Attribute-Based Identification: Definitions and Efficient Constructions*

Hiroaki Anada, Seiko Arita, Sari Handa, and Yosuke Iwabuchi

Graduate School of Information Security, Institute of Information Security, Japan
2-14-1 Tsuruya-Cho, Kanagawa-Ku, Yokohama-Shi, 221-0835, Japan
hiroaki.anada@gmail.com, {arita,mgs125502,mgs123101}@iisec.ac.jp

Abstract. We propose a notion of attribute-based identification (ABID) in two flavors: prover-policy ABID (PP-ABID) and verifier-policy ABID (VP-ABID). In a PP-ABID scheme, a prover has an authorized access policy written as a boolean formula over attributes, while each verifier maintains a set of attributes. The prover is accepted when his access policy fits the verifier's set of attributes. In a VP-ABID scheme, a verifier maintains an access policy written as a boolean formula over attributes, while each prover has a set of authorized attributes. The prover is accepted when his set of attributes satisfies the verifier's access policy. Our design principle is first to construct key-policy and ciphertext-policy attribute-based key encapsulation mechanisms (KP-ABKEM and CP-ABKEM). Second, we convert KP-ABKEM and CP-ABKEM into challenge-and-response PP-ABID and VP-ABID, respectively, by encapsulation-and-decapsulation. There, we show that KP-ABKEM and CP-ABKEM only have to be secure against chosen-ciphertext attacks on one-wayness (OW-CCA secure) for the obtained PP-ABID and VP-ABID to be secure against concurrent man-in-the-middle attacks (cMiM secure). According to the design principle, we construct concrete KP-ABKEM and CP-ABKEM with the OW-CCA security by enhancing the KP-ABKEM of Ostrovsky, Sahai and Waters and CP-ABKEM of Waters, respectively. Finally, we obtain concrete PP-ABID and VP-ABID schemes that are proved to be selectively secure in the standard model against cMiM attacks.

Keywords: access policy, attribute, identification, key encapsulation mechanism.

1 Introduction

An identification (ID) scheme enables a prover to convince a verifier that the prover certainly knows a secret key that corresponds to the matching public key. For example, Σ-protocols [7] such as the Schnorr protocol [14,5] are widely recognized. In these ID schemes, the public key to which the verifier refers limits the corresponding secret key uniquely, and also, the corresponding prover.

* Supported by MEXT-Supported Program for the Strategic Research Foundation at Private Universities, 2011-2013.

C. Boyd and L. Simpson (Eds.): ACISP 2013, LNCS 7959, pp. 168–186, 2013.

In this paper, we will describe an *attribute-based identification (ABID)*. In an ABID scheme, each entity has credentials called *attributes*. On the other hand, an *access policy* is written as a boolean formula over those attributes. Then, a verifier can identify that a prover certainly belongs to a set of entities that have authorized access policies that fit the verifier's attributes, or, in the dual flavor, a verifier can identify that a prover certainly belongs to a set of entities that possess authorized attributes that satisfy the verifier's access policy. Hence, ABID schemes can be considered as an expansion of the usual ID schemes.

However, ABID schemes are not a mere expansion, but have useful applications beyond those of the usual ID schemes. For example, the following scenarios of smart card systems motivate us to apply ABID.

Functional Tickets. Suppose that we are going to stay at a resort complex, a ski resort, for instance. We search Web sites or brochures for information about services: available dates, accommodation, ski lifts, restaurants in ski areas and hot springs around the areas. For each service, we usually buy a ticket, paying with money or using a credit card. However, acquiring many tickets and carrying a wallet is inconvenient, and therefore, it would be more convenient if we could gain access to these services by using only one smart card. In the smart card, a service authority writes an access policy in terms of the service names that we choose, for instance, [January 1 to 4, 2014] AND [[Hotel A] OR [Ski Lift AND [Day OR Night]] OR [Lunch OR Beer] OR [Hot Spring X]]. A *functional ticket* is a ticket embedded in a smart card that realizes an access policy as a boolean formula over services, as in this scenario. Here, the access policy is chosen according to our requirements.

Functional Gates. Suppose that we have to design a security gate system for an office building in which different kinds of people work: employees of several companies holding many different positions, security guards, food service staffs, cleaning staffs and janitors. There are also many types of security gates to be designed: building entrances, intelligent elevators to limit available floors, company gates, common refreshment areas and room doors for the above staffs. In this case, one solution is to use smart cards and gates with sensors. That is, an authority issues each person a smart card in which a set of attribute data is written. Each gate decides whether to "pass" each person carrying a smart card according to the gate's access policy, for instance, [Year 2014] AND [[[Company A] AND [Manager]] OR [Security Guard]]. A *functional gate* is a gate that maintains an access policy as a boolean formula over attributes of people, as in this scenario. Here, the access policy is chosen according to the kind of people that the gate should allow to pass.

1.1 Our Contributions

Bearing the above scenarios in mind, we propose a notion of attribute-based identification (ABID) that has two flavors corresponding to the scenarios: *prover-policy* ABID and *verifier-policy* ABID.

Prover-Policy ABID. In a prover-policy ABID scheme (PP-ABID, for short), a prover has his own authorized access policy, while each verifier maintains its attributes. Here, the access policy is described over the verifier's attributes. Sending his access policy, each prover queries an authority for his secret key. Then, using this secret key, each prover can convince the verifier that his access policy fits the verifier's attributes. Our PP-ABID defined in this way realizes a functional ticket system.

Verifier-Policy ABID. In a verifier-policy ABID scheme (VP-ABID, for short), a verifier maintains its access policy, while each prover has his own authorized attributes. Here, the access policy is described over the prover's attributes. Sending his attributes, each prover queries an authority for his secret key. Then, using this secret key, each prover convinces the verifier that his attributes satisfy the verifier's access policy. Our VP-ABID defined in this way realizes a functional gate system.

Attack and Security Analysis. An adversary's objective is *impersonation*: giving a *target* set of attributes (or, a *target* access policy) to a verifier, the adversary tries to make the verifier accept him.

First, to reflect a *collusion attack* (that is, an attack launched by collecting secret keys that satisfy a condition), we consider an attack model in which an adversary issues key-extraction queries, as is the case for attribute-based encryptions [13,15]. The condition is that the adversary cannot collect any secret key whose intrinsic access policy fits the target set of attributes (or, whose intrinsic set of attributes satisfies the target access policy).

Our main objective is to define a model of *concurrent man-in-the-middle attack (cMiM attack)* in the setting of ABID. "Concurrent" means that an adversary can invoke provers that have *different* secret keys corresponding to different access policies (or, different sets of attributes). The adversary interacts with these provers in an arbitrarily interleaved order of messages. Then, interacting with a verifier on a target set of attributes (or, on a target access policy, respectively) the adversary tries to impersonate a prover. The concurrent attack modeled in this way is a real threat, especially to smart card systems. On the other hand, "man-in-the-middle (MiM)" means that an adversary stands between a prover and a verifier simultaneously. Typically, the adversary first receives a message from the verifier, and then, the adversary begins to interact with the prover *adaptively* to the verifier's message. The MiM attack and the cMiM attack modeled in this way are real threats, especially to network applications.

As is the case for usual ID schemes, *reset attacks* should be considered. In a reset attack, an adversary aborts an interaction at any point, and then rewinds the interaction back to any other point to start the interaction again. At that re-starting point, the adversary is allowed to change messages as long as the interaction remains valid (as captured by the word "reset"). Such a reset attack is a strong threat, not only to smart card systems [4] (including the functional tickets and functional gates described above) but also to virtual machine services in cloud computing [17]. As our contribution, an ABID constructed using our

generic conversion becomes secure against the reset attacks in both senses of prover-resettable and verifier-resettable [4].

It is desirable that a verifier learns nothing about a prover more than that he belongs to the set of entities that have access policies fitting the verifier's attributes (or, belongs to the set of entities that possess attributes satisfying the verifier's access policy). In fact, by this property (*anonymity*), the prover's privacy is protected when using a functional ticket, as opposed to using a credit card the track of which is recorded. As our contribution, our concrete ABID in Section 5 possesses this anonymity.

Design Principle. First, we construct key-policy and ciphertext-policy attribute-based key encapsulation mechanisms (KP-ABKEM and CP-ABKEM [13,15]). Second, we convert the KP-ABKEM and CP-ABKEM into challenge-and-response PP-ABID and VP-ABID, respectively, by encapsulation-and-decapsulation. There, we show that KP-ABKEM and CP-ABKEM only have to be secure against chosen-ciphertext attacks on one-wayness (OW-CCA secure) for the obtained PP-ABID and VP-ABID to be secure against cMiM attacks (cMiM secure). We stress that the security of indistinguishability against chosen-ciphertext attacks (the IND-CCA security) is excessive, and the OW-CCA security is enough for constructing a cMiM secure ABID.

Concrete Constructions. We construct KP-ABKEM and CP-ABKEM with the OW-CCA security from the KP-ABKEM of Ostrovsky, Sahai and Waters [13] and CP-ABKEM of Waters [15]. Their KEMs are secure in the indistinguishability game of chosen-plaintext attack (IND-CPA secure). Our strategy is to apply the algebraic trick of Boneh and Boyen [3] and Kiltz [11] to attain CCA security. Then our generic conversions yield concrete PP-ABID and VP-ABID.

New Number Theoretic Assumptions. We will introduce the Computational Bilinear Diffie-Hellman Assumption with Gap on Target Group and the Computational q-Parallel Bilinear Diffie-Hellman Exponent Assumption with Gap on Target Group. These assumptions are reasonable, for example, for the bilinear map of a pairing on a elliptic curve. We need these assumptions for security proofs of concrete constructions.

1.2 Related Works

Anonymous Deniable Predicate Authentication. First, we should refer to the work of Yamada et al. [16]. Our generic construction of ABID can be considered as a special case of their predicate authentication; however, it differs in at least two points. The first point is that our objective is to provide a simple, fast ABID. In contrast, Yamada et al.'s objective is to apply their verifiable predicate encryption to yield an anonymous deniable message authentication. In fact, we simply consider a 2-round challenge-and-response ABID, whereas they proposed a 6-round protocol for deniability. The second point is that we provide more efficient concrete ABID by using the algebraic trick ([3,11]). In contrast, they used their generic transformation which causes a longer secret key,

a longer ciphertext and more computational costs for encryption and decryption than ours, because a verification key of a one-time signature is involved in their generic transformation.

Identification Scheme from KEM. Anada and Arita [1] proposed a design principle of obtaining a cMiM secure ID scheme by constructing KEM. Their concrete ID scheme is more efficient than known Σ-protocol-based cMiM secure ID schemes ([9]). Our scheme can be seen as an attribute-based version of theirs.

1.3 Organization of the Paper

In Section 2, we survey the required terms. In Section 3, we define the notions of PP-ABID and VP-ABID, cMiM attacks and security against it. In Section 4, we provide generic conversions from KP-ABKEM to PP-ABID and from CP-ABKEM to VP-ABID. In Section 5, we construct concrete KP-ABKEM and CP-ABKEM. Finally, we obtain concrete PP-ABID and VP-ABID. In Section 6, we present the conclusions of our study. Because of space limitation, the case of PP-ABID is described in the main text and the case of VP-ABID is only shortly described in the Appendix.

2 Preliminaries

The security parameter is denoted by λ. A prime of bit length λ is denoted by p. A multiplicative cyclic group of order p is denoted by $\mathbb{G}$. The ring of the exponent domain of $\mathbb{G}$, which consists of integers from 0 to $p-1$ with modulo p operation, is denoted by $\mathbb{Z}_p$. When an algorithm A with input a outputs z, we denote it as $z \leftarrow A(a)$. When A with input a and B with input b interact with each other and B outputs z, we denote it as $z \leftarrow \langle A(a), B(b)\rangle$. When A has oracle-access to $\mathcal{O}$, we denote it as $A^{\mathcal{O}}$. When A has concurrent oracle-access to n oracles $\mathcal{O}_1, \ldots, \mathcal{O}_n$, we denote it as $A^{\mathcal{O}_i|_{i=1}^n}$. Here "concurrent" means that A accesses to oracles in arbitrarily interleaved order of messages. A probability of an event E is denoted by Pr[E]. A probability of an event E on condition that events $\mathrm{E}_1, \ldots, \mathrm{E}_m$ occur in this order is denoted as $\Pr[\mathrm{E}_1; \cdots; \mathrm{E}_m : \mathrm{E}]$.

2.1 Access Structure

Let $\mathcal{U} = \{\chi_1, \ldots, \chi_u\}$ be an attribute universe, or simply set $\mathcal{U} = \{1, \ldots, u\}$. We must distinguish two cases: the case that $\mathcal{U}$ is small (i.e. $|\mathcal{U}| = u$ is bounded by some polynomial in λ) and the case that $\mathcal{U}$ is large (i.e. u is not necessarily bounded). We assume the *small case* unless we state the large case explicitly. An *access structure*, which reflects a given access policy, is defined as a collection $\mathbb{A}$ of non-empty subsets of $\mathcal{U}$. That is, $\mathbb{A} \subset 2^{\mathcal{U}} \backslash \{\phi\}$. An access structure $\mathbb{A}$ is called *monotone* if for any $B \in \mathbb{A}$ and $B \subset C$, $C \in \mathbb{A}$ holds. We will consider in this paper only monotone access structures.

2.2 Linear Secret-Sharing Scheme

A secret-sharing scheme Π over a set of parties $\mathcal{P}$ is called a linear secret-sharing scheme (LSSS) over $\mathbb{Z}_p$ ([2]), if Π satisfies the following conditions.

1. The shares for each party form a vector over $\mathbb{Z}_p$
2. There exists a matrix M called the *share-generating matrix* for Π, of size $l \times n$, and a function ρ which maps each row index i of M to a party $\mathcal{P}$, $\rho : \{1, ..., l\} \to \mathcal{P}$.

To make shares for a secret $s \in \mathbb{Z}_p$, we first choose $n-1$ random values $v_2, \ldots, v_n \in \mathbb{Z}_p$ and form a vector $\boldsymbol{v} = (s, v_2, \ldots, v_n)$. For $i = 1$ to l, we calculate each share $\lambda_i = \boldsymbol{v} \cdot M_i$, where M_i denotes the i-th row vector of M and $\cdot$ denotes the formal inner product. The share λ_i belongs to the party $\rho(i)$.

Looking at $\mathcal{P}$ as an attribute universe $\mathcal{U}$, Π determines an access structure $\mathbb{A}$ as (M, ρ) ([13,15]). Suppose that an attribute set $S \subset \mathcal{U}$ satisfies $\mathbb{A}$ ($S \in \mathbb{A}$) and put $I_S = \rho^{-1}(S) \subset \{1, \ldots, l\}$. Then, there exists a set of constants $\{\omega_i \in \mathbb{Z}_p; i \in I_S\}$ called *linear reconstruction constants* ([2]) that satisfies $\sum_{i \in I_S} \omega_i \lambda_i = s$. These constants $\{\omega_i\}_{i \in I_S}$ can be computed in time polynomial in the size of M. We denote the algorithm by $\mathrm{Recon}(I_S, M)$. If S does not satisfy $\mathbb{A}$ ($S \notin \mathbb{A}$), then no such constants $\{\omega_i\}_{i \in I_S}$ exist, but instead, there is a vector $\boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{Z}_p^n$ such that $w_1 = 1$ and $\boldsymbol{w} \cdot M_i = 0$ for all $i \in I_S$. $\boldsymbol{w}$ also can be computed in time polynomial in the size of M ([15]).

2.3 Key-Policy Attribute-Based KEM

Scheme. A key-policy ABKEM, `KP-ABKEM`, consists of four probabilistic polynomial time (PPT, for short) algorithms (Setup, KeyGen, Encap, Decap).

Setup$(\lambda, \mathcal{U}) \to (\mathbf{PK}, \mathbf{MSK})$**.** Setup takes as input the security parameter λ and the attribute universe $\mathcal{U}$. It returns a public key PK and a master secret key MSK.

KeyGen$(\mathbf{PK}, \mathbf{MSK}, \mathbb{A}) \to \mathbf{SK}_{\mathbb{A}}$**.** A key generation algorithm KeyGen takes as input the public key PK, the master secret key MSK and an access structure $\mathbb{A}$. It returns a secret key $\mathrm{SK}_{\mathbb{A}}$ that corresponds to $\mathbb{A}$.

Encap$(\mathbf{PK}, S) \to (\kappa, \psi)$**.** Encap takes as input the public key PK and an attribute set S. It returns a random KEM key κ and its encapsulation ψ (we also call it a ciphertext). We denote the set of all possible output (κ, ψ) of $\mathrm{Encap}(\mathrm{PK}, S)$ by $[\mathrm{Encap}(\mathrm{PK}, S)]$. If $(\tilde{\kappa}, \tilde{\psi}) \in [\mathrm{Encap}(\mathrm{PK}, S)]$, then $(\tilde{\kappa}, \tilde{\psi})$ is called *consistent* and otherwise, *inconsistent*.

Decap$(\mathbf{PK}, \mathbf{SK}_{\mathbb{A}}, \psi) \to \hat{\kappa}$**.** Decap takes as input the public key PK, an encapsulation ψ and a secret key $\mathrm{SK}_{\mathbb{A}}$. It returns a decapsulation result $\hat{\kappa}$ of ψ under $\mathrm{SK}_{\mathbb{A}}$. We demand correctness of `KP-ABKEM` that for any λ and $\mathcal{U}$, and if $S \in \mathbb{A}$, then $\Pr[(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathrm{Setup}(\lambda, \mathcal{U}); \mathrm{SK}_{\mathbb{A}} \leftarrow \mathrm{KeyGen}(\mathrm{PK}, \mathrm{MSK}, \mathbb{A}); (\kappa, \psi) \leftarrow \mathrm{Encap}(\mathrm{PK}, S); \hat{\kappa} \leftarrow \mathrm{Decap}(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}}, \psi) : \kappa = \hat{\kappa}] = 1$.

$\mathbf{Exprmt}^{\text{ow-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda,\mathcal{U})$: //Adaptive S^*	$\mathbf{Exprmt}^{\text{ow-sel-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda,\mathcal{U})$: //Selective S^*
$(\mathrm{PK},\mathrm{MSK}) \leftarrow \mathbf{Setup}(\lambda,\mathcal{U})$	$(\mathrm{PK},\mathrm{MSK}) \leftarrow \mathbf{Setup}(\lambda,\mathcal{U})$
$S^* \leftarrow \mathcal{A}^{\mathcal{KG}(\mathrm{PK},\mathrm{MSK},\cdot),\mathcal{DEC}(\mathrm{PK},\mathrm{SK}_\cdot,\cdot)}(\mathrm{PK},\mathcal{U})$	$S^* \leftarrow \mathcal{A}(\lambda,\mathcal{U})$
$(\kappa^*,\psi^*) \leftarrow \mathbf{Encap}(\mathrm{PK},S^*)$	$(\kappa^*,\psi^*) \leftarrow \mathbf{Encap}(\mathrm{PK},S^*)$
$\hat{\kappa^*} \leftarrow \mathcal{A}^{\mathcal{KG}(\mathrm{PK},\mathrm{MSK},\cdot),\mathcal{DEC}(\mathrm{PK},\mathrm{SK}_\cdot,\cdot)}(\psi^*)$	$\hat{\kappa^*} \leftarrow \mathcal{A}^{\mathcal{KG}(\mathrm{PK},\mathrm{MSK},\cdot),\mathcal{DEC}(\mathrm{PK},\mathrm{SK}_\cdot,\cdot)}(\mathrm{PK},\psi^*)$
If $\hat{\kappa^*} = \kappa^*$ Return WIN else Return LOSE	If $\hat{\kappa^*} = \kappa^*$ Return WIN else Return LOSE

Fig. 1. The experiment of an adversary $\mathcal{A}$ that executes a chosen-ciphertext attack on one-wayness of KP-ABKEM. The left side: the case of adaptive S^*; the right side: the case of selective S^*.

Chosen-Ciphertext Attack on One-Wayness of KP-ABKEM and Security. The following experiment $\mathbf{Exprmt}^{\text{ow-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda,\mathcal{U})$ of an adversary $\mathcal{A}$ defines the game of chosen-ciphertext attack on one-wayness of KP-ABKEM (the OW-CCA game).

In the experiment, $\mathcal{A}$ issues two types of queries. One is key-extraction queries to the key-generation oracle $\mathcal{KG}$. Giving an attribute set $\mathbb{A}_i$, $\mathcal{A}$ queries $\mathcal{KG}(\mathrm{PK}, \mathrm{MSK}, \cdot)$ for the secret key $\mathrm{SK}_{\mathbb{A}_i}$. Another is decapsulation queries to the decapsulation oracle $\mathcal{DEC}$. Giving a pair $(\mathbb{A}_j, \psi_j)$ of an attribute set and an encapsulation, $\mathcal{A}$ queries $\mathcal{DEC}(\mathrm{PK}, \mathrm{SK}_\cdot, \cdot)$ for the decapsulation result $\hat{\kappa}_j$. Here an attribute set S_j, which is used to generate a ciphertext, is included in ψ_j. When $S_j \notin \mathbb{A}_j$, $\hat{\kappa}_j = \perp$ is replied to $\mathcal{A}$.

The attribute set S^* declared by $\mathcal{A}$ is called a *target attribute set*. The encapsulation ψ^* is called a *challenge ciphertext*. Two restrictions are imposed on $\mathcal{A}$ concerning S^* and ψ^*. In key-extraction queries, each attribute set $\mathbb{A}_i$ must satisfy $S^* \notin \mathbb{A}_i$. In decapsulation queries, each pair $(\mathbb{A}_j, \psi_j)$ must satisfy $S^* \notin \mathbb{A}_j \vee \psi_j \neq \psi^*$. Both types of queries are at most q_k and q_d times in total, respectively, which are bounded by a polynomial in λ.

The *advantage* of $\mathcal{A}$ over KP-ABKEM in the OW-CCA game is defined as [1]

$$\mathbf{Adv}^{\text{ow-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda) \stackrel{\text{def}}{=} \Pr[\mathbf{Exprmt}^{\text{ow-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda,\mathcal{U}) \text{ returns WIN}].$$

KP-ABKEM is called *secure against chosen-ciphertext attacks on one-wayness* if, for any PPT $\mathcal{A}$ and for any $\mathcal{U}$, $\mathbf{Adv}^{\text{ow-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda)$ is negligible in λ.

Selective Security. In the *selective game on a target attribute set* (OW-sel-CCA game), $\mathcal{A}$ declares S^* *before* $\mathcal{A}$ receives PK. $\mathbf{Exprmt}^{\text{ow-sel-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda,\mathcal{U})$ defines the selective game. The *advantage* in the OW-sel-CCA game is defined as

$$\mathbf{Adv}^{\text{ow-sel-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda) \stackrel{\text{def}}{=} \Pr[\mathbf{Exprmt}^{\text{ow-sel-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda,\mathcal{U}) \text{ returns WIN}].$$

KP-ABKEM is called *selectively secure against chosen-ciphertext attacks on one-wayness* if, for any PPT $\mathcal{A}$ and for any $\mathcal{U}$, $\mathbf{Adv}^{\text{ow-sel-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda)$ is negligible in λ.

[1] Although we follow the convention, we should write the advantage as a function of λ and u: $\mathbf{Adv}^{\text{ow-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda, u)$, where u is the size of the attribute universe $\mathcal{U}$.

2.4 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. We call $\mathbb{G}$ a source group and $\mathbb{G}_T$ a target group. Let g be a generator of $\mathbb{G}$ and e be a bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The map e satisfies

1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq \mathrm{id}_{\mathbb{G}_T}$(: the identity element of the group $\mathbb{G}_T$).

Groups and a bilinear map are generated by a PPT algorithm **Grp** on input λ: $(p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathbf{Grp}(\lambda)$. We assume that the group operation in $\mathbb{G}$ and $\mathbb{G}_T$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ are computable in time polynomial in λ.

2.5 Computational Bilinear Diffie-Hellman Assumption with Gap on Target Group

We introduce in this paper a new number theoretic assumption, which we call the *Computational Bilinear Diffie-Hellman Assumption with Gap on Target Group*. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Let $a, b, c \in \mathbb{Z}_p, a, b, c \neq 0$, be chosen at random. Put $A := g^a, B := g^b, C := g^c$. Then our new assumption says it is at most with a negligible probability in λ that, for any PPT algorithm $\mathcal{B}$ given input (g, A, B, C), to output $Z = e(g, g)^{abc} \in \mathbb{G}_T$, even with the aid of the decisional Diffie-Hellman oracle for $\mathbb{G}_T$: $\mathcal{DDH}_{\mathbb{G}_T}(\cdot, \cdot, \cdot, \cdot)$. Here a tuple $(\overline{g}, \overline{g}^{z_1}, \overline{g}^{z_2}, \overline{g}^{z_3}) \in \mathbb{G}_T^4$ ($\overline{g} := e(g, g)$) is called a Diffie-Hellman tuple (in $\mathbb{G}_T$) if $z_1 z_2 = z_3$. The oracle $\mathcal{DDH}_{\mathbb{G}_T}$ returns TRUE or FALSE according to whether an input tuple is a Diffie-Hellman tuple or not, respectively. The probability for $\mathcal{B}$ to output $e(g, g)^{abc}$ is denoted as $\mathbf{Adv}^{\text{c-bdh-gap}}_{\mathcal{B},(e,\mathbb{G},\mathbb{G}_T)}(\lambda)$ (the advantage of $\mathcal{B}$ in the computational BDH game with gap on $\mathbb{G}_T$). Note that the above assumption is *different* from the *Gap Bilinear Diffie-Hellman Assumption* [6].

2.6 Target Collision Resistant Hash Functions

Target collision resistant (TCR) hash functions [12] are treated as a family $Hfam_\lambda = \{H_\mu\}_{\mu \in HKey_\lambda}$. The advantage $\mathbf{Adv}^{\text{tcr}}_{\mathcal{CF},Hfam_\lambda}(\lambda)$ of a PPT algorithm $\mathcal{CF}$ over $Hfam_\lambda$ is defined as the success probability to find a target collision.

3 Attribute-Based Identification

In this section, we define a notion of *prover-policy* attribute-based identification (PP-ABID), a concurrent man-in-the-middle attack on PP-ABID and security against it. The case of *verifier-policy* ABID is described in Appendix A in a dual manner to PP-ABID on an access structure $\mathbb{A}$ and an attribute set S.

3.1 Prover-Policy ABID

Scheme. `PP-ABID` consists of four PPT algorithms (Setup, KeyGen, P, V).

Setup$(\lambda, \mathcal{U}) \to (\mathbf{PK}, \mathbf{MSK})$**.** Setup takes as input the security parameter λ and the attribute universe $\mathcal{U}$. It outputs a public key PK and a master secret key MSK.

KeyGen$(\mathbf{PK}, \mathbf{MSK}, \mathbb{A}) \to \mathbf{SK}_{\mathbb{A}}$**.** A key-generation algorithm KeyGen takes as input the public key PK, the master secret key MSK and an access structure $\mathbb{A}$. It outputs a secret key $\mathrm{SK}_{\mathbb{A}}$ corresponding to $\mathbb{A}$.

P$(\mathbf{PK}, \mathbf{SK}_{\mathbb{A}})$ **and V**$(\mathbf{PK}, S)$**.** P and V are interactive algorithms called a *prover* and a *verifier*, respectively. P takes as input the public key PK and the secret key $\mathrm{SK}_{\mathbb{A}}$. Here the secret key $\mathrm{SK}_{\mathbb{A}}$ is given to P by an authority that runs KeyGen(PK,MSK,$\mathbb{A}$). V takes as input the public key PK and an attribute set S. P is provided V's attribute set S by the first round. P and V interact with each other for some, at most constant rounds. Then, V finally returns its decision bit b. $b = 1$ means that V *accepts* P in the sense P has a secret key $\mathrm{SK}_{\mathbb{A}}$ such that S satisfies $\mathbb{A}$. $b = 0$ means that V *rejects* P. We demand correctness of `PP-ABID` that for any λ and $\mathcal{U}$, and if $S \in \mathbb{A}$, then $\Pr[(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathrm{Setup}(\lambda, \mathcal{U}); \mathrm{SK}_{\mathbb{A}} \leftarrow \mathrm{KeyGen}(\mathrm{PK}, \mathrm{MSK}, \mathbb{A}); b \leftarrow \langle \mathrm{P}(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}}), \mathrm{V}(\mathrm{PK}, S)\rangle : b = 1] = 1$.

3.2 Concurrent Man-in-the-Middle Attack on PP-ABID and Security

An adversary $\mathcal{A}$'s objective is impersonation. $\mathcal{A}$ tries to make a verifier V accept with a target attribute set S^*. The following experiment $\mathbf{Exprmt}^{\mathrm{cmim}}_{\mathcal{A},\texttt{PP-ABID}}(\lambda, \mathcal{U})$ of an adversary $\mathcal{A}$ defines the game of concurrent man-in-the-middle attack (cMiM attack, for short) on PP-ABID.

$\mathbf{Exprmt}^{\mathrm{cmim}}_{\mathcal{A},\texttt{PP-ABID}}(\lambda, \mathcal{U})$: //Adaptive S^*
$(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathbf{Setup}(\lambda, \mathcal{U})$
$S^* \leftarrow \mathcal{A}^{\mathcal{KG}(\mathrm{PK},\mathrm{MSK},\cdot), \mathbf{P}_i(\mathrm{PK},\mathrm{SK}_{\mathbb{A}_i})|_{i=1}^{q'_{\mathrm{p}}}}(\mathrm{PK}, \mathcal{U})$
$b \leftarrow \langle \mathcal{A}^{\mathcal{KG}(\mathrm{PK},\mathrm{MSK},\cdot), \mathbf{P}_i(\mathrm{PK},\mathrm{SK}_{\mathbb{A}_i})|_{i=1}^{q_{\mathrm{p}}}}, \mathbf{V}(\mathrm{PK}, S^*)\rangle$
If $b = 1$ Return Win else Return Lose

$\mathbf{Exprmt}^{\text{sel-cmim}}_{\mathcal{A},\texttt{PP-ABID}}(\lambda, \mathcal{U})$: //Seletive S^*
$(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathbf{Setup}(\lambda, \mathcal{U})$
$S^* \leftarrow \mathcal{A}(\lambda, \mathcal{U})$
$b \leftarrow \langle \mathcal{A}^{\mathcal{KG}(\mathrm{PK},\mathrm{MSK},\cdot), \mathbf{P}_i(\mathrm{PK},\mathrm{SK}_{\mathbb{A}_i})|_{i=1}^{q_{\mathrm{p}}}}(\mathrm{PK}), \mathbf{V}(\mathrm{PK}, S^*)\rangle$
If $b = 1$ Return Win else Return Lose

Fig. 2. The experiment of an adversary $\mathcal{A}$ that executes a cMiM attack on `PP-ABID`. The left side: the case of adaptive S^*; the right side: the case of selective S^*.

In the experiment, $\mathcal{A}$ issues key-extraction queries to the key-generation oracle $\mathcal{KG}$. Giving an access structure $\mathbb{A}_i$, $\mathcal{A}$ queries $\mathcal{KG}(\mathrm{PK}, \mathrm{MSK}, \cdot)$ for the secret key $\mathrm{SK}_{\mathbb{A}_i}$. In addition, the adversary $\mathcal{A}$ invokes provers $\mathrm{P}_j(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}_j})$ ($j = 1, \ldots, q'_{\mathrm{p}}, \ldots, q_{\mathrm{p}}$) by giving an access structure $\mathbb{A}_j$ of $\mathcal{A}$'s choice. Acting as a verifier with an attribute set S_j, $\mathcal{A}$ interacts with each P_j.

The attribute set S^* declared by $\mathcal{A}$ is called a *target attribute set*. Two restrictions are imposed on $\mathcal{A}$ concerning S^*. In key-extraction queries, each access structure $\mathbb{A}_i$ must satisfy $S^* \notin \mathbb{A}_i$. In interactions with each prover, every transcript of messages of a whole interaction with a prover $\mathrm{P}_j(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}_j})$ must not be

equal to a transcript of messages of a whole interaction with a verifier $\mathrm{V}(\mathrm{PK}, S^*)$ (that is, a mere *relay of messages* is prohibited in the game of man-in-the-middle attack), or, $S^* \notin \mathbb{A}_j$. The number of key-extraction queries and the number of invoked provers are at most q_k and q_p in total, respectively, which are bounded by a polynomial in λ.

The *advantage* of $\mathcal{A}$ over PP-ABID in the game of cMiM attack is defined as

$$\mathbf{Adv}^{\mathrm{cmim}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda) \stackrel{\mathrm{def}}{=} \Pr[\mathbf{Exprmt}^{\mathrm{cmim}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda,\mathcal{U}) \text{ returns } \textsc{Win}].$$

PP-ABID is called *secure against cMiM attacks* if, for any PPT $\mathcal{A}$ and for any attribute universe $\mathcal{U}$, $\mathbf{Adv}^{\mathrm{cmim}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda)$ is negligible in λ.

Selective Security. In the *selective game on a target attribute set* (the game of sel-cMiM attack), $\mathcal{A}$ declares S^* *before* $\mathcal{A}$ receives PK. $\mathbf{Exprmt}^{\text{sel-cmim}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda,\mathcal{U})$ defines the selective game. The *advantage* in the game of sel-cMiM attack is defined as

$$\mathbf{Adv}^{\text{sel-cmim}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda) \stackrel{\mathrm{def}}{=} \Pr[\mathbf{Exprmt}^{\text{sel-cmim}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda,\mathcal{U}) \text{ returns } \textsc{Win}].$$

PP-ABID is called *selectively secure against cMiM attacks* if, for any PPT $\mathcal{A}$ and for any $\mathcal{U}$, $\mathbf{Adv}^{\text{sel-cmim}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda)$ is negligible in λ.

4 Generic Conversions from ABKEM to ABID

In this section, we provide a generic conversion from a key-policy ABKEM to a prover-policy ABID. The conversion yields a challenge-and-response protocol of encapsulation-and-decapsulation. We show that KP-ABKEM only has to be OW-CCA secure for the obtained PP-ABID to be cMiM secure. A generic conversion from a ciphertext-policy ABKEM to a verifier-policy ABID is provided in a similar way (in the full version).

4.1 Generic Conversion from KP-ABKEM to PP-ABID

Let KP-ABKEM= (KEM.Setup, KEM.KeyGen, KEM.Encap, KEM.Decap) be a KP-ABKEM. Then PP-ABID= (Setup, KeyGen, Encap, Decap) is obtained as a challenge-and-response protocol of encapsulation-and-decapsulation. Figure 3 shows this conversion. Setup of PP-ABID uses KEM.Setup. KeyGen of PP-ABID uses KEM.KeyGen. The verifier V, given a public key PK and an attribute set S as input, invokes the encapsulation algorithm KEM.Encap on (PK, S). V gets a return (κ, ψ). V sends the encapsulation ψ to the prover P as a challenge message. P, given a public key PK and the secret key $\mathrm{SK}_\mathbb{A}$ as input, and receiving ψ as a message, invokes the decapsulation algorithm KEM.Decap on $(\mathrm{PK}, \mathrm{SK}_\mathbb{A}, \psi)$. P gets a return $\hat{\kappa}$. P sends the decapsulation $\hat{\kappa}$ to V as a response message. Finally, V, receiving $\hat{\kappa}$ as a message, verifies whether $\hat{\kappa}$ is equal to κ. If so, then V returns 1 and otherwise, 0.

Setup$(\lambda, \mathcal{U})$:	**KeyGen**$(\mathrm{PK}, \mathrm{MSK}, \mathbb{A})$:
$(\mathrm{PK}, \mathrm{MSK}) \leftarrow$ **KEM.Setup**$(\lambda, \mathcal{U})$	$\mathrm{SK}_{\mathbb{A}} \leftarrow$ **KEM.KeyGen**$(\mathrm{PK}, \mathrm{MSK}, \mathbb{A})$
Return $(\mathrm{PK}, \mathrm{MSK})$	Return $(\mathrm{SK}_{\mathbb{A}})$

$\mathbf{P}(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}})$:		$\mathbf{V}(\mathrm{PK}, S)$:
		$(\kappa, \psi) \leftarrow$ **KEM.Encap**(PK, S)
Receiving ψ as input:	$\xleftarrow{\psi}$	Send ψ to $\mathbf{P}$
$\hat{\kappa} \leftarrow$ **KEM.Decap**$(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}}, \psi)$		
Send $\hat{\kappa}$ to $\mathbf{V}$	$\xrightarrow{\hat{\kappa}}$	Receiving $\hat{\kappa}$ as input:
		If $\hat{\kappa} = \kappa$ then $b := 1$ else $b := 0$, Return b

Fig. 3. A generic conversion from KP-ABKEM to PP-ABID

Theorem 1. *If KP-ABKEM is OW-CCA secure, then the derived PP-ABID is cMiM secure. More precisely, for any given PPT adversary $\mathcal{A}$ on PP-ABID in the game of cMiM attack, and for any given attribute universe $\mathcal{U}$, there exists a PPT adversary $\mathcal{B}$ on KP-ABKEM in the OW-CCA game that satisfies the following tight reduction.*

$$\mathbf{Adv}^{cmim}_{\mathcal{A},\mathit{PP\text{-}ABID}}(\lambda) \leqslant \mathbf{Adv}^{ow\text{-}cca}_{\mathcal{B},\mathit{KP\text{-}ABKEM}}(\lambda).$$

Proof. Employing any given PPT cMiM adversary $\mathcal{A}$ on PP-ABID in Theorem 1, we construct a PPT OW-CCA adversary $\mathcal{B}$ on KP-ABKEM. The left side of Figure 4 shows the construction.

On input $(\mathrm{PK}, \mathcal{U})$, $\mathcal{B}$ initializes its inner state and invokes $\mathcal{A}$ on $(\mathrm{PK}, \mathcal{U})$. When $\mathcal{A}$ issues a key-extraction query for $\mathbb{A}$, $\mathcal{B}$ queries its key-generation oracle $\mathcal{KG}(\mathrm{PK}, \mathrm{MSK}, \cdot)$ for the answer for $\mathbb{A}$ and gets a reply $\mathrm{SK}_{\mathbb{A}}$. $\mathcal{B}$ reply $\mathrm{SK}_{\mathbb{A}}$ to $\mathcal{A}$. When $\mathcal{A}$ sends a challenge message $(\mathbb{A}, \psi)$ to a prover $\mathbf{P}$, $\mathcal{B}$ queries its decapsulation oracle $\mathcal{DEC}(\mathrm{PK}, \mathrm{SK}_{\cdot}, \cdot)$ for the answer for $(\mathbb{A}, \psi)$ and gets a reply $\hat{\kappa}$. $\mathcal{B}$ reply $\hat{\kappa}$ to $\mathcal{A}$. When $\mathcal{A}$ outputs a target attribute set S^*, $\mathcal{B}$ output S^* as its target attribute set. Then $\mathcal{B}$ receives a challenge ciphertext ψ^* from its challenger. When $\mathcal{A}$ queries $\mathbf{V}$ for a challenge message, $\mathcal{B}$ sends ψ^* to $\mathcal{A}$ as a challenge message. When $\mathcal{A}$ sends the response message $\hat{\kappa^*}$ to $\mathbf{V}$, $\mathcal{B}$ returns $\hat{\kappa^*}$ as its guess.

The view of $\mathcal{A}$ in $\mathcal{B}$ is the same as the real view of $\mathcal{A}$. If $\mathcal{A}$ wins, then $\mathcal{B}$ wins. Hence the inequality in Theorem 1 holds. □

4.2 Discussion

Selective Security. The right side of Figure 4 shows the construction of $\mathcal{B}$ in the game of selective S^*. The inequality of advantages becomes

$$\mathbf{Adv}^{\text{sel-cmim}}_{\mathcal{A},\text{PP-ABID}}(\lambda) \leqslant \mathbf{Adv}^{\text{ow-sel-cca}}_{\mathcal{B},\text{KP-ABKEM}}(\lambda).$$

Resettable Security. We note that the derived PP-ABID is prover-resettable in the sense in [4] because underlying KP-ABKEM has the OW-CCA security. PP-ABID is also verifier-resettable because PP-ABID consists of two rounds interaction.

$\mathcal{B}(\mathrm{PK},\mathcal{U})$: //Adaptive S^*
//**Set up**
Initialize inner state, Invoke $\mathcal{A}$ on $(\mathrm{PK},\mathcal{U})$
//**Answering $\mathcal{A}$'s Queries**
When $\mathcal{A}$ issues a key-ext. query for $\mathbb{A}$
$\mathrm{SK}_{\mathbb{A}} \leftarrow \mathcal{KG}(\mathrm{PK}, \mathrm{MSK}, \mathbb{A})$
Reply $\mathrm{SK}_{\mathbb{A}}$ to $\mathcal{A}$
When $\mathcal{A}$ sends a chal. msg. $(\mathbb{A}, \psi)$ to **P**
$\hat{\kappa} \leftarrow \mathcal{DEC}(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}}, \psi)$
Send $\hat{\kappa}$ to $\mathcal{A}$ as the res. msg.
When $\mathcal{A}$ outputs a target attribute set S^*
Output S^* as its target attribute set
Receive ψ^* as a chal. ciphertext
When $\mathcal{A}$ queries **V** for a chal. msg.
Send ψ^* to $\mathcal{A}$ as a chal. msg.
When $\mathcal{A}$ sends the res. msg. $\hat{\kappa^*}$ to **V**
Return $\hat{\kappa^*}$

$\mathcal{B}(\mathrm{PK},\mathcal{U})$: //Selective S^*
//**Set up**
Initialize inner state, Invoke $\mathcal{A}$ on $(\lambda,\mathcal{U})$
//**Answering $\mathcal{A}$'s Queries**
$S^* \leftarrow \mathcal{A}(\lambda,\mathcal{U})$
Output S^* as its target attribute set
Receive ψ^* as a chal. ciphertext
Give PK to $\mathcal{A}$
When $\mathcal{A}$ issues a key-ext query for $\mathbb{A}$
$\mathrm{SK}_{\mathbb{A}} \leftarrow \mathcal{KG}(\mathrm{PK}, \mathrm{MSK}, \mathbb{A})$
When $\mathcal{A}$ sends a chal. msg. $(\mathbb{A}, \psi)$ to **P**
$\hat{\kappa} \leftarrow \mathcal{DEC}(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}}, \psi)$
Send $\hat{\kappa}$ to $\mathcal{A}$ as the res. msg.
When $\mathcal{A}$ queries **V** for a chal. msg.
Send ψ^* to $\mathcal{A}$ as a chal. msg.
When $\mathcal{A}$ sends the res. msg. $\hat{\kappa^*}$ to **V**
Return $\hat{\kappa^*}$

Fig. 4. A one-way CCA adversary $\mathcal{B}$ on `KP-ABKEM` that employs a given cMiM adversary $\mathcal{A}$ on the `PP-ABID`. The left side: the case of adaptive S^*; the right side: the case of selective S^*.

5 Concrete Constructions of ABKEM

In this section, we construct a concrete KP-ABKEM that is OW-sel-CCA secure. Using the algebraic trick of Boneh and Boyen [3] and Kiltz [11], we build an enhanced version, `KP-ABKEM`, of the KP-ABKEM of Ostrovsky, Sahai and Waters [13] (OSW, for short). Then we obtain our concrete `PP-ABID` by applying the generic conversion. (Our concrete `CP-ABKEM` and `VP-ABID` is described in Appendix C).

5.1 Our Enhanced OSW KP-ABKEM and PP-ABID

The construction of our concrete `KP-ABKEM` is described in Figure 5. We only explain the enhanced part from the original [13]. We indicate the part of the original scheme by the index: $_{\mathrm{cpa}}$. In Setup, a second component $\alpha_2 \in \mathbb{Z}_p$ is added to the master secret key $\mathrm{MSK}_{\mathrm{cpa}}$. Also, the corresponding $Y_2 := e(g,g)^{\alpha_2 b}$ and a hash key η is added to the public key $\mathrm{PK}_{\mathrm{cpa}}$. In KeyGen, components in $\mathrm{SK}_{\mathrm{cpa},\mathbb{A}}$ are doubled reflecting the index 2 (but randomness is chosen independently of index 1). So computational cost for KeyGen is doubled. In Encap, a temporal KEM key κ_2 is generated in the same way as κ_1. Next, a hash value $\tau \leftarrow H_\eta(\psi_{\mathrm{cpa}})$ and a *check sum* $d := \kappa_1^\tau \kappa_2$ are computed. Then $(\kappa, \psi) := (\kappa_1, (\psi_{\mathrm{cpa}}, d))$ is a new KEM key and encapsulation. In Decap, first, $\mathrm{Decap}_{\mathrm{cpa}}$ is executed twice for index 1 and 2 to yield $\hat{\kappa_1}$ and $\hat{\kappa_2}$. Then, whether ψ_{cpa} is a consistent ciphertext and $(e(g,g), Y_1^\tau Y_2, e(C',g), d)$ is a Diffie-Hellman tuple is verified. These two conditions are verified by one equation $\hat{\kappa_1}^\tau \hat{\kappa_2} = d$, though the verification equation overlooks inconsistent ψ_{cpa} only with a negligible probability. Finally, $\hat{\kappa} := \hat{\kappa_1}$ is returned only when the verification equation holds.

Setup$(\lambda, \mathcal{U})$:
$(p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathbf{Grp}(\lambda)$
For $x = 1$ to u: $T_x \leftarrow \mathbb{G}$
$b \leftarrow \mathbb{Z}_p, B := g^b, \alpha_1, \alpha_2 \leftarrow \mathbb{Z}_p$
$Y_1 := e(g,g)^{\alpha_1 b}, Y_2 := e(g,g)^{\alpha_2 b}$
$\eta \leftarrow HKey_\lambda$
$\mathrm{PK} := (g, T_1, \ldots, T_u, B, Y_1, Y_2, \eta)$
$\mathrm{MSK} := (\alpha_1, \alpha_2)$
Return $(\mathrm{PK}, \mathrm{MSK})$

KeyGen$(\mathrm{PK}, \mathrm{MSK}, \mathbb{A} = (M, \rho))$:
For $k = 1, 2$: For $j = 2$ to n: $v_{k,j} \leftarrow \mathbb{Z}_p$
For $k = 1, 2$: $\boldsymbol{v}_k := (\alpha_k, v_{k,2}, \ldots, v_{k,n})$
For $i = 1$ to l: $\lambda_{k,i} := \boldsymbol{v}_k \cdot M_i$
For $k = 1, 2$: For $i = 1$ to l:
$r_{k,i} \leftarrow \mathbb{Z}_p,\ K_{k,i} := B^{\lambda_{k,i}} T_{\rho(i)}^{r_{k,i}},\ L_{k,i} := g^{r_{k,i}}$
$\mathrm{SK}_{\mathbb{A}} := (((K_{k,i}, L_{k,i}); i = 1, \ldots, l); k = 1, 2)$
Return $\mathrm{SK}_{\mathbb{A}}$

Encap(PK, S):
$s \leftarrow \mathbb{Z}_p, C' = g^s$, For $x \in S : C_x = T_x^s$
$\psi_{\mathrm{cpa}} := (S, C', (C_x; x \in S))$
$\tau \leftarrow H_\eta(\psi_{\mathrm{cpa}})$
For $k = 1, 2$: $\kappa_k := Y_k^s;\ d := \kappa_1^\tau \kappa_2$
$(\kappa, \psi) := (\kappa_1, (\psi_{\mathrm{cpa}}, d))$
Return (κ, ψ)

Decap$(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}}, \psi)$:
If $S \notin \mathbb{A}$ Return $\hat{\kappa} := \perp$
else $I_S := \rho^{-1}(S)$
$\{\omega_i; i \in I_S\} \leftarrow \mathbf{Recon}(I_S, M)$
For $k = 1, 2$:
$\hat{\kappa_k} := \prod_{i \in I_S} (e(K_{k,i}, C')/e(L_{k,i}, C_{\rho(i)}))^{\omega_i}$
$\tau \leftarrow H_\eta(\psi_{\mathrm{cpa}})$
If $\hat{\kappa_1}^\tau \hat{\kappa_2} \neq d$ then $\hat{\kappa} := \perp$ else $\hat{\kappa} := \hat{\kappa_1}$
Return $\hat{\kappa}$

Fig. 5. Our concrete KP-ABKEM (an enhanced OSW KP-ABKEM)

Theorem 2. *If the computational bilinear Diffie-Hellman assumption with gap on target group holds, and an employed hash function family has target collision resistance, then our KP-ABKEM is OW-sel-CCA secure. More precisely, for any given PPT adversary $\mathcal{A}$ on KP-ABKEM in the OW-sel-CCA game and for any given attribute universe $\mathcal{U}$, there exist a PPT adversary $\mathcal{B}$ on $(e, \mathbb{G}, \mathbb{G}_T)$ in the computational BDH game with gap on $\mathbb{G}_T$ and a PPT target collision finder $\mathcal{CF}$ on Hfam_λ that satisfy the following tight reduction.*

$$\mathbf{Adv}^{ow\text{-}sel\text{-}cca}_{\mathcal{A}, KP\text{-}ABKEM}(\lambda) \leqslant \mathbf{Adv}^{c\text{-}bdh\text{-}gap}_{\mathcal{B}, (e, \mathbb{G}, \mathbb{G}_T)}(\lambda) + \mathbf{Adv}^{tcr}_{\mathcal{CF}, \mathrm{Hfam}_\lambda}(\lambda).$$

5.2 Proof for Theorem 2

Using any given OW-sel-CCA adversary $\mathcal{A}$ as a subroutine, we construct a PPT solver $\mathcal{B}$ of the problem of the computational bilinear Diffie-Hellman assumption with gap on target group, as follows.

Set Up. $\mathcal{B}$ is given a random instance of the problem, $g, A = g^a, B = g^b, C = g^c$, as input. $\mathcal{B}$ initializes its inner state. $\mathcal{B}$ chooses an attribute universe $\mathcal{U} = \{1, \ldots, u\}$ at random. $\mathcal{B}$ invokes $\mathcal{A}$ on input $(\lambda, \mathcal{U})$.

In return, $\mathcal{B}$ receives a target attribute set $S^* \in \mathcal{U}$ from $\mathcal{A}$, For each $x = 1, \ldots, u$, $\mathcal{B}$ puts each component T_x of PK as

$$\text{If } x \in S^* \text{ then } t_x \leftarrow \mathbb{Z}_p, T_x := g^{t_x} \text{ else } \theta_x, \eta_x \leftarrow \mathbb{Z}_p \text{ s.t. } \theta_x \neq 0, T_x := B^{\theta_x} g^{\eta_x}.$$

Here, in else case, we have implicitly set $t_x := b\theta_x + \eta_x$. $\mathcal{B}$ sets $Y_1 := e(A, B) = e(g,g)^{ab}$ and $\mathrm{PK}_{\mathrm{cpa}} := (g, T_1, \ldots, T_u, B, Y_1)$. Here we have implicitly set $\alpha_1 := a$.

Challenge ciphertext are computed as follows (we implicitly set $s^* = c$):

$$\psi^*_{\mathrm{cpa}} := (S^*, C'^* = g^{s^*} := C, (C^*_x := C^{t_x}, x \in S^*)).$$

Then a public key PK and a whole challenge ciphertext ψ^* is computed as

$$\eta \leftarrow HKey_\lambda,\ \tau^* \leftarrow H_\eta(\psi^*_{\mathrm{cpa}}),\ \mu \leftarrow \mathbb{Z}_p,\ Y_2 := e(B,g)^\mu / Y_1^{\tau^*},$$
$$\mathrm{PK} := (\mathrm{PK}_{\mathrm{cpa}}, Y_2, \eta),\ d^* := e(B, C'^*)^\mu,\ \psi^* := (\psi^*_{\mathrm{cpa}}, d^*).$$

Here we have an implicit relation $\alpha_2 b = b\mu - \alpha_1 b\tau^*, b \neq 0$. That is,

$$\alpha_2 = \mu - \alpha_1 \tau^*. \tag{1}$$

$\mathcal{B}$ inputs (PK, ψ^*) to $\mathcal{A}$.

Answering $\mathcal{A}$'s Queries. (1) Key-Extraction Queries. When $\mathcal{A}$ issues a key-extraction query for an attribute set $\mathbb{A} = (M, \rho)$, where M is of size $l \times n$, $\mathcal{B}$ has to reply a corresponding secret key $\mathrm{SK}_{\mathbb{A}}$.

$\mathcal{B}$ computes a vector $\boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{Z}_p^n$ such that $w_1 = 1$ and for all $i \in \rho^{-1}(S^*), \boldsymbol{w} \cdot M_i = 0$. Note here that $S^* \notin \mathbb{A}$, so such $\boldsymbol{w}$ surely exists. $\mathcal{B}$ chooses random values $u_{1,1}, \ldots, u_{1,n} \in \mathbb{Z}_p$ and put $\boldsymbol{u}_1 := (u_{1,1}, \ldots, u_{1,n})$. Then we implicitly set $\boldsymbol{v}_1 := \boldsymbol{u}_1 + (a - u_{1,1})\boldsymbol{w}$.

Here for each $i = 1, \ldots, l$, $\mathcal{B}$ can compute $g^{\lambda_{1,i}}$ as $g^{M_i \cdot \boldsymbol{v}_1} = g^{M_i \cdot (\boldsymbol{u}_1 - u_{1,1}\boldsymbol{w})} A^{M_i \cdot \boldsymbol{w}}$. Then $\mathcal{B}$ computes the index 1 components of SK_S as

$$\textsf{For } i = 1 \textsf{ to } l : \textsf{If } i \in \rho^{-1}(S^*) \textsf{ then } r_{1,i} \leftarrow \mathbb{Z}_p, K_{1,i} := B^{M_i \cdot \boldsymbol{u}_1} T_{\rho(i)}^{r_{1,i}}, L_{1,i} := g^{r_{1,i}}$$
$$\textsf{else } r'_{1,i} \leftarrow \mathbb{Z}_p, K_{1,i} := (g^{\lambda_{1,i}})^{-\eta_{\rho(i)}/\theta_{\rho(i)}} (B^{\theta_{\rho(i)}} g^{\eta_{\rho(i)}})^{r'_{1,i}}, L_{1,i} := (g^{\lambda_{1,i}})^{-1/\theta_{\rho(i)}} g^{r'_{1,i}}.$$

Here, in else case, we implicitly set $r_{1,i} := r'_{1,i} - \lambda_{1,i}/\theta_{\rho(i)}$.

Now $\mathcal{B}$ has to compute the index 2 components $K_{2,i}, L_{2,i}$ for $i = 1, \ldots, l$. To do so, $\mathcal{B}$ chooses random values $u_{2',1}, \ldots, u_{2',n}, r_{2',i}$(or $r'_{2',i}$) $\in \mathbb{Z}_p$ and computes $K_{2',i}, L_{2',i}, i = 1, \ldots, l$ *just in the same way* as to the index 1. Then $\mathcal{B}$ converts them as follows:

$$K_{2,i} := B^{M_{i,1}\mu} (K_{2',i})^{-\tau^*}, L_{2,i} := (L_{2',i})^{-\tau^*}, i = 1, \ldots, l.$$

Then $\mathcal{B}$ replies $\mathrm{SK}_{\mathbb{A}} = (((K_{k,i}, L_{k,i}); i = 1, \ldots, l); k = 1, 2)$ to $\mathcal{A}$.

(2) Decapsulation Queries. When $\mathcal{A}$ issues a decapsulation query for $(\mathbb{A}, \psi = (\psi_{\mathrm{cpa}}, d))$ (where ψ_{cpa} is about S), $\mathcal{B}$ has to reply the decapsulation $\hat{\kappa}$ to $\mathcal{A}$. To do so, $\mathcal{B}$ computes as follows. (Note that the oracle $\mathcal{DDH}_{\mathbb{G}_T}$ is accessed.)

$\textsf{If } S \notin \mathbb{A} \textsf{ then } \hat{\kappa} := \perp$
$\textsf{else If } \mathbf{Verify}(\mathrm{PK}_{\mathrm{cpa}}, \psi_{\mathrm{cpa}}) = \textsc{False} \textsf{ then } \hat{\kappa} := \perp$
$\textsf{else } \tau \leftarrow H_\eta(\psi_{\mathrm{cpa}})\ \textsf{If } \mathcal{DDH}_{\mathbb{G}_T}(e(g,g), Y_1^\tau Y_2, e(C', g), d) = \textsc{False} \textsf{ then } \hat{\kappa} := \perp$
$\textsf{else If } \tau = \tau^* \textsf{ then Abort } //\text{Call this case } \textsc{Abort}$
$\textsf{else } \hat{\kappa} := (d/e(B, C')^\mu)^{1/(\tau - \tau^*)}$

where **Verify** is the following PPT algorithm to check consistency of ψ_{cpa}:

$$\begin{aligned}\mathbf{Verify}(\mathrm{PK}_{\mathrm{cpa}}, \psi_{\mathrm{cpa}}) :&\mathsf{For}\ x \in S :\ \mathsf{If}\ e(T_x, C') \neq e(C_x, g)\ \mathsf{then\ Return}\ \textsc{False}\\ &\mathsf{Return}\ \textsc{True}.\end{aligned}$$

Guess. When $\mathcal{A}$ returns $\mathcal{A}$'s guess $\hat{\kappa^*}$, $\mathcal{B}$ returns $Z := \hat{\kappa^*}$ as $\mathcal{B}$'s guess.

$\mathcal{B}$ can perfectly simulate the real view of $\mathcal{A}$ until the case Abort happens. To see why, we prove the following claims. (The proofs will be in the full version.)

Claim 1. *The reply $SK_{\mathbb{A}}$ to a key-extraction query is a perfect simulation.* □

Claim 2. *The reply $\hat{\kappa}$ to a decapsulation query is a simulation that is computationally indistinguishable from a real, until the case* Abort *happens.* □

Claim 3. *The challenge ciphertext $\psi^* = (\psi^*_{cpa}, d^*)$ is correctly distributed.* □

Now we are ready to evaluate the advantage of $\mathcal{B}$ in the OW-sel-CCA game. First, the following claim holds. (The proof will be described in the full version.)

Claim 4. *The probability that* Abort *occurs is negligible in λ. More precisely, the following equality holds: $Pr[\textsc{Abort}] = \mathbf{Adv}^{tcr}_{\mathcal{CF},Hfam_\lambda}(\lambda)$.* □

By definition, $\mathcal{A}$ wins in the OW-sel-CCA game if and only if $\hat{\kappa}^*$ is correctly guessed. That is, $\hat{\kappa^*} = Y_1^{s^*} = e(g,g)^{abs^*} = e(g,g)^{abc}$. This is the definition that $\mathcal{B}$ succeeds in computing the answer for the given instance (g, A, B, C).

Therefore, the probability that $\mathcal{B}$ wins is equal to the probability that $\mathcal{A}$ wins and Abort never occurs. So we have:

$$\Pr[\mathcal{B}\ \text{wins}] = \Pr[(\mathcal{A}\ \text{wins}) \wedge (\neg\textsc{Abort})] \geqslant \Pr[\mathcal{A}\ \text{wins}] - \Pr[\textsc{Abort}].$$

Substituting advantages and using the equality in Claim 4, we have:

$$\mathbf{Adv}^{\text{c-bdh-gap}}_{\mathcal{B},(e,\mathbb{G},\mathbb{G}_T)}(\lambda) \geqslant \mathbf{Adv}^{\text{ow-sel-cca}}_{\mathcal{A},\texttt{KP-ABKEM}}(\lambda) - \mathbf{Adv}^{\text{tcr}}_{\mathcal{CF},Hfam_\lambda}(\lambda).$$

This is what we should prove. □

Theorem 3 (Corollary to Theorem 1 and 2). *Our PP-ABID is selectively secure against cMiM attacks. More precisely, the following inequality holds:*

$$\mathbf{Adv}^{sel\text{-}cmim}_{\mathcal{A},PP\text{-}ABID}(\lambda) \leqslant \mathbf{Adv}^{c\text{-}bdh\text{-}gap}_{\mathcal{B},(e,\mathbb{G},\mathbb{G}_T)}(\lambda) + \mathbf{Adv}^{tcr}_{\mathcal{CF},Hfam_\lambda}(\lambda).$$

Figure 6 shows an interaction of our PP-ABID.

$\mathbf{P}(\mathrm{PK} = (g, T_1, \ldots, T_u, B, Y_1, Y_2, \eta),$
$\mathrm{SK}_{\mathbb{A}} = (((K_{k,i} = B^{\lambda_{k,i}} T_{\rho(i)}^{r_{k,i}},$
$L_{k,i} = g^{r_{k,i}});$
$i = 1, \ldots, l); k = 1, 2)):$

$\mathbf{V}(\mathrm{PK}, S)$:
$s \leftarrow \mathbb{Z}_p, C' := g^s$, For $x \in S : C_x := T_x^s$
$\psi_{\mathrm{cpa}} := (S, C', (C_x; x \in S))$
$\tau \leftarrow H_\eta(\psi_{\mathrm{cpa}})$
For $k = 1, 2$: $\kappa_k := Y_k^s$; $d := \kappa_1^\tau \kappa_2$
$(\kappa, \psi) := (\kappa_1, (\psi_{\mathrm{cpa}}, d))$

$\xleftarrow{\psi}$ Send ψ to $\mathbf{P}$

Receiving ψ as input:
If $S \notin \mathbb{A}$ then $\hat{\kappa} := \perp$
else $\tau \leftarrow H_\eta(\psi_{\mathrm{cpa}}), I_S := \rho^{-1}(S)$
$\{\omega_i; i \in I_S\} \leftarrow \mathbf{Recon}(I_S, M)$
For $k = 1, 2$:
$\hat{\kappa_k} := \prod_{i \in I_S} (e(K_{K,i}, C')$
$/e(L_{K,i}, C_{\rho(i)}))^{\omega_i}$
If $\hat{\kappa_1}^\tau \hat{\kappa_2} \neq d$ then $\hat{\kappa} := \perp$ else $\hat{\kappa} := \hat{\kappa_1}$
Send $\hat{\kappa}$ to $\mathbf{V}$

$\xrightarrow{\hat{\kappa}}$ Receiving $\hat{\kappa}$ as input:
If $\hat{\kappa} = \kappa$ then $b := 1$ else $b := 0$, Return b

Fig. 6. An interaction of our concrete PP-ABID

5.3 Discussion

Anonymity. Consider the following experiment in Figure 7. (In the experiment, an adversary $\mathcal{A}$ interacts with $\mathbf{P}(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}_b})$ as a verifier with S^*.)

We say that PP-ABID have *anonymity* if, for any PPT $\mathcal{A}$ and for any $\mathcal{U}$, $\mathbf{Adv}^{\mathrm{anonym}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda) \stackrel{\mathrm{def}}{=} |\Pr[\mathbf{Exprmt}^{\mathrm{anonym}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda, \mathcal{U}) \text{ returns } \textsc{Win}] - 1/2|$ is negligible in λ. Our concrete PP-ABID possesses the anonymity because, in the case that ψ is inconsistent, the randomness in $\mathrm{SK}_{\mathbb{A}_b}$ is *not canceled out correctly* in the computation of a response message $\hat{\kappa_b}$.

$\mathbf{Exprmt}^{\mathrm{anonym}}_{\mathcal{A},\mathtt{PP\text{-}ABID}}(\lambda, \mathcal{U})$:
$(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathbf{Setup}(\lambda, \mathcal{U}), (\mathbb{A}_0, \mathbb{A}_1, S^*) \leftarrow \mathcal{A}(\mathrm{PK})$
s.t. $(S^* \in \mathbb{A}_0 \wedge S^* \in \mathbb{A}_1) \vee (S^* \notin \mathbb{A}_0 \wedge S^* \notin \mathbb{A}_1)$
$\mathrm{SK}_{\mathbb{A}_0} \leftarrow \mathbf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, \mathbb{A}_0), \mathrm{SK}_{\mathbb{A}_1} \leftarrow \mathbf{KeyGen}(\mathrm{PK}, \mathrm{MSK}, \mathbb{A}_1)$
$b \leftarrow \{0, 1\}, \hat{b} \leftarrow \mathcal{A}^{\mathbf{P}(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}_b})}(\mathrm{PK}, \mathrm{SK}_{\mathbb{A}_0}, \mathrm{SK}_{\mathbb{A}_1})$
If $b = \hat{b}$ Return Win else Return Lose

Fig. 7. The anonymity experiment of an adversary $\mathcal{A}$ on PP-ABID

Large Universe Case. If the attribute universe $\mathcal{U}$ is large, we have to modify our concrete schemes to make security reductions in time polynomial in λ. As is proposed by Waters [15], we use for $x \in \mathcal{U}$ a hashed value $H(x)$ instead of T_x (and hence T_x is removed from PK). Although the resulting schemes are proved to be secure only in the random oracle model, we get relief from rewriting the public key PK each time when a new attribute x is added.

Exiting the Gap Assumption. Instead of the oracle $\mathcal{DDH}_{\mathbb{G}_T}$, we can apply the twin Diffie-Hellman trapdoor test of Cash, Kiltz and Shoup [8] in the security

proofs. In compensation, the resulting schemes become to have a twice as long secret key and twice as much computational cost in decapsulation.

Security against Adaptive Target. To attain the adaptive security in the OW-CCA game, we can apply our enhancing technique to the dual system encryption of Lewko, Okamoto, Sahai, Takashima and Waters [10] in the random oracle model.

6 Conclusions

We proposed PP-ABID and VP-ABID. We established a design principle. We constructed concrete KP-ABKEM and CP-ABKEM with the OW-CCA security. Finally, we obtained concrete PP-ABID and VP-ABID. Functional tickets and functional gates are realized by those PP-ABID and VP-ABID, respectively.

References

1. Anada, H., Arita, S.: Identification Schemes from Key Encapsulation Mechanisms. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 59–76. Springer, Heidelberg (2011)
2. Beimel, A.: Secure schemes for secret sharing and key distribution. Ph.D. thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
3. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
4. Bellare, M., Fischlin, M., Goldwasser, S., Micali, S.: Identification Protocols Secure against Reset Attacks. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 495–511. Springer, Heidelberg (2001)
5. Bellare, M., Palacio, A.: GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 162–177. Springer, Heidelberg (2002)
6. Baek, J., Safavi-Naini, R., Susilo, W.: Efficient Multi-receiver Identity-Based Encryption and Its Application to Broadcast Encryption. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 380–397. Springer, Heidelberg (2005)
7. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty Computation from Threshold Homomorphic Encryption. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 280–300. Springer, Heidelberg (2001)
8. Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman Problem and Applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008); Full version available at Cryptology ePrint Archive, 2008/067, `http://eprint.iacr.org/`
9. Gennaro, R.: Multi-trapdoor Commitments and their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 220–236. Springer, Heidelberg (2004)
10. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010); Full version available at IACR Cryptology ePrint Archive, 2010/110, `http://eprint.iacr.org/`

11. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
12. Naor, M., Yung, M.: Universal One-Way Hash Functions and their Cryptographic Applications. In: The 21st Symposium on Theory of Computing 1989, pp. 33–43. Association for Computing Machinery, New York (1989)
13. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security, pp. 195–203 (2007)
14. Schnorr, C.P.: Efficient Identification and Signatures for Smart Cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)
15. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011), Full version available at IACR Cryptology ePrint Archive,2008/290, `http://eprint.iacr.org/`
16. Yamada, S., Attrapadung, N., Santoso, B., Schuldt, J.C.N., Hanaoka, G., Kunihiro, N.: Verifiable Predicate Encryption and Applications to CCA Security and Anonymous Predicate Authentication. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 243–261. Springer, Heidelberg (2012)
17. Yilek, S.: Resettable Public-Key Encryption: How to Encrypt on a Virtual Machine. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 41–56. Springer, Heidelberg (2010)

A Verifier-Policy ABID

We define a notion of *verifier-policy* attribute-based identification (VP-ABID).

Scheme. `VP-ABID` consists of four PPT algorithms (Setup, KeyGen, P, V).

Setup$(\lambda, \mathcal{U}) \to (\mathbf{PK}, \mathbf{MSK})$**.** Setup takes as input the security parameter λ and the attribute universe $\mathcal{U}$. It outputs public a public key PK and a master secret key MSK.

KeyGen$(\mathbf{PK}, \mathbf{MSK}, S) \to \mathbf{SK}_S$**.** A key-generation algorithm KeyGen takes as input the public key PK, the master secret key MSK and an attribute set S. It outputs a secret key SK_S corresponding to S.

P$(\mathbf{PK}, \mathbf{SK}_S)$ **and V**$(\mathbf{PK}, \mathbb{A})$**.** P and V are interactive algorithms called a *prover* and a *verifier*, respectively, which are defined in a similar way as in Section 3.1. A cMiM attack on `VP-ABID` is defined in a similar way as in Section 3.2. `CP-ABKEM`=(**Setup**,**KeyGen**,**Encap**,**Decap**) is defined in a dual manner to `KP-ABKEM` on an access structure $\mathbb{A}$ and an attribute set S. Then a generic conversion from `CP-ABKEM` to `VP-ABID` is defined in a similar way as in Section 4.1.

Theorem 4. *If `CP-ABKEM` is OW-CCA secure, then the derived `VP-ABID` is cMiM secure.*

B Computational Parallel Bilinear Diffie-Hellman Exponent Assumption with Gap on Target Group

Let $a, s, b_1, \ldots, b_q \in \mathbb{Z}_p$, all of which is not zero, be chosen at random. Let $\boldsymbol{y} := (g, g^s, g^a, \ldots, g^{(a^q)}, g^{(a^{q+2})}, \ldots, g^{(a^{2q})}, \forall_{1 \leqslant j \leqslant q} g^{sb_j}, g^{a/b_j}, \ldots, g^{(a^q/b_j)}, g^{(a^{q+2}/b_j)}, \ldots, g^{(a^{2q}/b_j)}, \forall_{1 \leqslant j,k \leqslant q, k \neq j} g^{asb_k/b_j}, \ldots, g^{a^q sb_k/b_j})$. Then our new assumption says it is at most with a negligible probability in λ that, for any PPT algorithm $\mathcal{B}$ given input $\boldsymbol{y}$ (parametrized by q), to output $Z = e(g,g)^{a^{q+1}s} \in \mathbb{G}_T$, even with the aid of the oracle $\mathcal{DDH}_{\mathbb{G}_T}(\cdot,\cdot,\cdot,\cdot)$.

C Our Enhanced Waters CP-ABKEM and VP-ABID

We can build an enhanced version, `CP-ABKEM`, of the CP-ABKEM of Waters [15]. Then we can obtain our concrete `VP-ABID` as in Figure 8.

$\mathbf{P}(\mathrm{PK} = (g, T_1, \ldots, T_u, A, Y_1, Y_2, \eta),$
$\mathrm{SK}_S = ((K_k = g^{\alpha_k} A^{l_k}, L_k = g^{l_k},$
$(K_{k,x} = T_x^{l_k}; x \in \mathcal{S}); k = 1, 2))):$

$\mathbf{V}(\mathrm{PK}, \mathbb{A}):$
$s \leftarrow \mathbb{Z}_p$, For $j = 2$ to n: $v_j \leftarrow \mathbb{Z}_p$,
$\boldsymbol{v} := (s, v_2, \ldots, v_n)$
For $i = 1$ to l: $\lambda_i := \boldsymbol{v} \cdot M_i$, $r_i \leftarrow \mathbb{Z}_p$
$C' := g^s$
For $i = 1$ to l: $C_i := A^{\lambda_i} T_{\rho(i)}^{-r_i}, D_i := g^{r_i}$
$\psi_{\mathrm{cpa}} := (\mathbb{A}, C', ((C_i, D_i); i = 1, \ldots, l))$
$\tau \leftarrow H_\eta(\psi_{\mathrm{cpa}})$
For $k = 1, 2$: $\kappa_k := Y_k^s; d := \kappa_1^\tau \kappa_2$
$(\kappa, \psi) := (\kappa_1, (\psi_{\mathrm{cpa}}, d))$

Receiving ψ as input:
If $S \notin \mathbb{A}$ then $\hat{\kappa} := \perp$
else $\tau \leftarrow H_\eta(\psi_{\mathrm{cpa}}), I_S := \rho^{-1}(S)$
$\{\omega_i; i \in I_S\} \leftarrow \mathbf{Recon}(I_S, M)$
For $k = 1, 2 : \hat{\kappa_k} := e(K_k, C')/$
$\prod_{i \in I_S} (e(L_k, C_i) e(K_{k,\rho(i)}, D_i))^{\omega_i}$
If $\hat{\kappa_1}^\tau \hat{\kappa_2} \neq d$ then $\hat{\kappa} := \perp$ else $\hat{\kappa} := \hat{\kappa_1}$
Send $\hat{\kappa}$ to $\mathbf{V}$

$\xleftarrow{\psi}$ Send ψ to $\mathbf{P}$

$\xrightarrow{\hat{\kappa}}$ Receiving $\hat{\kappa}$ as input:
If $\hat{\kappa} = \kappa$ then $b := 1$ else $b := 0$, Return b

Fig. 8. An interaction of our concrete `VP-ABID`

Theorem 5. *If the computational q-parallel bilinear Diffie-Hellman exponent assumption with gap on target group holds, and an employed hash function family has target collision resistance, then our `CP-ABKEM` is OW-sel-CCA secure with a challenge matrix of size $l^* \times n^*$, $l^*, n^* \leqslant q$.*

Theorem 6 (Corollary to Theorem 4 and 5). *Our `VP-ABID` is selectively secure against cMiM attacks.*

The proof of Theorem 5 will be described in the full version.