Searchable encryption with decryption in the standard model

Dennis Hofheinz[*] Enav Weinreb[†]

Abstract

A *searchable public key encryption (PEKS) scheme* allows to generate, for any given message W, a trapdoor T_W, such that T_W allows to check whether a given ciphertext is an encryption of W or not. Of course, T_W should not reveal any additional information about the plaintext. PEKS schemes have interesting applications: for instance, consider an email gateway that wants to prioritize or filter encrypted emails based on keywords contained in the message text. The email recipient can then enable the gateway to do so by releasing the trapdoors for the corresponding keywords. This way, the gateway can check emails for these keywords, but it learns nothing more about the email contents.

PEKS schemes have first been formalized and constructed by Boneh et al.. But with one exception, no known construction of a PEKS scheme supports the decryption of ciphertexts. That is, known constructions allow to *test* for a certain message, but they do not allow to *retrieve* the message, even when having the full secret key. Besides being somewhat unnatural for an encryption scheme, this "no-decryption"-property also limits the applicability of a PEKS scheme. The one exception, a PEKS scheme with decryption due to Fuhr and Paillier, is formulated in the random oracle model, and inherently relies on the statistical properties of the random oracle. In fact, Fuhr and Paillier leave it as an open problem to construct a PEKS scheme with decryption in the standard model.

In this paper, we construct the first PEKS scheme with decryption (PEKSD scheme) in the standard model. Our sole assumption is an anonymous IBE scheme. We explain the technical difficulties that arise with previous attempts to build a PEKS scheme with decryption and how we overcome these difficulties. Technically, we isolate a vital additional property of IBE schemes (a property we call *well-addressedness* and which states that a ciphertext is tied to an identity and will be rejected when trying to decrypt with respect to any other identity) and show how to generically achieve it.

Our construction of a PEKSD scheme from an anonymous IBE scheme provides a natural example of a *non-shielding* construction (in which the decryption algorithm queries the encryption algorithm). Gertner et al. have shown that an IND-CCA secure public key encryption scheme cannot be constructed and proven from an IND-CPA secure scheme in a black-box and *shielding* way. However, our results give evidence that encryption queries in the decryption algorithm may well prove useful in a security reduction.

[*]CWI, Amsterdam, The Netherlands. E-mail: Dennis.Hofheinz@cwi.nl
[†]CWI, Amsterdam, The Netherlands. E-mail: e.n.weinreb@cwi.nl

1 Introduction

Motivation. Consider an email gateway G that stores the email for a number of users. Suppose each email message is encrypted and tagged with a number of keywords (such as "meeting" or "price offer" or similar). We assume that the keywords are also encrypted for privacy reasons. Now imagine that a user U wants to retrieve all messages tagged with the keyword "meeting". Since U does not want to download all messages, U needs to delegate to G the capability to recognize and then filter emails tagged with keyword "meeting".

Searchable public key encryption (PEKS). This can be done with a *searchable public key encryption (PEKS) scheme* (as defined by Boneh et al. [3]). Basically, a PEKS scheme is a public key encryption (PKE) scheme, in which, *instead of decryption*, the secret key allows to generate trapdoors T_W for arbitrary messages W. Using T_W, it is possible to check whether an arbitrary given ciphertext c is an encryption of W or not. However, if c is not an encryption of W, the trapdoor T_W should not give any information about the true encrypted message W' (besides $W' \neq W$ of course). A PEKS scheme can be used in the above example to encrypt the keywords of an email. The user U can then delegate the capability of checking whether an email is tagged with a keyword $W =$ "meeting" simply by handing the trapdoor $T_{\text{"meeting"}}$ to the gateway G.

Searchable public key encryption with decryption (PEKSD). However, a PEKS scheme does not allow the user U to *decrypt* the encrypted keywords, and thus U cannot, say, sort her emails according to the keywords, or even just see the full list of keywords attached to a message. It might seem natural to then encrypt the keyword not only by a PEKS scheme, but additionally use a traditional PKE scheme. The user U could then retrieve the keyword by a PKE decryption. However, this solution does not ensure that the PEKS encryption and the PKE encryption are really consistent (i.e., referring to the same keyword). This can become problematic if U relies on the gateway's actions (which only depend on the PEKS encryptions) during local computations (which then only depend on the PKE encryptions). This leads to a definition of a *searchable public key encryption scheme with decryption (PEKSD scheme)*. A PEKSD scheme is identical to a PEKS scheme, only that the secret key allows to also decrypt ciphertexts (in the usual PKE sense).

Related literature. The definition of a PEKS scheme was first formalized by Boneh et al. [3], who also noticed a connection between identity-based encryption (IBE) schemes and PEKS schemes. This connection appears natural: in an IBE scheme, the master secret key can be used to generate user secret keys which allow to decrypt a certain subset of ciphertexts; this seems a natural starting point for trapdoors in the PEKS sense. The construction from [3] starts from a specific IBE scheme (specifically, the Boneh-Franklin IBE scheme [2]). A more general connection to (anonymous) IBE schemes was given by Abdalla et al. [1]; in particular, combining the results of [1] with the anonymous IBE scheme from Boyen and Waters [5] yields a PEKS construction without random oracles. Abdalla et al. also generalized the notion of PEKS consistency[1], and corrected a flaw concerning consistency from [3]. However, Abdalla et al. leave open the question to construct a *perfectly* consistent PEKS scheme.

Zhang and Imai [12] consider a "hybrid" of a PEKS and a PKE scheme, in which PKE encryptions are tagged with a PEKS encryption. While their solution provides decryption of the PKE part of the ciphertext, it does not allow to retrieve the PEKS keyword. Hence, while the solution of [12] allows to "tie together" a PEKS keyword and a PKE message, it does not guarantee any relation between message and keyword. (In particular, their construction does not imply a PEKSD scheme, as required for our purposes.)

Possibly closest to our work is the work of Fuhr and Paillier [8]: they construct a PEKSD scheme in the random oracle model. As we will argue below, the proof of their construction hinges on the statistical properties of the random oracle and cannot be easily transported to the standard model. This

[1]Roughly, consistency of a PEKS scheme ensures that the testing algorithms return results that are consistent with the actually encrypted message.

is also noticed by Fuhr and Paillier who specifically mention designing a solution in the standard model as an open problem.

Our contribution. We construct the first PEKSD scheme in the standard model (i.e., without random oracles). Our construction is surprisingly simple and, similar to previous PEKS constructions, only assumes an anonymous identity-based encryption (IBE) scheme as a basis.[2] We stress that anonymous IBE schemes can indeed be implemented from standard bilinear complexity assumptions, cf. Boyen and Waters [5]. For our construction, we isolate and define a useful property of the underlying IBE scheme we call *well-addressedness*. Informally, a well-addressed IBE scheme has ciphertexts which only decrypt correctly under at most one identity (i.e., a ciphertext is tied to an identity). We show how to turn any anonymous IBE scheme into an anonymous *and* well-addressed IBE scheme.[3] In the following, we will motivate and explain our construction in detail.

A first attempt. As a first attempt towards constructing a PEKSD scheme, assume an IBE scheme $\mathcal{IBE} = (\mathrm{IBG}, \mathrm{IBT}, \mathrm{IBE}, \mathrm{IBD})$. (For formal definitions of IBE and PKE schemes, see Section 2.) It seems natural to start from an IBE scheme, since an IBE master secret key allows to produce user secret keys T_{id} that allow to decrypt a certain class of ciphertexts (namely, those ciphertexts associated with an identity id). Hence, we might try to identify IBE identities with PEKS messages. Concretely, we could try to construct a PEKS encryption of W as

$$\mathrm{PEKS}_{PK}(W) = \mathrm{IBE}_{PK,W}(F),$$

i.e., as an IBE encryption under identity W of an arbitrary (for simplicity fixed) IBE plaintext F. The trapdoor for testing if a given ciphertext c is an encryption of W would be the IBE user secret key T_W for identity W. Accordingly, one can then test whether c is an encryption of W by checking if c decrypts to F under T_W.

Observe that secrecy of this naive PEKS scheme now requires anonymity from the IBE scheme (this was also noticed by Abdalla et al. [1], who consider a related but more complex generic construction for a PEKS scheme without decryption). Namely, given an IBE ciphertext, it should not be possible to determine under which identity this ciphertext was encrypted. Observe also that there are two problems with this naive scheme: first, it is unclear how to decrypt. Second, the usual security requirements on IBE schemes (including anonymity) give no guarantees what happens if a ciphertext is decrypted under an identity *different* from the one under which it was encrypted. Concretely, for all we know about $\mathcal{IBE}$, we might have that

$$\mathrm{IBD}_{T_{W'},W'}(\mathrm{IBE}_{PK,W}(F)) = F$$

for a cleverly chosen $W' \neq W$. (This would violate PEKS consistency, since now the PEKS test returns that $\mathrm{PEKS}_{PK}(W) = \mathrm{IBE}_{PK,W}(F)$ is an encryption of $W' \neq W$.)

Adding decryption. To solve the first problem of our naive scheme (i.e., the lack of decryption), we might add a PKE encryption of the PEKS message W to the ciphertext. (Also the scheme from Fuhr and Paillier [8] follows this path, see below for more information on their approach.) So assume a PKE scheme $\mathcal{PKE} = (\mathrm{PKG}, \mathrm{PKE}, \mathrm{PKD})$, and consider the construction:

$$\mathrm{PEKS}_{PK}(W) = c = (c_1, c_2) = (\mathrm{IBE}_{PK_1,W}(F), \mathrm{PKE}_{PK_2}(W)). \tag{1}$$

This obviously ensures decryptability (assuming that the PEKS secret key contains the PKE secret key SK_2), but it creates two new problems. First, combining two ciphertexts often invites malleability-style attacks on the (IND-CCA) security of an encryption scheme (cf. Zhang et al. [13], Dodis and Katz [7]).

[2] For our construction, we actually use a *well-addressed* anonymous IBE scheme *and* a PKE scheme. However, we show in Section 4 how to construct well-addressed anonymous IBE schemes from anonymous IBE schemes; also, it is known how to construct PKE schemes from IBE schemes (Canetti et al. [6]).

[3] Note that turning an IBE scheme into a well-addressed IBE scheme is trivial; simply add the identity to each ciphertext. The difficulty lies in preserving anonymity, which is vital for our application: constructing a PEKSD scheme.

The mutual dependency problem. However, a much graver problem is that now *PEKS consistency* is at stake, in the following sense. For consistency, we require that the PEKS testing algorithm (which tests whether c_1 is encrypted under a given identity W) yields results which are consistent with the actual PEKS decryption algorithm. That is, we want that $\mathrm{Test}_{PK}(c, T_W) =$ "yes" if and only if $\mathrm{PEKSD}_{MK}(c) = W$. Now the holder of the PEKS secret key MK can first extract W from the PKE part $c_2 = \mathrm{PKE}_{PK_2}(W)$ of the ciphertext and then check if the IBE part c_1 is consistent with W. However, there is no guarantee for the holder of a trapdoor T_W that the PKE part c_2 is indeed an encryption of W. In fact, more generally, the values of c_1 and c_2 depend on each other, since the results of testing and decryption must be "synchronized".

The approach of Fuhr and Paillier. Fuhr and Paillier [8] approach this "synchronization" problem as follows: essentially, they also encrypt the *randomness* used during both IBE and PKE encryptions. Concretely, their IBE ciphertext part is an encryption of the randomness used in the PKE encryption, and the PKE ciphertext part contains the actual PEKS message *and* the randomness used in the IBE encryption.[4] In particular, they create a cyclic dependency as follows:

$$\begin{array}{ccc} \text{PKE decryption} & \longrightarrow & \text{IBE randomness, PEKS message} \\ \uparrow & & \downarrow \\ \text{PKE randomness} & \longleftarrow & \text{IBE decryption} \end{array}$$

where an arrow $X \to Y$ means that X allows to *deterministically* obtain Y, given the full ciphertext of Fuhr and Paillier's scheme. The holder of the full PEKS secret key can "jump into" that cycle at the upper left corner (the PKE decryption) and check for consistency by following the arrows deterministically in a full circle. On the other hand, the holder of the trapdoor for a PEKS message W can "jump into" the cycle at the lower right corner (the IBE decryption) and check for consistency similarly.

While elegant from a conceptual point of view, this approach has the disadvantage that neither the randomness used in the IBE encryption nor that used in the PKE encryption is completely hidden; instead, all random coins are additionally encrypted. That implies that a straightforward reduction to IBE or PKE security is not possible. The reason why [8] still can prove security follows from their use of a random oracle: they first prove that the encryption randomness is statistically hidden, from which point on a "usual" reduction can be conducted.

Our approach. We solve the consistency problem in a different way. To see how, reconsider the construction from (1). The problem with this construction was that the holder of a PEKS trapdoor T_W cannot check that the PKE ciphertext c_2 is really an encryption of W. But now suppose that c_1 is an encryption of the *randomness* used in the PKE encryption c_2:

$$\mathrm{PEKS}_{PK}(W) = c = (c_1, c_2) = (\mathrm{IBE}_{PK_1,W}(R), \mathrm{PKE}_{PK_2}(W; R)).$$

Then, decrypting c_1 yields the randomness for c_2, which allows to check whether c_2 is an encryption of W. This is exactly the information that the holder of T_W needs to decide whether the whole ciphertext is consistent. On the other hand, the holder of the PEKS secret key can first decrypt c_2 to obtain a "candidate message" W, generate an IBE trapdoor T_W for c_1, and then proceed as the holder of T_W.

IND-CCA attacks and well-addressed IBE schemes. This simple construction thus ensures (perfect) consistency; however, we still might get into trouble if we strive for encryption security against chosen-ciphertext attacks (IND-CCA security). Indeed, suppose that an IND-CCA adversary A gets a challenge ciphertext

$$c^* = (c_1^*, c_2^*) = (\mathrm{IBE}_{PK_1,W^*}(R), \mathrm{PKE}_{PK_2}(W^*; R)),$$

and A's goal is to determine whether $W^* = W_0$ or $W^* = W_1$ (for adversarially chosen messages W_0 and W_1). Now suppose further that the IBE scheme has the property that $\mathrm{IBD}_{T_{W'},W'}(\mathrm{IBE}_{PK_1,W}(R)) =$

[4] Actually, [8] use a more "low-level" KEM/DEM based approach to avoid some technicalities of our high-level description.

0 for all R and $W' \neq W$. (This property does not violate anonymity or security of the IBE scheme.) Then, A can use its CCA oracle and request a decryption of

$$c = (c_1, c_2) = (c_1^*, \mathrm{PKE}_{PK_2}(W_0; 0)),$$

where $\mathrm{PKE}_{PK_2}(W_0; 0)$ denotes a PKE encryption of W_0 with randomness 0. Technically, $c \neq c^*$ with high probability, so A gets the correct decryption W of c. By definition, PEKSD will first decrypt c_2 (which yields W_0) and then decrypt $c_1 = c_1^*$ under identity W_0. If $W^* = W_0$, then decrypting c_1^* will yield the randomness R used to encrypt c_2^*, which is $\neq 0$ with high probability. After this, PEKSD will check $c_2 \stackrel{?}{=} \mathrm{PKE}_{PK_2}(W_0; R)$, which is most likely not the case, so PEKSD will reject the ciphertext c. Conversely, if $W^* = W_1$, then PEKSD will decrypt c_1^* under identity W_0, which by our assumption on $\mathcal{IBE}$ yields 0. Then, PEKSD will successfully verify that $\mathrm{PKE}_{PK_2}(W_0; 0)$ and output W_0. Summarizing, A can break the IND-CCA security of the PEKSD scheme with only one CCA query.

For the described attack, it is crucial that the IBE scheme does not reject ciphertexts when trying to decrypt under a "wrong" identity. In fact, we can show that when the IBE scheme is *well-addressed* (which means that the decryption algorithm rejects ciphertexts when trying to decrypt under the wrong identity), we can prove the described PEKSD scheme IND-CCA secure. As hinted, IBE schemes may or may not be well-addressed. However, we give a construction that turns any IND-CCA secure and anonymous IBE scheme into a well-addressed scheme, while preserving IND-CCA security and anonymity. (For more details on our construction, see Section 4.)

Perfect consistency. We stress that our PEKSD construction enjoys *perfect* consistency (i.e., the test performed by a holder of a trapdoor T_W will *always* be consistent with the output of the decryption algorithm). While already the PEKSD scheme of Fuhr and Paillier [8] achieves perfect consistency, our scheme is the first scheme that does so in the standard model.

Privacy preserving trapdoors. As a side remark, we informally introduce PEKS schemes with privacy preserving trapdoors, which allow to hide the keyword even from the server holding the trapdoor, assuming sufficient entropy in the message space. We discuss a construction in the RO model to compile a PEKS scheme (with or without decryption) into a scheme with privacy preserving trapdoors.

Importance of non-shielding constructions. Our PEKSD scheme constitutes a natural example of a *non-shielding* construction (that is, a construction of an encryption scheme from another encryption scheme, in which the constructed decryption algorithm queries the encryption algorithm of the underlying scheme). Gertner et al. have been shown that an IND-CCA secure public key encryption scheme cannot be constructed and proven from an IND-CPA secure scheme in a black-box and *shielding* way. Their work in fact raises the question whether non-shielding reductions are of importantance at all. Our results give evidence that the answer to that question might be "yes": encryption queries in the decryption algorithm may well prove useful in a security reduction. (We should mention that, independently, Rosen and Segev [11] gave another example of a non-shielding construction of an IND-CCA secure encryption scheme.)

IBE with powerful center. Boneh et al. [3, Section 2.1] prove that any PEKS scheme gives rise to an anonymous IBE scheme. If we plug our PEKSD scheme into the construction from [3, Lemma 2.3], then we obtain an IBE scheme in which the master secret key can be used to efficiently break the anonymity and to decrypt arbitrary ciphertexts. We call such an IBE scheme an *IBE scheme with powerful center*. We envision that an IBE scheme with powerful center can be useful in optimistic protocols (in which a trusted party knows the master secret key and only intervenes upon conflicts): generally, encryptions are anonymous; however, as soon as a conflict occurs, the trusted party can break anonymity and identify cheaters.

2 Preliminaries

A probabilistic polynomial-time (PPT) algorithm A is a randomized algorithm which runs in time polynomial in the length of its input. Sometimes we will want to make explicit the random coins that A uses; we write $A(x;r)$ to express that A should be run on input x and with random coins r. A function $f : \mathbb{N} \to \mathbb{R}$ is negligible iff it vanishes faster than any polynomial, i.e., iff $\forall c > 0 \, \exists k_0 \, \forall k > k_0 : |f(k)| < k^{-c}$. If $\mathcal{S}$ is a set, then $x \stackrel{\$}{\leftarrow} \mathcal{S}$ denotes the process of assigning x a value from $\mathcal{S}$ uniformly at random.

The definitions for families of pairwise independent hash functions, public-key encryption schemes, and identity-based encryption schemes have been outsourced into Appendix A due to space constraints.

3 Searchable public key encryption

We start with a definition of PEKS as it appears in [3].

Definition 3.1 (PEKS [3]). *A non interactive public key encryption with keyword search (PEKS) scheme consists of the following polynomial time randomized algorithms:*

1. $\mathrm{KeyGen}(1^k)$*: Takes a security parameter* 1^k *and generates a public/secret key pair* (PK, MK).
2. $\mathrm{PEKS}_{PK}(W)$*: For a public key* PK *and a word* W*, produces a searchable encryption of* W.
3. $\mathrm{Trapdoor}_{MK}(W)$*: For a secret key* MK *and a word* W*, produces a trapdoor* T_W.
4. $\mathrm{Test}_{PK}(S, T_W)$*: Given a public key* PK*, a searchable encryption* $S = \mathrm{PEKS}(PK, W')$*, and a trapdoor* $T_W = \mathrm{Trapdoor}_{MK}(W)$*, outputs 'yes' if* $W = W'$ *and 'no' otherwise.*

We continue to the definition of security against an active attacker as it appears in [3].

Definition 3.2 (PEKS security [3]). *A PEKS scheme* $\mathcal{PEKS} = (\mathrm{KeyGen}, \mathrm{Trapdoor}, \mathrm{PEKS}, \mathrm{Test})$ *is called* indistinguishable under chosen-trapdoor attacks (IBE-IND-CTA secure) *iff for every pair of PPT adversaries* $A = (A_1, A_2)$*, the function*

$$\mathsf{Adv}^{\mathsf{peks\text{-}ind\text{-}cta}}_{\mathcal{PEKS},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{peks\text{-}ind\text{-}cta}}_{\mathcal{PEKS},A}(k) = 1\right] - 1/2$$

is negligible in k, where $\mathsf{Exp}^{\mathsf{peks\text{-}ind\text{-}cta}}_{\mathcal{PEKS},A}(k)$ *is the following experiment:*

$$\begin{array}{l}
\textbf{Experiment } \mathsf{Exp}^{\mathsf{peks\text{-}ind\text{-}cta}}_{\mathcal{PEKS},A}(k) \\
\quad (MK, PK) \leftarrow \mathrm{KeyGen}(1^k) \\
\quad (m_0, m_1, st) \leftarrow A_1^{\mathrm{Trapdoor}_{MK}(\cdot)}(PK) \\
\quad b \stackrel{\$}{\leftarrow} \{0,1\} \\
\quad c^* \leftarrow \mathrm{PEKS}_{PK}(m_b) \\
\quad b' \leftarrow A_2^{\mathrm{Trapdoor}_{MK}(\cdot)}(st, c^*) \\
\quad \text{Return 1 iff } b = b'
\end{array}$$

To avoid trivialities, we require that A_1 *always returns* m_0, m_1 *with* $|m_0| = |m_1|$*, that* A_1 *never returns a value* m_i *on which* $\mathrm{Trapdoor}(MK, \cdot)$ *has been queried, and that* A_2 *never queries* $\mathrm{Trapdoor}_{MK}(m_0)$ *and* $\mathrm{Trapdoor}_{MK}(m_1)$.

We consider enhanced PEKS schemes which enable the holder of the secret key to decrypt.

Definition 3.3 (PEKS with decryption (PEKSD)). *A PEKS scheme with decryption (PEKSD scheme) is a PEKS scheme with the following extra polynomial time randomized algorithm:*

1. $\mathrm{PEKSD}_{MK}(S)$*: Given a secret key* MK *and a searchable encryption* $S = \mathrm{PEKS}_{PK}(W)$ *outputs* W.

We require correctness *in the sense that for* (MK, PK) *in the range of* $\mathrm{KeyGen}(1^k)$ *and all messages* $W \in \mathcal{M}$*, we have* $\mathrm{PEKSD}_{MK}(\mathrm{PEKS}_{PK}(W)) = W$ *always.*

We also require consistency *of* PEKSD_{MK} *with* Test*, even for inconsistent ciphertexts, in the following sense. We require that for all keypairs* (MK, PK) *in the range of* $\mathrm{KeyGen}(1^k)$*, for all syntactically possible encryptions* S *and words* W*, and all trapdoors* T_W *in the range of* $\mathrm{Trapdoor}_{MK}(W)$*, we have that*

$$\mathrm{Test}_{PK}(S, T_W) = \mathrm{yes} \textit{ if and only if } \mathrm{PEKSD}_{MK}(S) = W.$$

Definition 3.4 (PEKSD security)**.** *A PEKSD scheme* $\mathcal{PEKSD} = (\mathrm{KeyGen}, \mathrm{Trapdoor}, \mathrm{PEKS}, \mathrm{Test}, \mathrm{PEKSD})$ *is called* indistinguishable under chosen-ciphertext attacks (IBE-IND-CCA secure) *iff for every pair of PPT adversaries* $A = (A_1, A_2)$*, the function*

$$\mathsf{Adv}^{\mathsf{peksd\text{-}ind\text{-}cca}}_{\mathcal{PEKSD},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{peksd\text{-}ind\text{-}cca}}_{\mathcal{PEKSD},A}(k) = 1\right] - 1/2$$

is negligible in k, where $\mathsf{Exp}^{\mathsf{peksd\text{-}ind\text{-}cca}}_{\mathcal{PEKSD},A}(k)$ *is the following experiment:*

$$\begin{array}{l}
\textbf{Experiment } \mathsf{Exp}^{\mathsf{peksd\text{-}ind\text{-}cca}}_{\mathcal{PEKSD},A}(k) \\
\quad (MK, PK) \leftarrow \mathrm{KeyGen}(1^k) \\
\quad (m_0, m_1, st) \leftarrow A_1^{\mathrm{PEKSD}_{MK}(\cdot), \mathrm{Trapdoor}_{MK}(\cdot)}(PK) \\
\quad b \stackrel{\$}{\leftarrow} \{0,1\} \\
\quad c^* \leftarrow \mathrm{PEKS}_{PK}(m_b) \\
\quad b' \leftarrow A_2^{\mathrm{PEKSD}_{MK}(\cdot), \mathrm{Trapdoor}_{MK}(\cdot)}(st, c^*) \\
\quad \text{Return 1 iff } b = b'
\end{array}$$

To avoid trivialities, we require that A_1 *always returns* m_0, m_1 *with* $|m_0| = |m_1|$*, that* A_1 *never returns a value* m_i *on which* $\mathrm{Trapdoor}_{MK}(\cdot)$ *has been queried, and that* A_2 *never queries* $\mathrm{Trapdoor}_{MK}(m_0)$*,* $\mathrm{Trapdoor}_{MK}(m_1)$*, and* $\mathrm{PEKSD}_{MK}(c^*)$*.*

4 Well-addressed IBE schemes

The security definition. Informally, an IBE scheme is well-addressed if it is not feasible, given an encryption of a random message under an adversarially chosen identity, to find another identity under which the given ciphertext is not rejected, i.e., decrypts to an (arbitrary) message from the message space. For our results, we need that this property holds even if the adversary gets the master IBE key:

Definition 4.1 (Well-addressed IBE scheme)**.** *An IBE scheme* $\mathcal{IBE} = (\mathrm{IBG}, \mathrm{IBT}, \mathrm{IBE}, \mathrm{IBD})$ *is called* well-addressed *iff for every PPT adversary* $A = (A_1, A_2)$*, the function*

$$\mathsf{Adv}^{\mathsf{ibe\text{-}wa}}_{\mathcal{IBE},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{ibe\text{-}wa}}_{\mathcal{IBE},A}(k) = 1\right] - 1/2$$

is negligible in k, where $\mathsf{Exp}^{\mathsf{ibe\text{-}wa}}_{\mathcal{IBE},A}(k)$ *is the following experiment:*

$$\begin{array}{l}
\textbf{Experiment } \mathsf{Exp}^{\mathsf{ibe\text{-}wa}}_{\mathcal{IBE},A}(k) \\
\quad (MK, PK) \leftarrow \mathrm{IBG}(1^k) \\
\quad (id, st) \leftarrow A_1(MK, PK) \\
\quad m \stackrel{\$}{\leftarrow} \{0,1\}^k \\
\quad c \leftarrow \mathrm{IBE}_{PK,id}(m) \\
\quad id' \leftarrow A_2(st, m, c) \\
\quad m' \leftarrow \mathrm{IBD}_{MK,id'}(c) \\
\quad \text{Return 1 iff } id' \neq id \text{ and } m' \neq \perp
\end{array}$$

How to construct a well-adressed IBE scheme. Not every IBE scheme is well-addressed. For instance, a scheme might decrypt invalid ciphertexts to, say, 0, instead of rejecting them with $\perp$. Hence, formally, no ciphertext at all is rejected. This does not contradict security or anonymity, but obviously breaks well-addressedness. There is a trivial way to turn any IBE scheme into a well-addressed one: append the identity to every ciphertext and check that identity during decryption. It is easy to see that this transformation achieves Definition 4.1 and preserves IBE-IND-CCA security, but *breaks anonymity*. However, our purposes require an IBE scheme which is anonymous, IBE-IND-CCA secure, *and* well-addressed.

A seemingly better idea (that preserves anonymity) would be to encrypt the identity id *along* with the message m (so actually the tuple (id, m) is encrypted). Upon decryption, the identity can then be extracted from the message and checked. But with this idea, a particularly "uncooperative" IBE scheme might decrypt messages under a "wrong" identity $id' \neq id$ to $(id', 0)$; such ciphertexts are then accepted as valid, which breaks well-addressedness. Similar to the initial example above, this does not contradict anonymity or secrecy, since no information about the message is leaked. Note that in this example, we can view the IBE scheme that is used as a basis in fact as part of the adversary: it tries to lure the decryption construction around it into accepting the ciphertext as valid.

So we need a slightly more sophisticated way to achieve well-addressedness. Concretely, similar to the previous example, our approach is to hide the identity as part of the encrypted message, so that anonymity is preserved. But to avoid the attack on well-addressedness above, we will equip the identity with an "authentication tag" which is hard to guess from the basic IBE scheme's perspective.

Construction 4.2 (Well-addressed IBE scheme)**.** Let $\mathcal{IBE}' = (\mathrm{IBG}', \mathrm{IBT}', \mathrm{IBE}', \mathrm{IBD}')$ be an IBE scheme with identity space $\{0,1\}^k$ and message space $\{0,1\}^\ell$ for a polynomially bounded $\ell = \ell(k) > 3k$. Let $\mathcal{H} = (\mathcal{H}_k)_{k\in\mathbb{N}}$ be a family of pairwise independent hash functions mapping from $\{0,1\}^k$ to $\{0,1\}^{3k}$. In this situation, define an IBE scheme $\mathcal{IBE} = (\mathrm{IBG}, \mathrm{IBT}, \mathrm{IBE}, \mathrm{IBD})$, with message space $\{0,1\}^{\ell-3k}$ and identity space $\{0,1\}^k$, as follows:

- $\mathrm{IBG}(1^k)$ uniformly samples $h \stackrel{\$}{\leftarrow} \mathcal{H}_k$, runs $(MK', PK') \leftarrow \mathrm{IBG}'(1^k)$, and returns $(MK, PK) := ((MK', h), (PK', h))$.
- $\mathrm{IBT}(MK, id)$ parses $MK = (MK', h)$ and returns $T = (\mathrm{IBT}'(MK', id), h)$.
- $\mathrm{IBE}_{PK,id}(m)$ parses $PK = (PK', h)$ and returns $c := \mathrm{IBE}'_{PK',id}(h(id), m)$.
- $\mathrm{IBD}_{T,id}(c)$ parses $T = (T', h)$, computes $m' \leftarrow \mathrm{IBD}'_{T',id}(c)$, then parses $m' = (Y, m)$, and finally returns m if $Y = h(id)$, and $\perp$ otherwise.

So roughly speaking, Construction 4.2 encrypts $h(id)$ along with m. We will now formally prove that this modification does not damage the secrecy and the anonymity of the underlying IBE scheme, and we will prove that this modification achieves well-addressedness.

Lemma 4.3 (Construction 4.2 preserves IBE-IND-CCA)**.** *In the situation of Construction 4.2, if $\mathcal{IBE}'$ is IBE-IND-CCA secure, then so is $\mathcal{IBE}$.*

Proof. This can be shown using a merely syntactic reduction and will be given in Appendix B. □

Lemma 4.4 (Construction 4.2 preserves IBE-ANO-CCA)**.** *In the situation of Construction 4.2, if $\mathcal{IBE}'$ is IBE-IND-CCA secure and IBE-ANO-CCA secure, then $\mathcal{IBE}$ is IBE-ANO-CCA secure.*

Proof. The reduction used in this proof is slightly more complex than the one from Lemma 4.3, since the challenge message m in the IBE-ANO-CCA experiment with $\mathcal{IBE}$ corresponds to a challenge message $m' = (h(id_b), m)$ in the IBE-ANO-CCA experiment with $\mathcal{IBE}'$ which *depends on the used identity id_b*. We hence provide a game-based proof for clarity.

Assume an adversary A on $\mathcal{IBE}$'s IBE-ANO-CCA property, and let **Game 1** be the original IBE-ANO-CCA experiment $\mathsf{Exp}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE},A}$. Let out_1 denote the experiment's output bit, so that

$$\Pr\left[out_1 = 1\right] - 1/2 = \mathsf{Adv}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE},A}(k)$$

by definition.

In **Game 2,** we modify the generation of the challenge ciphertext c^*. Recall that in the original experiment $\mathsf{Exp}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE},A}$, we have $c^* \leftarrow \mathrm{IBE}_{PK,id_b}(m)$. If we write $PK = (PK', h)$, this is equivalent to $c^* \leftarrow \mathrm{IBE}'_{PK',id_b}(h(id_b), m)$. In Game 2, we now construct c^* as $c^* \leftarrow \mathrm{IBE}'_{PK',id_b}(0^{3k}, m)$. A straightforward reduction to $\mathcal{IBE}'$'s IBE-IND-CCA security shows that

$$\begin{aligned}&\mathsf{Pr}\left[out_2 = 1\right] - \mathsf{Pr}\left[out_1 = 1\right]\\ &\quad= \mathsf{Pr}\left[\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE}',A^*}(k) = 1 \mid b = 1\right] - \mathsf{Pr}\left[\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE}',A^*}(k) = 1 \mid b = 0\right] = 2\mathsf{Adv}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE}',A^*}(k)\end{aligned}$$

is negligible, where A^* is a suitable IBE-IND-CCA adversary on $\mathcal{IBE}'$ that chooses $m_0 = (h(id_b), m)$ and $m_1 = (0^{3k}, m)$, and out_2 denotes the experiment output in Game 2.

Now note that in Game 2, the message $(0^{3k}, m)$ encrypted in c^* does no longer depend on the identity id_b used for that encryption. Hence, we can now reduce to $\mathcal{IBE}'$'s IBE-ANO-CCA security. Namely, we can construct an adversary A' on $\mathcal{IBE}'$'s IBE-ANO-CCA security, such that A' simulates A, but translates A's oracle calls like in the proof of Lemma 4.3. A' constructs its challenge message as $m' = (0^{3k}, m)$, where m is A's challenge message. This perfectly simulates Game 2, so that

$$\mathsf{Pr}\left[out_2 = 1\right] - 1/2 = \mathsf{Pr}\left[\mathsf{Exp}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE}',A'}(k) = 1\right] - 1/2 = \mathsf{Adv}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE}',A'}(k)$$

is negligible. Summing up, also $\mathsf{Adv}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE},A}(k)$ must be negligible, which shows the claim. □

Lemma 4.5 (Construction 4.2 achieves well-addressedness)**.** *In the situation of Construction 4.2, $\mathcal{IBE}$ is well-adressed.*

Proof. Note that the claim is unconditional, so we will not rely on any computational assumptions, but only on the fact that $\mathcal{H}$ is a family of pairwise independent hash functions.

Consider the well-addressedness experiment $\mathsf{Exp}^{\mathsf{ibe\text{-}wa}}_{\mathcal{IBE},A}$ with an arbitrary adversary A. Let r_{IBE} and r_{IBD} denote the respective random coins used by the experiment for the computations of $c \leftarrow \mathrm{IBE}_{PK,id}(m)$ and $m' \leftarrow \mathrm{IBD}_{MK,id'}(c)$. Denote by r_A the random coins that A_1 and A_2 are run with.

Recall that $PK = (PK', h)$ and $MK = (MK', h)$. Now fix arbitrary values PK', MK', m, and $r := (r_{\mathrm{IBE}}, r_{\mathrm{IBD}}, r_A)$ (but not h). Then, any pair of identities $id, id' \in \{0,1\}^k$ and a value $y = h(id)$ deterministically induces a ciphertext

$$c = \mathrm{IBE}_{PK,id}(m) = \mathrm{IBE}'_{PK',id}(h(id), m; r_{\mathrm{IBE}})$$

and thus a decryption

$$(y', \tilde{m}') = \tilde{m} = \mathrm{IBD}_{MK',id'}(c; r_{\mathrm{IBD}}). \tag{2}$$

By the universal property of h, we hence have for any fixed tuple $(PK', MK', m, r, id, id')$ with $id \neq id'$:

$$\mathsf{Pr}_h\left[y' = h(id')\right] = 2^{-3k},$$

where y' is defined through (2). A union bound over all values of $id, id' \in \{0,1\}^k$ with $id \neq id'$ yields

$$\mathsf{Pr}_h\left[\exists id, id' \in \{0,1\}^k \ : \ y' = h(id')\right] \leq 2^{-k}. \tag{3}$$

Now observe that any successful adversary run (in which A_2 finally produces an id' such that c decrypts to $m' \neq \perp$ under identity id') implies that there exist $id \neq id'$ with $y' = h(id')$. Since PK', MK', m, and r are chosen *independently* by the experiment, and (3) holds for all fixed such values, we get that A's probability to succeed in the $\mathsf{Exp}^{\mathsf{ibe\text{-}wa}}_{\mathcal{IBE},A}$ experiment is upper bounded by 2^{-k}. □

5 The Construction

We show how to construct a PEKSD scheme from an IBE scheme and a PKE scheme.

Construction 5.1 (PEKSD scheme from IBE and PKE)**.** Let $\mathcal{IBE} = (\mathrm{IBG}, \mathrm{IBT}, \mathrm{IBE}, \mathrm{IBD})$ be an IBE scheme, and let $\mathcal{PKE} = (\mathrm{PKG}, \mathrm{PKE}, \mathrm{PKD})$ be a PKE scheme, such that:

- $\mathcal{IBE}$ is anonymous, well-addressed[5], IBE-IND-CCA secure, and has identity and message space $\{0,1\}^k$,
- $\mathcal{PKE}$ is IND-CCA secure, has message space $\{0,1\}^k$, and the encryption algorithm PKE always uses at most k bits of randomness[6].

Consider the following construction of a PEKSD scheme:

- $\mathrm{KeyGen}(k)$: Executes key generation algorithms of both the IBE and the PKE. The public (secret) key is a concatenation of the two corresponding public (secret, respectively) keys. That is $PK = (PK_1, PK_2)$ and $MK = (MK_1, SK_2)$ where $(MK_1, PK_1) \leftarrow \mathrm{IBG}(k)$ and $(SK_2, PK_2) \leftarrow \mathrm{PKG}(k)$.
- $\mathrm{PEKS}((PK_1, PK_2), W)$: Given a word W, the encryption algorithm works as follows:
  1. Choose two random strings $r_1, r_2 \in \{0,1\}^k$.
  2. Compute $c_1 = \mathrm{IBE}_{PK_1,W}(r_1; r_2)$ and $c_2 = \mathrm{PKE}_{PK_2}(W; r_1)$.
  3. Output (c_1, c_2).
- $\mathrm{Trapdoor}_{(MK_1,SK_2)}(W)$: Given a word W, to generate a trapdoor, execute the trapdoor algorithm $T_W = \mathrm{IBT}(MK_1, W)$ and output the resulting user secret ket (T_W, W).
- $\mathrm{Test}_{(PK_1,PK_2)}(S, (T_W, W))$: Given a public key (PK_1, PK_2) and a searchable encryption $S = (c_1, c_2)$, as well as an IBE trapdoor $T_W = \mathrm{Trapdoor}((MK_1, SK_2), W)$ along with a word W, do the following:
  1. Compute $r_1' = \mathrm{IBD}_{T_W,W}(c_1)$, using the decryption algorithm of the IBE scheme and the user secret key associated with the identity W.
  2. Compute $c_2' = \mathrm{PKE}_{PK_2}(W; r_1')$, using the encryption algorithm of the PKE scheme with r_1' as the random string.
  3. if $c_2 = c_2'$ output 'yes', and otherwise output 'no'.
- $\mathrm{PEKSD}_{(MK_1,SK_2)}(S)$, given a secret key (MK_1, SK_2) and a searchable encryption $S = (c_1, c_2)$ do the following:
  1. Compute $W' = \mathrm{PKD}_{SK_2}(c_2)$, using the secret key of the PKE scheme.
  2. Compute $T_{W'} = \mathrm{IBT}(MK_1, W')$ using the exatraction algorithm of the IBE scheme.
  3. Compute $r_1' = \mathrm{IBD}_{T_{W'},W'}(c_1)$ using the decryption algorithm of the IBE scheme.
  4. Compute $c_2' = \mathrm{PKE}_{PK_2}(W'; r_1')$ using the encryption algorithm of the PKE scheme, with r_1' as a random string.
  5. if $c_2 = c_2'$ output W', and otherwise output $\perp$.

We remark that for efficiency, an identity-based key encapsulation mechanism (instead of a full IBE scheme) can be used, similar to [8].

Correctness and consistency. The correctness of our construction is immediate. To see that also the consistency requirement of Definition 3.3 is met, consider an arbitrary PEKSD keypair $(MK, PK) = ((MK_1, SK_2), (PK_1, PK_2))$ (as produced by KeyGen), $S = (c_1, c_2)$, W. Let (T_W, W) denote the unique trapdoor produced by $\mathrm{Trapdoor}_{MK}(W)$.[7] Then, by definition, $\mathrm{PEKSD}_{MK}(S) = W$ means

$$W = \mathrm{PKD}_{SK_2}(c_2) \text{ and } c_2 = \mathrm{PKE}_{PK_2}(W; r_1) \text{ where } r_1 = \mathrm{IBD}_{T_W,W}(c_1).$$

By the perfect correctness of $\mathcal{PKE}$, this is equivalent to

$$c_2 = \mathrm{PKE}_{PK_2}(W; r_1) \text{ for } r_1 = \mathrm{IBD}_{T_W,W}(c_1).$$

[5]Recall that the assumption of well-addressedness is without loss of generality, considering Construction 4.2.

[6]The assumption about PKE's use of random coins is without loss of generality, since one can always use a pseudorandom number generator to stretch k bits of randomness suitably.

[7]Uniqueness follows from our requirement that IBT is deterministic, cf. Definition A.4.

Game	c_1^*	c_2^*	Decryption rule
G_0	$\mathrm{IBE}_{PK_1,W}(r_1;r_2)$	$\mathrm{PKE}_{PK_2}(W;r_1)$	
G_1	$\mathrm{IBE}_{PK_1,W}(r_1;r_2)$	$\mathrm{PKE}_{PK_2}(W;r_1)$	reject (c_1^*, c_2) for $c_2 \neq c_2^*$
G_2	$\mathrm{IBE}_{PK_1,W}(0;r_2)$	$\mathrm{PKE}_{PK_2}(W;r_1)$	reject (c_1^*, c_2) for $c_2 \neq c_2^*$
G_3	$\mathrm{IBE}_{PK_1,0}(0;r_2)$	$\mathrm{PKE}_{PK_2}(W;r_1)$	reject (c_1^*, c_2) for $c_2 \neq c_2^*$
G_4	$\mathrm{IBE}_{PK_1,0}(0;r_2)$	$\mathrm{PKE}_{PK_2}(0;r_1)$	reject (c_1^*, c_2) for $c_2 \neq c_2^*$

Table 1: Games in the security proof of the PEKSD construction.

But this is equivalent to $\mathrm{Test}_{(PK_1,PK_2)}(S, (T_W, W)) = \text{yes}$.

Privacy against the server. In case of Construction 5.1, trapdoors are of the form (T_W, W) and thus contain the word W being tested for in plain. This is crucial for our construction, since we want to enable Test to decrypt an IBE ciphertext with respect to the "right" identity W. But obviously, this way the holder of the trapdoor (T_W, W) learns the word W which the trapdoor allows to test for. To a certain degree, this property of the trapdoor is unavoidable in general: if the trapdoor holder suspects that the trapdoor is associated with a word W, he can always encrypt W and see if the trapdoor recognizes W.

In our example from the introduction, an email gateway uses trapdoors to test incoming user emails for a keyword W. With our scheme (in which trapdoors are of the form (T_W, W)), the server hence knows all relevant keywords W, and what messages contains which keywords. However, an email user might be interested in *not* letting the server know the keyword W itself, but *only* providing the server with a trapdoor test (which in itself does not leak the keyword, but only accepts or rejects). In other words, we might want that, given a trapdoor for a randomly chosen keyword W, one cannot efficiently find W. (As outlined above, we cannot expect indistinguishability, but only one-wayness here.) We informally say that PEKS schemes with this property have *privacy-preserving trapdoors*.

We briefly sketch how to construct PEKS (and PEKSD) schemes with privacy-preserving trapdoors. As a basis, we assume a PEKS scheme $\mathcal{PEKS}$. As a first attempt, we can use $\mathcal{PEKS}$ not with messages W, but instead with messages $f(W)$ for a one-way permutation f. (That is, we construct a new PEKS scheme $\mathcal{PEKS}'$ in which $\mathrm{PEKS}'_{PK}(W) = \mathrm{PEKS}_{PK}(f(W))$, etc.) This way, trapdoors are constructed for messages $f(W)$, and hence finding W from a trapdoor for (uniform) W requires breaking the one-way property of f. With a similar construction, one obtains a PEKSD scheme with privacy-preserving trapdoors from any PEKSD scheme and a *trapdoor* one-way permutation.

Of course, this first attempt has the drawback that privacy holds only for a *uniform* message W, where one might hope even for privacy as soon as the message comes from a distribution with significant min-entropy. And indeed, truly random permutation is one-way as soon as the input distribution has significant min-entropy. Since (almost) truly random permutations can be constructed in the random oracle model (cf. [10]), we obtain a PEKS (but not a PEKSD) scheme with privacy-preserving trapdoors in the random oracle model.

6 Security Proof

We prove the security of the PEKSD scheme presented in Construction 5.1 using a series of games. The first game is the IND-CCA-PEKSD security game, while in the last game the adversary has information theoretically no chance of winning. We prove that every two adjacent games are indistinguishable to a polynomial time adversary, relying on the different properties of the IBE and the PKE schemes. The games differ in the way challenge messages are encrypted and in the way the decryption queries are being answered. The games are depicted in Table 6. For simplicity of notation, we denote by $G_i(A)$ the probability that $\mathsf{Adv}^{\text{peks-ind-cta}}_{\mathcal{PEKS},A}(k) = 1$ while adapting the $\mathsf{Adv}^{\text{peks-ind-cta}}_{\mathcal{PEKS},A}$ experiment to the changes described in G_i.

The difference between games G_0 and G_1 is that in the latter, after the adversary gets his challenge ciphertext (c_1^*, c_2^*), we reject decryption queries of the form (c_1^*, c_2) where $c_2 \neq c_2^*$. The next claim asserts that since the IBE scheme is well-addressed these two games are indistinguishable.

Claim 6.1. *If the IBE scheme in Construction 5.1 is well-addressed then games G_0 and G_1 are indistinguishable.*

Proof. Suppose there exists an adversary $A = (A_1, A_2)$ such that $G_0(A) - G_1(A) = f(k)$ is a non-negligible function of the security parameter. We construct an adversary B for the $\mathsf{Exp}^{\text{ibe-wa}}_{\mathcal{IBE},B}(k)$ experiment that succeeds with probability $f(k)$. The adversary B is described as follows:

1. Get the parameters (MK, PK) of the IBE from the experiment $\mathsf{Exp}^{\text{ibe-wa}}_{\mathcal{IBE},B}(k)$.
2. Generate public and private parameters for the PKE scheme.
3. Hand A the public parameters for the PEKSD scheme as described in the construction.
4. Answer PEKSD and $\mathrm{Trapdoor}$ queries by following the description in Construction 5.1. PKE operations can be done since B holds the secret key for the PKE system. IBE operations can be done since B has the master secret key of the IBE scheme.
5. Get (m'_0, m'_1) from A_1.
6. Pick $b \xleftarrow{\$} \{0,1\}$ and $r_1 \xleftarrow{\$} \mathcal{M}$.
7. Hand m'_b to the experiment $\mathsf{Exp}^{\text{ibe-wa}}_{\mathcal{IBE},B}(k)$ to obtain a ciphertext $c^*_1 = \mathrm{IBE}_{PK.m'_b}(r_1)$ and a message r_1.
8. Give (c^*_1, c^*_2) to A_2, where $c^*_2 = \mathrm{PKE}_{PK_2}(m'_b; r_1)$.
9. Handling decryption queries: given a query of the type (c_1, c_2), where $c_1 \neq c^*_1$, answer exactly as in step 4. On queries of the form (c^*_1, c_2) with $c_2 \neq c^*_2$, decrypt c_2 to obtain an identity id'. If $id' = m_b$, then reject. (Since $c_2 \neq c^*_2$ but $c_1 = c^*_1$, this ciphertext would have been rejected in both G_0 and G_1.) For $id' \neq m_b$, check if $\mathrm{IBD}_{PK,id'}(c^*_1) = \perp$. If yes, reject (again, this ciphertext would have been rejected in G_0 and G_1). If not, we have found an identity useful for attacking the well-addressedness of $\mathcal{IBE}$, so we can return id' to the experiment $\mathsf{Exp}^{\text{ibe-wa}}_{\mathcal{IBE},B}(k)$.

It is clear that in order to detect a difference between G_0 and G_1, A has to submit a decryption query (c^*_1, c_2) such that $\mathrm{IBD}_{PK,id'}(c^*_1) \neq \perp$ for id' being the decryption of c_2. But these queries, B manages to extract id' from B's query and can use it to break $\mathcal{IBE}$'s well-addressedness. We have

$$|G_0(A) - G_1(A)| \leq \mathsf{Adv}^{\text{ibe-wa}}_{\mathcal{IBE},B}$$

which proves the claim. □

Next, game G_2 differs from G_1 in the fact that the message encrypted by the IBE scheme is no longer related to the random string used in the PKE encryption. The indistinguishability of these games in based on the secrecy property of the IBE scheme.

Claim 6.2. *If the IBE scheme in Construction 5.1 is IBE-IND-CCA secure, then games G_1 and G_2 are indistinguishable.*

Proof. Suppose there exists an adversary $A = (A_1, A_2)$ such that $G_1(A) - G_2(A) = f(k)$ is a non-negligible function of the security parameter. We construct an adversary B to the $\mathsf{Exp}^{\text{ibe-ind-cca}}_{\mathcal{IBE},B}$ experiment that gets advantage $f(k)/4$. The adversary B is described as follows:

1. Get public parameters for the IBE scheme from the $\mathsf{Exp}^{\text{ibe-ind-cca}}_{\mathcal{IBE},B}$ experiment.
2. Generate public and private parameters to the PKE scheme.
3. Hand A_1 the public parameters for the PEKSD scheme as described in Construction 5.1.
4. Answer PEKSD and $\mathrm{Trapdoor}$ queries of A_1 by following the algorithms as described in Construction 5.1. PKE operations can be done since B holds the secret key for the PKE system. IBE operations are done by using oracle calls to the IBD and IBT algorithms, which are allowed in the $\mathsf{Exp}^{\text{ibe-ind-cca}}_{\mathcal{IBE},B}$ experiment.
5. Get (m'_0, m'_1) from A_1.
6. Pick $b \xleftarrow{\$} \{0,1\}$ and $r_1 \xleftarrow{\$} \mathcal{M}$.
7. Hand the $\mathsf{Exp}^{\text{ibe-ind-cca}}_{\mathcal{IBE},B}$ the following values: $id^* = m'_b$,$m_0 = r_1$, and $m_1 = 0$.

8. Get c^* from $\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}$ and give (c_1^*, c_2^*) to A_2, where $c_1^* = c^*$ and $c_2^* = \mathrm{PKE}_{PK_2}(m_b'; r_1)$.
9. Answer PEKSD and Trapdoor queries of A_2 as follows: if the query is of the form (c_1^*, c_2), then reject the query. Otherwise, answer exactly as in step (4). The key point is that since $c_1 \neq c_1^*$, the adversary B can still use oracle calls to the IBD and IBT algorithms.
10. Get the output b' from A_2.
11. Pick $b_1, b_2 \xleftarrow{\$} \{0,1\}$. If $b_1 = 0$, output $b^* = b'$, else, output $b^* = b_2$.

Note that in case the experiment $\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}$ chose to encrypt the message $m_0 = r_1$, then the experiment is distributed identically to the game G_1 while if it chose to encrypt the message $m_1 = 0$, then the experiment is distributed identically to the game G_2.

$$\Pr[\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}(k) = 1] =$$

$$= 1/2 \Pr[\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}(k) = 1 | b_1 = 0] + 1/2 \Pr[\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}(k) = 1 | b_1 = 1] =$$

$$= 1/2 \Pr[\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}(k) = 1 | b_1 = 0] + 1/4 =$$

$$= 1/2(1/2 \Pr[\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}(k) = 1 | b_1 = 0 \text{ and } m_0 = r_1 \text{ was encrypted}] +$$

$$+ 1/2 \Pr[\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}(k) = 1 | b_1 = 0 \text{ and } m_1 = 0 \text{ was encrypted}]) + 1/4 =$$

$$= 1/2(1/2(1 - G_1(A)) + 1/2 G_2(A)) + 1/4 =$$

$$= 1/2(1/2 - 1/2(G_1(A) - G_2(A))) + 1/4 =$$

$$= 1/4 - 1/4 f(k) + 1/4 = 1/2 - 1/4 f(k)$$

Therefore, $\mathsf{Adv}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}(k) = -f(k)/4$ is non-negligible. □

The difference between game G_3 and G_2 is that the identity used to encrypt a message in game G_3 is now replaced to be the zero identity. The games are proved indistinguishable based on the anonymity property of the IBE scheme.

Claim 6.3. *If the IBE scheme in Construction 5.1 is anonymous then games G_2 and G_3 are indistinguishable.*

Proof. Suppose there exists an adversary $A = (A_1, A_2)$ such that $G_2(A) - G_3(A) = f(k)$ is a non-negligible function of the security parameter. We construct an adversary B for the $\mathsf{Exp}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE},B}$ experiment that gets advantage $f(k)/4$. The adversary B is described as follows:

1. Get public parameters for the IBE scheme from the $\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}$ experiment.
2. Generate public and private parameters to the PKE scheme.
3. Hand to A_1 the public parameters for the PEKSD scheme as described in Construction 5.1.
4. Answer PEKSD and Trapdoor queries of A by following the algorithms as described in the PEKSD construction. PKE operations can be done since B holds the secret key for the PKE system. IBE operations are done by using oracle calls to the IBD and IBT algorithms, which are allowed in the $\mathsf{Exp}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE},B}$ experiment.
5. Get (m_0', m_1') from A_1.
6. Pick $b \xleftarrow{\$} \{0,1\}$ and $r_1 \xleftarrow{\$} \mathcal{M}$.
7. Hand to $\mathsf{Exp}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE},B}$ the following values: $id_0 = m_b'$, $id_1 = 0$, and $m = 0$.
8. Get c^* from $\mathsf{Exp}^{\mathsf{ibe\text{-}ano\text{-}cca}}_{\mathcal{IBE},B}$ and give (c_1^*, c_2^*) to A_2, where $c_1^* = c^*$ and $c_2^* = \mathrm{PKE}_{PK_2}(m_b'; r_1)$.
9. Answer PEKSD and Trapdoor queries as follows: if the query is of the form (c_1, c_2), where $c_1 = c_1^*$, then reject the query. Otherwise, answer exactly as in step (4). The key point is that since $c_1 \neq c_1^*$, the adversary B can still use oracle calls to the IBD and IBT algorithms.
10. Get the output b' from A_2.
11. Pick $b_1, b_2 \xleftarrow{\$} \{0,1\}$. If $b_1 = 0$, output $b^* = b'$, else, output $b^* = b_2$.

Note that in case the experiment $\mathsf{Exp}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},B}$ chose to encrypt under the identity $id_0 = m_b$, then the experiment is distributed identically to the game G_2 while if it chose to encrypt under the identity $id_1 = 0$, then the experiment distributes identically to the game G_3. The analysis of the advantage of B is identical to the analysis from Claim 6.2. □

Finally, game G_4 differs from G_3 in that the message encrypted by the PKE scheme no longer depends on the message W. This makes the encryption challenge information theoretically independent of the message, and thus the adversary has no advantage in guessing which of W_0 and W_1 was encrypted. The games are proven indistiguishable using the secrecy property of the PKE scheme.

Claim 6.4. *If the PKE scheme in Construction 5.1 is IND-CCA secure, then games G_3 and G_4 are indistinguishable.*

Proof. Suppose there exists an adversary $A = (A_1, A_2)$ such that $G_3(A) - G_4(A) = f(k)$ is a non-negligible function of the security parameter. We construct an adversary B to the $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},B}$ experiment that gets advantage $f(k)/4$. The adversary B is described as follows:

1. Get public parameters for the PKE scheme from the $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},B}$ experiment.
2. Generate public and private parameters to the IBE scheme.
3. Hand A the public parameters for the PEKSD scheme as described in Construction 5.1.
4. Answer PEKSD and $\mathrm{Trapdoor}$ queries by following the algorithms as described in the Construction 5.1. IBE operations can be done since B holds the master secret key for the IBE scheme. PKE operations are done by using oracle calls to the PKD algorithm, which are allowed in the $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},B}$ experiment.
5. Get (m'_0, m'_1) from A_1.
6. Pick $b \xleftarrow{\$} \{0,1\}$ and $r_1, r_2 \xleftarrow{\$} \mathcal{M}$.
7. Hand the $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},B}$ the following values: $m_0 = m'_b$ and $m_1 = 0$.
8. Get c^* from $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},B}$ and give (c_1^*, c_2^*) to A_2, where $c_1^* = \mathrm{IBE}_0(0)$ and $c_2^* = c^*$.
9. Answer PEKSD and $\mathrm{Trapdoor}$ queries as follows: if the query is of the form (c_1, c_2), where $c_1 = c_1^*$, then reject the query. Otherwise, if $c_2 \neq c_2^*$, answer exactly as in Step 4. Finally, if $c_2 = c_2^*$ answer as follows (here we show how to answer a decryption query; a test query is answered similarly):
   (a) Compute $T_0 = \mathrm{IBT}(MK_1, 0)$ using the extraction algorithm of the IBE scheme.
   (b) Compute $r'_1 = \mathrm{IBD}_{T_0,0}(c_1)$ using the decryption algorithm of the IBE scheme.
   (c) Compute $c'_2 = \mathrm{PKE}_{PK_2}(0; r'_1)$ using the encryption algorithm of the PKE scheme, with r'_1 as a random string. If $c'_2 = c_2^*$ answer 0. otherwise continue.
   (d) Compute $T_{m_b} = \mathrm{IBT}(MK_1, m_b)$ using the extraction algorithm of the IBE scheme.
   (e) Compute $r''_1 = \mathrm{IBD}_{T_{m_b},m_b}(c_1)$ using the decryption algorithm of the IBE scheme.
   (f) Compute $c''_2 = \mathrm{PKE}_{PK_2}(m_b; r''_1)$ using the encryption algorithm of the PKE scheme, with r''_1 as a random string. If $c''_2 = c_2^*$ answer m_b. otherwise reject the query.
10. Get the output b' from A_2.
11. Pick $b_1, b_2 \xleftarrow{\$} \{0,1\}$. If $b_1 = 0$, output $b^* = b'$, else, output $b^* = b_2$.

Note that in case the experiment $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},B}$ chose to encrypt the message $m_0 = m'_b$, then the experiment is distributed identically to the game G_3 while if it chose to encrypt the message $m_1 = 0$, then the experiment distributes identically to the game G_4. The analysis of the advantage of B is identical to the analysis from Claim 6.2. □

Acknowledgements

We would like to express our gratitude to Eike Kiltz for helpful discussions. Furthermore, we would like to thank Dario Catalano who found a glitch in a previous version of our construction.

References

[1] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 205–222, Santa Barbara, CA, USA, August 14–18, 2005. Springer-Verlag, Berlin, Germany.

[2] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany.

[3] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.

[4] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):915–942, 2006.

[5] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 290–307, Santa Barbara, CA, USA, August 20–24, 2006. Springer-Verlag, Berlin, Germany.

[6] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.

[7] Yevgeniy Dodis and Jonathan Katz. Chosen-ciphertext security of multiple encryption. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 188–209, Cambridge, MA, USA, February 10–12, 2005. Springer-Verlag, Berlin, Germany.

[8] Thomas Fuhr and Pascal Paillier. Decryptable searchable encryption. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *Provable Security, First International Conference, ProvSec 2007*, volume 4784 of *LNCS*, pages 228–236, Wollongong, Australia, November 1–2, 2007. Springer-Verlag, Berlin, Germany.

[9] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and CCA security for public key encryption. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 434–455, Amsterdam, The Netherlands, February 21–24, 2007. Springer-Verlag, Berlin, Germany.

[10] Michael Luby and Charles Rackoff. How to construct pseudo-random permutations from pseudo-random functions (abstract). In Hugh C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, page 447, Santa Barbara, CA, USA, August 18–22, 1986. Springer-Verlag, Berlin, Germany.

[11] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. Cryptology ePrint Archive, Report 2008/116, 2008. `http://eprint.iacr.org/2008/116`.

[12] Rui Zhang and Hideki Imai. Generic combination of public key encryption with keyword search and public key encryption. In Feng Bao, San Ling, Tatsuaki Okamoto, Huaxiong Wang, and Chaoping Xing, editors, *CANS 07*, volume 4856 of *LNCS*, pages 159–174, Singapore, December 8–10, 2007. Springer-Verlag, Berlin, Germany.

[13] Rui Zhang, Goichiro Hanaoka, Junji Shikata, and Hideki Imai. On the security of multiple encryption or CCA-security+CCA-security=CCA-security? In Feng Bao, Robert Deng, and Jianying Zhou, editors, *PKC 2004*, volume 2947 of *LNCS*, pages 360–374, Singapore, March 1–4, 2004. Springer-Verlag, Berlin, Germany.

A Standard definitions

A.1 Universal hashing

Definition A.1 (Family of pairwise independent hash functions)**.** *Let* $\mathcal{H} = (\mathcal{H}_k)_{k\in\mathbb{N}}$ *and* $\ell : \mathbb{N} \to \mathbb{N}$ *with* $h : \{0,1\}^k \to \{0,1\}^{\ell(k)}$ *for all* k *and* $h \in \mathcal{H}_k$*. Then* $\mathcal{H}$ *is a* family of pairwise independent hash functions *iff for all* k*, for all* $X, X' \in \{0,1\}^k$ *with* $X \neq X'$*, and for all* $Y, Y' \in \{0,1\}^{\ell(k)}$

$$\Pr_h \left[h(X) = Y \text{ and } h(X') = Y'\right] = 2^{-2\ell(k)},$$

where the probability is over a uniform choice of $h \in \mathcal{H}_k$*.*

A.2 Public key encryption

Definition A.2 (PKE scheme)**.** *A* public key encryption (PKE) scheme $\mathcal{PKE} = (\mathrm{PKG}, \mathrm{PKE}, \mathrm{PKD})$ *with* message space $\mathcal{M}$ *consists of three PPT algorithms with the following syntactics:*

Key generation: $(PK, SK) \leftarrow \mathrm{PKG}(1^k)$ *samples a keypair* (PK, SK) *consisting of a public key* PK *along with a secret key* SK*.*

Encryption: $c \leftarrow \mathrm{PKE}_{PK}(m)$ *encrypts a message* $m \in \mathcal{M}$ *and produces a ciphertext* c*.*

Decryption: $m \leftarrow \mathrm{PKD}_{SK}(c)$ *decrypts a ciphertext* c *to a message* m*.*

We require that $\mathrm{PKD}_{SK}(\mathrm{PKE}_{PK}(m)) = m$ *always, for all* $m \in \mathcal{M}$ *and all possible* $(PK, SK) \leftarrow \mathrm{PKG}(1^k)$.

Definition A.3 (IND-CCA secure PKE scheme)**.** *A PKE scheme* $\mathcal{PKE} = (\mathrm{PKG}, \mathrm{PKE}, \mathrm{PKD})$ *is called* indistinguishable under chosen-ciphertext attacks (IND-CCA secure) *iff for every pair of PPT adversaries* $A = (A_1, A_2)$*, the function*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},A}(k) = 1\right] - 1/2$$

is negligible in k*, where* $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},A}(k)$ *is the following experiment:*

$$\begin{array}{l}
\textbf{Experiment } \mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cca}}_{\mathcal{PKE},A}(k) \\
\quad (SK, PK) \leftarrow \mathrm{PKG}(1^k) \\
\quad (m_0, m_1, st) \leftarrow A_1^{\mathrm{PKD}_{SK}(\cdot)}(PK) \\
\quad b \stackrel{\$}{\leftarrow} \{0,1\} \\
\quad c^* \leftarrow \mathrm{PKE}_{PK}(m_b) \\
\quad b' \leftarrow A_2^{\mathrm{PKD}_{SK}(\cdot)}(st, c^*) \\
\quad \text{Return 1 iff } b = b'
\end{array}$$

To avoid trivialities, we require that A_1 *always returns* $m_0, m_1 \in \mathcal{M}$ *with* $|m_0| = |m_1|$*, and that* A_2 *never queries* $\mathrm{PKD}_{SK}(c^*)$*.*

A.3 Identity based encryption

Definition A.4 (IBE scheme)**.** *An* identity-based encryption (IBE) scheme $\mathcal{IBE} = (\mathrm{IBG}, \mathrm{IBT}, \mathrm{IBE}, \mathrm{IBD})$ *with* identity space $\mathcal{ID} \subseteq \{0,1\}^*$ *and* message space $\mathcal{M} \subseteq \{0,1\}^*$ *is comprised of four PPT algorithms with the following syntactics:*

Key generation: $(MK, PK) \leftarrow \mathrm{IBG}(1^k)$ *returns a* master secret key MK *along with a* public key PK.

Trapdoor generation: $T \leftarrow \mathrm{IBT}(MK, id)$ *returns a* user secret key MK *for an identity* $id \in \mathcal{ID}$.

Encryption: $c \leftarrow \mathrm{IBE}_{PK,id}(m)$ *encrypts a message* $m \in \mathcal{M}$ *under public key* PK *and identity* $id \in \mathcal{ID}$.

Decryption: $m \leftarrow \mathrm{IBD}_{T,id}(c)$ *decrypts a ciphertext* c *under identity* id *with a user secret key* T.

Occasionally, we will write $c \leftarrow \mathrm{IBD}_{MK,id}(c)$ *as a shorthand for executing first* $T \leftarrow \mathrm{IBT}(MK, id)$ *and then* $c \leftarrow \mathrm{IBD}_{T,id}(c)$. *We require that for all* $id \in \mathcal{ID}$ *and* $m \in \mathcal{M}$, *we always have* $m \leftarrow \mathrm{IBD}_{MK,id}(\mathrm{IBE}_{PK,id}(m))$ *for all possible* $(MK, PK) \leftarrow \mathrm{IBG}(1^k)$. *As a technicality, we also require that algorithm* IBT *is deterministic (this is without loss of generality, cf. Boneh et al. [4]).*

Definition A.5 (IND-CCA secure IBE scheme)**.** *An IBE scheme* $\mathcal{IBE} = (\mathrm{IBG}, \mathrm{IBT}, \mathrm{IBE}, \mathrm{IBD})$ *is called* indistinguishable under chosen-ciphertext attacks (IBE-IND-CCA secure) *iff for every pair of PPT adversaries* $A = (A_1, A_2)$, *the function*

$$\mathsf{Adv}^{\text{ibe-ind-cca}}_{\mathcal{IBE},A}(k) := \Pr\left[\mathsf{Exp}^{\text{ibe-ind-cca}}_{\mathcal{IBE},A}(k) = 1\right] - 1/2$$

is negligible in k, where $\mathsf{Exp}^{\text{ibe-ind-cca}}_{\mathcal{IBE},A}(k)$ *is the following experiment:*

$$\begin{array}{l}
\textbf{Experiment } \mathsf{Exp}^{\text{ibe-ind-cca}}_{\mathcal{IBE},A}(k) \\
\quad (MK, PK) \leftarrow \mathrm{IBG}(1^k) \\
\quad (id^*, m_0, m_1, st) \leftarrow A_1^{\mathrm{IBT}(MK,\cdot),\mathrm{IBD}_{MK,\cdot}(\cdot)}(PK) \\
\quad b \stackrel{\$}{\leftarrow} \{0,1\} \\
\quad c^* \leftarrow \mathrm{IBE}_{PK,id^*}(m_b) \\
\quad b' \leftarrow A_2^{\mathrm{IBT}(MK,\cdot),\mathrm{IBD}_{MK,\cdot}(\cdot)}(st, c^*) \\
\quad \text{Return 1 iff } b = b'
\end{array}$$

To avoid trivialities, we require that A_1 *always returns* $m_0, m_1 \in \mathcal{M}$ *with* $|m_0| = |m_1|$, *that* A_1 *never returns an* id^* *on which* $\mathrm{IBT}(MK, \cdot)$ *has been queried, and that* A_2 *never queries* $\mathrm{IBT}(MK, id^*)$ *and* $\mathrm{IBD}_{MK,id^*}(c^*)$.

Definition A.6 (Anonymous IBE scheme)**.** *An IBE scheme* $\mathcal{IBE} = (\mathrm{IBG}, \mathrm{IBT}, \mathrm{IBE}, \mathrm{IBD})$ *is called* anonymous (IBE-ANO-CCA secure) *iff for every pair of PPT adversaries* $A = (A_1, A_2)$, *the function*

$$\mathsf{Adv}^{\text{ibe-ano-cca}}_{\mathcal{IBE},A}(k) := \Pr\left[\mathsf{Exp}^{\text{ibe-ano-cca}}_{\mathcal{IBE},A}(k) = 1\right] - 1/2$$

is negligible in k, where $\mathsf{Exp}^{\text{ibe-ano-cca}}_{\mathcal{IBE},A}(k)$ *is the following experiment:*

$$\begin{array}{l}
\textbf{Experiment } \mathsf{Exp}^{\text{ibe-ano-cca}}_{\mathcal{IBE},A}(k) \\
\quad (MK, PK) \leftarrow \mathrm{IBG}(1^k) \\
\quad (id_0, id_1, m, st) \leftarrow A_1^{\mathrm{IBT}(MK,\cdot),\mathrm{IBD}_{MK,\cdot}(\cdot)}(PK) \\
\quad b \stackrel{\$}{\leftarrow} \{0,1\} \\
\quad c^* \leftarrow \mathrm{IBE}_{PK,id_b}(m) \\
\quad b' \leftarrow A_2(st, c^*) \\
\quad \text{Return 1 iff } b = b'
\end{array}$$

To avoid trivialities, we require that A_1 *always returns* id_0, id_1 *with* $|id_0| = |id_1|$, *that* A_1 *never returns an* id_0 *or an* id_1 *on which* $\mathrm{IBT}(MK, \cdot)$ *has been queried, and that* A_2 *never queries* $\mathrm{IBT}(MK, id_i)$ *and* $\mathrm{IBD}_{MK,id_i}(c^*)$ *for* $i \in \{0, 1\}$.

B Postponed proofs

Proof of Lemma 4.3. Given an arbitrary IBE-IND-CCA adversary $A = (A_1, A_2)$ on $\mathcal{IBE}$, we construct an IBE-IND-CCA adversary $A' = (A'_1, A'_2)$ on $\mathcal{IBE}'$. $A'^{\,\mathrm{IBT}'(MK',\cdot),\mathrm{IBD}'_{MK',\cdot}(\cdot)}_1(PK')$ samples $h \stackrel{\$}{\leftarrow} \mathcal{H}_k$, sets $PK := (PK', h)$, and runs $(id^*, m_0, m_1, st) \leftarrow A_1^{\mathrm{IBT}(MK,\cdot),\mathrm{IBD}_{MK,\cdot}(\cdot)}(PK)$. Here, the trapdoor oracle $\mathrm{IBT}(MK, id)$ returns $(\mathrm{IBT}'(MK', id), h)$, and the decryption oracle $\mathrm{IBD}_{MK,id}(c)$ is implemented as follows: run $m' \leftarrow \mathrm{IBD}'_{MK',id}(c)$, parse $m' = (Y, m)$, and return m if $h(id) = Y$ (and $\perp$ otherwise). Next, A'_1 returns (id^*, m'_0, m'_1, st), where $m'_i = (h(id^*), m_i)$ for $i \in \{0, 1\}$. Finally, A'_2 runs A_2 with oracles IBT and IBD implemented as above, and outputs A_2's output. This perfectly emulates the IBE-IND-CCA experiment with $\mathcal{IBE}$ for A, and we have

$$\mathsf{Adv}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE},A}(k) = \mathsf{Adv}^{\mathsf{ibe\text{-}ind\text{-}cca}}_{\mathcal{IBE}',A'}(k),$$

which shows the lemma. $\square$