathcal{A}$ correctly guesses b with probability $1/2$. We have to take into account, as well, the event in which we abort, during the simulation of $\mathcal{A}$'s environment. Note that, when we abort, we guess the correct bit z with probability $1/2$. Putting all the pieces together, and denoting as ρ the probability that we do not abort in any phase, we have:

$$\Pr[\text{we succeed}] = \frac{1}{2}\Pr[\text{we succeed}/\ T = g^{xy}] \ + \ \frac{1}{2}\Pr[\text{we succeed}/\ T \text{ is random}] \geq$$

$$\geq \frac{1}{2}\left[\Pr[\text{we do not abort}] \cdot (\frac{1}{2} + \varepsilon) \ + \ \Pr[\text{we abort}] \cdot \frac{1}{2}\right] \ +$$

$$+\frac{1}{2} \cdot \frac{1}{2} \ \geq \ \frac{1}{4}\rho + \frac{1}{2}\rho\varepsilon + \frac{1}{4}(1-\rho) + \frac{1}{4} \ = \ \frac{1}{2} + \frac{\rho\varepsilon}{2}.$$

The probability that we do not abort at any point is $\rho \geq \mu^{q_c}(1-\mu)$. This value is maximized when $\mu = \frac{q_c}{q_c+1}$, which leads to

$$\rho \geq \left(\frac{1}{1+\frac{1}{q_c}}\right)^{q_c} \cdot \frac{1}{q_c+1} \geq \frac{1}{e} \cdot \frac{1}{q_c+1}.$$

Therefore, our advantage in solving the DDH problem is at least

$$\frac{\rho\varepsilon}{2} \geq \frac{\varepsilon}{2(q_c+1)e} \geq \frac{\varepsilon}{6(q_c+1)}.$$

□

B Security Analysis of the ABE Scheme

Selective CPA security for ABE schemes is defined by considering the following game that an attacker $\mathcal{A}$ plays against a challenger:

1. $\mathcal{A}$ selects a set $\mathcal{P}$ of attributes and a family $\Gamma \subset 2^{\mathcal{P}}$.
2. The challenger runs $(\texttt{params}, msk) \leftarrow \text{ABE.Setup}(1^k)$ and gives $\texttt{params}$ to $\mathcal{A}$.
3. Secret key queries: $\mathcal{A}$ adaptively sends subsets $B \notin \Gamma$, and must receive $sk_B \leftarrow \text{ABE.Ext}(\texttt{params}, B, msk)$ as answer.
4. $\mathcal{A}$ outputs two messages m_0, m_1 of the same length.
5. The challenger chooses a random bit $b \leftarrow \{0,1\}$, computes $C^* \leftarrow \text{ABE.Enc}(\texttt{params}, \Gamma, m_b)$ and gives C^* to $\mathcal{A}$.
6. Step 3 is repeated.
7. $\mathcal{A}$ outputs a bit b'.

If the specific ABE scheme employs some hash function H that is modeled as a random oracle in the security proof, then the attacker $\mathcal{A}$ can make hash queries to this oracle, for inputs x of his choice. $\mathcal{A}$ must receive as answer a completely random and independent value $H(x)$.

The advantage of such an adversary $\mathcal{A}$ in breaking the selective CPA-security of the ABE scheme is defined as

$$\varepsilon = \left| \Pr[b' = b] - \frac{1}{2} \right|$$

An ABE scheme is said to enjoy ε-selective CPA-security if the advantage of any polynomial time attacker $\mathcal{A}$ is at most ε.

The security of our ABE scheme will hold under the assumption that the following decisional problem related to bilinear pairings is hard.

Definition 2. *We define the decisional ℓ-Bilinear Diffie-Hellman Exponent (ℓ-BDHE, for short) problem in the group $\mathbb{G}_1 = \langle P \rangle$ as follows. Let $a, s \in \mathbb{Z}_q$ be chosen at random; the input of the problem is the tuple $y = (P, sP, aP, a^2P, \ldots, a^{\ell}P, , a^{\ell+2}P, \ldots, a^{2\ell}P)$. A challenger randomly chooses a bit $z \in \{0,1\}$, defining $R = e(P,P)^{a^{\ell+1}s}$ if $z = 1$ and R as a random element in $\mathbb{G}_2$ if $z = 0$. The goal for a solver of the problem is to find the correct bit z.*

A solver of the decisional ℓ-BDHE problem in $\mathbb{G}_1$, which outputs a bit z', has advantage ε if

$$\left| \Pr[z' = z] - \frac{1}{2} \right| \geq \varepsilon$$

This problem has been considered and studied in other works [7, 11]. In particular, the assumption that the decisional ℓ-Bilinear Diffie-Hellman Exponent is hard is proved to be generically secure in [7].

Again, we are going to use a reduction argument to prove that our ABE scheme enjoys selective CPA-security. We will assume the existence of an hypothetical attacker $\mathcal{A}$ against the selective CPA-security of the scheme, with advantage ε, and we will use $\mathcal{A}$ to construct a solver of the decisional ℓ-BDHE problem, with advantage $\varepsilon/2$. Under the assumption that this problem is hard, we conclude that ε must be negligible and so our ABE scheme is secure.

Theorem 2. *The ABE scheme in Section 5 has selective CPA-security, assuming that the decisional ℓ-BDHE problem is hard in $\mathbb{G}_1$, for $\ell = n$, the total number of real attributes.*

Proof. We assume the existence of a successful adversary $\mathcal{A}$ against the selective CPA-security of our scheme, with advantage ε, and we use $\mathcal{A}$ to solve an instance of the ℓ-BDHE problem. We start executing $\mathcal{A}$, which gives us the set $\mathcal{P} = \{at_1, \ldots, at_n\}$ of attributes and the access structure $\Gamma \subset 2^{\mathcal{P}}$ for the challenge ciphertext (by definition of selective security). We construct a suitable extended access structure $\Gamma' \subset 2^{\mathcal{P} \cup \tilde{\mathcal{P}}}$, where $\tilde{\mathcal{P}} = \{at_{n+1}, \ldots, at_{2n-d}\}$ is the set of dummy attributes. Let $\psi' : \mathcal{P} \cup \tilde{\mathcal{P}} \cup \{D\} \to (\mathbb{Z}_q)^n$ be the map realizing the extended access structure Γ'. Without loss of generality (applying if necessary a basis change), we can assume that $\psi'(D) = (1, 0, 0, \ldots, 0)$. We denote $\psi'(at_i) = (\psi'(at_i)^{(1)}, \ldots, \psi'(at_i)^{(n)})$.

Now we ask for an instance of the ℓ-BDHE problem, for $\ell = n$ (note that we will use both n and ℓ throughout the proof), and we receive (y, R), where $y = (P, sP, aP, a^2P, \ldots, a^{\ell}P, , a^{\ell+2}P, \ldots, a^{2\ell}P)$. Remember that the goal is to distinguish if $R = e(P,P)^{a^{\ell+1}s}$ or if R is a random element in $\mathbb{G}_2$.

We choose the public parameters of the ABE scheme as follows: we take at random $u, v \in \mathbb{Z}_q^*$, and we define $P_1 = aP$, $V = \frac{1}{v}P$ and $U = \frac{u}{v}P$. We give the resulting `params` to $\mathcal{A}$, which can then make queries for hash values (random oracle model) and for secret keys of subsets $B \notin \Gamma$.

Hash queries. Note that the only relevant queries are $H(at_i)$, for $at_i \in \mathcal{P}$. We define $Q = a^{\ell}P + \alpha_0 P$, for some random value $\alpha_0 \in \mathbb{Z}_q$. For each dummy attribute $at_j \in \tilde{\mathcal{P}}$, we take at random $\alpha_j \in \mathbb{Z}_q^*$ and define $Q_j = \alpha_j P$. Let $L \subset \mathcal{P}$, $L \in (\bar{\Gamma})_0$ be a maximal non-authorized subset. This implies that $\tilde{\mathcal{P}} \cup L \notin \Gamma'$, and that $\tilde{\mathcal{P}} \cup L \cup \{at_{i*}\} \in \Gamma'$ for any $at_{i*} \notin L$. For every $at_i \in L$, we take $\alpha_i \in \mathbb{Z}_q$ at random and we define

$$Q_i = H(at_i) = \psi'(at_i)^{(1)}(a^{\ell}P) + \psi'(at_i)^{(2)}(a^{\ell-1}P) + \ldots + \psi'(at_i)^{(n)}(aP) + \alpha_i P.$$

For each of the remaining real attributes $at_{i*} \notin L$, we know that $A' = \tilde{\mathcal{P}} \cup L \cup \{at_{i*}\} \in \Gamma'$, therefore there must exist coefficients $\lambda_{i0}^{A'}$ such that $\psi'(D) = \sum_{at_i \in A'} \lambda_{i0}^{A'} \psi'(at_i)$. We define

$$H(at_{i*}) = Q_{i*} = \frac{1}{\lambda_{i*0}^{A'}} \left(Q - \sum_{at_j \in \tilde{\mathcal{P}}} \lambda_{j0}^{A'} Q_j - \sum_{at_i \in L} \lambda_{i0}^{A'} Q_i \right) =$$

$$= \tilde{\psi}(at_{i*})^{(1)}(a^\ell P) + \ldots + \tilde{\psi}(at_{i*})^{(n)}(aP) + \alpha_{i*}P,$$

where $\alpha_{i*} = \frac{1}{\lambda_{i*0}^{A'}} \left(\alpha_0 - \sum_{at_j \in \tilde{\mathcal{P}}} \lambda_{j0}^{A'} \alpha_j - \sum_{at_i \in L} \lambda_{i0}^{A'} \alpha_i \right)$ and, for every $k = 1, \ldots, n$:

$$\tilde{\psi}(at_{i*})^{(k)} = \frac{1}{\lambda_{i*0}^{A'}} \left(\psi'(D)^{(k)} - \sum_{at_i \in L} \lambda_{i0}^{A'} \psi'(at_i)^{(k)} \right) = \psi'(at_{i*})^{(k)} + \frac{1}{\lambda_{i*0}^{A'}} \sum_{at_j \in \tilde{\mathcal{P}}} \lambda_{j0}^{A'} \psi'(at_j)^{(k)}.$$

Summing up, we have at the end $Q = a^\ell P + \alpha_0 P$, then $Q_j = \alpha_j P$ for $at_j \in \tilde{\mathcal{P}}$, and for $at_i \in \mathcal{P}$ we have

$$Q_i = H(at_i) = \tilde{\psi}(at_i)^{(1)}(a^\ell P) + \ldots + \tilde{\psi}(at_i)^{(n)}(aP) + \alpha_i P,$$

where $\tilde{\psi}(at_i)^{(k)}$ is either equal to $\psi'(at_i)^{(k)}$, when $at_i \in L$, or is otherwise equal to

$$\psi'(at_i)^{(k)} + \frac{1}{\lambda_{i0}^{A'}} \sum_{at_j \in \tilde{\mathcal{P}}} \lambda_{j0}^{A'} \psi(at_j)^{(k)}.$$

We must show that this is a consistent simulation of the random oracle model, i.e. that the values $H(at_i)$ are all random and independent. To see this, note that for every $at_i \in \mathcal{P} \cup \tilde{\mathcal{P}}$, we can write $Q_i = \beta_i P$ for some $\beta_i \in \mathbb{Z}_q$. For instance, for $at_j \in \tilde{\mathcal{P}}$ we have $\beta_j = \alpha_j$, and for $at_i \in L$ we have

$$\beta_i = \tilde{\psi}(at_i)^{(1)} a^\ell + \ldots + \tilde{\psi}(at_i)^{(n)} a + \alpha_i.$$

It is easy to check that these values $\{\beta_i\}_{at_i \in \mathcal{P} \cup \tilde{\mathcal{P}}}$ are a sharing, according to the secret sharing scheme defined by ψ', of the secret $a^\ell + \alpha_0$ (which is the discrete logarithm of Q in the basis P). This sharing has been randomly computed, by choosing at random the secret and the shares of the elements in a maximal non-authorized subset, $L \cup \tilde{\mathcal{P}}$. Therefore, this random sharing follows the same distribution as a random sharing in which the shares that are chosen at random are those of the minimal authorized subset $\mathcal{P}$. Since these shares in $\mathcal{P}$ are independent, we conclude that the values $H(at_i) = \beta_i P$, for $at_i \in \mathcal{P}$, are random and independent, as desired.

Secret key queries. If $\mathcal{A}$ requests a secret key for a subset of attributes $B \notin \Gamma$, we know that $B' = B \cup \tilde{\mathcal{P}} \notin \Gamma'$. By definition of the secrecy property of a vector space secret sharing scheme, any secret is equally possible given the set of shares of B'. In other words, there exists a vector $w = (w_1, \ldots, w_n)$ such that $w_1 = w \cdot \psi'(D) = \frac{1}{1-u}$ and such that $w \cdot \psi'(at_i) = 0$ for all $at_i \in B'$. We implicitly define $t = w_1 a + w_2 a^2 + \ldots + w_n a^n$. Then, from the data included in the instance y of the ℓ-BDHE problem, we can easily compute the values $T = tP$ and $T_u = u(tP)$. Finally, for the values $D_{i,u} = utQ_i + \gamma Q_i$ and $D_{i,v} = vtQ_i$, where $at_i \in B$, we have $D_{i,u} = (ut + a)Q_i$ and $D_{i,v} = vtQ_i$. The only problematic component of these two values is the one which multiplies $a^{\ell+1}P$, because the value $a^{\ell+1}P$ is not included in the instance y of the ℓ-BDHE problem. Recalling the special form of

$Q_i = \tilde{\psi}(at_i)^{(1)}(a^\ell P) + \ldots + \tilde{\psi}(at_i)^{(n)}(aP) + \alpha_i P$, we have that the coefficient of $a^{\ell+1}P$ in $D_{i,u}$ is

$$(uw_1 + 1)\tilde{\psi}(at_i)^{(1)} + w_2\tilde{\psi}(at_i)^{(2)} + \ldots + w_n\tilde{\psi}(at_i)^{(n)},$$

whereas the coefficient of $a^{\ell+1}P$ in $D_{i,v}$ is

$$w_1\tilde{\psi}(at_i)^{(1)} + w_2\tilde{\psi}(at_i)^{(2)} + \ldots + w_n\tilde{\psi}(at_i)^{(n)}.$$

Taking into account the form of $\tilde{\psi}(at_i)^{(k)}$, for $k = 1, \ldots, n$, the fact that $w_1 = \frac{1}{1-u}$ (which makes the two previous coefficients of $a^{\ell+1}P$, in both $D_{i,u}$ and $D_{i,v}$, equal) and the fact that $w \cdot \psi'(at_i) = 0$ for all attributes $at_i \in B' = B \cup \tilde{\mathcal{P}}$, it is easy to see that these problematic coefficients vanish, and so we can correctly simulate $D_{i,u}$ and $D_{i,v}$ by using the values included in y.

Challenge. At some point, $\mathcal{A}$ broadcasts two messages m_0, m_1 of the same length, to be challenged. We choose a bit $\beta \in \{0, 1\}$ at random, and compute an encryption C of m_β, as follows. We choose at random $x \in \mathbb{Z}_q^*$ and (implicitly) define $r_1 = s$ and r_2 such that $ur_1 + vr_2 = x$. In other words, we have $r_2 = \frac{x}{v} - \frac{us}{v}$.

For the elements of the challenge ciphertext C^*, remember that $Q = a^\ell P + \alpha_0 P$. We can compute $X = xQ$, $C_1 = sP$, $C_2 = r_2 P = \frac{x}{v}P - \frac{u}{v}(sP)$, $C_3 = m_\beta \cdot R \cdot e(aP, sP)^{\alpha_0}$, and then, for each $at_j \in \tilde{\mathcal{P}}$, the values $\kappa_{j,1} = sQ_j = \alpha_j(sP)$ and $\kappa_{j,x} = xQ_j$.

Note that the ciphertext C^* is consistent if and only if $R = e(P, P)^{a^{\ell+1}s}$. If R is a random value in $\mathbb{G}_2$, then the view of $\mathcal{A}$ is completely independent of the bit β, so the probability that $\mathcal{A}$ guesses β in this second case is $1/2$. We wait for $\mathcal{A}$'s answer $\beta' \in \{0, 1\}$. If $\beta' = \beta$, then we output $z' = 1$ as our answer to the ℓ-BDHE problem, meaning that $R = e(P, P)^{a^{\ell+1}s}$. Otherwise, if $\beta' \neq \beta$, we output $z' = 0$, meaning that R is a random element in $\mathbb{G}_2$.

By the definition of the decisional ℓ-BDHE problem, we have $R = e(P, P)^{a^{\ell+1}s}$ with probability $1/2$, and R is a random element with probability $1/2$, as well. Assuming that $\mathcal{A}$ guesses β with probability $1/2 + \varepsilon$, when the challenge ciphertext is consistent, we can compute our success probability in solving the decisional ℓ-BDHE problem as

$$\Pr[\text{we succeed}] = \frac{1}{2}\Pr[\text{we succeed}/\ R = e(P, P)^{a^{\ell+1}s}] +$$

$$+\frac{1}{2}\Pr[\text{we succeed}/\ R \text{ is random}] \geq$$

$$\geq \frac{1}{2} \cdot (\frac{1}{2} + \varepsilon) \ + \ \frac{1}{2} \cdot \frac{1}{2} \ \geq \ \frac{1}{2} + \frac{\rho\varepsilon}{2}.$$

The advantage that we obtain in solving the ℓ-BDHE problem is therefore half the advantage of $\mathcal{A}$ in breaking the selective CPA security of our ABE scheme.

□