Multi-authority attribute based encryption with honest-but-curious central authority

Vladimir Božović[1], Daniel Socek[2⋆], Rainer Steinwandt[1], and Viktória I. Villányi[1]

[1] Department of Mathematical Sciences, Florida Atlantic University,
777 Glades Road, Boca Raton, FL 33431, U.S.A.
[2] CoreTex Systems LLC,
2851 S Ocean Blvd. 5L, Boca Raton, FL 33432, U.S.A.
{vbozovic,dsocek,rsteinwa,vvillan}@fau.edu

Abstract. An attribute based encryption scheme capable of handling multiple authorities was recently proposed by Chase. The scheme is built upon a single-authority attribute based encryption scheme presented earlier by Sahai and Waters. Chase's construction uses a trusted central authority that is inherently capable of decrypting arbitrary ciphertexts created within the system. We present a multi-authority attribute based encryption scheme in which only the set of recipients defined by the encrypting party can decrypt a corresponding ciphertext. The central authority is viewed as "honest-but-curious": on the one hand it honestly follows the protocol, and on the other hand it is curious to decrypt arbitrary ciphertexts thus violating the intent of the encrypting party. The proposed scheme, which like its predecessors relies on the Bilinear Diffie-Hellman assumption, has a complexity comparable to that of Chase's scheme. We prove that our scheme is secure in the selective ID model and can tolerate an honest-but-curious central authority.

1 Introduction

In both standard *public key encryption* and *identity based encryption* a message is to be transmitted to a single recipient known at the time of encryption. Similarly, *broadcast encryption* addresses scenarios where a sender explicitly specifies a set of receivers (or revoked users) when encrypting a plaintext. In contrast, in an *attribute based encryption* scheme, the sender does not provide an explicit list of recipients or revoked users when encrypting a plaintext, but instead, the recipient of a ciphertext is specified through a set of credentials, also referred to as the *attributes*, which are sufficient to decrypt a ciphertext. Fuzzy identity based encryption proposed by Sahai and Waters [7] can be used to address such a setting, if all attributes are controlled by a single authority.

⋆ Work done in part at the Department of Mathematical Sciences of Florida Atlantic University.

The starting point of the current paper is a recent proposal of Chase [4] which considers *multi-authority attribute based encryption*, therewith solving an open problem from [7]. Chase's scheme is capable of handling disjoint sets of attributes that are distributed among multiple authorities. In this setting, an encrypting party specifies a set of attributes $\mathcal{A}_C$ with the attributes in $\mathcal{A}_C$ being controlled by several authorities. Let $\mathcal{A}_k$ be the set of attributes controlled by authority k. Then the ciphertext C associated with the attribute set $\mathcal{A}_C$ can only be decrypted by those users u with a set of attributes $\mathcal{A}_u$ for which the cardinality of the intersection $\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_C$ exceeds the respective threshold d_k, for each authority k.

As pointed out in [4], one of the primary challenges in implementing such a multi-authority attribute based encryption scheme is the prevention of collusion attacks among users that obtain secret key components from different authorities. Moreover, it is desirable that there be no communication between the individual authorities. To overcome these difficulties, Chase's scheme relies on a trusted central authority. The resulting scheme is capable of tolerating multiple corrupted authorities, but the honesty of the central authority remains of vital importance since, by the constriction from [4], the trusted authority has the capability of decrypting every ciphertext.

Our contribution. Building on Chase's proposal, we construct a threshold scheme for multi-authority attribute based encryption which offers the same security guarantees provided by Chase's construction, but in addition can tolerate an honest-but-curious central authority. Assuming the central authority is honest during the initialization phase, the indistinguishability of encryptions is guaranteed. As in [4], our security analysis is in the selective ID model and builds on the Decisional Bilinear Diffie Hellman assumption.

Related work. Since Shamir posed the problem of identity based encryption [8], various proposals have been made, a very partial list being the work in [6, 9, 10, 2, 5]. Building on the Bilinear Diffie Hellman assumption and the selective ID model [3, 1], at EUROCRYPT 2005 Waters presented an identity based encryption scheme in the standard model [11]. Sahai and Water's proposal for a fuzzy identity based encryption [7] provides an attribute based encryption with a single authority. Here, *fuzzy* refers to an identity id' being able to decrypt a ciphertext encrypted by an identity id if and only if id and id' are close to each other in the "set overlap" distance metric. This is of interest when dealing with noisy inputs, such as biometric templates. Building on the ideas from [7], Chase proposed a solution for multi-authority attribute based encryption, provided that a trusted central authority is available [4]. Our proposal aims at improving Chase's construction by imposing a weaker assumption on the central authority without paying a high cost in terms of efficiency.

2 Notation and preliminaries

As already mentioned, our proposal relies on the Decisional Bilinear Diffie Hellman assumption. For the sake of clarity, the next sections review the relevant terminology related to bilinear maps and multi-authority attribute based encryption. Section 2.3 discusses the security model where, like in [4], we make use of the selective ID model.

2.1 Bilinear maps and the Bilinear Diffie Hellman assumption

Let G_1, G_2 be groups of prime order p, and let P a generator of G_1. We assume q to be superpolynomial in the security parameter ℓ and that all group operations in G_1 and G_2 can be computed efficiently, i. e., in probabilistic polynomial time. We use additive notation for G_1 and multiplicative notation for G_2. By $e : G_1 \times G_1 \longrightarrow G_2$ we denote an admissible bilinear map, i. e., all of the following hold [2]:

- For all $P, Q \in G_1$ and for all $\alpha, \beta \in \mathbb{Z}$ we have $e(\alpha P, \beta Q) = e(P, Q)^{\alpha\beta}$.
- We have $e(P, P) \neq 1$, i. e., $e(P, P)$ is a generator of G_2.
- There is a probabilistic polynomial time algorithm that for arbitrary $P, Q \in G_1$ computes $e(P, Q)$.

In the above setting, the Decisional Bilinear Diffie Hellman (D-BDH) problem in (G_1, G_2, e) is the problem of distinguishing between the challenger's possible outputs in the following experiment: The challenger chooses $\alpha, \beta, \gamma, \eta \leftarrow \{0, 1, \ldots, p-1\}$ independently and uniformly at random, flips a fair binary coin $\delta \leftarrow \{0, 1\}$, and then outputs the tuple

$$(P, \alpha P, \beta P, \gamma P, e(P, P)^{\delta \cdot \alpha\beta\gamma + (1-\delta)\cdot\eta}).$$

In other words, with probability 1/2 the last component of the challenger's output is $e(P, P)^{\alpha\beta\gamma}$, and with probability 1/2 the last component is a uniformly at random chosen element from G_2. We define the *advantage* of algorithm $\mathcal{A}$ in solving the D-BDH problem as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{bdh}}(\ell) := \Pr(\delta' = \delta) - \frac{1}{2}$$

where δ' is the output of $\mathcal{A}$ when trying to guess the value of the fair binary coin δ. We say that an algorithm $\mathcal{A}$ has a *non-negligible advantage* in solving the D-BDH problem, if $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{bdh}}$ is not negligible[1] where the probability is over the randomly chosen $\alpha, \beta, \gamma, \eta$ and the random bits consumed by $\mathcal{A}$.

Definition 1 (Decisional Bilinear Diffie Hellman assumption). *The Decisional Bilinear Diffie Hellman assumption holds for* (G_1, G_2, e) *if there exists no probabilistic polynomial time algorithm having non-negligible advantage in solving the above* D-BDH *problem.*

2.2 Authorities, attributes and users

Let $\mathcal{K}$ be the polynomial size set of authorities and $\mathcal{U}$ the polynomial size set of users we consider, and denote by $\mathcal{A}_k$ the polynomial size set of attributes handled by authority $k \in \mathcal{K}$. We impose that the sets $\mathcal{A}_k$ are pairwise disjoint, i. e., the *universal attribute set*

$$\mathcal{A} := \biguplus_{k \in \mathcal{K}} \mathcal{A}_k$$

is the disjoint union of the $\mathcal{A}_k$. In addition to the authorities $k \in \mathcal{K}$, there is one central authority $k_{\mathsf{CA}} \notin \mathcal{K}$ which we will model as honest-but-curious—the central authority k_{CA} honestly follows the protocol, but will try to decrypt ciphertexts sent by users in the system. During an initialization phase we allow communication between k_{CA} and k for each authority $k \in \mathcal{K}$, but thereafter no communication between the central authority and the authorities $k \in \mathcal{K}$ is possible: while the central authority k_{CA} is involved in setting up the system, we do not want to rely on k_{CA} being available throughout the complete lifetime of the system. Also, we do not allow any communication among the authorities in $\mathcal{K}$.

To distinguish different users, we follow [4] and assume that each user $u \in \mathcal{U}$ has a unique identifier. Depending on the application, the identifier could refer to a social security number or a passport number, for instance. We denote the set of those attributes in $\mathcal{A}$ that are available to user $u \in \mathcal{U}$ by $\mathcal{A}_u$. Similarly, we write $\mathcal{A}_C$ for the set of attributes that is associated with a ciphertext C. This set $\mathcal{A}_C$ is chosen by the encrypting party as part of the input to the encryption algorithm, the other part of the input being the plaintext. We associate with each authority $k \in \mathcal{K}$ a threshold $d_k \in \mathbb{N}_{>0}$. The goal is that exactly those users u satisfying

[1] We refer to a function $f : \mathbb{N}_{>0} \longrightarrow \mathbb{R}$ as negligible, if $|f| = |f(\ell)| \in \frac{1}{\ell^{\omega(1)}}$.

$$|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_C| \geq d_k \text{ for every } k \in \mathcal{K}$$

are able to decrypt the ciphertext C. In other words, for each authority k, user u must have at least d_k of the attributes that have been specified at the time of encryption. To decrypt a ciphertext, user $u \in \mathcal{U}$ uses the secret keys obtained during the initialization phase from the authorities $k \in \mathcal{K}$. Figure 1 lists the main components of a multi-authority attribute based encryption scheme (cf. [4]).

Setup. A probabilistic polynomial time algorithm[a] that given the security parameter 1^ℓ, a list of pairwise disjoint sets of attributes $[\mathcal{A}_k]_{k\in\mathcal{K}}$ and thresholds $[d_k]_{k\in\mathcal{K}}$ generates

- a (public key, secret key)-pair for each attribute authority $k \in \mathcal{K}$
- public system parameters.

Attribute key generation. A probabilistic polynomial time algorithm that given an attribute authority k's secret key, the corresponding threshold d_k, a (unique identifier of a) user u and a subset $\mathcal{A}_u \subseteq \mathcal{A}_k$ outputs decryption keys for user u.

Encryption. A probabilistic polynomial time algorithm that given a plaintext, attributes $\mathcal{A}_C \subseteq \mathcal{A}$ and the public system parameters, outputs a ciphertext C.

Decryption. A deterministic polynomial time algorithm that given a set of decryption keys for a set of attributes $\mathcal{A}_u$ and a ciphertext C encrypted with attribute set $\mathcal{A}_C$, outputs the corresponding plaintext M if $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_C| \geq d_k$ for all attribute authorities $k \in \mathcal{K}$; otherwise it outputs an error symbol $\perp$.

[a] It may be preferable to realize this computation in a distributed fashion, involving individual attribute authorities and some central authority. Below we will use such a distributed realization.

Fig. 1. Algorithms in a multi-authority attribute based encryption scheme.

Remark 1. Unlike [4] we do not make use of a *central key generation algorithm*, run by the central authority k_{CA} to generate secret keys for users u. Without loss of generality, in the security model we therefore will not give the adversary the possibility to query k_{CA} for private user keys. In the scheme we discuss, private user keys are generated by the attribute authorities $k \in \mathcal{K}$ only.

A crucial feature of a multi-authority attribute based encryption scheme is the prevention of collusions among users: we want to prevent that any set of users, each of which is not able to decrypt a ciphertext C, can combine their information to decrypt C. The security definition discussed next tries to capture this design goal.

2.3 Security model

Like [4], we use a selective ID model for the security analysis. The adversary $\mathcal{H}$ has to specify the set of attributes that he wants to attack before receiving any public keys of the system. Figure 2 shows the game an adversary has to win to defeat the security of our scheme. As in [4], for our security analysis we impose the technical restriction that the adversary does not query the same attribute authority twice for private keys of the same user.

For a multi-authority attribute based encryption scheme to be secure, we require that there is no efficient algorithm achieving a non-negligible advantage in the game in Figure 2. More specifically, we define the advantage of an adversary $\mathcal{H}$ in the game in Figure 2 as

$$\mathrm{Adv}_{\mathcal{H}}^{\mathsf{sid}}(\ell) := \Pr(\delta' = \delta) - \frac{1}{2}$$

and make the following definition.

Definition 2 (Security in the selective ID model). *A scheme for multi-authority attribute based encryption is* secure in the selective ID model*, if for all probabilistic polynomial time adversaries $\mathcal{H}$, the advantage* $\mathrm{Adv}_{\mathcal{H}}^{\mathsf{sid}}(\ell)$ *is negligible.*

The security requirement in Definition 2 does not address the question which information is available to the central authority. Specifically, in Chase's scheme [4], the central authority has the capability of reading arbitrary ciphertexts constructed by the users within the system. To express a requirement that limits the possibilities of an honest-but-curious central authority, we take a more detailed look at the setup phase, which is combined into a single algorithm in Figure 1. More precisely, this step can be seen as a simple protocol where the central authority k_{CA} securely communicates with the attribute authorities.

Remark 2. From a practical perspective, it is desirable to have no communication among attribute authorities, and only very limited interaction of the central authority with each attribute authority. In the protocol in Section 3, the central authority sends one message to each attribute authority and derives the public system parameters from the replies.

The game in Figure 3 captures a setting where an honest-but-curious central authority tries to violate the indistinguishability of ciphertexts. We introduce a "curious" algorithm $\mathcal{B}$ which, similarly as the "outside

Setup

1. Given the security parameter 1^ℓ, the adversary $\mathcal{H}$ outputs
   - a non-empty list $\mathcal{U}$ of (unique identifiers of) users
   - a non-empty list $\mathcal{K}$ of (unique identifiers of) attribute authorities
   - a list $[(\mathcal{A}_k, \mathsf{corrupted}, d_k)]_{k\in\mathcal{K}}$ of non-empty, pairwise disjoint attribute sets, each along with a threshold $d_k \in \mathbb{N}_{>0}$ and a flag indicating if the respective authority is corrupted. There must be at least one uncorrupted authority.[a]
   - a non-empty set of attributes $\mathcal{A}_C \subseteq \biguplus_{k\in\mathcal{K}} \mathcal{A}_k$ that will be associated with the challenge ciphertext.
2. The public and secret keys are generated, and $\mathcal{H}$ learns
   - the public keys of all attribute authorities
   - the public system parameters
   - the complete history of all those authorities $k \in \mathcal{K}$ that are corrupted.

Secret key queries

The adversary can query the authorities $k \in \mathcal{K}$ for private user keys for attributes in $\mathcal{A}_k$ for user u. Whenever the adversary queries k for a secret key for attribute $a \in \mathcal{A}_k$ for user u, the attribute a is added to the (initially empty) set $\mathcal{A}_u$. The only restrictions for secret key queries are the following:
- at any time, for each user u there is at least one uncorrupted authority $\hat{k} = \hat{k}(u)$ with $|\mathcal{A}_u \cap \mathcal{A}_{\hat{k}} \cap \mathcal{A}_C| < d_{\hat{k}}$[b]
- for each user u, no authority $k \in \mathcal{K}$ is queried more than once for private keys of u.

Challenge

1. The adversary $\mathcal{H}$ outputs two equal length messages M_0, M_1.
2. The challenger flips a fair binary coin $\delta \leftarrow \{0, 1\}$ and then applies the encryption algorithm to M_δ and the attribute set $\mathcal{A}_C$.
3. The resulting ciphertext C is given to the adversary $\mathcal{H}$.

Further secret key queries

The adversary can query for further private keys of users, subject to the same restrictions as before: for each user u there is at least one uncorrupted authority $\hat{k} = \hat{k}(u)$ with $|\mathcal{A}_u \cap \mathcal{A}_{\hat{k}} \cap \mathcal{A}_C| < d_{\hat{k}}$, and for each user u, no authority $k \in \mathcal{K}$ is queried more than once for private keys of u.

Guess

The adversary $\mathcal{H}$ outputs a guess δ' for the challenger's secret coin δ.

[a] Note that the central authority k_{CA} is not included in this list and in particular cannot be corrupted.

[b] The uncorrupted authority $\hat{k} = \hat{k}(u)$ may be different for each user u.

Fig. 2. Attacking multi-authority attribute based encryption in the selective ID model.

adversary" $\mathcal{H}$ in Figure 2, fixes the attribute sets and their distribution among the attribute authorities. Further on, $\mathcal{B}$ specifies the set of attributes that will be associated with the challenge ciphertext. At the end of the setup phase, $\mathcal{B}$ learns the complete state of the central authority, and based on this knowledge then tries to violate the indistinguishability of ciphertexts.
For an algorithm $\mathcal{B}$, we define the advantage in the game in Figure 3 as

$$\mathrm{Adv}^{\mathsf{ca}}_{\mathcal{B}}(\ell) := \left|\Pr(\delta' = \delta) - \frac{1}{2}\right| \quad .$$

Setup

1. Given the security parameter 1^ℓ, the algorithm $\mathcal{B}$ outputs
   - a non-empty list $\mathcal{U}$ of (unique identifiers of) users
   - a non-empty list $\mathcal{K}$ of (unique identifiers of) attribute authorities
   - a list $[(\mathcal{A}_k, \mathsf{corrupted}, d_k)]_{k\in\mathcal{K}}$ of non-empty, pairwise disjoint attribute sets, each along with a threshold $d_k \in \mathbb{N}_{>0}$ and a flag indicating if the respective authority is corrupted. There must be at least one uncorrupted authority.[a]
2. The public and secret keys of all authorities $k \in \mathcal{K}$ are generated, and $\mathcal{B}$ learns
   - all public keys
   - the public system parameters
   - the complete history of all those authorities $k \in \mathcal{K}$ that are corrupted
   - the complete history of the central authority k_{CA}.

Challenge

1. The algorithm $\mathcal{B}$ outputs two equal length messages M_0, M_1 and a non-empty set of attributes $\mathcal{A}_C \subseteq \biguplus_{k\in\mathcal{K}} \mathcal{A}_k$.
2. The challenger flips a fair binary coin binary $\delta \leftarrow \{0,1\}$ and then applies the encryption algorithm to M_δ and the attribute set $\mathcal{A}_C$.
3. The resulting ciphertext C is given to $\mathcal{B}$.

Guess

The algorithm $\mathcal{B}$ outputs a guess δ' for the challenger's secret coin δ.

[a] Note that the central authority k_{CA} is not included in this list and in particular cannot be corrupted.

Fig. 3. Dealing with an honest-but-curious central authority.

Definition 3 (Tolerating an honest-but-curious central authority). *A scheme for multi-authority attribute based encryption* can tolerate an honest-but-curious central authority*, if for all probabilistic time algorithms $\mathcal{B}$, the advantage* $\mathrm{Adv}_{\mathcal{B}}^{\mathsf{ca}}(\ell)$ *is negligible.*

Remark 3. Unlike for the adversary $\mathcal{H}$ in Figure 2, we do not require that an honest-but-curious central authority specifies the challenge attributes $\mathcal{A}_C$ in advance: algorithm $\mathcal{B}$ in Figure 3 does not have to provide this set before the challenge phase.

We are now in the position to describe our suggestion for a multi-authority attribute based encryption scheme and to show it is secure in the sense of both Definition 2 and Definition 3.

3 Proposed protocol

We adopt the notation from Section 2 with G_1, G_2 being groups of prime order p, P a generator of G_1 and $e : G_1 \times G_1 \longrightarrow G_2$ an admissible bilinear map. We assume the unique identifiers for users u and for the attribute authorities $k \in \mathcal{K}$ to be public. Similarly, we assume the sets of attributes $\mathcal{A}_k$ and the corresponding threshold d_k to be public—in particular, all these values are known to the central authority k_{CA}, which we invoke (only) in the setup phase. In order to generate secret keys for users, we assume that each attribute $a \in \mathcal{A}$ can be identified with a number $\iota(a) \in \{1, \dots, p-1\}$—for practical purposes, $\iota(a)$ could be based on a hash value, for instance.

For the sake of clarity, we break the protocol description down into steps. We start with a basic protocol, which is then modified, yielding the final proposal.

3.1 The basic protocol

Setup. The setup phase requires one message to be sent from the central authority to each of the attribute authorities. It is assumed that the adversary has no possibility to interfere with or to access this communication:

The central authority k_{CA} chooses, for each pair $(k, u) \in \mathcal{K} \times \mathcal{U}$, uniformly at random a secret value $s_{k,u} \leftarrow \{0, \dots, p-1\}$. Then, the parameter σ is set as follows:

$$\sigma := \sum_{k \in \mathcal{K}} s_{k,u} \pmod{p}. \tag{1}$$

The sequence

$$\underbrace{[s_{k,u} \cdot P}_{=:S_{k,u}}]_{u \in \mathcal{U}}$$

is sent to attribute authority k ($k \in \mathcal{K}$), and k_{CA} publishes the public system parameter

$$\underbrace{e(P,P)^{\sigma}}_{=:\mathsf{pk}}.$$

Attribute authority $k \in \mathcal{K}$ receives the corresponding sequence of $S_{k,u}$-values from k_{CA} and chooses a value $r_k \leftarrow \{0, \ldots, p-1\}$ uniformly at random. Moreover, for each of its attributes $a \in \mathcal{A}_k$, a secret value $t_{k,a} \leftarrow (\mathbb{Z}/p\mathbb{Z})^*$ is chosen uniformly at random by k, and the pair

$$\left(e(P,P)^{r_k}, [\underbrace{t_{k,a} \cdot P}_{=:T_{k,a}}]_{a \in \mathcal{A}_k}\right)$$

forms k's public key. The secret key of k contains the aforementioned values r_k, $[S_{k,u}]_{u \in \mathcal{U}}$, and $[t_{k,a}]_{a \in \mathcal{A}_k}$. Finally, for each user $u \in \mathcal{U}$, attribute authority k chooses uniformly at random a secret polynomial $f_{k,u} \in \mathbb{F}_p[X]$ of degree $< d_k$.

Remark 4. The value $e(P,P)^{r_k}$ is only used during encryption and decryption to compute the product $\mathsf{pk} \cdot \prod_{k \in \mathcal{K}} e(P,P)^{r_k}$—which is ciphertext-independent. If one allows the attribute authorities to contribute to the generation of the public system parameters, the $e(P,P)^{r_k}$-component in the attribute authorities' public keys can be omitted. To do so, the public system parameter $\mathsf{pk} = e(P,P)^{\sigma}$ can be replaced with $e(P,P)^{\sigma + \sum_{k \in \mathcal{K}} r_k}$.

Attribute key generation. To extract the secret decryption key associated with an attribute $a \in \mathcal{A}_k \cap \mathcal{A}_u$ for a user $u \in \mathcal{U}$, attribute authority k proceeds as follows:

- The secret value $X_{k,u} := S_{k,u} + (r_k - f_{k,u}(0)) \cdot P$, which depends on k and u, but not the specific attribute a, is computed and given to u.
- The attribute-specific value $D_{k,u,a} := \frac{f_{k,u}(\iota(a))}{t_{k,a}} \cdot P$ is computed and given to u.

Encryption. To encrypt a plaintext $M \in G_2$ with associated attribute set $\mathcal{A}_C \subseteq \mathcal{A}$, the encrypting party chooses $s \leftarrow \{0, \ldots, p-1\}$ uniformly at random and computes the ciphertext

$$\left(\left(\mathsf{pk} \cdot \prod_{k \in \mathcal{K}} e(P,P)^{r_k}\right)^s \cdot M, s \cdot P, [s \cdot T_{k,a}]_{a \in \mathcal{A}_C}\right).$$

Decryption. Let $C = ((\mathsf{pk} \cdot \prod_{k\in\mathcal{K}} e(P,P)^{r_k})^s \cdot M, s \cdot P, [s \cdot T_{k,a}]_{a\in\mathcal{A}_C})$ be a ciphertext with associated attribute set $\mathcal{A}_C$, and suppose that user u's attribute set $\mathcal{A}_u$ satisfies $|\mathcal{A}_u \cap \mathcal{A}_k| \geq d_k$ for all $k \in \mathcal{K}$. Then u can recover the plaintext M as follows.

1. For each $k \in \mathcal{K}$, he chooses d_k attributes $a \in \mathcal{A}_u \cap \mathcal{A}_k$, and computes
$$e(s \cdot T_{k,a}, D_{k,u,a}) = e(P,P)^{f_{k,u}(\iota(a))\cdot s}.$$
Then, using Lagrange polynomial interpolation, u computes
$$e(P,P)^{f_{k,u}(0)\cdot s}.$$
2. Further on, for each $k \in \mathcal{K}$, user u can use the $X_{k,u}$-component of his secret key to compute $e(X_{k,u}, s \cdot P) = e(P,P)^{(s_{k,u}+r_k-f_{k,u}(0))\cdot s}$.
3. Finally, user u computes the following product:
$$\begin{aligned}
&\prod_{k\in\mathcal{K}} e(P,P)^{f_{k,u}(0)\cdot s} \cdot e(P,P)^{(s_{k,u}+r_k-f_{k,u}(0))\cdot s} \\
&= e(P,P)^{s\cdot\sum_{k\in\mathcal{K}}(s_{k,u}+r_k)} \\
&= e(P,P)^{s\cdot(\sigma+\sum_{k\in\mathcal{K}} r_k)} \\
&= \left(\mathsf{pk} \cdot \prod_{k\in\mathcal{K}} e(P,P)^{r_k}\right)^s .
\end{aligned}$$
By inverting this element and multiplying the result with the first component of the ciphertext, the plaintext M can be recovered.

3.2 Improving flexibility

We can make the aforementioned basic protocol more flexible by allowing the addition of new authorities to a previously established protocol. For this purpose, we will change the setup phase through the introduction of dummy values $s_{k_{\mathsf{CA}},u}$ $(u \in \mathcal{U})$. This causes a corresponding modification of the decryption algorithm.

Setup. The setup phase remains the same, except that now the central authority k_{CA} computes for each user $u \in \mathcal{U}$ the additional "dummy secret" $s_{k_{\mathsf{CA}},u} := \sigma - \sum_{k\in\mathcal{K}} s_{k,u}$. The corresponding "dummy public key" $s_{k_{\mathsf{CA}},u} \cdot P$ is sent to user u. Now, to add a new authority k^*, the central authority k_{CA} replaces the old value σ with a new uniformly at random chosen σ', and replaces each $s_{k_{\mathsf{CA}},u}$ with $\sigma' - \sum_{k\in\mathcal{K}\cup\{k^*\}} s_{k,u}$. Then the updated "dummy public keys" $s_{k_{\mathsf{CA}},u} \cdot P$ have to be communicated to the users, and the new authority k^* can compute its secret and public key as before.

Decryption. After a user receives the dummy public key $s_{k_{\mathsf{CA}},u} \cdot P$, he can perform the following decryption phase, which deviates only in the third step from the previous decryption algorithm.

1. For each $k \in \mathcal{K}$, he chooses d_k attributes $a \in \mathcal{A}_u \cap \mathcal{A}_k$, and computes

$$e(s \cdot T_{k,a}, D_{k,u,a}) = e(P,P)^{f_{k,u}(\iota(a)) \cdot s}.$$

   Then, using Lagrange polynomial interpolation, u computes

$$e(P,P)^{f_{k,u}(0) \cdot s}.$$

2. Further on, for each $k \in \mathcal{K}$, user u can use the $X_{k,u}$-component of his secret key to compute $e(X_{k,u}, s \cdot P) = e(P,P)^{(s_{k,u}+r_k-f_{k,u}(0)) \cdot s}$.
3. Multiplying $e(s \cdot P, s_{k_{\mathsf{CA}},u} \cdot P)$ with all of the above values yields

$$\begin{aligned}
&e(s \cdot P, s_{k_{\mathsf{CA}},u} \cdot P) \cdot \prod_{k \in \mathcal{K}} e(P,P)^{f_{k,u}(0) \cdot s} \cdot e(P,P)^{(s_{k,u}+r_k-f_{k,u}(0)) \cdot s} \\
&= e(P,P)^{s \cdot s_{k_{\mathsf{CA}},u}} \cdot e(P,P)^{s \cdot \sum_{k \in \mathcal{K}} (s_{k,u}+r_k)} \\
&= e(P,P)^{s \cdot (\sigma + \sum_{k \in \mathcal{K}} r_k)} \\
&= \left(\mathsf{pk} \cdot \prod_{k \in \mathcal{K}} e(P,P)^{r_k} \right)^s .
\end{aligned}$$

   By inverting this element and multiplying the result with the first component of the ciphertext, the plaintext M can be recovered.

3.3 The proposed protocol

In general, it is not desirable for the central authority to have to communicate with the users in every update phase. At the cost of an increased size of the public system parameters, an update can be performed without this communication. More specifically, in our final protocol, we use the pair

$$\Big([s_{k_{\mathsf{CA}},u} \cdot P]_{u \in \mathcal{U}}, \underbrace{e(P,P)^\sigma}_{=:\mathsf{pk}} \Big).$$

as public system parameters. As the decryption algorithm has access to the public system parameters, no modification to the decryption algorithm just described is necessary, and users can decrypt as before.

4 Security analysis

In this section, we prove security of the proposed protocol both in the sense of Definition 2 and Definition 3. We start with the former and show security in the selective ID model.

4.1 Security in the selective ID model

Our proof builds on the analysis of Chase's scheme in [4], and it is worth noting that the reduction to a $\mathsf{D\text{-}BDH}$ adversary $\mathcal{S}$ in the proof below is tight: Essentially, the advantage of the adversary $\mathcal{H}$ violating security in the selective ID model is only halved at the cost of simulating the attribute authorities k and the central authority k_{CA}.

Theorem 1. *Suppose there exists a probabilistic polynomial time adversary $\mathcal{H}$ against the protocol in Section 3 having a non-negligible advantage in the game in Figure 2. Then there is a probabilistic polynomial time algorithm $\mathcal{S}$ having a non-negligible advantage in solving the* $\mathsf{D\text{-}BDH}$*-problem.*

Proof. As explained in Section 2.1, the input of the $\mathsf{D\text{-}BDH}$ adversary $\mathcal{S}$ is a tuple

$$(P, \alpha P, \beta P, \gamma P, e(P,P)^{\delta\cdot\alpha\beta\gamma+(1-\delta)\cdot\eta}) \tag{2}$$

with $\delta \leftarrow \{0,1\}$ being chosen uniformly random. To find δ, the algorithm $\mathcal{S}$ runs a simulation of $\mathcal{H}$, and subsequently we refer to $\mathcal{S}$ as the *simulator*: it will simulate all attribute authorities and the central authority to $\mathcal{H}$, and $\mathcal{S}$ will answer all queries for user keys made by $\mathcal{H}$. More specifically, $\mathcal{S}$ mimics the individual phases of the game in Figure 2 as follows:

Setup. The simulator uses the attribute authorities, thresholds and attribute sets specified by $\mathcal{H}$. For corrupted authorities the simulator follows exactly the original protocol specification, so that the history of such an authority (which is revealed to $\mathcal{H}$) follows the same distribution as in the game in Figure 2. Honest attribute authorities are also simulated by $\mathcal{S}$, but instead of computing the public key of an uncorrupted authority k as $(e(P,P)^{r_k}, [t_{k,a}\cdot P]_{a\in\mathcal{A}_k})$, the simulator uses the public key $(e(P,P)^{r_k}, [t_{k,a}\cdot Q]_{a\in\mathcal{A}_k})$ where

$$Q := \begin{cases} P & \text{, if } a \in \mathcal{A}_C \\ \beta P & \text{, if } a \notin \mathcal{A}_C \end{cases}$$

with βP being part of the $\mathsf{D\text{-}BDH}$-challenge. In other words, for attributes $a \in \mathcal{A}_k \setminus \mathcal{A}_C$ handled by honest authorities, the random value $t_{k,a}$ is multiplied with the point βP instead of P. As G_1 is of prime order, with overwhelming probability βP generates G_1 and for $\mathcal{H}$ the distribution of the public keys does not change compared to the game in Figure 2. Reflecting the above modification of public keys, the computation of the polynomials $f_{k,u}$ by honest authorities will also be modified, and the simulator $\mathcal{S}$ will define the polynomials $f_{k,u}$ implicitly when answering secret key queries as detailed below.

When simulating the central authority k_{CA}, the simulator follows the steps of the original protocol, with the following exceptions:

– The value pk in the public system parameters is computed as

$$\mathsf{pk} := e(\alpha P, \beta P) \tag{3}$$

where αP and βP are part of the $\mathsf{D\text{-}BDH}$ challenge. For the adversary $\mathcal{H}$, the usage of this modified pk-value instead of $e(P,P)^\sigma$ makes no difference. Because of G_2 being of prime order, with overwhelming probability $\mathsf{pk} = e(P,P)^{\alpha\beta}$ is a uniformly distributed element in G_2. Similarly, the original value $e(P,P)^\sigma$ is for $\mathcal{H}$ indistinguishable from a uniformly at random chosen group element. The only information on σ that is potentially available to $\mathcal{H}$, are
  - $S_{k,u}$-values of corrupted authorities,
  - $[s_{k_{\mathsf{CA}},u} \cdot P]_{u \in \mathcal{U}}$,
  - $X_{k,u}$-values obtained from secret user key queries.

By assumption, for each $u \in \mathcal{U}$, at least one authority $\hat{k}(u)$ is uncorrupted, and hence the first two of the above listed items alone do not reveal any information on σ. Even with the knowledge of the $S_{k,u}$-values of all corrupted authorities and $[s_{k_{\mathsf{CA}},u} \cdot P]_{u \in \mathcal{U}}$, each value of σ remains equally likely, as for each $u \in \mathcal{U}$ Equation (1) contains at least one unknown random value $s_{\hat{k}(u),u}$. The only potentially available information on $s_{\hat{k}(u),u}$ is the value $X_{\hat{k}(u),u}$ obtained from a secret user key query. However, due to the subtraction of the random value $\tilde{f}_{k,u}(0) \cdot P$, each $X_{k,u}$ is an independent random value, containing no information on $s_{k,u}$ or σ.

– The simulator only chooses the "dummy secrets" $s_{k_{\mathsf{CA}},u}$ ($u \in \mathcal{U}$) and the $s_{k,u}$-values of corrupted authorities uniformly at random. For honest authorities, the $s_{k,u}$-values will be determined later as needed.

Secret key queries. We can w. l. o. g. assume that $\mathcal{H}$ does not query secret user keys from corrupted attribute authorities, as $\mathcal{H}$ can compute such user keys itself. For uncorrupted attribute authorities, the simulator $\mathcal{S}$

must be able to answer secret key queries from $\mathcal{H}$, and we distinguish two cases:[2]

1. $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_C| < d_k$ *and there has not been a previous secret key query for user* u *to an authority* $k' \neq k$ *with* $|\mathcal{A}_u \cap \mathcal{A}_{k'} \cap \mathcal{A}_C| < d_{k'}$*:* W. l. o. g., we may assume $|\mathcal{A}_k \cap \mathcal{A}_C| = d_k - 1$ (otherwise we can modify $\mathcal{H}$ to ask for further secret user keys which will be ignored). The simulator implicitly defines $f_{k,u}$ by specifying the values of $f_{k,u}$ at d_k points. Namely, the simulator chooses uniformly at random $\rho_{k,u,a} \in \mathbb{F}_p$ for all $a \in \mathcal{A}_k \cap \mathcal{A}_C$, a random value $\hat{\rho}_{k,u} \in \mathbb{F}_p$ and imposes
$$\begin{aligned} f_{k,u}(\iota(a)) &= \beta \cdot \rho_{k,u,a} \quad \text{for all } a \in \mathcal{A}_k \cap \mathcal{A}_C \text{ and} \\ f_{k,u}(0) &= \beta \cdot (\alpha + \hat{\rho}_{k,u}) \end{aligned}$$
with αP, βP being part of the D-BDH challenge. With overwhelming probability $\beta \neq 0$ and $f_{k,u}$ follows the same distribution as in the original protocol. Now $\mathcal{S}$ can use the values αP, βP from the D-BDH challenge to extract the requested secret key $(X_{k,u}, D_{k,u,a})$ for user $u \in \mathcal{U}$ and attribute $a \in \mathcal{A}_k \cap \mathcal{A}_u$:
   - For $a \in \mathcal{A}_C$, we have $D_{k,u,a} = (\rho_{k,u,a}/t_{k,a}) \cdot \beta P$.
   - Because of
$$\frac{f_{k,u}(0)}{t_{k,a} \cdot \beta} \cdot P = \frac{1}{t_{k,a}} \cdot (\alpha P + \hat{\rho}_{k,u} P)$$
the simulator $\mathcal{S}$ can compute the d_k points
$$\frac{f_{k,u}(0)}{t_{k,a} \cdot \beta} \cdot P, \left[\underbrace{\frac{f_{k,u}(\iota(a))}{t_{k,a} \cdot \beta}}_{\rho_{k,u,a}/t_{k,a}} \cdot P \right]_{a \in \mathcal{A}_k \cap \mathcal{A}_C}$$
and then use Lagrange interpolation to derive
$$D_{k,u,a} = \frac{f_{k,u}(\iota(a))}{t_{k,a} \cdot \beta} \cdot P$$
for $a \notin \mathcal{A}_C$.
   - Finally, the simulator computes
$$X_{k,u} := r_k \cdot P - \hat{\rho}_{k,u} \cdot \beta P - \left(\sum_{\kappa \in (\mathcal{K} \cup \{k_{\mathsf{CA}}\}) \setminus \{k\}} s_{\kappa,u} \right) \cdot P,$$

[2] Here we exploit that $\mathcal{H}$ never queries the same authority k twice with the same user u, and that for $k \neq k'$ we have $\mathcal{A}_k \cap \mathcal{A}_{k'} = \emptyset$ (cf. [4, Remark 1]). These assumptions ensure that the validity of $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_C| < d_k$ does not depend on the future secret key queries of $\mathcal{H}$.

choosing, for the user u, all $S_{\kappa,u}$ ($\kappa \in \mathcal{K} \setminus \{k\}$), that have not been fixed already, as $S_{\kappa,u} := s_{\kappa,u} \cdot P$ with a uniformly at random chosen $s_{\kappa,u}$. With the modified value of pk in (3), this choice of $X_{k,u}$ implicitly fixes $s_{k,u} := \alpha\beta - \sum_{\kappa \in (\mathcal{K} \cup \{k_{\mathsf{CA}}\}) \setminus \{k\}} s_{\kappa,u}$.

2. $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_C| \geq d_k$ *or there has been a previous secret key query for user* u *to an authority* $k' \neq k$ *with* $|\mathcal{A}_u \cap \mathcal{A}_{k'} \cap \mathcal{A}_C| < d_{k'}$*:* In this case, the simulator chooses a random polynomial $\tilde{f}_{k,u} \in \mathbb{F}_p[X]$ of degree $< d_k$ and implicitly defines $f_{k,u} := \beta \cdot \tilde{f}_{k,u}$ (with βP being part of the $\mathsf{D\text{-}BDH}$ challenge). Note that with overwhelming probability $\beta \neq 0$ and $f_{k,u}$ follows the same distribution as in the original protocol. Using the value βP from the $\mathsf{D\text{-}BDH}$ challenge, $\mathcal{S}$ can compute the respective secret key $(X_{k,u}, D_{k,u,a})$ for user $u \in \mathcal{U}$ and attribute $a \in \mathcal{A}_k \cap \mathcal{A}_u$ as follows:
$$X_{k,u} := S_{k,u} + r_k \cdot P - \tilde{f}_{k,u}(0) \cdot \beta P \text{ and}$$
$$D_{k,u,a} := \begin{cases} \frac{\tilde{f}_{k,u}(\iota(a))}{t_{k,a}} \cdot \beta P & \text{, if } a \in \mathcal{A}_C \\ \frac{\tilde{f}_{k,u}(\iota(a))}{t_{k,a}} \cdot P & \text{, if } a \notin \mathcal{A}_C \end{cases}$$
At this point, the value $S_{k,u}$, if not fixed already through a previous secret key query (see above), is chosen as $S_{k,u} := s_{k,u} \cdot P$ with a uniformly at random chosen $s_{k,u}$.

Challenge. Let $M_0, M_1 \in G_2$ be the challenge messages selected by $\mathcal{H}$, and let δ be the value to be found by the $\mathsf{D\text{-}BDH}$ adversary $\mathcal{S}$ (see (2)). Using a fair binary coin $\mu \leftarrow \{0,1\}$ and the last two components of the $\mathsf{D\text{-}BDH}$ challenge, the simulator hands the challenge ciphertext

$$\left(e(P,P)^{\delta \cdot \alpha\beta\gamma + (1-\delta)\cdot\eta} \cdot e(\gamma P, P)^{\sum_{k \in \mathcal{K}} r_k} \cdot M_\mu, \gamma P, [t_{k,a} \cdot \gamma P]_{a \in \mathcal{A}_C}\right) \quad (4)$$

for M_μ to $\mathcal{H}$. We consider both possible cases $\delta = 0$ and $\delta = 1$:

$\delta = 0$**:** Because of $e(P,P)^{\delta \cdot \alpha\beta\gamma + (1-\delta)\cdot\eta} = e(P,P)^\eta$ with a uniformly at random chosen $\eta \leftarrow \{0, \ldots, p-1\}$, the challenge ciphertext contains no information on M_μ.

$\delta = 1$**:** Because of $\mathsf{pk} = e(\alpha P, \beta P)$, in this case we can rewrite the challenge ciphertext (4) as

$$\left(\left(\mathsf{pk} \cdot \prod_{k \in \mathcal{K}} e(P,P)^{r_k}\right)^\gamma \cdot M_\mu, \gamma P, [\gamma \cdot t_{k,a} P]_{a \in \mathcal{A}_C}\right),$$

which is a valid encryption of M_μ.

Further secret key queries. Here the simulator proceeds exactly as with secret key queries prior to the challenge phase, maintaining consistency with already answered secret key queries.

Guess. Denote by μ' the output of $\mathcal{H}$. The output of the simulator $\mathcal{S}$ is given by

$$\delta' := \begin{cases} 1 \text{ , if } \mu = \mu' \\ 0 \text{ , if } \mu \neq \mu' \end{cases} \quad .$$

In other words, $\mathcal{S}$ considers the last component of the $\textsf{D-BDH}$ challenge to be $e(P,P)^{\alpha\beta\gamma}$ whenever $\mathcal{H}$ correctly identifies M_μ. As in case of $\delta = 0$ the challenge ciphertext contains no information on μ, the adversary's $\mathcal{H}$ probability to find the correct μ-value is $1/2$. Consequently, the probability that $\mathcal{S}$ returns a correct guess for δ in this case is $1/2$, too:

$$\Pr(\delta' = \delta \mid \delta = 0) = \frac{1}{2} \quad . \tag{5}$$

If $\delta = 1$, the adversary $\mathcal{H}$ faces a valid encryption of M_μ, and we obtain

$$\Pr(\delta' = \delta \mid \delta = 1) = \Pr(\mu' = \mu \mid \delta = 1) = \frac{1}{2} + \mathrm{Adv}^{\mathsf{sid}}_{\mathcal{H}}(\ell) \quad . \tag{6}$$

Combining (5) and (6), we can compute $\mathcal{S}$'s advantage in solving the $\textsf{D-BDH}$ challenge:

$$\begin{aligned} \mathrm{Adv}^{\mathsf{bdh}}_{\mathcal{S}}(\ell) &= \Pr(\delta' = \delta) - \tfrac{1}{2} \\ &= \tfrac{1}{2} \cdot (\Pr(\delta' = \delta \mid \delta = 0) + \Pr(\delta' = \delta \mid \delta = 1)) - \tfrac{1}{2} \\ &= \tfrac{1}{2} \cdot (\tfrac{1}{2} + \tfrac{1}{2} + \mathrm{Adv}^{\mathsf{sid}}_{\mathcal{H}}(\ell)) - \tfrac{1}{2} \\ &= \tfrac{1}{2} \cdot \mathrm{Adv}^{\mathsf{sid}}_{\mathcal{H}}(\ell). \end{aligned}$$

□

4.2 Security against an honest-but curious central authority

In order to show that the proposed scheme can tolerate an honest-but-curious central authority in the sense of Definition 3, we can use a similar argument as in the above proof of Theorem 1. It turns out that again there is a tight security reduction: Essentially, for the price of simulating the central authority and the attribute authorities, from an adversary $\mathcal{B}$ described in the game from Figure 3, we obtain a $\textsf{D-BDH}$ adversary whose advantage is half the advantage of $\mathcal{B}$.

Theorem 2. *Let $\mathcal{B}$ be a probabilistic polynomial time adversary against the protocol in Section 3 having a non-negligible advantage in the game in Figure 3. Then there is a probabilistic polynomial time algorithm $\mathcal{S}$ having a non-negligible advantage in solving the* D-BDH*-problem.*

Proof. As in the proof of Theorem 1, the input of the D-BDH adversary $\mathcal{S}$, which we have to derive, is a tuple of the form (2). Again we refer to $\mathcal{S}$ as the *simulator*, and to find δ, a simulation of $\mathcal{B}$ is run by $\mathcal{S}$. The individual phases of the game in Figure 3 are mimicked as follows:

Setup. The simulator uses the attribute authorities, users, thresholds and attribute sets specified by $\mathcal{B}$. For all corrupted authorities the simulator follows the original protocol specification. Moreover, as the central authority k_{CA} is honest-but-curious, the simulation of k_{CA} follows the original protocol specification also. In particular, σ and all the $s_{k,u}$-values ($k \in \mathcal{K} \cup \{k_{\mathsf{CA}}\}$) are chosen honestly. Let $\mathcal{K}_{\mathsf{hon}} \subseteq \mathcal{K}$ be the set of those attribute authorities that $\mathcal{B}$ specified as not being corrupted.

The simulator chooses one authority $\hat{k} \in \mathcal{K}_{\mathsf{hon}}$ uniformly at random. For $k \in \mathcal{K}_{\mathsf{hon}} \setminus \{\hat{k}\}$ the simulator generates k's public key as specified in the original protocol. For $\hat{k}$, the computation of the public value $e(P,P)^{r_{\hat{k}}}$ is modified. Namely, the latter value is computed as

$$e(\alpha P, \beta P) \cdot e(P,P)^{-\sum_{k \in \mathcal{K}\setminus\{\hat{k}\}} r_k} = e(P,P)^{\alpha\beta - \sum_{k \in \mathcal{K}\setminus\{\hat{k}\}} r_k}$$

with αP, βP being part of the D-BDH challenge. This implicitly fixes

$$r_{\hat{k}} := \alpha\beta - \sum_{k \in \mathcal{K}\setminus\{\hat{k}\}} r_k. \tag{7}$$

So for $\mathcal{B}$ the values learned at the end of the setup phase with overwhelming probability follow the same distribution as in the original game in Figure 3.

Challenge. Let $M_0, M_1 \in G_2$ be the challenge messages selected by $\mathcal{B}$, and let δ be the value to be found by the D-BDH adversary $\mathcal{S}$. Using a fair binary coin $\mu \leftarrow \{0,1\}$ and the last two components of the D-BDH challenge, the simulator hands the challenge ciphertext

$$\left(e(P,P)^{\delta\cdot\alpha\beta\gamma+(1-\delta)\cdot\eta} \cdot e(\gamma P, P)^{\sigma} \cdot M_\mu, \gamma P, [t_{k,a} \cdot \gamma P]_{a \in \mathcal{A}_C}\right) \tag{8}$$

for M_μ to $\mathcal{H}$. We consider both possible cases $\delta = 0$ and $\delta = 1$:

$\delta = 0$**:** Because of $e(P,P)^{\delta\cdot\alpha\beta\gamma+(1-\delta)\cdot\eta} = e(P,P)^{\eta}$ with a uniformly at random chosen $\eta \leftarrow \{0,\dots,p-1\}$, the challenge ciphertext contains no information on M_μ.

$\delta = 1$**:** We have that $e(P,P)^{\delta\cdot\alpha\beta\gamma+(1-\delta)\cdot\eta} = e(P,P)^{\alpha\beta\gamma}$, and Equation (7) yields $e(P,P)^{\alpha\beta\gamma} = e(P,P)^{\gamma\cdot\sum_{k\in\mathcal{K}} r_k}$. Hence the challenge ciphertext (8) becomes

$$\left(\left(\mathsf{pk}\cdot\prod\nolimits_{k\in\mathcal{K}} e(P,P)^{r_k}\right)^{\gamma}\cdot M_\mu, \gamma P, [\gamma\cdot t_{k,a}P]_{a\in\mathcal{A}_C}\right),$$

which is a valid encryption of M_μ.

Guess. Denote by μ' the output of $\mathcal{B}$. The output of the simulator $\mathcal{S}$ is given by

$$\delta' := \begin{cases} 1\ , \text{ if } \mu = \mu' \\ 0\ , \text{ if } \mu \neq \mu' \end{cases} .$$

In other words, $\mathcal{S}$ considers the last component of the $\mathsf{D\text{-}BDH}$ challenge to be $e(P,P)^{\alpha\beta\gamma}$ whenever $\mathcal{B}$ correctly identifies M_μ. With the same line of arguments as in the proof of Theorem 1, the advantage of $\mathcal{S}$ in solving the $\mathsf{D\text{-}BDH}$ challenge computes to

$$\mathrm{Adv}^{\mathsf{bdh}}_{\mathcal{S}}(\ell) = \tfrac{1}{2}\cdot\mathrm{Adv}^{\mathsf{ca}}_{\mathcal{B}}(\ell).$$

□

5 Conclusion

Building on the proposal for multi-authority based attribute based encryption from [4], we constructed a scheme where the central authority is no longer capable of decrypting arbitrary ciphertexts created within the system. In addition to showing security in the selective ID model, we showed that the proposed system can tolerate an honest-but-curious central authority. Since both Chase's scheme and the proposed scheme rely on the same hardness assumption, and have a comparable complexity, the new scheme seems a viable alternative to Chase's construction. However, since only the proposed method is capable of handling a curious yet honest central authority, the proposed scheme is recommended in applications where security against such a central authority is required.

References

1. D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 2004.
2. D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, 2001.
3. R. Canetti, S. Halevi, and J. Katz. A Forward-Secure Public-Key Encryption Scheme. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer-Verlag, 2003.
4. M. Chase. Multi-authority Attribute Based Encryption. In S.P. Vadhan, editor, *Theory of Cryptography – TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 515–534. Springer-Verlag, 2007.
5. C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In B. Honary, editor, *Cryptography and Coding, 8th IMA International Conference*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer-Verlag, 2001.
6. Y. Desmedt and J-J. Quisquater. Public-key Systems Based on the Difficulty of Tampering (Is there a difference between DES and RSA?). In A. M. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 111–117. Springer-Verlag, 1987.
7. A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer-Verlag, 2005.
8. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology – CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1985.
9. H. Tanaka. A Realization Scheme for the Identity-Based Cryptosystem. In C. Pomerance, editor, *Advances in Cryptology – CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 340–349. Springer-Verlag, 1988.
10. S. Tsujii and T. Itoh. An ID-Based Cryptosystem Based on the Discrete Logarithm Problem. *IEEE Journal on Selected Areas in Communications*, 7(4), May 1989.
11. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer-Verlag, 2005.