Collaborative Multi-Authority Key-Policy Attribute-Based Encryption for Shorter Keys and Parameters

Riccardo Longo*, Chiara Marcolla**, Massimiliano Sala***

Abstract. Architectures relying on a single central authority often offer a great efficiency, but suffer of resiliency problems and are quite vulnerable to attacks. In our proposal, a Multiple-Authorities Key-Policy Attribute-Based Encryption scheme is constructed including a collaboration phase between the authorities, in order to achieve shorter keys and parameters, thus enhancing the efficiency of encryption and decryption.
We prove our system secure under a variation of the bilinear Diffie-Hellman assumption, providing a lower bound on its complexity.

Keywords ABE, KP-ABE, Multi-Authority, Bilinear Groups, Diffie-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 external attackers, or even from insiders that abuse their clearance. In fact sometimes legitimate users try to gain access to someone else's data, and they should not be allowed to do that. One possible approach to this problem is to use Attribute-Based Encryption (ABE), a tool that provides cryptographically enhanced access control functionality in encrypted data.

ABE developed from Identity Based Encryption, a scheme proposed by Shamir [23] in 1985 with the first constructions obtained in 2001 by Boneh and Franklin [6] and Cocks [9]. In 2005 Sahai and Waters [22] proposed the first schemes of Attributed Based Encryption and in a consecutive work, Goyal, Pandey, Sahai, and Waters [11] 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

* riccardolongomath@gmail.com, *Department of Mathematics, University of Trento*
** chiara.marcolla@gmail.com, *Department of Mathematics, University of Turin*
*** maxsalacodes@gmail.com, *Department of Mathematics, University of Trento*

policies and ciphertexts are associated with sets of attributes. Several developments and generalizations have been obtained for KP-ABE [20, 2, 1, 13]. These schemes are constructed on bilinear groups (usually implemented through the Tate [24, 25] and Weil [28] 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. [4] 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 [27] 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. [14, 10] which exploit 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 ([12]).

The first multi authority KP-ABE scheme with expressive policies (i.e. that allow any monotone access structure as policy) was presented in [19]. 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 [7, 8] and [16]. In [8], that is a improvement of [7], the authors construct a simple-threshold schemes in the case where attributes are divided in disjoint sets, each controlled by a different authority. Comparing our scheme with those proposed by Chase et al ([7, 8]), 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 unauthorized accesses more effectively, and prevent the ability of authorities to intrude into users' privacy. Whereas, in [16] Lewko and Waters propose a scheme where there is no need for a central authority or for coordination between the authorities, each one controlling disjoint sets of attributes. Finally in [21] Rouselakis and Waters propose a multi authority CP-ABE with a large universe of attributes, that is any string can be used and they do not have to be enumerated beforehand. Their proof is under the random oracle model (that provides weaker security) and guarantees only static (compared to selective) security, however they achieve good efficiency.

Our construction The scheme that we propose in this paper evolves from the scheme presented in [19] exploiting a collaboration between authorities to improve the efficiency. It is a multi authority KP-ABE scheme in which the authorities collaborate to achieve shorter keys and parameters, thus 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 they collaborate together to create:

- a common *public key* used 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* that describes his clearance. 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 every key, he can merge them and obtain a single compact key. In this way Bob may store and use 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 [19]) 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 [19] 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 scheme, 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, they will provide correct keys to users but will also try to access to data beyond their competence. Obviously, if there is a single authority which is the unique responsible of issuing the keys, there is no way to prevent this kind of 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 (note however that CP-ABE is equally sensitive in this passage because the authority has to assign correct

attributes to users). 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 in our scheme, since these authorities set up their parameters independently and during encryption these parameters must be bound together irrevocably, then no authority can single-handedly decrypt any ciphertext and thus key escrow is removed. In our construction and security proof we never consider malicious authorities, that is authorities might become compromised and therefore attackers may gain access to master keys and distribute keys improperly but the set up of the parameters is beyond the control of any attacker.

So our KP-ABE scheme guarantees protection against both breaches and curiosity.

The scheme is proved secure under a slightly stronger variation of the classical BDH assumption (Definition 3).

Organization This paper is organized as follows. In Section 2 we present bilinear groups, alongside the original Decisional Bilinear Diffie Hellman assumption and its variation that we will use in our security proof. 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 of the security assumption 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 and its variation that we will use to prove our scheme secure.

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

Definition 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 $\mathbb{G}_1$, $e(g, g) \neq 1_{\mathbb{G}_2}$.*

Definition 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 [6] by Boneh and Franklin and then widely used in a variety of proofs, including the one of the first ABE in [11]. 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 $\mathcal{B}(A = g^a, B = g^b, S = g^s, T) \rightarrow \{0, 1\}$ to efficiently distinguish between the tuples $(A, B, S, e(g, g)^{abs})$ 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 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.*

Augment Decisional Bilinear Diffie-Hellman Assumption This assumption, introduced by Liang et al. in [17], is a variant of the basic BDH in which the attacker has an advantage due to one more element at their disposal. We formally define it as follows.

Let $a, b, s, z \in \mathbb{Z}_p$ be exponents chosen at random, let g be a generator of the bilinear group $\mathbb{G}_1$, and let $b \neq 0$. The augment decisional bilinear Diffie-Hellman (ABDH) problem consists in constructing an efficient algorithm $\mathcal{B}(A = g^a, B = g^b, C = g^{\frac{1}{b}}, S = g^s, Z) \rightarrow \{0, 1\}$ to efficiently distinguish between the tuples $(A, B, C, S, e(g, g)^{abs})$ and $(A, B, C, S, e(g, g)^z)$. The advantage of $\mathcal{B}$ is defined, following the standard convention, as:

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

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

In Section 5 we show an adaptation of these assumption to the generic group model and we are able to prove a related security bound.

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 encrypted data, listing the sets of attributes that have clearance.

Definition 5 (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 (that is k attributes are requested out of a set of n specific attributes), then the access structure is *monotonic*.

Definition 6 (Monotonic Access Structure). *An access structure $\mathbb{A}$ is said monotonic if given $S_0 \subseteq S_1 \subseteq U$ it holds*

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

An interesting property is that monotonic access structures 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 [3]).

Definition 7 (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 $i \in \{1, \dots, l\}$ M_i (the i-th row of M) is labeled via a function ρ, that associates it to the party $\rho(i)$. Considering the vector $\boldsymbol{v} = (s, r_2, \dots, r_n) \in \mathbb{Z}_p^n$, 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 $M\boldsymbol{v}$ is the vector of l shares of the secret s according to Π. The share $(M\boldsymbol{v})_i = M_i\boldsymbol{v}$ belongs to party $\rho(i)$.*

It is shown in [3] 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

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

Furthermore, it is shown in [3] 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 $\boldsymbol{u}$ such that $u_1 = c$ and

$$Mi \cdot \boldsymbol{w} = 0 \qquad \forall i \in I \tag{2}$$

In the first ABE schemes the access formulas are typically described in terms of access trees. The appendix of [16] is suggested for a discussion of how to perform a conversion from access trees to LSSS.

See [11], [3] and [18] for more details about LSSS and access structures.

4 Our Construction

This section is divided in three parts. First Collaborative Multi-Authority Key-Policy ABE and its CPA selective security are defined. In the second part the scheme is presented in detail and, finally, a variant of the BDH assumption (Definition 3) is used to prove the security of this scheme 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 keys 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 access policy to assign to the user, or equivalently the user should ask for the same policy to every authority. Note however that it is very reasonable that the same access policy is assigned since it is strictly related to the specific user. Moreover note that even if the policy of a user might contain sensitive data, it might be safely shared between authorities since they are entitled to access this kind of information anyway, and there is no randomness shared between authorities in doing so.

To encrypt, a user chooses a set of attributes that describes the message (and thus determines which access structures give access to it). The ciphertext is computed using the public key generated by the authorities in concert. When someone wants to decrypt, they need a key for every authority and once they obtain all the pieces they can merge 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 8 (Collaborative Multi-authority KP-ABE). *A collaborative 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 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 h-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 $\mathsf{AP}^{(h+1)}$ (at the first step, i.e. $h = 0$, they are simply initialized with the parameters of the first authority). When $h = x = |X|$ then $\mathsf{PK}^{(x)} = \mathsf{PK}$ and $\mathsf{AP}^{(x)} = \mathsf{AP}$, i.e. the public and authority parameters are completed once 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, the commonly constructed authority parameters AP, 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 $k \in X$. Let $\mathbb{A}$ be the access structure of each key SK_k. It outputs the message m' if and only if $S \in \mathbb{A}$.*

The security game is defined as follows.

Definition 9 (CMA-KP-ABE Security Game). *Take a CMA-KP-ABE scheme $\mathcal{E} = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{KeyGen}, \mathsf{Decrypt})$ 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 $\mathcal{A}$, security parameter λ and attribute universe U:*

Init. *The adversary declares the set of attributes S that they wish to be challenged upon. Moreover they select 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 individual public key and the authority parameters of every authority, alongside all the master keys of the non-honest authorities and every collaboration step.*

Phase I. *The adversary issues queries for private keys generated by k_0, however the access structures $\mathbb{A}$ relative to these keys can not authorize the target set, that is $S \notin \mathbb{A}$.*

Challenge. *The adversary submits two equal length messages m_0 and m_1. The challenger flips a random coin $b \in \{0, 1\}$, 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 10 (CMA-KP-ABE Selective Security). *The CMA-KP-ABE scheme* $\mathcal{E}$ *is CPA selective secure (or secure against chosen-plaintext attacks) for attribute universe* U *if for every probabilistic polynomial-time adversary* $\mathcal{A}$*, 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.

The scheme consists of three randomized algorithms (Setup, KeyGen, Encrypt) plus the collaborative step CollSetup and decryption 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.

$\mathsf{Setup}(U, g, \mathbb{G}_1) \rightarrow (\mathsf{PK}_k, \mathsf{MK}_k, \mathsf{AP}_k)$. 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, the master key MK_k, and the authority parameters AP_k are:

$$\mathsf{PK}_k = \left(Y_k = e(g,g)^{\alpha_k}, \{T_{k,i} = g^{z_{k,i}}\}_{i \in U}\right) \tag{3}$$

$$\mathsf{MK}_k = (\alpha_k, \{z_{k,i}\}_{i \in U}\}) \tag{4}$$

$$\mathsf{AP}_k = \left(\left\{V_{k,i} = g^{\frac{1}{z_{k,i}}}\right\}_{i \in U}\right) \tag{5}$$

$\mathsf{CollSetup}(\mathsf{MK}_k, \mathsf{PK}_k, \mathsf{AP}_k, \mathsf{PK}^{(h)}, \mathsf{AP}^{(h)}) \rightarrow (\mathsf{PK}^{(h+1)}, \mathsf{AP}^{(h+1)})$. 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

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

Then it is easy to see that when the construction is complete (i.e. every authority has contributed) the public key is:

$$\mathsf{PK}^{(x)} = \mathsf{PK} = \left(Y = e(g,g)^{\sum_{k\in X}\alpha_k}, \left\{T_i = g^{\prod_{k\in X} z_{k,i}}\right\}_{i\in U}\right) \tag{6}$$

$$\mathsf{AP}^{(x)} = \mathsf{AP} = \left(\left\{V_i = g^{\frac{1}{\prod_{k\in X} z_{k,i}}}\right\}_{i\in U}\right) \tag{7}$$

$\mathsf{KeyGen}_k(\mathsf{MK}_k, \mathsf{AP}, (M,\rho)) \to \mathsf{SK}_k$. The key generation algorithm for the authority k takes as input the master key MK_k, the authority parameters AP 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 $\boldsymbol{v}_k \in \mathbb{Z}_p^n$ such that $v_{k,1} = \alpha_k$. Then computes the shares $\lambda_{k,i} = M_i \boldsymbol{v}_k$ for $1 \le i \le l$ where M_i is the i-th row of M. Then the private key SK_k is:

$$\mathsf{SK}_k = \left\{K_{k,i} = V_{\rho(i)}^{\lambda_{k,i}} = g^{\frac{\lambda_{k,i}}{\prod_{k\in X} z_{k,\rho(i)}}}\right\}_{1\le i\le l} \tag{8}$$

$\mathsf{Encrypt}(m, S, \mathsf{PK}) \to \mathsf{CT}$. The encryption algorithm takes as input the public key PK, 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:

$$\mathsf{CT} = \left(S, C' = m \cdot (Y)^s, \{C_i = (T_i)^s\}_{i\in S}\right) \tag{9}$$

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

$$\sum_{i\in I} \lambda_{k,i} w_i = \alpha_k \qquad \forall k \in X \tag{10}$$

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 (10).

Note that once the user has obtained the keys from every authority they can be multiplied all together, so that only $\mathsf{SK} = \{K_i = \prod_{k \in X} K_{k,i}\}_{1 \le i \le l}$ has to be stored, since this is the only thing needed to perform the decryption. So actually only a key of size l is needed, hence the scheme is very efficient in terms of key-size.

4.3 Security

The scheme is proved secure under the ABDH assumption in the selective set security game described in Definition 9. Recall that every authority but one is supposed curious (or corrupted or breached) and then the attacker has access to their master keys and so is able to issue even keys that have enough clearance for the target set of attributes, while the honest authority 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 1. *If an adversary can break the scheme, then a simulator can be constructed to play the Decisional ABDH 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 ABDH game with advantage $\epsilon/2$. The simulation proceeds as follows.

Init The simulator takes in a ABDH challenge $\boldsymbol{y} = (g, g^a, g^b, g^{\frac{1}{b}}, g^s), Z$. The adversary gives to the simulator the challenged set of attributes S, and chooses the honest authority $k_0 \in X$, where X is the set of authorities.

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

$$Y_{k_0} = e(g,g)^{\alpha_{k_0}} = e(g^a, g^b) e(g,g)^{\sum_{k \in X \setminus \{k_0\}} r_k} \tag{11}$$

$$Y_k = e(g,g)^{\alpha_k} = e(g,g)^{-r_k} \qquad \forall k \in X \setminus \{k_0\} \tag{12}$$

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

$$T_{k_0,i} = \begin{cases} g^{z'_{k_0,i}} & \text{if } i \in S \\ (g^b)^{z'_{k_0,i}} & \text{if } i \notin S \end{cases} \tag{13}$$

$$T_{k,i} = g^{z'_{k,i}} \qquad \text{for } k \neq k_0 \tag{14}$$

And the authority parameters as:

$$V_{k_0,i} = \begin{cases} g^{\frac{1}{z'_{k_0,i}}} & \text{if } i \in S \\ (g^{\frac{1}{b}})^{\frac{1}{z'_{k_0,i}}} & \text{if } i \notin S \end{cases} \tag{15}$$

$$V_{k,i} = g^{\frac{1}{z'_{k,i}}} \qquad \text{for } k \neq k_0 \tag{16}$$

The simulator can now pass the master keys of the non-honest authorities, and every public and authority parameter to the adversary.

Then it proceeds to simulate the collaborative steps of the scheme. To formalize this let us introduce an ordering function $\psi : X \to \{j : 1 \leq j \leq |X|\}$ that simply specifies in which order the authorities collaborate (that is if $\psi(k_0) = 1$ then the honest authority begins the collaboration). Then the collaborative steps of the public parameters (for $i \in U$) are computed as:

$$Y^{(h)} = \begin{cases} e(g^a, g^b)e(g,g)^{\prod_{k\in X:\psi(k)>h} r_k} & \text{if } \psi(k_0) \leq h \\ e(g,g)^{-\prod_{k\in X:\psi(k)\leq h} r_k} & \text{otherwise} \end{cases} \tag{17}$$

$$T_i^{(h)} = \begin{cases} (g^b)^{\prod_{k\in X:\psi(k)\leq h} z'_{k,i}} & \text{if } i \notin S \wedge \psi(k_0) \leq h \\ g^{\prod_{k\in X:\psi(k)\leq h} z'_{k,i}} & \text{otherwise} \end{cases} \tag{18}$$

$$V_i^{(h)} = \begin{cases} (g^b)^{\frac{1}{\prod_{k\in X:\psi(k)\leq h} z'_{k,i}}} & \text{if } i \notin S \wedge \psi(k_0) \leq h \\ g^{\frac{1}{\prod_{k\in X:\psi(k)\leq h} z'_{k,i}}} & \text{otherwise} \end{cases} \tag{19}$$

Using the previously introduced notation $z'_i := \prod_{k\in X} z'_{k,i}$, for $h = |X|$ we have the complete public key and authority parameters:

$$Y = e(g,g)^{ab} \tag{20}$$

$$T_i = \begin{cases} g^{z'_i} & \text{if } i \in S \\ g^{bz'_i} & \text{if } i \notin S \end{cases} \tag{21}$$

$$V_i = \begin{cases} g^{\frac{1}{z'_i}} & \text{if } i \in S \\ g^{\frac{1}{bz'_i}} & \text{if } i \notin S \end{cases} \tag{22}$$

Phase I In this phase the simulator answers to private key queries made to the honest 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 property (2) of an LSSS it can find a vector $\boldsymbol{u} \in \mathbb{Z}_p^n$ with $u_1 = 1$ such that

$$M_i \boldsymbol{u} = 0 \qquad \forall i \text{ such that } \rho(i) \in S \tag{23}$$

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

$$\lambda_{k_0,i} = \sum_{j=1}^{n} M_{i,j}(bv_j + (ab + \sum_{k\in X\setminus\{k_0\}} r_k - bv_1)u_j) \tag{24}$$

Note that $\lambda_{k_0,i} = \sum_{j=1}^{n} M_{i,j} w_j$ where $w_j = bv_j + (ab + \sum_{k\in X\setminus\{k_0\}} r_k - bv_1)u_j$ thus $w_1 = bv_1 + (ab + \sum_{k\in X\setminus\{k_0\}} r_k - bv_1)1 = ab + \sum_{k\in X\setminus\{k_0\}} r_k = \alpha_{k_0}$ so the shares are valid.

Note also that from (23) it follows that

$$\begin{aligned}\lambda_{k_0,i} &= \sum_{j=1}^{n} M_{i,j} b v_j + \sum_{j=1}^{n} M_{i,j} u_j (ab + \sum_{k \in X \setminus \{k_0\}} r_k - b v_1) \\ &= b \sum_{j=1}^{n} M_{i,j} v_j \quad \forall i \text{ such that } \rho(i) \in S\end{aligned}$$

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

$$\begin{aligned}K_{k_0,i} &= g^{\frac{\sum_{j=1}^{n} M_{i,j}(v_j - v_1 u_j)}{z'_{\rho(i)}}} (g^a)^{\frac{\sum_{j=1}^{n} M_{i,j} u_j}{z'_{\rho(i)}}} (g^{\frac{1}{b}})^{\frac{\sum_{j=1}^{n} M_{i,j} u_j \sum_{k \in X \setminus \{k_0\}} r_k}{z'_{\rho(i)}}} \\ &= g^{\frac{b \sum_{j=1}^{n} M_{i,j}(v_j - v_1 u_j)}{b z'_{\rho(i)}}} g^{\frac{ab \sum_{j=1}^{n} M_{i,j} u_j}{b z'_{\rho(i)}}} g^{\frac{\sum_{j=1}^{n} M_{i,j} u_j \sum_{k \in X \setminus \{k_0\}} r_k}{b z'_{\rho(i)}}} \\ &= g^{\frac{\sum_{j=1}^{n} M_{i,j}(b v_j + (ab + \sum_{k \in X \setminus \{k_0\}} r_k - b v_1) u_j)}{b z'_{\rho(i)}}} \\ &= g^{\frac{\lambda_{k_0,i}}{z_{\rho(i)}}}\end{aligned}$$

Where the last equality follows from $z_{\rho(i)} = b z'_{\rho(i)}$.

Note that the adversary has the master keys of the other authorities, so they can create any other private key.

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

$$\begin{aligned}C' &= m_\mu \cdot Z \stackrel{*}{=} m_\mu \cdot e(g,g)^{abs} = m_\mu Y^s \\ C_{k,i} &= (g^s)^{z'_{\rho(i)}} = g^{s z_{\rho(i)}} \qquad i \in S\end{aligned}$$

Where the equality $\stackrel{*}{=}$ holds if and only if the ABDH challenge was a valid tuple (i.e. Z 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 1 to guess that $Z = e(g,g)^{abs}$ if $\mu' = \mu$; otherwise, they output 0 to indicate that they believes Z is a random group element in $\mathbb{G}_2$. In fact when Z is not random the simulator $\mathcal{B}$ gives a perfect simulation so it holds:

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

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

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

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

5 Generic Security of Diffie-Hellman Assumptions

In [5] 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 is 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 11 (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 2 (Theorem A.2 of [5]). *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 1 (Corollary A.3 of [5]). *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 1

We claim that the assumptions presented in Section 2 follow from Corollary 1 giving the sets P, Q that reduces them to the general bilinear Diffie-Hellman problem:

- BDH in $\mathbb{G}_1$: set $P = \{1, y, w, z\}, Q = \{1\}, f = y\ w\ z$.
- ABDH in $\mathbb{G}_1$: set $P = \{1, y, w, \frac{1}{w}, z\}, Q = \{1\}, f = y\ w\ z$.

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

ABDH, the complete list of terms that may be obtained combining any two polynomials of P follows:

$$1, w, \frac{1}{w}, y, yw, \frac{y}{w}, wz, \frac{z}{w}, z, yz$$

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 ywz as result, thus f is independent of P, Q.

Thus applying the Corollary 1 a lower bound on the computational complexity of these problems in the generic bilinear group is obtained.

6 Final Comments

Our construction evolves from the scheme presented in [19] 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:

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

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 derived from the user's clearance. So for the price of the 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, we obtain great improvement in encryption, decryption and key-storage.

Remark 1 (Security Definitions). This scheme has been proven *IND-CPA selective* secure, that is after selecting the target parameters (in this case the attribute set and the authorities) the attacker may not distinguish between chosen plaintexts after the encryption. We observe that although the scheme of [16] 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 [26] with techniques developed with Lewko [15]), 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. Therefore, we believe that our construction in prime groups retains validity and interest, considering also that the proof is in the standard model.

References

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

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

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

[4] Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: Proc. of SP 07. pp. 321–334 (2007)

[5] Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: Proc. of EUROCRYPT 05, LNCS, vol. 3494, pp. 440–456 (2005)

[6] Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Advances in CryptologyCRYPTO 2001. pp. 213–229. Springer (2001)

[7] Chase, M.: Multi-authority attribute based encryption. In: Theory of Cryptography, pp. 515–534. Springer (2007)

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

[9] Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Cryptography and Coding, pp. 360–363. Springer (2001)

[10] Giacon, F., Aragona, R., Sala, M.: A proof of security for a key-policy RS-ABE scheme. to appear in JP J. Alg. Number Theory & Appl. (2017), https://arxiv.org/abs/1603.06635

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

[12] Herranz, J., Ruiz, A., Sáez, G.: New results and applications for multi-secret sharing schemes. Designs, Codes and Cryptography 73(3), 841–864 (2014)

[13] Hohenberger, S., Waters, B.: Attribute-based encryption with fast decryption. In: Proc. of PKC 13, LNCS, vol. 7778, pp. 162–179 (2013)

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

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

[16] Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Proc. of EUROCRYPT 11, LNCS, vol. 6632, pp. 568–588 (2011)

[17] Liang, X., Cao, Z., Lin, H., Shao, J.: Attribute based proxy re-encryption with delegating capabilities. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. pp. 276–286. ASIACCS '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1533057.1533094

[18] Liu, Z., Cao, Z., Wong, D.S.: Efficient generation of linear secret sharing scheme matrices from threshold access trees. Cryptology ePrint Archive: Listing (2010)

[19] Longo, R., Marcolla, C., Sala, M.: Key-policy multi-authority attribute-based encryption. In: Maletti A. (eds) Algebraic Informatics. CAI 2015. Lecture Notes in Computer Science, vol 9270, pp. 152–164. Springer International Publishing (2015)

[20] Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Proc. of CCS 07. pp. 195–203 (2007)
[21] Rouselakis, Y., Waters, B.: Efficient statically-secure large-universe multi-authority attribute-based encryption. In: International Conference on Financial Cryptography and Data Security. pp. 315–332. Springer (2015)
[22] Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Advances in Cryptology–EUROCRYPT 2005, pp. 457–473. Springer (2005)
[23] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Advances in cryptology. pp. 47–53. Springer (1985)
[24] Tate, J.: WC-groups over p-adic fields, Séminaire Bourbaki; 10e année: 1957/1958. Textes des conférences; Exposés 152 à 168; 2e éd. corrigée, Exposé 156, vol. 13. Secrétariat mathématique, Paris (1958)
[25] Tate, J.: Duality theorems in galois cohomology over number fields. In: Proc. Internat. Congr. Mathematicians (Stockholm, 1962). pp. 288–295 (1962)
[26] Waters, B.: Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In: Proc. of CRYPTO 09, LNCS, vol. 5677, pp. 619–636 (2009)
[27] Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Proc. of PKC 11, LNCS, vol. 6571, pp. 53–70 (2011)
[28] Weil, A.: Sur les fonctions algébriques à corps de constantes fini. C. R. Acad. Sci. Paris 210, 592–594 (1940)