Plaintext Awareness in Identity-Based Key Encapsulation*

Mark Manulis[1] Bertram Poettering[2]
Douglas Stebila[3]

[1] *Department of Computing, University of Surrey, Guildford, Surrey, United Kingdom*
mark@manulis.eu
[2] *Information Security Group, Royal Holloway, University of London, Egham, Surrey, United Kingdom*
bertram.poettering@rhul.ac.uk
[3] *School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, Queensland, Australia*
stebila@qut.edu.au

September 29, 2012

Abstract

The notion of *plaintext awareness* (PA) has many applications in public key cryptography: it offers unique, stand-alone security guarantees for public key encryption schemes, has been used as a sufficient condition for proving indistinguishability against adaptive chosen ciphertext attacks (IND-CCA), and can be used to construct privacy-preserving protocols such as deniable authentication. Unlike many other security notions, plaintext awareness is very fragile when it comes to differences between the random oracle and standard models; for example, many implications involving PA in the random oracle model are not valid in the standard model and vice versa. Similarly, strategies for proving PA of schemes in one model cannot be adapted to the other model. Existing research addresses PA in detail only in the public key setting.

This paper gives the first formal exploration of plaintext awareness in the identity-based setting and, as initial work, proceeds in the random oracle model. The focus is laid mainly on identity-based key encapsulation mechanisms (IB-KEMs), for which the paper presents the first definitions of plaintext awareness, highlights the role of PA in proof strategies of IND-CCA security, and explores relationships between PA and other security properties.

On the practical side, our work offers the first, highly efficient, general approach for building IB-KEMs that are simultaneously plaintext-aware and IND-CCA-secure. Our construction is inspired by the Fujisaki-Okamoto (FO) transform, but demands weaker and more natural properties of its building blocks. This result comes from a new look at the notion of γ-*uniformity* that was inherent in the original FO transform. We show that for IB-KEMs (and PK-KEMs) this assumption can be replaced with a weaker computational notion, which is in fact implied by one-wayness. Finally, we give the first concrete IB-KEM scheme that is PA and IND-CCA-secure by applying our construction to a popular IB-KEM and optimizing it for better performance.

*This research work is part of the bilateral research project between Germany and Australia, funded jointly by the German Academic Exchange Service (DAAD) through grant Nr. 53361649 and by Australia's Department of Innovation, Industry, Science and Research (DIISR). Mark Manulis was also supported by the German Research Foundation (DFG) through grant MA 4096. He and Bertram Poettering wish further to acknowledge support from the Center of Advanced Security Research Darmstadt (CASED) and the European Center for Security and Privacy by Design (EC SPRIDE).

Keywords: plaintext awareness; identity-based encryption; key encapsulation mechanism; generic transformation

1 Introduction

A modern approach to hybrid encryption is given by the KEM/DEM paradigm [Sho00, CS02, CS03, AGKS05, DGKS10], which gives a modular construction of public key encryption (PKE) and identity-based encryption (IBE) schemes. The *key encapsulation mechanism* (KEM) is first applied by the sender to compute the key and its encapsulation. The output key is used in the *data encapsulation mechanism* (DEM) to encrypt the message. The recipient decapsulates the key and then decrypts the message. This mechanism is especially valuable for secure transmission of longer messages since the DEM part is usually based on fast, symmetric techniques.

Adaptive CCA Security and Transformations. The widely accepted security notion for KEMs is *indistinguishability against adaptive chosen ciphertext attacks* (IND-CCA[1]) [CS03, HHK10], and diverse KEM constructions satisfying this property have been proposed in the public key (PK-KEM) [Sho00, CS02, CS03, Den03, BMW05, Kil06] and the identity-based (IB-KEM) [BFMLS08, KG09] settings. Some [Sho00, Den03, BFMLS08] are secure in the random oracle model; others [Sho00, CS02, CS03, BMW05, Kil06, KG09] in the standard model.

In general, IND-CCA-secure KEMs can trivially be obtained from any IND-CCA-secure encryption scheme by encrypting (sufficiently many) randomly chosen messages that are then used as the encapsulated key. In the random oracle model, Dent [Den03] demonstrated that IND-CCA-secure PK-KEMs can further be obtained from PKE schemes that satisfy only a weaker property of *one-wayness* (OW). These results were extended by Bentahar et al. [BFMLS08] to the identity-based setting. This parallels prior work on IND-CCA-secure encryption: in the random oracle model, transformations by Fujisaki and Okamoto [FO99a, FO99b] and by Okamoto and Pointcheval [OP01] offer IND-CCA security for PKE schemes that are one-way and whose ciphertexts are close to uniformly distributed (so-called γ-*uniformity*); similar transformations were shown by Kitagawa et al. [KYH+06, YKH+06] to apply in the identity-based setting. In the standard model, IND-CCA-secure PK-KEMs can be obtained using transformations originally proposed for IND-CCA-secure PKE schemes, e.g., by adding a "proof of well-formedness" to an IND-CPA-secure scheme [Sah99, DDN00, ES02, CS02, CS03], or by converting a weakly secure IBE scheme [CHK04, BK05, BCHK07].

Plaintext Awareness and its Role. The notion of *plaintext awareness* (PA) was introduced in the context of PKE by Bellare et al. [BR94, BDPR98, BP04], both in the random oracle model [BR94, BDPR98] and the standard model [BP04], with two variants (weaker PA1 and stronger PA2) in the standard model[2]. Roughly speaking, plaintext awareness demands that no ciphertext can be created without explicit knowledge of the encrypted message. From a technical perspective, this is represented by the existence of a *plaintext extractor* that, when run on the same inputs as the *ciphertext creator*, can produce the plaintext behind the ciphertext. Originally [BR94], PA was seen as a strategy for proving IND-CCA security of PKE schemes (by using plaintext awareness to answer decryption queries). It is also useful in other contexts: for example, *deniable* authentication and key exchange protocols [DGK06] can be built from plaintext-aware KEMs.

[1] By CCA in this paper we mean adaptive chosen ciphertext attacks, often referred to CCA2.

[2] Alternative definitions of plaintext awareness in the standard model were earlier proposed by Herzog, Liskov, and Micali [HLM03]. They consider a specialized PKE setting where both senders and receivers generate individual secret/public key pairs and register their public keys with a trusted authority. In contrast, definitions from [BR94, BDPR98, BP04] are more standard in that they do not rely on registration authorities and assume that only recipients have public keys.

Bellare et al. [BDPR98] showed that, for PKE schemes in the random oracle model, combining PA and IND-CPA implies IND-CCA (but PA is not implied by IND-CCA in general). In the standard model, Teranishi and Ogata [TO06] proved that IND-CCA security of PKE schemes is implied by one-wayness (OW) and PA2, and also that, surprisingly, in the random oracle model this implication fails. This serves as a hint that not all standard model results on plaintext awareness carry over to the random oracle-based setting and vice versa.

Birkett and Dent [BD08, BD11] observed that, while PA was a useful strategy for proving IND-CCA security in the random oracle model, it was not always useful in the standard model, where some PA2 proofs often assumed that the PKE schemes were already IND-CCA-secure. They also developed a different proof strategy for PA2 security of PKE schemes (in the KEM/DEM paradigm) by considering a weaker notion of plaintext awareness combined with a new property, "simulatability", and used this to prove that Cramer-Shoup PK-KEM [CS03] is PA and IND-CCA-secure (see also Dent's earlier proof [Den06]). Teranishi and Ogata [TO08] proved PA2 for Cramer-Shoup PKE using a slightly weaker (but similar) property instead of simulatability.

With respect to plaintext awareness of hybrid PK-KEM/DEM constructions, Jiang and Wang [JW10] proved that a PK-KEM that simultaneously satisfies a variant of IND-CCA security and some weaker notion of plaintext awareness combined with a one-time unforgeable DEM results in a hybrid construction that offers PA2 (for both the random oracle and standard models).

Our Focus. Existing work on plaintext awareness and its relevance for proving IND-CCA security has been carried out for PKE and PK-KEMs. Current knowledge on plaintext awareness in the identity-based setting is very limited. To date, no definitions of plaintext awareness for IB-KEMs exist, neither in the random oracle nor in standard models. Are the techniques developed for relating plaintext awareness and IND-CCA security of public key schemes applicable to the identity-based setting? Can existing transformations be applied to construct identity-based schemes that are both plaintext-aware and IND-CCA-secure?

In fact, the standard model strategies [BD08, TO08, BD11] for proving plaintext awareness of PK-KEMs, demonstrated successfully on the Cramer-Shoup scheme [CS03], do not seem to apply to current standard model IB-KEMs. For example, the prominent IND-CCA-secure standard model IB-KEM by Kiltz and Galindo [KG09], which can be seen as a pairing-based variant of the Cramer-Shoup PK-KEM, does not have the simulatability property used by Birkett and Dent [BD11] (nor the relaxation used by Teranishi and Ogata [TO08]). This is due to the existence of public verification of the "well-formedness" of the ciphertext using a bilinear pairing (bilinear groups are the most popular technique for constructing identity-based primitives). This motivates a separate treatment of plaintext awareness in the identity-based setting, including its connection to IND-CCA security and the construction of generic transformations.

1.1 Our Results and Techniques

Our Results. In this work we initiate the study of plaintext awareness for IB-KEMs, aiming to draw parallels and indicate differences to the corresponding results for PK-KEMs. While it is important to investigate plaintext awareness for IB-KEMs both with and without random oracles, here we work in the random oracle model (which tends to result in very efficient, practical constructions) and postpone standard model research to future work. We introduce definitions of plaintext awareness for IB-KEMs, investigate its role and relationship to other IB-KEM security notions (in particular, one-wayness and indistinguishability), and develop a method for obtaining highly efficient IB-KEMs that are simultaneously plaintext-aware and IND-CCA-secure from ones that satisfy only the basic notion of one-wayness.

Our results, which are summarized in Figure 1, are as follows:

– We provide (in Section 2) a formalization of plaintext awareness for IB-KEMs in the random oracle model and explore the relationship of this notion to other security notions, in particular observing that a scheme that is both PA and OW is also OW-CCA (Section 3).
– We introduce in Section 5 a generic transformation $\mathcal{F}$ on IB-KEMs, somewhat related to the Fujisaki-Okamoto transformation [FO99b] for PKE, that transforms an OW-secure IB-KEM into one that is both OW-CCA-secure and plaintext-aware. This construction is quite efficient. Our construction has an advantage from the perspective of provable security: whereas the proof of plaintext awareness of Fujisaki-Okamoto requires that the KEM be both OW and *uniform*, our construction $\mathcal{F}$ eliminates the needs for uniformity. Instead, our proof relies on a new intermediate security notion, *computational uniformity*, introduced in Section 4, which we show is actually implied by OW; hence, OW suffices for proving that our new construction yields PA.
– As a consequence of our $\mathcal{F}$ construction, we observe a difference between OW, IND, and PA notions in the standard model versus the random oracle model. Teranishi and Ogata [TO06] showed that, in the standard model, OW and PA of PKE schemes implies IND-CCA security. For IB-KEMs in the random oracle model, we show that this does not hold.
– In Section 6, we describe a simple hash-based construction # that adds IND security to our $\mathcal{F}$ transform (and preserves its plaintext awareness).
– Together, the $\mathcal{F}$ and # constructions efficiently transform an OW-secure IB-KEM into one that provides plaintext awareness and IND-CCA security. As described in Section 7, some additional optimizations can be applied when these two techniques are used together. We show an application of this technique to the IB-KEMs that underlie the IBE schemes by Boneh and Franklin [BF01] and Sakai and Kasahara [SK03], to construct highly efficient IB-KEMs that are simultaneously PA and IND-CCA-secure.

Since IND-CCA security does not in general imply plaintext awareness (either in the PKE setting [BDPR98] or for IB-KEMs, as we prove), our $\# \circ \mathcal{F}$ constructions offers the only known provable approach for constructing IB-KEMs that are *simultaneously* PA and IND-CCA-secure.

Finally, we show that (adaptions of) our transformations for OW-secure IB-KEMs can also be applied to OW-secure PK-KEMs. This extends work of Dent [Den03] since PK-KEMs obtained with our transformations are both plaintext-aware and IND-CCA-secure. More importantly, PK-KEMs with PA and IND-CCA security readily admit construction of plaintext-aware hybrid PKE schemes based on the results from [JW10] for the KEM/DEM approach.

New Techniques. Central to our results is a novel perspective on the notion of γ-uniformity, which was originally defined in the public key setting [FO99b] and has been extended to IBE schemes [TO08]. Existing random oracle-based transformations from OW to IND-CCA [FO99a, FO99b, OP01, TO08] explicitly required γ-uniformity for the underlying encryption scheme. We show this requirement is unnecessary for plaintext awareness (and IND-CCA security) of KEMs.

Our proof technique uses a new, intermediate notion called *computational uniformity*. We prove that, for IB-KEMs, computational uniformity is immediately implied by one-wayness. The understanding of this implication helps to explain why generic constructions of IND-CCA-secure IB-KEMs in [BFMLS08] do not require uniformity as a separate condition on the underlying OW-secure IBE encryption scheme: in [BFMLS08], an IBE scheme is used to encrypt random messages (from which encapsulation keys are derived) and can thus be seen as an OW-secure IB-KEM. Our technique also applies to the public key setting: In Appendix A we extend our results towards PK-KEMs showing that their computational uniformity is also implied by OW security. This paves the way for using our transformations developed for IB-KEMs in the construction of plaintext-aware and IND-CCA-secure PK-KEMs.

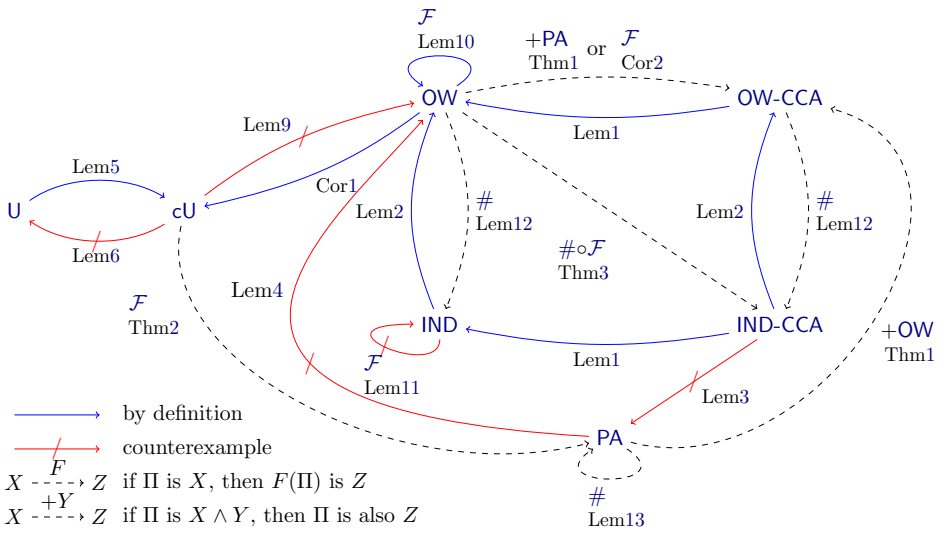

Figure 1: Relationships between security notions and results of generic transformations for IB-KEMs. Solid lines (→) denote an implication between notions (e.g., IND implies OW); solid lines with a cross (↛) denote a separation between notions (e.g., IND-CCA does not imply PA); and dashed lines indicate that one property can be obtained either by applying a transformation (e.g., $\mathcal{F}$ transforms an OW scheme into an OW-CCA scheme) or by satisfying two properties (e.g., a scheme that is both OW and PA is also OW-CCA).

1.2 Notation

Unless explicitly stated, all algorithms and adversaries throughout this paper are probabilistic and polynomial time. Let $A(x;r)$ denote the running of A on input x and random coins r. We use the notation $y \xleftarrow{\$} A(x)$ to denote setting y to the output of A run on input x with uniform randomness, and $y \in A(x)$ denotes that y is an element of the set of outputs of A when run on x with any randomness. If algorithm A outputs tuples, i.e. $(x, y, z) \leftarrow A$, then partial computation of the output vector is denoted with wildcards, e.g. $(\cdot, y, \cdot) \leftarrow A$, if the first and third component of the result, x and z, are neither computed nor assigned.

2 Secrecy notions and plaintext awareness of IB-KEMs

The modular concept of hybrid encryption, where asymmetric encryption schemes are constructed from two building blocks — a key encapsulation mechanism (KEM) and a data encapsulation mechanism (DEM) — was originated by [CS03] and is a popular design strategy for modern cryptosystems [KG09, KD04, Den03]. Specifically, in this paper we are interested in the identity-based variant of KEMs, although most of our results readily extend to the public key setting as well (cf. Appendix A).

Definition 1 (IB-KEM)**.** *An* identity-based key encapsulation mechanism (IB-KEM) *is a tuple* $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encap}, \mathsf{Decap})$ *of algorithms with, for each* $k \in \mathbb{N}$*, associated finite sets* $\mathtt{IDSp}(k)$*,* $\mathtt{CoinSp}(k)$*,* $\mathtt{CipherSp}(k)$*, and* $\mathtt{KeySp}(k)$*, where*

- $\mathsf{Setup}(1^k)$ *is a master key generation algorithm which outputs a pair of strings* (mpk, msk)*.*
- $\mathsf{Extract}(msk, id)$ *is a key extraction algorithm which, on input master secret key* msk *and an identity* $id \in \mathtt{IDSp}(k)$*, outputs a string* sk*.*
- $\mathsf{Encap}_{mpk}(id; r)$ *is a key encapsulation algorithm; it takes a master public key* mpk*, the recipient's identity* $id \in \mathtt{IDSp}(k)$*, and random coins* $r \in \mathtt{CoinSp}(k)$*, and outputs a ciphertext* $c \in \mathtt{CipherSp}(k)$ *and a key* $K \in \mathtt{KeySp}(k)$*. We use the notation* $\mathsf{Encap}_{mpk}(id)$ *as*

$\mathrm{Expt}^{\mathsf{OW[\text{-}CCA]}}_{\Pi,\mathcal{A}}(k)$:

(a) $\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k)$
(b) $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}^{\mathcal{H}}(1^k)$
(c) $(id^*, st) \xleftarrow{\$} \mathcal{A}_1^{\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X[, \mathcal{O}_D]}(mpk)$
- If $\mathcal{A}$ queries $\mathcal{O}_{\mathcal{H}}(H, m)$:
  (a) $x \leftarrow \mathcal{H}(H, m)$
  (b) Append (H, m, x) to HList.
  (c) Answer $\mathcal{A}$ with x.
- If $\mathcal{A}$ queries $\mathcal{O}_X(id)$:
  (a) $sk \xleftarrow{\$} \mathsf{Extract}^{\mathcal{H}}(msk, id)$
  (b) Append (id, sk) to XList.
  (c) Answer $\mathcal{A}$ with sk.
- If $\mathcal{A}$ queries $\mathcal{O}_D(id, c)$ (in CCA variant):
  (a) $sk \xleftarrow{\$} \mathsf{Extract}^{\mathcal{H}}(msk, id)$
  (b) $K \leftarrow \mathsf{Decap}^{\mathcal{H}}_{sk}(c)$
  (c) Append (id, c) to DList.
  (d) Answer $\mathcal{A}$ with K.

(d) $(c^*, K^*) \xleftarrow{\$} \mathsf{Encap}^{\mathcal{H}}_{mpk}(id^*)$
(e) $K' \xleftarrow{\$} \mathcal{A}_2^{\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X[, \mathcal{O}_D]}(st, c^*)$
- Answer $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X, \mathcal{O}_D$ queries as above.

(f) If $(id^*, \cdot) \in \mathsf{XList}$ return LOSE.
(g) If $(id^*, c^*) \in \mathsf{DList}$ return LOSE.
(h) If $K' \neq K^*$ return LOSE.
(i) Return WIN.

$\mathrm{Expt}^{\mathsf{IND[\text{-}CCA]},b}_{\Pi,\mathcal{A}}(k)$:

(a) $\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k)$
(b) $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}^{\mathcal{H}}(1^k)$
(c) $(id^*, st) \xleftarrow{\$} \mathcal{A}_1^{\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X[, \mathcal{O}_D]}(mpk)$
- If $\mathcal{A}$ queries $\mathcal{O}_{\mathcal{H}}(H, m)$:
  (a) $x \leftarrow \mathcal{H}(H, m)$
  (b) Append (H, m, x) to HList.
  (c) Answer $\mathcal{A}$ with x.
- If $\mathcal{A}$ queries $\mathcal{O}_X(id)$:
  (a) $sk \xleftarrow{\$} \mathsf{Extract}^{\mathcal{H}}(msk, id)$
  (b) Append (id, sk) to XList.
  (c) Answer $\mathcal{A}$ with sk.
- If $\mathcal{A}$ queries $\mathcal{O}_D(id, c)$ (in CCA variant):
  (a) $sk \xleftarrow{\$} \mathsf{Extract}^{\mathcal{H}}(msk, id)$
  (b) $K \leftarrow \mathsf{Decap}^{\mathcal{H}}_{sk}(c)$
  (c) Append (id, c) to DList.
  (d) Answer $\mathcal{A}$ with K.

(d) $(c^*, K_0^*) \xleftarrow{\$} \mathsf{Encap}^{\mathcal{H}}_{mpk}(id^*)$
(e) $K_1^* \xleftarrow{\$} \mathtt{KeySp}(k)$
(f) $d \xleftarrow{\$} \mathcal{A}_2^{\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X[, \mathcal{O}_D]}(st, c^*, K_b^*)$
- Answer $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X, \mathcal{O}_D$ queries as above.

(g) If $(id^*, \cdot) \in \mathsf{XList}$ return LOSE.
(h) If $(id^*, c^*) \in \mathsf{DList}$ return LOSE.
(i) Return d.

Figure 2: Experiments for one-way security (OW[-CCA]) and indistinguishability (IND[-CCA]) of IB-KEMs

shortcut for $\mathsf{Encap}_{mpk}(id; r)$ *with freshly drawn* $r \xleftarrow{\$} \mathtt{CoinSp}(k)$.
- $\mathsf{Decap}_{sk}(c)$ *is a key decapsulation algorithm which takes secret key* sk *and a ciphertext* $c \in \mathtt{CipherSp}(k)$*, and returns a key* $K \in \mathtt{KeySp}(k)$ *or the symbol* $\perp$.

IB-KEM Π *is* correct *if for all* $k \in \mathbb{N}$*, all* $(mpk, msk) \in \mathsf{Setup}(1^k)$*, all* $id \in \mathtt{IDSp}(k)$*, all* $sk \in \mathsf{Extract}(msk, id)$*, and all* $(c, K) \in \mathsf{Encap}_{mpk}(id)$*, we have* $\mathsf{Decap}_{sk}(c) = K$.

In the random oracle model, the four algorithms of an IB-KEM have access to a common set of random oracles $\mathcal{H} = (H_1, \ldots, H_h)$. This is explicitly indicated by the notation $\mathsf{Setup}^{\mathcal{H}}$, $\mathsf{Extract}^{\mathcal{H}}$, $\mathsf{Encap}^{\mathcal{H}}$, and $\mathsf{Decap}^{\mathcal{H}}$. The query to a specific oracle H in $\mathcal{H}$ on value m is denoted by $\mathcal{H}(H, m)$. By $\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k)$ we denote the process of picking at random an instantiation of random oracles $\mathcal{H} = (H_1, \ldots, H_h)$.

2.1 One-way security and indistinguishability

The classical secrecy notions of IB-KEMs are one-way security and key indistinguishability [BFMLS08], either with or without a decryption oracle: OW, OW-CCA, IND, and IND-CCA. The adversary's goal in OW[-CCA] is to (fully) compute the key that corresponds to a given ciphertext $c^* \in \mathtt{CipherSp}(k)$, while in IND[-CCA] it is sufficient to distinguish the real key from a random one, again given a ciphertext c^*. This intuition is formalized in experiments $\mathrm{Expt}^{\mathsf{OW[\text{-}CCA]}}$ and $\mathrm{Expt}^{\mathsf{IND[\text{-}CCA]}}$, in Figure 2. As most results established in this paper are proven in the random oracle model, besides giving adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ access to a key extraction oracle $\mathcal{O}_X$ and — in the CCA variants — a key decapsulation oracle $\mathcal{O}_D$, the security experiments for OW[-CCA] and IND[-CCA] explicitly express the option of $\mathcal{A}$ to make queries to hash oracle $\mathcal{O}_{\mathcal{H}}$.

Definition 2 (One-way security (OW[-CCA]))**.** *An IB-KEM* Π *is said to be* OW- *(resp.* OW-CCA-*)* secure *if, for all adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, the corresponding success probability is negligible*

in k, where we define

$$\mathrm{Succ}^{\mathsf{OW[\text{-}CCA]}}_{\Pi,\mathcal{A}}(k) = \Pr\left[\mathrm{Expt}^{\mathsf{OW[\text{-}CCA]}}_{\Pi,\mathcal{A}}(k) = \textsc{win}\right] \quad .$$

Definition 3 (Indistinguishability (IND[-CCA])). *An IB-KEM Π is said to be* IND- *(resp.* IND-CCA-*) secure if, for all adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the corresponding advantage is negligible in k, where we define*

$$\begin{aligned}\mathrm{Advt}^{\mathsf{IND[\text{-}CCA]}}_{\Pi,\mathcal{A}}(k) = \Big|&\Pr\left[\mathrm{Expt}^{\mathsf{IND[\text{-}CCA]},0}_{\Pi,\mathcal{A}}(k) = 1\right] \\ &- \Pr\left[\mathrm{Expt}^{\mathsf{IND[\text{-}CCA]},1}_{\Pi,\mathcal{A}}(k) = 1\right]\Big| \quad .\end{aligned}$$

Lemmas 1 and 2 follow immediately from the definitions:

Lemma 1 (OW-CCA $\Rightarrow$ OW and IND-CCA $\Rightarrow$ IND). *Let Π be an IB-KEM. If Π is* OW-CCA*-secure, then Π is* OW*-secure. If Π is* IND-CCA*-secure, then Π is* IND*-secure.*

Lemma 2 (IND $\Rightarrow$ OW and IND-CCA $\Rightarrow$ OW-CCA). *Let Π be an IB-KEM. If Π is* IND*-secure, then Π is* OW*-secure. If Π is* IND-CCA*-secure, then Π is* OW-CCA*-secure.*

2.2 Plaintext awareness

The notion of *plaintext awareness*, as established in the setting of public key encryption (PKE) in [BDPR98, BP04], captures the intuition that no one can generate a (valid) ciphertext c^* without knowing the message to which it will decrypt. This is a very strong notion of security, generally implying IND-CCA security of PKE schemes [BDPR98, BP04]. In this section, we define an analogous notion for KEMs, in the identity-based setting. In particular, we demand that, given a plaintext-aware IB-KEM, it is impossible to come up with a ciphertext c^* without knowing the key to which it will decapsulate[3].

In line with the PKE setting, we define plaintext awareness using two algorithms, the *ciphertext creator* $\mathcal{A}$ and the *plaintext extractor* $\mathcal{K}$. The ciphertext creator's task is to output a target identity id^* and a ciphertext c^*. The candidate scheme is understood to be plaintext-aware if any such algorithm $\mathcal{A}$ could additionally output the key encapsulated in c^*. This is formalized via the plaintext extractor $\mathcal{K}$, which receives as input the transcript of all of $\mathcal{A}$'s interactions, including $\mathcal{A}$'s input, all oracle queries posed by $\mathcal{A}$ together with corresponding answers, and the values output by $\mathcal{A}$. That is, $\mathcal{K}$ is in the position to retrace the computations that $\mathcal{A}$ performed to generate c^* (see also Remark 4), and we expect from $\mathcal{K}$ to use this knowledge to output the key encapsulated in c^*.

These ideas are formalized in Definition 4. As in [BDPR98], we explicitly regard plaintext awareness in the random oracle model. In particular, although usually unnecessary outside the secret key setting, we provide an encapsulation oracle $\mathcal{O}_E$ that allows creation of encapsulated keys, but without the adversary seeing random oracle queries asked from within corresponding Encap invocations. Availability of this oracle will become an important prerequisite for proving security against chosen-ciphertext attacks of plaintext-aware IB-KEMs.

Definition 4 (Plaintext awareness (PA)). *Let Π be an IB-KEM and $\mathcal{A}$ and $\mathcal{K}$ be algorithms (the "ciphertext creator" and "plaintext extractor", respectively). Consider experiment* $\mathrm{Expt}^{\mathsf{PA}}_{\Pi,\mathcal{A},\mathcal{K}}$ *from Figure 3 and define the* awareness *of Π as*

$$\mathrm{Aw}^{\mathsf{PA}}_{\Pi,\mathcal{A},\mathcal{K}}(k) = \Pr\left[\mathrm{Expt}^{\mathsf{PA}}_{\Pi,\mathcal{A},\mathcal{K}}(k) = \textsc{lose}\right] \quad .$$

[3] Strictly speaking, naming a KEM's property 'plaintext awareness' can be misleading: KEMs do not process messages in the classical sense, but only keys. Although one could argue that *key awareness* would be a better name for the intended property, in this paper we stick to 'plaintext awareness' that has been around in the context of public key encryption for the last two decades.

$\mathrm{Expt}^{\mathsf{PA}}_{\Pi,\mathcal{A},\mathcal{K}}(k)$:

(a) $\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k)$

(b) $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}^{\mathcal{H}}(1^k)$

(c) $(id^*, c^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_X,\mathcal{O}_E}(mpk)$
- Answer $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X$ queries as in Figure 2.
- If $\mathcal{A}$ queries $\mathcal{O}_E(id)$:
  (a) $(c, \cdot) \xleftarrow{\$} \mathsf{Encap}^{\mathcal{H}}_{mpk}(id)$
  (b) Append (id, c) to EList.
  (c) Answer $\mathcal{A}$ with c.

(d) If $(id^*, c^*) \in \mathsf{EList}$ return LOSE.

(e) $K^* \xleftarrow{\$} \mathsf{Decap}^{\mathcal{H}}_{\mathsf{Extract}^{\mathcal{H}}(msk,id^*)}(c^*)$

(f) $K' \xleftarrow{\$} \mathcal{K}^{\mathcal{H}}(id^*, c^*, mpk, \mathsf{HList}, \mathsf{XList}, \mathsf{EList})$

(g) If $K^* = K'$ return LOSE.

(h) Return WIN.

Figure 3: Plaintext awareness experiment PA for IB-KEMs. See Figure 2 for the definitions of HList and XList.

*An algorithm $\mathcal{K}$ is called a $\lambda(k)$-*extractor for Π *if*

$$\mathrm{Aw}^{\mathsf{PA}}_{\Pi,\mathcal{A},\mathcal{K}}(k) \geq \lambda(k) \ ,$$

for every adversary $\mathcal{A}$. IB-KEM Π is plaintext aware *if there exists a $\lambda(k)$-extractor $\mathcal{K}$ for Π such that $\lambda(k)$ is overwhelming, that is, $1 - \lambda(k)$ is negligible (in k).*

The following remarks discuss rationale and some interesting properties of Definition 4.

Remark 1 (Oracle access of plaintext extractor)**.** *We emphasize a subtle difference between plaintext awareness of PKE according to [BDPR98], and our Definition 4. In [BDPR98], the plaintext extractor $\mathcal{K}$ is not given access to the random oracle $\mathcal{H}$. This limitation is not motivated in [BDPR98], but instead back in [BR94, Section 5]: "Note we don't give $\mathcal{K}$ oracle access to $\mathcal{H}$: it is required to find the plaintext corresponding to [the ciphertext] given only $\mathcal{A}$'s view of the oracle". However, in the KEM setting, we disagree that this restriction is reasonable and argue*[4] *that $\mathcal{K}$ should be allowed to query the same random oracles that* Decap *is allowed to query, namely $\mathcal{H}$ (compare lines (e) and (f) of experiment $\mathrm{Expt}^{\mathsf{PA}}_{\Pi,\mathcal{A},\mathcal{K}}$).*

Remark 2 (Success probability of plaintext extractor)**.** *For an IB-KEM to achieve plaintext awareness according to Definition 4, we demand plaintext extractor $\mathcal{K}$ to have overwhelming success probability. At first sight, this seems to be a very conservative requirement, but we argue that it is indispensable. Consider the case of a plaintext extractor $\mathcal{K}$ where $\mathrm{Aw}^{\mathsf{PA}}_{\Pi,\mathcal{A},\mathcal{K}}(k)$ is growing fast, but not overwhelmingly fast, for all $\mathcal{A}$. We construct an adversary $\mathcal{A}^*$ that picks a random identity id, runs polynomially many executions of $(c, K) \xleftarrow{\$} \mathsf{Encap}(id)$ independently of each other in parallel, and applies $\mathcal{K}$ on each computed ciphertext c. If any such run of $\mathcal{K}$ fails to correctly extract the key (and it will, according to our assumption), corresponding c is the ciphertext c^* output by $\mathcal{A}^*$ (together with identity id). Clearly, when run on $\mathcal{A}^*$, we expect $\mathcal{K}$ to fail with high probability, in contradiction to assumed good performance of $\mathcal{K}$. Following this intuition, it can be shown that $\mathcal{K}$ cannot be a $\lambda(k)$-extractor for any $\lambda(k) > 0$.*

[4] Consider, for a justification, the following example (see also Lemma 13 for a more detailed discussion). Given an IB-KEM Π, derive from it IB-KEM Π' such that $\mathsf{Setup}' \equiv \mathsf{Setup}$, $\mathsf{Extract}' \equiv \mathsf{Extract}$, $\mathsf{Encap}'_{mpk}(id) \equiv [(c, K) \leftarrow \mathsf{Encap}_{mpk}(id), K' \leftarrow H(c, K), \text{return } (c, K')]$, $\mathsf{Decap}'_{sk}(c) \equiv [K \leftarrow \mathsf{Decap}_{sk}(c), \text{return } H(c, K)]$, where H is an independent random oracle. Intuitively, if Π is plaintext-aware then so is Π', as a plaintext extractor $\mathcal{K}$ could extract K from c and then derive $K' \leftarrow H(c, K)$. This reasoning, however, assumes that $\mathcal{K}$ has access to random oracle H.

Remark 3 (Universality of plaintext extractor)**.** *Observe that plaintext extractor $\mathcal{K}$, as defined in Definition 4, is a* universal extractor*: it does not depend on specific ciphertext creator $\mathcal{A}$, but only on scheme Π. Although this is a very strong notion of plaintext awareness, in Section 5 we see that it can be achieved even by rather efficient IB-KEMs. When leaving the random oracle model, however, it seems that the concept of universal extractability is too strong. For instance, in [BP04], the demand for a universal extractor is dropped. Instead, to achieve plaintext awareness, it suffices to have a specific plaintext extractor $\mathcal{K}_{\mathcal{A}}$ for each ciphertext creator $\mathcal{A}$. Note that, in this paper, we exploit universality of plaintext extractors, e.g. in the proof of Theorem 1.*

Remark 4 (Random tape of ciphertext creator)**.** *We will shortly discuss a variant of $\mathrm{Expt}^{\mathsf{PA}}_{\Pi,\mathcal{A},\mathcal{K}}$ where $\mathcal{K}$ gets as additional input the tape $\mathcal{R}[\mathcal{A}]$ of random coins used by $\mathcal{A}$ to craft (id^*, c^*). While this modification would be meaningful from a modeling point of view, it is already noticed in [BR94] that the randomness used by $\mathcal{A}$ is usually not needed to prove schemes to be plaintext-aware, at least in the random oracle model. As this observation holds for the IB-KEMs analyzed in this paper, we decided to stay with (stronger) Definition 4. However, in standard model, $\mathcal{K}$'s knowledge of random tape $\mathcal{R}[\mathcal{A}]$ is an unavoidable prerequisite for plaintext awareness, as pointed out in [BP04].*

3 Plaintext awareness and one-way security implies OW-CCA security

Although being defined independently of each other, the different notions of secrecy and plaintext awareness considered in Section 2 are not unrelated to each other. In particular, we will establish a strong result in Theorem 1: IB-KEMs that are both OW-secure and plaintext-aware are also OW-CCA-secure, i.e. availability of a decapsulation oracle does not threaten OW security of plaintext-aware IB-KEMs. The intuition behind this implication is simple: given a plaintext-aware IB-KEM, an OW-CCA adversary $\mathcal{A}$ cannot gain advantage from its decapsulation oracle, as $\mathcal{A}$ 'knows' the keys of all queried ciphertexts anyway. Nonetheless, the proof of Theorem 1 is technically involved as it has to cope with the fact that OW-CCA adversary $\mathcal{A}$ may query its $\mathcal{O}_D$ oracle polynomially many times, while plaintext extractor $\mathcal{K}$ is applicable only to adversaries that output exactly one ciphertext.

Taken together, the following results separate the security notions PA and IND[-CCA], and illustrate once more the strength of the notion of plaintext awareness.

Theorem 1 ($\mathsf{OW} \wedge \mathsf{PA} \Rightarrow \mathsf{OW\text{-}CCA}$)**.** *Let Π be an IB-KEM. If Π is* OW*-secure and* PA*, then Π is* OW-CCA*-secure.*

Proof. Let $\mathcal{K}$ be a $\lambda(k)$-extractor for Π, for an overwhelming $\lambda(k)$. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be any adversary against OW-CCA of Π. We construct an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against OW of Π such that $\mathcal{B}_1$ and $\mathcal{B}_2$ run $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively, as subroutines. While all queries to $\mathcal{O}_{\mathcal{H}}$ and $\mathcal{O}_X$ oracles posed by $\mathcal{A}$ will just be relayed by $\mathcal{B}$ to its own challenger, $\mathcal{B}$ will answer decapsulation queries to $\mathcal{O}_D$ itself, by applying plaintext extractor $\mathcal{K}$ on (partially) simulated $\mathcal{A}$. The precise specification of $\mathcal{B}_1$ and $\mathcal{B}_2$ is given in Figure 4 (left part).

Let q_D be the total number of decapsulation queries that $\mathcal{A}$ will ask. For $1 \leq j \leq q_D$, denote by (id_j, c_j) the arguments of the j-th $\mathcal{O}_D$ query, and let E_j be the event that the query is correctly answered by $\mathcal{B}$, i.e. that corresponding execution of $\mathcal{K}$ succeeds in computing the correct decapsulation K_j of (id_j, c_j). We will lowerbound the probability that E_j occurs. Let $\tau_j = (id_j, c_j, mpk, \mathtt{HLi}_j, \mathtt{XLi}_j, \mathtt{ELi}_j)$ denote the parameters provided by $\mathcal{B}$ to $\mathcal{K}$ in its j-th execution. Moreover, considering PA adversary $\mathcal{C}_j$ as defined in Figure 4 (right part), let $\tau'_j = (id'_j, c'_j, mpk, \mathsf{HList}_j, \mathsf{XList}_j, \mathsf{EList}_j)$ be a parameter set that plaintext extractor $\mathcal{K}$ would receive when run on $\mathcal{C}_j$, i.e. in experiment $\mathrm{Expt}^{\mathsf{PA}}_{\Pi,\mathcal{C}_j,\mathcal{K}}$, initialized with the same mpk and instantiation of $\mathcal{H}$ as $\mathcal{B}$. Careful comparison of the specifications of $\mathcal{B}$ and $\mathcal{C}_j$ reveals that τ_j and τ'_j are equally

$\mathcal{B}_1^{\mathcal{O}'_{\mathcal{H}},\mathcal{O}'_X}(mpk)$:
(a) $\texttt{HLi} \leftarrow \emptyset, \texttt{XLi} \leftarrow \emptyset$
(b) Run $\mathcal{A}_1^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_X,\mathcal{O}_D}(mpk)$
– If $\mathcal{A}_1$ queries $\mathcal{O}_{\mathcal{H}}(H,m)$:
(a) $x \leftarrow \mathcal{O}'_{\mathcal{H}}(H,m)$
(b) Append (H,m,x) to $\texttt{HLi}$.
(c) Answer $\mathcal{A}_1$ with x.
– If $\mathcal{A}_1$ queries $\mathcal{O}_X(id)$:
(a) $sk \leftarrow \mathcal{O}'_X(id)$
(b) Append (id,sk) to $\texttt{XLi}$.
(c) Answer $\mathcal{A}_1$ with sk.
– If $\mathcal{A}_1$ queries $\mathcal{O}_D(id,c)$:
(a) $K \xleftarrow{\$} \mathcal{K}^{\mathcal{O}_{\mathcal{H}}}(id,c,mpk,\texttt{HLi},\texttt{XLi},\emptyset)$
– If $\mathcal{K}$ queries $\mathcal{O}_{\mathcal{H}}(H,m)$:
(a) $x \leftarrow \mathcal{O}'_{\mathcal{H}}(H,m)$
(b) Append (H,m,x) to $\texttt{HLi}$.
(c) Answer $\mathcal{K}$ with x.
(b) Answer $\mathcal{A}_1$ with K.
(c) Finally $\mathcal{A}_1$ halts, outputting (id^*, st).
(d) $st' \leftarrow (st, mpk, \texttt{HLi}, \texttt{XLi})$
(e) Return (id^*, st').

$\mathcal{B}_2^{\mathcal{O}'_{\mathcal{H}},\mathcal{O}'_X}(st', c^*)$:
(a) Parse $(st, mpk, \texttt{HLi}, \texttt{XLi}) \leftarrow st'$
(b) Run $\mathcal{A}_2^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_X,\mathcal{O}_D}(st,c^*)$
– Answer $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X$ queries as above.
– If $\mathcal{A}_2$ queries $\mathcal{O}_D(id,c)$:
(a) $K \xleftarrow{\$} \mathcal{K}^{\mathcal{O}_{\mathcal{H}}}(id,c,mpk,\texttt{HLi},\texttt{XLi},(c^*))$
– If $\mathcal{K}$ queries $\mathcal{O}_{\mathcal{H}}(H,m)$:
(a) $x \leftarrow \mathcal{O}'_{\mathcal{H}}(H,m)$
(b) Append (H,m,x) to $\texttt{HLi}$.
(c) Answer $\mathcal{K}$ with x.
(b) Answer $\mathcal{A}_2$ with K.
(c) Finally $\mathcal{A}_2$ halts, outputting K'.
(d) Return K'.

$\mathcal{C}_j^{\mathcal{O}'_{\mathcal{H}},\mathcal{O}'_X,\mathcal{O}'_E}(mpk)$:
(a) $\texttt{HLi} \leftarrow \emptyset, \texttt{XLi} \leftarrow \emptyset, i \leftarrow 1$
(b) Run $\mathcal{A}_1^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_X,\mathcal{O}_D}(mpk)$
– If $\mathcal{A}_1$ queries $\mathcal{O}_{\mathcal{H}}(H,m)$:
(a) $x \leftarrow \mathcal{O}'_{\mathcal{H}}(H,m)$
(b) Append (H,m,x) to $\texttt{HLi}$.
(c) Answer $\mathcal{A}_1$ with x.
– If $\mathcal{A}_1$ queries $\mathcal{O}_X(id)$:
(a) $sk \leftarrow \mathcal{O}'_X(id)$
(b) Append (id,sk) to $\texttt{XLi}$.
(c) Answer $\mathcal{A}_1$ with sk.
– If $\mathcal{A}_1$ queries $\mathcal{O}_D(id,c)$:
(a) If $i = j$ return (id,c) else $i \leftarrow i+1$.
(b) $K \xleftarrow{\$} \mathcal{K}^{\mathcal{O}_{\mathcal{H}}}(id,c,mpk,\texttt{HLi},\texttt{XLi},\emptyset)$
– If $\mathcal{K}$ queries $\mathcal{O}_{\mathcal{H}}(H,m)$:
(a) $x \leftarrow \mathcal{O}'_{\mathcal{H}}(H,m)$
(b) Append (H,m,x) to $\texttt{HLi}$.
(c) Answer $\mathcal{K}$ with x.
(c) Answer $\mathcal{A}_1$ with K.
(c) Finally $\mathcal{A}_1$ halts, outputting (id^*, st).
(d) $c^* \leftarrow \mathcal{O}'_E(id^*)$
(e) Run $\mathcal{A}_2^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_X,\mathcal{O}_D}(st,c^*)$
– Answer $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X$ queries as above.
– If $\mathcal{A}_2$ queries $\mathcal{O}_D(id,c)$:
(a) If $i = j$ return (id,c) else $i \leftarrow i+1$.
(b) $K \xleftarrow{\$} \mathcal{K}^{\mathcal{O}_{\mathcal{H}}}(id,c,mpk,\texttt{HLi},\texttt{XLi},(c^*))$
– If $\mathcal{K}$ queries $\mathcal{O}_{\mathcal{H}}(H,m)$:
(a) $x \leftarrow \mathcal{O}'_{\mathcal{H}}(H,m)$
(b) Append (H,m,x) to $\texttt{HLi}$.
(c) Answer $\mathcal{K}$ with x.
(c) Answer $\mathcal{A}_2$ with K.

Figure 4: LEFT: Construction of OW adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ from OW-CCA adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and plaintext extractor $\mathcal{K}$. RIGHT: Specification of PA adversaries $\mathcal{C}_j$ whose transcripts match the parameters provided by $\mathcal{B}$ to the j-th execution of $\mathcal{K}$.

distributed (more precisely, if $\mathcal{B}$ and $\mathcal{C}_j$ are executed on the same random tape, then we have $\tau_j = \tau'_j$). If K'_j denotes the decapsulation of (id'_j, c'_j), then also (τ_j, K_j) and (τ'_j, K'_j) obey the same distribution, and hence

$$\begin{aligned}\Pr[E_j] = \Pr\left[\mathcal{K}^{\mathcal{H}}(\tau_j) = K_j\right] &= \Pr\left[\mathcal{K}^{\mathcal{H}}(\tau'_j) = K'_j\right] \\ &= \mathrm{Aw}^{\mathsf{PA}}_{\Pi,\mathcal{C}_j,\mathcal{K}}(k) \geq \lambda(k) \ .\end{aligned}$$

Setting $E = E_1 \wedge \ldots \wedge E_{q_D}$ and applying Lemma 16 (Appendix B) gets us to

$$\begin{aligned}\mathrm{Succ}^{\mathsf{OW}}_{\Pi,\mathcal{B}}(k) &\geq \Pr\left[\mathrm{Expt}^{\mathsf{OW}}_{\Pi,\mathcal{B}}(k) = \textsc{win} \wedge E\right] \\ &= \Pr\left[\mathrm{Expt}^{\mathsf{OW}}_{\Pi,\mathcal{B}}(k) = \textsc{win} \middle| E\right] \Pr[E] \\ &\geq \Pr\left[\mathrm{Expt}^{\mathsf{OW\text{-}CCA}}_{\Pi,\mathcal{A}}(k) = \textsc{win}\right] \\ &\quad \cdot (1 - q_D(1-\lambda(k))) \\ &= \mathrm{Succ}^{\mathsf{OW\text{-}CCA}}_{\Pi,\mathcal{A}}(k) \cdot (1 - q_D(1-\lambda(k))) \ .\end{aligned}$$

This concludes the proof, as $1 - q_D(1-\lambda(k))$ is an overwhelming quantity. □

$\mathsf{Setup}'(1^k)$:
(a) $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}(1^k)$
(b) $(c', \cdot) \xleftarrow{\$} \mathsf{Encap}_{mpk}(id_0)$
(c) $mpk' \leftarrow (mpk, c')$
(d) Return (mpk', msk).

$\mathsf{Extract}'(msk, id)$:
(a) Return $\mathsf{Extract}(msk, id)$.

$\mathsf{Encap}'_{mpk'}(id; r)$:
(a) Parse $(mpk, c') \leftarrow mpk'$.
(b) Return $\mathsf{Encap}_{mpk}(id; r)$.

$\mathsf{Decap}'_{sk}(c)$:
(a) Return $\mathsf{Decap}_{sk}(c)$.

Figure 5: Constructing Π' from Π. We assume that id_0 is a fixed identity in $\mathtt{IDSp}(k)$, and all queries to $\mathcal{H}$ oracle are relayed without modification.

Remark 5 (On the proving technique used in Theorem 1)**.** *A substantial element in the proof of Theorem 1 is the estimation of $\mathcal{K}$'s ability to compute the correct answers to $\mathcal{A}$'s $\mathcal{O}_D$ queries. Note that for arbitrary $\mathtt{HLi}, \mathtt{XLi}, \mathtt{ELi}$ lists we cannot expect $\mathcal{K}$ to correctly extract keys with any reasonable probability. Indeed, in order to argue that the parameters τ_j provided by $\mathcal{B}$ are sufficient for extraction of the desired keys, we had to specify auxiliary* PA *adversaries $(\mathcal{C}_j)_{1 \le j \le q_D}$, and had to show that these would give raise to the same parameter sets when run in $\mathrm{Expt}^{\mathsf{PA}}_{\Pi, \mathcal{C}_j, \mathcal{K}}$. Interestingly enough, these adversaries $\mathcal{C}_j$ were not designed to be ever executed: it is their bare existence that is necessary for the proof of Theorem 1 to go through.*

Remark 6 (On IND∧PA ⇒ IND-CCA)**.** *The techniques used to prove Theorem 1, that* OW∧PA ⇒ OW-CCA*, cannot be applied to prove the* IND *analogue that* IND *security and* PA *imply* IND-CCA *security. The reason for this is a subtle difference between the* OW *notion and the* IND *notion that, in the* IND *case, renders inapplicable the proving technique we discuss in Remark 5: In experiment $\mathrm{Expt}^{\mathsf{IND\text{-}CCA}}$, the adversary obtains ciphertext c^* and (possibly) corresponding key K^* from the challenger, but this cannot be reflected in parameters $\mathtt{HLi}, \mathtt{XLi}, \mathtt{ELi}$ that are provided to plaintext extractor $\mathcal{K}$. Hence, in a proof constructed like in Figure 4, we cannot argue about $\mathcal{K}$'s success probability to answer decapsulation queries of $\mathcal{A}_2$ correctly. In experiment $\mathrm{Expt}^{\mathsf{OW\text{-}CCA}}$, however, adversary just obtains a random ciphertext c^* encapsulated for identity id^*. This finds its perfect correspondence in $\mathcal{C}_j$ calling the $\mathcal{O}_E$ oracle on id^* (cf. line (d) in Figure 4, right part). Basically, this also explains why we have to set $\mathtt{ELi} = (c^*)$ when processing $\mathcal{A}_2$'s $\mathcal{O}_D$ queries.*

Observe that implication IND∧PA ⇒ IND-CCA *has been proven in the PKE setting [BDPR98]. The difference is that, in the* IND-CCA *experiment of public key encryption, the adversary has partial knowledge about the message that is encrypted in c^*: it is either m_0 or m_1, so adversary can query these to its encryption oracle $\mathcal{O}_E$.*

Although IND-CCA security is considered the strongest confidentiality goal for IB-KEMs, the following lemma shows that this property is to weak to imply plaintext awareness by itself.

Lemma 3 (IND-CCA ⇏ PA)**.** *Not all* IND-CCA*-secure IB-KEMs are* PA*.*

Proof. Let Π be an IND-CCA-secure IB-KEM. Consider IB-KEM Π' that arises when modifying Π's Setup to include into master public key mpk an encapsulation to some fixed identity $id_0 \in \mathtt{IDSp}(k)$, as specified in Figure 5. We will show that Π' is not plaintext-aware, but remains to be IND-CCA-secure.

Observe that any adversary $\mathcal{A}'$ against IND-CCA of Π' can be trivially turned into an adversary $\mathcal{A}$ against IND-CCA of Π, by letting $\mathcal{A}_1$ simulate the $(c', \cdot) \xleftarrow{\$} \mathsf{Encap}_{mpk}(id_0)$ step of line (b) of Setup', and, apart from that, letting $\mathcal{A}_1$ and $\mathcal{A}_2$ run $\mathcal{A}'_1$ and $\mathcal{A}'_2$, respectively, relaying all oracle queries to the own challenger. As the IND-CCA advantages of $\mathcal{A}$ and $\mathcal{A}'$ would be the same, this proves IND-CCA security of Π'.

To see that Π' is not plaintext-aware, we show that any corresponding plaintext extractor would break OW security of Π. Assume towards contradiction the existence of a $\lambda(k)$-extractor $\mathcal{K}_{\Pi'}$ for Π', where $\lambda(k)$ is overwhelming. Define adversary $\mathcal{B}$ against OW of Π as follows:

$\mathcal{B}_1(mpk)$ just outputs $(id^*, st) = (id_0, mpk)$, while $\mathcal{B}_2(st, c^*)$ sets $mpk' \leftarrow (mpk, c^*)$, computes $K \xleftarrow{\$} \mathcal{K}^{\mathcal{H}}_{\Pi'}(id_0, c^*, mpk', \emptyset, \emptyset, \emptyset)$, and outputs K. We use the technique discussed in Remark 5 to justify that $\mathcal{K}_{\Pi'}$ will succeed with high probability in revealing correct key K: consider the simple ciphertext creator $\mathcal{A}'$ for IB-KEM Π' that, on input mpk', parses $(mpk, c^*) \leftarrow mpk'$ and outputs (id_0, c^*). As $\mathcal{A}'$ doesn't ask any queries to its oracles, i.e. $\mathsf{HList} = \mathsf{XList} = \mathsf{EList} = \emptyset$, execution of $\mathcal{K}^{\mathcal{H}}_{\Pi'}(id_0, c^*, mpk', \emptyset, \emptyset, \emptyset)$ will readily extract the key encapsulated in c^* for identity id_0. More precisely, we have shown that $\mathrm{Succ}^{\mathsf{OW}}_{\Pi,\mathcal{B}}(k) = \mathrm{Aw}^{\mathsf{PA}}_{\Pi',\mathcal{A}',\mathcal{K}_{\Pi'}}(k) \geq \lambda(k) \geq 1 - \mathrm{negl}(k)$, in contradiction to $\mathsf{IND\text{-}CCA}$ security of Π (cf. Lemmas 1 and 2). This shows that Π' is not plaintext-aware. □

At the same time, plaintext awareness does not imply the basic confidentiality goal of one-wayness:

Lemma 4 ($\mathsf{PA} \not\Rightarrow \mathsf{OW}$)**.** *Not all plaintext-aware IB-KEMs are* OW*-secure.*

Proof. Consider the (correct, but insecure) IB-KEM in which $\mathsf{Encap}_{mpk}(id; r)$ returns $(c, K) = (r, r)$, i.e. keys are equal to their encapsulation. The scheme is clearly not OW-secure. However, it is plaintext aware: an appropriate plaintext extractor $\mathcal{K}$ would just output the ciphertext. □

4 Computational uniformity of IB-KEMs and its relation to one-wayness

We introduce two new security notions for IB-KEMs. The first one, *uniformity* (U), is defined mainly for historical reasons: it is the analogue to the uniformity notion for PKE coined in [FO99b]. We will see that its (strictly weaker) variant, *computational uniformity* (cU), is a very natural notion of IB-KEMs, already being implied by the basic secrecy requirement of one-way security (OW, Definition 2). In Section 5, it will find its application in the analysis of our generic transformation that achieves plaintext awareness for any IB-KEM. We give a treatment of the corresponding notions for PK-KEMs in Appendix A.

Definition 5 (Uniformity (U))**.** *Let* Π *be an IB-KEM with associated spaces* $\mathtt{IDSp}$*,* $\mathtt{CoinSp}$*,* $\mathtt{CipherSp}$*, and* $\mathtt{KeySp}$*. For* $k \in \mathbb{N}$*,* $\mathcal{H} \in \mathsf{Hash}(k)$*,* $(mpk, msk) \in \mathsf{Setup}^{\mathcal{H}}(1^k)$*,* $id \in \mathtt{IDSp}(k)$*, and* $c \in \mathtt{CipherSp}(k)$ *define*

$$\gamma^{\mathcal{H}}_{mpk}(id, c) = \Pr\left[r \xleftarrow{\$} \mathtt{CoinSp}(k); (c, \cdot) = \mathsf{Encap}^{\mathcal{H}}_{mpk}(id; r)\right] .$$

We say that Π *is* γ-uniform *for a function* $\gamma : \mathbb{N} \to \mathbb{R}^{\geq 0}$ *if for (almost) all* $k \in \mathbb{N}$*,* $\mathcal{H} \in \mathsf{Hash}(k)$*,* $(mpk, msk) \in \mathsf{Setup}^{\mathcal{H}}(1^k)$*,* $id \in \mathtt{IDSp}(k)$*, and* $c \in \mathtt{CipherSp}(k)$ *we have* $\gamma^{\mathcal{H}}_{mpk}(id, c) \leq \gamma(k)$*. IB-KEM* Π *is simply called* uniform (U) *if* Π *is* γ*-uniform for all non-negligible* γ*.*

A weaker notion of uniformity is that of *computational uniformity*, where the adversary's aim is to explicitly spot pairs (id, c) with $\gamma^{\mathcal{H}}_{mpk}(id, c) > \gamma(k)$. This is formalized in Definition 6.

Definition 6 (Computational uniformity (cU))**.** *Let* Π *be an IB-KEM and* $\mathcal{A}$ *be an algorithm. For a given function* $\gamma : \mathbb{N} \to \mathbb{R}^{\geq 0}$*, consider experiment* $\mathrm{Expt}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}$ *from Figure 6 and define the success probability of* $\mathcal{A}$ *as*

$$\mathrm{Succ}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k) = \Pr\left[\mathrm{Expt}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k) = \textsc{win}\right] .$$

IB-KEM Π *is said to be* computationally γ-uniform *if* $\mathrm{Succ}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k)$ *is negligible in* k*, for all adversaries* $\mathcal{A}$*. IB-KEM* Π *is simply called* computationally uniform (cU) *if* Π *is computationally* γ*-uniform for all non-negligible* γ*.*

$\mathrm{Expt}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k)$:

(a) $\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k)$

(b) $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}^{\mathcal{H}}(1^k)$

(c) $(id, c) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_X}(mpk)$
- Answer $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X$ queries as in Figure 2.

(d) If $\gamma^{\mathcal{H}}_{mpk}(id, c) > \gamma(k)$ return WIN.

(e) Return LOSE.

Figure 6: Experiment for computational uniformity (cU) of IB-KEMs

$\mathsf{Setup}^*(1^k)$:

(a) $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}(1^k)$

(b) $h^* \leftarrow ow(id^*)$ for $id^* \xleftarrow{\$} \mathtt{IDSp}(k)$

(c) $r^* \xleftarrow{\$} \mathtt{CoinSp}(k)$

(d) $mpk^* \leftarrow (mpk, h^*, r^*)$

(e) Return (mpk^*, msk).

$\mathsf{Extract}^*(msk, id)$:

(a) Return $\mathsf{Extract}(msk, id)$.

$\mathsf{Encap}^*_{mpk^*}(id; r)$:

(a) Parse $(mpk, h^*, r^*) \leftarrow mpk^*$.

(b) If $ow(id) = h^*$ return $\mathsf{Encap}_{mpk}(id; r^*)$.

(c) Return $\mathsf{Encap}_{mpk}(id; r)$.

$\mathsf{Decap}^*_{sk}(c)$:

(a) Return $\mathsf{Decap}_{sk}(c)$.

Figure 7: Constructing Π^* from Π. We assume that all queries to $\mathcal{H}$ oracle are relayed without modification.

We note that line (d) of $\mathrm{Expt}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k)$ in Figure 6 is not necessarily efficient, but the experiment is still well-defined.

We now analyze the relation between uniformity and computational uniformity. While Lemma 5 follows directly from the definitions, the proof of Lemma 6 is a bit more involved.

Lemma 5 ($\mathsf{U} \Rightarrow \mathsf{cU}$)**.** *Let Π be an IB-KEM and $\gamma : \mathbb{N} \to \mathbb{R}^{\geq 0}$ be a function. If Π is γ-uniform, then Π is computationally γ-uniform. In particular, if Π is uniform, then Π is computationally uniform.*

Lemma 6 ($\mathsf{cU} \not\Rightarrow \mathsf{U}$)**.** *Let γ be any function $\mathbb{N} \to [0, 1)$. Then not all computationally γ-uniform IB-KEMs are γ-uniform, assuming the existence of a one-way function $\mathtt{IDSp}(k) \to \{0,1\}^k$. In particular, if Π is computationally uniform, then Π is not necessarily uniform.*

Proof. Let Π be a γ-uniform IB-KEM and let $ow : \mathtt{IDSp}(k) \to \{0,1\}^k$ be a one-way function. Using Π and ow, construct IB-KEM Π^* according to Figure 7. The new scheme is mostly identical to Π, but for a very small set of identities the encapsulation algorithm is artificially made deterministic. We will show that Π^* is computationally γ-uniform, but not γ-uniform.

Consider identity id^* that is randomly picked in Setup^* and observe that encapsulation for id^* is deterministic (line (b) of Encap^*). More precisely, we have $\mathsf{Encap}^*_{mpk^*}(id^*; r) = (c, \cdot)$ for all $r \in \mathtt{CoinSp}(k)$, where $(c, \cdot) = \mathsf{Encap}^*_{mpk^*}(id^*; r^*)$. It follows that $\gamma^{\mathcal{H}}_{mpk^*}(id^*, c) = 1 > \gamma(k)$, i.e. Π^* is not γ-uniform.

However, Π^* is computationally γ-uniform: Note that for all identities $id \in \mathtt{IDSp}(k)$ with $ow(id) \neq h^* = ow(id^*)$ we have $\gamma^{\mathcal{H}}_{mpk^*}(id, c) = \gamma^{\mathcal{H}}_{mpk}(id, c) \leq \gamma(k)$, for all $c \in \mathtt{CipherSp}(k)$. Hence, a successful adversary against cU would have to provide $id \in \mathtt{IDSp}(k)$ such that $ow(id) = h^*$, i.e. break one-wayness of ow. This is infeasible by assumption. □

Surprisingly, computational uniformity is implied by OW security and can thus be assumed for any reasonable IB-KEM. To prove this statement, we introduce a new intermediate security notion, *(computational) collision uniformity* (cCU), and show in Lemmas 7 and 8 that OW implies cCU and that cCU implies cU.

$\mathrm{Expt}^{\mathsf{cCU},\Gamma}_{\Pi,\mathcal{B}}(k)$:

(a) $\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k)$

(b) $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}^{\mathcal{H}}(1^k)$

(c) $id \xleftarrow{\$} \mathcal{B}^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_X}(mpk)$

– Answer $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X$ queries as in Figure 2.

(d) If $\Gamma^{\mathcal{H}}_{mpk}(id) > \Gamma(k)$ return WIN.

(e) Return LOSE.

Figure 8: Experiment for computational collision uniformity (cCU) of IB-KEMs

We start by establishing the notion of *collision uniformity*. We only regard the computational variant, as its information-theoretical analogue is not needed in this paper. Intuitively, an IB-KEM Π is collision-uniform if collisions of ciphertexts occur rarely, i.e. if $(c_1, K_1) \leftarrow \mathsf{Encap}_{mpk}(id; r_1)$ and $(c_2, K_2) \leftarrow \mathsf{Encap}_{mpk}(id; r_2)$ are run with independently drawn random coins $r_1, r_2 \in \mathtt{CoinSp}(k)$, then $c_1 = c_2$ happens only with negligible probability.

Definition 7 (Computational collision uniformity (cCU))**.** *Let Π be an IB-KEM with associated spaces $\mathtt{IDSp}$, $\mathtt{CoinSp}$, $\mathtt{CipherSp}$, and $\mathtt{KeySp}$. For $k \in \mathbb{N}$, $\mathcal{H} \in \mathsf{Hash}(k)$, $(mpk, msk) \in \mathsf{Setup}^{\mathcal{H}}(1^k)$, and $id \in \mathtt{IDSp}(k)$ define*

$$\Gamma^{\mathcal{H}}_{mpk}(id) = \Pr\left[r_1, r_2 \xleftarrow{\$} \mathtt{CoinSp}(k); \mathsf{Encap}^{\mathcal{H}}_{mpk}(id; r_1) = \mathsf{Encap}^{\mathcal{H}}_{mpk}(id; r_2)\right] .$$

For a given function $\Gamma : \mathbb{N} \to \mathbb{R}^{\geq 0}$, consider experiment $\mathrm{Expt}^{\mathsf{cCU},\Gamma}_{\Pi,\mathcal{B}}$ from Figure 8. IB-KEM Π is said to be computationally Γ-collision-uniform *if, for all algorithms $\mathcal{B}$, success probability*

$$\mathrm{Succ}^{\mathsf{cCU},\Gamma}_{\Pi,\mathcal{B}}(k) = \Pr\left[\mathrm{Expt}^{\mathsf{cCU},\Gamma}_{\Pi,\mathcal{B}}(k) = \text{WIN}\right]$$

is negligible in k. A scheme Π is simply called computationally collision-uniform (cCU) *if Π is computationally Γ-collision-uniform for all non-negligible Γ.*

Similarly to what we pointed out in the context of Definition 6, line (d) of $\mathrm{Expt}^{\mathsf{cCU},\Gamma}_{\Pi,\mathcal{B}}(k)$ in Figure 8 is not necessarily efficient, but the experiment is still well-defined.

The following lemma shows that computational collision uniformity is stronger than plain computational uniformity:

Lemma 7 ($\mathsf{cCU} \Rightarrow \mathsf{cU}$)**.** *Let Π be an IB-KEM and $\gamma, \Gamma : \mathbb{N} \to \mathbb{R}^{\geq 0}$ be functions such that $\Gamma(k) = \gamma^2(k)$ for all k. If Π is computationally Γ-collision-uniform, then Π is computationally γ-uniform. In particular, if Π is computationally collision-uniform, then Π is computationally uniform.*

Proof. For any $k \in \mathbb{N}$, $\mathcal{H} \in \mathsf{Hash}(k)$, $(mpk, msk) \in \mathsf{Setup}^{\mathcal{H}}(1^k)$, $id \in \mathtt{IDSp}(k)$, and $c \in \mathtt{CipherSp}(k)$ we have

$$\begin{aligned}\Gamma^{\mathcal{H}}_{mpk}(id) &\geq \Pr\left[r_1, r_2 \xleftarrow{\$} \mathtt{CoinSp}(k); \mathsf{Encap}^{\mathcal{H}}_{mpk}(id; r_1) = (c, \cdot) = \mathsf{Encap}^{\mathcal{H}}_{mpk}(id; r_2)\right] \\ &= \gamma^{\mathcal{H}}_{mpk}(id, c)^2.\end{aligned}$$

Now, if adversary $\mathcal{A}$ against computational γ-uniformity manages to output a pair (id, c) such that $\gamma^{\mathcal{H}}_{mpk}(id, c) > \gamma(k)$, then $\Gamma^{\mathcal{H}}_{mpk}(id) \geq \gamma^{\mathcal{H}}_{mpk}(id, c)^2 > \gamma^2(k) = \Gamma(k)$ holds as well. In other words, an adversary $\mathcal{B}$ against computational Γ-collision-uniformity is given by a modified adversary $\mathcal{A}$ that outputs only id but not c. Observe that $\mathcal{B}$ has at least the same success probability as $\mathcal{A}$. □

$\mathcal{A}_1^{\mathcal{O}'_{\mathcal{H}},\mathcal{O}'_X}(mpk)$:
(a) $id^* \leftarrow \mathcal{B}^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_X}(mpk)$
– Relay $\mathcal{O}_{\mathcal{H}}, \mathcal{O}_X$ queries to $\mathcal{A}_1$.
(b) $st \leftarrow (mpk, id^*)$
(c) Return (id^*, st).

$\mathcal{A}_2^{\mathcal{O}'_{\mathcal{H}},\mathcal{O}'_X}(st, c^*)$:
(a) Parse $(mpk, id^*) \leftarrow st$.
(b) $(c', K') \xleftarrow{\$} \mathsf{Encap}_{mpk}^{\mathcal{H}}(id^*)$
(c) If $c' \neq c^*$ return $\perp$.
(d) Return K'.

Figure 9: Constructing OW adversary $\mathcal{A}$ from collision-uniformity adversary $\mathcal{B}$

Remark 7 (Colliding ciphertexts imply matching keys)**.** *In the proof of Lemma 7, equality "*$\mathsf{Encap}_{mpk}^{\mathcal{H}}(id; r_1) = (c, \cdot) = \mathsf{Encap}_{mpk}^{\mathcal{H}}(id; r_2)$*" expresses that the ciphertexts output by* Encap *shall collide, but a requirement on the corresponding keys K is not explicitly given. Observe that it follows from correctness of IB-KEM that in case of colliding ciphertexts also keys K will match.*

Remark 8 (Equivalent variants of Definitions 6 and 7)**.** *When coining the notion of computational uniformity in Definition 6, we allowed adversary $\mathcal{A}$ to query its $\mathcal{O}_X$ oracle on any identity, including on the identity id output by $\mathcal{A}$. In the next paragraph, we justify that this definition is equivalent to one where $\mathcal{A}$ is forbidden to query $\mathcal{O}_X$ on id (but continues to be allowed to query $\mathcal{O}_X$ on all other identities in* $\mathtt{IDSp}(k)$*). The same observation applies to the notion of computational collision-uniformity in Definition 7.*

Let $\mathcal{A}$ be a cU *adversary in accordance with Definition 6. Without loss of generality we may assume that $\mathcal{A}$ queries id to its $\mathcal{O}_X$ oracle. Let q_X denote the total number of (distinct) $\mathcal{O}_X$ queries that $\mathcal{A}$ poses. We construct* cU *adversary $\mathcal{A}'$ from $\mathcal{A}$ as follows: In a first step, $\mathcal{A}'$ picks a number $1 \leq t \leq q_X$ at random, basically guessing the number of the $\mathcal{O}_X$ query which $\mathcal{A}$ will pose to ask $\mathcal{O}_X(id)$. Then $\mathcal{A}'$ simulates $\mathcal{A}$ until reaching the t-th $\mathcal{O}_X$ query, relaying the first $t-1$ queries to its own $\mathcal{O}_X$ oracle. Let id_t be the identity revealed in the t-th query. Adversary $\mathcal{A}'$ outputs id_t and stops. The tightness factor appearing in the analysis of this reduction is q_X. Nevertheless, this shows that the notions are asymptotically equivalent.*

We prove next that one-wayness implies collision uniformity. Intuitively, if cCU security of an IB-KEM Π is not given, then an OW adversary $\mathcal{A}$ could simply run $(c', K') \xleftarrow{\$} \mathsf{Encap}_{mpk}(id^*)$ and would hit the challenge ciphertext with non-negligible probability, i.e. $c' = c^*$. As matching ciphertexts imply matching keys (cf. Remark 7), $\mathcal{A}$ can break OW security by just outputting K'. We formalize this idea as follows:

Lemma 8 (OW $\Rightarrow$ cCU)**.** *Let Π be an IB-KEM. If Π is* OW*-secure, then Π is computationally collision-uniform.*

Proof. Assume that Π is not computationally collision-uniform, i.e. there exists a non-negligible Γ and a successful adversary $\mathcal{B}$ against Γ-collision-uniformity. Given $\mathcal{B}$, we construct an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against OW of Π, as shown in Figure 9. Observe that $\mathcal{A}_1$ is well-defined due to Remark 8.

Consider line (c) of the specification of adversary $\mathcal{A}_2$ and denote by E the event that $c' = c^*$. Remark 7 shows that occurrence of E implies $K' = K^*$, i.e. correct decapsulation of c^* by $\mathcal{A}_2$. Careful comparison of Definition 7 with line (d) of $\mathrm{Expt}^{\mathsf{OW}}$ (Figure 2) and line (b) of $\mathcal{A}_2$ reveals that the probability for event E to occur is exactly $\Gamma_{mpk}^{\mathcal{H}}(id^*)$. Moreover, denoting by B the event that $\Gamma_{mpk}^{\mathcal{H}}(id^*) > \Gamma(k)$ occurs, we know that B happens with probability $\mathrm{Succ}_{\Pi,\mathcal{B}}^{\mathsf{cCU},\Gamma}(k)$ (cf. line (a) of $\mathcal{A}_1$ and Definition 7). That is, we constructed an adversary $\mathcal{A}$ against OW of Π with

$$\begin{aligned}\mathrm{Succ}_{\Pi,\mathcal{A}}^{\mathsf{OW}}(k) = \Pr[E] &\geq \Pr[E \wedge B] = \Pr[E|B]\Pr[B] \\ &> \Gamma(k) \cdot \mathrm{Succ}_{\Pi,\mathcal{B}}^{\mathsf{cCU},\Gamma}(k) > \mathrm{negl}(k) \ .\end{aligned}$$

In particular, Π is not OW-secure. □

$\mathsf{Setup}^*(1^k)$:
(a) Return $\mathsf{Setup}(1^k)$.

$\mathsf{Encap}^*_{mpk}(id; r)$ for $r \in \mathtt{KeySp}(k)$:
(a) $r_1 \leftarrow H(id, r)$
(b) $(c_1, K) \leftarrow \mathsf{Encap}_{mpk}(id; r_1)$
(c) $c_2 \leftarrow r + K$
(d) $c \leftarrow (c_1, c_2)$
(e) Return (c, K).

$\mathsf{Extract}^*(msk, id)$:
(a) Return $\mathsf{Extract}(msk, id)$.

$\mathsf{Decap}^*_{sk}(c)$:
(a) Parse $(c_1, c_2) \leftarrow c$.
(b) $\hat{K} \leftarrow \mathsf{Decap}_{sk}(c_1)$
(c) If $\hat{K} = \perp$ return $\perp$.
(d) $\hat{r} \leftarrow c_2 - \hat{K}$
(e) $\hat{r}_1 \leftarrow H(id, \hat{r})$
(f) If $(c_1, \cdot) \neq \mathsf{Encap}_{mpk}(id; \hat{r}_1)$ return $\perp$.
(g) Return $\hat{K}$.

Figure 10: Constructing $\mathcal{F}(\Pi)$ from Π. Queries to hash functions other than H are relayed without modification.

Taken together, Lemmas 7 and 8 establish the following corollary:

Corollary 1 ($\mathsf{OW} \Rightarrow \mathsf{cU}$)**.** *Let* Π *be an IB-KEM. If* Π *is* OW*-secure, then* Π *is computationally uniform.*

For completeness we show that the notions OW and cU are not equivalent, i.e. that OW is strictly stronger than cU.

Lemma 9 ($\mathsf{cU} \not\Rightarrow \mathsf{OW}$)**.** *Not all computationally uniform IB-KEMs are* OW*-secure.*

Proof. Consider the (correct, but insecure) IB-KEM in which $\mathsf{Encap}_{mpk}(id; r)$ returns $(c, K) = (r, r)$, i.e. keys are equal to their encapsulation. The scheme is clearly not OW-secure. However, it is computationally uniform, provided that $\mathtt{CoinSp}(k)$ is super-polynomially large. □

5 Obtaining plaintext awareness for IB-KEMs

After seeing in Sections 1 and 3 that plaintext awareness is a valuable property of (IB-)KEMs, the question arises on how this property can be achieved. For this purpose, we introduce a generic transformation $\mathcal{F}$ that (a) converts a computationally uniform IB-KEM into a plaintext-aware scheme, and (b) strengthens its security from OW to OW-CCA. Our transformation $\mathcal{F}$ is inspired by the Fujisaki-Okamoto (FO) approach [FO99b] for PKE schemes, but with some differences in structure and properties. Unlike [FO99b], our transformation does not achieve the highest level of security, IND-CCA, directly, but when combined with the hash transformation # in Section 6, we obtain IND-CCA security *and* plaintext awareness simultaneously.

Definition 8 (Transformation $\mathcal{F}$)**.** *Let* $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encap}, \mathsf{Decap})$ *be an IB-KEM with random oracles* $\mathcal{H}$ *and associated spaces* $\mathtt{IDSp}$, $\mathtt{CoinSp}$, $\mathtt{CipherSp}$, *and* $\mathtt{KeySp}$, *where* $(\mathtt{KeySp}, +)$ *forms a group. Let* $H : \mathtt{IDSp} \times \mathtt{KeySp} \to \mathtt{CoinSp}$ *be a new hash function (independent of* $\mathcal{H}$*), modeled as a random oracle in the security analysis. Then IB-KEM* $\mathcal{F}(\Pi) = (\mathsf{Setup}^*, \mathsf{Extract}^*, \mathsf{Encap}^*, \mathsf{Decap}^*)$ *is specified in Figure 10, with random oracles* $\mathcal{H}' = \mathcal{H} \cup \{H\}$ *and associated spaces* $\mathtt{IDSp}^* = \mathtt{IDSp}$, $\mathtt{CoinSp}^* = \mathtt{KeySp}$, $\mathtt{CipherSp}^* = \mathtt{CipherSp} \times \mathtt{KeySp}$, $\mathtt{KeySp}^* = \mathtt{KeySp}$.

It is easy to verify that, if Π is a correct IB-KEM, then so is $\mathcal{F}(\Pi)$. Observe that Definition 8 views $\mathtt{KeySp}$ as a group. This is quite common in practice, as fixed-length bit strings with the XOR operation form a group. Alternatively, as detailed in Section 7.1, one can simply hash the key to a fixed-length bit string.

We will see in Section 7.1 that even if $\mathtt{KeySp}$ is already a group, hashing to a bit string may provide a performance optimization which is attractive for the design of practical schemes.

Compared to the original FO [FO99b], our transformation $\mathcal{F}$ has many similarities, but a few differences as well. Considering that KEMs are not fully-fledged encryption schemes and thus

need not process arbitrary-length messages but rather keys from a finite key set, our construction eschews symmetric encryption (which is used for messages in hybrid encryption schemes) for a simple one-time pad of the key K, with randomness r taking the role of the pad. The latter is necessary for verifying the decapsulation. These changes lead to a proof of security for $\mathcal{F}$, as we will see, under weaker and more natural assumptions as those used in [FO99b].

We can now move to our main technical result regarding the $\mathcal{F}$ transformation: it results in an IB-KEM that is plaintext-aware, assuming the original scheme was computationally uniform. The plaintext awareness comes from the query to the new random oracle H, i.e. the ciphertext will not decapsulate unless the intermediate value r was queried to H, in which case that same value r can be used to derive the key.

Theorem 2 ($\mathcal{F} : \mathsf{cU} \mapsto \mathsf{PA}$)**.** *Let* Π *be an IB-KEM. If* Π *is computationally uniform, then* $\mathcal{F}(\Pi)$ *is plaintext-aware.*

Proof. We use notation from Figure 10: $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encap}, \mathsf{Decap})$ and $\mathcal{F}(\Pi) = (\mathsf{Setup}^*, \mathsf{Extract}^*, \mathsf{Encap}^*, \mathsf{Decap}^*)$. Our goal is to specify a plaintext extractor $\mathcal{K}$ for $\mathcal{F}(\Pi)$ with awareness $\mathrm{Aw}^{\mathsf{PA}}_{\mathcal{F}(\Pi),\mathcal{A},\mathcal{K}}$ overwhelming for all ciphertext creators $\mathcal{A}$. This is the case if, in $\mathrm{Expt}^{\mathsf{PA}}_{\mathcal{F}(\Pi),\mathcal{A},\mathcal{K}}$, the output of $\mathcal{K}$ matches the output of $\mathsf{Decap}^*_{sk^*}(c^*)$ except for some negligible probability, where sk^* denotes the secret key corresponding to identity id^* and (id^*, c^*) was output by ciphertext creator $\mathcal{A}$. Consider the following plaintext extractor:

$\mathcal{K}^{\mathcal{H}}(id^*, c^*, mpk, \mathsf{HList}, \mathsf{XList}, \mathsf{EList})$:
- (a) Parse $(c_1^*, c_2^*) \leftarrow c^*$.
- (b) $\mathcal{R} \leftarrow \{(r, r_1) : (H, (id^*, r), r_1) \in \mathsf{HList}\}$
- (c) For each $(r, r_1) \in \mathcal{R}$:
  - (a) $(c, K) \leftarrow \mathsf{Encap}_{mpk}(id^*; r_1)$
  - (b) If $(c_1^* = c) \wedge (r = c_2^* - K)$, return K.
- (d) Return $\perp$.

Let r^* and $r_1^* = H(id^*, r^*)$ be the values $(\hat{r}, \hat{r}_1)$ that occur in lines (d) and (e) of algorithm Decap^* when executing $\mathsf{Decap}^*_{sk^*}(c^*)$. By close inspection of Decap^* we observe that $\mathsf{Decap}^*_{sk^*}$ on input $c^* = (c_1^*, c_2^*)$ outputs a key K^* $(\neq \perp)$ if and only if

$$\big((c_1^*, K^*) = \mathsf{Encap}_{mpk}(id^*; r_1^*)\big) \wedge (r^* = c_2^* - K^*) \quad . \tag{1}$$

Given that it exists, this key K^* is unique and can be readily computed given r_1^*. To analyze the effectiveness of $\mathcal{K}$, we distinguish three cases:

(1) r^* is queried to $H(id^*, \cdot)$ by $\mathcal{A}$, i.e. $(r^*, r_1^*) \in \mathcal{R}$ (cf. line (b) of $\mathcal{K}$).
Plaintext extractor $\mathcal{K}$ has access to $\mathcal{R}$ and can, using equation (1), easily decide whether $\mathsf{Decap}^*_{sk^*}$ will output a key or not. If it will, then $\mathcal{K}$ computes and outputs the same key. This is handled in line (c) of $\mathcal{K}$.

(2) r^* is queried to $H(id^*, \cdot)$ from within an $\mathcal{O}_E(id^*)$ query.
In this case, there exists some $(id^*, \hat{c}) \in \mathsf{EList}$ where $(\hat{c}, \hat{K}) \leftarrow \mathsf{Encap}^*_{mpk}(id^*; r^*)$. Let $\hat{c} = (\hat{c}_1, \hat{c}_2)$. Clearly, by line (f) of Decap^*, if $\hat{c}_1 \neq c_1^*$ then $\mathsf{Decap}^*_{sk^*}(c^*)$ will output $\perp$. Case $\hat{c}_1 = c_1^*$ will not occur as it would imply $\hat{c}_2 = c_2^*$ (cf. line (d) in Decap^* and recall that $\hat{K}$ and r^* are already fixed), but $\hat{c} = c^*$ is prohibited by line (d) of $\mathrm{Expt}^{\mathsf{PA}}_{\mathcal{F}(\Pi),\mathcal{A},\mathcal{K}}$.

(3) r^* is not queried to $H(id^*, \cdot)$ before reaching line (e) of $\mathrm{Expt}^{\mathsf{PA}}_{\mathcal{F}(\Pi),\mathcal{A},\mathcal{K}}(k)$.
In this case, r_1^* is randomly assigned when executing $\mathsf{Decap}^*_{sk^*}(c^*)$. Therefore $(c_1^*, \cdot) = \mathsf{Encap}_{mpk}(id^*, r_1^*)$ (cf. line (f) of Decap^*) occurs only with negligible probability, due to our assumption that Π is computationally uniform. (Otherwise, this experiment combined with $\mathcal{A}$ would constitute an efficient algorithm for finding, with non-negligible probability, an identity id^* for which $\gamma^{\mathcal{H}}_{mpk}(id^*, c^*)$ is non-negligible.) Thus, $\mathsf{Decap}^*_{sk^*}(c^*)$ will return $\perp$ with overwhelming probability.

We see that there is a perfect match of the decapsulation capabilities of $\mathsf{Decap}^*_{sk^*}$ and $\mathcal{K}$ in cases (1) and (2). In case (3), $\mathcal{K}$ correctly simulates the behavior of $\mathsf{Decap}^*_{sk^*}$ with overwhelming probability. This shows that $\mathcal{F}(\Pi)$ is plaintext-aware. □

Having shown that $\mathcal{F}$ introduces plaintext awareness, we now examine its secrecy properties, aiming to prove that $\mathcal{F}$ transforms any OW-secure IB-KEM into an OW-CCA-secure scheme. First we will show that $\mathcal{F}$ *preserves* OW security.

Lemma 10 ($\mathcal{F} : \mathsf{OW} \mapsto \mathsf{OW}$)**.** *Let* Π *be an IB-KEM. If* Π *is* OW*-secure, then* $\mathcal{F}(\Pi)$ *is* OW*-secure.*

Proof. The proof in general proceeds as a sequence of games in which related values are gradually decoupled. Although the concept is straightforward, the proof requires some care.

We define a sequence of games. Game G_1 will be the original OW experiment $\mathrm{Expt}^{\mathsf{OW}}_{\mathcal{F}(\Pi),\mathcal{A}}(k)$ for $\mathcal{F}(\Pi)$. In subsequent games, we will modify how the challenger generates the challenge ciphertext to allow it to, at various stages, insert a challenge from an OW challenger for Π. This will allow us to show that an adversary that can break OW of $\mathcal{F}(\Pi)$ can be used to break OW of Π. In Figure 11, we specify precisely how the challenge ciphertext is generated in each game. This corresponds to line (d) in the original $\mathrm{Expt}^{\mathsf{OW}}_{\mathcal{F}(\Pi),\mathcal{A}}(k)$ experiment defined in Figure 2. For each game G_i, we define S_i to be the probability that game G_i returns WIN; our goal is to bound S_1. The first several games are simple rewriting steps.

Game G_2. We replace random oracle $H : \mathtt{IDSp}(k) \times \mathtt{KeySp}(k) \to \mathtt{CoinSp}(k)$ with the composition of a random function $H' : \mathtt{IDSp}(k) \times \mathtt{KeySp}(k) \to \mathtt{CoinSp}(k)$ and a random permutation $P : \mathtt{KeySp}(k) \to \mathtt{KeySp}(k)$, and give $\mathcal{A}$ oracle access to $H' \circ P$ which on queries of the form $(id, r) \in \mathtt{IDSp}(k) \times \mathtt{KeySp}(k)$ returns $H'(id, P(r))$. Since $H' \circ P$ has the same distribution as H, $S_1 = S_2$.

Game G_3. We rewrite the generation of the input to the hash function in terms of a separate variable r'. This is a bridging step, so $S_2 = S_3$.

Game G_4. We simplify the generation of the variable r'. Since P is a permutation, this does not change the distribution, and hence $S_3 = S_4$.

Game G_5. The hop from game G_4 to game G_5 contains the essential step of generating r_1 independently from r'. This serves to decouple ciphertext components c_1 and c_2. We will show that this change cannot be detected unless Π is not OW.

Let $\mathcal{A}$ be an algorithm that can distinguish G_4 from G_5. To distinguish the two games, $\mathcal{A}$ needs to induce an (id^*, r') query to H'. Since $\mathcal{A}$ only has access to $H' \circ P$, and since P is a permutation, the only way $\mathcal{A}$ can induce such a query to H' is if $\mathcal{A}$ queries $(id^*, P^{-1}(r'))$ to $H' \circ P$. Observe that $P^{-1}(r') = c_2 - \mathsf{Decap}_{sk}(c_1)$ in both G_4 and G_5. Hence, given (c_1, c_2), distributed either as in G_4 or as in G_5, $\mathcal{A}$ queries $(id^*, c_2 - \mathsf{Decap}_{sk}(c_1))$ to $H' \circ P$. Observe in addition that c_1 and c_2 are statistically independent in G_5.

We now use such an $\mathcal{A}$ to break the OW security of Π. Let $\hat{c}^*$ be an OW challenge for Π. Choose $\hat{c}^*_2 \xleftarrow{\$} \mathtt{KeySp}(k)$. This pair $(\hat{c}^*, \hat{c}^*_2)$ is distributed as in G_5. Hence, when $\mathcal{A}$ is run on $(\hat{c}^*, \hat{c}^*_2)$, $\mathcal{A}$ will query its oracle $H' \circ P$ on $(id^*, \hat{c}^*_2 - \mathsf{Decap}_{sk}(\hat{c}^*))$. From the list of queries to the oracle $H' \circ P$, pick a query (id, z) at random, constraint to $id = id^*$. Return $\hat{c}^*_2 - z$ as the guess for the key that corresponds to the challenge ciphertext $\hat{c}^*$.

We have shown how to use a distinguisher $\mathcal{A}$ for G_4 and G_5 to construct an algorithm $\mathcal{B}$ for breaking the OW security of Π. Hence, $|S_4 - S_5| \leq q_H \cdot \mathrm{Succ}^{\mathsf{OW}}_{\Pi,\mathcal{B}}(k)$, where q_H denotes the total number of queries to $H' \circ P$. This difference is negligible, assuming that Π is OW-secure.

Game G_6. We replace c_2 with a random value. In G_5, r' is independent of c_1 so c_2 is independent of c_1, and moreover r' is uniformly random, so $P^{-1}(r')$ is uniform and hence c_2 is uniform. In G_6, c_2 is clearly independent of c_1 and is clearly uniform. Thus, the distributions of (c_1, c_2) in G_5 and G_6 are identical, so $S_5 = S_6$.

Finally, we need to bound S_6. Suppose $\mathcal{A}$ wins in game G_6. We run $\mathcal{A}$ on the same environment as G_6 but this time we replace ciphertext component c_1 in G_6 with the challenge

$G_1 = \mathrm{Expt}^{\mathsf{OW}}_{\mathcal{F}(\Pi),\mathcal{A}}(k)$
...
(c) $(id^*, st) \xleftarrow{\$} \mathcal{A}_1^{H,\mathcal{O}_X}(mpk)$
(d).1 $r \xleftarrow{\$} \mathtt{KeySp}(k)$
(d).2 $r_1 \leftarrow H(id^*, r)$
(d).3 $(c_1, K^*) \leftarrow \mathsf{Encap}(id^*; r_1)$
(d).4 $c_2 \leftarrow r + K^*$
(d).5 $c^* \leftarrow (c_1, c_2)$
(e) $K' \xleftarrow{\$} \mathcal{A}_2^{H,\mathcal{O}_X}(st, c^*)$
...

G_2
...
(c) $(id^*, st) \xleftarrow{\$} \mathcal{A}_1^{H'\circ P,\mathcal{O}_X}(mpk)$
(d).1 $r \xleftarrow{\$} \mathtt{KeySp}(k)$
(d).2 $r_1 \leftarrow H'(id^*, P(r))$
(d).3 $(c_1, K^*) \leftarrow \mathsf{Encap}(id^*; r_1)$
(d).4 $c_2 \leftarrow r + K^*$
(d).5 $c^* \leftarrow (c_1, c_2)$
(e) $K' \xleftarrow{\$} \mathcal{A}_2^{H'\circ P,\mathcal{O}_X}(st, c^*)$
...

G_3
...
(c) $(id^*, st) \xleftarrow{\$} \mathcal{A}_1^{H'\circ P,\mathcal{O}_X}(mpk)$
(d).1 $r \xleftarrow{\$} \mathtt{KeySp}(k)$
(d).2 $r' \leftarrow P(r)$
(d).3 $r_1 \leftarrow H'(id^*, r')$
(d).4 $(c_1, K^*) \leftarrow \mathsf{Encap}(id^*; r_1)$
(d).5 $c_2 \leftarrow P^{-1}(r') + K^*$
(d).6 $c^* \leftarrow (c_1, c_2)$
(e) $K' \xleftarrow{\$} \mathcal{A}_2^{H'\circ P,\mathcal{O}_X}(st, c^*)$
...

G_4
...
(c) $(id^*, st) \xleftarrow{\$} \mathcal{A}_1^{H'\circ P,\mathcal{O}_X}(mpk)$
(d).1 $r' \xleftarrow{\$} \mathtt{KeySp}(k)$
(d).2 $r_1 \leftarrow H'(id^*, r')$
(d).3 $(c_1, K^*) \leftarrow \mathsf{Encap}(id^*; r_1)$
(d).4 $c_2 \leftarrow P^{-1}(r') + K^*$
(d).5 $c^* \leftarrow (c_1, c_2)$
(e) $K' \xleftarrow{\$} \mathcal{A}_2^{H'\circ P,\mathcal{O}_X}(st, c^*)$
...

G_5
...
(c) $(id^*, st) \xleftarrow{\$} \mathcal{A}_1^{H'\circ P,\mathcal{O}_X}(mpk)$
(d).1 $r' \xleftarrow{\$} \mathtt{KeySp}(k)$
(d).2 $r_1 \xleftarrow{\$} \mathtt{CoinSp}(k)$
(d).3 $(c_1, K^*) \leftarrow \mathsf{Encap}(id^*; r_1)$
(d).4 $c_2 \leftarrow P^{-1}(r') + K^*$
(d).5 $c^* \leftarrow (c_1, c_2)$
(e) $K' \xleftarrow{\$} \mathcal{A}_2^{H'\circ P,\mathcal{O}_X}(st, c^*)$
...

G_6
...
(c) $(id^*, st) \xleftarrow{\$} \mathcal{A}_1^{H'\circ P,\mathcal{O}_X}(mpk)$
(d).1 $r_1 \xleftarrow{\$} \mathtt{CoinSp}(k)$
(d).2 $(c_1, K^*) \leftarrow \mathsf{Encap}(id^*; r_1)$
(d).3 $c_2 \xleftarrow{\$} \mathtt{KeySp}(k)$
(d).4 $c^* \leftarrow (c_1, c_2)$
(e) $K' \xleftarrow{\$} \mathcal{A}_2^{H'\circ P,\mathcal{O}_X}(st, c^*)$
...

Figure 11: Sequence of games for the proof of Lemma 10, showing lines changed from $\mathrm{Expt}^{\mathsf{OW}}_{\mathcal{F}(\Pi),\mathcal{A}}(k)$ in Figure 2. Changes between games are marked. H' is a random function and P is a random permutation.

from an OW challenger for Π. This exactly matches the distribution in G_6. Thus, $\mathcal{A}$ can be used to break OW security of Π, and thus we have an algorithm $\mathcal{B}$ such that $S_6 = \mathrm{Succ}^{\mathsf{OW}}_{\Pi,\mathcal{B}}(k)$ which is negligible, assuming Π is OW-secure.

Combining the intermediate results demonstrates that $S_1 = \mathrm{Succ}^{\mathsf{OW}}_{\mathcal{F}(\Pi),\mathcal{A}}(k)$ is negligible, assuming Π is OW-secure. □

Remark 9 (Better tightness in proof of Lemma 10)**.** *In Lemma 10, we saw that the $\mathcal{F}$ construction preserves one-way-ness. The proof introduces a gap in tightness due to the random choice of z from the q_H queries to $H' \circ P$ in the analysis of the hop from G_4 to G_5. However, if Π is in fact* IND*-secure, then we can achieve a tight reduction: in the* IND *experiment, $\mathcal{B}_2$ is given a potential key K and can check whether that K is equal to the one output by $\mathcal{A}_2$, or whether K is consistent with one of the queries to $H' \circ P$. If one of these is the case, then $\mathcal{B}_2$ answers that K is a real key, otherwise $\mathcal{B}_2$ answers that K is a random value.*

We can now combine the results of this section with some previous results to show that the $\mathcal{F}$ transformation results in an IB-KEM which is both plaintext-aware and OW-CCA-secure.

Corollary 2 ($\mathcal{F} : \mathsf{OW} \mapsto \mathsf{PA} \wedge \mathsf{OW\text{-}CCA}$)**.** *Let Π be an IB-KEM. If Π is* OW*-secure, then $\mathcal{F}(\Pi)$ is plaintext-aware and* OW-CCA*-secure.*

Proof. Since Π is OW, then by Corollary 1 it is also computationally uniform. By Theorem 2, $\mathcal{F}$ converts a computationally uniform scheme into a PA scheme. By Lemma 10, $\mathcal{F}$ preserves OW. Thus we have that $\mathcal{F}(\Pi)$ is OW and PA, by Theorem 1 it is also OW-CCA. □

Corollary 2 stands in contrast to the original FO transformation [FO99b], which required uniformity as a separate precondition in addition to OW; we have identified a (strictly) weaker precondition, computational uniformity, which follows from OW security.

Given an identity id, a ciphertext $c = (c_1, c_2)$, and a potential corresponding key K, anyone can check if K does in fact correspond to c: compute $\hat{r} \leftarrow c_2 - K$, $\hat{r}_1 \leftarrow H(id, \hat{r})$, and check if (c_1, K) is equal to $\mathsf{Encap}^*_{mpk}(id; \hat{r}_1)$. In other words, there is a public consistency check of ciphertexts and keys. As a result, $\mathcal{F}(\Pi)$ can never be secure in the sense of an indistinguishability experiment such as IND or IND-CCA, which proves the following lemma:

Lemma 11 ($\mathcal{F}(\Pi)$ never IND)**.** *For any IB-KEM Π, $\mathcal{F}(\Pi)$ is not* IND*-secure.*

This result leads to an important observation contrasting random oracle model results from standard model results. Teranishi and Ogata [TO06] showed that, for public key encryption in the standard model, $\mathsf{OW} \wedge \mathsf{PA2} \Rightarrow \mathsf{IND\text{-}CCA}$. Corollary 2 and Lemma 11 show that this is not the case for IB-KEMs in the random oracle model: assuming Π is OW, we have that $\mathcal{F}(\Pi)$ is PA and OW, but $\mathcal{F}(\Pi)$ is not IND.

Remark 10 (On the implication $\mathsf{OW} \wedge \mathsf{PA} \Rightarrow \mathsf{IND\text{-}CCA}$ from [TO06])**.** *We briefly discuss the intuition behind the surprising implication $\mathsf{OW} \wedge \mathsf{PA} \Rightarrow \mathsf{IND}$[-CCA], discovered in [TO06] in the context of public key encryption and plaintext awareness defined in the standard model [BP04]. In the proof, assuming towards contradiction that the encryption scheme is not* IND*-secure, the authors construct a (stateful) plaintext creator $\mathcal{P}$ that encrypts a random message and transmits resulting ciphertext c to* PA *adversary $\mathcal{A}$ via a series of $\mathcal{O}_E$ queries. This ciphertext is output by $\mathcal{A}$, and obviously no plaintext extractor $\mathcal{K}$ can recover the encrypted message. Regarding a standard model definition of plaintext awareness for KEMs, we doubt that above implication would hold as well. The key argument is that there would be no plaintext creator $\mathcal{P}$, which plays the central role in the attack scenario described above.*

$\mathsf{Setup}^{\#}(1^k)$:
(a) Return $\mathsf{Setup}(1^k)$.

$\mathsf{Encap}^{\#}_{mpk}(id; r)$:
(a) $(c, \hat{K}) \leftarrow \mathsf{Encap}_{mpk}(id; r)$
(b) $K \leftarrow H^{\#}(\hat{K})$
(c) Return (c, K).

$\mathsf{Extract}^{\#}(msk, id)$:
(a) Return $\mathsf{Extract}(msk, id)$.

$\mathsf{Decap}^{\#}_{sk}(c)$:
(a) $\hat{K} \leftarrow \mathsf{Decap}_{sk}(c)$
(b) If $\hat{K} = \perp$ return $\perp$.
(c) $K \leftarrow H^{\#}(\hat{K})$
(d) Return K.

Figure 12: Constructing $\#(\Pi)$ from Π. Queries to hash functions other than $H^{\#}$ are relayed without modification.

6 From one-way security to indistinguishability

We present now a simple transformation — hashing an IB-KEM's key — that allows us to provably jump from OW to IND security, and from OW-CCA to IND-CCA. Bear in mind that our $\mathcal{F}$ transform does not achieve IND(-CCA) security and so # will play an important role in achieving this property. We admit that this technique has been used earlier for the same purpose [BFMLS08, Den03]; however, we additionally prove that it preserves plaintext awareness.

Definition 9 (Transformation #)**.** *Let* $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encap}, \mathsf{Decap})$ *be an IB-KEM with random oracles* $\mathcal{H}$ *and associated spaces* $\mathtt{IDSp}$, $\mathtt{CoinSp}$, $\mathtt{CipherSp}$, *and* $\mathtt{KeySp}$. *Let* $H^{\#} : \mathtt{KeySp} \to \mathtt{KeySp}$ *be a new hash function (independent of* $\mathcal{H}$*), modeled as random oracle in the security analysis. Then (hashed) IB-KEM* $\#(\Pi) = (\mathsf{Setup}^{\#}, \mathsf{Extract}^{\#}, \mathsf{Encap}^{\#}, \mathsf{Decap}^{\#})$ *is specified in Figure 12, with random oracles* $\mathcal{H}' = \mathcal{H} \cup \{H^{\#}\}$ *and associated spaces* $\mathtt{IDSp}^{\#} = \mathtt{IDSp}$, $\mathtt{CoinSp}^{\#} = \mathtt{CoinSp}$, $\mathtt{CipherSp}^{\#} = \mathtt{CipherSp}$, *and* $\mathtt{KeySp}^{\#} = \mathtt{KeySp}$.

It is easy to verify that, if Π is a correct IB-KEM, then so is $\#(\Pi)$. The following Lemmas 12 and 13 show that # converts arbitrary OW-secure schemes into IND-secure schemes and preserves plaintext awareness.

Lemma 12 ($\# : \mathsf{OW}[\text{-CCA}] \mapsto \mathsf{IND}[\text{-CCA}]$)**.** *Let* Π *be an IB-KEM. If* Π *is* OW*-secure, then* $\#(\Pi)$ *is* IND*-secure. Analogously, if* Π *is* OW-CCA*-secure, then* $\#(\Pi)$ *is* IND-CCA*-secure.*

Proof. Let $\mathcal{A}^{\#} = (\mathcal{A}_1^{\#}, \mathcal{A}_2^{\#})$ be an IND[-CCA] adversary against $\#(\Pi)$. We can construct an OW[-CCA] adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against Π as follows: $\mathcal{A}_1$ executes $\mathcal{A}_1^{\#}$, recording and answering all queries to random oracle $H^{\#}$, and relaying all oracle queries to $\mathcal{O}_X$, $\mathcal{O}_D$ (in the CCA case), and $\mathcal{O}_{\mathcal{H}}(H, \cdot)$, for $H \neq H^{\#}$, to its own challenger. $\mathcal{A}_1$ outputs the same (id^*, st) as is output by $\mathcal{A}_1^{\#}$. On receiving challenge ciphertext c^*, $\mathcal{A}_2$ picks a random key K^{**} and runs $\mathcal{A}_2^{\#}$ on input (st, c^*, K^{**}), recording and answering all queries to $H^{\#}$ as $\mathcal{A}_1$ did. Eventually $\mathcal{A}_2^{\#}$ stops. $\mathcal{A}_2$ ignores $\mathcal{A}_2^{\#}$'s output and returns a randomly chosen entry from $\mathtt{HLi}$ as its guess for the key corresponding to c^*, where $\mathtt{HLi}$ denotes the list of all answers given to $H^{\#}$ queries made by $\mathcal{A}_1^{\#}$ and $\mathcal{A}_2^{\#}$.

Let K^* be the (real) key corresponding to c^*. Let E be the event that K^* is queried to the $H^{\#}$ oracle by $\mathcal{A}^{\#}$. We have $\Pr\left[\mathrm{Expt}^{\mathsf{IND}[\text{-CCA}],0}_{\#(\Pi),\mathcal{A}^{\#}}(k) = \text{WIN} \middle| \neg E\right] = \Pr\left[\mathrm{Expt}^{\mathsf{IND}[\text{-CCA}],1}_{\#(\Pi),\mathcal{A}^{\#}}(k) = \text{WIN} \middle| \neg E\right]$, and thus $\Pr[E] \geq \mathrm{Advt}^{\mathsf{IND}[\text{-CCA}]}_{\#(\Pi),\mathcal{A}^{\#}}(k)$ (cf. Lemma 17). If E occurs, then K^* is contained in $\mathtt{HLi}$ and $\mathcal{A}_2$ returns it with probability (at least) $1/q$, where $q = |\mathtt{HLi}|$ is polynomially bounded. Hence

$$\begin{aligned}\mathrm{Succ}^{\mathsf{OW}[\text{-CCA}]}_{\Pi,\mathcal{A}}(k) &\geq \Pr\left[\mathrm{Expt}^{\mathsf{OW}[\text{-CCA}]}_{\Pi,\mathcal{A}}(k) = \text{WIN} \middle| E\right] \Pr[E] \\ &\geq \mathrm{Advt}^{\mathsf{IND}[\text{-CCA}]}_{\#(\Pi),\mathcal{A}^{\#}}(k)/q \ .\end{aligned}$$

This proves the lemma. □

Lemma 13 (# : PA $\mapsto$ PA)**.** *Let* Π *be an IB-KEM. If* Π *is plaintext-aware, then* $\#(\Pi)$ *is plaintext-aware.*

Proof. Let $\mathcal{K}$ be a plaintext extractor for Π and let $\mathcal{A}^{\#}$ be a PA adversary (ciphertext creator) for $\#(\Pi)$. Observe that, syntactically, it is possible to run $\mathcal{K}$ on transcripts from $\mathcal{A}^{\#}$. Let $(id^*, c^*, mpk, \mathsf{HList}, \mathsf{XList}, \mathsf{EList})$ be a transcript of $\mathcal{A}^{\#}$ and K be the key output by $\mathcal{K}$ on that input. Since $\mathcal{K}$ is a plaintext extractor for Π, we have that $K = \mathsf{Decap}^{\mathcal{H}}_{\mathsf{Extract}(msk,id^*)}(c^*)$ with high probability and hence

$$\begin{aligned}\mathsf{Decap}^{\#,\mathcal{H}'}_{\mathsf{Extract}^{\#}(msk,id^*)}(c^*) &= H^{\#}(\mathsf{Decap}^{\mathcal{H}}_{\mathsf{Extract}(msk,id^*)}(c^*)) \\ &= H^{\#}(K) \ .\end{aligned}$$

That is, the plaintext extractor $\mathcal{K}^{\#} = H^{\#} \circ \mathcal{K}$, obtained by combining $\mathcal{K}$ with $H^{\#}$, is a valid extractor for $\#(\Pi)$. In particular, $\mathcal{K}^{\#}$ has the same success probability as $\mathcal{K}$. □

Observe that, in the above proof of Lemma 13, we didn't exploit the fact that $H^{\#}$ is a hash function (or even a random oracle). Indeed, the proof would have worked as well if the construction in Definition 9 had applied on the key any other kind of deterministic function (for example, the truncation of a fixed number of bits). However, the use of a random oracle in the # construction is an essential prerequisite of the proof of Lemma 12.

We are now ready to state the central result of this paper: by combining transformation # with transformation $\mathcal{F}$, we can convert any OW-secure IB-KEM into an IB-KEM that is simultaneously plaintext-aware and IND-CCA-secure.

Theorem 3 (Main result: $\# \circ \mathcal{F}$: OW $\mapsto$ PA $\wedge$ IND-CCA)**.** *Let* Π *be an IB-KEM. If* Π *is* OW*-secure, then* $\#(\mathcal{F}(\Pi))$ *is plaintext-aware and* IND-CCA*-secure.*

Proof. By Corollary 2, $\mathcal{F}(\Pi)$ is PA and OW-CCA-secure. By Lemma 13, $\#(\mathcal{F}(\Pi))$ is PA and by Lemma 12 it is further IND-CCA-secure. □

This overall generic construction is powerful yet extremely efficient: encapsulation has just two additional random oracle queries, and decapsulation requires two random oracle queries and a partial encapsulation. We do not require any additional randomness or public parameters. In Section 7.1 we show how to further optimize the overall transformation.

7 Plaintext-aware IB-KEMs in practice: applications and optimizations

We apply our generic transformations from Sections 5 and 6 to two popular IB-KEMs[5], namely to BF-IB-KEM by Boneh and Franklin [BF01], and to SK-IB-KEM by Sakai and Kasahara [SK03]. Both schemes are defined in the pairing-based setting and proven (OW-)secure in the random oracle model. In order to obtain plaintext-aware IB-KEMs of maximum efficiency, we additionally develop a simple yet general and highly effective optimization of our transformation $\mathcal{F}$ from Definition 8.

We start by providing background on the cryptographic setting in which BF-IB-KEM and SK-IB-KEM are defined. Let $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$, and $G_T = \langle g_T \rangle$ be cyclic groups of prime order q. Let $e : G_1 \times G_2 \to G_T$ be an efficient bilinear map (pairing) such that $a, b \in \mathbb{Z}_q \Rightarrow e({g_1}^a, {g_2}^b) = {g_T}^{ab}$. Moreover, let $\psi : G_2 \to G_1$ be an efficient homomorphism such that $\psi(g_2) = g_1$. This setting, for which efficient instantiations are known [AKL+10, BSSC05, GPS08], is usually called a Type II setting [GPS08], and hardness of BDH and t-BDHI problems [CCMLS06] are widely assumed:

[5] More precisely, the schemes presented in [BF01, SK03] are not IB-KEMs, but rather IBE schemes. In this paper we consider the IB-KEMs that underlie their (hybrid) constructions.

– BDH: Given ${g_2}^x, {g_2}^y, {g_2}^z$ for $x, y, z \xleftarrow{\$} \mathbb{Z}_q$, it is infeasible to compute ${g_T}^{xyz}$.
– t-BDHI: Given ${g_2}^x, \ldots, {g_2}^{x^t}$ for $x \xleftarrow{\$} \mathbb{Z}_q$, it is infeasible to compute ${g_T}^{1/x}$.

7.1 Improving efficiency of transformation $\mathcal{F}$

Although the $\mathcal{F}$ transformation from Section 5 is already reasonably efficient, we identify a way to further optimize its performance in practice. For a motivation of our optimization consider the schemes BF-IB-KEM and SK-IB-KEM, as specified in Definitions 11 and 13, respectively. In both schemes we have $\mathtt{KeySp}(k) = G_T$. This set is a group, i.e. the corresponding requirement on $\mathtt{KeySp}(k)$ from Definition 8 is naturally fulfilled and $\mathcal{F}$ transformation can be directly applied to both IB-KEMs. However, sampling randomness r uniformly from G_T, as needed in Encap^* algorithm (cf. Figure 10), is a relatively costly operation. Indeed, in practice it will require an additional exponentiation: pick random exponent $h \xleftarrow{\$} \mathbb{Z}_q$ and compute r via $r \leftarrow {g_T}^h$. We propose a general optimization to the $\mathcal{F}$ transformation which does not require sampling from $\mathtt{KeySp}(k)$, and potentially notably increases obtained scheme's performance.

To derive the new conversion, consider Encap^* algorithm from $\mathcal{F}$ transformation, but replace line (b) by $(c_1, \overline{K}) \leftarrow \mathsf{Encap}_{mpk}(id; r_1)$ followed by $K \leftarrow H'(\overline{K})$, where $H' : \mathtt{KeySp}(k) \to \{0,1\}^k$ is an auxiliary hash function. The range $\{0,1\}^k$ of H' forms an (Abelian) group with respect to the bitwise XOR operation, so that we can apply the $\mathcal{F}$ and $\#$ transformations using this new key space $\mathtt{KeySp}(k) = \{0,1\}^k$. Observe that sampling from $\{0,1\}^k$ is a highly efficient operation, and that binary representation of its elements is (a) computationally trivial, and (b) usually more compact than the representation of elements of G_T, i.e. IB-KEMs obtained through the modified transformation have shorter ciphertexts.

A precise specification of the adapted $\# \circ \mathcal{F}$ transformation is given in Definition 10 and Figure 13. For reasons of efficiency, we conflate $H^\# \circ H'$ into a single hash function $H^\#$ (see lines (f) and (g) of Encap^+ and Decap^+ algorithms, respectively) and assume that $H^\#$ maps to $\{0,1\}^k$. As a further optimization, as H' and $H^\#$ are evaluated solely on $\overline{K}$ (respectively, on $\hat{K}$), one could compute $H'(\overline{K}), H^\#(\overline{K})$ via a single call to a double-length hash function, i.e. $H' \,\|\, H^\# : \mathtt{KeySp}(k) \to \{0,1\}^k \times \{0,1\}^k$. This trick reduces the total number of required hash functions to just two.

Definition 10 (Optimized transformation $\# \circ \mathcal{F}$)**.** *Let* $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encap}, \mathsf{Decap})$ *be an IB-KEM with random oracles* $\mathcal{H}$ *and associated spaces* $\mathtt{IDSp}$, $\mathtt{CoinSp}$, $\mathtt{CipherSp}$, *and* $\mathtt{KeySp}$. *Let* $H : \mathtt{IDSp} \times \{0,1\}^k \to \mathtt{CoinSp}$, $H', H^\# : \mathtt{KeySp} \to \{0,1\}^k$ *be new hash functions (independent of* $\mathcal{H}$*), modeled as a random oracles in the security analysis. Then IB-KEM* $\Pi^+ = (\mathsf{Setup}^+, \mathsf{Extract}^+, \mathsf{Encap}^+, \mathsf{Decap}^+)$ *is specified in Figure 13, with random oracles* $\mathcal{H}' = \mathcal{H} \cup \{H, H', H^\#\}$ *and associated spaces* $\mathtt{IDSp}^+ = \mathtt{IDSp}$, $\mathtt{CoinSp}^+ = \{0,1\}^k$, $\mathtt{CipherSp}^+ = \mathtt{CipherSp} \times \{0,1\}^k$, *and* $\mathtt{KeySp}^+ = \{0,1\}^k$.

Technically speaking, the proposed transformation involves twice the application of the $\#$ transformation from Section 6 to the respective IB-KEM. Lemmas 12 and 2 show that its first application preserves OW security (it even yields an IND-secure IB-KEM), and hence, together with Theorem 3, establish soundness of the overall transformation[6]:

Theorem 4 (Def. 10 : $\mathsf{OW} \mapsto \mathsf{PA} \wedge \mathsf{IND\text{-}CCA}$)**.** *Let* Π *be an IB-KEM and* Π^+ *its conversion. If* Π *is* OW*-secure, then IB-KEM* Π^+ *is plaintext-aware and* IND-CCA*-secure.*

[6]On first sight it seems that by applying the H' hash function to intermediate key $\overline{K}$ we lose a factor of $q_{H'}$ in the security reduction. However, Remark 9 shows that this is not true: as $H'(\overline{K})$ is an IND-secure key, we correspondingly get tighter security for the $\mathcal{F}$ construction.

$\mathsf{Setup}^+(1^k)$:
(a) Return $\mathsf{Setup}(1^k)$.

$\mathsf{Encap}^+_{mpk}(id; r)$ for $r \in \{0,1\}^k$:
(a) $r_1 \leftarrow H(id, r)$
(b) $(c_1, \overline{K}) \leftarrow \mathsf{Encap}_{mpk}(id; r_1)$
(c) $c_2 \leftarrow r \oplus H'(\overline{K})$
(d) $c \leftarrow (c_1, c_2)$
(e) $K \leftarrow H^{\#}(\overline{K})$
(f) Return (c, K).

$\mathsf{Extract}^+(msk, id)$:
(a) Return $\mathsf{Extract}(msk, id)$.

$\mathsf{Decap}^+_{sk}(c)$:
(a) Parse $(c_1, c_2) \leftarrow c$.
(b) $\hat{K} \leftarrow \mathsf{Decap}_{sk}(c_1)$
(c) If $\hat{K} = \perp$ return $\perp$.
(d) $\hat{r}_1 \leftarrow H(id, c_2 \oplus H'(\hat{K}))$
(e) If $(c_1, \cdot) \neq \mathsf{Encap}_{mpk}(id; \hat{r}_1)$ return $\perp$.
(f) $K \leftarrow H^{\#}(\hat{K})$
(g) Return K.

Figure 13: Optimized combination of $\mathcal{F}$ and $\#$ transformations from Figures 10 and 12. Queries to hash functions other than $H, H', H^{\#}$ are relayed without modification.

$\mathsf{Setup}(1^k)$:
(a) $msk \xleftarrow{\$} \mathbb{Z}_q$
(b) $mpk \leftarrow {g_2}^{msk}$
(c) Return (mpk, msk).

$\mathsf{Extract}(msk, id)$:
(a) $sk \leftarrow H_1(id)^{msk}$
(b) Return sk.

$\mathsf{Encap}_{mpk}(id; r)$ for $r \in \mathbb{Z}_q$:
(a) $c \leftarrow {g_2}^r$
(b) $K \leftarrow e(H_1(id), mpk)^r$
(c) Return (c, K).

$\mathsf{Decap}_{sk}(c)$:
(a) $K \leftarrow e(sk, c)$
(b) Return K.

$\mathsf{Setup}^+(1^k)$:
(a) Return $\mathsf{Setup}(1^k)$.

$\mathsf{Extract}^+(msk, id)$:
(a) Return $\mathsf{Extract}(msk, id)$.

$\mathsf{Encap}^+_{mpk}(id; r)$ for $r \in \{0,1\}^k$:
(a) $r_1 \leftarrow H(id, r)$
(b) $\overline{K} \leftarrow e(H_1(id), mpk)^{r_1}$
(c) $c \leftarrow ({g_2}^{r_1}, r \oplus H'(\overline{K}))$
(d) $K \leftarrow H^{\#}(\overline{K})$
(e) Return (c, K).

$\mathsf{Decap}^+_{sk}(c)$:
(a) Parse $(c_1, c_2) \leftarrow c$.
(b) $\hat{K} \leftarrow e(sk, c_1)$
(c) $\hat{r}_1 \leftarrow H(id, c_2 \oplus H'(\hat{K}))$
(d) If $c_1 \neq {g_2}^{\hat{r}_1}$ return $\perp$.
(e) $K \leftarrow H^{\#}(\hat{K})$
(f) Return K.

Figure 14: Specification of Π = BF-IB-KEM and Π^+ = BF-IB-KEM$^+$. Observe that Π^+ is constructed from Π following the stencil from Figure 13.

7.2 Plaintext-aware and IND-CCA-secure BF-IB-KEM

BF-IB-KEM is the underlying primitive to the Boneh-Franklin IBE scheme [BF01]. More precisely, the BasicIdent scheme from [BF01] is the canonical transformation of BF-IB-KEM into an IBE scheme, using a KEM/DEM construction where one-time pad encryption is used for the DEM part. BF-IB-KEM is OW-secure under Bilinear Diffie-Hellman (BDH) assumption [CCMLS06].

Definition 11 (BF-IB-KEM)**.** *The* Boneh-Franklin identity-based key encapsulation mechanism *is specified in Figure 14 (left part), where* $H_1 : \{0,1\}^* \to G_1$ *denotes a hash function. We have* $\mathtt{IDSp}(k) = \{0,1\}^*$, $\mathtt{CoinSp}(k) = \mathbb{Z}_q$, $\mathtt{CipherSp}(k) = G_2$, *and* $\mathtt{KeySp}(k) = G_T$.

Correctness of BF-IB-KEM follows from equality

$$e(H_1(id), mpk)^r = e(H_1(id)^{msk}, {g_2}^r) = e(sk, c).$$

Note that encapsulation requires one pairing evaluation and two exponentiations, while decapsulation takes a single pairing operation. By applying to BF-IB-KEM the transformation from Definition 10 we obtain BF-IB-KEM$^+$:

Definition 12 (BF-IB-KEM$^+$)**.** BF-IB-KEM$^+$ *is specified in Figure 14 (right part), where* $H_1 : \{0,1\}^* \to G_1$, $H : \{0,1\}^* \times \{0,1\}^k \to \mathbb{Z}_q$, *and* $H', H^{\#} : G_T \to \{0,1\}^k$ *are hash functions,*

$\mathsf{Setup}(1^k)$:
(a) $msk \xleftarrow{\$} \mathbb{Z}_q$
(b) $mpk \leftarrow {g_1}^{msk}$
(c) Return (mpk, msk).

$\mathsf{Extract}(msk, id)$:
(a) $sk \leftarrow {g_2}^{1/(msk+H_1(id))}$
(b) Return sk.

$\mathsf{Encap}_{mpk}(id; r)$ for $r \in \mathbb{Z}_q$:
(a) $c \leftarrow \big(mpk \cdot {g_1}^{H_1(id)}\big)^r$
(b) $K \leftarrow {g_T}^r$
(c) Return (c, K).

$\mathsf{Decap}_{sk}(c)$:
(a) $K \leftarrow e(c, sk)$
(b) Return K.

$\mathsf{Setup}^+(1^k)$:
(a) Return $\mathsf{Setup}(1^k)$.

$\mathsf{Extract}^+(msk, id)$:
(a) Return $\mathsf{Extract}(msk, id)$.

$\mathsf{Encap}^+_{mpk}(id; r)$ for $r \in \{0,1\}^k$:
(a) $r_1 \leftarrow H(id, r)$
(b) $\overline{K} \leftarrow {g_T}^{r_1}$
(c) $c_1 \leftarrow \big(mpk \cdot {g_1}^{H_1(id)}\big)^{r_1}$
(d) $c_2 \leftarrow r \oplus H'(\overline{K})$
(e) $c \leftarrow (c_1, c_2)$
(f) $K \leftarrow H^{\#}(\overline{K})$
(g) Return (c, K).

$\mathsf{Decap}^+_{sk}(c)$:
(a) Parse $(c_1, c_2) \leftarrow c$.
(b) $\hat{K} \leftarrow e(c_1, sk)$
(c) $\hat{r}_1 \leftarrow H(id, c_2 \oplus H'(\hat{K}))$
(d) If $c_1 \neq \big(mpk \cdot {g_1}^{H_1(id)}\big)^{\hat{r}_1}$ return $\perp$.
(e) $K \leftarrow H^{\#}(\hat{K})$
(f) Return K.

Figure 15: Specification of Π = SK-IB-KEM and Π^+ = SK-IB-KEM$^+$. Observe that Π^+ is constructed from Π following the stencil from Figure 13.

and $\mathtt{IDSp}^+(k) = \{0,1\}^*$, $\mathtt{CoinSp}^+(k) = \{0,1\}^k$, $\mathtt{CipherSp}^+(k) = G_2 \times \{0,1\}^k$, *and* $\mathtt{KeySp}^+(k) = \{0,1\}^k$.

The security properties of BF-IB-KEM$^+$ are implied by Theorem 4:

Theorem 5. *BF-IB-KEM$^+$ is plaintext-aware* (PA) *and* IND-CCA*-secure, under the BDH assumption in the random oracle model.*

Our transformation almost preserves the efficiency of the original scheme: the only substantial difference between BF-IB-KEM and BF-IB-KEM$^+$ is the additional exponentiation in line (d) of Decap^+, which can be considered marginal in comparison to the pairing evaluation in line (b). Observe that the length of ciphertexts is increased by only k bits and that all newly introduced hash functions $H, H', H^{\#}$ are straight-forward to instantiate.

7.3 Plaintext-aware and IND-CCA-secure SK-IB-KEM

The second IB-KEM that we analyze in this section was proposed by Sakai and Kasahara in [SK03]. Its main advantage over BF-IB-KEM is its efficiency: encapsulation operations do not require any pairing evaluation. Moreover, there is no need to efficiently hash into G_1 or G_2, but only into $\mathbb{Z}_q$.

Definition 13 (SK-IB-KEM)**.** *The* Sakai-Kasahara identity-based key encapsulation mechanism *is specified in Figure 15 (left part), where $H_1 : \{0,1\}^* \to \mathbb{Z}_q$ denotes a hash function. We have $\mathtt{IDSp}(k) = \{0,1\}^*$, $\mathtt{CoinSp}(k) = \mathbb{Z}_q$, $\mathtt{CipherSp}(k) = G_1$, and $\mathtt{KeySp}(k) = G_T$.*

SK-IB-KEM is proven OW-secure in [CC05, CCMLS06], assuming hardness of t-BDHI problem. Observe that secret keys cannot be extracted if $H_1(id) = -msk$, but this occurs only with negligible probability. In all other cases correctness is established by

$$\begin{aligned} e(c, sk) &= e\big(\big(mpk \cdot {g_1}^{H_1(id)}\big)^r, {g_2}^{1/(msk+H_1(id))}\big) \\ &= e\big({g_1}^{msk+H_1(id)}, {g_2}^{1/(msk+H_1(id))}\big)^r \\ &= {g_T}^r \ . \end{aligned}$$

Encapsulation requires two exponentiations but no pairing operation. Efficiency of decapsulation is the same as in BF-IB-KEM, namely one pairing evaluation. We apply the transformation from Definition 10 to SK-IB-KEM and obtain SK-IB-KEM$^+$:

Definition 14 (SK-IB-KEM$^+$)**.** *SK-IB-KEM$^+$ is specified in Figure 15 (right part), where $H_1 : \{0,1\}^* \to \mathbb{Z}_q$, $H : \{0,1\}^* \times \{0,1\}^k \to \mathbb{Z}_q$ and $H', H^\# : G_T \to \{0,1\}^k$ are hash functions, and $\mathtt{IDSp}^+(k) = \{0,1\}^*$, $\mathtt{CoinSp}^+(k) = \{0,1\}^k$, $\mathtt{CipherSp}^+(k) = G_1 \times \{0,1\}^k$, and $\mathtt{KeySp}^+(k) = \{0,1\}^k$.*

We let Theorem 4 establish the security properties of SK-IB-KEM$^+$:

Theorem 6. *SK-IB-KEM$^+$ is plaintext-aware* (PA) *and* IND-CCA*-secure, under the t-BDHI assumption in the random oracle model.*

SK-IB-KEM$^+$ is slightly less efficient than SK-IB-KEM. This difference is caused by two additional exponentiations in Decap^+, in line (d). Note that one of them, namely ${g_1}^{H_1(id)}$, can be precomputed. Observe that the length of ciphertexts is increased by only k bits and that all newly introduced hash functions $H, H', H^\#$ are straight-forward to instantiate.

8 Conclusion

We have defined and analyzed the notion of plaintext awareness for key encapsulation mechanisms in the identity-based setting and the public key setting in the random oracle model. We have explored the relationships between plaintext awareness and other security notions, noting that a scheme that is both PA and OW is also OW-CCA. We have also introduced a generic transformation $\mathcal{F}$, somewhat related to the Fujisaki-Okamoto transformation for public key encryption, that can be used to construct a plaintext-aware KEM from an OW-secure KEM. Notably, our $\mathcal{F}$ construction uses a weaker security notion which we call computational uniformity (and which is in fact implied by OW) than the distinct notion of uniformity required by Fujisaki-Okamoto; this security notion may be of independent interest. We further propose a simple hash-based construction # that, combined with our $\mathcal{F}$ construction, converts a weak (OW-secure) KEM into a strong (plaintext-aware and IND-CCA-secure) KEM. We apply this construction to several existing IB-KEMs, resulting in the first provably plaintext-aware secure IB-KEMs. This transformation can also be applied in the public key KEM setting.

Our techniques and results apply in the random oracle model. Existing proof strategies for plaintext awareness in the standard model, such as the simulatability approach of Birkett and Dent [BD11], do not seem to apply current standard model IB-KEMs, and we identify the development of proof strategies for plaintext awareness of identity-based constructions in the standard model as open question.

References

[AGKS05] Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In Ronald Cramer, editor, *EUROCRYPT 2005*, *LNCS*, volume 3494, pp. 128–146. Springer, 2005. DOI:10.1007/b136415.

[AKL+10] D. F. Aranha, K. Karabina, P. Longa, C. H. Gebotys, and J. López. Faster Explicit Formulas for Computing Pairings over Ordinary Curves. In K. Paterson, editor, *30th International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2011)*, *LNCS*, volume 6632, pp. 48–68. Springer, 2010.

[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.

[BD08] James Birkett and Alexander W. Dent. Relations Among Notions of Plaintext Awareness. In Ronald Cramer, editor, *PKC 2008*, *LNCS*, volume 4939, pp. 47–64. Springer, 2008.

[BD11] James Birkett and Alexander W. Dent. Security Models and Proof Strategies for Plaintext-Aware Encryption. Manuscript, http://www.isg.rhul.ac.uk/~alex/papers/plaintext_journal.pdf, 2011.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *CRYPTO '98*, *LNCS*, volume 1462, pp. 26–45. Springer-Verlag, 1998. DOI:10.1007/BFb0055718.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, *LNCS*, volume 2139, pp. 213–229. Springer, 2001. DOI:10.1007/3-540-44647-8_13.

[BFMLS08] Kamel Bentahar, Pooya Farshim, John Malone-Lee, and Nigel P. Smart. Generic Constructions of Identity-Based and Certificateless KEMs. *J. Cryptology*, **21**(2):178–199, 2008.

[BK05] Dan Boneh and Jonathan Katz. Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In Alfred J. Menezes, editor, *CT-RSA 2005*, *LNCS*, volume 3376, pp. 87–103. Springer, 2005.

[BMW05] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct Chosen Ciphertext Security from Identity-Based Techniques. In *ACM CCS 2005*, pp. 320–329. ACM, 2005.

[BP04] Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In Pil Joong Lee, editor, *ASIACRYPT 2004*, *LNCS*, volume 3329, pp. 37–52. Springer, 2004. DOI:10.1007/978-3-540-30539-2_4. Full version available as http://cseweb.ucsd.edu/users/mihir/papers/pa.pdf.

[BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT '94*, *LNCS*, volume 950, pp. 92–111. Springer-Verlag, 1994. DOI:10.1007/BFb0053428. Full version available as http://www-cse.ucsd.edu/~mihir/papers/oaep.html.

[BSSC05] I. Blake, G. Seroussi, N. Smart, and J. W. S. Cassels. *Advances in Elliptic Curve Cryptography (London Mathematical Society Lecture Note Series)*. Cambridge University Press, New York, NY, USA, 2005.

[CC05] Liqun Chen and Zhaohui Cheng. Security proof of Sakai-Kasahara's identity-based encryption scheme. In Nigel P. Smart, editor, *Cryptography and Coding – 10th IMA International Conference*, *LNCS*, volume 3796, pp. 442–459. Springer, 2005. DOI:10.1007/11586821.

[CCMLS06] Liqun Chen, Zhaohui Cheng, John Malone-Lee, and Nigel P. Smart. Efficient ID-KEM based on the Sakai-Kasahara key construction. *Information Security, IEE Proceedings*, **153**(1):19–26, March 2006. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1613725&isnumber=33872.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, *LNCS*, volume 3027, pp. 207–222. Springer, 2004.

[CS02] Ronald Cramer and Victor Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In Lars Knudsen, editor, *EUROCRYPT 2002*, *LNCS*, volume 2332, pp. 45–64. Springer, 2002.

[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**(1):167–226, 2003.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable Cryptography. *SIAM Journal on Computing*, **30**(2):391–437, 2000.

[Den03] Alexander W. Dent. A Designer's Guide to KEMs. In Kenneth G. Paterson, editor, *Cryptography and Coding*, *LNCS*, volume 2898, pp. 133–151. Springer, 2003. Updated version available at http://eprint.iacr.org/2002/174.

[Den06] Alexander W. Dent. The Cramer-Shoup Encryption Scheme Is Plaintext Aware in the Standard Model. In Serge Vaudenay, editor, *EUROCRYPT 2006*, *LNCS*, volume 4004, pp. 289–307. Springer, 2006.

[DGK06] Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk. Deniable authentication and key exchange. In Rebecca Wright, Sabrina De Capitani de Vimercati, and Vitaly Shmatikov, editors, *ACM CCS 2006*, pp. 400–409. ACM, 2006. DOI:10.1145/1180405.1180454. Full version available as http://eprint.iacr.org/2006/280.

[DGKS10] Yvo Desmedt, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. A New and Improved Paradigm for Hybrid Encryption Secure Against Chosen-Ciphertext Attack. *J. Cryptology*, **23**(1):91–120, 2010.

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO*, *Lecture Notes in Computer Science*, volume 196, pp. 10–18. Springer, 1984.

[ES02] Edith Elkind and Amit Sahai. A Unified Methodology For Constructing Public-Key Encryption Schemes Secure Against Adaptive Chosen-Ciphertext Attack. Cryptology ePrint Archive, Report 2002/042, 2002. http://eprint.iacr.org/.

[FO99a] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In Hideki Imai and Yuliang Zheng, editors, *PKC 1999*, *LNCS*, volume 1560, pp. 53–68. Springer, 1999. DOI:10.1007/3-540-49162-7_5.

[FO99b] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael Wiener, editor, *CRYPTO '99*, *LNCS*, volume 1666, pp. 537–554. Springer, 1999. DOI:10.1007/3-540-48405-1_34.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, **156**(16):3113–3121, September 2008. DOI:10.1016/j.dam.2007.12.010.

[HHK10] Javier Herranz, Dennis Hofheinz, and Eike Kiltz. Some (In)Sufficient Conditions for Secure Hybrid Encryption. *Information and Computation*, **208**(11):1243–1257, 2010.

[HLM03] Jonathan Herzog, Moses Liskov, and Silvio Micali. Plaintext awareness via key registration. In Dan Boneh, editor, *CRYPTO 2003*, *LNCS*, volume 2729, pp. 548–564. Springer, 2003. DOI:10.1007/978-3-540-45146-4_32.

[JW10] Shaoquan Jiang and Huaxiong Wang. Plaintext-awareness of hybrid encryption. In Josef Pieprzyk, editor, *CT-RSA 2010*, *LNCS*, volume 5985, pp. 57–72. Springer, 2010. DOI:10.1007/978-3-642-11925-5_5. Full version available at http://sites.google.com/site/shaoquan0825/DHIES-8.pdf.

[KD04] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matt Franklin, editor, *CRYPTO*, *LNCS*, volume 3152, pp. 426–442. Springer, 2004.

[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 Shai Halevi and Tal Rabin, editors, *TCC 2006*, *LNCS*, volume 3876, pp. 581–600. Springer, 2006.

[KYH+06] Takashi Kitagawa, Peng Yang, Goichiro Hanaoka, Rui Zhang, Hajime Watanabe, Kanta Matsuura, and Hideki Imai. Generic transforms to acquire CCA-security for identity based encryption: The cases of FOpkc and REACT. In Lynn Margaret Batten and Reihaneh Safavi-Naini, editors, *ACISP 2006*, *LNCS*, volume 4058, pp. 348–359. Springer, 2006. DOI:10.1007/11780656_29.

[OP01] Tatsuaki Okamoto and David Pointcheval. REACT: Rapid Enhanced-Security Asymmetric Cryptosystem Transform. In David Naccache, editor, *CT-RSA 2001*, *LNCS*, volume 2020, pp. 159–175. Springer, 2001.

[Sah99] Amit Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *FOCS 1999*, pp. 543–553. IEEE, 1999.

[Sho00] Victor Shoup. Using Hash Functions as a Hedge against Chosen Ciphertext Attack. In Bart Preneel, editor, *EUROCRYPT 2000*, *LNCS*, volume 1807, pp. 275–288. Springer, 2000.

[SK03] Ryuichi Sakai and Masao Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054, 2003. http://eprint.iacr.org/.

[TO06] Isamu Teranishi and Wakaha Ogata. Relationship between standard model plaintext awareness and message hiding. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, *LNCS*, volume 4284, pp. 226–240. Springer, 2006. DOI:10.1007/11935230_15.

[TO08] Isamu Teranishi and Wakaha Ogata. Cramer-shoup satisfies a stronger plaintext awareness under a weaker assumption. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *SCN 2008*, *LNCS*, volume 5229, pp. 109–125. Springer, 2008. DOI:10.1007/978-3-540-85855-3_8.

[YKH+06] Peng Yang, Takashi Kitagawa, Goichiro Hanaoka, Rui Zhang, Kanta Matsuura, and Hideki Imai. Applying Fujisaki-Okamoto to identity-based encryption. In Marc P.C. Fossorier, Hideki Imai, Shu Lin, and Alain Poli, editors, *AAECC-16 2006*, *LNCS*, volume 3857, pp. 183–192, 2006. DOI:10.1007/11617983_18.

A Extending our results to PK-KEMs

We explore the applicability of our IB-KEM-related results from Sections 3–7 to the setting of public key KEMs. A PK-KEM is generally given by a set of three algorithms: KeyGen, Encap, and Decap. We observe that IB-KEM-related definitions of syntax, secrecy (OW[-CCA] and IND[-CCA]), computational uniformity (cU), and plaintext awareness (PA) can readily be adapted to the PK-KEM setting by replacing in the security experiments $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}(1^k)$ by $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k)$, and by further leaving out identities and availability of $\mathcal{O}_X$ oracles.

We prove in Section A.1 that for these adapted definitions the implication $\mathsf{OW} \Rightarrow \mathsf{cU}$ still holds (cf. Corollary 1 in Section 4). Under this premise, a close inspection of our lemmas and proofs from Sections 3 and 5–7 shows that their statements remain valid in the PK-KEM world. In particular, we have correspondence for Theorem 1 ($\mathsf{OW} \wedge \mathsf{PA} \Rightarrow \mathsf{OW\text{-}CCA}$), Corollary 2 ($\mathcal{F} : \mathsf{OW} \mapsto \mathsf{PA} \wedge \mathsf{OW\text{-}CCA}$) and Theorem 3 ($\# \circ \mathcal{F} : \mathsf{OW} \mapsto \mathsf{PA} \wedge \mathsf{IND\text{-}CCA}$), with slightly adapted $\mathcal{F}$ and $\#$ transformations. We refrain here from giving corresponding proofs as it would suffice to marginally adjust those from named sections to fit the public key setting.

In Section A.2 we adapt the optimized $\# \circ \mathcal{F}$ transformation from Section 7 to the PK-KEM setting, and, for concreteness, apply this transformation to ElGamal-KEM, in Section A.3.

A.1 Computational uniformity of PK-KEMs and its relation to one-wayness

We define the notion of computational uniformity (cU) for PK-KEMs and show that it is implied by one-way security. As we did for IB-KEMs in Section 4, we prove this implication via an intermediate notion (CU), i.e. we prove $\mathsf{OW} \Rightarrow \mathsf{CU}$ and $\mathsf{CU} \Rightarrow \mathsf{cU}$ separately. Observe that CU is an information-theoretic notion, while the intermediate notion cCU in the identity-based setting was computational (cf. Definition 7). This difference comes from the fact that cCU is based on an adversary $\mathcal{B}$ that outputs an identity for which collisions among ciphertexts become noticeable. In the public key setting, however, this step becomes obsolete and we are only interested in collisions among ciphertexts generated for one specific (randomly generated) public key. As the notion of OW security is readily derived from Definition 2 by "stripping off" all parameters and oracles related to the identity-based setting, we start by defining the notion of collision uniformity (CU).

Definition 15 (Collision uniformity (CU))**.** *Let* Π *be a PK-KEM with associated spaces* $\mathtt{CoinSp}$, $\mathtt{CipherSp}$, *and* $\mathtt{KeySp}$. *For* $k \in \mathbb{N}$, $\mathcal{H} \in \mathsf{Hash}(k)$, *and* $(pk, sk) \in \mathsf{KeyGen}^{\mathcal{H}}(1^k)$ *define*

$$\Gamma^{\mathcal{H}}_{pk} = \Pr\left[r_1, r_2 \xleftarrow{\$} \mathtt{CoinSp}(k); \mathsf{Encap}^{\mathcal{H}}_{pk}(r_1) = \mathsf{Encap}^{\mathcal{H}}_{pk}(r_2)\right] \ .$$

$\mathcal{A}^{\mathcal{O}_{\mathcal{H}}}(pk, c^*)$:
(a) $(c', K') \xleftarrow{\$} \mathsf{Encap}^{\mathcal{H}}_{pk}()$
(b) If $c' \neq c^*$ return $\perp$.
(c) Return K'.

Figure 16: Construction of OW adversary $\mathcal{A}$ against Π

We say that Π *is* Γ-collision-uniform *for a given function* $\Gamma : \mathbb{N} \to \mathbb{R}^{\geq 0}$ *if the following probability is negligible in* k*:*

$$\Pr_{\substack{\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k) \\ (pk,sk) \xleftarrow{\$} \mathsf{KeyGen}^{\mathcal{H}}(1^k)}} \left[\Gamma^{\mathcal{H}}_{pk} > \Gamma(k)\right].$$

Π *is called* collision-uniform (CU) *if* Π *is* Γ*-collision-uniform for all non-negligible* Γ*.*

Lemma 14 (OW $\Rightarrow$ CU)**.** *Let* Π *be a PK-KEM. If* Π *is* OW*-secure, then* Π *is collision-uniform.*

Proof. Assume that Π is not collision-uniform, i.e. there exists a non-negligible Γ such that Π is not Γ-collision-uniform. Consider adversary $\mathcal{A}$ against OW of Π from Figure 16.

Denote by E the event that $c' = c^*$ in line (b) of adversary $\mathcal{A}$. Clearly, if E occurs then also $K' = K^*$ by correctness of Π, and hence $\mathrm{Succ}^{\mathsf{OW}}_{\Pi,\mathcal{A}}(k) = \Pr[E]$. On the other hand, inspection of adversary $\mathcal{A}$ and OW experiment reveals that $\Pr[E] = \Gamma^{\mathcal{H}}_{pk}$. Let B denote the event that $\Gamma^{\mathcal{H}}_{pk} > \Gamma(k)$ in the execution of OW experiment. By assumption we have $\Pr[B] > \mathrm{negl}(k)$. We thus obtain

$$\begin{aligned}\mathrm{Succ}^{\mathsf{OW}}_{\Pi,\mathcal{A}}(k) = \Pr[E] &\geq \Pr[E \wedge B] = \Pr[E|B]\Pr[B] \\ &> \Gamma(k) \cdot \Pr[B] > \mathrm{negl}(k)\end{aligned}$$

and conclude that Π is not OW-secure. □

We next adapt the notion of computational uniformity (cU, Definition 6) from the identity-based to the public key setting. Note that it is a (strictly weaker) variant of the uniformity notion for public key encryption schemes, as put forward by Fujisaki and Okamoto in [FO99b].

Definition 16 (Computational uniformity (cU))**.** *Let* Π *be a PK-KEM with associated spaces* $\mathtt{CoinSp}$*,* $\mathtt{CipherSp}$*, and* $\mathtt{KeySp}$*. For* $k \in \mathbb{N}$*,* $\mathcal{H} \in \mathsf{Hash}(k)$*,* $(pk, sk) \in \mathsf{KeyGen}^{\mathcal{H}}(1^k)$*, and* $c \in \mathtt{CipherSp}(k)$ *define*

$$\gamma^{\mathcal{H}}_{pk}(c) = \Pr\left[r \xleftarrow{\$} \mathtt{CoinSp}(k); (c, \cdot) = \mathsf{Encap}^{\mathcal{H}}_{pk}(r)\right].$$

For any algorithm $\mathcal{A}$ *and function* $\gamma : \mathbb{N} \to \mathbb{R}^{\geq 0}$*, consider experiment* $\mathrm{Expt}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}$ *from Figure 17 and define the success probability of* $\mathcal{A}$ *as*

$$\mathrm{Succ}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k) = \Pr\left[\mathrm{Expt}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k) = \textsc{win}\right].$$

PK-KEM Π *is* computationally γ-uniform *if* $\mathrm{Succ}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k)$ *is negligible in* k*, for all adversaries* $\mathcal{A}$*. PK-KEM* Π *is simply called* computationally uniform (cU) *if* Π *is computationally* γ*-uniform for all non-negligible* γ*.*

Analogously to what we show in the identity-based setting (cf. Lemma 7), collision uniformity of PK-KEMs implies computational uniformity:

Lemma 15 (CU $\Rightarrow$ cU)**.** *Let* Π *be a PK-KEM and* $\gamma, \Gamma : \mathbb{N} \to \mathbb{R}^{\geq 0}$ *be functions such that* $\Gamma(k) = \gamma^2(k)$ *for all* k*. If* Π *is* Γ*-collision-uniform, then* Π *is computationally* γ*-uniform. In particular, if* Π *is collision-uniform, then* Π *is computationally uniform.*

$\mathrm{Expt}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k)$:
(a) $\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k)$
(b) $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}^{\mathcal{H}}(1^k)$
(c) $c \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathcal{H}}}(pk)$
– Answer $\mathcal{O}_{\mathcal{H}}$ queries as in Figure 2.
(d) If $\gamma^{\mathcal{H}}_{pk}(c) > \gamma(k)$ return WIN.
(e) Return LOSE.

Figure 17: Experiment for computational uniformity (cU) of PK-KEMs

Proof. For any $k \in \mathbb{N}$, $\mathcal{H} \in \mathsf{Hash}(k)$, $(pk, sk) \in \mathsf{KeyGen}^{\mathcal{H}}(1^k)$, and $c \in \mathtt{CipherSp}(k)$ we have

$$\Gamma^{\mathcal{H}}_{pk} \geq \Pr\Big[r_1, r_2 \xleftarrow{\$} \mathtt{CoinSp}(k); \mathsf{Encap}^{\mathcal{H}}_{pk}(r_1) = (c, \cdot) \\ = \mathsf{Encap}^{\mathcal{H}}_{pk}(r_2)\Big] = \gamma^{\mathcal{H}}_{pk}(c)^2 \ .$$

Now, if an adversary $\mathcal{A}$ against computational γ-uniformity of Π manages to output a ciphertext c such that $\gamma^{\mathcal{H}}_{pk}(c) > \gamma(k)$, then $\Gamma^{\mathcal{H}}_{pk} \geq \gamma^{\mathcal{H}}_{pk}(c)^2 > \gamma^2(k) = \Gamma(k)$ holds as well. Hence we have

$$\mathrm{Succ}^{\mathsf{cU},\gamma}_{\Pi,\mathcal{A}}(k) \leq \Pr_{\substack{\mathcal{H} \xleftarrow{\$} \mathsf{Hash}(k) \\ (pk,sk) \xleftarrow{\$} \mathsf{KeyGen}^{\mathcal{H}}(1^k)}} \Big[\Gamma^{\mathcal{H}}_{pk} > \Gamma(k)\Big] \ .$$

If Π is Γ-collision-uniform then the right term is negligible, i.e. Π is computationally γ-uniform. □

Taken together, Lemmas 14 and 15 imply the following corollary:

Corollary 3 ($\mathsf{OW} \Rightarrow \mathsf{cU}$)**.** *Let Π be a PK-KEM. If Π is OW-secure, then Π is computationally uniform.*

A.2 Obtaining plaintext awareness for PK-KEMs

In Sections 5 and 6 we developed techniques that convert any OW-secure IB-KEM into a plaintext-aware $\mathsf{IND\text{-}CCA}$-secure one. This conversion was further optimized (in regards to efficiency) in Section 7. In the PK-KEM setting, we directly propose an optimized transformation that turns any OW-secure PK-KEM into one that offers plaintext awareness and $\mathsf{IND\text{-}CCA}$ security.

Definition 17 (Optimized transformation for PK-KEMs)**.** *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ be a PK-KEM with random oracles $\mathcal{H}$ and associated spaces $\mathtt{CoinSp}$, $\mathtt{CipherSp}$, and $\mathtt{KeySp}$. Let $H : \{0,1\}^k \to \mathtt{CoinSp}$, $H', H^{\#} : \mathtt{KeySp} \to \{0,1\}^k$ be three hash functions (independent of $\mathcal{H}$), modeled as random oracles in the security analysis. Then PK-KEM $\Pi^+ = (\mathsf{KeyGen}^+, \mathsf{Encap}^+, \mathsf{Decap}^+)$ is specified in Figure 18, with random oracles $\mathcal{H}' = \mathcal{H} \cup \{H, H', H^{\#}\}$ and associated spaces $\mathtt{CoinSp}^+ = \{0,1\}^k$, $\mathtt{CipherSp}^+ = \mathtt{CipherSp} \times \{0,1\}^k$, and $\mathtt{KeySp}^+ = \{0,1\}^k$.*

As, in Figure 18, hash functions H' and $H^{\#}$ are evaluated solely on $\overline{K}$ (respectively, on $\hat{K}$), computing them via a single call to a double-length hash function $H' \parallel H^{\#} : \mathtt{KeySp}(k) \to \{0,1\}^k \times \{0,1\}^k$ is possible and reduces the total number of required hash functions to just two. The security of the transformation is established as in Theorem 4:

Theorem 7 (Def. 17 : $\mathsf{OW} \mapsto \mathsf{PA} \wedge \mathsf{IND\text{-}CCA}$)**.** *Let Π be a PK-KEM and Π^+ its conversion. If Π is OW-secure, then PK-KEM Π^+ is plaintext-aware and $\mathsf{IND\text{-}CCA}$-secure.*

$\mathsf{KeyGen}^+(1^k)$:
(a) Return $\mathsf{KeyGen}(1^k)$.

$\mathsf{Encap}^+_{pk}(r)$ for $r \in \{0,1\}^k$:
(a) $r_1 \leftarrow H(r)$
(b) $(c_1, \overline{K}) \leftarrow \mathsf{Encap}_{pk}(r_1)$
(c) $c_2 \leftarrow r \oplus H'(\overline{K})$
(d) $c \leftarrow (c_1, c_2)$
(e) $K \leftarrow H^{\#}(\overline{K})$
(f) Return (c, K).

$\mathsf{Decap}^+_{sk}(c)$:
(a) Parse $(c_1, c_2) \leftarrow c$.
(b) $\hat{K} \leftarrow \mathsf{Decap}_{sk}(c_1)$
(c) If $\hat{K} = \perp$ return $\perp$.
(d) $\hat{r}_1 \leftarrow H(c_2 \oplus H'(\hat{K}))$
(e) If $(c_1, \cdot) \neq \mathsf{Encap}_{pk}(\hat{r}_1)$ return $\perp$.
(f) $K \leftarrow H^{\#}(\hat{K})$
(g) Return K.

Figure 18: Adaption of the optimized transformation for IB-KEMs from Figure 13 to the PK-KEM setting. Queries to hash functions other than $H, H', H^{\#}$ are relayed without modification.

$\mathsf{KeyGen}(1^k)$:
(a) $sk \xleftarrow{\$} \mathbb{Z}_q$
(b) $pk \leftarrow g^{sk}$
(c) Return (pk, sk).

$\mathsf{Encap}_{pk}(r)$ for $r \in \mathbb{Z}_q$:
(a) $c \leftarrow g^r$
(b) $K \leftarrow (pk)^r$
(c) Return (c, K).

$\mathsf{Decap}_{sk}(c)$:
(a) $K \leftarrow c^{sk}$
(b) Return K.

$\mathsf{KeyGen}^+(1^k)$:
(a) Return $\mathsf{KeyGen}(1^k)$.

$\mathsf{Encap}^+_{pk}(r)$ for $r \in \{0,1\}^k$:
(a) $r_1 \leftarrow H(r)$
(b) $\overline{K} \leftarrow (pk)^{r_1}$
(c) $c_1 \leftarrow g^{r_1}$
(d) $c_2 \leftarrow r \oplus H'(\overline{K})$
(e) $c \leftarrow (c_1, c_2)$
(f) $K \leftarrow H^{\#}(\overline{K})$
(g) Return (c, K).

$\mathsf{Decap}^+_{sk}(c)$:
(a) Parse $(c_1, c_2) \leftarrow c$.
(b) $\hat{K} \leftarrow {c_1}^{sk}$
(c) $\hat{r}_1 \leftarrow H(c_2 \oplus H'(\hat{K}))$
(d) If $c_1 \neq g^{\hat{r}_1}$ return $\perp$.
(e) $K \leftarrow H^{\#}(\hat{K})$
(f) Return K.

Figure 19: Specification of Π = ElGamal-PK-KEM and Π^+ = ElGamal-PK-KEM$^+$. Observe that Π^+ is constructed from Π following the stencil from Figure 18.

A.3 Plaintext-aware and IND-CCA-secure ElGamal-PK-KEM

We apply the transformation from Definition 17 to ElGamal's PK-KEM [ElG84]. The latter scheme is OW-secure under the CDH assumption.

Definition 18 (ElGamal-PK-KEM)**.** *Let $G = \langle g \rangle$ denote a cyclic group of prime order q. The* ElGamal key encapsulation mechanism $\Pi = (\mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap})$ *is specified in Figure 19 (left part). We have* $\mathtt{CoinSp}(k) = \mathbb{Z}_q$*,* $\mathtt{CipherSp}(k) = G$*, and* $\mathtt{KeySp}(k) = G$*.*

Our transformation turns ElGamal-PK-KEM into ElGamal-PK-KEM$^+$:

Definition 19 (ElGamal-PK-KEM$^+$)**.** *Let $G = \langle g \rangle$ denote a cyclic group of prime order q. ElGamal-PK-KEM$^+$ is specified in Figure 19 (right part), where $H : \{0,1\}^k \to \mathbb{Z}_q$ and $H', H^{\#} : G \to \{0,1\}^k$ denote hash functions,* $\mathtt{CoinSp}^+(k) = \{0,1\}^k$*,* $\mathtt{CipherSp}^+(k) = G \times \{0,1\}^k$*, and* $\mathtt{KeySp}^+(k) = \{0,1\}^k$*.*

Security of ElGamal-PK-KEM$^+$ is established by Theorem 7:

Theorem 8. *ElGamal-PK-KEM$^+$ is plaintext-aware* (PA) *and* IND-CCA*-secure under the CDH assumption in the random oracle model.*

Decapsulation in ElGamal-PK-KEM$^+$ takes one more exponentiation than decapsulation in the original scheme, i.e. the overhead introduced by the transformation is small. The length of ciphertexts is increased by only k bits and all newly introduced hash functions $H, H', H^{\#}$ are straight-forward to instantiate.

B Auxiliary lemmas

Lemma 16. *Let $0 \leq p \leq 1$ be a real number. Let $E_1, \ldots, E_n$ be a set of events that occur with at least probability p, i.e. $\Pr[E_i] \geq p$ for all $1 \leq i \leq n$. Then event $E = E_1 \wedge \ldots \wedge E_n$ occurs with probability at least $1 - n(1-p)$.*

Proof. We have $\Pr[E] = 1 - \Pr[\neg E] = 1 - \Pr[\neg E_1 \vee \ldots \vee \neg E_n] \geq 1 - \sum_{i=1}^{n} \Pr[\neg E_i] \geq 1 - n(1-p)$. □

Lemma 17. *Let S_0, S_1, E be events that satisfy $\Pr[S_0|\neg E] = \Pr[S_1|\neg E]$. Then we have $\Pr[E] \geq |\Pr[S_0] - \Pr[S_1]|$.*

Proof. We have

$$\begin{aligned} |\Pr[S_0] - \Pr[S_1]| &= |\Pr[S_0|E]\Pr[E] + \Pr[S_0|\neg E]\Pr[\neg E] \\ &\quad - \Pr[S_1|E]\Pr[E] - \Pr[S_1|\neg E]\Pr[\neg E]| \\ &= |\Pr[S_0|E]\Pr[E] - \Pr[S_1|E]\Pr[E]| \\ &= |\Pr[S_0|E] - \Pr[S_1|E]| \cdot \Pr[E] \\ &\leq \Pr[E] \ . \end{aligned}$$

□