Circular chosen-ciphertext security with compact ciphertexts

Dennis Hofheinz

June 11, 2012

Abstract

A key-dependent message (KDM) secure encryption scheme is secure even if an adversary obtains encryptions of messages that depend on the secret key. Such key-dependent encryptions naturally occur in scenarios such as harddisk encryption, formal cryptography, or in specific protocols. However, there are not many provably secure constructions of KDM-secure encryption schemes. Moreover, only one construction, due to Camenisch, Chandran, and Shoup (Eurocrypt 2009) is known to be secure against active (i.e., CCA) attacks.

In this work, we construct the first public-key encryption scheme that is KDM-secure against active adversaries and has compact ciphertexts. As usual, we allow only circular key dependencies, meaning that encryptions of arbitrary *entire* secret keys under arbitrary public keys are considered in a multi-user setting.

Technically, we follow the approach of Boneh, Halevi, Hamburg, and Ostrovsky (Crypto 2008) to KDM security, which however only achieves security against passive adversaries. We explain an inherent problem in adapting their techniques to active security, and resolve this problem using a new technical tool called "lossy algebraic filters" (LAFs). We stress that we significantly deviate from the approach of Camenisch, Chandran, and Shoup to obtain KDM security against active adversaries. This allows us to develop a scheme with compact ciphertexts that consist only of a constant number of group elements.

Keywords: key-dependent messages, chosen-ciphertext security, public-key encryption.

1 Introduction

KDM security. An encryption scheme is key-dependent message (KDM) secure if it is secure even against an adversary who has access to encryptions of messages that depend on the secret key. Such a setting arises, e.g., in harddisk encryption [10], computational soundness results in formal methods [6, 2], or specific protocols [13]. KDM security does not follow from standard security [1, 15], and there are indications [19, 5] that KDM security (at least in its most general form) cannot be proven using standard techniques; it seems that dedicated constructions and proof techniques are necessary.[1]

The BHHO approach to KDM-CPA security. Boneh, Halevi, Hamburg, and Ostrovsky [10] (henceforth BHHO) were the first to construct and prove a public-key encryption (PKE) scheme that is KDM secure under chosen-plaintext attacks (KDM-CPA-secure) in the standard model, under the Decisional Diffie-Hellman (DDH) assumption. While they did not prove their scheme secure under messages that *arbitrarily* depend on the secret key, their result encompasses the important case of *circular (CIRC-CPA) security*. Loosely speaking, a PKE scheme is circular secure if it is secure even in a multi-user setting where encryptions of arbitrary secret keys under arbitrary public keys are known. This notion is sufficient for certain applications [13], and can often be extended to stronger forms of KDM security [5, 12]. Inspired by BHHO, KDM-CPA-secure PKE schemes from other computational assumptions followed [4, 11, 22].

Since we will be using a similar approach, we give a high-level intuition of BHHO's approach. The crucial property of their scheme is that it is *publicly* possible to construct encryptions of the

[1] We mention, however, that semi-generic transformations exist that *enhance* the KDM security of an already "slightly" KDM-secure scheme [5, 12, 3].

secret key (under the corresponding public key). Thus, encryptions of the secret key itself do not harm the (IND-CPA) security of that scheme. Suitable homomorphic properties of both keys and ciphertexts allow to extend this argument to circular security (for arbitrarily many users/keys), and to affine functions of all keys.

Why the BHHO approach fails to achieve KDM-CCA security. When considering an *active* adversary, we require a stronger form of KDM security. Namely, KDM-CCA, resp. CIRC-CCA security requires security against an adversary who has access to key-dependent encryptions *and* a decryption oracle. (Naturally, to avoid a trivial notion, the adversary is not allow to submit any of those given KDM encryptions to its decryption oracle.) Now if we want to extend BHHO's KDM-CPA approach to an adversary with a decryption oracle, the following problem arises: since it is publicly possible to construct (fresh) encryptions of the secret key, an adversary can generate such an encryption and then submit it to its decryption oracle, thus obtaining the full secret key. Hence, the very property that BHHO use to prove KDM-CPA security seemingly contradicts chosen-ciphertext security.

Our technical tool: lossy algebraic filters (LAFs). Before we describe our approach to KDM-CCA security, let us present the core technical tool we use. Namely, a *lossy algebraic filter* (LAF) is a family of functions, indexed by a public key and a tag. A function from that family takes a vector $X = (X_i)_{i=1}^{\mathtt{n}}$ as input. Now if the tag is *lossy*, then the output of the function reveals only a linear combination of the X_i. If the tag is *injective*, however, then so is the function. We require that there are many lossy tags, which however require a special trapdoor to be found. On the other hand, lossy and injective tags are computationally indistinguishable. This concept is very similar to (parameterized) lossy trapdoor functions [24], and in particular to all-but-many lossy trapdoor functions (ABM-LTFs [20]). In our setting, we do not require efficient inversion, but we do require that lossy functions always reveal *the same* linear combination about the input. In particular, evaluating the same input under many lossy tags will still leave the input (partially) undetermined.

We give a construction of LAFs under the Decision Linear (DLIN) assumption in pairing-friendly groups. Similar to ABM-LTFs, lossy tags correspond to suitably blinded signatures. (This in particular allows to release many lossy tags, while still making the generation of a fresh lossy tag hard for an adversary.) However, unlike with ABM-LTFs, functions with lossy tags always release the same information about its input. Our construction has compact tags with $\mathbf{O}(1)$ group elements, which will be crucial for our KDM-CCA secure encryption scheme.

Our approach to KDM-CCA security. We can now describe our solution to the KDM-CCA dilemma explained above. We will start from a hybrid between the BHHO-like PKE schemes of Brakerski and Goldwasser [11], resp. Malkin et al. [22]. This scheme has compact ciphertexts ($\mathbf{O}(1)$ group elements), and its KDM-CPA security can be proved under the Decisional Composite Residuosity (DCR) assumption. As with the BHHO scheme, the scheme's KDM-CPA security relies on the fact that encryptions of its secret key can be publicly generated. Essentially, our modification consists of adding a suitable authentication tag to each ciphertext. This authentication tag comprises the (encrypted) image of the plaintext message under an LAF. During decryption, a ciphertext is rejected in case of a wrong authentication tag.

In our security proof, all authentication tags for the key-dependent encryptions the adversary gets are made with respect to lossy filter tags. This means that information-theoretically, little information about the secret key is released (even with many key-dependent encryptions, resp. LAF evaluations). However, any decryption query the adversary makes must refer (by the LAF properties) to an injective tag. Hence, in order to place a valid key-dependent decryption query, the adversary would have to correctly guess the whole secret key (which is hidden).

Thus, in a nutshell, adding a suitable authentication tag allows to leverage the techniques by BHHO, resp. Brakerski and Goldwasser, Malkin et al. to chosen-ciphertext attacks. In particular, we obtain a CIRC-CCA-secure PKE scheme with compact ciphertexts (of $\mathbf{O}(1)$ group elements). We prove security under the conjunction of the following assumptions: the DCR assumption (in $\mathbb{Z}_{N^3}^*$), the DLIN assumption (in a pairing-friendly group), and the DDH assumption (somewhat curiously, in the subgroup of order $(P-1)(Q-1)/4$ of $\mathbb{Z}^*_{N^3}$, where $N = PQ$).[2]

Relation to Camenisch et al.'s CIRC-CCA-secure scheme. Camenisch, Chandran, and Shoup [14] present the only other known CIRC-CCA-secure PKE scheme in the standard model. They also build upon BHHO techniques, but instead use a Naor-Yung-style double encryption technique [23] to achieve chosen-ciphertext security. As an authentication tag, they attach to each ciphertext a non-interactive zero-knowledge proof that *either* the encryption is consistent (in the usual Naor-Yung sense), *or* that they know a signature for the ciphertext. Since they build on the original, DDH-based BHHO scheme, they can use Groth-Sahai proofs [18] to prove consistency. Compared to our scheme, their system is less efficient: they require $\mathbf{O}(k)$ group elements per ciphertext, and the secret key can only be encrypted bitwise. However, their sole computational assumption to prove circular security is the DLIN assumption in pairing-friendly groups. One interesting thing to point out is their implicit use of a (one-time) signature scheme. Their argument is conceptually not unlike our LAF argument. However, since they can apply a hybrid argument to substitute all key-dependent encryptions with random ciphertexts, they only require one-time signatures. Furthermore, the meaning of "consistent ciphertext" and "proof" in our case is technically very different. (Unlike Camenisch et al., we apply an argument that rests on the *information* that the adversary has at a certain point about the secret key.)

Note about concurrent work. In a work concurrent to ours, Galindo, Herranz, and Villar [17] define and instantiate a strong notion of KDM security for identity-based encryption (IBE) schemes. Using the IBE→PKE transformation of Boneh, Canetti, Halevi, and Katz [9], they derive a KDM-CCA-secure PKE scheme. Their concrete construction is entropy-based and achieves only a bounded form of KDM security, much like the KDM-secure SKE scheme from [21]. Thus, while their ciphertexts are very compact, they can only tolerate a number of (arbitrary) KDM queries that is linear in the size of the secret key. In particular, it is not clear how to argue that the encryption of a full secret key in their scheme is secure.

2 Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. Throughout the paper, $k \in \mathbb{N}$ denotes the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For a probabilistic algorithm A, we denote $y \leftarrow A(x; R)$ the process of running A on input x and with randomness R, and assigning y the result. We write $y \leftarrow A(x)$ for $y \leftarrow A(x; R)$ with uniformly chosen R. If A's running time is polynomial in k, then A is called probabilistic polynomial-time (PPT).

DCR assumption. The Decisional Composite Residuosity (DCR) assumption over a group $\mathbb{Z}^*_{N^{s+1}}$ (for $N = PQ$ with primes P, Q, and $s \geq 1$) states that for every PPT adversary A,

$$\mathsf{Adv}^{\mathsf{dcr}}_{\mathbb{Z}^*_{N^{s+1}},A}(k) \;:=\; \Pr\left[A(N, Z) = 1\right] - \Pr\left[A(N, Z^{N^s}) = 1\right],$$

is negligible, where $Z \leftarrow \mathbb{Z}^*_{N^{s+1}}$ is uniformly chosen. Damgård and Jurik [16] have showed that the DCR assumptions over $\mathbb{Z}^*_{N^{s+1}}$ and $\mathbb{Z}^*_{N^{s'}+1}$ are equivalent for any s, s'.

DDH and DLIN assumptions. The Decisional Diffie-Hellman (DDH), resp. Decision Linear [7] (DLIN) assumptions over a group $\mathbb{G}$ of (not necessarily prime) order q state that for every PPT adversary A, the respective following functions are negligible:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G},A}(k) &:= \Pr\left[A(g, g^x, g^y, g^{xy}) = 1\right] - \Pr\left[A(g, g^x, g^y, g^z) = 1\right], \\
\mathsf{Adv}^{\mathsf{dlin}}_{\mathbb{G},A}(k) &:= \Pr\left[A(g, U_1, U_2, g^{s_0}, U_1^{s_1}, U_2^{s_0+s_1}) = 1\right] - \Pr\left[A(g, U_1, U_2, g^{s_0}, U_1^{s_1}, U_2^{s_2}) = 1\right],
\end{aligned}$$

[2] Very roughly, we resort to the DDH assumption since we release *partial* information about our secret keys. Whereas the argument of [11, 22] relies on the fact that the secret key sk is completely hidden modulo a certain N, where $\mathbb{Z}_N$ is message space, we cannot avoid to leak some information modulo about sk mod N by releasing LAF images of sk. However, using a suitable encoding of messages, we *can* argue that sk is completely hidden modulo the coprime modulus $(P-1)(Q-1)/4$, which enables a reduction to the DDH assumption.

where g is a uniform generator of $\mathbb{G}$, and $U_1, U_2 \leftarrow \mathbb{G}$ and $x, y, z, s_0, s_1, s_2 \leftarrow \mathbb{Z}_q$ are uniform.

Pairings. A (symmetric) pairing is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ between two cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ that satisfies $e(g, g) \neq 1$ and $e(g^a, g^b) = e(g, g)^{ab}$ for all generators g of $\mathbb{G}$ and all $a, b \in \mathbb{Z}$.

PKE schemes. A public-key encryption (PKE) scheme PKE consists of four[3] PPT algorithms $(\mathsf{Pars}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The parameter generator $\mathsf{Pars}(1^k)$ outputs public parameters pp such as a group description. Key generation $\mathsf{Gen}(pp)$ outputs a public key pk and a secret key sk. Encryption $\mathsf{Enc}(pp, pk, M)$ takes parameters pp, a public key pk, and a message M, and outputs a ciphertext C. Decryption $\mathsf{Dec}(pp, sk, C)$ takes public parameters pp, a secret key sk, and a ciphertext C, and outputs a message M. For correctness, we want $\mathsf{Dec}(pp, sk, C) = M$ for all M, all $pp \leftarrow \mathsf{Pars}(1^k)$, all $(pk, sk) \leftarrow \mathsf{Gen}(pp)$, and all $C \leftarrow \mathsf{Enc}(pk, M)$.

Key-unique SKE schemes. A secret-key encryption (SKE) scheme (E, D) consists of two PPT algorithms. Encryption $\mathsf{E}(K, M)$ takes a key K and a message M, and outputs a ciphertext C. Decryption $\mathsf{D}(K, C)$ takes a key K and a ciphertext C, and outputs a message M. For correctness, we want $\mathsf{Dec}(K, C) = M$ for all M, all K, and all $C \leftarrow \mathsf{E}(K, M)$. We say that (E, D) is *key-unique* if for every ciphertext C, there is at most one key K with $\mathsf{D}(K, C) \neq \perp$. For instance, ElGamal encryption can be interpreted as a key-unique SKE scheme through $\mathsf{E}(x, M) := (g^x, g^y, g^{xy} \cdot M)$ (and the obvious D). This example assumes a publicly known group $\mathbb{G} = \langle g \rangle$ in which the DDH assumption holds.[4] If a larger message space (e.g., $\{0, 1\}^*$) is desired, hybrid encryption techniques (which preserve key-uniqueness) can be employed.

IND-CPA security. An SKE scheme is IND-CPA secure iff no efficient adversary A wins the following game with probability non-negligibly away from $1/2$. First, A selects two equal-length messages M_0, M_1, then gets an encryption $\mathsf{E}(K, M_b)$ (for random K and $b \leftarrow \{0, 1\}\}$, and then takes a guess $b' \in \{0, 1\}$. During this, A gets access to an encryption oracle $\mathsf{E}(K, \cdot)$. We say that A wins iff $b = b'$. For concrete security analyses, let $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{(\mathsf{E},\mathsf{D}),A}(k)$ denote the probability that A wins this game. This definition can be adapted to the PKE setting by initially giving A the public key pk instead of access to an encryption oracle.

Signature schemes. A signature scheme Sig consists of three PPT algorithms $(\mathsf{SGen}, \mathsf{Sig}, \mathsf{Ver})$. Key generation $\mathsf{SGen}(1^k)$ outputs a verification key $verk$ and a signing key $sigk$. The signature algorithm $\mathsf{Sig}(sigk, M)$ takes a signing key $sigk$ and a message M and outputs a signature σ. Verification $\mathsf{Ver}(verk, M, \sigma)$ takes a verification key $verk$, a message M and a potential signature σ and outputs a verdict $b \in \{0, 1\}$. For correctness, we require that $\mathsf{Ver}(verk, M, \sigma) = 1$ for all M, all $(verk, sigk) \leftarrow \mathsf{SGen}(1^k)$, and all $\sigma \leftarrow \mathsf{Sig}(sigk, M)$.

(One-time, strong) existential unforgeability. A signature scheme Sig is existentially unforgeable (EUF-CMA secure) iff no PPT forger F wins the following game with non-negligible probability. First, F gets a verification key $verk$ as well as access to a signature oracle $\mathsf{Sig}(sigk, \cdot)$. A wins iff it finally outputs a valid signature σ for a fresh message M that has not yet been queried to $\mathsf{Sig}(sigk, \cdot)$. Let $\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{Sig},A}(k)$ denote the probability that A wins this game. Sig is called one-time existentially unforgeable (OT-EUF-CMA secure) iff no PPT forger F that makes at most one signature query wins the above game with non-negligible probability. Finally, Sig is strongly (OT-)EUF-CMA secure iff it is (OT-)EUF-CMA secure as above, but in a game in which an adversary already wins already if it generates a fresh signature for a (perhaps already signed) message. We let $\mathsf{Adv}^{\mathsf{seuf\text{-}cma}}_{\mathsf{Sig},A}(k)$ denote the probability that A wins this strong EUF-CMA security game.

Waters signatures. In [25], Waters proves the following signature scheme EUF-CMA secure:[5]

- $\mathsf{Gen}(1^k)$ chooses groups $\mathbb{G}, \mathbb{G}_T$ of prime order p, along with a pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, a generator $g \in \mathbb{G}$, and uniform group elements $g^\omega, H_0, \ldots, H_k \in \mathbb{G}$. Output is

$$verk = (\mathbb{G}, \mathbb{G}_T, e, p, g, (H_i)_{i=0}^k, e(g, g)^\omega), \qquad sigk = (verk, g^\omega).$$

[3] We will only use public parameters for PKE schemes, but not, e.g., for signature schemes.

[4] In view of our application, $\mathbb{G}$ can be part of the public parameters of our KDM-secure PKE scheme.

[5] In fact, our description is a slight folklore optimization of Waters [25]. The original scheme features elements g^α, g^β in $verk$, so that $e(g^\alpha, g^\beta)$ takes the role of $e(g, g)^\omega$.

- $\mathsf{Sig}(sigk, M)$, for $M = (M_i)_{i=1}^k \in \{0,1\}^k$, picks $r \leftarrow \mathbb{Z}_p$, and lets $\sigma := (g^r, g^\omega \cdot (H_0 \prod_{i=1}^k H_i^{M_i})^r)$.
- $\mathsf{Ver}(verk, M, \sigma)$, for $\sigma = (\sigma_0, \sigma_1)$, outputs 1 iff $e(g, \sigma_1) = e(g,g)^\omega \cdot e(\sigma_0, H_0 \prod_{i=1}^k H_i^{M_i})$.

KDM-CCA and CIRC-CCA security. Let $n = n(k)$ and let PKE be a PKE scheme with message space $\mathcal{M}$. PKE is chosen-ciphertext secure under key-dependent message attacks (n-KDM-CCA secure) iff

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

is negligible for all PPT A, where experiment $\mathsf{Exp}^{\mathsf{kdm\text{-}cca}}_{\mathsf{PKE},n,A}$s defined as follows. First, the experiment tosses a coin $b \leftarrow \{0,1\}$, and samples public parameters $pp \leftarrow \mathsf{Pars}(1^k)$ and n keypairs $(pk_i, sk_i) \leftarrow \mathsf{Gen}(pp)$. Then A is invoked with input pp and $(pk_i)_{i=1}^n$, and access to two oracles:

- a KDM oracle $\mathcal{KDM}_b(\cdot,\cdot)$ that maps $i \in [n]$ and a function $f : (\{0,1\}^*)^n \to \{0,1\}^*$ to a ciphertext $C \leftarrow \mathsf{Enc}(pp, pk_i, M)$. If $b = 0$, then $M = f((sk_i)_{i=1}^n)$; else, $M = 0^{|f((sk_i)_{i=1}^n)|}$.
- a decryption oracle $\mathcal{DEC}(\cdot,\cdot)$ that takes as input an index $i \in [n]$ and a ciphertext C, and outputs $\mathsf{Dec}(pp, sk_i, C)$.

When A finally generates an output $b' \in \{0,1\}$, the experiment outputs 1 if $b = b'$ (and 0 else). We require that (a) A never inputs a ciphertext C to $\mathcal{DEC}$ that has been produced by $\mathcal{KDM}_b$ (for the same index i), and (b) A only specifies PPT-computable functions f that always output messages of the same length. As a relevant special case, PKE is *n-CIRC-CCA-secure* if it is n-KDM-CCA secure against all A that only query $\mathcal{KDM}_b$ with functions $f \in \mathcal{F}$ for

$$\mathcal{F} \;:=\; \{f_j \;:\; f_j((sk_i)_{i=1}^n) = sk_j\}_{j\in[n]} \cup \{f_M \;:\; f_M((sk_i)_{i=1}^n) = M\}_{M\in\mathcal{M}}\,.$$

(Technically, what we call "circular security" is called "clique security" in [10]. We stress, however, that our notion of circular security implies that of [10].) Our main result will be a PKE scheme that is n-CIRC-CCA-secure for all polynomials $n = n(k)$.

3 Lossy algebraic filters

3.1 Definition

Informal description. Informally, an $(\ell_{\mathsf{LAF}}, \mathfrak{n})$-lossy algebraic filter (LAF) is a family of functions indexed by a public key Fpk and additionally by a tag t. A function $\mathsf{LAF}_{Fpk,t}$ from the family maps an input $X = (X_i)_{i=1}^{\mathfrak{n}} \in \mathbb{Z}_p^{\mathfrak{n}}$ to an output $\mathsf{LAF}_{Fpk,t}(X)$, where p is an ℓ_{LAF}-bit prime contained in the public key.

The crucial property of an LAF is its lossiness. Namely, for a given public key Fpk, we distinguish *injective* and *lossy* tags.[6] For an injective tag t, the function $\mathsf{LAF}_{Fpk,t}(\cdot)$ is injective, and thus has an image of size $p^{\mathfrak{n}}$. However, if t is lossy, then $\mathsf{LAF}_{Fpk,t}(\cdot)$ only depends on a linear combination $\sum_{i=1}^{\mathfrak{n}} \omega_i X_i \bmod p$ of its input. In particular, different X with the same value $\sum_{i=1}^{\mathfrak{n}} \omega_i X_i \bmod p$ are mapped to the same image. Here, the coefficients $\omega_i \in \mathbb{Z}_p$ only depend on Fpk (but not on t). For a lossy tag t, the image of $\mathsf{LAF}_{Fpk,t}(\cdot)$ is thus of size at most p. Note that the modulus p is public, while the coefficients ω_i may be (and in fact will have to be) computationally hidden.

For this concept to be useful, we require that (a) lossy and injective tags are computationally indistinguishable, (b) lossy tags can be generated using a special trapdoor, but (c) new lossy (or, rather, non-injective) tags cannot be found efficiently without that trapdoor, even when having seen polynomially many lossy tags before.

For technical reasons, and in view of our application, we will work with structured tags: each tag $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ consists of a *core tag* t_{c} and an *auxiliary tag* t_{a}. In our application, the auxiliary tag will be a ciphertext part that is authenticated by a filter image.

Definition 3.1 (LAF)**.** *An* $(\ell_{\mathsf{LAF}}, \mathfrak{n})$*-lossy algebraic filter (LAF)* LAF *consists of three PPT algorithms:*

[6] Technically, there may also be tags that are neither injective nor lossy.

Key generation. $\mathsf{FGen}(1^k)$ *samples a keypair* (Fpk, Ftd). *Fpk is the public key and contains an* ℓ_{LAF}*-bit prime* p *and the description of a tag space* $\mathcal{T} = \mathcal{T}_{\mathsf{c}} \times \{0,1\}^*$, *where* $\mathcal{T}_{\mathsf{c}}$ *is efficiently samplable. Each tag* $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ *consists of a core tag* $t_{\mathsf{c}} \in \mathcal{T}_{\mathsf{c}}$ *and an auxiliary tag* $t_{\mathsf{a}} \in \{0,1\}^*$. *A tag may be either injective, or lossy, or neither. Ftd is the trapdoor (to Fpk) that will allow to sample lossy tags.*

Evaluation. $\mathsf{FEval}(Fpk, t, X)$, *for a public key Fpk and a tag* $t = (t_{\mathsf{c}}, t_{\mathsf{a}}) \in \mathcal{T}$, *maps an input* $X = (X_i)_{i=1}^{\mathfrak{n}}) \in \mathbb{Z}_p^{\mathfrak{n}}$ *to a unique output* $\mathsf{LAF}_{Fpk,t}(X)$.

Lossy tag generation. $\mathsf{FTag}(Ftd, t_{\mathsf{a}})$, *for a trapdoor Ftd and* $t_{\mathsf{a}} \in \{0,1\}^*$, *samples a core tag* t_{c} *such that* $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ *is lossy.*

We require the following:

Lossiness. *The function* $\mathsf{LAF}_{Fpk,t}(\cdot)$ *is injective if* t *is injective. If* t *is lossy, then* $\mathsf{LAF}_{Fpk,t}(X)$ *depends only on* $\sum_{i=1}^{\mathfrak{n}} \omega_i X_i \bmod p$ *for* $\omega_i \in \mathbb{Z}_p$ *that only depend on Fpk.*

Indistinguishability. *Lossy tags are indistinguishable from random tags. Formally,*

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},A}(k) := \Pr\left[A(1^k, Fpk)^{\mathsf{FTag}(Ftd,\cdot)} = 1\right] - \Pr\left[A(1^k, Fpk)^{\mathcal{O}_{\mathcal{T}_{\mathsf{c}}}(\cdot)} = 1\right]$$

is negligible for all PPT A, where $(Fpk, Ftd) \leftarrow \mathsf{FGen}(1^k)$, *and* $\mathcal{O}_{\mathcal{T}_{\mathsf{c}}}(\cdot)$ *is the oracle that ignores its input and samples a random core tag* t_{c}.

Evasiveness. *Non-injective (and in particular lossy) tags are hard to find, even given multiple lossy tags:*

$$\mathsf{Adv}^{\mathsf{eva}}_{\mathsf{LAF},A}(k) := \Pr\left[t \text{ non-injective} \;\middle|\; t \leftarrow A(1^k, Fpk)^{\mathsf{FTag}(Ftd,\cdot)}\right]$$

is negligible with $(Fpk, Ftd) \leftarrow \mathsf{FGen}(1^k)$, *and for any PPT algorithm A that never outputs a tag obtained through oracle queries (i.e., A never outputs* $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ *when* t_{c} *has been obtained by an oracle query* $\mathsf{FTag}(Ftd, t_{\mathsf{a}})$*).*

3.2 Construction

Intuition. We present a construction based on the DLIN problem in a group $\mathbb{G}$ of order p with symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Essentially, each tag corresponds to $\mathfrak{n}$ DLIN-encrypted Waters signatures. If the signatures are valid, then the tag is lossy. The actual filter maps an input $X = (X_i)_{i=1}^{\mathfrak{n}} \in \mathbb{Z}_p^{\mathfrak{n}}$ to the tuple

$$\mathsf{LAF}_{Fpk,t}(X) := Z \circ X := (\prod_{j=1}^{\mathfrak{n}} Z_{i,j}^{X_j})_{j=1}^{\mathfrak{n}} \in \mathbb{G}_T^{\mathfrak{n}}, \tag{1}$$

where the matrix $Z = (Z_{i,j})_{i,j \in [\mathfrak{n}]} \in \mathbb{G}_T^{\mathfrak{n}\times\mathfrak{n}}$ is computed from public key and tag. Note that this mapping is lossy if and only if the matrix

$$\widetilde{Z} := (\widetilde{Z}_{i,j}) := (\mathrm{dlog}_{e(g,g)}(Z_{i,j}))_{i,j} \in \mathbb{Z}_p^{\mathfrak{n}\times\mathfrak{n}} \tag{2}$$

of discrete logarithms (to some arbitrary basis $e(g,g) \in \mathbb{G}_T$) is non-invertible.

For a formal description, let $\ell_{\mathsf{LAF}}(k), \mathfrak{n}(k)$ be two functions.

Key generation. $\mathsf{FGen}(1^k)$ generates cyclic groups $\mathbb{G}, \mathbb{G}_T$ of prime order p (where p is of bitlength $\lfloor \log_2(p) \rfloor = \ell_{\mathsf{LAF}}(k)$), and a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Then FGen chooses

- a generator $g \in \mathbb{G}$,
- a uniform exponent $\omega \leftarrow \mathbb{Z}_p$,
- uniform group elements $U_1, \ldots, U_{\mathfrak{n}} \leftarrow \mathbb{G}$, $H_0, \ldots, H_k \leftarrow \mathbb{G}$, and
- a keypair (Hpk, Htd) for a chameleon hash function $\mathsf{CH} : \{0,1\}^* \to \{0,1\}^k$.

FGen finally outputs

$$\begin{aligned} Fpk &:= (\mathbb{G}, \mathbb{G}_T, e, p, g, (H_i)_{i=0}^k, (U_i)_{i=1}^{\mathfrak{n}}, W := e(g,g)^{\omega}, Hpk) \\ Ftd &:= (Fpk, g^{\omega}, Htd). \end{aligned}$$

For convenience, write $U_i = g^{u_i}$ for suitable exponents u_i.

Tags. (Core) tags are of the form

$$t_{\mathsf{c}} := (R, S_0, (S_{i,j})_{i,j=1}^{\mathfrak{n}}, R_{\mathsf{CH}}) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}^{\mathfrak{n}\times\mathfrak{n}} \times \mathcal{R}_{\mathsf{CH}},$$

where we require $e(U_{j'}, S_{i,j}) = e(U_j, S_{i,j'})$ whenever $i \notin \{j, j'\}$. This means we can write

$$R = g^r, \qquad S_0 = g^{s_0}, \qquad S_{i,j} = U_j^{s_i} \quad (i \neq j)$$

for suitable r, s_i. To a tag $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ (with auxiliary part $t_{\mathsf{a}} \in \{0,1\}^*$), we associate the matrix $Z = (Z_{i,j})_{i,j=1}^{\mathfrak{n}} \in \mathbb{G}_T^{\mathfrak{n}\times\mathfrak{n}}$ with

$$\begin{aligned} Z_{i,j} &= e(U_j, S_0) \cdot e(g, S_{i,j}) = e(g,g)^{u_j(s_0+s_i)} \qquad (i \neq j) \\ Z_{i,i} &= \frac{e(g, S_{i,i})}{W \cdot e(H_0 \prod_{i=1}^k H_i^{h_i}, R)} \end{aligned} \tag{3}$$

for $(h_i)_{i=1}^k := \mathsf{CH}_{Hpk}(R, S_0, (S_{i,j})_{i,j=1}^{\mathfrak{n}}, t_{\mathsf{a}}; R_{\mathsf{CH}})$. If the matrix $\widetilde{Z}$ of discrete logarithms (see (2)) is invertible, we say that t is injective; if $\widetilde{Z}$ has rank 1, then t is lossy. Note that for lossy tags, thus $Z_{i,j} = e(g,g)^{u_j(s_0+s_i)}$ for *all* i, j.

Evaluation. $\mathsf{FEval}(Fpk, t, X)$, for $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$, $t_{\mathsf{a}} \in \{0,1\}^*$, $X = (X_i)_{i=1}^{\mathfrak{n}} \in \mathbb{Z}_p^{\mathfrak{n}}$, and Fpk and t_{c} as above, computes Z as in (3) and then $(Y_i)_{i=1}^{\mathfrak{n}} := \mathsf{LAF}_{Fpk,t}(X) \in \mathbb{G}_T^{\mathfrak{n}}$ as in (1).

Lossiness. If we write $Y_i = e(g,g)^{y_i}$, the definition of FEval implies $(y_i)_{i=1}^{\mathfrak{n}} = \widetilde{Z} \cdot X$. Since injective tags satisfy that $\widetilde{Z}$ is invertible, they lead to injective functions $\mathsf{LAF}_{Fpk,t}(\cdot)$. On the other hand, for a lossy tag, $\widetilde{Z}_{i,j} = u_j(s_0 + s_i)$, so that

$$y_i = \sum_{j=1}^{\mathfrak{n}} u_j(s_0+s_i)X_j = (s_0+s_i) \cdot \sum_{j=1}^{\mathfrak{n}} u_j X_j \mod p.$$

Specifically, $\mathsf{LAF}_{Fpk,t}(X)$ depends only on $\sum_i \omega_i X_i \bmod p$ for $\omega_i := u_i$.

Lossy tag generation. $\mathsf{FTag}(Ftd, t_{\mathsf{a}})$, for Ftd as above and $t_{\mathsf{a}} \in \{0,1\}^*$, first chooses a random CH-image $h = (h_i)_{i=1}^k \in \{0,1\}^k$ that can later be explained, using Htd, as the CH-image of an arbitrary preimage. FTag then chooses uniform $r, s_0, \ldots, s_{\mathfrak{n}} \leftarrow \mathbb{Z}_p$ and sets

$$R := g^r, \quad S_0 := g^{s_0}, \quad S_{i,j} := U_j^{s_i} \quad (i \neq j), \quad S_{i,i} := U_i^{s_0+s_i} \cdot g^{\omega} \cdot \left(H_0 \prod_{i=1}^k H_i^{h_i}\right)^r. \tag{4}$$

Finally, FTag chooses CH-randomness R_{CH} such that $\mathsf{CH}_{Hpk}(R, S_0, (S_{i,j})_{i,j=1}^{\mathfrak{n}}, t_{\mathsf{a}}; R_{\mathsf{CH}}) = h$ and outputs $t_{\mathsf{c}} = (R, S, (S_{i,j})_{i,j=1}^{\mathfrak{n}}, R_{\mathsf{CH}})$. Intuitively, t_{c} consists of $\mathfrak{n}$ DLIN encryptions (with correlated randomness s_i) of Waters signatures $(g^r, g^{\omega} \cdot (H_0 \prod_{i=1}^k H_i^{h_i})^r)$ for message h. Indeed, substituting into (3) yields

$$Z_{i,i} := \frac{e(g,g)^{u_i(s_0+s_i)} \cdot W \cdot e(g, (H_0 \prod_{i=1}^k H_i^{h_i})^r)}{W \cdot e(g^r, H_0 \prod_{i=1}^k H_i^{h_i})} = e(g,g)^{u_i(s_0+s_i)}.$$

Hence, $\widetilde{Z}_{i,j} = u_j(s_0 + s_i)$ for *all* i, j, and thus the resulting tag $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ is lossy.

3.3 Security proof

Theorem 3.2. *If the DLIN assumption holds in* $\mathbb{G}$*, and* CH *is a chameleon hash function, then the LAF construction* LAF *from Section 3.2 satisfies Definition 3.1.*

The lossiness of LAF has already been discussed in Section 3.2. We prove indistinguishability and evasiveness separately.

Lemma 3.3. *For every adversary A on* LAF*'s indistinguishability, there exists a DLIN distinguisher B such that*

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},A}(k) \;=\; \frac{\mathsf{Adv}^{\mathsf{dlin}}_{B}(k)}{\mathfrak{n}}. \tag{5}$$

Intuitively, to see Lemma 3.3, observe that lossy tags differ from random tags only in their $S_{i,i}$ components, and in how the CH randomness R_{CH} is generated. For lossy tags, the $S_{i,i}$ are (parts of) DLIN ciphertexts, which are pseudorandom under the DLIN assumption. Furthermore, the uniformity property of CH guarantees that the distribution of R_{CH} is the same for lossy and random tags.

Proof. Assume a PPT adversary A. We proceed in games. In Game i, A gets an input Fpk and interacts with an oracle $\mathcal{O}_i$. Let out_i denote the A's output in Game i.

In **Game** 1, we let $\mathcal{O}_1(\cdot) := \mathsf{FTag}(Ftd, \cdot)$, where Ftd is the trapdoor initially sampled alongside Fpk. Thus, $\mathcal{O}_1(t_{\mathsf{a}})$ outputs core tags $t_{\mathsf{c}} = (R, S, (S_{i,j})_{i,j=1}^{\mathfrak{n}}, R_{\mathsf{CH}})$ generated as in (4).

In **Game** $2.j^*$ (for $0 \le j^* \le \mathfrak{n}$), we let $\mathcal{O}_2$ generate core tags as in Game 1, but with independently and uniformly chosen $S_{i,i} \in \mathbb{G}$ for $i \le j^*$. Note that Game 2.0 is equivalent to Game 1. Let furthermore Game 2 be defined as Game $2.\mathfrak{n}$. We claim

$$\Pr[out_1 = 1] - \Pr[out_2 = 1] \;=\; \Pr[out_{2.0} = 1] - \Pr[out_{2.\mathfrak{n}} = 1] \;=\; \frac{\mathsf{Adv}^{\mathsf{dlin}}_{B}(k)}{\mathfrak{n}} \tag{6}$$

for a suitable DLIN distinguisher B. Namely, B uniformly chooses $j^* \in [\mathfrak{n}]$, sets $j' := (j^* \bmod \mathfrak{n})+1$, and parses its DLIN challenge as $(g, U_{j'}, U_{j^*}, g^{s_0}, U_{j'}^{s_{j^*}}, C)$, where $C = U_{j^*}^{s_0+s_{j^*}}$ or $C \in \mathbb{G}$ is uniform. B then first re-randomizes its input to obtain many tuples $(g^{s_{0,\ell}}, U_{j'}^{s_{j^*,\ell}}, C_\ell)$, where (a) the $s_{0,\ell}, s_{j^*,\ell}$ are independently and uniformly random, and (b) $C_\ell = U_{j^*}^{s_{0,\ell}+s_{j^*,\ell}}$ iff $X = U_{j^*}^{s_0+s_{j^*}}$ (otherwise, all C_ℓ are independently and uniformly random). Next, B simulates Game $2.(j^*-1)$ or Game $2.j^*$, depending on its own challenge C. Concretely, to prepare a key Fpk for A, B sets $U_j = U_{j'}^{\alpha_j}$ for all $j \notin \{j', j^*\}$ and uniform $\alpha_j \leftarrow \mathbb{Z}_p$. (Like Game $2.j^*$, B chooses $\omega \leftarrow \mathbb{Z}_p$ and a CH keypair (Hpk, Htd) on its own.) When answering A's ℓ-th oracle query, B proceeds as in Game $2.j^*$, but sets up (a) $S_0 = g^{s_{0,\ell}}$, (b) $S_{i,i}$ as in Game 1 for $i > j^*$, (c) $S_{i,i} \leftarrow \mathbb{G}$ uniformly (as in Game 2) for $i < j^*$, (d) $S_{i,j^*} = (U_{j'}^{s_{j^*,\ell}})^{\alpha_i} = U_i^{s_{j^*,\ell}}$ for $i \neq j^*$, (e) $S_{j^*,j^*} = C_\ell \cdot g^\omega \cdot \left(H_0 \prod_{i=1}^k H_i^{h_i}\right)^r$. This implicitly sets $s_{j^*} = s_{j^*,\ell}$. (All other s_i are chosen by B.) Furthermore, if $C = U_{j^*}^{s_0+s_{j^*}}$, this setting of S_{i,j^*} yields Game $2.(j^*-1)$; but if C is uniform, then all C_i are independently uniform, and we obtain Game $2.j^*$. We get (6).

In **Game** 3, we choose the hash values R_{CH} in the core tags output by $\mathcal{O}_3$ uniformly and independently. Recall that up to Game 2, R_{CH} was instead chosen as follows: first choose a random CH-output h, and later select R_{CH} such that $\mathsf{CH}_{Hpk}(R, S_0, (S_{i,j})_{i,j=1}^{\mathfrak{n}}; R_{\mathsf{CH}}) = h$ holds. By definition of chameleon hashing, this induces a uniform distribution of R_{CH}. Moreover, h is not used in Game 2 or Game 3. Hence, the change in Game 3 is merely conceptual, and we obtain

$$\Pr[out_3 = 1] \;=\; \Pr[out_2 = 1].$$

Now note that in Game 3, the tags t_{c} output by $\mathcal{O}_3$ are random tags. Taking things together, (5) follows as desired. □

Lemma 3.4. *For every adversary A on* LAF*'s evasiveness, there exist adversaries B, C, and F such that*

$$\mathsf{Adv}^{\mathsf{eva}}_{\mathsf{LAF},A}(k) \le \left|\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},B}(k)\right| + \mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},C}(k) + \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{Sig_{Wat}},F}(k). \tag{7}$$

Intuitively, Lemma 3.4 holds because lossy (or, rather, non-injective) tags correspond to DLIN-encrypted Waters signatures. Hence, even after seeing many lossy tags (i.e., encrypted signatures), an adversary cannot produce a fresh encrypted signature. We note that the original Waters signatures from [25] are re-randomizable and thus not *strongly* unforgeable. To achieve evasiveness, we have thus combined Waters signatures with a chameleon hash function, much like Boneh et al. [8] did to make Waters signatures strongly unforgeable.

Proof. Assume a PPT adversary A. Again, we proceed in games. Let $\mathsf{bad_i}$ denote the event that A's output in Game i is a fresh non-injective tag. In **Game** 1, A gets input Fpk and interacts with an $\mathsf{FTag}(Ftd, \cdot)$ oracle. By definition,

$$\Pr[\mathsf{bad}_1] \;=\; \mathsf{Adv}^{\mathsf{eva}}_{\mathsf{LAF},A}(k).$$

To describe **Game** 2, denote A's output by $t^* = (t_{\mathsf{c}}{}^*, t_{\mathsf{a}}{}^*)$, for $t_{\mathsf{c}}{}^* = (R^*, S_0^*, (S_{i,j}^*)_{i,j=1}^{\mathfrak{n}}; R_{\mathsf{CH}}^*)$ Denote by $\mathsf{bad}_{\mathsf{coll}}$ the event that t^* induces a CH-collision in the sense that

$$h^* = \mathsf{CH}_{Hpk}(R^*, S_0^*, (S_{i,j}^*)_{i,j=1}^{\mathfrak{n}}; R_{\mathsf{CH}}^*) = \mathsf{CH}_{Hpk}(R, S_0, (S_{i,j})_{i,j=1}^{\mathfrak{n}}; R_{\mathsf{CH}}) = h$$

for some hash value h associated with an FTag-output $t_{\mathsf{c}} = (R, S_0, (S_{i,j})_{i,j=1}^{\mathfrak{n}}; R_{\mathsf{CH}})$ (and the corresponding query t_{a}). In Game 2, we abort (and do not raise event bad_2) if $\mathsf{bad}_{\mathsf{coll}}$ occurs. Intuitively, we would expect to use CH's collision resistance directly to argue that $\mathsf{bad}_{\mathsf{coll}}$ occurs only negligibly often. However, both in Game 1 and Game 2, we use CH's trapdoor Htd to construct lossy tags for A.

Hence, we first argue that $\mathsf{bad}_{\mathsf{coll}}$ occurs with essentially the same probability in a modified **Game** $1'$, in which A gets random tags instead of lossy tags as oracle answers. Indeed, since lossy and random tags are indistinguishable by Lemma 3.3, and $\mathsf{bad}_{\mathsf{coll}}$ is efficiently recognizable from A's view, we obtain

$$\Pr\left[\mathsf{bad}_{\mathsf{coll}} \text{ in Game } 1'\right] - \Pr[\mathsf{bad}_{\mathsf{coll}} \text{ in Game } 1] \;=\; \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},B}(k)$$

for a suitable adversary B on LAF's indistinguishability. Furthermore, since in Game $1'$, the CH-trapdoor Htd is not required, we have

$$\Pr\left[\mathsf{bad}_{\mathsf{coll}} \text{ in Game } 1'\right] \;=\; \mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},C}(k)$$

for a suitable collision-finder C. However, Game 1 and Game 2 only differ when $\mathsf{bad}_{\mathsf{coll}}$ occurs, and so we finally get

$$|\Pr[\mathsf{bad}_2] - \Pr[\mathsf{bad}_1]| \;\le\; \Pr[\mathsf{bad}_{\mathsf{coll}} \text{ in Game } 1] \;\le\; \left|\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},B}(k)\right| + \mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},C}(k).$$

The final reduction. Now that CH-collisions are excluded, we can finally conclude that any occurence of bad_2 means that A has forged a Waters signature. Concretely, we show that

$$\Pr[\mathsf{bad}_2] \;=\; \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{Sig}_{\mathsf{Wat}},F}(k) \tag{8}$$

for a suitable forger F that attacks $\mathsf{Sig}_{\mathsf{Wat}}$ and internally simulates Game 2 with A. Namely, F gets as input a $\mathsf{Sig}_{\mathsf{Wat}}$ public key $(\mathbb{G}, \mathbb{G}_T, e, p, g, (H_i)_{i=0}^k, W := e(g,g)^\omega)$. F extends this public key to an LAF public key Fpk by picking $U_i = g^{u_i}$ and Hpk. (In particular, F knows all u_i and Htd.) Upon an FTag-query from A, F constructs elements S_0 and $S_{i,j}$ (for $i \neq j$) exactly as in (4); note, however, that F cannot directly compute the $S_{i,i}$, since F does not know g^ω. Instead, F requests a $\mathsf{Sig}_{\mathsf{Wat}}$ signature for the message $h \in \{0,1\}^k$ (as derived in (4)). Such a signature is of the form

$$(g^r, g^\omega \cdot \left(H_0 \prod_{i=1}^k H_i^{h_i}\right)^r),$$

from which F can compute the elements R and $S_{i,i}$ as in (4). Since F also knows the CH-trapdoor Htd, this allows to construct lossy tags exactly as FTag would do in Game 2.

It remains to describe how F extracts a $\mathsf{Sig}_{\mathsf{Wat}}$-signature out of a lossy tag $t = (t_{\mathsf{c}}, t_{\mathsf{a}})$ that A finally outputs. By our definition of tags, we may assume that $t_{\mathsf{c}} = (R, S_0, (S_{i,j})_{i,j=1}^{\mathfrak{n}}, R_{\mathsf{CH}})$ is of the form $R = g^r$, $S_0 = g^{s_0}$, and $S_{i,j} = U_j^{s_i}$ for suitable r, s_i and all $i \neq j$. Furthermore, since t_{c} is lossy,

$$\operatorname{rank}(\widetilde{Z}) < \mathfrak{n} \implies \exists i : \widetilde{Z}_{i,i} = u_i(s_0 + s_i) \implies \exists i : S_{i,i} = U_i^{s_0+s_i} \cdot g^\omega \cdot \left(H_0 \prod_{i=1}^k H_i^{h_i}\right)^r. \tag{9}$$

Since F knows all u_i, it can compute

$$\sigma_i := \frac{S_{i,i}}{S_0^{u_i} \cdot S_{i,j}^{u_i/u_j}} = \frac{S_{i,i}}{U_i^{s_0+s_i}}$$

for all i (and some $j \neq i$). By (9), for some i, the pair (R, σ_i) forms a valid $\mathsf{Sig}_{\mathsf{Wat}}$ signature for $h = \mathsf{CH}_{Hpk}(R, S_0, (S_{i,j})_{i,j=1}^{\mathfrak{n}}; R_{\mathsf{CH}})$. Because Game 2 aborts in case of a CH-collision, we may further assume that h is a message for which F has not yet requested a signature. Consequently, F can output a forged signature for a fresh message whenever bad_2 occurs. This yields (8). Putting things together finally gives (7). □

Combining Lemma 3.3, Lemma 3.4, and the fact that Waters signatures are EUF-CMA secure already under the CDH assumption, we obtain Theorem 3.2.

4 CIRC-CCA-secure encryption scheme

4.1 The scheme

Setting and ingredients. First, we assume an algorithm GenN that outputs ℓ_N-bit Blum integers $N = PQ$ along with their prime factors P and Q. If N is clear from the context, we write $\mathbb{G}_{\mathsf{rnd}}$ and $\mathbb{G}_{\mathsf{msg}}$ for the unique subgroups of $\mathbb{Z}_{N^3}^*$ of order $(P-1)(Q-1)/4$, resp. N^2. We also write $h := 1 + N \bmod N^3$, so $\langle h \rangle = \mathbb{G}_{\mathsf{msg}}$. Note that it is efficiently possible to compute $\mathrm{dlog}_h(X) := x$ for $X := h^x \in \mathbb{G}_{\mathsf{msg}}$ and $x \in \mathbb{Z}_{N^2}$. Specifically, it is efficiently possible to test for membership in $\mathbb{G}_{\mathsf{msg}}$. In our scheme, $\mathbb{G}_{\mathsf{msg}}$ will be used to embed a suitably encoded message, and $\mathbb{G}_{\mathsf{rnd}}$ will be used for blinding. We will require that

- P and Q are safe primes of bitlength between $\ell_N/2 - k$ and $\ell_N/2 + k$,
- $\gcd((P-1)(Q-1)/4, N) = 1$ (which holds, e.g., for uniform P, Q of a certain length),
- $\ell_N \geq 25k + 8$ (e.g., $k = 80$ and $\ell_N = 2048$)[7]
- the DCR assumption holds in $\mathbb{Z}_{N^3}^*$, and the DDH assumption holds in $\mathbb{G}_{\mathsf{rnd}}$.

We also assume an $(\ell_{\mathsf{LAF}}, \mathfrak{n})$-lossy algebraic filter LAF for $\mathfrak{n} = 6$ and $\ell_{\mathsf{LAF}} = (\ell_N + k + 1)/(\mathfrak{n} - 2)$. Our scheme will encrypt messages from the domain

$$\mathcal{M} := \mathbb{Z}_{2^{3k}} \times \mathbb{Z}_{p \cdot 2^k} \times \mathbb{Z}_{N \cdot 2^{k-2}},$$

where p is the modulus of the used LAF. (The reason for this weird-looking message space will become clearer in the proof.) During encryption, we will have to treat a message $M = (a, b, c) \in \mathcal{M}$ both as an element of $\mathbb{Z}_{N^2}$ and as an LAF-input from $\mathbb{Z}_p^{\mathfrak{n}}$. In these cases, we can encode

$$[M]_{\mathbb{Z}} := a + 2^{3k} \cdot b + p \cdot 2^{4k} \cdot c \in \mathbb{Z}, \qquad [M]_{\mathbb{Z}_p^{\mathfrak{n}}} := (a, b \bmod p, c_0, \ldots, c_{\mathfrak{n}-3}) \in \mathbb{Z}_p^{\mathfrak{n}} \tag{10}$$

for the natural interpretation of $\mathbb{Z}_i$-elements as integers between 0 and $i - 1$, and c's p-adic representation $(c_i)_{i=0}^{\mathfrak{n}-3} \in \mathbb{Z}_p^{\mathfrak{n}-2}$ with $c = \sum_{i=0}^{\mathfrak{n}-3} c_i \cdot p^i$. Note that by our requirements on ℓ_N and ℓ_{LAF}, we have $0 \leq [M]_{\mathbb{Z}} < N^2 - 2^k$. However we stress that the encoding $[M]_{\mathbb{Z}_p^{\mathfrak{n}}}$ is *not* injective, since it only depends on $b \bmod p$ (while $0 \leq b < p \cdot 2^k$).

Finally, we assume a strongly OT-EUF-CMA secure signature scheme $\mathsf{Sig} = (\mathsf{SGen}, \mathsf{Sig}, \mathsf{Ver})$, and a key-unique IND-CPA secure symmetric encryption scheme (E, D) (see Section 2) with k-bit symmetric keys K and message space $\{0,1\}^*$.

Now consider the following PKE scheme PKE:

Public parameters. $\mathsf{Pars}(1^k)$ first runs $(N, P, Q) \leftarrow \mathsf{GenN}(1^k)$. Recall that this fixes the groups $\mathbb{G}_{\mathsf{rnd}}$ and $\mathbb{G}_{\mathsf{msg}}$. Then, Pars selects two generators g_1, g_2 of $\mathbb{G}_{\mathsf{rnd}}$. Finally, Pars runs $(Fpk, Ftd) \leftarrow \mathsf{FGen}(1^k)$, and outputs

$$pp = (N, g_1, g_2, Fpk).$$

[7] Depending on the parameter $\mathfrak{n}$ below, shorter ℓ_N are possible. The relevant inequality that must hold is (15).

In the following, we denote with p the LAF modulus contained in Fpk.

Key generation. $\mathsf{Gen}(pp)$ uniformly selects two messages $s_j = (a_j, b_j, c_j) \in \mathcal{M}$ (for $j \in \{1, 2\}$) as secret key, and sets

$$pk := \tilde{g} := \left(g_1^{[s_1]_\mathbb{Z}} g_2^{[s_2]_\mathbb{Z}}\right)^{2^k} \qquad\qquad sk := (s_1, s_2).$$

Encryption. $\mathsf{Enc}(pp, pk, M)$, for pp and pk as above, and $M \in \mathcal{M}$, uniformly selects an exponent $r \leftarrow \mathbb{Z}_{N/4}$, a random filter core tag t_c, a Sig-keypair $(verk, sigk) \leftarrow \mathsf{SGen}(1^k)$, and a random symmetric key $K \in \{0,1\}^k$ for (E, D), and computes

$$\begin{gathered} C_1 := g_1^r, \quad C_2 := g_2^r, \quad \widetilde{C} := \tilde{g}^r \cdot h^{K + 2^k \cdot [M]_\mathbb{Z}} \\ C_\mathsf{E} \leftarrow \mathsf{E}(K, \mathsf{LAF}_{Fpk,t}([M]_{\mathbb{Z}_p^\mathfrak{n}})), \\ \sigma \leftarrow \mathsf{Sig}(sigk, (C_1, C_2, \widetilde{C}, C_\mathsf{E})) \\ C := (C_1, C_2, \widetilde{C}, C_\mathsf{E}, t_\mathsf{c}, verk, \sigma) \end{gathered}$$

for the auxiliary tag $t_\mathsf{a} := verk$, and the resulting filter tag $t := (t_\mathsf{c}, t_\mathsf{a})$.

Decryption. $\mathsf{Dec}(pp, sk, C)$, for pp, sk and C as above, first checks the signature σ and rejects with $\perp$ if $\mathsf{Ver}(verk, (C_1, C_2, \widetilde{C}, C_\mathsf{E}), \sigma) = 0$. Then Dec computes

$$\widehat{C} := \left(C_1^{[s_1]_\mathbb{Z}} C_2^{[s_2]_\mathbb{Z}}\right)^{2^k}$$

and then $K \in \{0,1\}^k, M \in \mathcal{M}$ with

$$K + 2^k \cdot [M]_\mathbb{Z} \;:=\; \mathrm{dlog}_h(\widetilde{C}/\widehat{C}).$$

If $\widetilde{C}/\widehat{C} \notin \mathbb{G}_\mathsf{msg}$, or no such M exists, or $\mathsf{D}(K, C_\mathsf{E}) \neq \mathsf{LAF}_{Fpk,t}([M]_{\mathbb{Z}_p^\mathfrak{n}})$ (for $t = (t_\mathsf{c}, t_\mathsf{a})$ computed from C as during encryption), then Dec rejects with $\perp$. Else, Dec outputs M.

Secret keys as messages. Our scheme has secret keys $s = (s_1, s_2) \in \mathcal{M}^2$; hence, we can only encrypt one half s_j of a secret key at a time. In the security proof below, we will thus only consider KDM queries that ask to encrypt a specific secret key *part*. Alternatively, we can change our scheme, so that pairs of $\mathcal{M}$-elements are encrypted. To avoid malleability (which would destroy CCA security), we of course have to use only one LAF tag for this. Our CIRC-CCA proof below applies to such a changed scheme with minor syntactic changes.

Efficiency. When instantiated with our DLIN-based LAF construction from Section 3, and taking $\mathfrak{n} = 6$ as above, our scheme has ciphertexts with 38 $\mathbb{G}$-elements, 3 $\mathbb{Z}_{N^3}$-elements, plus chameleon hash randomness, a one-time signature and verification key, and a symmetric ciphertext (whose size could be in the range of one $\mathbb{Z}_{N^2}$-element plus some encryption randomness). The number of group elements in the ciphertext is constant, and does not grow in the security parameter. The public parameters contain $\mathbf{O}(k)$ group elements (most of them from $\mathbb{G}$), and public keys contain only one $\mathbb{Z}_{N^3}$-element; secret keys consist of two $\mathbb{Z}_{N^2}$-elements. While these parameters are not competitive with current non-KDM-secure schemes, they are significantly better than those from the circular-secure scheme of Camenisch et al. [14].

4.2 Security proof (1-user case)

It is instructive to first consider the one-user case. In this case, we essentially only require that PKE is IND-CCA secure, even if encryptions of its secret key are made public. Already the one-user case will allow us to showcase most of the techniques required for the multi-user case.

Theorem 4.1. *Assume the DCR assumption holds in $\mathbb{Z}_{N^3}$, the DDH assumption holds in $\mathbb{G}_\mathsf{rnd}$,* LAF *is an LAF,* Sig *is a strongly OT-EUF-CMA secure signature scheme, and* (E, D) *is a key-unique IND-CPA secure symmetric encryption scheme. Then* PKE *is 1-CIRC-CCA-secure.*

Proof. Assume a PPT adversary A on PKE's 1-CIRC-CCA security. Say that A always makes $q = q(k)$ KDM queries. We proceed in games. Let out_i denote the output of Game i.

Game 1 is the 1-KDM-CCA experiment with PKE and A. Thus, by definition,

$$\Pr[out_1 = 1] - 1/2 \;=\; \mathsf{Adv}^{\mathsf{kdm\text{-}cca}}_{\mathsf{PKE},A}(k).$$

In **Game** 2, we slightly change how answers to KDM queries are prepared. Namely, in each KDM ciphertext, we set up $\widetilde{C}$ not as $\widetilde{C} \;=\; \tilde{g}^r \cdot h^{K+2^k\cdot[M]_\mathbb{Z}}$, but instead as

$$\widetilde{C} \;=\; \left(C_1^{[s_1]_\mathbb{Z}} \cdot C_2^{[s_2]_\mathbb{Z}}\right)^{2^k} \cdot h^{K+2^k\cdot([M]_\mathbb{Z})}.$$

This change is only conceptual, as $C_i = g_i^r$ implies $(C_1^{[s_1]_\mathbb{Z}} \cdot C_2^{[s_2]_\mathbb{Z}})^{2^k} = (g_1^{r\cdot[s_1]_\mathbb{Z}} \cdot g_2^{r\cdot[s_2]_\mathbb{Z}})^{2^k} = \tilde{g}^r$.

In **Game** 3, we again change how KDM ciphertexts are prepared. Namely, for KDM queries to encrypt an s_j (and only for those), we let $C_j := g_j^r/h^{2^k}$, and prepare the remaining parts of C as in Game 2. We claim that

$$\Pr[out_3 = 1] - \Pr[out_2 = 1] \;\le\; 2 \cdot \mathsf{Adv}^{\mathsf{dcr}}_{\mathbb{Z}^*_{N^3},B}(k) + \mathbf{O}(2^{-k}) \tag{11}$$

for a suitable DCR distinguisher B that simulates Game 2, resp. Game 3. Concretely, B gets as input a value $\widetilde{Z} \in \mathbb{Z}^*_{N^3}$. Note that if we set $Z := (\widetilde{Z}^2)^{2^{-1} \bmod N^2}$, we have $Z = g^{\widehat{r}} \cdot h^b \in \mathbb{Z}^*_{N^3}$, with uniform $g^{\widehat{r}} \in \mathbb{G}_{\mathsf{rnd}}$ and $b \in \{0,1\}$. First, B guesses a value of $j \in \{1,2\}$. (This gives a very small hybrid argument, in which in the j-th step, only encryptions of s_j are changed.) Both g_i are computed from Z as $g_j = Z^{N^2}$ and $g_{3-j} = g_j^{\alpha\cdot N^2}$ for uniform $\alpha \leftarrow \mathbb{Z}_{N/4}$. Furthermore, KDM encryptions of s_j are prepared using

$$C_j = Z^{2^k} \cdot g_j^\beta, \qquad\qquad C_{3-j} = Z^{2^k\cdot\alpha\cdot N^2} \cdot g_{3-j}^\beta$$

for a fresh uniform $\beta \leftarrow \mathbb{Z}_{N/4}$. This gives s_j-encryptions as in Game $2+b$, and thus yields (11). (The $\mathbf{O}(2^{-k})$ term in (11) accounts for the statistical defect caused by choosing random $\mathbb{G}_{\mathsf{rnd}}$-exponents from $\mathbb{Z}_{N/4}$.)

Our change in Game 3 implies $C_1^{[s_1]_\mathbb{Z}} \cdot C_2^{[s_2]_\mathbb{Z}} = \tilde{g}^r/h^{2^k\cdot[s_j]_\mathbb{Z}}$, and so $\widetilde{C} \;=\; \tilde{g}^r \cdot h^K$ when s_j is to be encrypted. This means that A still obtains information about the s_j (beyond what is public from pk) from its KDM queries, but this information is limited to values $\mathsf{LAF}_{Fpk,t}([s_j]_{\mathbb{Z}_p^\mathfrak{n}})$. We will now further cap this leaked information by making $\mathsf{LAF}_{Fpk,t}(\cdot)$ lossy.

In **Game** 4, we use the LAF trapdoor Ftd initially sampled with Fpk. Concretely, when preparing a KDM ciphertext $C = (C_1, C_2, \widetilde{C}, C_\mathsf{E}, t_\mathsf{c}, verk, \sigma)$ for A, we sample the tag t_c using $t_\mathsf{c} \leftarrow \mathsf{FTag}(Ftd, t_\mathsf{a})$ for the corresponding auxiliary tag $t_\mathsf{a} = verk$. A straightforward reduction shows

$$\Pr[out_4 = 1] - \Pr[out_3 = 1] \;=\; \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{LAF},C}(k)$$

for a suitable adversary C on LAF's indistinguishability.

In **Game** 5, we reject all decryption queries of A that re-use a tag t from one of the KDM ciphertexts. To show that this change does not significantly affect A's view, assume a decryption query $C = (C_1, C_2, \widetilde{C}, C_\mathsf{E}, t_\mathsf{c}, verk, \sigma)$ that re-uses a tag t from a KDM ciphertext $C^* = (C_1^*, C_2^*, \widetilde{C}^*, C_\mathsf{E}^*, t_\mathsf{c}{}^*, verk^*, \sigma^*)$. Observe that C contains a signature σ under an honestly generated Sig-verification-key $verk = t_\mathsf{a} = t_\mathsf{a}{}^* = verk^*$. Since A may not submit unchanged challenge ciphertexts for decryption, and we assumed $(t_\mathsf{c}, t_\mathsf{a}) = (t_\mathsf{c}{}^*, t_\mathsf{a}{}^*)$, either the signed message $(C_1, C_2, \widetilde{C}, C_\mathsf{E})$ or the signature σ must be fresh (i.e., different from $(C_1^*, C_2^*, \widetilde{C}^*, C_\mathsf{E}^*)$, resp. σ^*). Hence, if σ is valid, then A has produced a fresh valid Sig-signature. Thus, a straightforward reduction to the strong OT-EUF-CMA security of Sig yields

$$\Pr[out_5 = 1] - \Pr[out_4 = 1] \;=\; q(k) \cdot \mathsf{Adv}^{\mathsf{seuf\text{-}cma}}_{\mathsf{LAF},F_\mathsf{Sig}}(k)$$

for a forger F_Sig against Sig that makes at most one signature query.

In **Game** $6.i$ (for $0 \leq i \leq q$), the first i KDM ciphertexts are prepared using $\widetilde{C} = \widehat{g} \cdot h^K$ (if a key component s_j is to be encrypted), resp. $\widetilde{C} = \widehat{g} \cdot h^{[M]_\mathbb{Z}}$ (if a constant $M \in \mathcal{M}$ is to be encrypted) for an independently uniform $\widehat{g} \leftarrow \mathbb{G}_{\mathsf{rnd}}$ drawn freshly for each ciphertext. Obviously, Game 6.0 is identical to Game 5, and in Game $6.q$, all $\mathbb{G}_{\mathsf{rnd}}$-components of all $\widetilde{C}$ are fully randomized. Specifically,

$$\Pr[out_{6.0} = 1] \; = \; \Pr[out_5 = 1].$$

We will move from Game $6.i$ to Game $6.(i+1)$ in several steps. During these steps, let $C = (C_1, C_2, \widetilde{C}, C_\mathsf{E}, t_\mathsf{c}, verk, \sigma)$ denote the $(i+1)$-st KDM ciphertext.

In **Game** $6.i.1$, we change the $\mathbb{G}_{\mathsf{rnd}}$ parts of C_1, C_2 from a Diffie-Hellman tuple (with respect to g_1, g_2) to a random tuple. Concretely, if an s_j is to be encrypted, we prepare $(C_j, C_{3-j}) = (g_j^{r_j}/h^{2^k}, g_{3-j}^{r_{3-j}})$; if a constant M is encrypted, we set $(C_1, C_2) = (g_1^{r_1}, g_2^{r_2})$, in both cases for independently uniform $r_1, r_2 \leftarrow \mathbb{Z}_{N/4}$. The $\mathbb{G}_{\mathsf{msg}}$ parts of C_1, C_2 are thus unchanged compared to Game $6.i$. A straightforward reduction to the DDH assumption in $\mathbb{G}_{\mathsf{rnd}}$ yields that

$$\sum_{i=1}^{q(k)} \left(\Pr[out_{6.i} = 1] - \Pr[out_{6.i.1} = 1]\right) \; = \; q(k) \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_{\mathsf{rnd}}, D_1}(k) + \mathbf{O}(2^{-k})$$

for a suitable D_1. The $\mathbf{O}(2^{-k})$ error term accounts for the statistical difference caused by the choice of exponents $r \leftarrow \mathbb{Z}_{N/4}$, which induces an only almost-uniform distribution on group elements g^r. Note that at this point, $\widetilde{C}$ is still computed as $\widetilde{C} = (C_1^{[s_1]_\mathbb{Z}} C_2^{[s_2]_\mathbb{Z}})^{2^k} \cdot h^{K+[M]_\mathbb{Z}}$ (even if a message $M = s_j$ is to be encrypted).

In **Game** $6.i.2$, we compute $\widetilde{C}$ as $\widetilde{C} = \widehat{g} \cdot h^{K+[M]_\mathbb{Z}}$ for a fresh $\widehat{g} \leftarrow \mathbb{G}_{\mathsf{rnd}}$. Thus, the difference to Game $6.i.1$ is that we substitute a $\mathbb{G}_{\mathsf{rnd}}$-element computed as $(g_1^{r_1[s_1]_\mathbb{Z}} \cdot g_2^{r_1[s_2]_\mathbb{Z}})^{2^k}$ with a fresh random $\widehat{g}$. To show that this change affects A's view only negligibly, it suffices to show that A's statistical information about

$$X \; := \; \mathrm{dlog}_g\left(g_1^{r_1[s_1]_\mathbb{Z}} \cdot g_2^{r_2[s_2]_\mathbb{Z}}\right) \; = \; r_1\alpha_1[s_1]_\mathbb{Z} + r_2\alpha_2[s_2]_\mathbb{Z} \mod |\mathbb{G}_{\mathsf{rnd}}|$$

(for some arbitrary generator g of $\mathbb{G}_{\mathsf{rnd}}$ and $\alpha_i = \mathrm{dlog}_g(g_i)$) is negligible. This part of the proof will be rather delicate, since we will have to argue that both A's KDM queries and A's decryption queries yield (almost) no information about X.

First, observe that A gets the following information about the s_j:

- pk reveals (through $\tilde{g}$) precisely one equation $\alpha_1[s_1]_\mathbb{Z} + \alpha_2[s_2]_\mathbb{Z} \mod |\mathbb{G}_{\mathsf{rnd}}|$ about the s_j, where the α_i are as above. Hence, for uniform r_1, r_2, X is (almost) independent of pk.
- By LAF's lossiness, KDM ciphertexts reveal (through $C_\mathsf{E} = \mathsf{E}(K, \mathsf{LAF}_{Fpk,t}([s_j]_{\mathbb{Z}_p^\mathsf{n}}))$ at most one equation $\omega_1 a_j + \omega_2 b_j + \sum_{i=0}^{\mathsf{n}-2} \omega_{3+i} c_{j,i} \mod p$ for each j, where $(a_j, b_j, c_{j,0}, \ldots, c_{j,\mathsf{n}-3}) := [s_j]_{\mathbb{Z}_p^\mathsf{n}}$, and the ω_i are the (fixed) coefficients from LAF's lossiness property.

Recall the encodings $[s_j]_\mathbb{Z}, [s_j]_{\mathbb{Z}_p^\mathsf{n}}$ of the $s_j = (a_j, b_j, c_j) \in \mathcal{M}$ from (10). Note that the $b_j \in \mathbb{Z}_{p \cdot 2^k}$ fully blind the information released about the $c_j \in \mathbb{Z}_{2^{k-2}N}$ through the KDM ciphertexts. Furthermore, pk reveals only information about $\alpha_1 c_1 + \alpha_2 c_2 \mod |\mathbb{G}_{\mathsf{rnd}}|$, where the $c_j \in \mathbb{Z}_{2^{k-2}N}$ are statistically close to uniform modulo $|\mathbb{G}_{\mathsf{rnd}}|$. Since this latter equation (in the unknowns c_j) is linearly independent from $r_1\alpha_1 c_1 + r_2\alpha_2 c_2 \mod |\mathbb{G}_{\mathsf{rnd}}|$ with high probability over uniform r_1, r_2, also X is (almost) uniformly random and independent from A's view.

This already shows that our change from Game $6.i.2$ affects A's view only negligibly if A makes no decryption queries. It remains to show that decryption queries yield no additional information about the s_j. To do so, let us say that a ciphertext $C = (C_1, C_2, \widetilde{C}, C_\mathsf{E}, t_\mathsf{c}, verk, \sigma)$ is *inconsistent* iff (C_1, C_2) is not of the form (g_1^r, g_2^r) for some r. Note that the decryption of a consistent ciphertext yields no information about the s_j beyond pk. (pk and C_1, C_2 determine the value $\widehat{C}$ computed during decryption; everything else follows from $\widehat{C}$ and C.) Thus, it suffices to prove the following lemma (which we do after the main proof):

Lemma 4.2. *In the situation of Game* $6.i.j$ *(for* $j \in \{1,2\}$*), let* $\mathsf{bad}_{\mathsf{query.i.j}}$ *be the event that* A *places an inconsistent decryption query that is not rejected. Then*

$$\sum_{i=1}^{q(k)} \left(\Pr\left[\mathsf{bad}_{\mathsf{query.i.1}}\right] + \Pr\left[\mathsf{bad}_{\mathsf{query.i.2}}\right]\right) \;\leq\; 2 \cdot q(k) \cdot \mathsf{Adv}^{\mathsf{eva}}_{\mathsf{LAF},F}(k) + \mathbf{O}(2^{-3k}).$$

for a suitable evasiveness adversary F *on* LAF.

By our discussion above and Lemma 4.2, we obtain that

$$\sum_{i=1}^{q(k)} \left|\Pr\left[out_{6.i.2} = 1\right] - \Pr\left[out_{6.i.1} = 1\right]\right| \;\leq\; 2 \cdot q(k) \cdot \mathsf{Adv}^{\mathsf{eva}}_{\mathsf{LAF},F}(k) + \mathbf{O}(2^{-3k}).$$

In **Game** $6.i.3$, we reverse the change from Game $6.i.1$. Concretely, we prepare $(C_j, C_{3-j}) = (g_j^r/h^{2^k}, g_{3-j}^r)$ (if an s_j is encrypted), resp. $(C_1, C_2) = (g_1^r, g_2^r)$ (if a constant M is encrypted) for $r \leftarrow \mathbb{Z}_{N/4}$. Another straightforward reduction to the DDH assumption in $\mathbb{G}_{\mathsf{rnd}}$ yields that

$$\sum_{i=1}^{q(k)} \left(\Pr\left[out_{6.i.3} = 1\right] - \Pr\left[out_{6.i.2} = 1\right]\right) \;=\; q(k) \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathbb{G}_{\mathsf{rnd}},D_2}(k) + \mathbf{O}(2^{-k})$$

for a suitable D_2. To close the hybrid argument, note that Game $6.i.3$ and Game $6.(i+1)$ are identical.

In **Game** 7, we completely randomize the $\widetilde{C}$ component of all KDM ciphertexts prepared for A. That is, instead of computing $\widetilde{C} = \widehat{g} \cdot h^{K+[M]_{\mathbb{Z}}}$ for a freshly uniform $\widehat{g} \leftarrow \mathbb{G}_{\mathsf{rnd}}$, we sample $\widetilde{C} \leftarrow \mathbb{Z}^*_{N^3}$. Since already all $\widetilde{C}$ have an independently uniform $\mathbb{G}_{\mathsf{rnd}}$-component, a straightforward reduction to the DCR assumption yields

$$\Pr\left[out_{6.q} = 1\right] - \Pr\left[out_7 = 1\right] \;=\; \mathsf{Adv}^{\mathsf{dcr}}_{\mathbb{Z}^*_{N^3},E}(k) + \mathbf{O}(2^{-k})$$

for a DCR distinguisher E. Note that because of the re-randomizability of DCR, there is no factor of $q(k)$, even though we substitute many group elements at once. However, since the precise order of $\mathbb{G}_{\mathsf{rnd}}$ is not known, this re-randomization costs us an error term of $\mathbf{O}(2^{-k})$).

In **Game** 8, we substitute the symmetric ciphertexts C_{E} in all KDM ciphertexts by encryptions of random messages. By our change in Game 7, we do not use the symmetric keys K used to produce C_{E} anywhere else. Thus, a reduction to the IND-CPA security of (E, D) gives

$$\Pr\left[out_7 = 1\right] - \Pr\left[out_8 = 1\right] \;=\; q(k) \cdot \mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{(\mathsf{E},\mathsf{D}),G}(k)$$

for an IND-CPA adversary G.

Finally, note that in Game 8, A's view is independent of the challenge bit b initially selected by the KDM challenger. Hence, we have

$$\Pr\left[out_8 = 1\right] \;=\; 1/2.$$

Taking things together yields the theorem. □

It remains to prove Lemma 4.2, which we do now:

Proof. Let $\mathsf{bad}_{\mathsf{tag.i.j}}$ be the event that in Game $6.i.j$, A submits a decryption query that refers to a lossy tag t. By our change in Game 5, we may assume that t is fresh, i.e., has not been generated through FTag by the experiment. Thus, by LAF's evasiveness, $\mathsf{bad}_{\mathsf{tag}}$ can occur only with negligible probability. Concretely, it is easy to construct an evasiveness adversary F with

$$\sum_{i=1}^{q(k)} \left(\Pr\left[\mathsf{bad}_{\mathsf{tag.i.1}}\right] + \Pr\left[\mathsf{bad}_{\mathsf{tag.i.2}}\right]\right) \;\leq\; 2 \cdot q(k) \cdot \mathsf{Adv}^{\mathsf{eva}}_{\mathsf{LAF},F}(k). \tag{12}$$

Now suppose that we are in Game $6.i.j$, and say that $\mathsf{bad}_{\mathsf{tag.i.j}}$ does not occur. Consider an inconsistent decryption query $C = (C_1, C_2, \widetilde{C}, C_{\mathsf{E}}, t_{\mathsf{c}}, \mathit{verk}, \sigma)$ from A. Write $C_i = g_i^{r_i} \cdot h^{\gamma_i}$ (for $i \in \{1,2\}$) and $\widetilde{C} = \tilde{g}^{\tilde{r}} \cdot h^{\tilde{\gamma}}$. Recall that decryption first computes

$$\widehat{C} \;=\; \left(C_1^{[s_1]_{\mathbb{Z}}} C_2^{[s_2]_{\mathbb{Z}}}\right)^{2^k} \;=\; \left(g_1^{r_1[s_1]_{\mathbb{Z}}} g_2^{r_2[s_2]_{\mathbb{Z}}}\right)^{2^k} \cdot h^{(\gamma_1[s_1]_{\mathbb{Z}} + \gamma_2[s_2]_{\mathbb{Z}}) \cdot 2^k}, \tag{13}$$

and from this values $K \in \{0,1\}^k, M \in \mathcal{M}$ with

$$K + 2^k \cdot [M]_{\mathbb{Z}} \;=\; \mathrm{dlog}_h(\widetilde{C}/\widehat{C}) \;=\; \tilde{\gamma} - (\gamma_1[s_1]_{\mathbb{Z}} + \gamma_2[s_2]_{\mathbb{Z}}) \cdot 2^k \bmod N^2. \tag{14}$$

As usual, we write $s_j = (a_j, b_j, c_j) \in \mathcal{M} = \mathbb{Z}_{2^{3k}} \times \mathbb{Z}_{p \cdot 2^k} \times \mathbb{Z}_{N \cdot 2^{k-2}}$ for $i \in \{1,2\}$.

h-inconsistent ciphertexts. First, consider the case that there is an $i^* \in \{1,2\}$ with $\gamma_{i^*} \neq 0 \bmod N^2$. (In that case, we may say that C is *h-inconsistent.*) Then, we claim that either C is rejected, or A has (information-theoretically) successfully narrowed down the value of

$$S \;:=\; \gamma_1[s_1]_{\mathbb{Z}} + \gamma_2[s_2]_{\mathbb{Z}} \bmod N^2$$

to a set of size at most 2^k. Indeed, C_{E} determines K and thus $\mathsf{LAF}_{Fpk,t}([M]_{\mathbb{Z}_p^{\mathfrak{n}}}) = \mathsf{D}(K, C_{\mathsf{E}})$ by (E, D)'s key-uniqueness. Moreover, since we assumed $\neg\mathsf{bad}_{\mathsf{tag.i.j}}$, the used tag t is injective, and so $\mathsf{LAF}_{Fpk,t}([M]_{\mathbb{Z}_p^{\mathfrak{n}}})$ determines M up to $\lfloor b/p \rfloor \in \mathbb{Z}_{2^k}$. (Recall that the encoding $[M]_{\mathbb{Z}_p^{\mathfrak{n}}}$ only depends on $b \bmod p$.) Thus, a non-rejected ciphertext allows to infer (a 2^k-candidate set for) S by substituting K, M, and $\tilde{\gamma}$ (as defined by $\widetilde{C}$) into (14).

However, we will now argue that S has min-entropy at least $5k$, even given pk, the KDM ciphertexts, and s_{3-i^*}. Hence, A cannot predict a correct 2^k-candidate set for S (and thus cannot supply an h-inconsistent decryption query that is not rejected) with non-negligible probability. To prove our claim, we need some preparations. Since $\gamma_{i^*} \neq 0 \bmod N^2$, either $P^2 \nmid \gamma_{i^*}$ or $Q^2 \nmid \gamma_{i^*}$ (or both) for the factors P, Q of N. Without loss of generality, say that $P^2 \nmid \gamma_{i^*}$, so that the subterm $\gamma_{i^*} \cdot [s_{i^*}]_{\mathbb{Z}} \bmod N^2$ of S reveals $[s_{i^*}]_{\mathbb{Z}} \bmod P$. Furthermore,

$$\begin{aligned}
[s_{i^*}]_{\mathbb{Z}} \overset{(10)}{<} 2^{5k-2} \cdot p \cdot N \;\leq\; 2^{5k+1} \cdot 2^{\ell_{\mathsf{LAF}}} \cdot |\mathbb{G}_{\mathsf{rnd}}| &\overset{\ell_{\mathsf{LAF}} = \frac{\ell_N + k + 1}{\mathfrak{n} - 2}}{<} 2^{(5+1/(\mathfrak{n}-2))k+2} \cdot 2^{\ell_N/(\mathfrak{n}-2)} \cdot |\mathbb{G}_{\mathsf{rnd}}| \\
&\overset{P \geq 2^{(\ell_N/2)-k}}{\leq} 2^{(6+1/(\mathfrak{n}-2))k+2-(1/2-1/(\mathfrak{n}-2))\ell_N} \cdot |\mathbb{G}_{\mathsf{rnd}}| \cdot P \overset{\substack{\ell_N \geq 25k+8 \\ \mathfrak{n}=6}}{\leq} |\mathbb{G}_{\mathsf{rnd}}| \cdot P.
\end{aligned} \tag{15}$$

Using $\gcd(P, |\mathbb{G}_{\mathsf{rnd}}|) = 1$, the Chinese Remainder Theorem hence gives that $[s_{i^*}]_{\mathbb{Z}} \bmod |\mathbb{G}_{\mathsf{rnd}}|$ and $[s_{i^*}]_{\mathbb{Z}} \bmod P$ uniquely determine $[s_{i^*}]_{\mathbb{Z}}$. Thus, since $[s_{i^*}]_{\mathbb{Z}}$ initially has min-entropy at least $5k - 2 + \ell_{\mathsf{LAF}} + \ell_N$, revealing $[s_{i^*}]_{\mathbb{Z}} \bmod |\mathbb{G}_{\mathsf{rnd}}|$ (through pk) leaves at least $5k + \ell_{\mathsf{LAF}}$ bits of min-entropy in $[s_{i^*}]_{\mathbb{Z}} \bmod P$. The KDM ciphertexts reveal no more than ℓ_{LAF} bits of entropy about $[s_{i^*}]_{\mathbb{Z}} \bmod P$, so that $[s_{i^*}]_{\mathbb{Z}} \bmod P$ has min-entropy at least $5k$.

However, C implies 2^k candidates for S which, given s_{3-i^*}, in turn determine 2^k candidates for $[s_{i^*}]_{\mathbb{Z}} \bmod P$. So, assuming $\neg\mathsf{bad}_{\mathsf{tag.i.j}}$, the probability that a given h-inconsistent C implies "the correct $[s_{i^*}]_{\mathbb{Z}} \bmod P$" (which is a prerequisite for non-rejection), is at most 2^{-4k}.

g-inconsistent ciphertexts. Now assume that $\gamma_1 = \gamma_2 = 0 \bmod N^2$. Since C is inconsistent, $r_1 \neq r_2 \bmod |\mathbb{G}_{\mathsf{rnd}}|$. We may call such ciphertexts *g-inconsistent.* Recall that $|\mathbb{G}_{\mathsf{rnd}}| = (P-1)(Q-1)/4$, where P, Q are safe primes. Hence, without loss of generality, we can assume that $r_1 \neq r_2 \bmod (P-1)/2$, where $(P-1)/2$ is prime. We now claim that the subterm $g_1^{r_1[s_1]_{\mathbb{Z}}} g_2^{r_2[s_2]_{\mathbb{Z}}}$ of (13) is (up to a small statistical defect) independently and uniformly random modulo $(P-1)/2$. This can be seen as in the discussion after Game $6.i.2$, where the value

$$X = \mathrm{dlog}_g\left(g_1^{r_1[s_1]_{\mathbb{Z}}} \cdot g_2^{r_2[s_2]_{\mathbb{Z}}}\right)$$

is seen as essentially uniform. In particular, pk contains a linear equation that is independent of X, and the information about X from the KDM challenges is suitably blinded by the b_j-components of the s_j. (The difference to Game 6.i.2 is that the r_i in our case are adversarially chosen, and so could be equal modulo a factor of $|\mathbb{G}_{\mathsf{rnd}}|$. Thus, we can only conclude linear independence modulo $(P-1)/2$.) Since a ciphertext is rejected when $\widetilde{C}/\widehat{C} \notin \mathbb{G}_{\mathsf{rnd}}$, A has to (information-theoretically) guess the right value of $X \bmod (P-1)/2$ to achieve non-rejection. However, $X \bmod (P-1)/2$ is essentially independent of A's view, so A's chance to produce a g-inconsistent ciphertext that is not rejected is no more than $|\mathbb{G}_{\mathsf{rnd}}|^{-1} \cdot 2^{\varepsilon} \leq 2^{-4k}$.

Summarizing, and using a union bound, we obtain that

$$\Pr\left[\mathsf{bad}_{\mathsf{query.i.j}} \mid \neg\mathsf{bad}_{\mathsf{tag.i.j}}\right] \;\leq\; q'(k) \cdot 2^{-4k} \;=\; \mathbf{O}(2^{-3k})$$

for the number $q'(k)$ of A's decryption queries. Combining with (12) shows the lemma. We stress that in this proof, it appears that several bounds have been chosen too conservatively. In particular, we arrive at an error bound that is significantly smaller than, e.g., $\mathbf{O}(2^{-k})$. These extra "entropy cushions" are used in the multi-user case. □

4.3 Security proof (multi-user case)

Theorem 4.3. *Assume the DCR assumption holds in $\mathbb{Z}_{N^3}$, the DDH assumption holds in $\mathbb{G}_{\mathsf{rnd}}$,* LAF *is an LAF,* Sig *is a strongly OT-EUF-CMA secure signature scheme, and* (E,D) *is a key-unique IND-CPA secure symmetric encryption scheme. Then* PKE *is n-CIRC-CCA-secure for every polynomial $n = n(k)$.*

Proof sketch. The proof is very similar to the proof of Theorem 4.1. The way we achieve multi-user KDM security is to have n "virtual" secret keys s^i that are set up as

$$s^i \;=\; (s_1^i, s_2^i) \;=\; (s_1, s_2) + (\hat{s}_1^i, \hat{s}_2^i) \tag{16}$$

(with component-wise addition) for uniformly chosen $\hat{s}^i = (\hat{s}_1^i, \hat{s}_2^i) \leftarrow \mathcal{M}^2$. Intuitively, the $\hat{s}^i$ blind a single $s = (s_1, s_2) \in \mathcal{M}^2$ in several instances. While the $\hat{s}^i$ are all uniform, however, we choose the $s_j = (a_j, b_j, c_j) \in \mathcal{M}$ with "small" components. Concretely, we pick $(a_j, b_j, c_j) \leftarrow \mathbb{Z}_{3k} \times \mathbb{Z}_p \times \mathbb{Z}_{N/4}$ and embed s_j into $\mathcal{M}$ in the natural way. This choice guarantees that $[s_j^i]_{\mathbb{Z}} = [s_j]_{\mathbb{Z}} + [\hat{s}_j^i]_{\mathbb{Z}}$ and $[s_j^i]_{\mathbb{Z}_p^{\mathfrak{n}}} = [s_j]_{\mathbb{Z}_p^{\mathfrak{n}}} + [\hat{s}_j^i]_{\mathbb{Z}_p^{\mathfrak{n}}}$, except with probability $\mathbf{O}(2^{-k})$. Intuitively, the $\hat{s}^i$ can be known to A at all times, while we will try to argue that the information A has about s is very limited.

We will now go through the proof of Theorem 4.1, and sketch the necessary modifications for the multi-user case. Generally, we assume a setup of keys as in (16) (which guarantees independently uniform s^i). **Games** 1 **to** 6.i.2 are as with Theorem 4.1, where the changes apply of course to KDM queries under all public keys. The corresponding reductions to DCR, DDH, the indistinguishability of LAF, and the security of Sig apply almost verbatim. The only noteworthy change occurs in the justification of the change from Game 6.i.2.

Here, we have to argue that A obtains no useful information about the $s_j^i \bmod |\mathbb{G}_{\mathsf{rnd}}|$ from all public keys pk^i, all KDM ciphertexts, and all decryption queries. First, each $pk^i = g_1^{[s_1^i]_{\mathbb{Z}}} g_2^{[s_2^i]_{\mathbb{Z}}}$ yields exactly one linear equation

$$\alpha_1[s_1^i]_{\mathbb{Z}} + \alpha_2[s_2^i]_{\mathbb{Z}} \;=\; (\alpha_1[s_1]_{\mathbb{Z}} + \alpha_2[s_2]_{\mathbb{Z}}) + \big(\alpha_1[\hat{s}_1^i]_{\mathbb{Z}} + \alpha_2[\hat{s}_2^i]_{\mathbb{Z}}\big) \bmod |\mathbb{G}_{\mathsf{rnd}}|$$

about $s = (s_1, s_2)$. Obviously, this equation only depends on $\alpha_1[s_1]_{\mathbb{Z}} + \alpha_2[s_2]_{\mathbb{Z}} \bmod |\mathbb{G}_{\mathsf{rnd}}|$, just like in the single-user case. Similarly, since $[s_j^i]_{\mathbb{Z}_p^{\mathfrak{n}}} = [s_j]_{\mathbb{Z}_p^{\mathfrak{n}}} + [\hat{s}_j^i]_{\mathbb{Z}_p^{\mathfrak{n}}}$, all KDM ciphertexts depend only on

$$\omega_1 a_j + \omega_2 b_j + \sum_{i=0}^{\mathfrak{n}-2} \omega_{3+i} c_{j,i} \bmod p$$

(for $(a_j, b_j, c_{j,0}, \ldots, c_{j,\mathfrak{n}-3}) := [s_j]_{\mathbb{Z}_p^{\mathfrak{n}}}$) and the $\hat{s}_j^i$. This equation is fully blinded by $b_j \in \mathbb{Z}_p$. Next, carefully considering the (slightly reduced) entropy in the s_j, we can prove an analog of Lemma 4.2 for the multi-user case. (Because of the reduced entropy, the $\mathbf{O}(2^{-3k})$ bound from the lemma will become $\mathsf{poly} \cdot 2^{-k}$.) Finally, to justify the change from Game 6.i.2, it suffices to note that hence, A's view is essentially independent of

$$r_1\alpha_1[s_1^i]_{\mathbb{Z}} + r_2\alpha_2[s_2^i]_{\mathbb{Z}} \mod |\mathbb{G}_{\mathsf{rnd}}|$$

(where $r_1 \neq r_2$ are the $\mathbb{G}_{\mathsf{rnd}}$-exponents of the considered C_1, C_2).

The remaining **Games** 6.i.3 **to Game** 8 are again as with Theorem 4.1, of course again applied to KDM queries under all public keys. The corresponding reductions to DDH, DCR, and the IND-CPA security of (E, D) apply verbatim. □

Acknowledgements

I would like to thank Hoeteck Wee for pointing out to me the connection between hash proof systems and KDM security. In particular, his interpretation of the schemes of Boneh et al. [10] and Brakerski and Goldwasser [11] were the starting point for this work. Furthermore, I am grateful to Tibor Jager for very useful feedback and discussions. Finally, I am thankful to the Crypto 2012 referees who found an annoying cyclic dependency in an earlier version of the construction.

References

[1] Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. Cryptographic agility and its relation to circular encryption. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 403–422, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[2] Pedro Adão, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *ESORICS 2005*, volume 3679 of *LNCS*, pages 374–396, Milan, Italy, September 12–14, 2005. Springer, Berlin, Germany.

[3] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 527–546, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.

[4] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany.

[5] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 423–444, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[6] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75, St. John's, Newfoundland, Canada, August 15–16, 2003. Springer, Berlin, Germany.

[7] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Berlin, Germany.

[8] Dan Boneh, Emily Shen, and Brent Waters. Strongly unforgeable signatures based on computational Diffie-Hellman. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006*, volume 3958 of *LNCS*, pages 229–240, New York, NY, USA, April 24–26, 2006. Springer, Berlin, Germany.

[9] 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.

[10] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany.

[11] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 1–20, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany.

[12] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 201–218, Providence, RI, USA, March 28–30, 2011. Springer, Berlin, Germany.

[13] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118, Innsbruck, Austria, May 6–10, 2001. Springer, Berlin, Germany.

[14] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.

[15] David Cash, Matthew Green, and Susan Hohenberger. New definitions and separations for circular security. Cryptology ePrint Archive, Report 2010/144, 2010. `http://eprint.iacr.org/`.

[16] Ivan Damgård and Mats Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 119–136, Cheju Island, South Korea, February 13–15, 2001. Springer, Berlin, Germany.

[17] David Galindo, Javier Herranz, and Jorge Villar. Identity-based encryption with master key-dependent message security and applications. Cryptology ePrint Archive, Report 2012/142, 2012. `http://eprint.iacr.org/`.

[18] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.

[19] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 202–219. Springer, Berlin, Germany, March 15–17, 2009.

[20] Dennis Hofheinz. All-but-many lossy trapdoor functions. In *EUROCRYPT 2012*, LNCS, pages 209–227. Springer, Berlin, Germany, 2012.

[21] Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 108–126, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.

[22] Tal Malkin, Isamu Teranishi, and Moti Yung. Efficient circuit-size independent public key encryption with KDM security. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 507–526, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.

[23] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.

[24] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press.

[25] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Germany.