

# Efficient Chosen-Ciphertext Security from Selective-ID Secure Identity-Based Key Encapsulation

Jonas Schrieb\*

University of Paderborn, Germany  
jonas@uni-paderborn.de

**Abstract.** A chosen-ciphertext secure key encapsulation mechanism (KEM)—“constructed directly from identity-based techniques” by Boyen, Mei and Waters—is investigated. A transformation is given that obtains chosen-ciphertext security from arbitrary chosen-plaintext secure *partitioned* identity-based KEMs analogously. Further extensions and their occurrence in literature are briefly mentioned.

## 1 Introduction

Designing encryption schemes secure against chosen-ciphertext attacks is a hot topic in cryptographic research. In such schemes, an attacker cannot learn anything about the message underlying a ciphertext  $C^*$ , even if she gets decryptions for any ciphertext  $C \neq C^*$  she likes. This security notion is useful, e. g., in cryptographic protocols that should resist active attackers or if non-malleability of ciphertexts<sup>1</sup> is needed. An apparently weaker notion is security against chosen-plaintext attacks. Here, no such decryptions are given.

Identity-based encryption (IBE) is a variant of public-key cryptography where the public key of each user can be deduced from some public parameters and the user’s “identity”. Secret keys are issued by a trusted third party knowing a secret corresponding to the public parameters. Despite its direct applications in simplifying public-key management, another celebrated feature of IBE is the possibility to obtain chosen-ciphertext secure public-key schemes from only chosen-plaintext secure identity-based schemes.

The transformation of Canetti, Halevi and Katz [6] is applicable to *any* chosen-plaintext secure IBE. Their construction results from a clever combination with one-time signatures. Unfortunately, this introduces a computational overhead and enlarges ciphertexts compared to the original IBE. Boyen, Mei and Waters [4] optimize the above construction when applied to specific IBEs. They directly define two very efficient chosen-ciphertext secure public-key schemes (“BMW-PKE” and “BMW-KEM”) that are reminiscent of the original IBEs (Waters’ IBE [12] and Boneh-Boyen’s IBE [2, §4]), but do not result from a generic transformation rule. Specific properties of the “underlying” IBEs are exploited to avoid the overhead of a signature scheme. Abe, Cui, Imai and Kiltz [1, §5.3] analyze the connection of BMW-PKE to Waters’ IBE and distill the specific properties of the latter guaranteeing security of the former. Any scheme with these properties can be transformed efficiently into a chosen-ciphertext secure public-key scheme. Their transformation is as efficient as the “Waters’-IBE-to-BMW-PKE” construction but broadens its applicability to other IBEs.

Boneh-Boyen’s IBE has weaker security properties than Water’s IBE (selective-identity as opposed to adaptive-identity security) and gives only a key encapsulation mechanism BMW-KEM. That is a restricted form of encryption, only capable of encrypting randomly generated session keys instead of arbitrary messages chosen by the sender. It is a natural question whether one can obtain results analogous to that of Abe et al.: a definition of “properties like Boneh-Boyen’s IBE” and an efficient transformation from those schemes into a chosen-ciphertext secure KEM. As special case, this transformation applied to Boneh-Boyen’s IBE should give BMW-KEM. Actually, Abe et al. expect this to be unlikely [1, §5.4]. In this paper, their conjecture will be refuted. The “Boneh-Boyen’s-IBE-to-BMW-KEM” construction will be generalized in a similar (and practical) way, as Abe et al. do for “Waters’-IBE-to-BMW-PKE”.

\* Supported by a grant from the International Graduate School “Dynamic Intelligent Systems”, Univ. of Paderborn.

<sup>1</sup> E. g., if in a private auction Alice encrypts “My bid is  $n$ .” with the auctioneer’s public key it should be hard for Eve to compute the encryption of “My bid is  $n + 1$ .” without knowing  $n$  or the auctioneer’s secret key.

## 2 Preliminaries

**Key Encapsulation and Hybrid Encryption.** A key encapsulation mechanism KEM is a collection of three PPT algorithms: The key generator  $(PK, SK) \leftarrow \text{KeyGen}(1^\lambda)$  takes a security parameter  $\lambda \in \mathbb{N}$  and computes a public/secret key pair  $PK, SK$ . The encapsulation algorithm  $(K, C) \leftarrow \text{Encaps}(PK)$  computes a session key  $K$  and encapsulates it in a ciphertext  $C$  under the public key  $PK$ . It is assumed that  $K$  is uniform in some session key space  $\mathcal{K}(\lambda)$  that is of size superpolynomially in  $\lambda$ . The decapsulation algorithm  $K \leftarrow \text{Decaps}(SK, C)$  recovers the session key  $K$  from a ciphertext  $C$  with help of the secret key  $SK$ . For “invalid” ciphertexts, an arbitrary session key or a special rejection symbol  $\perp$  may be output.

An identity-based key encapsulation mechanism IBKEM is defined by the following five PPT algorithms: The setup algorithm  $(PK, SK) \leftarrow \text{Setup}(1^\lambda)$  computes a master public/secret key pair  $PK, SK$ . The (deterministic) public key generator  $PK_{ID} \leftarrow \text{PubKey}(PK, ID)$  deduces the user public key  $PK_{ID}$  for identity  $ID \in \{0, 1\}^*$  from the master public key  $PK$ . The secret key generator  $SK_{ID} \leftarrow \text{SecKey}(SK, ID)$  computes a user secret key  $SK_{ID}$  for identity  $ID \in \{0, 1\}^*$  using the master secret key  $SK$ . Encapsulation and decapsulation are defined as for KEM, except that  $PK_{ID}$  or  $SK_{ID}$  are used instead of  $PK$  and  $SK$ .

Any KEM can be combined with an appropriately secure symmetric encryption scheme to obtain *hybrid* public-key encryption [7]. Analogue results hold for the identity-based setting. Thus, constructing a KEM instead of full encryption is no limitation. Starting from IBKEM instead of IBE gives even more generality.

**Security.** The goal is to obtain a KEM with chosen ciphertext security (IND-CCA). An attacker  $\mathcal{A}$  should not be able to distinguish an encapsulated session key from a random one, even when given a Decaps-oracle. Starting point is an IBKEM with (a soon defined variant of) security against selective-identity and chosen-plaintext attacks (IND-sID-CPA), where such an oracle is not present. However, to model collusion of multiple users,  $\mathcal{A}$  is given a SecKey-oracle. The notions are formalized by the typical experiments (below on the right  $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$  and  $\mathcal{A}_1$  is assumed to know all values computed by  $\mathcal{A}_0$ )

$$\begin{array}{ll}
 \text{Exp}_{\text{IND-CCA}}^{\text{KEM}, \mathcal{A}}(\lambda): & \text{Exp}_{\text{IND-sID-CPA}}^{\text{IBKEM}, \mathcal{A}}(\lambda): \\
 PK, SK \leftarrow \text{KeyGen}(1^\lambda) & ID^* \leftarrow \mathcal{A}_0(1^\lambda) \\
 & PK, SK \leftarrow \text{Setup}(1^\lambda) \\
 & PK_{ID^*} \leftarrow \text{PubKey}(PK, ID^*) \\
 K_0^*, C^* \leftarrow \text{Encaps}(PK) & K_0^*, C^* \leftarrow \text{Encaps}(PK_{ID^*}) \\
 K_1^* \leftarrow \mathcal{K} & K_1^* \leftarrow \mathcal{K} \\
 b \leftarrow \{0, 1\} & b \leftarrow \{0, 1\} \\
 b' \leftarrow \mathcal{A}^{\text{Decaps}}(PK, K_b^*, C^*) & b' \leftarrow \mathcal{A}_1^{\text{SecKey}}(PK, K_b^*, C^*) \\
 \text{if } b = b', \text{ then output } win & \text{if } b = b', \text{ then output } win
 \end{array}$$

where the Decaps-oracle on input  $C \neq C^*$  returns  $K \leftarrow \text{Decaps}(SK, C)$  and the SecKey-oracle on input  $ID \neq ID^*$  outputs  $SK_{ID} \leftarrow \text{SecKey}(SK, ID)$ . A Scheme  $\in \{\text{KEM}, \text{IBKEM}\}$  is said to be *secure in the sense of NOTION*  $\in \{\text{IND-CCA}, \text{IND-sID-CPA}\}$  if for any PPT algorithm  $\mathcal{A}$  its advantage  $\text{Adv}_{\text{NOTION}}^{\text{Scheme}, \mathcal{A}}(\lambda) := |\Pr[\text{Exp}_{\text{NOTION}}^{\text{Scheme}, \mathcal{A}}(\lambda) = win] - \frac{1}{2}|$  is a negligible function in  $\lambda$ .

Note that unlike the original definition [7], in this IND-CCA experiment there is no oracle-phase for  $\mathcal{A}$  before she gets the challenge ciphertext. It is shown in [9] that both definitions are equivalent up the negligible additive term of  $q(\lambda)/|\mathcal{K}(\lambda)|$  where  $q(\lambda)$  is an upper bound for the number of oracle-queries of  $\mathcal{A}$ . The IND-sID-CPA experiment is trivially equivalent (without any loss) to the variant with two oracle-phases.

## 3 Transforming *partitioned* IBKEMs into CCA-secure KEMs

First, a special class of *partitioned* IBKEMs is defined. This partition property can easily be verified and is very common among identity-based schemes from bilinear groups (e. g., [12, 2, 3, 5]). Subsequently, a transformation from those schemes into KEMs will be given. To prove security of the resulting KEM, the IBKEM must satisfy a slightly strengthened security notion called *strong* IND-sID-CPA. This notion is widespread in partitioned IBKEMs, albeit not always easy to verify. To avoid this limitation of applicability, finally a very simple and efficient transformation from standard to strong IND-sID-CPA security is given.

**Definition 1.** An IBKEM is partitioned if it has the following properties. Let  $C$  be output by  $\text{Encaps}(PK_{ID})$ :

- Split of ciphertext: The ciphertext can be split into two parts,  $C = (C_1, C_2)$ , where the first part  $C_1$  only depends on the master public key  $PK$  and not the identity  $ID$  or user public key  $PK_{ID}$ .
- $C_2$ -uniqueness: The part  $C_2$  is uniquely determined by the user public key  $PK_{ID}$  and  $C_1$ .
- Simulatable rejection: There is an efficient algorithm  $\text{Reject}(PK_{ID}, C)$  that has the same output distribution as  $\text{Decaps}(\text{SecKey}(SK, ID), C)$  for any invalid ciphertext  $C$ . A ciphertext  $C$  is called invalid if (for fixed  $PK_{ID}$ ) it is never output by  $\text{Encaps}(PK_{ID})$ , regardless of the algorithm's internal coin tosses.

Typically, the output of the  $\text{Reject}$  algorithm would be uniform in  $\mathcal{K}$  or always  $\perp$ . Note that there is no requirement on the output distribution of  $\text{Reject}$  for *valid* ciphertexts as input! Furthermore, note that the distribution  $\text{Reject}$  has to mimic is meant over the random coin tosses of both  $\text{Decaps}$  and  $\text{SecKey}$ .

The split property allows to rewrite  $\text{Encaps}$  as two stages:  $\text{Encaps}_1(PK)$  outputs  $C_1$  and some state  $\sigma$  (e. g., all randomness produced by the internal coin tosses). Note that  $\sigma$  is meant only as input for the second stage and must be deleted afterwards.  $\text{Encaps}_2(PK_{ID}, C_1, \sigma)$  outputs  $C_2$  and session key  $K$ . The auxiliary input  $\sigma$  is crucial for  $\text{Encaps}_2$ , as computing either  $C_2$  or  $K$  from only  $PK_{ID}$  and  $C_1$  would typically imply inverting a one-way function. Due to this split property, the following transformation is (syntactically) sound:

$\text{KEM} := T_1(\text{IBKEM})$	$\text{KEM.KeyGen}(1^\lambda) = \text{IBKEM.Setup}(1^\lambda)$
$\text{KEM.Encaps}(PK) :$ $C_1, \sigma \leftarrow \text{IBKEM.Encaps}_1(PK)$ $ID \leftarrow C_1$ $PK_{ID} \leftarrow \text{IBKEM.PubKey}(PK, ID)$ $K, C_2 \leftarrow \text{IBKEM.Encaps}_2(PK_{ID}, C_1, \sigma)$ Return $K$ and $C \leftarrow (C_1, C_2)$	$\text{KEM.Decaps}(SK, C) :$ $C_1, C_2 \leftarrow C$ $ID \leftarrow C_1$ $SK_{ID} \leftarrow \text{IBKEM.SecKey}(SK, ID)$ $K \leftarrow \text{IBKEM.Decaps}(SK_{ID}, C)$ Return $K$

Note that by definition  $ID$  may be a binary string of arbitrary length (this can always be accomplished using collision resistant hashing). Hence, the binary encoding of  $C_1$  may be interpreted as identity.

**Definition 2.** For a partitioned IBKEM, the strong IND-sID-CPA experiment is defined below. (For comparison, the standard IND-sID-CPA experiment is rewritten equivalently using  $\text{Encaps}_{1/2}$  instead of  $\text{Encaps}$ .)

$\text{Exp}_{\text{strong IND-sID-CPA}}^{\text{partitioned IBKEM}, \mathcal{A}}(\lambda) :$ $PK, SK \leftarrow \text{Setup}(1^\lambda)$ $C_1^*, \sigma^* \leftarrow \text{Encaps}_1(PK)$ $ID^* \leftarrow \mathcal{A}_0(C_1^*)$ $PK_{ID^*} \leftarrow \text{PubKey}(PK, ID^*)$ $K_0^*, C_2^* \leftarrow \text{Encaps}_2(PK_{ID^*}, C_1^*, \sigma^*)$ $K_1^* \leftarrow \mathcal{K}$ $b \leftarrow \{0, 1\}$ $b' \leftarrow \mathcal{A}_1^{\text{SecKey}}(PK, K_b^*, C_1^*, C_2^*)$ if $b = b'$ , then output win	$\left( \text{Exp}_{\text{IND-sID-CPA}}^{\text{partitioned IBKEM}, \mathcal{A}}(\lambda) : \right.$ $ID^* \leftarrow \mathcal{A}_0(1^\lambda)$ $PK, SK \leftarrow \text{Setup}(1^\lambda)$ $PK_{ID^*} \leftarrow \text{PubKey}(PK, ID^*)$ $C_1^*, \sigma^* \leftarrow \text{Encaps}_1(PK)$ $K_0^*, C_2^* \leftarrow \text{Encaps}_2(PK_{ID^*}, C_1^*, \sigma^*)$ $K_1^* \leftarrow \mathcal{K}$ $b \leftarrow \{0, 1\}$ $b' \leftarrow \mathcal{A}_1^{\text{SecKey}}(PK, K_b^*, C_1^*, C_2^*)$ if $b = b'$ , then output win
---	---

Many proofs of IND-sID-CPA security for partitioned IBKEMs can easily be rearranged to prove even the apparently stronger notion. This might be surprising, as requiring to compute  $C_1^*$  (and thus  $PK$ ) before  $ID^*$  is chosen seems to hinder common proof techniques for IND-sID-CPA security, where the simulator chooses parts of  $PK$  dependent on  $ID^*$ . However, these parts are often not needed compute  $C_1^*$  and choosing them can be delayed until after  $ID^*$  is determined.

**Theorem 3.** If IBKEM is strong IND-sID-CPA secure, then  $\text{KEM} := T_1(\text{IBKEM})$  is IND-CCA secure. In particular for any attacker  $\mathcal{A}$  of KEM there exists an attacker  $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$  of IBKEM with:

$$\text{Adv}_{\text{IND-CCA}}^{\text{KEM}, \mathcal{A}}(\lambda) = \text{Adv}_{\text{strong IND-sID-CPA}}^{\text{partitioned IBKEM}, \mathcal{B}}(\lambda)$$

*Proof (Sketch).*  $\mathcal{B}_0$  on input  $C_1^*$  outputs  $ID^* \leftarrow C_1^*$ .  $\mathcal{B}_1$  forwards its input  $PK, K_b^*, C_1^*, C_2^*$  to  $\mathcal{A}$  and answers the KEM.Decaps-queries for  $(C_1, C_2)$  as follows: if  $C_1 \neq C_1^*$  it computes  $K$  as defined in the KEM.Decaps-algorithm replacing the IBKEM.SecKey step by an oracle call (which is allowed as  $ID = C_1 \neq C_1^* = ID^*$ ). If  $(C_1, C_2) = (C_1^*, C_2) \neq (C_1^*, C_2^*)$  it computes  $ID \leftarrow C_1$ ,  $PK_{ID} \leftarrow \text{IBKEM.PubKey}(PK, ID)$  and returns the output of  $\text{Reject}(PK_{ID}, (C_1, C_2))$ . The  $C_2$ -uniqueness and simulatable rejection properties guarantee the correctness of this answer. When  $\mathcal{A}$  outputs a guess  $b'$ ,  $\mathcal{B}$  simply forwards this value. It is easy to see that  $\mathcal{B}$  gives a perfect simulation to  $\mathcal{A}$  and has the same success probability.  $\square$

As argued above the strong IND-sID-CPA security should be no limit to the transformation's applicability as it is satisfied by many schemes. Unfortunately, to validate this for a particular scheme one has to look at its security proof. Interestingly, there is a simple transformation from normal to strong IND-sID-CPA security adding only one XOR computation. From now on, identities are assumed to be of fixed length  $\ell(\lambda)$  that is polynomially bounded in  $\lambda$ , but still large enough to hold the binary encoding of  $C_1$ .

$\text{IBKEM.Setup}(1^\lambda) :$ $PK, SK \leftarrow \overline{\text{IBKEM.Setup}}(1^\lambda)$ $x \leftarrow \{0, 1\}^{\ell(\lambda)}$ Return $(PK, x)$ and $(SK, x)$	$\text{IBKEM.PubKey}((PK, x), ID \in \{0, 1\}^{\ell(\lambda)}) :$ Return $PK_{ID} \leftarrow \overline{\text{IBKEM.PubKey}}(PK, ID \oplus x)$ $\text{IBKEM.SecKey}((SK, x), ID \in \{0, 1\}^{\ell(\lambda)}) :$ Return $SK_{ID} \leftarrow \overline{\text{IBKEM.SecKey}}(SK, ID \oplus x)$
$\text{IBKEM} := T_2(\overline{\text{IBKEM}})$	Encaps and Decaps remain unchanged.

This lifts standard to strong IND-sID-CPA security.

**Theorem 4.** *If  $\overline{\text{IBKEM}}$  is IND-sID-CPA secure, then  $\text{IBKEM} := T_2(\overline{\text{IBKEM}})$  is strong IND-sID-CPA secure. In particular for any attacker  $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$  of IBKEM there exists an attacker  $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$  of  $\overline{\text{IBKEM}}$  with:*

$$Adv_{\text{strong IND-sID-CPA}}^{\text{partitioned IBKEM}, \mathcal{A}}(\lambda) = Adv_{\text{IND-sID-CPA}}^{\text{partitioned } \overline{\text{IBKEM}}, \mathcal{B}}(\lambda)$$

*Proof (Sketch).*  $\mathcal{B}_0$  on input  $1^\lambda$  uniformly selects  $\overline{ID}^* \leftarrow \{0, 1\}^{\ell(\lambda)}$  as target identity. Then,  $\mathcal{B}_1$  is given  $PK, K_b^*, C_1^*, C_2^*$ . First, it starts  $\mathcal{A}_0$  with input  $C_1^*$  to obtain  $\mathcal{A}$ 's target identity  $ID^* \in \{0, 1\}^{\ell(\lambda)}$ . Then, it computes  $x \leftarrow ID^* \oplus \overline{ID}^*$  and starts  $\mathcal{A}_1$  on input  $(PK, x), K_b^*, C_1^*, C_2^*$ . SecKey-queries are forwarded to the own oracle after applying  $ID \oplus x$ . It is not hard to see that  $\mathcal{A}$ 's view has the expected distribution.  $\square$

## 4 Application to Boneh-Boyen's IBKEM and Further Extensions

Let  $(\mathbb{G}, \mathbb{G}_T, e)$  be bilinear groups of prime order  $p > 2^\lambda$  with generator  $g \in \mathbb{G}$  (for a definition see [2]). Furthermore, let  $H: \{0, 1\}^* \rightarrow \mathbb{Z}_p$  be a collision resistant hash function. Below, Boneh-Boyen's IBKEM and the result of applying transformation  $T_1$  is shown. The latter is almost BMW-KEM except that it has a slower, randomized decapsulation algorithm (see below how to get exactly BMW-KEM).

common IBKEM.Setup / KEM.KeyGen( $1^\lambda$ ) $a, b, c \leftarrow \mathbb{Z}_p; \quad g_1, g_2, g_3 \leftarrow g^a, g^b, g^c; \quad Z \leftarrow e(g_1, g_2) \quad PK \leftarrow (g_1, g_2, g_3, Z); \quad SK \leftarrow g^{ab}$		
$\text{IBKEM.PubKey}(PK, ID)$ $PK_{ID} \leftarrow g_1^{H(ID)} \cdot g_3$	$\text{IBKEM.Encaps}(PK_{ID})$ $s \leftarrow \mathbb{Z}_p$ $C \leftarrow (g^s, PK_{ID}^s)$ $K \leftarrow Z^s$	$\text{IBKEM.Decaps}(SK_{ID}, C)$ $(S_1, S_2) \leftarrow SK_{ID}$ $(C_1, C_2) \leftarrow C$ $K \leftarrow e(S_2, C_1) / e(C_2, S_1)$
$\text{IBKEM.SecKey}(SK, ID)$ $r \leftarrow \mathbb{Z}_p; SK_{ID} \leftarrow (g^r, SK \cdot PK_{ID}^r)$		
$\text{KEM.Encaps}(PK)$ $s \leftarrow \mathbb{Z}_p$ $C \leftarrow (C_1, C_2) = (g^s, (g_1^{H(C_1)} \cdot g_3)^s)$ $K \leftarrow Z^s$		$\text{KEM.Decaps}(SK, C)$ $(C_1, C_2) \leftarrow C$ $r \leftarrow \mathbb{Z}_p$ $K \leftarrow e(SK \cdot (g_1^{H(C_1)} \cdot g_3)^r, C_1) / e(C_2, g^r)$

The ciphertext split and  $C_2$ -uniqueness of IBKEM is obvious. A little calculation shows that for a malformed ciphertext, i. e.,  $(g, PK_{ID}, C_1, C_2)$  is not a Diffie-Hellman tuple, the output of  $\text{Decaps}(\text{SecKey}(SK, ID), C)$  is uniform in  $\mathbb{G}_T$ . Here, the randomness comes from the choice of  $r$  in  $\text{SecKey}$ . Consequently, IBKEM also has simulatable rejections. The security proof of IBKEM, in fact, already shows strong security, as the simulator simply sets  $C_1^*$  to one value of its own input and thus can give it to  $\mathcal{A}$  before knowing  $ID^*$ .

**Extensions.** Throughout all constructions one can replace the IBKEM by selective-tag weakly CCA-secure *tag-based key encapsulation* (TBKEM) which is—despite the “weakly CCA”—weaker than IBKEM [8]. This generalizes the applicability, gives some insight in existing constructions ([8, §6] and [9]), implies BMW-KEM with the more efficient  $\text{Decaps}$ -algorithm and explains why both schemes in [2] give “surprisingly similar KEMs” [4, full version §4.4]: Their TBKEM versions are essentially the same. Indeed, tag-based schemes can be seen as natural basis for this type of transformation as already observed in [8].

The construction can be generalized to the hierarchical setting (HIBKEM) [10] where the deepest level may be tag-based instead of identity-based and only needs to be selective-tag secure. This explains how the CCA secure IBKEM in [10] is obtained from Waters’ HIBKEM [12] with the deepest level being a Boneh-Boyen level [2].<sup>2</sup> Similarly, this explains the construction of a compact CCA secure IBKEM in [11, §4.3] from Boneh-Boyen-Goh’s HIBKEM with constant size ciphertexts [3].

Furthermore, the transformation can be generalized to anonymous (H)IBKEM, where a ciphertext leaks no information on the receiver’s identity, and threshold cryptography as briefly mentioned in [5, 4].

The  $C_2$ -uniqueness can be relaxed to some form of  $C_2$ -integrity, roughly: given a valid ciphertext  $(C_1^*, C_2^*)$  no efficient attacker can come up with another valid ciphertext  $(C_1^*, C_2)$ . Note that  $C_2$ -uniqueness implies  $C_2$ -integrity. With this relaxation (an IBKEM version of) the generic transformation due to Canetti et al. [6] can be split into two parts where the first one lifts *any* IND-sID-CPA secure IBKEM to *partitioned* IBKEM with *strong* IND-sID-CPA security and the second is the  $T_1$  defined above. This may be interpreted as affirmation that transformation  $T_1$  captures the common underlying idea of the direct and generic transformations.

**Conclusion.** KEM from any *partitioned* IBKEM. Here, being partitioned is a very natural and easily verifiable property. This transformation is a generalization of Boyen, Mei and Waters’ “direct construction” and can be extended even further to explain several other constructions. Having a single transformation rule instead of several ad hoc constructions gives possibly broader applicability, easier proofs (as only the partition property has to be proven instead of the security of a whole scheme), and a better understanding of those constructions.

**Acknowledgments.** I would like to thank Johannes Blömer for helpful discussions and comments.

## References

1. M. Abe, Y. Cui, H. Imai, and E. Kiltz. Efficient hybrid encryption from ID-based encryption. Technical Report 23, Cryptology ePrint Archive, 2007.
2. D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT '04*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.
3. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT '05*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005.
4. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security, CCS '05*, pages 320–329. ACM, 2005.
5. X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO '06*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.

<sup>2</sup> Also, this gives their “implicit rejection” a new interpretation of always re-randomizing the level-1 part of a user secret key when computing the level-2 part in order to obtain the simulatable rejection property.

6. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT '04*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2004.
7. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2004.
8. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *Theory of Cryptography Conference '06*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, 2006.
9. E. Kiltz. Chosen-ciphertext secure key-encapsulation based on gap hashed diffie-hellman. In *Public Key Cryptography '07*, volume 4450 of *Lecture Notes in Computer Science*, pages 282–297. Springer, 2007.
10. E. Kiltz and D. Galindo. Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. In *Australasian Conf. on Inf. Sec. and Priv., ACISP '06*, volume 4058 of *LNCS*, pages 336–347. Springer, 2006.
11. E. Kiltz and Y. Vahlis. CCA2 secure IBE: Standard model efficiency through authenticated symmetric encryption. In *Cryptographers' Track at the RSA Conf., CT-RSA '08*, volume 4964 of *LNCS*, pages 221–238. Springer, 2008.
12. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT '05*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.