Two novel applications of bilinear groups to ABE encryption

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

Chiara Marcolla chiara.marcolla@unitn.it
Department of Mathematics, University of Trento, 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 particular, they have been used to create an ABE system with multi authorities, but limited to the ciphertext-policy instance. Here, for the first time, we propose two multi-authority key-policy ABE systems.
In our first proposal, the authorities may be set up in any moment and without any coordination. A party can simply act as an ABE authority by creating its own public parameters and issuing private keys to the users. A user can thus encrypt data choosing both a set of attributes and a set of trusted authorities, maintaining full control unless all his chosen authorities collude against him.
In our second system, the authorities are allowed to collaborate to achieve shorter keys and parameters, enhancing the efficiency of encryption and decryption.
We prove our systems secure under algebraic assumptions on the bilinear groups: the bilinear Diffie-Hellmann assumption and an original variation of the former.

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

1 Introduction

The key feature that makes the Cloud so attracting nowadays is the great accessibility it provides: users can access their data through the Internet from anywhere. Unfortunately, at the moment the protection offered for sensitive information is questionable and access control is one of the greatest concerns. Illegal access may come from outsiders or even from insiders without proper clearance. One possible approach for this problem is

to use Attribute-Based Encryption (ABE) that provides cryptographically enhanced access control functionality in encrypted data.

ABE 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]. The use of bilinear groups, in particular the Tate and Weil pairings on elliptic curves [BF01], was the winning strategy that finally allowed to build schemes following the seminal Shamir's idea. Bilinear groups came in nicely when a preliminary version of ABE was invented by Sahai and Waters [SW05] in 2005. Immediately afterwards, Goyal, Pandey, Sahai, and Waters [GPSW06] formulated the two complimentary forms of ABE which are nowadays standard: ciphertext-policy ABE and key-policy ABE. In a ciphertext-policy ABE system, keys are associated with sets of attributes and ciphertexts are associated with access policies. In a KP-ABE system, the situation is reversed: keys are associated with access policies and ciphertexts are associated with sets of attributes. Several developments in efficiency and generalizations have been obtained for key-policy ABE, e.g. [OSW07], [ALDP11], [AHL+12], [HW13]. All the latter key-policy schemes have a proof of security based on the original Diffie-Helmann assumption on bilinear groups or some slight variation (more on this in Section 2). 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. Related to the work we propose in this pare is the construction for multiple authorities (ciphertext-policy ABE) that have been proposed in [Cha07], [CC09] and [LW11].
However, before the present paper no multi-authority KP-ABE scheme has appeared in the literature with a proof of security.

Our construction In this paper we present the first multi authority KP-ABE schemes. In our first system, after the creation of an initial set of common parameters, the authorities may be set up in any moment and without any coordination. A party can simply act as an ABE authority by creating a public parameters and issuing private keys to different users (assigning access policies while doing so). A user can encrypt data under any set of attributes specifying also a set of *trusted* authorities, so the encryptor maintains high control. Also, the system does not require any central authority.

Our scheme has both very short single-authority keys, that compensates the need of multiple keys (one for authority), and also very short ciphertexts. Moreover, the pairing computations in the bilinear group are involved only during the decryption phase, obtaining this way significant advantages in terms of encryption times.

In our second system, the authorities are allowed to collaborate to achieve shorter keys and parameters, enhancing the efficiency of encryption and decryption. Unless they all collude, even if the authorities are collaborating, the existence of just one non-cheating authority guarantees that no illegitimate party (including authorities) has access to the encrypted data.

We prove our first scheme using the classical bilinear Diffie-Helmann assumption, while for the second scheme we use a slightly stronger variation of the same assumption.

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 multi authority KP-ABE scheme. In Section 4 we explain in detail our multi authority KP-ABE scheme and its security is proven under standard, non-interactive assumptions in the selective set model. In Section 5 we explain the collaborative variant and also prove its security. In Section 6 a lower-bound on the complexity in generic bilinear groups is shown. Finally conclusions are drawn in Section 7.

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 reminder of this section $\mathbb{G}_1$ and $\mathbb{G}_2$ are understood.

2.1 Security assumption on prime order bilinear groups

The security assumptions to be presented relate to the difficulty of the following problem: guess the type of an element T given a vector $\vec{y}$. In particular there are two cases: either $T = e(g,g)^\alpha \in \mathbb{G}_2$ where α depends on $\vec{y}$ (in this case $(\vec{y}, T)$ is called *valid* tuple), or T is a random element $R \in \mathbb{G}_2$, unrelated to $\vec{y}$.
An algorithm $\mathcal{B}$ that tries to solve this problem should output 0 in the first case and 1 in the second. This algorithm $\mathcal{B}$ is supposed to have access to a source of entropy that allows to make random choices. Moreover, $\mathcal{B}$ terminates in a time that is polynomial in $\log_2(p)$ where p is the order of the bilinear group. For these reasons $\mathcal{B}$ is called a *probabilistic polynomial-time algorithm*. It may be useful for the reader to think of $\mathcal{B}$ as an attacker. By giving the attacker a polynomial-time algorithm, we are bounding his computational power in a realistic way (if she could use exponential-time algorithms, no system would resist her attacks). On the other hand, we are giving the attacker a source of entropy as input (or a *random tape* in Turing machine terminology), which is essential to mount dangerous attacks.

Definition 2.3 (Advantage of $\mathcal{B}$)**.** *We say that the advantage $Adv_{\mathcal{B}}$ of $\mathcal{B}$ solving the decisional problem in $\mathbb{G}_1$ is ϵ if*

$$\left| Pr\left[\mathcal{B}(\vec{y}, T = e(g,g)^\alpha) = 0\right] - Pr\left[\mathcal{B}(\vec{y}, T = R) = 0\right] \right| \geq \epsilon.$$

In other words, we hope that the enemy has a small advantage. Indeed, usually people build systems in such a way that they can be broken only by an attacker with a large advantage. We provide now a precise definition for the size of the advantage. We recall that $\eta(k)$ is a negligible function in k, that is, for every c and for every γ there exists k_0 such that $|\eta(k)| < \left|\frac{1}{ck^\gamma}\right|$ for all $k > k_0$. The security assumptions hold if no probabilistic polynomial-time algorithm $\mathcal{B}$ has a *non-negligible* advantage in solving the decisional problems. $\mathcal{B}$ has a *negligible* advantage if $Adv_{\mathcal{B}} = \eta(\log_2(p))$.

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 2.4 (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 2.5 (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 2.6 (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 and g be a generator of the bilinear group $\mathbb{G}_1$. Let $b = c^x$. 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 2.7 (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

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

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 2.8 (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 6 we show an adaption of these assumptions to the generic group model and we are able to prove a related security bound.

Lemma 2.9. *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:

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

If we set $b = a^q$, 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 2.10. *BDH x-Power Assumption and BDH x-Roots Assumption imply BDH Assumption. Moreover x-RBDH implies x-PBDH, and so we have:*

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

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 $i \in \{1, \ldots, x\}$. Since BDH assumption is verified for any b and so in particular for $b = c^x$, we have that $\mathcal{S}_{\text{BDH}} \subseteq \mathcal{S}_{\text{x-PBDH}}$. Moreover $\mathcal{S}_{\text{x-PBDH}} \subseteq \mathcal{S}_{\text{q-PBDHE}}$. 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, if we choose c as a primitive element of $\mathbb{Z}_p$, then we have that $\mathbb{Z}_p$ is generated by c^x iff x and $p-1$ are coprime:

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

□

Finally, we have the following lemma.

Lemma 2.11. *q-BDHE implies both x-BDHE and x-PBDH.*

Proof. If we set $b = a^q$ we obtain that $\mathcal{S}_{\text{x-PBDH}} \subseteq \mathcal{S}_{\text{q-BDHE}}$. Moreover if we also set $a = c$ we have that q-BDHE implies x-RBDH. In fact, we have that both g^{c^i} and $g^{ac^{i-1}}$ become g^{a^i} for $i = 1, \ldots, q$, so $\mathcal{S}_{\text{x-RBDH}} \subseteq \mathcal{S}_{\text{q-BDHE}}$. □

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

Theorem 2.12. *The security assumptions above satisfy the following relations:*

$$\begin{array}{rccc} \text{BDH} & \Longleftarrow & \text{q-BDHE} & \Longleftarrow \quad \text{q PBDHE.} \\ {\scriptstyle GCD(x,p-1)=1}\Downarrow\Uparrow & \not\swarrow & \Downarrow & \\ \text{x-PBDH} & \Longleftarrow & \text{x-RBDH.} & \end{array}$$

2.3 Other security assumptions

In this subsection we present for completeness the assumptions used to prove the full security of a scheme via the *dual system encryption* technique as in [LOS+10]. This technique gained wide interest and success because it permits to achieve full security, however it requires a construction in bilinear groups of composite order, that are significantly less efficient. Note that the following assumption are not used in our work, are only introduced to give a more thorough overview on the algebraic assumptions on which the security of the various ABE schemes lay.
At the end of this section, we are going to compare the assumption on bilinear groups of composite order and the ones on bilinear groups of prime order.

Subgroup Decision Assumption (*A*1) Let $(N, \mathbb{G}, \mathbb{G}_T, e)$ be the description of the bilinear group of composite order $N = p_1p_2p_3$. Let $g_{p_1}, g_{p_2}, g_{p_3}$ be generators of the subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ respectively. The Subgroup Decision problem is that given the challenge tuple

$$\left(D = (N, \mathbb{G}, \mathbb{G}_T, e, g_{p_1}, g_{p_3}), Z\right)$$

an algorithm $\mathcal{A}$ must distinguish between $Z = Z_0 = X_1 \in \mathbb{G}_{p_1}$ and $Z = Z_1 = X_1R_1 \in \mathbb{G}_{p_1p_2}$. The advantage of $\mathcal{A}$ is defined as

$$Adv^{A1}_{\mathcal{A}} = |Pr[\mathcal{A}(D, Z_0) = 0] - Pr[\mathcal{A}(D, Z_1) = 0]|$$

where the probability is taken over the random choices of $X_1 \in \mathbb{G}_{p_1}$ and $R_1 \in \mathbb{G}_{p_2}$.

Definition 2.13 (Assumption *A*1)**.** *We say that the assumption A*1 *holds if no polynomial-time algorithm $\mathcal{A}$ has a non-negligible advantage in solving the Subgroup Decision problem.*

General Subgroup Decision Assumption (*A*2) Let $(N, \mathbb{G}, \mathbb{G}_T, e)$ be the description of the bilinear group of composite order $N = p_1p_2p_3$. Let $g_{p_1}, g_{p_2}, g_{p_3}$ be generators of the subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ respectively. The General Subgroup Decision problem is: given the challenge tuple

$$\left(D = (N, \mathbb{G}, \mathbb{G}_T, e, g_{p_1}, g_{p_3}, X_1R_1, R_2Y_1), Z\right)$$

an algorithm $\mathcal{A}$ must distinguish between $Z = Z_0 = X_2Y_2 \in \mathbb{G}_{p_1p_2}$ and $Z = Z_1 = X_2R_3Y_2 \in \mathbb{G}$. The advantage of $\mathcal{A}$ is defined as

$$Adv^{A2}_{\mathcal{A}} = |Pr[\mathcal{A}(D, Z_0) = 0] - Pr[\mathcal{A}(D, Z_1) = 0]|$$

where the probability is taken over the random choices of $X_1, X_2 \in \mathbb{G}_{p_1}$, $R_1, R_2, R_3 \in \mathbb{G}_{p_2}$ and $Y_1, Y_2 \in \mathbb{G}_{p_3}$, and the random bits possibly consumed by $\mathcal{B}$.

Definition 2.14 (Assumption $A2$)**.** *We say that the assumption $A2$ holds if no polynomial-time algorithm $\mathcal{A}$ has a non-negligible advantage in solving the General Subgroup Decision problem.*

Composite Diffie-Hellman Assumption ($A3$**)** Let $(N, \mathbb{G}, \mathbb{G}_T, e)$ be the description of the bilinear group of composite order $N = p_1p_2p_3$. Let $g_{p_1}, g_{p_2}, g_{p_3}$ be generators of the subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ respectively. The Composite Diffie-Hellman problem is: given the challenge tuple

$$\left(D = (N, \mathbb{G}, \mathbb{G}_T, e, g_{p_1}, g_{p_2}, g_{p_3}, g_{p_1}^a R_1, g_{p_1}^b R_2), Z\right) \tag{1}$$

an algorithm $\mathcal{A}$ must distinguish between $Z = Z_0 = e(g_{p_1}, g_{p_1})^{ab}$ and $Z = Z_1 = e(g_{p_1}, g_{p_1})^c$. The advantage of $\mathcal{A}$ is defined as

$$Adv_{\mathcal{A}}^{A1} = Pr[\mathcal{A}(D, Z_0) = 0] - Pr[\mathcal{A}(D, Z_1) = 0] \tag{2}$$

where the probability is taken over the random choices of $a, b, c \in \mathbb{Z}_N$ and $R_1, R_2 \in \mathbb{G}_{p_2}$, and the random bits possibly consumed by $\mathcal{B}$.

Definition 2.15 (Assumption $A3$)**.** *We say that the assumption $A3$ holds if no polynomial-time algorithm $\mathcal{A}$ has a non-negligible advantage in solving the Composite Diffie-Hellman problem.*

Note that, the Assumptions $A1$ and $A2$ do not hold if the group order can be factorized in polynomial time. So the underlying problems are easier than the problem of Section 2.1. Whereas, $A3$ is comparable with the previous assumptions since is based on Discrete Logarithm on Elliptic Curves.

Moreover, we observe that the group operations in prime order bilinear groups are more efficient than those in composite order bilinear groups.

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 $i \in \{1, \dots, l\}$ the i-th row of M is labeled via a function ρ, that associates M_i to the party $\rho(i)$. Considering the vector $\vec{v} = (s, r_2, \dots, r_n) \in \mathbb{Z}_p^n$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_i \in \mathbb{Z}_p$, with $i \in \{2, \dots, n\}$ 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 $I \subseteq \{1, \dots, l\}$ 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

$$\sum_{i \in I} w_i \lambda_i = s \tag{3}$$

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 $(1, 0, \dots, 0)$ 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 First Construction

This section is divided in three parts. We start with definitions of Multi-Authority Key-Policy ABE and of CPA selective security. In the second part we present in detail our first scheme and, finally, we prove the security of this scheme under the classical BDH assumption in the selective set model.

A security parameter will be used to determine the size of the bilinear group used in the construction, this parameter represents the order of complexity of the assumption that provides the security of the scheme. Namely, first the complexity is chosen thus fixing the security parameter, then this value is used to compute the order that the bilinear group must have in order to guarantee the desired complexity, and finally a suitable group is picked and used.

4.1 Multi Authority KP-ABE Structure and Security

In this scheme, after the common universe of attributes and bilinear group are agreed, the authorities set up independently their master key and public parameters. The master key is subsequently used to generate the private keys requested by users. Users ask to an authority for keys that embed a specific access structure, and the authority issues the key only if it judges that the access structure suits the user that requested it. Equivalently an authority evaluates a user that requests a key and assigning an access structure, and gives to the user a key that embeds it. When someone wants to encrypt, it chooses a set of attributes that describes the message (and thus determines which access structures may read it) and a set of trusted authority. The ciphertext is computed using the public parameters of the chosen authorities, and may be decrypted only using a valid key for each of these authorities. A key with embedded access structure $\mathbb{A}$ may be used to decrypt a ciphertext that specifies a set of attributes S if and only if $S \in \mathbb{A}$, that is the structure considers the set authorized. 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, A a set of authorities and $\mathbb{A}$ an access structure on a universe of attributes U.

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

$\mathsf{Setup}(U, g, \mathbb{G}_1) \to (\mathsf{PK}_\mathbf{k}, \mathsf{MK}_\mathbf{k})$. *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 and a master key MK_k for that authority.*

$\mathsf{KeyGen}_k(\mathsf{MK}_k, (M_k, \rho_k)) \to \mathsf{SK}_k$. *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_k, ρ_k). It outputs a decryption key SK_k for that access structure.*

$\mathsf{Encrypt}(m, S, \{\mathsf{PK}_k\}_{k\in A}) \to \mathsf{CT}$. *The encryption algorithm takes as input the public parameters PK_k of every authority of the set $A \subseteq X$ chosen, 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 and the set of authorities A.*

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

The previous scheme is secure under the classical BDH assumption in the selective set model. The security of the scheme is based on a security games. This kind of games defines an adversary that tries to violate the scheme and describes what it can and cannot do, formalizing also what does *break the scheme* mean. In a security game the *challenger* runs the scheme interacting with an *adversary* that tries to break it. The adversary knows the algorithms and is allowed to perform any computation it wishes with the informations it has access to (such as public parameters and the keys, ciphertexts or other elements it may obtain from the challenger). There is of course the limitation that such computations have to be performed in polynomial time, since the focus is on computationally-bounded realistic adversaries.

In the game to be presented the security is defined in terms of chosen-ciphertext indistinguishability. This means that the adversary should have no advantage in guessing which message has been encrypted of the two she chose and gave to the challenger. Since the adversary is probabilistic it may be lucky with a wild guess on a single shot, but the security states that when the game is repeated the chances of the adversary decrease more and more, and the distribution of the random variable that models its guesses converges towards a uniform distribution. That is, however long she tries, the adversary does not do better than a coin-flip.

The security game is formally defined as follows.

Let $\mathcal{E}$ = (Setup, Encrypt, KeyGen, Decrypt) be a MA-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 MA-KP-ABE experiment $\mathsf{MA\text{-}KP\text{-}ABE\text{-}Exp}_{\mathcal{A},\mathcal{E}}(\lambda, U)$ for an adversary $\mathcal{A}$, parameter λ and attribute universe U:

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

Setup. The challenger runs the Setup algorithm, initializes the authorities and gives to the adversary the public parameters.

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 for the set of authorities A. The ciphertext is passed to the adversary.

Phase II. Phase `I` is repeated.

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

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

The schemes plans a set X of independent authorities, each with their own parameters, and it sets up an encryption algorithm that lets the encryptor choose a set $A \subseteq X$ of authorities, and combines the public parameters of these in such a way that an authorized key for each authority in A is required to successfully decrypt.
Our scheme consists of three randomized algorithms (Setup, KeyGen, Encrypt) plus the decryption Decrypt. The techniques used are inspired from the scheme of Goyal et al. in [GPSW06]. 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.

$\mathsf{Setup}(U, g, \mathbb{G}_1) \rightarrow (\mathsf{PK}_k, \mathsf{MK}_k)$. 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:

$$\mathsf{PK}_k = (e(g,g)^{\alpha_k}, \{g^{z_{k,i}}\}_{i \in U}) \qquad \mathsf{MK}_k = (\alpha_k, \{z_{k,i}\}_{i \in U}\})$$

$\mathsf{KeyGen}_k(\mathsf{MK}_k, (M_k, \rho_k)) \to \mathsf{SK}_k$. The key generation algorithm for the authority k takes as input the master secret key MK_k and an LSSS access structure (M_k, ρ_k), where M_k is an $l \times n$ matrix on $\mathbb{Z}_p$ and ρ_k is a function which associates rows of M_k 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:

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

$\mathsf{Encrypt}(m, S, \{\mathsf{PK}_k\}_{k \in A}) \to \mathsf{CT}$. 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:

$$\mathsf{CT} = \left(S, C' = m \cdot \left(\prod_{k \in A} e(g,g)^{\alpha_k} \right)^s, \{C_{k,i} = (g^{z_{k,i}})^s\}_{k \in A,\ i \in S} \right)$$

$\mathsf{Decrypt}(\mathsf{CT}, \{\mathsf{SK}_k\}_{k \in A}) \to m'$. The input is a ciphertext for a set of attributes S and a set of authorities A and an authorized key for every authority cited by the ciphertext. Let (M_k, ρ_k) be the LSSS associated to the key k, and suppose that S is authorized for each $k \in A$. The algorithm for each $k \in A$ finds $w_{k,i} \in \mathbb{Z}_p, i \in I_k$ such that

$$\sum_{i \in I_k} \lambda_{k,i} w_{k,i} = \alpha_k \tag{4}$$

for appropriate subsets $I_k \subseteq S$ and then proceeds to reconstruct the original message computing:

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

Where $\stackrel{*}{=}$ follows from property (4).

4.3 Security

The scheme is proved secure under the BDH assumption (Definition 2.4) in a selective set security game in which 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, then a simulator can be constructed to play the Decisional BDH 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 a simulator $\mathcal{B}$ can be built that can play the Decisional BDH game with advantage $\epsilon/2$. The simulation proceeds as follows.

Init The simulator takes in a BDH challenge g, g^a, g^b, g^s, T. The adversary gives the algorithm the challenge access structure S.

Setup The simulator chooses random $r_k \in \mathbb{Z}_p$ for $k \in A \setminus \{k_0\}$ and implicitly sets $\alpha_k = -r_k b$ for $k \in A \setminus \{k_0\}$ and $\alpha_{k_0} = ab + b\sum_{k\in A\setminus\{k_0\}} r_k$ by computing:

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

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

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

Then it can publish the public parameters computing the remaining values as:

$$g^{z_{k,i}} = \begin{cases} g^{z'_{k,i}} & \text{if } i \in S \\ (g^b)^{z'_{k,i}} & \text{if } i \notin S \end{cases}$$

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

$$M_i \vec{y} = 0 \qquad \forall i \text{ such that } \rho(i) \in S \tag{5}$$

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

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

Note that $\lambda_{k_0,i} = \sum_{j=1}^{n} M_{i,j} u_j$ where $u_j = b(v_j + (a + \sum_{k \in A \setminus \{k_0\}} r_k - v_1) y_j)$ thus $u_1 = b(v_1 + (a + \sum_{k \in A \setminus \{k_0\}} r_k - v_1)1) = ab + b \sum_{k \in A \setminus \{k_0\}} r_k = \alpha_{k_0}$ so shares are valid. Note also that from (5) 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

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

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

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

Remembering that in this case $z_{k_0,\rho(i)} := b z'_{k_0,\rho(i)}$. Finally for the queries to the other authorities $k \in A \setminus \{k_0\}$, 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

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

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

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

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 Collaborative Authorities Variant

This section is also 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 2.4) to prove the security of this scheme under in the selective set model.

5.1 Collaborative Multi Authority KP-ABE Structure and Security

In this scheme, the authorities set up independently their master key, but then collaborate to create a common public key. They collaborate also to generate the private keys requested by users, in fact, after one authority agrees to give a key for a specific access structure to a user, this key is *validated* by every other authority and only then it is ready and passed to the user. Users need a key for every authority, but once they obtained all the pieces they may unite them and thus store and use them as a single key. When someone wants to encrypt, it 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, and may be decrypted only using a conglomerate valid key (equivalently a valid key for every authority). 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 5.1 (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:*

$\mathsf{Setup}(U, g, \mathbb{G}_1) \to (\mathsf{PK}_k, \mathsf{MK}_k)$. *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 and a master key MK_k for that authority.*

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

$\mathsf{KeyGen}_k(\mathsf{MK}_k, (M, \rho)) \to \mathsf{SK}_k^{(1)}$. *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 the initial step of the construction of a decryption key $\mathsf{SK}_k^{(0)}$ for that access structure.*

$\mathsf{CollKeygen}(\mathsf{MK}_k, \mathsf{SK}_{\bar{k}}^{(h)}) \to \mathsf{SK}_{\bar{\mathbf{k}}}^{(h+\mathbf{1})}$. *The collaborative part of setup asks the authority $k \in X$ to add their part to the final decryption key of authority $\bar{k} \in X, \bar{k} \neq k$. It takes as input the master key MK_k for that authority and the i-th step of construction of the decryption key $\mathsf{SK}_k^{(h)}$. It outputs the next step of construction of the decryption key $\mathsf{SK}_k^{(h+1)}$. When $h = x$ then $\mathsf{SK}_k^{(x)} = \mathsf{SK}_k$ i.e. the key is completed since every authority has contributed.*

$\mathsf{Encrypt}(m, S, \mathsf{PK}) \to \mathsf{CT}$. *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. However the authorities collaborate every time they are requested to, so if the authority answers to the query the final key will be passed to the adversary.

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.

Definition 5.2 (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:*

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

5.2 The Scheme

This variant 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.

$\mathsf{Setup}(U, g, \mathbb{G}_1) \rightarrow (\mathsf{PK}_k, \mathsf{MK}_k)$. 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:

$$\mathsf{PK}_k = (Y_k = e(g,g)^{\alpha_k}, \{T_{k,i} = g^{z_{k,i}}\}_{i \in U}) \qquad \mathsf{MK}_k = (\alpha_k, \{z_{k,i}\}_{i \in U}\})$$

$\mathsf{CollSetup}(\mathsf{MK}_k, \mathsf{PK}_k, \mathsf{PK}^{(h)}) \to \mathsf{PK}^{(h+1)}$. 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 $\mathsf{PK}^{(1)} = \mathsf{PK}_k$
- if $h > 0$ then $\mathsf{PK}^{(h)} = (Y^{(h)}, \{T_i^{(h)}\}_{i \in U})$

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

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

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

$\mathsf{KeyGen}_k(\mathsf{MK}_k, (M, \rho)) \to \mathsf{SK}_k^{(1)}$. The key generation algorithm for the authority k takes as input the master secret key MK_k 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 first step of the private key $\mathsf{SK}_k^{(1)}$ is:

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

$\mathsf{CollKeygen}(\mathsf{MK}_k, \mathsf{SK}_{\bar{k}}^{(h)}) \to \mathsf{SK}_{\bar{\mathbf{k}}}^{(h+1)}$. The collaborative construction of the public key for the authority $k \neq \bar{k}$ takes the previous step $\mathsf{SK}_{\bar{k}}^{(h)} = \left\{K_{\bar{k},i}^{(h)}\right\}_{1 \le i \le l}$ and then computes

$$K_{\bar{k},i}^{(h+1)} = \left(K_{\bar{k},i}^{(h)}\right)^{\frac{1}{z_{k,i}}} \quad \forall 1 \le i \le l$$

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

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

$\mathsf{Encrypt}(m, S, \mathsf{PK}) \to \mathsf{CT}$. 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:

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

$\mathsf{Decrypt}(\mathsf{CT}, \{\mathsf{SK}_k\}_{k\in X}) \to m'$. 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

$$\sum_{i\in I} \lambda_{k,i} w_i = \alpha_k \qquad \forall k \in X \tag{6}$$

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 (6).

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.

5.3 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 5.1. 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 5.3. *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 challenge access structure S.

Setup 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:

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

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

$$M_i \vec{y} = 0 \qquad \forall i \text{ such that } \rho(i) \in S \tag{7}$$

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 (7) 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 $z_{\rho(i)} = bz'_{\rho(i)}$. 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

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

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

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

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 $\frac{\epsilon}{2}$. □

6 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.

6.1 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 6.1 (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 $d_P = \max\{d_f : f \in P\}$.

6.2 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 6.2 (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 6.3 (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* $\Omega(\frac{p}{d} - s)$.

6.3 Using Corollary 6.3

We claim that the assumptions presented in Section 2 follow from Corollary 6.3 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 $i \in \{1, \dots, 2q\} \setminus \{q+1\}, Q = \{1\}, f = x^q y$.
- 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 $P = \{1, w^i, yw^{i-1}, z\}$ 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:

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

Since every monomial in which both y and z appear has degree strictly lesser than $x+2$ 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 6.3 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].

7 Related Works and Final Comments

Our scheme gives a solution addressing the problem of faith in the authority, specifically the concerns arisen by key escrow and clearance check. Key escrow is a setting in which a party (in this case the authority) may obtain access to private keys and thus it can decrypt any ciphertext. Normally the users have faith in the authority and assume that it will not abuse of its powers. The problem arises when the application does not plan a predominant role and there are trust issues selecting any third party that should manage the keys. In this situation the authority is seen as *honest but curious*, in the sense that it will provide correct keys to users (then it is not malicious) but will also try to access to data beyond its competence. It is clear that as long as a single authority will be the unique responsible to issue the keys, there is no way to prevent key escrow. Thus the need for multi-authority schemes arises.

The second 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 not only to give correct keys and to not violate the privacy, but also to perform correct checks of the users' clearances and to assign correct access structures accordingly. So alongside to *not malicious* and *not curious* the authority has also to be *not breached*, in the sense that the keys of a user must embed access structures that are coherent with its actual clearance, and no

one has access to keys beyond their pertinence. In this case, to add multiple authorities to the scheme gives to the encryptor the opportunity to request more guarantees about the legitimacy of the decryptor's clearance. In fact, each authority checks the users independently, so the idea is to request that the decryption proceeds successfully only when a key for each authority of a given set A is used. This means that the identity of the user has been checked by every selected authority, and the choice of these by the encryptor models the trust that he has in them. Note that if these authorities set up their parameters independently and during encryption these parameters are bound together indissolubly, 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 first scheme proposed has very short single-keys (just one element per row of the access matrix) that compensates the need of multiple single-keys (one for cited authority) in the decryption. Ciphertexts are also very short (the number of elements is linear in the number of authorities times the number of attributes under which it has been encrypted) thus the scheme is efficient under this aspect. Moreover, there are *no* pairing computations involved during encryption and this means significant advantages in terms of encryption times. Decryption time is not constant in the number of pairings (e.g. as in the scheme presented in [HW13] or the one in [Wat11]) but requires $\cdot \sum_{k \in A} l_k$ pairings where A is the set of authorities involved in encryption and l_k is the number of rows of the access matrix of the key given by authority k. Although many ABE schemes do not have a constant number of pairings in decryption, it is evident that decryption slows down linearly with the number of authorities required by the encryptor, so for an efficient scheme only few authorities have to be requested. On the contrary the assumption under which the scheme is proven secure (BDH) is weaker than that of the schemes with fast decryption (q-BDHE), so it is not unreasonable that a variant of the scheme will be developed that achieves faster decryption with stronger security assumptions.

The second construction manages to achieve even more efficiency in the number of parameters needed, since the collaboration between the authorities 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, and it is easy to plan the key generation in such a way that all the pieces are generated and validated together requiring a single passage among the authorities, so the collapsed key is obtained directly. 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 setup and key generation (the phases that have to be done fewer times when the scheme runs), encryption, decryption and key-storage are greatly improved.

Taking a more historical perspective, the problem of multi-authority ABE is not novel and a few solution have been proposed for ciphertext-policy ABE. The problem of building ABE systems with multiple authorities was proposed by Sahai and Waters and first considered by Chase [Cha07] and Chase and Chow [CC09]. In those works the main goal is to relieve the central authority of the burden of generating key material for every user and add resiliency to the system. Multiple authorities manage the attributes, so that each has less work and the whole system does not get stuck if one is down. The most recent and interesting result are in [LW11], where Lewko and Waters propose a scheme where no central authority or coordination between the authorities, each controlling disjoint sets of attributes, is needed. They used composite bilinear groups and via Dual System Encryption (introduced by Waters [Wat09] with techniques developed with Lewko [LW10]) proved their scheme fully secure following the example of Lewko et al. [LOS+10]. They allow the adversary to statically corrupt authorities choosing also their master key. Note however that they did not specifically address key escrow but distributed workload.

Our results of this article retain relevance since they address a different setting. In fact, with this expansions the differences in the situations of ciphertext-policy ABE and KP-ABE model become more distinct. For example a situation that suits the scheme proposed here, but not the one of Lewko and Waters is the following. Consider company branches dislocated on various parts of the world, each checking its personnel and giving to each an access policy (thus act as authorities). This scheme allows encryptions that may be decrypted by the manger of the branch (simply use only one authority as in classic ABE) but also more secure encryptions that require the identity of the decryptor to be guaranteed by more centers, basing the requirements on which branches are still secure and/or where a user may actually authenticate itself.

Moreover, 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, 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. These considerations demonstrate that our constructions in prime groups retain validity and interest, considering also that the proofs are under quite weak assumptions and in the standard model.

Remark 7.1 (Security Assumptions). In the proof of the second scheme (Proof of Theorem 5.3) it is supposed that only the final public key is actually public, that is, the parameters of the authorities 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 2.7) 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^i} for $i \in \{1, \ldots, x\}$ to correctly simulate the collaboration steps during setup, while for the step that creates the keys further values are needed: g^{ac^i} for $i \in \{1, \ldots, x-1\}$ so instead of the x-PBDH, the stronger x-RBDH is needed.

Remark 7.2 (Security Definitions). Both original schemes have 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. The definition of security may however 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* (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. For the first scheme the target is the set of attributes S, the set of authorities A 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. The only restriction on A is that it must contain k_0. For the second scheme the target is the set of attributes S and the honest authority k_0.

Also 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.

In both schemes 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 results in this paper have partially appeared in the first author's Master's thesis. Therefore, he would like to thank his supervisors (the other two authors). 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.

[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.

[LOS+10] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, *Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption*, Proc. of EUROCRYPT 10, LNCS, vol. 7881, 2010, pp. 62–91.

[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.