Collaborative Multi-Authority KP-ABE for Shorter Keys and Parameters

Riccardo Longo (riccardolongomath@gmail.com)
Department of Mathematics, University of Trento, Italy

Chiara Marcolla (chiara.marcolla@unito.it)
Department of Mathematics, University of Turin, Italy

Massimiliano Sala (maxsalacodes@gmail.com)
Department of Mathematics, University of Trento, Italy

Abstract

Bilinear groups are often used to create Attribute-Based Encryption (ABE) algorithms. In our proposal, a Multiple-Authorities Key-Policy Attribute-Based Encryption scheme is constructed in which the authorities collaborate to achieve shorter keys and parameters, enhancing the efficiency of encryption and decryption.
We prove our system secure under an original variation of the bilinear Diffie-Hellman assumption, we also show its relation with other similar assumptions.

Keywords: ABE, KP-ABE, Multi Authority, Bilinear Groups, Diffie-Hellman Assumptions.

1 Introduction

The Attribute-Based Encryption (ABE), that provides access control functionality in encrypted data, developed from Identity Based Encryption, a scheme proposed by Shamir [Sha85] in 1985 with the first constructions obtained in 2001 by Boneh and Franklin [BF01] and Cocks [Coc01]. In 2005 Sahai and Waters [SW05] proposed the first schemes of Attributed Based Encryption and in a consecutive work, Goyal, Pandey, Sahai, and Waters [GPSW06] formulated the two complimentary forms of ABE which are nowadays standard: *ciphertext-policy ABE*, where the keys are associated with sets of attributes and ciphertexts are associated with access policies, and *key-policy ABE*, which is a scheme where the keys are associated with access policies and ciphertexts are associated with sets of attributes. Several developments and generalizations have been obtained for KP-ABE

[OSW07,ALDP11,AHL+12,HW13]. These schemes are constructed on bilinear groups (usually implemented through the Tate and Weil pairings on elliptic curves), and have a proof of security based on the original Diffie-Hellman assumption on bilinear groups or some slight variation. A first implementation of ciphertext-policy ABE has been achieved by Bethencourt et al. [BSW07] in 2007 but the proofs of security of the ciphertext-policy ABE remained unsatisfactory since they were based on an assumption independent of the algebraic structure of the group (the generic group model). It is only with the work of Waters [Wat11] that the first non-restricted ciphertext-policy ABE scheme was built with a security dependent on variations of the DH assumption on bilinear groups. Noteworthy are also the latest developments that aim to control dynamic users via revocation, e.g. [LCL+13] which exploits even more sophisticated assumptions on bilinear groups, including a variant of the subgroup decision problem. Recently new methods to construct ABE schemes have also been approached ([HRS14]).

The first multi authority KP-ABE scheme was presented in [LMS15]. In this system the authorities may be set up in any moment and without any coordination. Any party can act as an ABE authority by creating a public parameters and issuing private keys to different users. Moreover the encryptor can select a set of trusted authorities that will have to authenticate the potential decryptors.

Related works on multiple authorities (but limited to ciphertext-policy ABE) are [Cha07,CC09] and [LW11]. In [CC09], that is a improvement of [Cha07], the authors construct a simple-threshold schemes in the case where attributes are divided in disjoint sets, each controlled by a different authority. Whereas, in [LW11] Lewko and Waters propose a scheme where is not needed a central authority or coordination between the authorities, each controlling disjoint sets of attributes.

Our construction The scheme that we propose in this paper evolves from the scheme presented in [LMS15] exploiting the collaboration between authorities to improve the efficiency. It is a multi authority KP-ABE scheme and the authorities collaborate to achieve shorter keys and parameters, enhancing the efficiency of encryption and decryption.
Basically our scheme proceeds as follows: the first step is the creation of the parameters. Namely, each authority sets up independently its *master key* and then it collaborates together with the other authorities to create:

- a common *public key* utilized by users to encrypt,
- the *authority parameters* that will be used to generate *secret keys* (used to decrypt).

Once the *public key* is published, a user, who we will call Alice, chooses a set of attributes that describe her message and encrypts it using this key. Let Bob be another user, so he has an *access policy*. Suppose that Bob wants to decrypt Alice's message (note that he can do so if and only if the message has the attributes prescribed by his policy). Bob requests a *secret key* for his policy to every authority. Independently, each authority checks the policy pertinence and generates a secret key. Once he has obtained all keys, he can merge them and obtain a single compact key. In this way Bob may store and use them as a single key.

Note that, even if there are drawbacks in the overheads caused by the collaboration in the setup phase, we achieve much greater performances (with respect to [LMS15]) in encryption, decryption and key storage (the essential parts of a protocol) thanks to the drastic reduction of both public parameters and decryption keys. In fact, where in [LMS15] there are multiple sets of public parameters and multiple secret keys, here we compress them into only one set of public parameters and one secret key, therefore the size of the ciphertext is greatly reduced and the decryption becomes considerably faster.

Concerning the security of our schemes, unless every authority colludes, the existence of just one non-cheating authority guarantees that no illegitimate party (including authorities) has access to the encrypted data. More specifically, our schemes give a solution to address the following two problems:

(1) The authority is *honest but curious*, namely, it will provide correct keys to users but will also try to access to data beyond its competence. Obviously, if there is a single authority, which is the unique responsible to issue the keys, there is no way to prevent key escrow. Using a multi-authority schemes we bypass this problem.

(2) The authority has been *breached*, this happens when a user's keys embed access structures that *do not* faithfully represent that user's level of clearance, and so someone has access to keys with a higher level of clearance than the one they are due. This problem is more specific for KP-ABE. In fact, the authority has to assign to each user an appropriate access structure that represents what the user can and cannot decrypt. Therefore, the authority has to be trusted also to perform correct checks of the users' clearances and to assign correct access structures accordingly. Adding multiple authorities to the scheme gives to the encryptor the opportunity to request more guarantees about the legitimacy of the decryptor's clearance since each authority checks the users independently. The idea is to request that the decryption proceeds successfully only when a key for each authority is used. Note that if these authorities set up their parameters independently and during encryption

these parameters are bound together irrevocably, then no authority can single-handedly decrypt any ciphertext and thus key escrow is removed.

So our KP-ABE schemes guarantee a protection against both breaches and curiosity.

The scheme is proved secure under a slightly stronger variation of the classical BDH assumption (Definition 2.3).

Comparing our scheme with those proposed by Chase ([Cha07,CC09]), which are the first ABE schemes with "multiple-authorities", we note that it enjoys more general and expressive policies. Furthermore, it models a different setting, since we aim at adding a layer of security rather than distributing the control of the attributes. Indeed, we request redundant checks, therefore preventing more effectively unauthorized accesses, and prevent the ability of authorities to intrude into users' privacy.

Organization This paper is organized as follows. In Section 2 we present bilinear groups and the main security assumptions used for ABE schemes, alongside our original assumptions and a comparison between these assumptions. In Section 3 we present the main mathematical tools used in the construction of ABE schemes. In Section 4 we explain our scheme and also prove its security. In Section 5 a lower-bound on the complexity in generic bilinear groups is shown. Finally, conclusions are drawn in Section 6.

2 Complexity Assumptions on Bilinear Groups

This section covers background information necessary to understand KP-ABE schemes and their security. In particular, we give some mathematical notions about bilinear groups and our cryptographic assumption, that is, the decisional bilinear Diffie-Hellman assumption, with particular emphasis on its variations used to prove ABE schemes and their relations.

Let $\mathbb{G}_1, \mathbb{G}_2$ be groups of the same prime order p.

Definition 2.1 (Pairing)**.** *A symmetric pairing is a bilinear map e such that $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ has the following properties:*

- *Bilinearity:* $\forall g, h \in \mathbb{G}_1, \forall a, b \in \mathbb{Z}_p, \quad e(g^a, h^b) = e(g, h)^{ab}$.
- *Non-degeneracy: for g generator of $\mathbb{G}_1$, $e(g, g) \neq 1$.*

Definition 2.2 (Bilinear Group)**.** *$\mathbb{G}_1$ is a Bilinear group if the conditions above hold and both the group operations in $\mathbb{G}_1$ and $\mathbb{G}_2$ as well as the bilinear map e are efficiently computable.*

In the remainder of this section $\mathbb{G}_1$ and $\mathbb{G}_2$ are understood.

2.1 Security assumption on prime order bilinear groups

Decisional Bilinear Diffie-Hellman Assumption The Decisional Bilinear Diffie-Hellman (BDH) assumption is the basilar assumption used for proofs of indistinguishability in pairing-based cryptography. It has been first introduced in [BF01] by Boneh and Franklin and then widely used in a variety of proofs, including the one of the first ABE in [GPSW06]. It is defined as follows.

Let $a, b, s, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of the bilinear group $\mathbb{G}_1$. The decisional bilinear Diffie-Hellman (BDH) problem consists in constructing an algorithm $\mathcal{B}(A = g^a, B = g^b, S = g^s, T) \to \{0, 1\}$ to efficiently distinguish between the tuples $(A, B, S, e(g, g)^{abc})$ and $(A, B, S, e(g, g)^z)$ outputting respectively 1 and 0. The advantage of $\mathcal{B}$ in this case is clearly written as:

$$Adv_{\mathcal{B}} = \left| \Pr\left[\mathcal{B}(A, B, S, e(g, g)^{abs}) = 1\right] - \Pr\left[\mathcal{B}(A, B, S, e(g, g)^z) = 1\right] \right|$$

where the probability is taken over the random choice of the generator g, of a, b, s, z in $\mathbb{Z}_p$, and the random bits possibly consumed by $\mathcal{B}$ to compute the response.

Definition 2.3 (BDH Assumption)**.** *The decisional BDH assumption holds if no probabilistic polynomial-time algorithm $\mathcal{B}$ has a non-negligible advantage in solving the decisional BDH problem.*

Decisional Bilinear Diffie-Hellman Exponent Assumption The decisional q-Bilinear Diffie-Hellman Exponent (q-BDHE) problem has been used in various security proofs, starting from Boneh et. al. in [BBG05] to prove their hierarchical identity-based encryption scheme with constant-size ciphertext. Subsequently it has been used in various ABE proofs, e.g. [Wat11] and [HW13]. It is defined as follows.

Let $a, s \in \mathbb{Z}_p$ be chosen at random and g be a generator of $\mathbb{G}_1$. If an adversary is given

$$\vec{y} = \left(g^s, g^{a^i},\ i \in \{1, \dots, 2q\} \setminus \{q + 1\}\right)$$

it must be hard to distinguish $e(g, g)^{a^{q+1}s} \in \mathbb{G}_2$ from a random element $R \in \mathbb{G}_2$. $\mathcal{B}$ clearly has advantage ϵ in solving the decisional q-BDHE in $\mathbb{G}_1$ if

$$\left| Pr\left[\mathcal{B}\left(y, T = e(g, g)^{a^{q+1}s}\right) = 0\right] - Pr\left[\mathcal{B}\left(y, T = R\right) = 0\right] \right| \geq \epsilon$$

Definition 2.4 (q-BDHE Assumption)**.** *The decisional q-BDHE assumption holds if no polynomial-time algorithm $\mathcal{B}$ has a non-negligible advantage in solving the decisional q-BDHE problem.*

Decisional Parallel Bilinear Diffie-Hellman Exponent Assumption The decisional q-parallel Bilinear Diffie-Hellman Exponent (q-PBDHE) problem has been first introduced by Waters in [Wat11] to prove the security of his more general construction of a ciphertext-policy ABE scheme. It is defined as follows.

Let $a, s, b_j \in \mathbb{Z}_p,\ j = 1, \ldots, q$, be chosen at random and g be a generator of $\mathbb{G}_1$. If an adversary is given

$$\vec{y} = \begin{cases} g^{a^i}, g^{\frac{a^i}{b_j}} & i \in \{1, \ldots, 2q\} \setminus \{q+1\}, \forall j \in \{1, \ldots, q\} \\ g, g^s, g^{sb_j}, g^{sa^i \frac{b_k}{b_j}} & \forall i, j, k \in \{1, \ldots, q\}, k \neq j \end{cases}$$

it must be hard to distinguish $e(g,g)^{a^{q+1}s} \in \mathbb{G}_2$ from a random element $R \in \mathbb{G}_2$. $\mathcal{B}$ as usual has advantage ϵ in solving the decisional q-PBDHE in $\mathbb{G}_1$ if

$$\left| Pr\left[\mathcal{B}\left(\vec{y}, T = e(g,g)^{a^{q+1}s}\right) = 0\right] - Pr\left[\mathcal{B}\left(\vec{y}, T = R\right) = 0\right] \right| \geq \epsilon$$

Definition 2.5 (q-PBDHE Assumption)**.** *The decisional q-PBDHE assumption holds if no polynomial-time algorithm $\mathcal{B}$ has a non-negligible advantage in solving the decisional q-PBDHE problem.*

Decisional Bilinear x-Power Diffie-Hellman Assumption This is our first original assumption introduced. It is a variant of the basic BDH in which the attacker has an advantage not due to more elements at its disposal (as in the previous cases), but rather from more knowledge of the algebraic properties of its input elements. We formally define it as follows.

Let $a, c, s, z \in \mathbb{Z}_p$ be chosen at random, g be a generator of the bilinear group $\mathbb{G}_1$, $b = c^x$. The x-power decisional bilinear Diffie-Hellman (x-PBDH) problem consists in constructing an algorithm

$$\mathcal{B}(A = g^a, B = g^b, S = g^s, Z) \rightarrow \{0, 1\}$$

to efficiently distinguish between the tuples $(A, B, S, e(g,g)^{abs})$ and $(A, B, S, e(g,g)^z)$. For example, when $x = 2$ we are telling the attacker that b is a Quadratic Residue modulo p and so the attacker knows something on the private exponents. The advantage of $\mathcal{B}$ is defined, following the standard convention as:

$$Adv_{\mathcal{B}} = \left| \Pr\left[\mathcal{B}(A, B, S, e(g,g)^{abs}) = 1\right] - \Pr\left[\mathcal{B}(A, B, S, e(g,g)^z) = 1\right] \right|$$

where the probability is taken over the random choice of the generator g, of a, c, s, z in $\mathbb{Z}_p$, and the random bits possibly consumed by $\mathcal{B}$ to compute the response.

Definition 2.6 (x-PBDH Assumption)**.** *The decisional x-PBDH assumption holds if no probabilistic polynomial-time algorithm $\mathcal{B}$ has a non-negligible advantage in solving the decisional x-PBDH problem.*

Decisional Bilinear x-Roots Diffie-Hellman Assumption This is our other original assumption. It develops from the x-PBDH taking the direction taken by q-BDHE and q-PBDHE of giving to the attacker more group elements in input. In this case, we add to the algebraic insight on b the actual x-root (and also its powers). This stronger assumption is defined as follows.

Let $a, c, s, z \in \mathbb{Z}_p$ be chosen at random and g be a generator of the bilinear group $\mathbb{G}_1$, moreover set $b = c^x$. The x-roots decisional bilinear Diffie-Hellman (x-RBDH) problem consists in constructing an algorithm $\mathcal{B}(\vec{y}, Z) \to \{0, 1\}$ that given the values

$$\vec{y} = \left(g^s, g^{c^{\pm i}}, g^{ac^{i-1}},\ i \in \{1, \dots, x\}\right)$$

efficiently distinguishes between the tuples $(\vec{y}, e(g,g)^{abs})$ and $(\vec{y}, e(g,g)^z)$. The advantage of $\mathcal{B}$ is then:

$$Adv_{\mathcal{B}} = \left|\Pr[\mathcal{B}(\vec{y}, e(g,g)^{abs}) = 1] - \Pr[\mathcal{B}(\vec{y}, e(g,g)^z) = 1]\right|$$

where the probability is taken over the random choice of the generator g, of a, c, s, z in $\mathbb{Z}_p$, and the random bits possibly consumed by $\mathcal{B}$.

Definition 2.7 (x-RBDH Assumption)**.** *The decisional x-RBDH assumption holds if no probabilistic polynomial-time algorithm $\mathcal{B}$ has a non-negligible advantage in solving the decisional x-RBDH problem.*

2.2 Comparison between security assumptions

In this section, we prove the relations between the security assumptions that we have defined in the previous section. In Section 5 we show an adaption of these assumptions to the generic group model and we are able to prove a related security bound.

Lemma 2.8. *Decisional Parallel Bilinear Diffie-Hellman Exponent Assumption implies BDH Exponent Assumption that implies, in turn, Decisional Bilinear Diffie-Hellman Assumption:*

$$q\text{-}PBDHE \implies q\text{-}BDHE \implies BDH\,.$$

Proof. In these three problems we assign three different sets as input to the attacker: for the BDH problem $\mathcal{S}_{\text{BDH}} = \{g^a, g^b, g^s, e(g,g)^{abs}\}$, for the q-BDHE problem

$$\mathcal{S}_{\text{q-BDHE}} := \{g^a, g^{a^q}, g^s, e(g,g)^{a^{q+1}s}\} \cup \{g^{a^i} : 2 \le i \le 2q, \quad i \ne q, q+1\}.$$

For the q-PBDHE problem:

$$\mathcal{S}_{\text{q-BDHE}} := \left\{g^a, g^{a^q}, g^s, e(g,g)^{a^{q+1}s}\right\} \cup \left\{g, g^{sb_j}, g^{a^\iota}, g^{\frac{a^\iota}{b_j}}, g^{sa^i\frac{b_k}{b_j}}\right\}_{\substack{i,j,k\in\{1,\dots,q\},\, k\ne j \\ 2\le\iota\le 2q,\, \iota\ne q, q+1}}.$$

If one can beat the BDH assumption with b random, then it can also beat it with the extra information that $b = a^q$, in this case we have $e(g,g)^{a^{q+1}s} = e(g,g)^{aa^qs} = e(g,g)^{abs}$ and so:

$$\mathcal{S}_{\text{BDH}} \subseteq \mathcal{S}_{\text{q-BDHE}} \subseteq \mathcal{S}_{\text{qPBDHE}}.$$

So q-PBDHE Assumption implies q-BDHE Assumption that implies BDH Assumption. □

Lemma 2.9. *BDH x-Power Assumption and BDH x-Roots Assumption imply BDH Assumption. Moreover x-RBDH implies x-PBDH, and so we have:*

$$x\text{-}RBDH \implies x\text{-}PBDH \implies BDH.$$

If $GCD(x, p-1) = 1$, *then x-PBDH is equivalent to BDH.*

Proof. We recall that $\mathcal{S}_{\text{BDH}} := \{g^a, g^b, g^s, e(g,g)^{abs}\}$, whereas in Decisional Bilinear x-Power Diffie-Hellman problem we have the same set of BDH but with $b = c^x$. In Decisional Bilinear x-Roots Diffie-Hellman problem we have $\mathcal{S}_{\text{x-RBDH}} := \{g^s, g^{c^i}, g^{ac^{i-1}}, e(g,g)^{abs}\}$, where $b = c^x$ and $i \in \{1, \dots, x\}$. If one can beat the BDH assumption with b random, then it can also beat it with the extra information that $b = c^x$, so x-PBDH Assumption implies BDH Assumption. Moreover $\mathcal{S}_{\text{x-PBDH}} \subseteq \mathcal{S}_{\text{x-RBDH}}$. So x-RBDH Assumption implies x-PBDH Assumption that implies BDH Assumption.

Now we prove that if $GCD(x, p-1) = 1$, then x-PBDH is equivalent to BDH. In fact, let d be a generator of $\mathbb{Z}_p^*$, then d^x is also a generator of $\mathbb{Z}_p^*$ iff x and $p-1$ are coprime. Then for every $b \in \mathbb{Z}_p^*$ there is some i such that $b = (d^x)^i$, thus $b = c^x$ for $c = d^i$. Therefore we have:

$$\text{x-PBDH} \underset{GCD(x,p-1)=1}{\cong} \text{BDH}.$$

□

Finally, we have the following lemma.

Lemma 2.10. *q-BDHE implies x-PBDH.*

Proof. As in the proof of Lemma 2.8, if we set $b = a^q$ we obtain that $\mathcal{S}_{\text{x–PBDH}} \subseteq \mathcal{S}_{\text{q–BDHE}}$. □

We summarize what we have just proved in the following Theorem.

Theorem 2.11. *The security assumptions above satisfy the following relations:*

$$\begin{array}{ccccc} \text{BDH} & \Longleftarrow & \text{q-BDHE} & \Longleftarrow & \text{q PBDHE.} \\ {\scriptstyle GCD(x,p-1)=1}\Downarrow\ \Uparrow & & \swarrow & & \\ \text{x-PBDH} & \Longleftarrow & \text{x-RBDH.} & & \end{array}$$

3 Access Structures and Linear Secret Sharing Schemes

We do not prove original results here, we only provide what we need for our construction. See the cited references for more details on these arguments.

Access structures define who may and who may not access to the data, giving the sets of attributes that have clearance.

Definition 3.1 (Access Structure)**.** *An access structure $\mathbb{A}$ on a universe of attributes U is the set of the subsets $S \subseteq U$ that are authorized. That is, a set of attributes S satisfies the policy described by the access structure $\mathbb{A}$ if and only if $S \in \mathbb{A}$.*

They are used to describe a policy of access, that is the rules that prescribe who may access to the information. If these rules are constructed using only AND, OR and THRESHOLD operators on the attributes, then the access structure is *monotonic*.

Definition 3.2 (Monotonic Access Structure)**.** *An access structure $\mathbb{A}$ is said monotonic if given $S_0 \subseteq S_1 \subseteq U$ it holds*

$$S_0 \in \mathbb{A} \Longrightarrow S_1 \in \mathbb{A}$$

An interesting property is that monotonic access structures (i.e. access structures $\mathbb{A}$ such that if S is an authorized set and $S \subseteq S'$ then also S' is an authorized set) may be associated to linear secret sharing schemes (LSSS). In this setting the parties of the LSSS are the attributes of the access structure.

A LSSS may be defined as follows (adapted from [Bei96]).

Definition 3.3 (Linear Secret-Sharing Schemes (LSSS)). *A secret-sharing scheme* Π *over a set of parties* P *is called linear (over* $\mathbb{Z}_p$*) if*

(1) The shares for each party form a vector over $\mathbb{Z}_p$.

(2) There exists a matrix M *with* l *rows and* n *columns called the share-generating matrix for* Π. *For all* $i \in \{1, \ldots, l\}$ *the* i*-th row of* M *is labeled via a function* ρ, *that associates* M_i *to the party* $\rho(i)$. *Considering the vector* $\vec{v} = (s, r_2, \ldots, r_n) \in \mathbb{Z}_p^n$, *where* $s \in \mathbb{Z}_p$ *is the secret to be shared, and* $r_i \in \mathbb{Z}_p$, *with* $i \in \{2, \ldots, n\}$ *are randomly chosen, then* $M\vec{v}$ *is the vector of* l *shares of the secret* s *according to* Π. *The share* $(M\vec{v})_i = M_i\vec{v}$ *belongs to party* $\rho(i)$.

It is shown in [Bei96] that every linear secret sharing-scheme according to the above definition also enjoys the linear reconstruction property, defined as follows: suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subseteq \{1, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $w_i \in \mathbb{Z}_p$, with $i \in I$ such that, if λ_i are valid shares of any secret s according to Π, then

$$\sum_{i \in I} w_i \lambda_i = s \tag{1}$$

Furthermore, it is shown in [Bei96] that these constants w_i can be found in time polynomial in the size of the share-generating matrix M.

Note that the vector $(1, 0, \ldots, 0)$ is the "target" vector for the linear secret sharing scheme. Then, for any set of rows I in M, the target vector is in the span of I if and only if I is an authorized set. This means that if I is not authorized, then for any choice of $c \in \mathbb{Z}_p$ there will exist a vector $\vec{u}$ such that $u_1 = c$ and

$$Mi \cdot \vec{w} = 0 \qquad \forall i \in I$$

In the first ABE schemes the access formulas are typically described in terms of access trees. The appendix of [LW11] is suggested for a discussion of how to perform a conversion from access trees to LSSS.

See [GPSW06], [Bei96] and [LC10] for more details about LSSS and access structures.

4 Our Construction

This section is divided in three parts. We start with definitions of Collaborative Multi-Authority Key-Policy ABE and of CPA selective security. In the second part we present in detail our scheme and, finally, we use a variant of the BDH assumption (Definition 2.3) to prove the security of this scheme under in the selective set model.

4.1 Collaborative Multi Authority KP-ABE Structure and Security

In this scheme, the authorities set up independently their master key and they collaborate to create a common public key and some authority parameters that will be used to generate secret keys. There is a minimum collaboration during key generation, in the sense that authorities have to agree on the policy to assign to the user, or equivalently the user should ask for the same policy to every authority. To encrypt, the user chooses a set of attributes that describes the message (and thus determines which access structures may read it). The ciphertext is computed using the public key generated by the authorities in concert. When someone wants to decrypt, he needs a key for every authority and once he obtains all the pieces he can merge them and use them as a single key.
The formal definition of the scheme follows.

Let $\mathbb{G}_1$ be a bilinear group (chosen accordingly to an implicit security parameter λ), $g \in \mathbb{G}_1$ a generator of the group, and $\mathbb{A}$ an access structure on a universe of attributes U.

Definition 4.1 (Collaborative Multi-authority KP-ABE)**.** *A collaborative multi-authority Key-Policy ABE system for a message space $\mathcal{M}$, a universe of authorities X with $x = |X|$, and an access structure space $\mathcal{G}$ is composed of the following four algorithms:*

$\mathsf{Setup}(U, g, \mathbb{G}_1) \to (\mathsf{PK}_k, \mathsf{MK}_k, \mathsf{AP}_k)$. *The setup algorithm for the authority $k \in X$ takes as input the universe of attributes U and the bilinear group $\mathbb{G}_1$ alongside its generator g. It outputs the public parameters PK_k, the master key MK_k, and the authority parameters AP_k for that authority.*

$\mathsf{CollSetup}(\mathsf{MK}_k, \mathsf{PK}_k, \mathsf{AP}_k, \mathsf{PK}^{(h)}, \mathsf{AP}^{(h)}) \to (\mathsf{PK}^{(h+1)}, \mathsf{AP}^{(h+1)})$. *The collaborative part of setup asks the authority $k \in X$ to add their part to the final public key and authority parameters. It takes as input the master key MK_k for that authority and the i-th step of construction of the public key $\mathsf{PK}^{(h)}$, and of the authority parameters $\mathsf{AP}^{(h)}$. It outputs the next step of construction of the public key $\mathsf{PK}^{(h+1)}$ and authority parameters $\mathsf{AP}^{(h+1)}$. When $h = x = |X|$ then $\mathsf{PK}^{(x)} = \mathsf{PK}$ and $\mathsf{AP}^{(x)} = \mathsf{AP}$ i.e. the public and authority parameters key is completed since every authority has contributed. At this point PK is distributed among all users, while AP is shared only between authorities.*

$\mathsf{KeyGen}_k(\mathsf{MK}_k, \mathsf{AP}, (M, \rho)) \to \mathsf{SK}_k$. *The key generation algorithm for the authority $k \in X$ takes as input the master key MK_k of the authority and an access structure $\mathbb{A}$ in the form of an LSSS (M, ρ). It outputs a decryption key SK_k for that access structure.*

$\mathsf{Encrypt}(m, S, \mathsf{PK}) \to \mathsf{CT}$. *The encryption algorithm takes as input the public parameters PK, a message $m \in \mathcal{M}$ and a set of attributes $S \subseteq U$. It outputs the*

ciphertext CT *associated with the attribute set* S.

$\mathsf{Decrypt}(\mathsf{CT}, \{\mathsf{SK}_k\}_{k\in X}) \to m'$. *The decryption algorithm takes as input a ciphertext* CT *that was encrypted under a set* S *of attributes and a decryption key* SK_k *for every authority* $k \in A$. *Let* $\mathbb{A}$ *be the access structure of the keys* SK_k. *It outputs the message* m' *if and only if* $S \in \mathbb{A}$.

The security game is defined as follows.

Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{KeyGen}, \mathsf{Decrypt})$ be a CMA-KP-ABE scheme for a message space $\mathcal{M}$, a universe of authorities X and an access structure space $\mathcal{G}$ and consider the following CMA-KP-ABE experiment $\mathsf{CMA\text{-}KP\text{-}ABE\text{-}Exp}_{\mathcal{A},\mathcal{E}}(\lambda, U)$ for an adversary $\mathcal{A}$, parameter λ and attribute universe U:

Init. The adversary declares the set of attributes S that it wishes to be challenged upon. Moreover it selects the *honest authority* $k_0 \in X$.

Setup. The challenger runs the Setup and Collaborative Setup algorithms initializing the authorities, and gives to the adversary the the public key.

Phase I. The adversary issues queries for private keys of any authority, but k_0 answers only to queries for keys for access structures $\mathbb{A}$ such that $S \notin \mathbb{A}$. On the contrary the other authorities respond to every query.

Challenge. The adversary submits two equal length messages m_0 and m_1. The challenger flips a random coin b, and encrypts m_b with S. The ciphertext is passed to the adversary.

Phase II. Phase `I` is repeated.

Guess. The adversary outputs a guess b' of b.

The output of the experiment is 1 if $b' = b$, 0 otherwise.

Definition 4.2 (MA-KP-ABE Selective Security)**.** *The MA-KP-ABE scheme* $\mathcal{E}$ *is CPA selective secure (or secure against chosen-plaintext attacks) for attribute universe* U *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$, *there exists a negligible function* $negl$ *such that:*

$$\Pr[\mathsf{MA\text{-}KP\text{-}ABE\text{-}Exp}_{\mathcal{A},\mathcal{E}}(\lambda, U) = 1] \leq \frac{1}{2} + negl(\lambda).$$

4.2 The Scheme

This scheme plans a set X of authorities, each with their own parameters, that collaborate to create a common public key and it sets up an encryption algorithm that uses this public key so that an authorized key for each authority in X is required to successfully decrypt.
Our scheme consists of three randomized algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt})$

plus the collaborative steps $\mathsf{CollSetup}$, $\mathsf{CollKeygen}$ and decryption $\mathsf{Decrypt}$. The scheme works in a bilinear group $\mathbb{G}_1$ of prime order p, and uses LSSS matrices to share secrets according to the various access structures. Attributes are seen as elements of $\mathbb{Z}_p$.

The description of the algorithms follows.

$\mathsf{Setup}(U, g, \mathbb{G}_1) \to (\mathsf{PK}_k, \mathsf{MK}_k, \mathsf{AP}_k)$. Given the universe of attributes U and a generator g of $\mathbb{G}_1$ each authority sets up independently its parameters. For $k \in X$ the Authority k chooses uniformly at random $\alpha_k \in \mathbb{Z}_p$, and $z_{k,i} \in \mathbb{Z}_p$ for each $i \in U$. Then the public parameters PK_k and the master key MK_k are:

$$\mathsf{PK}_k = \left(Y_k = e(g,g)^{\alpha_k}, \left\{T_{k,i} = g^{z_{k,i}}\right\}_{i \in U}\right), \ \mathsf{MK}_k = \left(\alpha_k, \{z_{k,i}\}_{i \in U}\right), \ \mathsf{AP}_k = \left(\left\{V_{k,i} = g^{\frac{1}{z_{k,i}}}\right\}_{i \in U}\right)$$

$\mathsf{CollSetup}(\mathsf{MK}_k, \mathsf{PK}_k, \mathsf{AP}_k, \mathsf{PK}^{(h)}, \mathsf{AP}^{(h)}) \to (\mathsf{PK}^{(h+1)}, \mathsf{AP}^{(h+1)})$. The collaborative construction of the public key proceeds as follows:

- if $h = 0$ then the authority k is the first to participate, then it simply sets $\mathsf{PK}^{(1)} = \mathsf{PK}_k$, $\mathsf{AP}^{(1)} = \mathsf{AP}_k$
- if $h > 0$ then $\mathsf{PK}^{(h)} = \left(Y^{(h)}, \left\{T_i^{(h)}\right\}_{i \in U}\right)$, $\mathsf{AP}^{(h)} = \left(\left\{V_i^{(h)}\right\}_{i \in U}\right)$, so it sets

$$Y^{(h+1)} = Y^{(h)} \cdot Y_k, \qquad T_i^{(h+1)} = \left(T_i^{(h)}\right)^{z_{k,i}}, \quad V_i^{(h+1)} = \left(V_i^{(h)}\right)^{\frac{1}{z_{k,i}}} \quad \forall i \in U$$

Then it is easy to see that when the construction is complete the public key is:

$$\mathsf{PK}^{(x)} = \mathsf{PK} = \left(Y = e(g,g)^{\sum_{k \in X} \alpha_k}, \left\{T_i = g^{\prod_{k \in X} z_{k,i}}\right\}_{i \in U}\right), \ \mathsf{AP}^{(x)} = \mathsf{AP} = \left(\left\{V_i = g^{\frac{1}{\prod_{k \in X} z_{k,i}}}\right\}_{i \in U}\right)$$

$\mathsf{KeyGen}_k(\mathsf{MK}_k, \mathsf{AP}, (M, \rho)) \to \mathsf{SK}_k$. The key generation algorithm for the authority k takes as input the master secret key MK_k, the public parameters PK and an LSSS access structure (M, ρ), where M is an $l \times n$ matrix on $\mathbb{Z}_p$ and ρ is a function which associates rows of M to attributes. It chooses uniformly at random a vector $\vec{v}_k \in \mathbb{Z}_p^n$ such that $v_{k,1} = \alpha_k$. Then computes the shares $\lambda_{k,i} = M_{k,i}\vec{v}_k$ for $1 \le i \le l$ where $M_{k,i}$ is the i-th row of M_k. Then the private key SK_k is:

$$\mathsf{SK}_k = \left\{K_{k,i}^{(1)} = V_{\rho(i)}^{\lambda_{k,i}} = g^{\frac{\lambda_{k,i}}{\prod_{k \in X} z_{k,\rho(i)}}}\right\}_{1 \le i \le l}$$

$\mathsf{Encrypt}(m, S, \mathsf{PK}) \to \mathsf{CT}$. The encryption algorithm takes as input the public parameters, a set S of attributes and a message m to encrypt. It chooses $s \in \mathbb{Z}_p$ uniformly at random and then computes the ciphertext as:

$$\mathsf{CT} = \left(S, C' = m \cdot (Y)^s, \{C_i = (T_i)^s\}_{i \in S}\right)$$

$\mathsf{Decrypt}(\mathsf{CT}, \{\mathsf{SK}_k\}_{k\in X}) \to m'$. The input is a ciphertext for a set of attributes S and an authorized key for every authority. Let (M, ρ) be the LSSS associated to the keys, and suppose that S is authorized. The algorithm finds $w_i \in \mathbb{Z}_p, i \in I$ such that

$$\sum_{i\in I} \lambda_{k,i} w_i = \alpha_k \qquad \forall k \in X \tag{2}$$

for an appropriate subset $I \subseteq S$. To simplify the notation let $z_i := \prod_{k\in X} z_{k,i}$, the algorithm then proceeds to reconstruct the original message computing:

$$\begin{aligned} m' &= \frac{C'}{\prod_{i\in I} e(\prod_{k\in X} K_{k,i}, C_{\rho(i)})^{w_i}} \\ &= \frac{m \cdot \left(e(g,g)^{(\sum_{k\in X}\alpha_k)}\right)^s}{\prod_{i\in I} e\left(\prod_{k\in X} g^{\frac{\lambda_{k,i}}{z_{\rho(i)}}}, (g^{z_{\rho(i)}})^s\right)^{w_i}} \\ &= \frac{m \cdot e(g,g)^{s(\sum_{k\in X}\alpha_k)}}{e(g,g)^{s\sum_{k\in X}\sum_{i\in I} w_i\lambda_{k,i}}} \\ &\overset{*}{=} \frac{m \cdot e(g,g)^{s(\sum_{k\in X}\alpha_k)}}{e(g,g)^{s(\sum_{k\in X}\alpha_k)}} = m \end{aligned}$$

Where $\overset{*}{=}$ follows from the property (2).

Note that once the user has obtained the keys from every authority it can multiply these all together and store only $\mathsf{SK} = \{K_i = \prod_{k\in X} K_{k,i}\}_{1\le i\le l}$ since this is all he needs to perform the decryption, so actually only a key is needed with size l, hence the scheme is very efficient in terms of key-size.

4.3 Security

The scheme is proved secure under the x-PBDH assumption (where $x = |X|$ is the number of authorities) in the selective set security game described in Section 4.1. Recall that every authority but one is supposed curious (or corrupted or breached) and then it will issue even keys that have enough clearance for the target set of attributes, while the honest one issues only unauthorized keys. Thus if at least one authority remains trustworthy the scheme is secure.
The security is provided by the following theorem.

Theorem 4.3. *If an adversary can break the scheme with x authorities, then a simulator can be constructed to play the Decisional x-PBDH game with a non-negligible advantage.*

Proof. Suppose there exists a polynomial-time adversary $\mathcal{A}$, that can attack the scheme in the Selective-Set model with advantage ϵ. Then we claim that

a simulator $\mathcal{B}$ can be built that can play the Decisional x-PBDH game with advantage $\epsilon/2$. The simulation proceeds as follows.

Init The simulator takes in a x-PBDH challenge g, g^a, g^b, g^s, T. The adversary gives the algorithm the challenged set of attributes S.

Setup In the scheme the only public things outputted during this phase are the public parameters PK, the simulator emulates them as follows. The simulator chooses random $r_k \in \mathbb{Z}_p$ for $k \in X \setminus \{k_0\}$ and implicitly sets $\alpha_k = -r_k b$ for $k \in X \setminus \{k_0\}$ and $\alpha_{k_0} = ab + b\sum_{k\in X\setminus\{k_0\}} r_k$ by computing:

$$e(g,g)^{\alpha_{k_0}} = e(g^a, g^b) \prod_{k\in X\setminus\{k_0\}} (g^b, g^{r_k})$$
$$e(g,g)^{\alpha_k} = e(g^b, g^{-r_k}) \qquad \forall k \in X \setminus \{k_0\}$$

Then it chooses $z'_{k,i} \in \mathbb{Z}_p$ uniformly at random for each $i \in U$, $k \in X$ and implicitly sets

$$z_{k,i} = \begin{cases} z'_{k,i} & \text{if } i \in S \\ c z'_{k,i} & \text{if } i \notin S \end{cases}$$

Then it can compute the public key as:

$$Y = e(g^a, g^b), \qquad T_i = \begin{cases} g^{z'_i} & \text{if } i \in S \\ (g^b)^{z'_i} & \text{if } i \notin S \end{cases} \tag{3}$$

Using the previously introduced notation $z'_i := \prod_{k\in X} z'_{k,i}$ and noting that for $i \notin S$

$$z_i = \prod_{k\in X} z_{k,i} = \prod_{k\in X} c z'_{k,i} = c^x \prod_{k\in X} z'_{k,i} = b z'_i$$

Phase I In this phase the simulator answers private key queries. For the queries made to the authority k_0 the simulator has to compute the $K_{k_0,i}$ values of a key for an access structure (M, ρ) with dimension $l \times n$ that is not satisfied by S. Therefore for the properties of an LSSS it can find a vector $\vec{y} \in \mathbb{Z}_p^n$ with $y_1 = 1$ fixed such that

$$M_i \vec{y} = 0 \qquad \forall i \text{ such that } \rho(i) \in S \tag{4}$$

Then it chooses uniformly at random a vector $\vec{v} \in \mathbb{Z}_p^n$ and implicitly sets the shares of $\alpha_{k_0} = b(a + \sum_{k\in X\setminus\{k_0\}} r_k)$ as

$$\lambda_{k_0,i} = b \sum_{j=1}^{n} M_{i,j}(v_j + (a + \sum_{k\in X\setminus\{k_0\}} r_k - v_1) y_j)$$

Note that $\lambda_{k_0,i} = \sum_{j=1}^{n} M_{i,j} u_j$ where $u_j = b(v_j + (a + \sum_{k \in X \setminus \{k_0\}} r_k - v_1) y_j)$ thus $u_1 = b(v_1 + (a + \sum_{k \in X \setminus \{k_0\}} r_k - v_1)1) = ab + b \sum_{k \in X \setminus \{k_0\}} r_k = \alpha_{k_0}$ so the shares are valid. Note also that from (4) it follows that

$$\lambda_{k_0,i} = b \sum_{j=1}^{n} M_{i,j} v_j \quad \forall i \text{ such that } \rho(i) \in S$$

Thus if i is such that $\rho(i) \in S$ the simulator can compute

$$K_{k_0,i} = (g^b)^{\frac{\sum_{j=1}^{n} M_{i,j} v_j}{z'_{\rho(i)}}} = g^{\frac{\lambda_{k_0,i}}{z_{\rho(i)}}}$$

Otherwise, if i is such that $\rho(i) \notin S$ the simulator computes

$$K_{k_0,i} = g^{\frac{\sum_{j=1}^{n} M_{i,j}(v_j + (r - v_1) y_j)}{z'_{\rho(i)}}} (g^a)^{\frac{\sum_{j=1}^{n} M_{i,j} y_j}{z'_{\rho(i)}}} = g^{\frac{\lambda_{1,i}}{z_{\rho(i)}}}$$

Where the last equality follows from $z_{\rho(i)} = b z'_{\rho(i)}$. Finally for the queries to the other authorities $k \in X \setminus \{k_0\}$, the simulator chooses uniformly at random a vector $\vec{t_k} \in \mathbb{Z}_p^n$ such that $t_{k,1} = -r_k$ and implicitly sets the shares $\lambda_{k,i} = b \sum_{j=1}^{n} M_{i,j} t_{k,j}$ by computing

$$K_{k,i} = \begin{cases} (g^b)^{\frac{\sum_{j=1}^{n} M_{i,j} t_{k,j}}{z'_{\rho(i)}}} = g^{\frac{b \sum_{j=1}^{n} M_{i,j} t_{k,j}}{z'_{\rho(i)}}} = g^{\frac{\lambda_{k,i}}{z_{\rho(i)}}} & \text{if } \rho(i) \in S \\ g^{\frac{\sum_{j=1}^{n} M_{i,j} t_{k,j}}{z'_{\rho(i)}}} = g^{\frac{b \sum_{j=1}^{n} M_{i,j} t_{k,j}}{b z'_{\rho(i)}}} = g^{\frac{\lambda_{k,i}}{z_{\rho(i)}}} & \text{if } \rho(i) \notin S \end{cases}$$

Challenge The adversary gives two messages m_0, m_1 to the simulator. He flips a coin μ. He creates:

$$\begin{aligned} C' &= m_\mu \cdot T \stackrel{*}{=} m_\mu \cdot e(g,g)^{sab} \\ &= m_\mu \cdot \left(e(g,g)^{ab + b\left(\sum_{k \in X \setminus \{k_0\}} r_k\right)} \prod_{k \in X \setminus \{k_0\}} e(g,g)^{-b r_k} \right)^s \\ C_{k,i} &= (g^s)^{z'_{\rho(i)}} = g^{s z_{\rho(i)}} \qquad i \in S \end{aligned}$$

Where the equality $\stackrel{*}{=}$ holds if and only if the BDH challenge was a valid tuple (i.e. T is non-random).

Phase II During this phase the simulator acts exactly as in *Phase I*.

Guess The adversary will eventually output a guess μ' of μ. The simulator then outputs 0 to guess that $T = e(g,g)^{abs}$ if $\mu' = \mu$; otherwise, it outputs 1 to

indicate that it believes T is a random group element in $\mathbb{G}_2$. In fact when T is not random the simulator $\mathcal{B}$ gives a perfect simulation so it holds:

$$Pr\left[\mathcal{B}\left(\vec{y}, T = e(g,g)^{abs}\right) = 0\right] = \frac{1}{2} + \epsilon$$

On the contrary when T is a random element $R \in \mathbb{G}_2$ the message m_μ is completely hidden from the adversary point of view, so:

$$Pr\left[\mathcal{B}\left(\vec{y}, T = R\right) = 0\right] = \frac{1}{2}$$

Therefore, $\mathcal{B}$ can play the decisional BDH game with non-negligible advantage $\frac{\epsilon}{2}$. □

5 Generic Security of Diffie-Hellman Assumptions

In [BBG05] Boneh et. al. stated and proved a theorem that gives a lower bound on the advantage of a generic algorithm in solving a class of decisional Diffie-Hellman problem. Despite a lower bound in generic groups does not imply a lower bound in any specific group, it still provides evidence of soundness of the assumptions. In this section: first the general Diffie-Hellman Exponent Problem is defined, then the lower bound will be stated and finally we will show our claim, i.e., how the problems introduced in Section 2 may be seen as particular cases of the general problem.

5.1 General Diffie-Hellman Exponent Problem

Let p be a prime and let s, n be positive integers. Let $P, Q \in \mathbb{F}_p[X_1, \ldots, X_n]^s$ be two s-tuples of n-variate polynomials over $\mathbb{F}_p$ and let $f \in \mathbb{F}_p[X_1, \ldots, X_n]$. Let $P = (p_1, p_2, \ldots, p_s)$ and $Q = (q_1, q_2, \ldots, q_s)$, we require that $p_1 = q_1 = 1$. Moreover define:

$$P(x_1, \ldots, x_n) = (p_1(x_1, \ldots, x_n), \ldots, p_s(x_1, \ldots, x_n)) \in (\mathbb{F}_p)^s.$$

And similarly for the s-tuple Q. Let $\mathbb{G}_1, \mathbb{G}_2$ be groups of order p and let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a non-degenerate bilinear map. Let $g \in \mathbb{G}_1$ be a generator of $\mathbb{G}_1$ and set $g_2 = e(g,g) \in \mathbb{G}_2$. Let

$$H(x_1, \ldots, x_n) = \left(g^{P(x_1,\ldots,x_n)}, g_2^{Q(x_1,\ldots,x_n)}\right) \in \mathbb{G}_1^s \times \mathbb{G}_2^s,$$

we say that an algorithm $\mathcal{B}$ that outputs $b \in \{0,1\}$ has advantage ϵ in solving the Decision (P, Q, f)-Diffie-Hellman problem in $\mathbb{G}_1$ if

$$\left|Pr\left[\mathcal{B}\left(H(x_1, \ldots, x_n), g_2^{f(x_1,\ldots,x_n)}\right) = 0\right] - Pr\left[\mathcal{B}(H(x_1, \ldots, x_n), T) = 0\right]\right| > \epsilon$$

where the probability is over the random choice of generator $g \in \mathbb{G}_1$, the random choice of $x_1, \ldots, x_n$ in $\mathbb{F}_p$, the random choice of $T \in \mathbb{G}_2$, and the random bits consumed by $\mathcal{B}$.

Definition 5.1 (Dependence on (P, Q))**.** *Let $P, Q \in \mathbb{F}_p[X_1, \ldots, X_n]^s$ be two s-tuples of n-variate polynomials over $\mathbb{F}_p$. We say that a polynomial $f \in \mathbb{F}_p[X_1, \ldots, X_n]$ is dependent on the sets (P, Q) if there exist $s^2 + s$ constants $\{a_{i,j}\}_{i,j=1}^s, \{b_k\}_{k=1}^s$ such that*

$$f = \sum_{i,j=1}^{s} a_{i,j} p_i p_j + \sum_{k=1}^{s} b_k q_k$$

We say that f is independent of (P, Q) if f is not dependent on (P, Q).

For a polynomial $f \in \mathbb{F}_p[X_1, \ldots, X_n]^s$, we let d_f denote the total degree of f. For a set $P \subseteq \mathbb{F}_p[X_1, \ldots, X_n]^s$ we let $d_P = \max\{d_f : f \in P\}$.

5.2 Complexity Lower Bound in Generic Bilinear Groups

We state the following lower bound in the framework of the generic group model. We consider two random encodings ξ_0, ξ_1 of the additive group $\mathbb{Z}_p$, i.e. injective maps $\xi_0, \xi_1 : \mathbb{Z}_p \to \{0,1\}^m$. For $i = 0, 1$ we write $\mathbb{G}_i = \{\xi_i(x) : x \in \mathbb{Z}_p\}$. We are given oracles to compute the induced group action on $\mathbb{G}_1, \mathbb{G}_2$, and an oracle to compute a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. We refer to $\mathbb{G}_1$ as a *generic* bilinear group. The following theorem gives a lower bound on the advantage of a generic algorithm in solving the decision (P, Q, f)-Diffie-Hellman problem. We emphasize, however, that a lower bound in generic groups does not imply a lower bound in any specific group.

Theorem 5.2 (Theorem A.2 of [BBG05])**.** *Let $P, Q \in \mathbb{F}_p[X_1, \ldots, X_n]^s$ be two s-tuples of n-variate polynomials over $\mathbb{F}_p$ and let $f \in \mathbb{F}_p[X_1, \ldots, X_n]$. Let $d = \max(2d_P, d_Q, d_f)$. Let ξ_0, ξ_1 and $\mathbb{G}_1, \mathbb{G}_2$ be defined as above. If f is independent of (P, Q) then for any $\mathcal{A}$ that makes a total of at most q queries to the oracles computing the group operation in $\mathbb{G}_1, \mathbb{G}_2$ and the bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ we have:*

$$\left| Pr\left[\mathcal{A}(p, \xi_0(P(x_1, \ldots, x_n)), \xi_1(Q(x_1, \ldots, x_n)), \xi_1(t_0), \xi_1(t_1)) = b\right] - \frac{1}{2} \right| \leq \frac{(q + 2s + 2)^2 d}{2p}$$

Where $x_1, \ldots, x_n, y$ are chosen uniformly at random from $\mathbb{F}_p$, b is chosen uniformly at random from $\{0, 1\}$ and $t_b = f(x_1, \ldots, x_n), t_{1-b} = y$.

Corollary 5.3 (Corollary A.3 of [BBG05])**.** *Let $P, Q \in \mathbb{F}_p[X_1, \ldots, X_n]^s$ be two s-tuples of n-variate polynomials over $\mathbb{F}_p$ and let $f \in \mathbb{F}_p[X_1, \ldots, X_n]$. Let $d = \max(2d_P, d_Q, d_f)$. If f is independent of (P, Q) then any $\mathcal{A}$ that has advantage $\frac{1}{2}$ in solving the decision (P, Q, f)-Diffie-Hellman Problem in a generic bilinear group G must take time at least $\Omega(\frac{p}{d} - s)$.*

5.3 Using Corollary 5.3

We claim that the assumptions presented in Section 2 follow from Corollary 5.3 giving the sets P, Q that reduces them to the general bilinear Diffie-Hellman problem:

- BDH in $\mathbb{G}_1$: set $P = \{1, y, w, z\}, Q = \{1\}, f = ywz$.
- q-BDHE in $\mathbb{G}_1$: set $P = \{1, y, w^i\}$ with $i \in \{1, \dots, 2q\} \backslash \{q+1\}, Q = \{1\}, f = x^q y$.
- x-PBDH in $\mathbb{G}_1$: set $P = \{1, y, w^x, z\}, Q = \{1\}, f = y\, w^x\, z$.
- x-RBDH in $\mathbb{G}_1$: set $P = \{1, y, w^i, \frac{1}{w}^i, z\}$ with $i \in \{1 \dots, x\}, Q = \{1\}$, $f = yw^x z$.

It is easy to see that each f is independent to the respective sets P and Q, in fact multiplying any two polynomials in the sets P and then combining them linearly does not give the polynomial f. To see this explicitly in the case of x-RBDH, the complete list of terms that may be obtained combining any two polynomials of P follows:

$$1, w^i, w^{2i}, w^{-i}, w^{-2i}, y, yw^i, yw^{-i}, w^i z, w^{-i} z, z, yz \qquad i \in \{1, \dots, x\}$$

Since there is no monomial in which y, w, and z appear together, it is apparent that no linear combination of these terms may give $yw^x z$ as result, thus f is independent of P, Q.

Thus applying the Corollary 5.3 a lower bound on the computational complexity of these problems in the generic bilinear group is obtained.

For the q-PBDHE the argument is slightly less direct, see [Wat11].

6 Final Comments

Our construction evolves from the scheme presented in [LMS15] exploiting the collaboration between authorities to improve the efficiency. This scheme needs fewer parameters, since the collaboration permits to collapse the various public parameters in a single public key, significantly reducing the length of ciphertexts. Moreover, once all the single-keys have been obtained they may be collapsed into one too:

$$SK = \left\{ K_i = \prod_{k \in X} K_{k,i} = g^{\frac{\sum_{k \in X} \lambda_{k,i}}{\prod_{k \in X} z_{k,\rho(i)}}} \right\}_{1 \le i \le l} .$$

This scheme requires that each authority uses the same LSSS matrix to generate the single-key, but the assumption is not unreasonable since the matrix is directly related to the user's clearance. So for the price of collaboration steps that weigh down the setup (a phase that has to be executed

only once when the scheme is used), and an additional parameter shared by authorities, encryption, decryption and key-storage are greatly improved.

Remark 6.1 (Security Assumptions). In the proof of security (Theorem 4.3) it is supposed that only the final public key is actually public, that is, the authority parameters and the collaboration steps remain secret and the simulator has not to simulate them to the adversary. This allows us to use only the x-PBDH assumption (Definition 2.6) that is a weak version of the BDH assumption as seen in Section 2. If however we want to weaken the scheme and keep all the collaboration steps public, then the simulator needs to emulate these passages and in order to do this she needs more values. Specifically she needs the values $g^{c^{\pm i}}$ for $i \in \{1, \dots, x\}$ to correctly simulate the collaboration steps during setup and the authority parameters. In particular in Equation (3) instead of $(g^b)^{z'_i}$ use $(g^{c^i})^{\prod z_i}$ to simulate the construction steps of T_i and $(g^{c^{-i}})^{\frac{1}{\prod z_i}}$ to simulate V_i and its construction steps. So instead of the x-PBDH, the stronger x-RBDH is needed.

Remark 6.2 (Security Definitions). This scheme has been proven *IND-CPA selective* secure, that is after selecting the target parameters (in this case attribute set and authorities) the attacker may not distinguish between chosen ciphertext after encryption. We observe that although the scheme of [LW11] is proven *fully secure* (against selective security), the construction is made in composite bilinear groups. It is in fact compulsory when using Dual System encryption (introduced by Waters [Wat09] with techniques developed with Lewko [LW10]), but this has drawbacks in terms of group size (integer factorization has to be avoided) and the computations of pairings and group operations are less efficient. This fact leads to an alternative construction in prime order groups in the same paper, that however is proven secure only in the generic group and random oracle model. Therefor, we believe that our constructions in prime groups retain validity and interest, considering also that the proofs are in the standard model.

However, the definition of security may be extended modifying the the security games. To extend the definition of security to *CCA* (chosen ciphertext attacks) it is enough to add decryption queries to Phase I and Phase II (with the obvious restriction that the challenge ciphertext may not be the subject of a decryption query).

Moreover, to define *full security* (as opposed to selective security) it is sufficient to remove the Init stage and move the choice of targets by the adversary in the Challenge phase. In our scheme the target is the set of attributes S and the honest authority k_0. Note that in this case the restrictions in the queries of Phase I are eliminated to become restrictions in the choice of the targets: in fact the honest authority k_0 has to be chosen among the authorities that have not issued authorized keys for the target attribute set S about to be selected. Phase II is left unaltered, in the sense that the restrictions to the queries are the same as the ones in the Phase II of selective security.

Acknowledgements

The authors would like to thank the anonymous referees for their insightful comments.
This research has been partially supported by TELSY S.p.A.

References

[AHL+12] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. De Panafieu, C. Ràfols, et al., *Attribute-based encryption schemes with constant-size ciphertexts*, Theoretical Computer Science **422** (2012), 15–38.

[ALDP11] N. Attrapadung, B. Libert, and E. De Panafieu, *Expressive key-policy attribute-based encryption with constant-size ciphertexts*, Public Key Cryptography–PKC 2011, Springer, 2011, pp. 90–108.

[BBG05] D. Boneh, X. Boyen, and E.-J. Goh, *Hierarchical identity based encryption with constant size ciphertext*, Proc. of EUROCRYPT 05, LNCS, vol. 3494, 2005, pp. 440–456.

[Bei96] A. Beimel, *Secure schemes for secret sharing and key distribution*, Ph.D. thesis, Technion-Israel Institute of technology, Faculty of computer science, 1996.

[BF01] D. Boneh and M. Franklin, *Identity-based encryption from the weil pairing*, Advances in CryptologyCRYPTO 2001, Springer, 2001, pp. 213–229.

[BSW07] J. Bethencourt, A. Sahai, and B. Waters, *Ciphertext-policy attribute-based encryption*, Proc. of SP 07, 2007, pp. 321–334.

[CC09] M. Chase and S. SM Chow, *Improving privacy and security in multi-authority attribute-based encryption*, Proceedings of the 16th ACM conference on Computer and communications security, ACM, 2009, pp. 121–130.

[Cha07] M. Chase, *Multi-authority attribute based encryption*, Theory of Cryptography, Springer, 2007, pp. 515–534.

[Coc01] C. Cocks, *An identity based encryption scheme based on quadratic residues*, Cryptography and Coding, Springer, 2001, pp. 360–363.

[GPSW06] V. Goyal, O. Pandey, A. Sahai, and B. Waters, *Attribute-based encryption for fine-grained access control of encrypted data*, Proc. of CCS 06, 2006, pp. 89–98.

[HRS14] Javier Herranz, Alexandre Ruiz, and Germán Sáez, *New results and applications for multi-secret sharing schemes*, Designs, Codes and Cryptography **73** (2014), no. 3, 841–864.

[HW13] S. Hohenberger and B. Waters, *Attribute-based encryption with fast decryption*, Proc. of PKC 13, LNCS, vol. 7778, 2013, pp. 162–179.

[LC10] Z. Liu and Z. Cao, *On efficiently transferring the linear secret-sharing scheme matrix in ciphertext-policy attribute-based encryption.*, IACR Cryptology ePrint Archive (2010).

[LCL+13] K. Lee, Seung G. Choi, D. H. Lee, J. H. Park, and M. Yung, *Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency*, Proc. of ASIACRYPT 13, LNCS, vol. 8270, 2013, pp. 235–254.

[LMS15] R. Longo, C. Marcolla, and M. Sala, *Key-policy multi-authority attribute-based encryption*, Algebraic Informatics, Springer, 2015, pp. 152–164.

[LW10] A. Lewko and B. Waters, *New techniques for dual system encryption and fully secure hibe with short ciphertexts*, Theory of Cryptography, LNCS, vol. 5978, 2010, pp. 455–479.

[LW11] ———, *Decentralizing attribute-based encryption*, Proc. of EUROCRYPT 11, LNCS, vol. 6632, 2011, pp. 568–588.

[OSW07] R. Ostrovsky, A. Sahai, and B. Waters, *Attribute-based encryption with non-monotonic access structures*, Proc. of CCS 07, 2007, pp. 195–203.

[Sha85] A. Shamir, *Identity-based cryptosystems and signature schemes*, Advances in cryptology, Springer, 1985, pp. 47–53.

[SW05] A. Sahai and B. Waters, *Fuzzy identity-based encryption*, Advances in Cryptology–EUROCRYPT 2005, Springer, 2005, pp. 457–473.

[Wat09] B. Waters, *Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions*, Proc. of CRYPTO 09, LNCS, vol. 5677, 2009, pp. 619–636.

[Wat11] ———, *Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization*, Proc. of PKC 11, LNCS, vol. 6571, 2011, pp. 53–70.