Direct Chosen-Ciphertext Secure Attribute-Based Key Encapsulations without Random Oracles

Johannes Blömer bloemer@upb.de
Gennadij Liske gennadij.liske@upb.de

Department of Computer Science, University of Paderborn, Germany*

October 8, 2013

Abstract. We present a new technique to realize attribute-based encryption (ABE) schemes secure in the standard model against chosen-ciphertext attacks (CCA-secure). Our approach is to extend certain concrete chosen-plaintext secure (CPA-secure) ABE schemes to achieve more efficient constructions than the known generic constructions of CCA-secure ABE schemes. We restrict ourselves to the construction of attribute-based key encapsulation mechanisms (KEMs) and present two concrete CCA-secure schemes: a key-policy attribute-based KEM that is based on Goyal's key-policy ABE and a ciphertext-policy attribute-based KEM that is based on Waters' ciphertext-policy ABE. To achieve our goals, we use an appropriate hash function and need to extend the public parameters and the ciphertexts of the underlying CPA-secure encryption schemes only by a single group element. Moreover, we use the same hardness assumptions as the underlying CPA-secure encryption schemes.

Keywords: attribute-based key encapsulation mechanism, attribute-based encryption, chosen-ciphertext security, bilinear maps

1 Introduction

Traditionally, encryption is viewed as a method of transmitting messages confidentially between a sender and a receiver, whereas nowadays encryption schemes are used for various applications. Consequently, conventional encryptions often do not satisfy additional requirements of these applications and need to be enhanced. An example is the realization of access control mechanisms for secure shared storage via encryption. The encrypted data in such a system is usually not meant for a single user but for a subgroup of users. However, every user should have their own secret key and data should be encrypted only once. To provide such a powerful access control of encrypted data, attribute-based encryption (ABE) schemes were introduced as a natural extension of public-key encryption (PKE) schemes.

ABE in their basic form were introduced by Sahai and Waters [SW05]. Later, Goyal et al. proposed two different types of ABE schemes [GPSW06]. In the so-called key-policy attribute-based encryption (KP-ABE), ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertext a user will

* This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre On-The-Fly Computing (SFB 901).

be able to decrypt. In the ciphertext-policy attribute-based encryption (CP-ABE), the keys are associated with attribute sets and the data is encrypted under access structures.

Since the work of Sahai and Waters, various extensions for ABEs have been considered. Some work focuses on extending the expressiveness of policies [GPSW06, OSW07, Wat11]. Other schemes consider the multi-authority setting [Cha07, CC09] or constructions based on lattices [ABV$^+$12, Boy13]. In [Wat09, LW10, LOS$^+$10], the authors present a first method to achieve adaptive security rather then selective security for ABE. This line of research was followed up in [LW11a, LW11b, LW12] and many further works.

In this paper, we focus on another important issue in the area of ABE. We consider the aspect of attaining security against chosen-ciphertext attacks (CCAs) in the standard model. CCA-security has emerged as the right security notion for encryption schemes. Most ABE schemes originally are only chosen plaintext secure (CPA-secure). Usually, the Fujisaki-Okamoto transformation [FO99] is proposed to achieve CCA-security for ABE schemes as well. However, the Fujisaki-Okamoto transformation is only secure in the random oracle model.

In the standard model, Goyal et al. [GPSW06] achieve CCA-security for their KP-ABE by a method based on the technique of Canetti et al. [BCHK07]. It has been proposed to apply this technique to several ABE constructions [GPSW06, CN07, Wat11]. The idea was generalized and extended by [YAHK11], where generic constructions for the CCA-secure KP-ABE and for the CCA-secure CP-ABE were presented. These constructions use one-time signatures as a building block in addition to a CPA-secure ABE scheme. First, the ciphertexts of the CPA-secure ABE are extended by the public key of the one-time signature scheme. Then, the one-time signature is applied to the resulting ciphertext. It is shown in [YAHK11] that under some mild conditions on the underlying CPA-secure ABE scheme this yields a CCA-secure ABE scheme.

Our contribution. In this paper we follow another approach and transfer the direct chosen-ciphertext techniques already known from public-key cryptography [BMW05, Kil06] and from identity-based cryptography [KG09] to attribute-based encryption.

Similar to [KG09] we restrict ourselves to the construction of key encapsulation mechanisms (KEMs) [CS03]. The construction of CCA-secure KEMs is conceptually easier than the construction of CCA-secure encryption schemes. As in the public-key and identity-based setting it is not hard to see that combining any CCA-secure attribute-based key encapsulation mechanisms (AB-KEMs) with a symmetric scheme (also called data encapsulation mechanism (DEM)) with appropriate security properties leads to a fully functional CCA-secure ABE scheme. For identity-based encryption schemes this was proven in [BFMLS08]. The proof can easily be adapted to ABE settings. For many practical reasons the modular KEMs/DEMs approach is preferable over ABE schemes, in particular for applications such as secure shared storage, which require the encryption of big data using public-key techniques.

To construct CCA-secure AB-KEMs, first, from the CPA-secure KP-ABE of [GPSW06] and from the CPA-secure CP-ABE of [Wat11], we derive AB-KEMs in the usual way. Then, we enhance these KEMs by the property called public verifiability of encapsulations [NMP$^+$12]. Achieving public verifiability is straightforward for the key-policy AB-KEM (KP-AB-KEM). For the ciphertext-policy AB-KEM (CP-AB-KEM) this is more involved. Next, we add some redundancy to the encapsulation to make it CCA-secure. As in the identity-based key encapsulation mechanism (IB-KEM) of [KG09], the redundancy consists of a single group element based on a hash of parts of the original encapsulation. In [KG09], target collision resistant hash functions were sufficient to prove CCA-security of their IB-KEM. Due to the rich internal structure in the attribute-based setting, we need cryptographically stronger hash functions, e.g. the universal one-way hash function (UOWHF). Altogether, we achieve CCA-secure AB-KEMs with the same hardness assumptions as the underlying CPA-secure ABE schemes. Note that in [CZF11], the authors use a construction similar to that of [KG09]. However, their scheme realizes only restricted access structures described by a single non-monotone AND gate.

The ABE schemes in [GPSW06] and [Wat11] are CPA-secure only in the selective-security model. Hence, for our schemes we can prove CCA-security only in this model as well. It turns out that achieving CCA-security for CP-AB-KEM is harder than to obtain CCA-security for KP-AB-KEM, therefore in this extended abstract we restrict ourselves to the more involved construction of CP-AB-KEM.

Organization. In Section 2, we present the basic concepts of ABE schemes including monotone span programs (MSPs), security definitions and security assumptions. We also discuss the main ingredients of our construction and prove some useful lemmata from linear algebra. In Section 3, based on Waters' construction, we present our CCA-secure CP-AB-KEM and its security proof. Due to space limitations, we restrict ourselves to the additional ingredients that allow us to achieve CCA-security.

2 Background

In this section we recall fundamental notions for attribute-based encryption schemes.

2.1 Access Structures and Monotone Span Programs (MSPs)

Definition 1. (Access Structure [Bei96]) *Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a set of parties, $\gamma, \beta \subseteq \mathcal{X}$. A collection $\mathbb{A} \subseteq 2^{\mathcal{X}}$ is monotone if $\gamma \in \mathbb{A}$ and $\gamma \subseteq \beta$ implies $\beta \in \mathbb{A}$. A monotone access structure is a monotone collection $\mathbb{A}$ of non-empty subsets of $\mathcal{X}$. The sets in $\mathbb{A}$ are called the authorized sets and the sets not in $\mathbb{A}$ are called the unauthorized sets.*

We always assume that access structures are not empty. Based on this definition we obtain:

Definition 2. (Monotone Span Program [Bei96]) *Let p be a prime and $\mathcal{X} = \{x_1, \ldots, x_n\}$ a set of parties. An* MSP *$\mathcal{M}$ over $\mathbb{Z}_p$ is a labeled matrix (M, ρ), where $M \in \mathbb{Z}_p^{l \times d}$ is a matrix and $\rho : \{1, \ldots, l\} \to \mathcal{X}$ is a labeling of rows of M with parties. The size of $\mathcal{M}$ is defined by the number of rows l.*

$\mathcal{M}$ accepts $\gamma \subseteq \mathcal{X}$ if vector $\boldsymbol{e}_1$ is in the span of the rows of M with labels in γ and rejects it otherwise. $\mathcal{M}$ realizes an access structure $\mathbb{A}$ (write $\mathcal{M}_{\mathbb{A}}$) if $\mathcal{M}$ accepts every authorized set and rejects every unauthorized set of $\mathbb{A}$.

Beimel proved in [Bei96] the equivalence of existence of linear secret sharing scheme (LSSS) for a monotone access structure $\mathbb{A}$ and the existence of monotone span program $\mathcal{M}_{\mathbb{A}} = (M, \rho)$ of the same size. In order to share a secret $s \in \mathbb{Z}_p$ using $\mathcal{M}_{\mathbb{A}}$, choose $v_2, \ldots, v_d \leftarrow \mathbb{Z}_p$, set $\boldsymbol{v} = (s, v_2, \ldots, v_d)$ and compute the vector $\boldsymbol{\lambda}$ of shares by $\boldsymbol{\lambda} := M \cdot \boldsymbol{v}$. The share λ_i belongs to party $\rho\left(i\right)$. For an authorized set $\gamma \in \mathbb{A}$ there exists a vector $\boldsymbol{w}$ so that $\boldsymbol{w} \cdot M = \boldsymbol{e}_1$ and $w_i = 0$ for all i with $\rho(i) \notin \gamma$. Such a vector can be computed efficiently and the parties in γ can reconstruct the secret by $\sum_{i \in \{1,\ldots,l\}, w_i \neq 0} w_i \cdot \lambda_i = \boldsymbol{w} \cdot M \cdot \boldsymbol{v} = s$. For an unauthorized set β such a vector does not exist and the parties in β get no information about the secret from their shares.

The following definition turns out to be helpful. Let $\mathcal{M}_{\mathbb{A}} = (M, \rho)$ be an MSP over $\mathbb{Z}_p$. Since $\mathbb{A} \neq \emptyset$, there exists a vector $\boldsymbol{z}$ with $\boldsymbol{z} \cdot M = \boldsymbol{e}_1$. The affine vector space V_M is defined as

$$V_M := \left\{ \boldsymbol{w} \in \mathbb{Z}_p^l \;\middle|\; \boldsymbol{w} \cdot M = \boldsymbol{e}_1 \right\}, \tag{1}$$

i.e. the vectors in V_M can be used to reconstruct the secret. Clearly, $V_M = \boldsymbol{z} + \ker\left(M^{tr}\right)$ and V_M can efficiently be computed given M.

Reduced echelon form of MSPs. Different MSPs can be used to realize the same access structure. This leads us to the major problem for the construction of CCA-secure CP-AB-KEMs, where the MSP is a part of the encapsulation. To solve this problem we will use the following special form of MSPs that is based on the reduced column echelon form of matrices. Moreover, we restrict ourselves to MSPs with injective labeling functions. As already observed and exploited in [Wat11], by using several distinct copies of parties, every MSP can be converted into MSP with an injective labeling function.

Definition 3. *A monotone span program $\mathcal{M} = (M, \rho)$ with $M \in \mathbb{Z}_p^{l \times d}$ and injective labeling function ρ is in* reduced echelon form *if:*

- *The columns of M are linearly independent, thus $d \leq l$.*
- *The rows are ordered according to some fixed order on the labels.*
- *The submatrix of M consisting of the last $d-1$ columns is in reduced column echelon form. [Sho06]*
- *The entries in the first column that correspond to the pivot elements in the other columns are zero.*

Using Gauss-Jordan elimination, every MSP with injective labeling function can be converted into reduced echelon form without changing the access structure.

Example 1. We use the notation of [LC10]. A boolean formula can be expressed as $(\phi_1, \ldots, \phi_n, t)$. The root node is a t of n threshold gate and its children are threshold gates of the same form or leaf nodes corresponding to parties. Let $\mathbb{A}$ be the access structure given by the set of satisfying assignments of $\phi = (B, (A, F, 2), (C, D, E, G, 3), 1)$. The following two MSPs over $\mathbb{Z}_{17}$ realize this access structure:

$$\begin{array}{cc} \begin{array}{c} B \\ A \\ F \\ C \\ D \\ E \\ G \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 2 & 4 \\ 1 & 0 & 3 & 9 \\ 1 & 0 & 4 & 16 \end{pmatrix} \end{array} \longrightarrow \begin{array}{cc} \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \end{array} & \left(\begin{array}{c|ccc|} \mathbf{0} & \mathbf{1} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \mathbf{0} & 0 & \mathbf{1} & 0 \\ \mathbf{0} & 0 & 0 & \mathbf{1} \\ 1 & 0 & 14 & 3 \\ 16 & 2 & 0 & 0 \\ 3 & 0 & 9 & 6 \end{array}\right) \end{array} .$$

The first MSP is constructed by the algorithm of [LC10]. Conversion of this MSP leads to the second MSP, which is in reduces echelon form.

The following general lemma about matrices is critical for our construction (see Appendix A for the proof).

Lemma 1. *Let p be a prime, $M = [m_1, \dots, m_d], N = [n_1, \dots, n_d] \in \mathbb{Z}_p^{l \times d}$ be matrices with* $\operatorname{span}(m_1, \dots, m_d) = \operatorname{span}(n_1, \dots, n_d)$. *Then, the reduced column echelon forms of M and N are equal.*

2.2 Attribute-Based Key Encapsulation Mechanisms (AB-KEMs)

As already mentioned in the introduction, we will restrict ourselves to the construction of AB-KEMs. Here, attributes play the role of parties. Following the usual terminology, from now one we call parties attributes.

Definition 4. *A* ciphertext-policy attribute-based key encapsulation mechanism Π *over an attribute universe $\mathcal{U}$ for symmetric key space $\mathbb{K}$ consists of four probabilistic polynomial time (ppt) algorithms:*

Setup: *The setup algorithm gets as input the security parameter 1^η and computes the public parameters and the master secret key:* $(\text{params}, \text{msk}) \leftarrow \text{Setup}(1^\eta)$. *(The public parameters will be implicitly used by all the other algorithms.)*

KeyGen: *The key generation algorithm on input $\gamma \subseteq \mathcal{U}$ and* msk *computes the secret key: $sk_\gamma \leftarrow \text{KeyGen}_{\text{msk}}(\gamma)$.*

Encaps: *The encapsulation algorithm gets as input an access structure $\mathbb{A}$ over $\mathcal{U}$, generates a key $k \leftarrow \mathbb{K}$ and its encapsulation: $(k, E_{\mathbb{A}}) \leftarrow \text{Encaps}(\mathbb{A})$.*

Decaps: *The decapsulation algorithms on input sk_γ and $E_{\mathbb{A}}$ recovers the key $k := \text{Decaps}_{sk_\gamma}(E_{\mathbb{A}})$.*

Correctness: *We require that for every access structure $\mathbb{A}$ over $\mathcal{U}$, every set of attributes $\gamma \in \mathbb{A}$, every $(\text{params}, \text{msk}) \leftarrow \text{Setup}(1^\eta)$, $sk_\gamma \leftarrow \text{KeyGen}_{\text{msk}}(\gamma)$ and every $(k, E_{\mathbb{A}}) \leftarrow \text{Encaps}(\mathbb{A})$ it holds that $\text{Decaps}_{sk_\gamma}(E_{\mathbb{A}}) = k$.*

For KP-AB-KEM the roles of access structures and sets of attributes are reversed.

Public verifiability of encapsulations will be an important ingredient of our constructions and security proofs. Let $A(x; \tau)$ denote the execution of a ppt algorithm A on input x with random bits τ.

Definition 5. *A ciphertext-policy attribute-based key encapsulation mechanism Π has* publicly verifiable encapsulations *if there exists a ppt algorithm* Verify *that on input the public parameters* params *and a possibly malformed encapsulation $E_{\mathbb{A}}$ accepts $E_{\mathbb{A}}$ if and only if there exist random bits τ such that* $\mathrm{Encaps}(\mathbb{A};\tau) = (k, E_{\mathbb{A}})$*. Encapsulations produced by the algorithm* Encaps *are called* consistent*.*

Security model for CP-AB-KEM. In [YAHK11, CZF11], the authors use similar CCA-security definitions of ABE adapted from the context of identity-based encryption (IBE). We extend these definitions and give the adversary potentially additional power through more specific decapsulation queries (decryption queries for ABE) as explained below. Note, however, that the constructions in [YAHK11, CZF11] also satisfy this stronger security notion.

In ciphertext-policy attribute-based settings users with the same attributes will have different private keys. Hence, we should model this in our security definition and allow the adversary to force the challenger to generate several keys for the same set of attributes. Then, in a decapsulation query the adversary should be allowed to specify which of the generated keys will be used for decapsulation. In the following experiment we formalize this by so-called covered key generation queries.

The experiment $\text{sCP-AB-KEM}^{\text{aCCA}}_{\mathcal{A},\Pi}(\eta)$ for a ciphertext-policy attribute-based key encapsulation mechanism Π and a ppt adversary $\mathcal{A}$ is as follows:

Init: $\mathcal{A}$ on input 1^η commits to an access structure $\mathbb{A}^*$. Challenger $\mathcal{C}$ initializes an empty list L of secret keys.

Setup: $\mathcal{C}$ generates $(\text{params}, \text{msk}) \leftarrow \text{Setup}(1^\eta)$ and gives params to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ adaptively queries the key generation $\text{KeyGen}(\gamma)$ for $\gamma \notin \mathbb{A}^*$, covered key generation $\text{CoveredKeyGen}(\gamma)$ for $\gamma \in \mathbb{A}^*$ and decapsulation queries $\text{Decaps}(E_{\mathbb{A}}, i)$ for arbitrary $E_{\mathbb{A}}$ and $i \in \mathbb{N}$, $i \leq |L|$. $\mathcal{C}$ replies to the queries as follows:

- $\text{KeyGen}(\gamma)$: Output $sk_\gamma \leftarrow \text{KeyGen}_{\text{msk}}(\gamma)$.
- $\text{CoveredKeyGen}(\gamma)$: Generate $sk_\gamma \leftarrow \text{KeyGen}_{\text{msk}}(\gamma)$ and add $(|L|+1, sk_\gamma)$ to L. $\mathcal{C}$ returns no output.
- $\text{Decaps}(E_{\mathbb{A}}, j)$: Let $(j, sk_\gamma) \in L$. Output $k := \text{Decaps}_{sk_\gamma}(E_{\mathbb{A}})$.

Challenge: $\mathcal{C}$ runs the encapsulation algorithm $(k_1, E^*_{\mathbb{A}^*}) \leftarrow \text{Encaps}(\mathbb{A}^*)$, chooses $k_0 \leftarrow \mathbb{K}$, $\nu \leftarrow \{0,1\}$, sets $k^* := k_\nu$ and outputs the challenge $(k^*, E^*_{\mathbb{A}^*})$.

Phase 2: Similar to Phase 1 under the restriction that decapsulation queries on $E^*_{\mathbb{A}^*}$ are not allowed.

Guess: $\mathcal{A}$ outputs a guess ν' and the output of the experiment is 1 iff $\nu' = \nu$.

Definition 6. *A CP-AB-KEM Π is selectively secure against adaptive chosen-ciphertext attacks if for every ppt algorithm $\mathcal{A}$ there exists a negligible function* negl *such that it holds* $\Pr\left[\text{sCP-AB-KEM}^{\text{aCCA}}_{\mathcal{A},\Pi}(\eta) = 1\right] \leq 1/2 + \text{negl}(\eta)$*.*

2.3 Security Assumptions

Our constructions use symmetric bilinear maps (see e.g. [BF03]). We use standard terminology for bilinear maps and cryptographic assumption. Let $\mathcal{G}$ be an algorithm that generates bilinear groups $(p, \mathbb{G}, \mathbb{G}_T, e, g)$, where p is a prime, $\mathbb{G}$ and $\mathbb{G}_T$ are groups of order p, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an admissible bilinear map and $g \in \mathbb{G}$ is a generator.

The decisional Bilinear Diffie-Hellman experiment $\mathrm{BDH}_{\mathcal{A},\mathcal{G}}(\eta)$ is as follows:

- Challenger $\mathcal{C}$ generates $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^\eta)$ and gives it to adversary $\mathcal{A}$.
- $\mathcal{C}$ chooses $a, b, c, z \leftarrow \mathbb{Z}_p$, $\nu \leftarrow \{0, 1\}$. $\mathcal{A}$ receives $\left(g^a, g^b, g^c, Z\right)$, where $Z := e(g, g)^z$ if $\nu = 0$ and $Z := e(g, g)^{abc}$ if $\nu = 1$.
- $\mathcal{A}$ outputs a guess ν' and the output of the experiment is 1 iff $\nu' = \nu$.

Definition 7. *The* decisional Bilinear Diffie-Hellman problem *relatively to $\mathcal{G}$ is hard if for every ppt algorithm $\mathcal{A}$ there exists a negligible function* negl *such that it holds* $\Pr\left[\mathrm{BDH}_{\mathcal{A},\mathcal{G}}(\eta) = 1\right] \le 1/2 + \mathrm{negl}(\eta)$.

The q-Bilinear Diffie-Hellman Exponent experiment $\text{q-BDHE}_{\mathcal{A},\mathcal{G}}(\eta)$ is as follows:

- Challenger $\mathcal{C}$ generates $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^\eta)$ and gives it to adversary $\mathcal{A}$.
- $\mathcal{C}$ chooses $a, s, z \leftarrow \mathbb{Z}_p$, $\nu \leftarrow \{0, 1\}$. $\mathcal{A}$ receives $\left(g^s, g^a, \dots g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2\cdot q}}, Z\right)$, where $Z = e(g, g)^z$ if $\nu = 0$ and $Z = e(g, g)^{s \cdot a^{q+1}}$ if $\nu = 1$.
- $\mathcal{A}$ outputs a guess ν' and the output of the experiment is 1 iff $\nu' = \nu$.

Definition 8. *The* q-Bilinear Diffie-Hellman Exponent problem *relatively to $\mathcal{G}$ is hard if for every ppt algorithm $\mathcal{A}$ there exists a negligible function* negl *such that it holds* $\Pr\left[\text{q-BDHE}_{\mathcal{A},\mathcal{G}}(\eta) = 1\right] \le 1/2 + \mathrm{negl}(\eta)$.

2.4 Hash Functions

Our constructions require universal one-way hash functions (UOWHFs) as introduced in [NY89]. Target collision resistant hash functions presented in [CS03] and used by [KG09] are not sufficient for our construction, since the input of the hash function will depend on the choices of the adversary. In practice, both types of hash functions are instantiated by dedicated cryptographic hash function like SHA-2 (cf. [CS03]).

Definition 9. *(cf. [Gol04]) Let $\mathcal{UOWHF} = \left\{h_s : \{0,1\}^* \to \{0,1\}^{l(|s|)}\right\}_{s \in \{0,1\}^*}$ with $l : \mathbb{N} \to \mathbb{N}$ be a collection of efficiently computable keyed functions. $\mathcal{UOWHF}$ is called a family of universal one-way hash functions if there exists a ppt algorithm I such that for all ppt adversaries $\mathcal{A}$ the probability to win the following game is negligible in η:*

- *$\mathcal{A}$ on input 1^η outputs x.*
- *$\mathcal{A}$ is given $s \leftarrow I(1^\eta)$.*
- *$\mathcal{A}$ outputs x' and wins the game if $x' \neq x$ but $h_s(x') = h_s(x)$.*

In our constructions we need to hash tuples consisting of a natural number of certain length and a bounded number of group elements. Although we require an injective encoding of such tuples through bit strings, we will not explicitly mention this in our constructions.

2.5 Useful Lemmata

Next we present two lemmata that will be useful to obtain public verifiability of our CP-AB-KEM. See Appendix A for the proofs.

Lemma 2. *Let p be a prime, $M \in \mathbb{Z}_p^{l \times d}$ with* $\ker\left(M^{tr}\right) = \operatorname{span}\left(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\right) \subseteq \mathbb{Z}_p^l$, *then $\boldsymbol{\lambda} \in \operatorname{im}(M)$ if and only if for all $j \in \{1, \ldots, k\}$ it holds $\boldsymbol{u}_j \cdot \boldsymbol{\lambda} = 0$.*

Lemma 3. *Let $\mathbb{G}$ be a group of prime order p, $h \in \mathbb{G}$ a generator and $s, \lambda_1, \ldots, \lambda_l \in \mathbb{Z}_p$. Let $M \in \mathbb{Z}_p^{l \times d}$ be a matrix with $\boldsymbol{e}_1 \in \operatorname{im}\left(M^{tr}\right)$ and* $\ker\left(M^{tr}\right) = \operatorname{span}\left(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\right)$. *Let $\boldsymbol{w}$ be arbitrary with $\boldsymbol{w} \cdot M = \boldsymbol{e}_1$. The following statements are equivalent:*

1. $\prod_{i=1}^{l} \left(h^{\lambda_i}\right)^{w_i} = h^s \wedge \forall j \in \{1, \ldots, k\} : \prod_{i=1}^{l} \left(h^{\lambda_i}\right)^{u_{j,i}} = 1.$
2. $\boldsymbol{w} \cdot \boldsymbol{\lambda} = s \wedge \forall j \in \{1, \ldots, k\} : \boldsymbol{u}_j \cdot \boldsymbol{\lambda} = 0.$
3. *There exists a vector $\boldsymbol{b}$ with $M \cdot \boldsymbol{b} = \boldsymbol{\lambda}$ and $b_1 = s$.*

3 Ciphertext-Policy Attribute-Based Key Encapsulation

In this section we present our construction of a CP-AB-KEM, selectively secure against adaptive chosen-ciphertext attacks. Our starting point for this construction is the large universe CP-ABE of [Wat11]. As in this construction we restrict ourselves to MSPs with injective labeling functions. The generalization is as described in Section 5 of [Wat11].

In the following, constants Attr_{max} and l_{max} specify the maximal number of attributes per key and the maximal size of supported MSPs, respectively.

Setup(1^η, Attr_{max}, l_{max}) Generate bilinear groups $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^\eta)$ and elements $\alpha \leftarrow \mathbb{Z}_p$, $g_1, g_3 \leftarrow \mathbb{G}$. Set $\mathcal{U} := \mathbb{Z}_p$, $Y := e(g,g)^\alpha$ and choose a universal one-way hash function UHF $\leftarrow \mathcal{UOWHF}$ with appropriate injective encoding such that:

$$\mathrm{UHF} : \{0,1\}^{\lfloor \log(l_{max}) \rfloor + 1} \times \mathbb{G}_T^{m \leq l_{max}+1} \to \mathbb{Z}_p.$$

Set $n := l_{max} + \mathit{Attr}_{max} - 1$ and choose $\{H_i \leftarrow \mathbb{G}\}_{i \in \{0,\ldots,n\}}$. As in [Wat11], the elements H_i define a publicly computable function:

$$\begin{aligned} H : \mathcal{U} \to & \quad \mathbb{G} \\ x \mapsto & \prod\nolimits_{i \in \{0,\ldots,n\}} H_i^{\Delta_{i,\{0,\ldots,n\}}(x)} \end{aligned}$$

where $\Delta_{i,\{0,\ldots,n\}}(x)$ are the Lagrange interpolation polynomials. Hence, $H(x) = g^{h(x)}$ for some polynomial $h(x)$ of degree at most n. Since the H_i are chosen uniformly and independently at random, h is also chosen uniformly at random.

The master secret is msk $:= \alpha$ and the public parameters are

$$\mathrm{params} := \Big(\mathit{Attr}_{max}, l_{max}, (p, \mathbb{G}, \mathbb{G}_T, e, g), g_1, g_3, Y, \{H_i\}_{i \in \{0,\ldots,n\}}, \mathrm{UHF}\Big).$$

The key space for the DEM is $\mathbb{K} = \mathbb{G}_T$.

Compared to the large universe construction of [Wat11], we add the group element g_3 and the hash function UHF to public parameters.

KeyGen$_{\mathrm{msk}}$**(**γ**)** with $\gamma \subseteq \mathcal{U}$, $|\gamma| \leq Attr_{max}$. Choose $r \leftarrow \mathbb{Z}_p$ and set $D := g^{\alpha} \cdot g_1^r$, $D' := g^r$ and $\{D_x := H(x)^r\}_{x \in \gamma}$. The secret key for γ is $sk_\gamma := \left(\gamma, D, D', \{D_x\}_{x\in\gamma}\right)$. This algorithm is as in [Wat11].

Encaps($\mathcal{M}$**)** with $\mathcal{M} = (M, \rho)$ in reduced echelon form, $M \in \mathbb{Z}_p^{l \times d}$, $l \leq l_{max}$. Choose $s, b_2, \ldots, b_d \leftarrow \mathbb{Z}_p$ and set $E' := g^s$, $\boldsymbol{b} := (s, b_2, \ldots, b_d)$. Compute the vector $\boldsymbol{\lambda} \in \mathbb{Z}_p^l$ of shares by $\boldsymbol{\lambda} := M \cdot \boldsymbol{b}$ and set $\left\{E_i := g_1^{\lambda_i} \cdot H\left(\rho(i)\right)^{-s}\right\}_{i\in\{1,\ldots,l\}}$. Compute $t :=$ UHF $\left(d, e(g, g_1)^s, e(g, g_1)^{\lambda_1}, \ldots, e(g, g_1)^{\lambda_l}\right)$ and set $E'' := \left(g_1^t \cdot g_3\right)^s$. The symmetric key is $K = Y^s$ and the encapsulation of K is $E_{\mathcal{M}} := \left(\mathcal{M}, E', \{E_i\}_{i\in\{1,\ldots,l\}}, E''\right)$. Compared to the original scheme we only add the group element E''.

Decaps$_{sk_\gamma}$**(**$E_{\mathcal{M}}$**)** with $\mathcal{M} = (M, \rho)$ in reduced echelon form. Compute a reconstruction vector $\boldsymbol{w} \in V_M$ with $w_i = 0$ for all $i \in \{0,1\}^l$ with $\rho(i) \notin \gamma$. Reject, if such a vector does not exist. Construct ker $\left(M^{tr}\right) =$ span $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$. Compute

$$\left\{X_i := e\left(E_i, g\right) \cdot e\left(H\left(\rho(i)\right), E'\right)\right\}_{i\in\{1,\ldots,l\}}$$

and $t' :=$ UHF $(d, e(E', g_1), X_1, \ldots, X_l)$. Reject, if one of the following consistency checks fails

$$\prod_{i=1}^{l} X_i^{w_i} \stackrel{?}{=} e(E', g_1), \tag{2}$$

$$\forall j \in \{1, \ldots, k\} : \prod_{i=1}^{l} X_i^{u_{j,i}} \stackrel{?}{=} 1, \tag{3}$$

$$e(E', g_1^{t'} \cdot g_3) \stackrel{?}{=} e(g, E''). \tag{4}$$

Using $\boldsymbol{w}$ from above, compute $Z_i := e\left(E_i, D'\right) \cdot e\left(D_{\rho(i)}, E'\right)$ for all i with $w_i \neq 0$ and output the key $K := e\left(E', D\right) \cdot \prod_{i\in\{0,\ldots,l\}, w_i \neq 0} Z_i^{-w_i}$.

Compared to Waters CP-ABE, we only add the consistency checks, whereas the reconstruction works as before. Furthermore, we show later a more efficient version of the tests in (2) and (3).

Lemma 4. *$E_{\mathcal{M}}$ passes the tests in (2), (3) and (4) if and only if it is consistent (see Definition 5). Furthermore, these tests implicitly define an algorithm* Verify$_{\mathrm{params}}$ $(E_{\mathcal{M}})$ *as required by Definition 5.*

Proof. Let $E_{\mathcal{M}} = \left(\mathcal{M}, E', \{E_i\}_{i\in\{1,\ldots,l\}}, E''\right)$ be a possibly not consistent encapsulation with $\mathcal{M} = (M, \rho)$ in reduced echelon form and $E', E_1, \ldots, E_l, E'' \in \mathbb{G}$ (this can be checked efficiently). Notice, that tests in (2), (3) and (4) do not use any parts of the secret key. Furthermore, given $E_{\mathcal{M}}$ and without using secret key we may directly compute $V_M = \boldsymbol{w} +$ span $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$. It follows that anybody can perform these tests.

By Definition 5 the encapsulation is consistent, if and only if there exists a $\boldsymbol{b} \in \mathbb{Z}_p^d$ such that the encapsulation algorithm with random choices $\boldsymbol{b}$ produces $E_{\mathcal{M}}$. First note that because of the prime order of $\mathbb{G}$, the element E' in $E_{\mathcal{M}}$ uniquely defines $r \in \mathbb{Z}_p$ such that $E' = g^r$. Given r, the element E'' uniquely defines $t \in \mathbb{Z}_p$ such that $E'' = \left(g_1^t \cdot g_3\right)^r$. Finally, the exponent r, the labeling function ρ, and the elements E_i in $E_{\mathcal{M}}$ uniquely define a vector $\boldsymbol{\lambda} \in \mathbb{Z}_p^l$ such that for all i it holds $E_i = g_1^{\lambda_i} \cdot H\left(\rho\left(i\right)\right)^{-r}$.

By the construction of X_i we have

$$X_i = e\left(E_i, g\right) \cdot e\left(H\left(\rho(i)\right), E'\right) = e\left(g, g_1\right)^{\lambda_i}$$

for all i. Lemma 3 applied to the group $\mathbb{G}_T$ and generator $e\left(g, g_1\right) \in \mathbb{G}_T$ implies the existence of a preimage $\boldsymbol{b}$ of $\boldsymbol{\lambda}$ with $b_1 = r$ if and only if $E_{\mathcal{M}}$ passes (2) and (3). Encapsulation $E_{\mathcal{M}}$ passes the test in (4) if and only if $t = t'$, which implies

$$t = t' = \mathrm{UHF}\left(d, e(g, g_1)^r, e(g, g_1)^{\lambda_1}, \ldots, e(g, g_1)^{\lambda_l}\right).$$

Hence, t' is the correct hash value for $E_{\mathcal{M}}$. The lemma follows. □

Correctness of the scheme. Every consistent encapsulation passes the tests in (2), (3) and (4) by Lemma 4. Therefore, correctness follows directly from the correctness of the original ABE.

Notice, that the consistency checks are essential for our proof. The fact that these checks can be performed publicly can be seen as a useful byproduct. Such a property has been exploited in the context of public-key cryptography and in the context of identity-based cryptography [NMP$^+$12].

3.1 Security Proof

Our construction is based on the scheme of [Wat11]. In the security proof given below, we will concentrate on the additional arguments necessary to achieve CCA-security.

Theorem 1. *Assume $\mathcal{UOWHF}$ is a family of universal one-way hash functions. Let $Attr_{max}$ and l_{max} be appropriate constants defining the maximal number of attributes per key and the maximal size of supported MSPs respectively. Under the q-BDHE assumption relative to a group generator $\mathcal{G}$, our ciphertext-policy attribute-based key encapsulation mechanism is selectively secure against adaptive chosen-ciphertext attacks, where the challenge matrix is of size $l^* \times d^*$ with $d^* + Attr_{max} \leq q$ and $d^* \leq l^* \leq l_{max}$.*

Proof. We only need to prove the security of our construction for MSPs with injective labeling functions. The generalization to MSPs with non-injective labeling function is as in [Wat11].

Let $\mathcal{A}$ be an adversary against the scheme. We will construct an algorithm $\mathcal{B}$ which simulates $\mathcal{A}$, answers $\mathcal{A}$'s queries and uses $\mathcal{A}$'s output to win the q-BDHE (η) experiment.

$\mathcal{B}$ is given a q-BDHE challenge $\left(g^s, g^a, \ldots, g^{a^q}, g^{a^{q+2}}, \ldots, g^{a^{2q}}, Z\right)$ along with the bilinear group description $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}\left(1^\eta\right)$ from the q-BDHE challenger $\mathcal{C}$.

Init: $\mathcal{A}$ on input 1^η and $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ commits to an MSP $\mathcal{M}^* = (M^*, \rho^*)$ over $\mathbb{Z}_p$. Let $M^* \in \mathbb{Z}_p^{l^* \times d^*}$ in reduced echelon form such that $d^* + Attr_{max} \leq q$ and $l^* \leq l_{max}$. We define

$$\gamma^* := \{x \in \mathcal{U} | \exists i : \rho^*(i) = x\}. \tag{5}$$

This is the set of attributes appearing in the challenge MSP.

Setup: Except for the new element g_3, $\mathcal{B}$ generates public parameters as in the proof of [Wat11]. We briefly recall Waters' construction.

$\mathcal{B}$ sets $n := l_{max} + Attr_{max} - 1$, $g_1 := g^a$. Furthermore, $\mathcal{B}$ chooses $\alpha' \leftarrow \mathbb{Z}_p$ and sets $Y := e(g,g)^{\alpha'} \cdot e\left(g^a, g^{a^q}\right)$. Hence, the master secret is implicitly $\alpha = \alpha' + a^{q+1}$. Next, $\mathcal{B}$ chooses $d^* + Attr_{max} + 1$ polynomials $\{p_i(x)\}_{i \in \{0,\dots,d^*+Attr_{max}\}}$ of degree n as follows:

- Choose $p_0(x)$ uniformly at random.
- Choose $p_1(x), \dots, p_{d^*}(x)$ randomly with $p_j(x_i) = m^*_{i,j}$ for all $x_i = \rho^*(i)$.
- Choose $p_{d^*+1}(x), \dots, p_{d^*+Attr_{max}}(x)$ randomly with $p_j(x_i) = 0$ for all $x_i \in \gamma^*$.

At last, $\mathcal{B}$ sets $H_i := \prod_{j=0}^{d^*+Attr_{max}} \left(g^{a^j}\right)^{p_j(i)}$ for all $i \in \{0, \dots, n\}$. Hence, the corresponding polynomial $h(x)$ is implicitly set to $h(x) = p_0(x) + \sum_{j=1}^{d^*+Attr_{max}} a^j \cdot p_j(x)$.

Next we show how $\mathcal{B}$ simulates g_3. The idea is basically the same as in [KG09]. The unknown exponent s from the q-BDHE-challenge will be used as the random exponent of the challenge encapsulation $E^*_{\mathcal{M}^*}$. In particular, $E''^* = (g_1^s \cdot g_3)^{t^*}$ depends on s and the corresponding hash value t^*. Without a specific choice of g_3, $\mathcal{B}$ would not be able to generate E''^* correctly. More precisely, $\mathcal{B}$ will set g_3 such that E''^* depends only on the value t^* rather than on s. Since $\mathcal{A}$'s challenge in the simulation of [Wat11] is independent of the queries made by adversary in Phase 1, $\mathcal{B}$ can make this choice already in the Setup-Phase.

Consequently, $\mathcal{B}$ sets $E'^* := g^s$, chooses $b'_2, \dots, b'_{d^*} \leftarrow \mathbb{Z}_p$ and computes for all i the elements $E^*_i := g_1^{\sum_{j=2}^{d^*} m^*_{i,j} \cdot b'_j} \cdot (g^s)^{-p_0(\rho^*(i))}$. This is as in [Wat11]. Next, $\mathcal{B}$ computes X^*_i as in the decapsulation algorithm from these elements and chooses UHF $\leftarrow \mathcal{UOWHF}$. Using E'^* and X^*_i, $\mathcal{B}$ computes t^* also as in the decapsulation algorithm. Hence, t^* is the correct hash value for the challenge. $\mathcal{B}$ chooses $c' \leftarrow \mathbb{Z}_p$ and sets $g_3 := g^{c'} \cdot (g^a)^{-t^*} = g^{c' - a \cdot t^*}$.

The element g_3 is uniformly distributed by the choice of c'. Together with the choice of UHF and the arguments from [Wat11], this shows that $\mathcal{B}$ gives $\mathcal{A}$ correctly distributed public parameters.

Challenge: In addition to the already defined $E'^* := g^s$ and $\{E^*_i\}_{i \in \{1,\dots,l^*\}}$, $\mathcal{B}$ sets $E''^* := (g^s)^{c'}$ and $K^* := Z \cdot e\left(g^s, g^{\alpha'}\right)$. $\mathcal{B}$ outputs the challenge $(E^*_{\mathcal{M}^*}, K^*)$ with $E^*_{\mathcal{M}^*} := \left(\mathcal{M}^*, E'^*, \{E^*_i\}_{i \in \{1,\dots,l^*\}}, E''^*\right)$.

By the definition of g_3 we have:

$$\left(g_1^{t^*} \cdot g_3\right)^s = \left(g^{a \cdot t^*} \cdot g^{c' - a \cdot t^*}\right)^s = (g^s)^{c'} = E''^*.$$

Hence, E''^* is correctly formed. As shown in [Wat11], E'^* and $\{E_i^*\}_{i\in\{1,\ldots,l^*\}}$ implicitly define

$$\boldsymbol{b}^* = \left(s, b_2' + s\cdot a, b_3' + s\cdot a^2, \ldots, b_{d^*}' + s\cdot a^{d^*-1}\right). \tag{6}$$

By the choice of $\boldsymbol{b}^*$ the challenge $E_{\mathcal{M}^*}^*$ is correctly distributed.

In the case of $Z = e(g,g)^{s\cdot a^{q+1}}$, we get

$$K^* := e(g,g)^{s\cdot a^{q+1}} \cdot e\left(g^s, g^{\alpha'}\right) = e(g,g)^{\left(\alpha'+a^{q+1}\right)\cdot s} = Y^s.$$

Thus, $E_{\mathcal{M}^*}^*$ is the correct encapsulation of K^*. In the other case, by the choice of Z the key K^* is distributed uniformly and independently from $E_{\mathcal{M}^*}^*$.

Phase 1 and 2: $\mathcal{B}$ answers the queries as follows.

KeyGen (γ): $\mathcal{B}$ answers the key generation queries as in the original construction of [Wat11]. Hence, the correctness follows as in [Wat11].

CoveredKeyGen (γ): $\mathcal{B}$ adds $(|L|+1, \gamma)$ to the initially empty list L and returns. Note that unlike the covered key generation queries in the definition of sCP-AB-KEM$_{\mathcal{A},\Pi}^{\text{aCCA}}(\eta)$ experiment on page 6, $\mathcal{B}$ only stores the set of attributes γ. This is sufficient to answer decapsulation queries correctly, since the consistency tests ensure that the output of the decapsulation algorithm depends only on γ and not on a particular secret key sk_γ.

Decapsulate $(E_{\mathcal{M}}, j)$: Let γ be the j-th entry in L. $\mathcal{B}$ rejects, if the consistency checks in (2), (3) or (4) fail or γ is not accepted by $\mathcal{M}$, and thus γ is unauthorized.

If $E_{\mathcal{M}} = E_{\mathcal{M}^*}^*$ in Phase 1, before the adversary has seen the challenge, the query is valid, but $\mathcal{B}$ cannot win the game any more and aborts. We call this event $Abort_1$. In Phase 2 such a query is not allowed.

Now, we handle the case that γ is authorized, $E_{\mathcal{M}} \neq E_{\mathcal{M}^*}^*$ and $E_{\mathcal{M}}$ is consistent. Let $E_{\mathcal{M}} = \left((M,\rho), E', \{E_i\}_{i\in\{1,\ldots,l\}}, E''\right)$, $M \in \mathbb{Z}_p^{l\times d}$ in reduced echelon form, $E' = g^r$, $E_i = g_1^{\lambda_i} \cdot H(\rho(i))^{-r}$ for all i and $E'' = \left(g_1^t \cdot g_3\right)^r$. Note that t and t^* can be computed given $E_{\mathcal{M}}$ and $E_{\mathcal{M}^*}$ as shown in the proof of Lemma 4. We consider the following cases separately:

1. $t \neq t^*$. $\mathcal{B}$ computes

$$\left(E'' \cdot \left(E'\right)^{-c'}\right)^{(t-t^*)^{-1}} = \left(\left(g_1^t \cdot g^{c'-a\cdot t^*}\right)^r \cdot \left(g^r\right)^{-c'}\right)^{(t-t^*)^{-1}} = g^{a\cdot r}.$$

Using this value $\mathcal{B}$ answers the decapsulation query correctly by

$$K := e\left(E', g\right)^{\alpha'} \cdot e\left(g^{a^q}, g^{a\cdot r}\right) = e\left(g,g\right)^{\left(\alpha'+a^{q+1}\right)\cdot r} = Y^r.$$

2. $t = t^*$ and the inputs of the hash function UHF for $E_{\mathcal{M}}$ and $E_{\mathcal{M}^*}^*$ are different. This implies a collision for UHF. Abort the simulation. We call this event $Abort_2$.
3. $t = t^*$ and the inputs of the hash function UHF for $E_{\mathcal{M}}$ and $E_{\mathcal{M}^*}^*$ are equal. That is, $d = d^*$, $e\left(E', g_1\right) = e\left(E'^*, g_1\right)$, $l = l^*$ (implicitly by the number of E_i's) and $X_i = X_i^*$ for all $i \in \{1, \ldots l^*\}$. This immediately implies $g^r = E' = E'^* = g^s$. From $s = r$ and $t = t^*$, we deduce $E'' = E''^*$. Consider three subcases:

(a) $\rho \neq \rho^*$. Since both functions are injective, $l = l^*$ and since the rows are ordered, there exists an index j such that $\rho(j) = \hat{x}$, $\hat{x} \notin \gamma^*$ (see (5) for the definition of γ^*). Thus $\hat{x} \neq x_j^* = \rho^*(j)$. The equality $X_j = X_j^*$ implies $\lambda_j = \lambda_j^*$ and $E_j = g_1^{\lambda_j} \cdot H\left(\hat{x}\right)^{-s} = g_1^{\lambda_j^*} \cdot H\left(\hat{x}\right)^{-s}$. Next, $\mathcal{B}$ computes:

$$\begin{aligned} E_j \cdot \left(E_j^*\right)^{-1} &= g^{s\left(h\left(x_j^*\right)-h(\hat{x})\right)} \\ &= g^{s\cdot\sum_{i=0}^{d^*+Attr_{max}} a^i\cdot\left(p_i\left(x_j^*\right)-p_i(\hat{x})\right)}. \end{aligned}$$

Let k be the maximal number $0 < k \leq d^* + Attr_{max} \leq q$ such that $p_k\left(x_j^*\right) - p_k\left(\hat{x}\right) \neq 0$. Then

$$\begin{aligned} e\left(E_j \cdot \left(E_j^*\right)^{-1}, g^{a^{q+1-k}}\right) = &\prod_{i=0}^{k-1} e\left(g^s, g^{a^{q+1+i-k}}\right)^{p_i\left(x_j^*\right)-p_i(\hat{x})} \\ &\cdot e(g,g)^{s\cdot a^{q+1}\cdot\left(p_k(x_j^*)-p_k(\hat{x})\right)}. \end{aligned}$$

From this, $\mathcal{B}$ is able to extract the value $e(g,g)^{s\cdot a^{q+1}}$ and solve its challenge directly. The probability that k does not exist for $\hat{x} \notin \gamma^*$ and $x_j^* \in \gamma^*$, is given by

$$\Pr\left[\forall i \in \{1, \ldots, d^* + Attr_{max}\} : p_i\left(\hat{x}\right) = p_i\left(x_i^*\right)\right],$$

where the probability is over the random choices of polynomials p_i. Since $\hat{x} \notin \gamma^*$, the value $p_i(\hat{x})$ is distributed uniformly and independently of $p_i(x_j^*)$. This shows that the probability is negligible.

(b) $\rho = \rho^*$ and there exists $\boldsymbol{z} \in V_M$, $\boldsymbol{z} \notin V_{M^*}$ (see (1) for the definition of V_M). Since $\rho = \rho^*$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$ we obtain $E_i = E_i^*$ for all i. Hence, all the elements of $E_{\mathcal{M}}$ and $E_{\mathcal{M}^*}^*$ are equal, except possibly the matrices. Recall that we also know that $l = l^*$ and $d = d^*$. Since $\mathcal{M}$ is in reduced echelon form, its columns are linearly independent. We deduce

$$\dim\left(\ker\left(M^{tr}\right)\right) = \dim\left(\ker\left(\left(M^*\right)^{tr}\right)\right). \tag{7}$$

Consider the affine vector spaces V_M and V_{M^*}. By their definition in (1) and from (7) we deduce that $\dim(V_M) = \dim(V_{M^*})$.

Consistency of $E_{\mathcal{M}}$ and $\boldsymbol{z} \in V_M$ imply $\boldsymbol{z} \cdot \boldsymbol{\lambda} = r = s$. But $\boldsymbol{\lambda} = \boldsymbol{\lambda^*} = M^* \cdot \boldsymbol{b}^*$, and thus $\boldsymbol{z} \cdot M^* \cdot \boldsymbol{b}^* = s$. Since $\boldsymbol{z} \notin V_{M^*}$, vector $\boldsymbol{v} := \boldsymbol{z} \cdot M^*$ satisfies $\boldsymbol{v} \neq \boldsymbol{e}_1$ and $\boldsymbol{v} \cdot \boldsymbol{b}^* = s$. Hence, the following equation holds by the definition of $\boldsymbol{b^*}$ in (6):

$$\boldsymbol{v} \cdot \boldsymbol{b}^* = v_1 \cdot s + \sum_{i=2}^{d^*} v_i \cdot \left(b_i' + s \cdot a^{i-1}\right) = s.$$

Since $\boldsymbol{v} \neq \boldsymbol{e}_1$, at least one of the elements v_i in the sum is not equal to zero. Let k be the largest number $2 \leq k \leq d^* < q$ with $v_k \neq 0$. Then

$$\left(v_1 \cdot s + \sum_{i=2}^{k} v_i \cdot \left(b_i' + s \cdot a^{i-1}\right)\right) \cdot a^{q+2-k} = s \cdot a^{q+2-k}.$$

We deduce

$$v_k \cdot s \cdot a^{q+1} = (1 - v_1) \cdot s \cdot a^{q+2-k} - \sum_{i=2}^{k} v_i \cdot b_i' \cdot a^{q+2-k} - \sum_{j=2}^{k-1} v_j \cdot s \cdot a^{q+1+j-k}.$$

Vector $\boldsymbol{z}$ and $\boldsymbol{v} = \boldsymbol{z} \cdot M^*$ from above can be efficiently computed. Hence, $\mathcal{B}$ can compute $e(g,g)^{s \cdot a^{q+1}}$ from:

$$e\left(g,g\right)^{s \cdot a^{q+1} \cdot v_k} := e\left(g^s, g^{a^{q+2-k}}\right)^{1-v_1} \cdot \prod_{i=2}^{k} e\left(g, g^{a^{q+2-k}}\right)^{-v_i \cdot b_i'} \cdot \prod_{j=2}^{k-1} e\left(g^s, g^{a^{q+1+j-k}}\right)^{-v_j}$$

and solve its challenge.

(c) $\rho = \rho^*$ and for all $\boldsymbol{z} \in V_M : \boldsymbol{z} \in V_{M^*}$. As in the last subcase $\dim\left(V_M\right) = \dim\left(V_{M^*}\right)$. We deduce $V_M = V_{M^*}$, which implies $\ker\left(M^{tr}\right) = \ker\left(\left(M^*\right)^{tr}\right)$. Therefore, there exists an invertible matrix $T \in \mathbb{Z}_p^{d \times d}$ such that $M = M^* \cdot T$. For $\boldsymbol{w} \in V_M$ arbitrary, we get $\boldsymbol{e}_1 = \boldsymbol{w} \cdot M = \boldsymbol{w} \cdot M^* \cdot T = \boldsymbol{e}_1 \cdot T$. Hence, T has the form

$$T = \begin{bmatrix} 1 & 0 \cdots 0 \\ a_2 & \\ \vdots & T' \\ a_d & \end{bmatrix},$$

where T' is invertible. This implies that the last $d-1$ columns of M and the last $d-1$ columns of M^* span the same vector space. From Lemma 1 and reduced echelon form of MSPs, we deduce that T' is the identity matrix and these columns are equal.

Assume, that one of the a_i's is not equal to zero. This implies that at least one of the matrices M, M^* violates condition (d) in Definition 5, contradicting the fact that both MSPs are in reduced echelon form. Hence, the assumption is wrong and T is the identity matrix, which contradicts $E_{\mathcal{M}} \neq E^*_{\mathcal{M}^*}$. Therefore, this subcase never occurs.

Guess: $\mathcal{B}$ outputs the guess of $\mathcal{A}$.

Next, we analyze the success probability of $\mathcal{B}$. Since we have to abort if events $Abort_1$ or $Abort_2$ occur, $\mathcal{B}$'s simulation is not perfect. Furthermore, the simulation of key generation queries from [Wat11] aborts with negligible probability over the random choice of polynomials $p_i(x)$.

The event $Abort_1$ happens with negligible probability over the random choice of the exponent s from the challenge, since the view of $\mathcal{A}$ in Phase 1 is independent of s.

To analyze the probability for the event $Abort_2$ consider the following algorithm $\mathcal{B}'$. This algorithm computes a collision for $\mathcal{UOWHF}$ in case of $Abort_2$:

- $\mathcal{B}'$ plays the role of the q-BDHE challenger and the role of $\mathcal{B}$ until the complete input $X = \langle d^*, e\left(g^s, g_1\right), X_1^*, \ldots, X_{l^*}^* \rangle$ of the hash function is computed in the Setup-Phase.
- $\mathcal{B}'$ commits to X, gets UHF $\leftarrow \mathcal{UOWHF}$ and continues to simulate $\mathcal{A}$.

– If event $Abort_2$ occurs, $\mathcal{B}'$ gets $X' \neq X$ with $\mathrm{UHF}(X) = \mathrm{UHF}(X')$ and outputs X'.

Hence, under the assumption that $\mathcal{UOWHF}$ is a family of universal one-way hash function event $Abort_2$ occurs with negligible probability. These facts together with the negligible probability for aborts in the simulation of [Wat11] imply the theorem. □

Note that by the choice of the challenge MSP, A controls parts of X. Therefore, unlike [KG09] target collision resistant hash functions are not sufficient for our construction.

Improvement. The tests in (2) and (3) can be performed more efficiently. The elements X_i do not have to be computed explicitly. The test in (2) is equivalent to

$$e\left(\prod_{i=1}^{l} E_i^{w_i}, g\right) \cdot e\left(\prod_{i=1}^{l} H(\rho(i))^{w_i}, E'\right) \overset{?}{=} e\left(E', g_1\right). \tag{8}$$

The tests in (3) ensure that for all $j \in \{1, \ldots, k\}$ it holds $\boldsymbol{\lambda} \cdot \boldsymbol{u}_j = 0$ (cf. Lemma 3). These tests can be replaced by a single randomized test similar to [KG09]:

$$e\left(\prod_{i=1}^{l} E_i^{\sum_{j=1}^{k} r_j \cdot u_{j,i}}, g\right) \cdot e\left(\prod_{i=1}^{l} H(\rho(i))^{\sum_{j=1}^{k} r_j \cdot u_{j,i}}, E'\right) \overset{?}{=} 1, \tag{9}$$

where $r_1, \ldots, r_k \leftarrow \mathbb{Z}_p$. This test ensures that $\boldsymbol{\lambda} \cdot \sum_{j=1}^{k} r_j \cdot \boldsymbol{u}_j = 0$. Hence, with negligible probability $1/p$ an encapsulation passes this randomized test although there exists a vector $\boldsymbol{u}_j$ with $\boldsymbol{u}_j \cdot \boldsymbol{\lambda} \neq 0$.

Whereas the original tests require $\mathcal{O}(l)$ pairings and $\mathcal{O}\left(l^2\right)$ exponentiations in $\mathbb{G}_T$, the improved tests require only $\mathcal{O}(1)$ pairings and $\mathcal{O}(l)$ exponentiations in $\mathbb{G}$.

Comparison. In [YAHK11] generic constructions of fully functional CCA-secure KP-ABE schemes and fully functional CCA-secure CP-ABE schemes based on any one-time signature scheme are presented. In the CP-ABE scheme, the complete ciphertext including the access structure is signed. Then, a weaker notion of verifiability than the one used in our construction is sufficient to prove CCA-security. Basically, the Verify algorithms in their constructions ensure that decryption queries can be correctly answered even for ciphertexts that are not consistent.

In contrast, we construct a CCA-secure KP-AB-KEM and a CCA-secure CP-AB-KEM rather than ABE schemes. For these KEMs we are able to construct a fully functional Verify algorithm. As a consequence, this allows us to use a single hash function instead of an one-time signature scheme. Moreover, we only authenticate parts of the encapsulation, mostly consisting of group elements. Alternatively, we can drop the consistency test in (3) and hash the complete encapsulation. Similar to [YAHK11] this is sufficient to answer decapsulation queries also for encapsulations that are not consistent. This alternative construction already exploits the simpler structure of KEMs as compared to fully functional encryption schemes by replacing one-time signatures by a hash function. However, the construction presented in this paper has the additional benefit that we do not have to hash a complex encapsulation consisting of various different data types. Moreover, it supports full public verifiability of encapsulations. See [NMP+12] for a thorough discussion of applications and advantages of encryptions schemes with publicly verifiable ciphertexts.

References

[ABV$^+$12] Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee. Functional encryption for threshold functions (or fuzzy IBE) from lattices. In *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 280–297. Springer, 2012.

[BCHK07] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

[Bei96] Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[BFMLS08] Kamel Bentahar, Pooya Farshim, John Malone-Lee, and Nigel P. Smart. Generic constructions of identity-based and certificateless KEMs. *Journal of Cryptology*, 21:178–199, 2008.

[BMW05] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security*, pages 320–329. ACM, 2005.

[Boy13] Xavier Boyen. Attribute-based functional encryption on lattices. In *10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 122–142. Springer, 2013.

[CC09] Melissa Chase and Sherman S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *ACM Conference on Computer and Communications Security*, pages 121–130. ACM, 2009.

[Cha07] Melissa Chase. Multi-authority attribute based encryption. In *4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 515–534. Springer, 2007.

[CN07] Ling Cheung and Calvin C. Newport. Provably secure ciphertext policy ABE. In *ACM Conference on Computer and Communications Security*, pages 456–465. ACM, 2007.

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen-ciphertext attack. *SIAM Journal on Computing*, 33:167–226, 2003.

[CZF11] Cheng Chen, Zhenfeng Zhang, and Dengguo Feng. Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In *Provable Security - 5th International Conference*, volume 6980 of *Lecture Notes in Computer Science*, pages 84–101. Springer, 2011.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.

[KG09] Eike Kiltz and David Galindo. Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. *Theoretical Computer Science*, 410(47-49):5093–5111, 2009.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, 2006.

[LC10] Zhen Liu and Zhenfu Cao. On efficiently transferring the linear secret-sharing scheme matrix in ciphertext-policy attribute-based encryption. *IACR Cryptology ePrint Archive*, 374, 2010.

[LOS$^+$10] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91. Springer, 2010.

[LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479. Springer, 2010.

[LW11a] Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 568–588. Springer, 2011.

[LW11b] Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 547–567. Springer, 2011.

[LW12] Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 180–198. Springer, 2012.

[NMP$^+$12] Juan Manuel González Nieto, Mark Manulis, Bertram Poettering, Jothi Rangasamy, and Douglas Stebila. Publicly verifiable ciphertexts. In *Security and Cryptography for Networks*, volume 7485 of *Lecture Notes in Computer Science*, pages 393–410. Springer, 2012.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing*, pages 33–43. ACM, 1989.

[OSW07] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pages 195–203. ACM, 2007.

[Sho06] Thomas S. Shores. *Applied Linear Algebra and Matrix Analysis.* Springer, 2006.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.

[Wat11] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer, 2011.

[YAHK11] Shota Yamada, Nuttapong Attrapadung, Goichiro Hanaoka, and Noboru Kunihiro. Generic constructions for chosen-ciphertext secure attribute based encryption. In *Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 71–89. Springer, 2011.

A Appendix A

Proof of Lemma 1. (cf. Corollary 1.1 in [Sho06]) Using only elementary column operations one can convert M into N, since $\text{span}\,(m_1, \ldots, m_d) = \text{span}\,(n_1, \ldots, n_d)$. The lemma follows by the uniqueness of the reduced column echelon form of matrices (see Theorem 1.3 in [Sho06]). □

Proof of Lemma 2. Let $\boldsymbol{\lambda} \in \text{im}\,(M)$, then there exists a vector $\boldsymbol{b}$ such that $\boldsymbol{\lambda} = M \cdot \boldsymbol{b}$. Hence, $\forall j \in \{1, \ldots, k\} : \boldsymbol{u}_j \cdot \boldsymbol{\lambda} = \boldsymbol{u}_j \cdot M \cdot \boldsymbol{b} = 0$.

By the rank-nullity theorem we have $\dim\left(\ker\left(M^{tr}\right)\right) + \dim\left(\text{im}\left(M^{tr}\right)\right) = l$. Furthermore, $\dim\left(\ker\left(M^{tr}\right)\right) = k$ and $\dim\left(\text{im}\left(M^{tr}\right)\right) = \dim\left(\text{im}\,(M)\right)$. Hence, $\dim\left(\text{im}\,(M)\right) = l - k$. But the kernel of matrix with k linear independent rows $\boldsymbol{u}_j$ also has dimension $l - k$. The claim follows. □

Proof of Lemma 3. We prove the lemma by "$1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$".

"$1 \Rightarrow 2$": The first step follows immediately since $\mathbb{G}$ has prime order.

"$2 \Rightarrow 3$": By Lemma 2 it holds $\lambda \in \mathrm{im}\,(M)$. Hence, there exists $\boldsymbol{b}$ with $M \cdot \boldsymbol{b} = \boldsymbol{\lambda}$ and by the first requirement $s = \boldsymbol{w} \cdot \boldsymbol{\lambda} = \boldsymbol{w} \cdot M \cdot \boldsymbol{b} = b_1$.

"$3 \Rightarrow 1$": Using the property of $\boldsymbol{b}$ we easily show:

$$\prod_{i=1}^{l} \left(h^{\lambda_i}\right)^{w_i} = h^{\boldsymbol{w} \cdot \boldsymbol{\lambda}} = h^{\boldsymbol{w} \cdot M \cdot \boldsymbol{b}} = h^{b_1} = h^s$$

and analogously

$$\forall j \in \{1, \dots, k\} : \prod_{i=1}^{l} \left(h^{\lambda_i}\right)^{u_{j,i}} = h^{\boldsymbol{u}_j \cdot M \cdot \boldsymbol{b}} = 1.$$

Hence, the lemma follows. □

B Appendix B: CCA-Secure KP-AB-KEM

In this section we present our KP-AB-KEM construction selectively secure against adaptive chosen-ciphertext attacks. Our starting point is an encryption scheme selectively secure against chosen-plaintext attacks. Basically, this is the large universe construction from [GPSW06]. However, we use monotone span programs to realize access structures rather than boolean trees. The modifications are straightforward. To achieve security against chosen-ciphertext attacks we use the technique of [KG09]. In the following, n is the maximum number of attributes in an encapsulation. The four algorithms of the KP-AB-KEM are defined as follows.

Setup(1^η, n) Generate bilinear groups $(p, \mathbb{G}, \mathbb{G}_T, e, g \in \mathbb{G}) \leftarrow \mathcal{G}(1^\eta)$ and elements $g_2, g_3 \leftarrow \mathbb{G}, \{H_i \leftarrow \mathbb{G}\}_{i \in \{0,\dots,n\}}, \alpha \leftarrow \mathbb{Z}_p$. Set $g_1 := g^\alpha$ and $Y := e(g_1, g_2)$. Choose an UOWHF UHF : $\{0,1\}^{\lfloor \log(n) \rfloor + 1} \times \mathbb{G} \to \mathbb{Z}_p$. The master secret is α and the public parameters are params $:= \Big((p, \mathbb{G}, \mathbb{G}_T, e, g), g_1, g_2, g_3, \{H_i\}_{i \in \{0,\dots,n\}}, Y, \mathrm{UHF}\Big)$. The key space for the DEM is $\mathbb{K} = \mathbb{G}_T$ and the universe of attributes is $\mathcal{U} = \mathbb{Z}_p$.

Compared to the modified large universe construction of [GPSW06], we add the group element g_3 and the hash function UHF.

As in [Wat11], the elements H_i define a publicly computable function:

$$\begin{aligned} H : \mathcal{U} \to & \quad \mathbb{G} \\ x \mapsto & \prod_{i \in \{0,\dots,n\}} H_i^{\Delta_{i,\{0,\dots,n\}}(x)} \end{aligned}$$

where $\Delta_{i,\{0,\dots,n\}}(x), i = 0, \dots, n$ are the Lagrange interpolation polynomials. Hence, $H(x) = g^{h(x)}$ for some polynomial $h(x)$ of degree at most n. Since the H_i are chosen uniformly and independently at random, h is also chosen uniformly at random.

KeyGen$_{\mathrm{msk}}$($\mathcal{M}$) with MSP $\mathcal{M} = (M, \rho)$, $M \in \mathbb{Z}_p^{l \times d}$ for some arbitrary access structure $\mathbb{A}$. Choose $b_2, \dots, b_d \leftarrow \mathbb{Z}_p$. Set $\boldsymbol{b} := (\alpha, b_2, \dots, b_d)$ and $\boldsymbol{\lambda} := M \cdot \boldsymbol{b} \in \mathbb{Z}_p^l$. Choose $\boldsymbol{r} \leftarrow \mathbb{Z}_p^l$

and set $D_i := g_2^{\lambda_i} \cdot H(\rho(i))^{-r_i}$ and $D'_i := g^{r_i}$ for all $i \in \{1, \ldots, l\}$. The secret key is $sk_{\mathcal{M}} := \left(\mathcal{M}, \{D_i, D'_i\}_{i \in \{1,\ldots,l\}}\right)$. This is as in [GPSW06].

Encaps(γ) with $|\gamma| \leq n$. Choose $s \leftarrow \mathbb{Z}_p$. Set $E' := g^s$ and $\{E_x := H(x)^s\}_{x \in \gamma}$. Compute $t := \mathrm{UHF}\left(|\gamma|, E'\right)$ and $E'' := \left(g_1^t \cdot g_3\right)^s$. The encapsulated key is $K := Y^s$ and the encapsulation of K is: $E_\gamma := \left(\gamma, E', \{E_x\}_{x \in \gamma}, E''\right)$. Compared to the original scheme we only add the group element E''.

Decaps$_{sk_{\mathcal{M}}}(E_\gamma)$ with MSP $\mathcal{M} = (M, \rho)$, $M \in \mathbb{Z}_p^{l \times d}$ for some arbitrary access structure $\mathbb{A}$. Compute a vector $\boldsymbol{w} \in \mathbb{Z}_p^l$ with $\boldsymbol{w} \cdot M = \boldsymbol{e}_1$ and $w_i = 0$ for all $i \in \{1, \ldots, l\}$ with $\rho(i) \notin \gamma$ and reject, if such an vector does not exists, since $\mathcal{M}$ rejects γ.

Compute $t' := \mathrm{UHF}\left(|\gamma|, E'\right)$. Reject, if one of the following consistency checks fails

$$\forall x \in \gamma : e\left(E', H(x)\right) \overset{?}{=} e\left(g, E_x\right), \tag{10}$$

$$e\left(E', g_1^{t'} \cdot g_3\right) \overset{?}{=} e\left(g, E''\right), \tag{11}$$

For all $i \in \{1, \ldots, l\}, w_i \neq 0$, compute $Z_i = e\left(D_i, E'\right) \cdot e\left(D'_i, E_{\rho(i)}\right)$ and output $K := \prod_{i \in \{1,\ldots,l\}, w_i \neq 0} Z_i^{w_i} = Y^s$.

Compared to the modified large universe construction of [GPSW06], we only added the consistence checks in (10) and in (11). Furthermore the tests in (10) can be exchanged by a single randomized test similar to [KG09].

Public verifiability: One easily checks that test in (10) ensures that there exists $r \in \mathbb{Z}_p$ such that $E' = g^r$ and $\forall x \in \gamma : E_x = H(x)^r$. Test in (11) checks the correct form of the additional element E'' and is required to achieve CCA-security. The secret key is not involved into the consistency tests.

Correctness Every correct generated encapsulation pass the consistency check. Correctness follows directly from correctness of the original ABE, where K is the element used to obscure the message.

Security proof: The next theorem can be proved by combining the proof techniques in [KG09] and [GPSW06]. Moreover, the proof is a simpler version of the security proof for our chosen-ciphertext secure ciphertext-policy attribute-based encapsulation mechanism.

Theorem 2. *Assume $\mathcal{UOWHF}$ is a family of universal one-way hash function. Under the DBDH-assumption relative to a group generator $\mathcal{G}$, our key-policy attribute-based key encapsulation mechanism is selectively secure against adaptive chosen-ciphertext attacks.*