Master-Key KDM-Secure IBE from Pairings

Sanjam Garg[1⋆], Romain Gay[2⋆⋆], Mohammad Hajiabadi[1]

[1] University of California, Berkeley
[2] Cornell Tech, New York

Abstract. Identity-based encryption (IBE) is a generalization of public-key encryption (PKE) by allowing encryptions to be made to user identities. In this work, we seek to obtain IBE schemes that achieve key-dependent-message (KDM) security with respect to messages that depend on the master secret key. Previous KDM-secure schemes only achieved KDM security in simpler settings, in which messages may only depend on user secret keys.

An important motivation behind studying master-KDM security is the application of this notion in obtaining generic constructions of KDM-CCA secure PKE, a primitive notoriously difficult to realize.

We give the first IBE that achieves master-KDM security from standard assumptions in pairing groups. Our construction is modular and combines techniques from KDM-secure PKE based from hash-proof systems, together with IBE that admits a tight security proof in the multi-challenge setting, which happens to be unexpectedly relevant in the context of KDM security. In fact, to the best of our knowledge, this is the first setting where techniques developed in the context of realizing tightly secure cryptosystems have led to a new feasibility result.

As a byproduct, our KDM-secure IBE, and thus the resulting KDM-CCA-secure PKE both enjoy a tight security reduction, independent of the number of challenge ciphertexts, which was not achieved before.

1 Introduction

Key-dependent-message (KDM) security is a strengthening of the classical notion of semantic security, by allowing the adversary to obtain encryptions of messages

⋆ Supported in part from AFOSR Award FA9550-19-1-0200, AFOSR YIP Award, NSF CNS Award 1936826, DARPA and SPAWAR under contract N66001-15-C-4065, a Hellman Award and research grants by the Okawa Foundation, Visa Inc., and Center for Long-Term Cybersecurity (CLTC, UC Berkeley). The views expressed are those of the authors and do not reflect the official policy or position of the funding agencies.

⋆⋆ Supported in part by NSF Award SATC-1704788 and in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2019-19-020700006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

that depend on the secret key. Originally introduced in [?] in the setting of public/private key encryption, KDM security has since found applications in such contexts as fully-homomorphic encryption [?], function secret sharing [?], and more recently in obtaining CCA-secure PKE and designated-verifier non-interactive zero knowledge (NIZK) [?,?].

For a function class $\mathcal{F}$, an encryption scheme is $\mathcal{F}$-KDM secure if no adversary can distinguish between encryptions of $f(sk)$, where $f \in \mathcal{F}$ and sk is the secret key, and encryptions of fixed messages. We know how to obtain KDM-secure encryption for arbitrarily-large classes of functions from various specific assumptions. These results are achieved by first realizing KDM security for a 'minimal' class of functions, e.g., affine functions [?,?,?,?], and then expanding the function family using KDM-amplification theorems [?,?].

KDM security for identity-based encryption (IBE). Alperin-Sheriff and Peikert [?] introduced notions of KDM security in the setting of IBE, under which one may securely encrypt functions of user secret keys (as opposed to the master secret key). In more detail, these notions (that we call *user-KDM security*) extend the semantic-security notion of IBE by allowing the adversary, who has specified a challenge identity id, to ask for encryptions of functions of sk_{id}, the user-specific secret key for id, under id itself. They showed how to build user-KDM secure IBE schemes from the learning with errors (LWE) assumption.

KDM security for master secret keys. In this work, we seek to realize stronger notions of KDM-security for IBE where the adversary may obtain ciphertexts encrypting functions of the master secret key, as opposed to user secret keys. In more detail, we would like the system to retain security even if the adversary obtains encryptions of functions of the master secret key made with respect to "uncorrupted identities." We call this notion master-KDM security (Definition 3).

Why should we care about master-KDM secure IBE? Theoretically speaking, we believe that the notion of master-KDM security for IBE is more natural than the user-KDM notion, as it implies KDM-CCA security for public-key encryption, via the transformation of [?]. In other words, just as IBE implies CCA2 security, master-KDM security implies KDM CCA2 security. In contrast, the weaker user-KDM security does not seem to imply KDM-CCA security.

Generically and simultaneously realizing both KDM security and CCA2 security for public-key encryption has been beset with challenges; thus, also pointing to the challenge in realizing master-KDM IBE. One reason that makes this combination challenging is the fact that KDM-secure PKE schemes typically come with *KDM-oblivious* algorithms, which allow one to sample KDM ciphertexts — without knowledge of the secret key — in such a way that such oblivious ciphertexts will even fool a real decryptor who is in possession of the secret key. This obliviousness property is exactly the intuition behind KDM security: that real KDM ciphertexts may be simulated by publicly samplable ciphertexts.

On the other hand, this KDM-obliviousness property is exactly what destroys CCA security: an adversary may query the decryption oracle on such oblivious ciphertexts to retrieve the secret key.

Previous works showed how to get around the above obstacle against KDM-CCA2 PKE by using NIZK along with CPA-KDM secure PKE [**?**], or more directly from pairing-based assumptions [**?**], or by using the specific properties of hash-proof systems, and hence from DDH, QR and DCR [**?**]. Very recently, the work of [**?**] shows the equivalence of KDM-CPA and KDM-CCA PKE schemes, via non-blackbox constructions that make use of designated-verifier NIZK and garbled circuits. However, it is not yet clear whether the more challenging notion of master-KDM secure IBE is at all realizable in the standard model, and if so from what assumptions. In particular, by trying to build this latter notion from a variety of assumptions, we will have an overarching approach for obtaining KDM-CCA secure PKE.

In summary, in addition to being interesting in its own right, master-KDM secure IBE offers a pathway to realizing new KDM-CCA public-key encryption schemes.

Prior work on master-KDM secure IBE. The observation that master-KDM security for IBE suffices for KDM-CCA secure PKE was first made by [**?**], who gave constructions of *bounded-master-KDM* secure IBE from pairing assumptions. Their constructions, however, only achieve bounded-KDM in the sense that (a) the number of KDM queries should be bounded beforehand, meaning that the sizes of various IBE parameters do grow with this fixed number; and (b) the set of identities against which KDM encryption are allowed should also be chosen beforehand, and not adaptively.

1.1 Our Contributions and Open Problems

In this work, we show constructions of IBE systems satisfying master-KDM security with respect to affine functions from standard assumptions in bilinear groups. Our construction does not suffer from any of the limitations of [**?**], which resulted in bounded master-KDM secure IBE. As a special case, our KDM notion allows us to encrypt the bits as well as the negations of the bits of the master secret key. As shown in [**?**,**?**], KDM security with respect to affine functions is sufficient for obtaining KDM security with respect to any a-priori bounded function family.

At a high level, our construction is obtained via a modular combination of the KDM-secure public-key encryption from [**?**] and a tightly-secure IBE inspired by prior works [**?**,**?**,**?**,**?**]. This connection between tight security and KDM-security is novel to this work and made explicit by abstract definitions that we put forth to capture the modular nature of our construction. Namely, we define a set of properties that our IBE and an abstract underlying public-key encryption must satisfy to obtain KDM security. These properties are naturally fulfilled by prior schemes relying on the standard dual system encryption proof paradigm, introduced by [**?**] in the context of fully-secure IBE; and by KDM-secure encryption

schemes such as [?,?,?] that all rely on hash-proof systems, as unified in [?]. Our IBE is an instance of this new abstract framework with a combination of tightly-secure IBE and the KDM-secure PKE from [?]. As a byproduct, our IBE also achieves *tight* security. Namely, the security loss is independent of the number of challenge ciphertexts, but is only a small constant times the security parameter. In fact, to the best of our knowledge, this is the first setting where techniques developed in the context of realizing tightly secure cryptosystems have led to new feasibility results.

Moreover, our IBE scheme implies KDM-CCA2 secure public-key encryption scheme. One of the benefits of our approach is that we are able to build on the techniques realized in the context of IBE and leverage them in the context of realizing KDM-CCA2 secure schemes. For example, this gives the first *tightly* secure KDM-CCA2 secure public-key encryption scheme. We give more details on our construction in Section 1.2.

Open problems. The main open problem that arises from our work is to build master-KDM secure IBE from other assumptions such as DDH, or factoring-based assumptions. One possible approach toward this is to investigate what properties will allow us to prove the DDH-based IBE schemes of [?,?,?] KDM-secure, and whether those properties are realizable under standard assumptions.

1.2 General Overview of our Construction

Modular construction of IBE from public-key encryption. We start with the observation that most pairing-based IBE schemes are built upon traditional PKE schemes in the following way. The public key of the IBE is the public key of the underlying PKE, plus some extra components that are generated from the latter and some independently generated parameters params. The master secret key of the IBE is simply the secret key of the underlying PKE. The IBE encryption algorithm outputs a ciphertext ct_0, which is an encryption of the plaintext m under the underlying PKE, and extra components that are generated from ct_0, the identity id, and the parameters params.

Put simply, it is possible to generate the public key and a ciphertext of the IBE from an existing public key and ciphertext of the underlying public-key encryption, which is not attribute-based, simply by sampling independent parameters params, and running the algorithms $\mathsf{Expand}_{\mathsf{pk}}$ and $\mathsf{Expand}_{\mathsf{ct}}$. The key generation algorithm of the IBE uses as input the master secret key, which is the secret key of the underlying public-key encryption, and the public key of the IBE.

KDM-secure IBE. For modular IBE, we can hope to achieve KDM-security by replacing the underlying PKE used in existing schemes with a KDM-secure PKE. This approach actually works for what we call *modular IBE* schemes (Definition 4) whose security proof follows the dual system encryption paradigm, originally put forth in [?], in the simplified security model where the adversary gets

$$(\mathsf{PKE.pk}, \mathsf{PKE.sk}) \leftarrow \mathsf{PKE.Setup}(1^\lambda)$$

$$\mathsf{IBE.msk} := \mathsf{PKE.sk}$$

$$\mathsf{IBE.pk} := (\mathsf{pk}_0, \mathsf{pk}_1)$$
$$\text{with } \mathsf{pk}_0 := \mathsf{PKE.pk}, \mathsf{pk}_1 := \mathsf{Expand}_{\mathsf{pk}}(\mathsf{pk}_0, \mathsf{params})$$

$$\mathsf{IBE.Enc}(m, \mathsf{id}) := (\mathsf{ct}_0, \mathsf{ct}_1)$$
$$\text{with } \mathsf{ct}_0 := \mathsf{PKE.Enc}(\mathsf{PKE.pk}, m), \mathsf{ct}_1 := \mathsf{Expand}_{\mathsf{ct}}(\mathsf{params}, \mathsf{ct}_0, \mathsf{id}).$$

Fig. 1. Modular IBE. Here, $(\mathsf{PKE.Setup}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ is a public-key encryption, and params are parameters that are generated independently.

to see only *one* challenge ciphertext. Note that in the standard IND-CPA security game, one challenge ciphertext is equivalent to many challenge ciphertexts, using a standard hybrid argument (this is valid for any public-key encryption). However, this argument fails for KDM security, since the plaintexts depend on the secret key. We describe the construction based on the dual system methodology, which is instructive despite the fact that its security only handles one challenge ciphertext. Next, we explain how to modify this first attempt and get KDM security with many challenge ciphertexts.

1.3 First Attempt: Dual System Encryption

Dual system encryption. For schemes using the dual system encryption paradigm, the security proof makes use of the fact that the master secret key of the IBE consists of two independent components: $\mathsf{IBE.msk} = \mathsf{PKE.sk} := (\mathsf{msk}_{\mathsf{N}}, \mathsf{msk}_{\mathsf{SF}})$, typically referred to as normal and semi-functional components, respectively. The corresponding public key $\mathsf{PKE.pk}$ (and thus, honestly generated ciphertexts) only depends on the normal component $\mathsf{msk}_{\mathsf{N}}$. The security proof consists of a sequence of hybrid games, where the first transition switches the distribution of the challenge ciphertext to a semi-functional distribution, where the ciphertext now also depends on the component $\mathsf{msk}_{\mathsf{SF}}$. In the next step of the security proof, the distribution of the functional secret keys is changed so that they do not depend on the semi-functional component $\mathsf{msk}_{\mathsf{SF}}$. This change of distribution should not be noticeable to the adversary, which implies that these semi-functional keys still correctly decrypt honestly generated ciphertext. However, they fail to decrypt the challenge ciphertext, which means the simulator can leverage the adversary's ability to break semantic security on the challenge ciphertext. At this point, the security relies on a statistical argument: the component $\mathsf{msk}_{\mathsf{SF}}$, which only appears in the challenge ciphertext, is used to mask the plaintext.

Making IBE KDM-secure, for one challenge ciphertext. As in prior works [**?**,**?**,**?**], we consider KDM-security for the class of affine functions, where

Hybrid game:	ct	sk
IND-CPA security game	N	N
game 1	SF	N
game 2	SF	SF

sk \ ct	N	SF
N	✓	-
SF	✓	✗

Fig. 2. The dual system encryption proof paradigm. The leftmost table depicts the sequence of hybrid games used in the security proof, starting with the original IND-CPA security game, and the rightmost table illustrates when decryption succeeds, depending on whether the ciphertexts and keys are normal (N) or semi-functional (SF). We denote by ct here the challenge ciphertext, and by sk the user secret keys generated in the security game.

the message space is a group $\mathbb{G}$ of order p, generated by g, and the secret key is of the form $\mathsf{msk} := (g_1, \ldots, g_\ell) \in \mathbb{G}^\ell$, an encoding of an ℓ-bit string. The adversary can choose an affine combination $(w_1, \ldots, w_\ell) \in \mathbb{Z}_p^\ell$ and $M \in \mathbb{G}$, and obtain an encryption of the message $\prod_{i\in[\ell]} g_i^{w_i} \cdot M$. For convenience, we use bracket notations, where for any exponent $a \in \mathbb{Z}_p$, we denote by $[a] := g^a$. With this notation, we can write $\mathsf{msk} := [\mathbf{k}] \in \mathbb{G}^\ell$, and the adversary gets an encryption of $[\mathbf{k}^\top \mathbf{w} + m]$. For simplicity, we focus on the single instance case, where only one public key, secret key pair is generated, and we consider the simplified security model where the adversary gets to see only *one* challenge ciphertext. We will see how to remove that restriction later, thereby allowing the adversary to obtain multiple challenge ciphertexts for many identities and affine combinations of its choice.

We take a modular IBE where the underlying PKE is compatible with the dual system encryption methodology, that is, a PKE whose ciphertext can be turned to a semi-functional distribution, even given the secret key. Thus, the secret key can be used to simulate the user secret keys queried by the adversary during the security proof, as well as the challenge ciphertext, whose underlying plaintext may depend on the secret key. Then, user secret keys of the IBE are turned to semi-functional, following the standard dual system encryption paradigm, except that this must be done with encryption of key-dependent messages. At this point, user secret keys can be generated only knowing the normal component of the secret key msk_N, as opposed to the full master secret key. Finally, we rely on the KDM security of the underlying PKE, which must hold even if the value msk_N is revealed to the adversary. This value permits to simulate semi-functional keys. This is achieved using a statistical argument which only involves msk_SF (and not msk_N). Indeed, since the value msk_SF only shows up in the challenge ciphertext, it can be used to hide the plaintext, and conclude the security proof. As it turns out, most existing KDM-secure encryption, such as [**?**,**?**,**?**] can be shown to satisfy these additional properties (and in fact, as noted in [**?**], all PKE based on hash-proof systems).

We show a concrete exposition of this technique by combining the modular IBE from [**?**] and the KDM-secure PKE from [**?**], both of which rely on prime-order groups, and thus are compatible. This construction gives some insight and

prepares for the IBE satisfying full-fledged KDM security, where the adversary gets to see many challenge ciphertexts, that we present later.

Chen et. al. Identity-Based Encryption. We illustrate the dual system encryption methodology with the IBE from [**?**]. We use a pairing group $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are all cyclic groups of prime order p, generated respectively by g_1, g_2, and $e(g_1, g_2)$, where e is a non-degenerate bilinear map, that is, for all $a, b \in \mathbb{Z}_p$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$. We use bracket notations, where for all exponents $a \in \mathbb{Z}_p$ and all groups $s \in \{1, 2, T\}$, we denote by $[a]_s$ the group element g_s^a. We generalize this notation for any matrix $\mathbf{A} = \begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ & \ddots & \\ a_{m,1} & \dots & a_{m,n} \end{pmatrix} \in \mathbb{Z}_p^{m \times n}$, that is, we denote by $[\mathbf{A}]_s$ the matrix of group elements $\begin{pmatrix} g_s^{a_{1,1}} & \dots & g_s^{a_{1,n}} \\ & \ddots & \\ g_s^{a_{m,1}} & \dots & g_s^{a_{m,n}} \end{pmatrix} \in \mathbb{G}_s^{m \times n}$.

The IBE from [**?**] is a modular IBE that uses the following underlying public-key encryption, which is essentially Damgård El-Gamal encryption [**?**], with message space $\mathbb{G}_T$.

- $\mathsf{PKE.Setup}(1^\lambda)$: $\mathbf{a}, \mathbf{k} \leftarrow_\mathsf{R} \mathbb{Z}_p^2$, return $\mathsf{pk} := ([\mathbf{a}]_1, [\mathbf{a}^\top \mathbf{k}]_T)$, and $\mathsf{sk} := \mathbf{k}$.
- $\mathsf{PKE.Enc}(\mathsf{pk}, M \in \mathbb{G}_T)$: $r \leftarrow_\mathsf{R} \mathbb{Z}_p$, return $([\mathbf{a}r]_1, [\mathbf{a}r^\top \mathbf{k}]_T \cdot M)$.
- $\mathsf{PKE.Dec}(\mathsf{pk}, \mathsf{ct}, \mathbf{k})$: parse $\mathsf{ct} := ([\mathbf{c}]_1 \in \mathbb{G}_1^2, [c']_T \in \mathbb{G}_T)$, and return $[c']_T / e([\mathbf{c}^\top \mathbf{k}]_1, [1]_2)$.

The rest of the IBE parameters are computed as follows. Note that the identity space is $\mathbb{Z}_p$.

- $\mathsf{params} := (\mathbf{W}_0, \mathbf{W}_1)$, where $\mathbf{W}_0, \mathbf{W}_1 \leftarrow_\mathsf{R} \mathbb{Z}_p^{2 \times 2}$.
- $\mathsf{Expand}_\mathsf{pk}(\mathsf{pk}_0)$: given $\mathsf{pk}_0 := ([\mathbf{a}]_1, [\mathbf{a}^\top \mathbf{k}]_T)$, samples $\mathbf{b} \leftarrow_\mathsf{R} \mathbb{Z}_p^2$, and returns $\mathsf{pk}_1 := ([\mathbf{W}_0 \mathbf{a}]_1, [\mathbf{W}_1 \mathbf{a}]_1, [\mathbf{W}_0^\top \mathbf{b}]_2, [\mathbf{W}_1^\top \mathbf{b}]_2)$.
- $\mathsf{Expand}_\mathsf{ct}(\mathsf{params}, \mathsf{ct}_0, \mathsf{id} \in \mathbb{Z}_p)$: given $\mathsf{ct}_0 := ([\mathbf{c}]_1, [c']_T)$, returns $\mathsf{ct}_1 := [(\mathbf{W}_0 + \mathsf{id}\mathbf{W}_1)\mathbf{c}]_1$.
- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{pk}, \mathsf{id} \in \mathbb{Z}_p)$: samples $s \leftarrow_\mathsf{R} \mathbb{Z}_p$, and returns $\mathsf{sk}_\mathsf{id} := ([\mathbf{b}s]_2, [\mathbf{k} + (\mathbf{W}_0 + \mathsf{id}\mathbf{W}_1)^\top \mathbf{b}s]_2)$.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}_\mathsf{id})$: parse $\mathsf{ct} := (\mathsf{ct}_0, \mathsf{ct}_1)$ with $\mathsf{ct}_0 := ([\mathbf{c}]_1, [c']_T)$, $\mathsf{ct}_1 := [\mathbf{c}_1]_1$, $\mathsf{sk}_\mathsf{id} := ([\mathbf{d}]_2, [\mathbf{d}']_2)$ and return $[c']_T \cdot e([\mathbf{c}_1]_1^\top, [\mathbf{d}]_2) / e([\mathbf{c}]_1^\top, [\mathbf{d}']_2)$.

We know there is an orthogonal vector $\mathbf{a}^\perp \in \mathbb{Z}_p^2$, such that $\mathbf{a}^\perp \neq \mathbf{0}$, and $\mathbf{a}^\top \mathbf{a}^\perp = 0$. Assuming $\mathbf{a} \leftarrow_\mathsf{R} \mathbb{Z}_p^2$ is different from the zero vector $\mathbf{a} \neq \mathbf{0}$, which happens with all but negligible probability over the choice of $\mathbf{a} \leftarrow_\mathsf{R} \mathbb{Z}_p^2$, we have that $(\mathbf{a} | \mathbf{a}^\perp)$ is a basis of $\mathbb{Z}_p^2$, and we can write $\mathbf{k} := \mathsf{msk}_\mathsf{N} + \mathsf{msk}_\mathsf{SF}$, where msk_N, the normal component, is of the form $k_0 \cdot \mathbf{a}$ with $k_0 \leftarrow_\mathsf{R} \mathbb{Z}_p$, and msk_SF, the semi-functional component, is of the form $k_1 \cdot \mathbf{a}^\perp$ with $k_1 \leftarrow_\mathsf{R} \mathbb{Z}_p$. That is, msk_N (resp. msk_SF) is the projection of the vector $\mathbf{k}$ onto the vector $\mathbf{a}$ (resp. onto $\mathbf{a}^\perp$). This way, the public key only depends on msk_N, since it only contains $[\mathbf{a}^\top \mathbf{k}]_T$, and $\mathbf{a}^\top \mathbf{a}^\perp = 0$.

The semi-functional distribution of ciphertexts is illustrated in fig. 3. We can change the distribution of the challenge ciphertext using the DDH assumption in $\mathbb{G}_1$, which says that $([\mathbf{a}]_1, [\mathbf{a}r]_1)$ is computationally indistinguishable from $([\mathbf{a}]_1, [\mathbf{u}]_1)$, where $\mathbf{a}, \mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$, and $r \leftarrow \mathbb{Z}_p$. Otherwise stated, DDH is a subgroup membership problem, which states that it is hard to distinguish a vector of group elements that is proportional to $[\mathbf{a}]$, from a uniformly random vector over $\mathbb{G}_1$. The consequence is that the semi-functional ciphertext depends on the component $\mathsf{msk}_{\mathsf{SF}}$, since the vector $[\mathbf{u}]_1$ that is part of the ciphertext (see fig. 3) is not orthogonal to $\mathbf{a}^\perp$ (with all but negligible probability), unlike $\mathbf{a}$.

Normal:	Semi-functional:
$\mathsf{ct}_0 := ([\mathbf{a}r], [\mathbf{a}r^\top \mathbf{k}]_T \cdot M)$	$\mathsf{ct}_0 := ([\mathbf{u}], [\mathbf{u}^\top \mathbf{k}]_T \cdot M)$

Fig. 3. Normal and semi-functional distributions for the challenge ciphertext. Here, $\mathbf{a}, \mathbf{k}, \mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$, and $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$. The rest of the ciphertext is computed from ct_0 using $\mathsf{Expand}_{\mathsf{ct}}$ and params.

Then, in [**?**], the distribution of all the user secret keys generated in the security game is changed, so that they depend on $\mathsf{msk}_{\mathsf{N}}$, but are independent of $\mathsf{msk}_{\mathsf{SF}}$. Namely, all the keys are switched from $\mathsf{KeyGen}(\mathbf{k}, \mathsf{pk}, \mathsf{id})$ to $\mathsf{KeyGen}(\mathsf{msk}_{\mathsf{N}}, \mathsf{pk}, \mathsf{id})$. Finally, we can use the component $\mathsf{msk}_{\mathsf{SF}}$ as a one-time pad to mask the plaintext in the challenge ciphertext.

We observe that if we trade the underlying public-key encryption used here, namely Damgård ElGamal [**?**], for the KDM-secure public-key encryption from [**?**], we obtain an overall IBE that enjoys KDM-security. Roughly speaking, the dual system encryption is compatible with the proof techniques used in [**?**].

Boneh et. al. KDM-secure public-key encryption. We now recall the public-key encryption from [**?**], which is KDM-secure for the class of affine functions. For simplicity, we focus on the single instance case, where only one public key, secret key pair is generated.

It is a modification of the Damgårg ElGamal encryption scheme where the key space is changed to $\mathbb{G}_T^\ell$ instead of $\mathbb{Z}_p^2$, so that affine combinations of the secret key $[\mathbf{k}]_T \in \mathbb{G}_T^\ell$ belong to the message space. To preserve correctness of the encryption scheme, the authors of [**?**] choose a secret key $[\mathbf{k}]_T$ where the discrete logarithm $\mathbf{k}$ can be obtained efficiently, and decryption can proceed as for the Damgård ElGamal encryption scheme. Namely, $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$. To have enough entropy in the secret key, it is necessary to take a dimension $\ell = \Theta(\log p)$. The dimension of the vector $[\mathbf{a}]_1$ which is part of the public key is modified accordingly. The security proof follows a similar pattern as outlined previously: the ciphertexts are switched to semi-functional, using a computational assumption that holds even when the secret key is revealed. Then the plaintexts are made independent of the key, using a perfect statistical argument. Finally, $\mathsf{msk}_{\mathsf{SF}}$, the semi-functional component of $\mathbf{k}$, is used to mask the plaintext, using a statistical argument.

Namely, we use the Left Over Hash Lemma [**?**] with entropy source $\mathsf{msk}_{\mathsf{SF}}$. An overview is given fig. 4.

Hybrid game:	challenge ct:	explanation
KDM security game	$[\mathbf{a}r]_1, [\mathbf{k}^\top \mathbf{a}r]_T \cdot [\mathbf{k}^\top \mathbf{w} + m]_T$	the adversary chooses an affine combination $\mathbf{w} \in \mathbb{Z}_p^\ell$, $[m] \in \mathbb{G}$
Game 1	$[\mathbf{u}]_1, [\mathbf{k}^\top \mathbf{u}]_T \cdot [\mathbf{k}^\top \mathbf{w} + m]_T$	ct is switched to semi-functional using DDH in $\mathbb{G}_1$
Game 2	$[\mathbf{u} - \mathbf{w}]_1, [\mathbf{k}^\top \mathbf{u}]_T \cdot [m]_T$	statistical change, the encrypted plaintext is not key-dependent
Game 3	$[\mathbf{u} - \mathbf{w}]_1, [\mathbf{k}^\top \mathbf{u}]_T$	LOHL, with seed $\mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$

Fig. 4. KDM security proof of [**?**]. Here, $[\mathbf{a}]_1 \leftarrow_{\mathsf{R}} \mathbb{G}_1^\ell$ is part of pk, and the secret key is $[\mathbf{k}]_T$ with $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$, $\ell = \Theta(\log p)$, and $\mathbf{w} \in \mathbb{Z}_p^\ell$, $[m] \in \mathbb{G}$ are chosen by the adversary. The randomness $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, $\mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$ is sampled upon creation of the challenge ciphertext. LHOL stands for Left Over Hash Lemma [**?**].

Combining Boneh et. al. PKE with Chen et. al. IBE.

We change the IBE from [**?**], which uses as an underlying PKE Damgård ElGamal encryption scheme, to a similar modular IBE which uses the Boneh et. al. KDM-secure PKE instead. Namely, we have: $\mathbf{a} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$, and $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$ for $\ell = \Theta(\log p)$, $\mathsf{pk} := ([\mathbf{a}]_1, [\mathbf{k}^\top \mathbf{a}]_T)$, and $\mathsf{sk} := [\mathbf{k}]_T$. The parameters are modified accordingly: $\mathsf{params} := (\mathbf{W}_0, \mathbf{W}_1)$ where $\mathbf{W}_0, \mathbf{W}_1 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{2\times\ell}$.

This way, we can prove KDM security of the IBE simply by following the first steps of the KDM security proof of [**?**]: the challenge ciphertext is switched to semi-functional, then the functional keys are switched to semi-functional; the plaintext is made independent of the master secret key, using a hash proof system style statistical argument; finally we use the Left Over Hash lemma with entropy source $\mathsf{msk}_{\mathsf{SF}}$ to mask the plaintext in the challenge ciphertext. The security proof is illustrated in fig. 5.

Dual system encryption, in more details. The proof of Chen et. al. IBE (and more generally, of any scheme using the dual system encryption methodology) crucially relies on the fact that there is only one challenge ciphertext. Recall that this is equivalent to many challenge ciphertexts for IND-CPA public-key IBE, however, this doesn't hold for KDM-secure IBE.

Indeed, to switch the functional keys to semi-functional, the proof uses an underlying statistical argument that is only valid in the presence of one challenge ciphertext. Namely, the distribution of each functional key is switched to a pseudo distribution, one by one. Doing so releases some entropy from the parameters params in the pseudo functional key, while that entropy remains hidden from all others keys, and from the public key, but not from the challenge ciphertext. At this point, the security relies on the fact the identity of the pseudo key

Game:	challenge c_0:	$\mathsf{sk}_{\mathsf{id}}$	explanation
Game 0	$[\mathbf{a}r]_1, [\mathbf{k}^\top \mathbf{a}r]_T \cdot [\mathbf{k}^\top \mathbf{w} + m]_T$	$\mathsf{KeyGen}(\mathsf{mpk}, [\mathbf{k}]_T, \mathsf{id})$	the adversary chooses an affine combination $\mathbf{w} \in \mathbb{Z}_p^\ell$, $[m] \in \mathbb{G}$
Game 1	$[\mathbf{u}]_1, [\mathbf{k}^\top \mathbf{u}]_T \cdot [\mathbf{k}^\top \mathbf{w} + m]_T$	$\mathsf{KeyGen}(\mathsf{mpk}, [\mathbf{k}]_T, \mathsf{id})$	ct is switched to semi-functional using DDH in $\mathbb{G}_1$
Game 2	$[\mathbf{u}]_1, [\mathbf{k}^\top \mathbf{u}]_T \cdot [\mathbf{k}^\top \mathbf{w} + m]_T$	$\mathsf{KeyGen}(\mathsf{mpk}, [\mathsf{msk}_{\mathsf{N}}]_T, \mathsf{pk}, \mathsf{id})$	$\mathsf{sk}_{\mathsf{id}}$ are switched to semi-functional
Game 3	$[\mathbf{u} - \mathbf{w}]_1, [\mathbf{k}^\top \mathbf{u}]_T \cdot [m]_T$	$\mathsf{KeyGen}(\mathsf{mpk}, [\mathsf{msk}_{\mathsf{N}}]_T, \mathsf{pk}, \mathsf{id})$	statistical change, the encrypted plaintext is not key-dependent
Game 4	$[\mathbf{u} - \mathbf{w}]_1, [\mathbf{k}^\top \mathbf{u}]_T$	$\mathsf{KeyGen}(\mathsf{mpk}, [\mathsf{msk}_{\mathsf{N}}]_T, \mathsf{pk}, \mathsf{id})$	LOHL, with seed $\mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$ and entropy source $\mathsf{msk}_{\mathsf{SF}}$

Fig. 5. KDM security proof of the IBE. Here, $[\mathbf{a}]_1 \leftarrow_{\mathsf{R}} \mathbb{G}_1^\ell$ is part of mpk, and the secret key is $[\mathbf{k}]_T$ with $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$, $\ell = \Theta(\log p)$, and $\mathbf{w} \in \mathbb{Z}_p^\ell$, $[m] \in \mathbb{G}$ are chosen by the adversary. The randomness $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, $\mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$ is sampled upon creation of the challenge ciphertext. Recall that $\mathsf{msk} := [\mathbf{k}]_T$, $\mathbf{k} := \mathsf{msk}_{\mathsf{N}} + \mathsf{msk}_{\mathsf{SF}}$, where $\mathsf{msk}_{\mathsf{N}}$, and $\mathsf{msk}_{\mathsf{SF}}$ are the projections of $\mathbf{k}$ onto $\mathbf{a}$ and $\mathbf{A}^\perp$, respectively.

and semi-functional ciphertext don't match, using a statistical *one-time* argument. This argument fails for many semi-functional ciphertexts, the presence of which is unavoidable in the KDM security proof.

More concretely, the pseudo keys in Chen et. al. IBE are of the form: $([\mathbf{v}]_2, [\mathbf{k} + (\mathbf{W}_0 + \mathsf{id}\mathbf{W}_1)^\top \mathbf{v}]_2)$, for a uniformly random $[\mathbf{v}]_2 \leftarrow_{\mathsf{R}} \mathbb{G}_2$, instead of $[\mathbf{v}]_2 := [\mathbf{b}s]_2$ with $s \leftarrow_{\mathsf{R}} \mathbb{Z}_p$ in normal keys. This releases entropy from $\mathbf{W}_0, \mathbf{W}_1 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell \times 2}$ that is not revealed from the public key which only contains $([\mathbf{W}_0\mathbf{a}]_1, [\mathbf{W}_1\mathbf{a}]_1, [\mathbf{W}_0^\top \mathbf{b}]_2, [\mathbf{W}_1^\top \mathbf{b}]_2)$. Namely, the component from these matrices that is orthogonal to $\mathbf{a}$ and $\mathbf{b}$ can be used to perform a statistical one-time argument with the semi-functional challenge ciphertext, which contains: $([\mathbf{u}]_1, [(\mathbf{W}_0 + \mathsf{id}^\star \mathbf{W}_1)\mathbf{u}]_1)$ for $[\mathbf{u}]_1 \leftarrow_{\mathsf{R}} \mathbb{G}_1^\ell$. This essentially uses the fact that the map $\mathsf{id} \to \mathbf{W}_0 + \mathsf{id}\mathbf{W}_1$ is a pairwise independent hash function, aka 2-universal hash function. This argument fails when there are several challenge ciphertexts, each of which associated with a different identity.

1.4 Final Attempt: Handling Many Challenge Ciphertexts

To prove KDM security, we need to consider many challenge ciphertexts simultaneously. Ultimately, in the security proof, we use the entropy from the semi-functional component $\mathsf{msk}_{\mathsf{SF}}$ of the master secret key to hide the plaintexts in all the challenge ciphertexts. Since there number of challenge ciphertexts is unbounded, this will require a computational argument, as opposed to the statistical argument used previously, in the single challenge ciphertext setting. To

that end, we first need to make the user secret keys and the plaintexts in the challenge ciphertexts independent from $\mathsf{msk}_{\mathsf{SF}}$. As explained previously, to do so, we make use of the fact that the plaintext in semi-functional challenge ciphertexts can be made independent from the master secret key, statistically (this is the transition from game 2 to game 3 in fig. 5). Thus, to make the plaintext independent from msk in *all* challenge ciphertexts, we need to switch them to semi-functional distribution *all* at the same time. More details are provided in Section 2.1.

Traditional dual system encryption, as explained previously, is incapable of handling many semi-functional challenge ciphertext at once. Instead, we adapt techniques from [?,?,?] that build IBE where the security proof can handle many challenge ciphertexts at once. These techniques, which builds upon [?,?,?], were developed for a whole different purpose than KDM security, namely, they were used to obtain IBE that are secure in the multi-challenge setting, where the security loss is independent of the number of challenge ciphertexts. These tight security reductions yield shorter concrete parameters for a given security level.

2 Preliminaries

2.1 Pairing groups

Let GGen be a PPT algorithm that on input the security parameter 1^λ, returns a description $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2, e)$ where for all $s \in \{1, 2, T\}$, $\mathbb{G}_s$ is a cyclic group of order p for a 2λ-bit prime p. $\mathbb{G}_1$ and $\mathbb{G}_2$ are generated by P_1 and P_2 respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable (non-degenerate) bilinear map. Define $P_T := e(P_1, P_2)$, which is a generator of $\mathbb{G}_T$, of order p. We use implicit representation of group elements. For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_p$, define $[a]_s = a \cdot P_s \in \mathbb{G}_s$ as the implicit representation of a in G_s. More generally, for a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_p^{n \times m}$ we define $[\mathbf{A}]_s$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}_s$:

$$[\mathbf{A}]_s := \begin{pmatrix} a_{11} \cdot P_s & \dots & a_{1m} \cdot P_s \\ & & \\ a_{n1} \cdot P_s & \dots & a_{nm} \cdot P_s \end{pmatrix} \in \mathbb{G}_s^{n \times m}.$$

Given $[a]_1$ and $[b]_2$, one can efficiently compute $[a \cdot b]_T$ using the pairing e. For matrices $\mathbf{A}$ and $\mathbf{B}$ of matching dimensions, define $e([\mathbf{A}]_1, [\mathbf{B}]_2) := [\mathbf{AB}]_T$. For any matrix $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_p^{n \times m}$, any group $s \in \{1, 2, T\}$, we denote by $[\mathbf{A}]_s + [\mathbf{B}]_s = [\mathbf{A} + \mathbf{B}]_s$.

For any prime p, we define the following distributions. The DDH distribution over $\mathbb{Z}_p^2$: $a \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, output $\mathbf{a} := \binom{1}{a}$.

Definition 1 (DDH assumption). *For any adversary $\mathcal{A}$, any group $s \in \{1, 2, T\}$ and any security parameter λ, let*

$$\mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_s, \mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathcal{PG}, [\mathbf{a}]_s, [\mathbf{a}r]_s)] - \Pr[1 \leftarrow \mathcal{A}(\mathcal{PG}, [\mathbf{a}]_s, [\mathbf{u}]_s)]|,$$

where the probabilities are taken over $\mathcal{PG} \leftarrow_{\mathsf{R}} \mathsf{GGen}(1^\lambda, d)$, $\boldsymbol{a} \leftarrow_{\mathsf{R}} \mathsf{DDH}$, $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, $\mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$, *and the random coins of* $\mathcal{A}$. *We say DDH holds in* $\mathbb{G}_s$ *if for all PPT adversaries* $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_s,\mathcal{A}}(\lambda)$ *is a negligible function of* λ.

Definition 2 (SXDH assumption). *For a pairing group* $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2, e) \leftarrow_{\mathsf{R}} \mathsf{GGen}(1^\lambda)$, *we say SXDH holds in* $\mathcal{PG}$ *if DDH holds in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$.

We define the (ℓ, Q)-fold DDH assumption below. Note that the DDH assumption corresponds to the $(1,1)$-fold DDH assumption.

Lemma 1 (Random self reducibility of DDH). *For any* $\ell, Q \geq 1$, *any PPT adversary* $\mathcal{A}$, *we define:*

$$\begin{aligned}\mathsf{Adv}^{\ell,Q\text{-}\mathsf{DDH}}_{\mathbb{G}_s,\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(\mathcal{PG}, [\mathbf{a}]_s, \{[r_i]_s, [\mathbf{a}r_i]_s\}_{i\in[Q]})] \\ - \Pr[1 \leftarrow \mathcal{A}(\mathcal{PG}, [\mathbf{a}]_s, \{[r_i]_s, [\mathbf{u}_i]_s\}_{i\in[Q]})]|,\end{aligned}$$

where the probabilities are taken over $\mathcal{PG} \leftarrow_{\mathsf{R}} \mathsf{GGen}(1^\lambda, d)$, $\mathbf{a} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$, $r_i \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, $\mathbf{u}_i \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$ *for all* $i \in [Q]$, *and the random coins of* $\mathcal{A}$.

There exists a PPT adversary $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\ell,Q\text{-}\mathsf{DDH}}_{\mathbb{G}_s,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_s,\mathcal{B}}(\lambda).$$

2.2 Entropy Extraction

We give a particular case of the left over hash lemma, that is tailored to our purpose.

Lemma 2 (Leftover hash lemma [?]). *Let* p *be a* 2λ*-bit prime, and* $\ell := 4\lceil \log_2(p) \rceil$. *The following distribution are within* $2^{-\lambda}$ *statistical distance:*

$$(\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{k}^\top\mathbf{a}, \mathbf{k}^\top\mathbf{b}, \mathbf{k}^\top\mathbf{u}) \text{ and } (\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{k}^\top\mathbf{a}, \mathbf{k}^\top\mathbf{b}, r),$$

where $\mathbf{a}, \mathbf{b}, \mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$, $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$, *and* $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$.

2.3 Identity Based Encryption

An Identity Based Encryption for identity space $\mathcal{I}$ and message space $\mathcal{M}$ is a tuple of PPT algorithms:

- $\mathsf{Setup}(1^\lambda)$: on input the security parameter λ, returns a master public key mpk which defines an identity space $\mathcal{I}$, and a master secret key msk.
- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id} \in \mathcal{I}, m \in \mathcal{M})$: returns a ciphertext ct.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id} \in \mathcal{I})$: returns $\mathsf{sk}_{\mathsf{id}}$, a user secret key for identity id.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk})$: deterministic algorithm that returns a message, or a special symbol $\perp$ if it fails.

Correctness. For any security parameter λ, any $\mathsf{id} \in \mathcal{I}$, any message m, $\Pr[\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}_{\mathsf{id}}) = m] = 1$, where the probability is taken over $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)$, $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$.

Remark 1 (Public-key encryption (PKE)). Note that a public-key encryption is a special case of IBE with identity space $\mathcal{I} := \{\varepsilon\}$. Of course, the interesting case of IBE is when $\mathcal{I}$ is of exponential size in the security parameter.

Definition 3 (Master-KDM security). *An IBE scheme* IBE *for identity space* $\mathcal{I}$ *and message space* $\mathcal{M}$ *is said to be KDM-secure for the class of (efficiently computable) functions* $\mathcal{F}$ *if for all PPT adversaries* $\mathcal{A}$*, the following advantage is a negligible function of the security parameter* λ*:*

$$\mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{IBE},\mathcal{A}}(\lambda) := 2 \cdot \left| 1/2 - \Pr\left[b' = b \;\middle|\; \begin{array}{c} b \leftarrow_{\mathsf{R}} \{0,1\} \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{Enc}}(\cdot,\cdot), \mathsf{O}_{\mathsf{KeyGen}}(\cdot)}(\mathsf{mpk}) \end{array} \right] \right|,$$

where the oracle $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, f)$*, on input an identity* $\mathsf{id} \in \mathcal{I}$ *and a function* $f \in \mathcal{F}$*, computes* $y := f(\mathsf{msk}) \in \mathcal{M}$*, returns* $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, f(\mathsf{msk}))$ *if* $b = 0$*, and computes a uniformly random message* $M \leftarrow_{\mathsf{R}} \mathcal{M}$*, and returns* $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, M)$ *if* $b = 1$*; the oracle* $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$*, on input an identity* $\mathsf{id} \in \mathcal{I}$*, returns* $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$*. We require that the identities queried by the adversary to the oracle* $\mathsf{O}_{\mathsf{Enc}}(\cdot,\cdot)$ *are different from the identities queried to* $\mathsf{O}_{\mathsf{KeyGen}}(\cdot)$*. This is in order to avoid trivial attacks, where the adversary can win the game simply using the correctness of the scheme.*

In this paper, as in prior works [**?**,**?**], we consider the class of affine functions, that is, we consider IBE where the message space is a group $\mathbb{G}$ of order p, and $\mathsf{msk} := [\mathbf{k}] \in \mathbb{G}^\ell$ for some integer ℓ. The adversary is allowed to query encryption of affine functions on msk, that is, encryption of messages of the form $[\mathbf{k}^\top \mathbf{w} + \gamma]$, for $\mathbf{w} \in \mathbb{Z}_p^\ell$, $[\gamma] \in \mathbb{G}$ of its choice. In [**?**,**?**], the authors showed that this can be boosted to KDM-security with respect to the class of circuits of a-priori bounded size.

The work of Alperin-Sheriff and Peikert [**?**] gives KDM-secure IBE schemes that only support KDM messages that depend on user secret keys. Also, the work of Galindo et al. [**?**] only achieved a restricted version of master-KDM security, on in which (a) the number of KDM queries is bounded and (b) the oracle $\mathsf{O}_{\mathsf{KeyGen}}$ may only be called on identities that were fixed at the beginning of the game.

3 KDM-Secure IBE from Pairings

In this section we give our construction of KDM-secure IBE from pairing assumptions. To make our construction modular, we first introduce an intermediate primitive (which we call modular IBE), and show that any modular IBE with some specific properties is already KDM secure. We then show how to realize the notion of modular IBE with those required properties.

3.1 Ingredients of Our Construction

We first start with the definition of modular IBE. Informally, we call an IBE scheme modular if it is built upon a PKE scheme in the sense we define below.

Definition 4 (Modular IBE). *We say an IBE* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *for identity space* $\mathcal{I}$ *is modular if there exists a PKE* $(\mathsf{PKE.Setup}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$*, and PPT algorithms* $\mathsf{SampParams}$*,* $\mathsf{Expand}_{\mathsf{pk}}$ *and* $\mathsf{Expand}_{\mathsf{ct}}$ *such that:*

1. $\mathsf{Setup}(1^\lambda)$*:* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Setup}(1^\lambda)$*,* $\mathsf{params} \leftarrow \mathsf{SampParams}(\mathsf{pk}, \mathcal{I})$*,* $\mathsf{pk}' \leftarrow \mathsf{Expand}_{\mathsf{pk}}(\mathsf{params}, \mathsf{pk})$*,* $\mathsf{mpk} := (\mathsf{pk}, \mathsf{pk}', \mathcal{I})$*,* $\mathsf{msk} := \mathsf{sk}$*, returns* $(\mathsf{mpk}, \mathsf{msk})$*.*
2. *For all identities* $\mathsf{id} \in \mathcal{I}$ *and all messages* m*, the following are identically distributed:*

$$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m),$$

and

$$(\mathsf{ct}_0, \mathsf{ct}_1) \text{ where } \mathsf{ct}_0 \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m), \mathsf{ct}_1 \leftarrow \mathsf{Expand}_{\mathsf{ct}}(\mathsf{pk}, \mathsf{params}, \mathsf{ct}_0, \mathsf{id}).$$

In both distributions, we have $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Setup}(1^\lambda)$*,* $\mathsf{params} \leftarrow \mathsf{SampParams}(\mathsf{pk}, \mathcal{I})$*,* $\mathsf{pk}' \leftarrow \mathsf{Expand}_{\mathsf{pk}}(\mathsf{params}, \mathsf{pk})$*, and* $\mathsf{mpk} := (\mathsf{pk}, \mathsf{pk}', \mathcal{I})$*.*

The definition implies that there are two ways to compute the encryption of a message m under identity id: either using Enc on input mpk, id and m; or using the underlying PKE encryption algorithm on input pk and message m, and using the $\mathsf{Expand}_{\mathsf{ct}}$ algorithm that takes as input the PKE ciphertext, pk, and id. These two ways are identically distributed.

We will now define the properties that need to be fulfilled by our IBE and its underlying PKE in order to achieve KDM security. Recall that we denote by $\mathsf{IBE} := (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ the modular IBE, with underlying pke $\mathsf{PKE} := (\mathsf{PKE.Setup}, \mathsf{PKE.Enc}, \mathsf{PKE.Enc})$ whose message space is a group $\mathbb{G}$ of order p, and whose secret key is of the form $\mathsf{sk} := [\mathbf{k}] \in \mathbb{G}^\ell$ for some $\ell \in \mathbb{N}$. We can write $\mathbf{k} := \mathsf{msk}_{\mathsf{N}} + \mathsf{msk}_{\mathsf{SF}} \in \mathbb{Z}_p^\ell$, where $\mathsf{msk}_{\mathsf{N}}$ is the normal component of sk, and $\mathsf{msk}_{\mathsf{SF}}$ is the semi-functional component of sk.

Property 1 (semi-functional encryption). There exists a PPT algorithm $\widetilde{\mathsf{Enc}}$ that takes as input $\mathsf{pk}, \mathsf{sk}, M$ and returns a ciphertext. For all PPT adversaries $\mathcal{A}$, the following advantage is a negligible function of the security parameter λ:

$$\mathsf{Adv}^{\mathsf{SF\text{-}ct}}_{\mathsf{PKE},\mathcal{A}}(\lambda) := 2 \cdot \left| 1/2 - \Pr\left[b' = b \;\middle|\; \begin{array}{c} b \leftarrow_{\mathsf{R}} \{0,1\} \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Setup}(1^\lambda) \\ b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{Enc}}(\cdot)}(\mathsf{pk}, \mathsf{sk}) \end{array} \right] \right|,$$

where the oracle $\mathsf{O}_{\mathsf{Enc}}(M)$, on input a message $M \in \mathbb{G}$, outputs $\mathsf{PKE.Enc}(\mathsf{pk}, M)$ if $b = 0$, or $\widetilde{\mathsf{Enc}}(\mathsf{pk}, \mathsf{sk}, M)$ if $b = 1$. Note that the message M can depend on sk since the latter is given to $\mathcal{A}$.

Property 2 (semi-functional keys). There exists a PPT algorithm $\widetilde{\mathsf{KeyGen}}$ that takes as input $\mathsf{pk}, \mathsf{msk}_\mathsf{N}$ where $\mathsf{sk} = [\mathsf{msk}_\mathsf{N} + \mathsf{msk}_\mathsf{SF}]$ and $(\mathsf{pk}, \mathsf{sk})$ is generated by $\mathsf{Setup}(1^\lambda)$, together with an identity, and outputs a user secret key. We require that for all PPT adversaries $\mathcal{A}$, the following advantage is a negligible function of λ:

$$\mathsf{Adv}^{\mathsf{SF\text{-}sk}}_{\mathsf{IBE},\mathcal{A}}(\lambda) := 2 \cdot \left| 1/2 - \Pr \left[b' = b \left| \begin{array}{c} b \leftarrow_\mathsf{R} \{0,1\} \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{params} \leftarrow \mathsf{SampParams}(\mathsf{pk}, \mathcal{I}) \\ \mathsf{pk}' \leftarrow \mathsf{Expand}_{\mathsf{pk}}(\mathsf{params}, \mathsf{pk}) \\ \mathsf{mpk} := (\mathsf{pk}, \mathsf{pk}', \mathcal{I}), \mathsf{msk} := \mathsf{sk} \\ b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{Enc}}(\cdot,\cdot), \mathsf{O}_{\mathsf{KeyGen}}{}^{(b)}(\cdot)}(\mathsf{mpk}) \end{array} \right. \right] \right|,$$

where the oracle $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, (\mathbf{w}, [m]))$, on input an identity $\mathsf{id} \in \mathcal{I}$, a vector $\mathbf{w} \in \mathbb{Z}_p^\ell$, and a message $[m] \in \mathbb{G}$, computes $\mathsf{ct}_0 \leftarrow \widetilde{\mathsf{Enc}}(\mathsf{pk}, \mathsf{sk}, [\mathbf{k}^\top \mathbf{w} + m])$, $\mathsf{ct}_1 \leftarrow \mathsf{Expand}_{\mathsf{ct}}(\mathsf{pk}, \mathsf{params}, \mathsf{ct}_0, \mathsf{id})$ an returns $(\mathsf{ct}_0, \mathsf{ct}_1)$. The oracle $\mathsf{O}_{\mathsf{KeyGen}}{}^{(b)}(\mathsf{id})$, on input an identity $\mathsf{id} \in \mathcal{I}$, returns $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ if $b = 0$ or $\mathsf{KeyGen}(\mathsf{mpk}, [\mathsf{msk}_\mathsf{N}], \mathsf{id})$ if $b = 1$. Recall that $\mathsf{msk} := [\mathsf{msk}_\mathsf{N} + \mathsf{msk}_\mathsf{SF}]$. We require that the identities queried by $\mathcal{A}$ to $\mathsf{O}_{\mathsf{Enc}}$ are distinct to the identities it queries to $\mathsf{O}_{\mathsf{KeyGen}}$.

Property 3 (KDM security). For all PPT adversaries $\mathcal{A}$, the following advantage is a negligible function of the security parameter λ:

$$\mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{PKE},\mathcal{A}}(\lambda) := 2 \cdot \left| 1/2 - \Pr \left[b' = b \left| \begin{array}{c} b \leftarrow_\mathsf{R} \{0,1\} \\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda), \\ b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{Enc}}(\cdot)}(\mathsf{pk}, [\mathsf{msk}_\mathsf{N}]_T) \end{array} \right. \right] \right|,$$

where the oracle $\mathsf{O}_{\mathsf{Enc}}(\mathbf{w}, [m])$, on input a vector $\mathbf{w} \in \mathbb{Z}_p^\ell$ and a message $[m] \in \mathbb{G}$, outputs $\widetilde{\mathsf{Enc}}(\mathsf{pk}, \mathsf{sk}, [\mathbf{w}^\top \mathbf{k} + m])$ if $b = 0$, or $\widetilde{\mathsf{Enc}}(\mathsf{pk}, \mathsf{sk}, [r])$ for a fresh random $r \leftarrow_\mathsf{R} \mathbb{Z}_p$ if $b = 1$. Recall that $\mathsf{sk} := [\mathbf{k}]$, with $\mathbf{k} := \mathsf{msk}_\mathsf{N} + \mathsf{msk}_\mathsf{SF}$.

3.2 KDM-Secure IBE Construction

We now give our theorem statement for KDM-secure IBE.

Theorem 1 (KDM-security). *Any modular IBE that satisfies properties 1 to 3 is KDM-secure for the class of affine functions.*

Proof. The proof goes through a hybrid argument, starting with game G_0, which is the KDM security experiment from Definition 3. Let $\mathcal{A}$ be a PPT adversary. For any game G, we denote by $\mathsf{Adv}_\mathcal{A}(\mathsf{G})$ the advantage of $\mathcal{A}$ in the game G.

Game G_0. This is the KDM security experiment for the class of affine functions. The message space is a group $\mathbb{G}$ of order p, the master secret key is of the form $[\mathbf{k}] \in \mathbb{G}^\ell$, and the adversary gets access to encryption of affine combinations of the form $[\mathbf{k}^\top \mathbf{w} + m]$, for $\mathbf{w} \in \mathbb{Z}_p^\ell$, $[m] \in \mathbb{G}$ of its choice. Namely, the adversary $\mathcal{A}$ first receives mpk. Then it can adaptively query $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, (\mathbf{w}, [m]))$, to receive

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, [\mathbf{k}^\top \mathbf{w} + m])$ if $b = 0$, $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, [r])$ for a fresh $[r] \leftarrow_{\mathsf{R}} \mathbb{G}$ if $b = 1$. Upon querying $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$, $\mathcal{A}$ receives $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$.

Game G_1. We change the challenge ciphertexts to semi-functional. That is, in game G_0, $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, (\mathbf{w}, [m]))$ computes $[m_0] := [\mathbf{k}^\top \mathbf{w} + m]$, $[m_1] \leftarrow_{\mathsf{R}} \mathbb{G}$, $\mathsf{ct}_0 := \mathsf{PKE.Enc}(\mathsf{pk}, [m_b])$; whereas $\mathsf{ct}_0 := \widetilde{\mathsf{Enc}}(\mathsf{pk}, \mathsf{sk}, [m_b])$ in game G_1, where $\widetilde{\mathsf{Enc}}$ is the PPT algorithm that generates semi-functional ciphertexts (see Property 1). The rest of the challenge ciphertext is computed as $\mathsf{ct}_1 := \mathsf{Expand}_{\mathsf{ct}}(\mathsf{pk}, \mathsf{params}, \mathsf{ct}_0, \mathsf{id})$ in both games. We show there exists a PPT adversary $\mathcal{B}_0$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_0) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_1)| \leq \mathsf{Adv}^{\mathsf{SF\text{-}ct}}_{\mathsf{PKE}, \mathcal{B}_0}(\lambda),$$

which is negligible by Property 1. The reduction $\mathcal{B}_0$ receives $(\mathsf{pk}, \mathsf{sk} := [\mathbf{k}] \in \mathbb{G}^\ell)$ from its own experiment, samples $b \leftarrow_{\mathsf{R}} \{0, 1\}$, $\mathsf{params} \leftarrow \mathsf{SampParams}(\mathsf{pk}, \mathcal{I})$, computes $\mathsf{pk}' \leftarrow \mathsf{Expand}_{\mathsf{pk}}(\mathsf{params}, \mathsf{pk})$, and returns $\mathsf{mpk} := (\mathsf{pk}, \mathsf{pk}', \mathcal{I})$ to $\mathcal{A}$. $\mathcal{B}_0$ can simulate the oracle $\mathsf{O}_{\mathsf{KeyGen}}$ straightforwardly using sk and mpk. To simulate $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, (\mathbf{w}, [m]))$, it computes $[m_0] := [\mathbf{k}^\top \mathbf{w} + m]$, $[m_1] \leftarrow_{\mathsf{R}} \mathbb{G}$, and uses its own encryption oracle on input the message $[m_b]$ to obtain a challenge ciphertext ct_0. Then it computes $\mathsf{ct}_1 \leftarrow \mathsf{Expand}_{\mathsf{ct}}(\mathsf{pk}, \mathsf{params}, \mathsf{ct}_0, \mathsf{id})$, and returns the challenge ciphertext $(\mathsf{ct}_0, \mathsf{ct}_1)$. If $\mathcal{A}$'s guess b' is such that $b' = b$ and identities queried by $\mathcal{A}$ to its encryption oracle are distinct from the identities queried to its key generation oracle, then $\mathcal{B}_0$ returns 1. Otherwise, it returns 0.

Game G_2. We change the user secret keys to semi-functional. That is, in game G_1, $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ returns $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$, whereas it returns $\mathsf{KeyGen}(\mathsf{mpk}, [\mathsf{msk}_{\mathsf{N}}]_T, \mathsf{id})$ in game G_2. Recall that $\mathsf{msk} := [\mathbf{k}]_T$, and $\mathbf{k} := \mathsf{msk}_{\mathsf{N}} + \mathsf{msk}_{\mathsf{SF}}$.

We show there exists a PPT adversary $\mathcal{B}_1$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_1) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_2)| \leq \mathsf{Adv}^{\mathsf{SF\text{-}sk}}_{\mathsf{IBE}, \mathcal{B}_1}(\lambda),$$

which is negligible by Property 2. The reduction $\mathcal{B}_1$ receives mpk from its own experiment, which it forwards to $\mathcal{A}$, and simulates the oracles to $\mathcal{A}$ straightforwardly using its own oracles. Here, we make use of the fact that the the identities queried by $\mathcal{A}$ to its encryption oracle $\mathsf{O}_{\mathsf{Enc}}$ must be distinct to the identities it queries to its key generation oracle $\mathsf{O}_{\mathsf{KeyGen}}$, since this condition must also be fulfilled in the security game from Property 2.

Game G_3. We use the KDM security of the underlying PKE to change the challenge ciphertexts to encryptions of random message $[r] \leftarrow_{\mathsf{R}} \mathbb{G}$. That is, $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, (\mathbf{w}, [m]))$ computes $[m_0] := [\mathbf{w}^\top \mathbf{k} + m]$, $[m_1] \leftarrow_{\mathsf{R}} \mathbb{G}$, $\mathsf{ct}_0 := \widetilde{\mathsf{Enc}}(\mathsf{pk}, \mathsf{sk}, [m_b])$ in game G_3, whereas it computes $\widetilde{\mathsf{Enc}}(\mathsf{pk}, \mathsf{sk}, [r])$ for a fresh random $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$ in game G_3. The rest of the challenge ciphertext is computed as $\mathsf{ct}_1 := \mathsf{Expand}_{\mathsf{ct}}(\mathsf{pk}, \mathsf{params}, \mathsf{ct}_0, \mathsf{id})$ in both games. It is clear that the challenge ciphertexts do not depend on the random bit $b \leftarrow_{\mathsf{R}} \{0, 1\}$ chosen by the experiment in game G_3, since the plaintexts are random, regardless of the value of b. Thus, we have:

$$\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_3) = 0.$$

Now, we show there exists a PPT adversary $\mathcal{B}_3$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_3) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_3)| \leq \mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{PKE},\mathcal{B}_3}(\lambda),$$

which is negligible by Property 3. The reduction $\mathcal{B}_3$ receives $(\mathsf{pk}, [\mathsf{msk}_\mathsf{N}]_T)$ from its own experiment, samples $b \leftarrow_\mathsf{R} \{0,1\}$, $\mathsf{params} \leftarrow \mathsf{SampParams}(\mathsf{pk}, \mathcal{I})$, computes $\mathsf{pk}' \leftarrow \mathsf{Expand}_\mathsf{pk}(\mathsf{params}, \mathsf{pk})$, and returns $\mathsf{mpk} := (\mathsf{pk}, \mathsf{pk}', \mathcal{I})$ to $\mathcal{A}$. When $\mathcal{A}$ queries $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$, $\mathcal{B}_3$ returns $\mathsf{KeyGen}(\mathsf{mpk}, [\mathsf{msk}_\mathsf{N}]_T, \mathsf{id})$. When $\mathcal{A}$ queries $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, (\mathbf{w}, [m]))$, $\mathcal{B}_3$ computes $[m_0] := [m]$, $[m_1] \leftarrow_\mathsf{R} \mathbb{G}$, and queries its own encryption oracle on input $(\mathbf{w}, [m_b])$ to obtain a challenge ciphertext ct_0. Then, $\mathcal{B}_3$ computes $\mathsf{ct}_1 \leftarrow \mathsf{Expand}_\mathsf{ct}(\mathsf{pk}, \mathsf{params}, \mathsf{ct}_0, \mathsf{id})$ and returns the challenge ciphertext $(\mathsf{ct}_0, \mathsf{ct}_1)$ to $\mathcal{A}$. If $\mathcal{A}$'s guess b' is such that $b' = b$ and identities queried by $\mathcal{A}$ to its encryption oracle are distinct from the identities queried to its key generation oracle, then $\mathcal{B}_0$ returns 1. Otherwise, it returns 0.

Overall, we have:

$$\mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{IBE},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SF\text{-}ct}}_{\mathsf{PKE},\mathcal{B}_0}(\lambda) + \mathsf{Adv}^{\mathsf{SF\text{-}sk}}_{\mathsf{IBE},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{PKE},\mathcal{B}_3}(\lambda).$$

□

3.3 Concrete Instantiations

We instantiate the framework presented in the previous section with a modular IBE inspired from [**?**], and the KDM-secure PKE from [**?**]. Both of them rely on prime-order groups, which make them compatible. In Figure 6, we give a description of the [**?**] when adapted to fit pairing groups, and in Figure 7, we show how to extent it in a modular way to obtain a KDM-secure IBE. A concrete description of our IBE is given in Figure 8.

$\underline{\mathsf{PKE.Setup}(1^\lambda)}$:
$\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2, e) \leftarrow \mathsf{GGen}(1^\lambda)$, $\ell := 4\lceil \log_2(p) \rceil$, $\mathbf{a} \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$, $\mathbf{k} \leftarrow_\mathsf{R} \{0,1\}^\ell$, return $\mathsf{pk} := (\mathcal{PG}, [\mathbf{a}]_1, [\mathbf{k}^\top \mathbf{a}]_1)$ and $\mathsf{sk} := [\mathbf{k}]_T$.

$\underline{\mathsf{PKE.Enc}(\mathsf{pk}, [m]_T \in \mathbb{G}_T)}$:
$r \leftarrow_\mathsf{R} \mathbb{Z}_p$, return $([\mathbf{a}r]_1, [\mathbf{k}^\top \mathbf{a}r]_T + [m]_T)$

$\underline{\mathsf{PKE.Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})}$:
Recover $\mathbf{k} \in \{0,1\}^\ell$ from $\mathsf{sk} := [\mathbf{k}] \in \mathbb{G}_T^\ell$.
Parse $\mathsf{ct} := ([\mathbf{c}_0]_1, [c_1]_T)$, return $[c_1]_T - e([\mathbf{k}^\top \mathbf{c}_0]_1, [1]_2)$.

$\underline{\widetilde{\mathsf{Enc}}(\mathsf{sk}, \mathsf{pk}, [m]_T \in \mathbb{G}_T)}$:
$\mathbf{u} \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$, return $([\mathbf{u}]_1, [\mathbf{k}^\top \mathbf{u}]_T + [m]_T)$

Fig. 6. KDM-secure public-key encryption from [**?**].

$\underline{\mathsf{SampParams}(\mathsf{pk})}$:
For all $i \in [\lambda]$, $b \in \{0,1\}$, $\mathbf{W}_{i,b} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{2\times\ell}$. $\mathbf{b} \leftarrow_{\mathsf{R}} \mathsf{DDH}$. Return $\mathsf{params} := \big(\mathbf{b}, \{\mathbf{W}_{i,b}\}_{i\in[\lambda],b\in\{0,1\}}\big)$.

$\underline{\mathsf{Expand}_{\mathsf{pk}}(\mathsf{params}, \mathsf{pk})}$:
Parse $\mathsf{pk} := (\mathcal{PG}, [\mathbf{a}]_1, [\mathbf{k}^\top \mathbf{a}]_1)$.
Return $\mathsf{pk}' := \big([\mathbf{b}]_2, \{[\mathbf{W}_{i,b}\mathbf{a}]_1, [\mathbf{W}_{i,b}^\top \mathbf{b}]_2\}_{i\in[\lambda],b\in\{0,1\}}\big)$

$\underline{\mathsf{Expand}_{\mathsf{ct}}(\mathsf{pk}, \mathsf{params}, \mathsf{ct}_0, \mathsf{id} \in \{0,1\}^\lambda)}$:
Parse $\mathsf{ct}_0 := ([\mathbf{c}]_1, [c']_T)$, return $\mathsf{ct}_1 := \sum_{i\in[\lambda]} [\mathbf{W}_{i,\mathsf{id}_i}\mathbf{c}]_1$.

Fig. 7. KDM-secure modular IBE, for the identity space $\{0,1\}^\lambda$. We denote by id_i the i'th bit of $\mathsf{id} \in \{0,1\}^\ell$. It builds upon the PKE from Figure 6.

We now proceed to prove the required properties from our concrete instantiation of the modular framework presented in the previous section.

Property 1 (semi-functional encryption). The difference between normal and semi-functional ciphertexts is that the vector $[\mathbf{a}r]_1$, with $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$ that is part of each challenge ciphertext is switched to a uniformly random vector over $\mathbb{G}_1^\ell$, using the (ℓ, Q)-fold DDH assumption, where Q denotes the number of encryption queries. By Lemma 1, this assumption is implied by the DDH assumption. Upon receiving a (ℓ, Q)-DDH challenge $([\mathbf{a}]_1, \{[\mathbf{z}_i]_1\}_{i\in[Q]})$, where either $[\mathbf{z}_i]_1 = [\mathbf{a}r_i]_1$ for $r_i \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, or $[\mathbf{z}_i]_1 \leftarrow_{\mathsf{R}} \mathbb{G}_1^\ell$, the reduction samples $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$, and returns $\mathsf{pk} := ([\mathbf{a}]_1, [\mathbf{k}^\top \mathbf{a}]_T)$ and $\mathsf{sk} := [\mathbf{k}]_T$ to $\mathcal{A}$. On the i'th query $\mathsf{O}_{\mathsf{Enc}}([m]_T \in \mathbb{G}_T)$, the reduction answers with $([\mathbf{z}_i]_1, [\mathbf{k}^\top \mathbf{z}_i]_T + [m]_T)$, for $i \in [Q]$.

Property 2, semi-functional keys. The proof goes through a sequence of hybrid games, defined in Figure 9. Let $\mathcal{A}$ be a PPT adversary. For each game G, we denote by $\mathsf{Adv}_{\mathcal{A}}(\mathsf{G})$ the advantage of $\mathcal{A}$ if game G. We start with game G_0, which is the security game defined in Property 2.

Game G_1: we change the vector $[\mathbf{u}]_1 \leftarrow_{\mathsf{R}} \mathbb{G}_1^\ell$ used in each challenge ciphertext to $[\mathbf{a}_0 r]$, for $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, and $\mathbf{a}_0 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$, independent of $\mathbf{a}$ used in the public key, using the (ℓ, Q)-fold DDH assumption in $\mathbb{G}_1$, where Q denotes the number of queries to $\mathsf{O}_{\mathsf{Enc}}$. By Lemma 1, this is implied by the DDH assumption. We build a PPT adversary $\mathcal{B}_0$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_0) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_1)| \leq \mathsf{Adv}^{\ell,Q\text{-}\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}_0}(\lambda).$$

Upon receiving a (ℓ, Q)-DDH challenge $([\mathbf{a}_0]_1, \{[\mathbf{z}_i]_1\}_{i\in[Q]}$, $\mathcal{B}_0$ samples $b \leftarrow_{\mathsf{R}} \{0,1\}$, $\mathbf{a} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$, $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$, and for all $i \in [\lambda]$, $b \in \{0,1\}$: $\mathbf{W}_{i,b} \leftarrow_{\mathsf{R}}$

$\underline{\mathsf{Setup}(1^\lambda)}$:
$\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2, e) \leftarrow \mathsf{GGen}(1^\lambda)$, $\ell := 4\lceil \log_2(p) \rceil$, $\mathbf{a} \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$, $\mathbf{b} \leftarrow_\mathsf{R} \mathsf{DDH}$, $\mathbf{k} \leftarrow_\mathsf{R} \{0,1\}^\ell$. For all $i \in [\lambda]$, $b \in \{0,1\}$, $\mathbf{W}_{i,b} \leftarrow_\mathsf{R} \mathbb{Z}_p^{2\times\ell}$. Return $\mathsf{mpk} := (\mathcal{PG}, [\mathbf{a}]_1, [\mathbf{b}]_2, \{[\mathbf{W}_{i,b}\mathbf{a}]_1, [\mathbf{W}_{i,b}^\top \mathbf{b}]_2\}_{i\in[\lambda], b\in\{0,1\}}, [\mathbf{k}^\top \mathbf{a}]_T)$ and $\mathsf{msk} := [\mathbf{k}]_T$

$\underline{\mathsf{Enc}(\mathsf{mpk}, \mathsf{id} \in \{0,1\}^\lambda, [m]_T \in \mathbb{G}_T)}$:
$r \leftarrow_\mathsf{R} \mathbb{Z}_p$, return: $\mathsf{ct} := ([\mathbf{a}r]_1, [\sum_{i\in[\lambda]} \mathbf{W}_{i,\mathsf{id}_i}\mathbf{a}r]_1, [\mathbf{k}^\top \mathbf{a}r]_T + [m]_T)$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id} \in \{0,1\}^\lambda)}$:
Recover $\mathbf{k}$ from $[\mathbf{k}]_T$, $s \leftarrow_\mathsf{R} \mathbb{Z}_p$, return $\mathsf{sk}_\mathsf{id} := ([\mathbf{b}s]_2, [\mathbf{k} + \sum_{i\in[\lambda]} \mathbf{W}_{i,\mathsf{id}_i}^\top \mathbf{b}s]_2)$.

$\underline{\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}_\mathsf{id})}$:
Parse $\mathsf{ct} := ([\mathbf{c}]_1, [\mathbf{c}']_1, [c'']_T) \in \mathbb{G}_1^\ell \times \mathbb{G}_1^2 \times \mathbb{G}_T$ and $\mathsf{sk}_\mathsf{id} := ([\mathbf{d}]_2, [\mathbf{d}']_2) \in \mathbb{G}_2^2 \times \mathbb{G}_2^\ell$.
Return $[c'']_T - e([\mathbf{c}]_1^\top, [\mathbf{d}']_2) + e([\mathbf{c}']_1^\top, [\mathbf{d}]_2)$.

Fig. 8. Concrete description of our KDM-secure IBE.

$\mathbb{Z}_p^{2\times\ell}$, thanks to which it can compute mpk and simulate $\mathsf{O}_{\mathsf{KeyGen}}$ to $\mathcal{A}$ as described in Figure 9. On the i'th query of $\mathcal{A}$ to $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, \mathbf{w}, [m]_T)$, $\mathcal{B}_0$ returns $([\mathbf{z}_i]_1, [\mathbf{W}_\mathsf{id}\mathbf{z}_i]_1, [\mathbf{k}^\top \mathbf{z}_i + \mathbf{k}^\top \mathbf{w} + m]_T)$, where $\mathbf{W}_\mathsf{id} := \sum_{i\in[\lambda]} \mathbf{W}_{i,\mathsf{id}_i}$.

Game G_2: we change the vector $[\mathbf{d}]_2$ in each user secret key from $[\mathbf{b}s]_2$ for $s \leftarrow_\mathsf{R} \mathbb{Z}_p$ to uniformly random over $\mathbb{G}_2^2$, using the DDH assumption in $\mathbb{G}_2$. We build a PPT adversary $\mathcal{B}_1$ such that:

$$|\mathsf{Adv}_\mathcal{A}(\mathsf{G}_1) - \mathsf{Adv}_\mathcal{A}(\mathsf{G}_2)| \leq \mathsf{Adv}^{1,Q_\mathsf{sk}\text{-}\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}_1}(\lambda),$$

where Q_sk denotes the number of queries to $\mathsf{O}_{\mathsf{KeyGen}}$.

Upon receiving a $1, Q_\mathsf{sk}$-fold DDH challenge $([\mathbf{b}]_2, \{[\mathbf{z}_i]_2\}_{i\in[Q_\mathsf{sk}]})$, $\mathcal{B}_1$ samples $b \leftarrow_\mathsf{R} \{0,1\}$, $\mathbf{a}, \mathbf{a}_0 \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$, $\mathbf{k} \leftarrow_\mathsf{R} \{0,1\}^\ell$, and for all $i \in [\lambda]$, $b \in \{0,1\}$: $\mathbf{W}_{i,b} \leftarrow_\mathsf{R} \mathbb{Z}_p^{2\times\ell}$, thanks to which it can compute mpk and simulate $\mathsf{O}_{\mathsf{Enc}}$ to $\mathcal{A}$ as described in Figure 9. On the i'th query of $\mathcal{A}$ to $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$, $\mathcal{B}_0$ returns $([\mathbf{z}_i]_2, [\mathbf{k}_b + \mathbf{W}_\mathsf{id}\mathbf{z}_i]_2)$, where $\mathbf{W}_\mathsf{id} := \sum_{i\in[\lambda]} \mathbf{W}_{i,\mathsf{id}_i}$, $\mathbf{k}_0 := \mathbf{k}$ and $\mathbf{k}_1 := \frac{\mathbf{k}^\top \mathbf{a}}{\|\mathbf{a}\|_2^2}$.

Game G_3: we change the way $\mathbf{W}_\mathsf{id}$ is computed, as described in Figure 9. In Lemma 3, we show that there exists a PPT adversary $\mathcal{B}_2$ such that:

$$|\mathsf{Adv}_\mathcal{A}(\mathsf{G}_2) - \mathsf{Adv}_\mathcal{A}(\mathsf{G}_3)| \leq 3\lambda \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2,\mathcal{B}_2}(\lambda) + \frac{2\lambda Q_\mathsf{sk}}{p},$$

where Q_sk denotes the number of queries to $\mathsf{O}_{\mathsf{KeyGen}}$.

Game G_4: we change the distribution of the user secret keys as described in Figure 9.

First, we use the fact that the following distributions are statistically $1/p$-close:

$$\mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2 \text{ and } \gamma \cdot \mathbf{d}, \text{ with } \gamma \leftarrow_{\mathsf{R}} \mathbb{Z}_p, \mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2.$$

Thus, we can write the output of $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ as

$$([\gamma \cdot \mathbf{d}]_2, [\mathbf{k}_b + \sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}^\top (\gamma \cdot \mathbf{d}) + \mathbf{A}^\perp \gamma \cdot \mathsf{RF}(\mathsf{id}) \cdot (\mathbf{b}^\perp)^\top \mathbf{d}]_2),$$

with fresh $\mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$ and $\gamma \leftarrow_{\mathsf{R}} \mathbb{Z}_p$. Using the DDH assumption in $\mathbb{G}_2$, for any identity id queried to $\mathsf{O}_{\mathsf{KeyGen}}$ (and therefore, not queried to $\mathsf{O}_{\mathsf{Enc}}$), we can switch $([\gamma]_2, [\mathsf{RF}(\mathsf{id})]_2, [\gamma \cdot \mathsf{RF}(\mathsf{id})]_2)$ to $([\gamma]_2, [\mathsf{RF}(\mathsf{id})]_2, [\mathbf{t}]_2)$, where $\gamma \leftarrow_{\mathsf{R}} \mathbb{Z}_p$ and $\mathbf{t} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell-1}$. Note that we make crucial use of the fact the value $\mathsf{RF}(\mathsf{id})$ for an identity id queried to $\mathsf{O}_{\mathsf{KeyGen}}$ only appears in the output of $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$, since this identity must not be queried to $\mathsf{O}_{\mathsf{Enc}}$ by $\mathcal{A}$. This means the output of $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ becomes:

$$([\gamma \cdot \mathbf{d}]_2, [\mathbf{k}_b + \sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}^\top (\gamma \cdot \mathbf{d}) + \mathbf{A}^\perp \mathbf{t} \cdot (\mathbf{b}^\perp)^\top \mathbf{d}]_2),$$

where $\gamma \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, $\mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$ and $\mathbf{t} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell-1}$ are sampled freshly upon generation of each user secret key.

Finally, we switch back $\gamma \cdot \mathbf{d}$ to $\mathbf{d}$, for $\mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$, $\gamma \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, which are $1/p$ statistically close, such that $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ becomes:

$$([\mathbf{d}]_2, [\mathbf{k}_b + \sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}^\top \mathbf{d} + \mathbf{A}^\perp \mathbf{t} \cdot (\mathbf{b}^\perp)^\top \mathbf{d}]_2),$$

which exactly as in game G_4. We have successfully transitioned from game G_3 to G_4; overall we have a PPT adversary $\mathcal{B}_4$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_3) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_4)| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2, \mathcal{B}_4}(\lambda) + \frac{2Q_{\mathsf{sk}}}{p},$$

where Q_{sk} denotes the number of queries to $\mathsf{O}_{\mathsf{KeyGen}}$.

Now, we show that:

$$\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_4) \leq \frac{Q_{\mathsf{sk}}}{p}.$$

This is due to the fact that in game G_4, the semi-functional component of msk is statistically hidden in the generated user secret keys.

Indeed, $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ outputs $([\mathbf{d}]_2, [\mathbf{k}_b + \sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}^\top \mathbf{d} + \mathbf{A}^\perp \mathbf{t} \cdot (\mathbf{b}^\perp)^\top \mathbf{d}]_2)$, where $\mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$, and $\mathbf{t} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell-1}$ are sampled freshly for each generated user secret key. Using the basis $(\mathbf{a}|\mathbf{A}^\perp)$ of $\mathbb{Z}_p^\ell$, we can write $\mathbf{k} := \mathbf{a} \cdot \mathsf{msk}_{\mathsf{N}} + \mathbf{A}^\perp \cdot \mathsf{msk}_{\mathsf{SF}}$, where $\mathsf{msk}_{\mathsf{N}} \in \mathbb{Z}_p$ and $\mathsf{msk}_{\mathsf{SF}} \in \mathbb{Z}_p^{\ell-1}$ denotes the normal and semi-functional components of $\mathbf{k}$, respectively. The component $\mathsf{msk}_{\mathsf{SF}}$ is completely hidden by the random vector $\mathbf{t} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell-1}$. Namely, conditioned on the fact that $\mathbf{d}^\top \mathbf{b}^\perp \neq 0$,

which holds with probability $1/p$ over the choice of $\mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$, the output of $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ is identically distributed to:

$$([\mathbf{d}]_2, [\mathbf{a} \cdot \mathsf{msk}_{\mathsf{N}} + \sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}^{\top} \mathbf{d} + \mathbf{A}^{\perp} \mathbf{t} \cdot (\mathbf{b}^{\perp})^{\top} \mathbf{d}]_2),$$

where $\mathsf{msk}_{\mathsf{N}} := \frac{\mathbf{k}^{\top}\mathbf{a}}{\|\mathbf{a}\|_2^2}$. At this point, the output is independent of the random bit $b \leftarrow_{\mathsf{R}} \{0,1\}$ picked by the experiment. □

Game G_0, $\boxed{\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3, \mathsf{G}_4}$:

$b \leftarrow_{\mathsf{R}} \{0,1\}$, $\mathcal{PG} \leftarrow \mathsf{GGen}(1^{\lambda})$, $\ell := 4\lceil \log_2(p) \rceil$, $\mathbf{a} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell}$, $\boxed{\mathbf{a}_0 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell}}$, $\mathbf{A}^{\perp} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell \times (\ell-1)}$ s.t. $\mathbf{a}^{\top}\mathbf{A}^{\perp} = \mathbf{0}$, $\mathbf{b} \leftarrow_{\mathsf{R}} \mathsf{DDH}$, $\mathbf{b}^{\perp} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$ s.t. $\mathbf{b}^{\top}\mathbf{b}^{\perp} = 0$, $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^{\ell}$. For all $i \in [\lambda]$, $b \in \{0,1\}$, $\mathbf{W}_{i,b} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{2 \times \ell}$.
$\mathsf{mpk} := (\mathcal{PG}, [\mathbf{a}]_1, \{[\mathbf{W}_{i,b}\mathbf{a}]_1, [\mathbf{W}_{i,b}^{\top}\mathbf{b}]_2\}_{i \in [\lambda], b \in \{0,1\}}, [\mathbf{k}^{\top}\mathbf{a}]_T)$
$b' \leftarrow_{\mathsf{R}} \mathcal{A}^{\mathsf{O}_{\mathsf{Enc}}(\cdot,\cdot), \mathsf{O}_{\mathsf{KeyGen}}{}^{(b)}(\cdot)}(\mathsf{mpk})$
Return 1 if $b' = b$ and identities queried to $\mathsf{O}_{\mathsf{Enc}}$ are distinct from identities queried to $\mathsf{O}_{\mathsf{KeyGen}}$.
Return 0 otherwise.

$\underline{\mathsf{O}_{\mathsf{Enc}}(\mathsf{id} \in \{0,1\}^{\lambda}, \mathbf{w} \in \mathbb{Z}_p^{\ell}, [m]_T \in \mathbb{G}_T)}$: $\quad \mathsf{G}_0, \boxed{\mathsf{G}_1, \mathsf{G}_2, \boxed{\mathsf{G}_3}, \mathsf{G}_4}$

$\mathbf{u} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell}$, $[\mathbf{c}]_1 := [\mathbf{u}]_1$, $\boxed{r \leftarrow_{\mathsf{R}} \mathbb{Z}_p, [\mathbf{c}]_1 := [\mathbf{a}_0 r]_1}$
$\mathbf{W}_{\mathsf{id}} := \sum_{i \in [\lambda]} \mathbf{W}_{i,\mathsf{id}_i} + \boxed{\mathbf{b}^{\perp}(\mathbf{A}^{\perp}\mathsf{RF}(\mathsf{id}))^{\top}}$
$\mathsf{ct} := ([\mathbf{c}]_1, [\mathbf{W}_{\mathsf{id}}\mathbf{c}]_1, [\mathbf{k}^{\top}\mathbf{c}]_T + [\mathbf{k}^{\top}\mathbf{w} + m]_T)$

$\underline{\mathsf{O}_{\mathsf{KeyGen}}{}^{(b)}(\mathsf{id} \in \{0,1\}^{\lambda})}$: $\quad \mathsf{G}_0, \mathsf{G}_1, \boxed{\mathsf{G}_2, \boxed{\mathsf{G}_3}, \mathsf{G}_4}$

$s \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, $[\mathbf{d}]_2 := [\mathbf{b}s]_2$, $\boxed{[\mathbf{d}]_2 \leftarrow_{\mathsf{R}} \mathbb{G}_2^2}$, $\mathbf{k}_0 := \mathbf{k}$, $\mathbf{k}_1 := \frac{\mathbf{k}^{\top}\mathbf{a}}{\|\mathbf{a}\|_2^2} \cdot \mathbf{a}$, $\mathbf{t} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell-1}$
$\mathbf{W}_{\mathsf{id}} := \sum_{i \in [\lambda]} \mathbf{W}_{i,\mathsf{id}_i} + \boxed{\mathbf{b}^{\perp}(\mathbf{A}^{\perp}\mathsf{RF}(\mathsf{id}))^{\top}} + \mathbf{b}^{\perp}(\mathbf{A}^{\perp}\mathbf{t})^{\top}$
Return $\mathsf{sk}_{\mathsf{id}} := ([\mathbf{d}]_2, [\mathbf{k}_b + \mathbf{W}_{\mathsf{id}}^{\top}\mathbf{d}]_2)$.

Fig. 9. Games for the proof of Property 2. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame. Here, $\mathsf{RF} : \{0,1\}^{\lambda} \to \mathbb{Z}_p^{\ell-1}$ denotes a random function that is computed on the fly.

Lemma 3 (From game G_2 to game G_3). *There exists a PPT adversary $\mathcal{B}_2$ such that:*

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_2) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_3)| \leq 3\lambda \cdot \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{DDH}}(\lambda) + \frac{2\lambda Q_{\mathsf{sk}}}{p},$$

where Q_{sk} denotes the number of queries to $\mathsf{O}_{\mathsf{KeyGen}}$.

Proof. The proof goes over a series of hybrid games defined in Figure 10. We progressively increase the entropy in the matrices $\mathbf{W}_{\mathsf{id}}$, originally set as $\mathbf{W}_{\mathsf{id}} := \sum_{j\in[\lambda]} \mathbf{W}_{j,\mathsf{id}_j}$ in game G_2, up to $\mathbf{W}_{\mathsf{id}} := (\sum_{j\in[\lambda]} \mathbf{W}_{j,\mathsf{id}_j}) + (\mathbf{A}^\perp \mathsf{RF}(\mathsf{id}))^\top$ in game G_3, where RF is a random function, computed on the fly by the experiment. Namely, in game $\mathsf{G}_{2.i}$, we have $\mathbf{W}_{\mathsf{id}} := (\sum_{j\in[\lambda]} \mathbf{W}_{j,\mathsf{id}_j}) + (\mathbf{A}^\perp \mathsf{RF}_i(\mathsf{id}))^\top$, where RF_i is a random function that only depends on the first i'th bits on its input. It is clear that $\mathsf{G}_{2.\lambda}$ is the same as G_3. We prove that G_2 is statistically close to $\mathsf{G}_{2.0}$ (note that RF_0 is a constant function, that ignores its input), and we show that for all $i \in [\lambda]$, G_{i-1} is computationally indistinguishable from G_i, in a way that is reminiscent to the security proof from [**?**]. One difference here is that the vector $\mathbf{k}$ is not uniformly random over $\mathbb{Z}_p$, which adds technical difficulties.

Game $\mathsf{G}_{2.0}$. This game is as G_1, except the matrix $\mathbf{W}_{\mathsf{id}}$ is switched from $\mathbf{W}_{\mathsf{id}} := \sum_{j\in[\lambda]} \mathbf{W}_{j,\mathsf{id}_j}$ to $\mathbf{W}_{\mathsf{id}} := \sum_{j\in[\lambda]} \mathbf{W}_{j,\mathsf{id}_j} + \mathbf{b}^\perp(\mathbf{A}^\perp \mathsf{RF}_0(\mathsf{id}))^\top$, where $\mathsf{RF}_0(\mathsf{id})$ is a random vector in $\mathbb{Z}_p^{\ell-1}$, independent of id (the extra term is highlighted in gray to better see the difference between G_2 and $\mathsf{G}_{2.0}$). This does change the distribution of the game, since $(\mathbf{W}_{1,0}, \mathbf{W}_{1,1})$ is identically distributed to $(\mathbf{W}_{1,0} + \mathbf{b}^\perp(\mathbf{A}^\perp \mathsf{RF}_0(\mathsf{id}))^\top, \mathbf{W}_{1,1} + \mathbf{b}^\perp(\mathbf{A}^\perp \mathsf{RF}_0(\mathsf{id}))^\top)$. Note that these extra terms don't appear in the public key, since $\mathbf{a}^\top \mathbf{A}^\perp = \mathbf{0}$ and $\mathbf{b}^\top \mathbf{b}^\perp = 0$. Thus, we have:

$$\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_1) = \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_{2.0}).$$

Games $\mathsf{G}_{2.i-1.1}$, for all $i \in [\lambda+1]$. This game is as $\mathsf{G}_{2.i-1}$, except the vector $[\mathbf{c}]_1$ output $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, \mathbf{w}, [m]_T)$ is switched from $[\mathbf{a}_0 r]_1$ to $[\mathbf{a}_{\mathsf{id}_i} r]_1$, with $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, where id_i denotes the i'th bit of id, and $\mathbf{a}_0, \mathbf{a}_1 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$ are two independent random vectors. We use the DDH assumption in $\mathbb{G}_1$, to first switch $[\mathbf{a}_0 r]_1$ to uniformly random over $\mathbb{G}_1^2$ when necessary, that is, when $\mathsf{id}_i = 1$; then we use the DDH assumption again to switch the uniformly random vector to $[\mathbf{a}_1 r]_1$ with $r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$. Overall we have a PPT adversary $\mathcal{B}_i$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_{2.i-1}) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_{2.i-1.1})| \leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}_i}(\lambda).$$

Games $\mathsf{G}_{2.i-1.2}$, for all $i \in [\lambda+1]$. See the description in Figure 10.

As in the security proof of the CCA-secure pke from [**?**], we use a basis $(\mathbf{A}_0^\perp | \mathbf{A}_1^\perp) \in \mathbb{Z}_p^{\ell-1}$ of $\mathbf{A}^\perp$ where $\mathbf{a}_0^\top \mathbf{A}_0^\perp = \mathbf{a}_1^\top \mathbf{A}_1^\perp = \mathbf{0}$, where both $\mathbf{a}_0$ and $\mathbf{a}_1$ are uniformly random vectors from $\mathbb{Z}_p^\ell$, sampled independently.

Namely, we sample $\mathbf{A}_0^\perp \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell \times \ell/2}$ and $\mathbf{A}_1^\perp \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell \times (\ell/2-1)}$ such that $(\mathbf{A}_0^\perp | \mathbf{A}_1^\perp) \in \mathbb{Z}_p^{\ell-1}$ is full rank, and $\mathbf{a}^\top \mathbf{A}_0^\perp = \mathbf{a}_0^\top \mathbf{A}_0^\perp = \mathbf{a}^\top \mathbf{A}_1^\perp = \mathbf{a}_1^\top \mathbf{A}_1^\perp = \mathbf{0}$.

Using this basis, we can decompose $\mathbf{A}^\perp \mathsf{RF}_{i-1}(\mathsf{id}) := \mathbf{A}_0^\perp \mathsf{RF}_{i-1}^{(0)}(\mathsf{id}) + \mathbf{A}_1^\perp \mathsf{RF}_{i-1}^{(1)}(\mathsf{id})$, where $\mathsf{RF}_{i-1}^{(0)} : \{0,1\}^\lambda \to \mathbb{Z}_p^{\ell/2}$ and $\mathsf{RF}_{i-1}^{(1)} : \{0,1\}^\lambda \to \mathbb{Z}_p^{\ell/2-1}$ are independent random functions that only read the first $i-1$'th bits of their inputs.

We define

$$\mathsf{RF}_i^{(0)}(\mathsf{id}) := \begin{cases} \mathsf{RF}_{i-1}^{(0)}(\mathsf{id}) + \widetilde{\mathsf{RF}}_{i-1}^{(0)}(\mathsf{id}) & \text{if } \mathsf{id}_i = 0 \\ \mathsf{RF}_{i-1}^{(0)}(\mathsf{id}) & \text{if } \mathsf{id}_i = 1 \end{cases},$$

and

$$\mathsf{RF}_i^{(1)}(\mathsf{id}) := \begin{cases} \mathsf{RF}_{i-1}^{(1)}(\mathsf{id}) & \text{if } \mathsf{id}_i = 0 \\ \mathsf{RF}_{i-1}^{(1)}(\mathsf{id}) + \widetilde{\mathsf{RF}}_{i-1}^{(1)}(\mathsf{id}) & \text{if } \mathsf{id}_i = 1 \end{cases},$$

where $\widetilde{\mathsf{RF}}_{i-1}^{(0)} : \{0,1\}^\lambda \to \mathbb{Z}_p^{\ell/2}$ and $\widetilde{\mathsf{RF}}_{i-1}^{(1)} : \{0,1\}^\lambda \to \mathbb{Z}_p^{\ell/2-1}$ are random functions that only read the first $i-1$'th bits of their inputs, that are independent of $\mathsf{RF}_{i-1}^{(0)}$ and $\mathsf{RF}_{i-1}^{(1)}$. Note that the random functions $\mathsf{RF}_i^{(0)}$ and $\mathsf{RF}_i^{(1)}$ now depend on the first i'th bits of their inputs: we added a dependency on the i'th bit. Thus, writing $\mathbf{A}^\perp \mathsf{RF}_i(\mathsf{id}) := \mathbf{A}_0^\perp \mathsf{RF}_i^{(0)}(\mathsf{id}) + \mathbf{A}_1^\perp \mathsf{RF}_i^{(1)}(\mathsf{id})$, we have $\mathbf{A}^\perp \mathsf{RF}_i(\mathsf{id}) = \mathbf{A}^\perp \mathsf{RF}_{i-1}(\mathsf{id}) + \boxed{\mathbf{A}_{\mathsf{id}_i}^\perp \widetilde{\mathsf{RF}}_{i-1}^{(\mathsf{id}_i)}(\mathsf{id})}$. The game $\mathsf{G}_{2.i-1.2}$ is the same as $\mathsf{G}_{2.i-1.1}$, except the latter uses $\mathbf{W}_{\mathsf{id}} := (\sum_{j\in[\lambda]} \mathbf{W}_{j,\mathsf{id}_j}) + \mathbf{b}^\perp (\mathbf{A}^\perp \mathsf{RF}_{i-1}(\mathsf{id}))^\top$, and the former uses $\mathbf{W}_{\mathsf{id}} + \boxed{\mathbf{A}_{\mathsf{id}_i}^\perp \widetilde{\mathsf{RF}}_{i-1}^{(\mathsf{id}_i)}(\mathsf{id})}$.

Note that this change doesn't appear in the challenge ciphertexts, since $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, \mathbf{w}, [m]_T)$ outputs:

$$\begin{aligned} \mathsf{ct} :&= ([\mathbf{a}_{\mathsf{id}_i} r]_1, [(\mathbf{W}_{\mathsf{id}} + \mathbf{b}^\perp (\mathbf{A}_{\mathsf{id}_i}^\perp \widetilde{\mathsf{RF}}_{i-1}^{(\mathsf{id}_i)}(\mathsf{id}))^\top \mathbf{a}_{\mathsf{id}_i} r]_1, [\mathbf{k}^\top \mathbf{a}_{\mathsf{id}_i} r + \mathbf{k}^\top \mathbf{w} + m]_T) \\ &= ([\mathbf{a}_{\mathsf{id}_i} r]_1, [(\mathbf{W}_{\mathsf{id}} \mathbf{a}_{\mathsf{id}_i} r]_1, [\mathbf{k}^\top \mathbf{a}_{\mathsf{id}_i} r + \mathbf{k}^\top \mathbf{w} + m]_T), \end{aligned}$$

since $\mathbf{a}_0^\top \mathbf{A}_0^\perp = \mathbf{a}_1^\top \mathbf{A}_1^\perp = \mathbf{0}$. Thus, the output of the oracle $\mathsf{O}_{\mathsf{Enc}}$ is identically distributed in $\mathsf{G}_{2.i-1.1}$ and $\mathsf{G}_{2.i-1.2}$. We now turn our attention to the output of $\mathsf{O}_{\mathsf{KeyGen}}$.

First, we use the fact that the following are identically distributed:

$$\mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2 \text{ and } \widehat{\mathsf{RF}}_{i-1}(\mathsf{id}) \cdot \mathbf{d}, \text{ with } \mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2,$$

where $\widehat{\mathsf{RF}}_{i-1} : \{0,1\}^\lambda \to \mathbb{Z}_p$ is a random function that only reads the first $i-1$'th bits of its input. That is, $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ uses a random vector $[\widehat{\mathsf{RF}}_{i-1}(\mathsf{id}) \cdot \mathbf{d}]_2$ instead of $[\mathbf{d}]_2 \leftarrow_{\mathsf{R}} \mathbb{G}_2^2$.

Then, we use the fact that following distributions are within statistical distance $1/p$:

$$(\mathbf{W}_{i,0}, \mathbf{W}_{i,1}) \text{ and } (\mathbf{W}_{i,0} + \mathbf{b}^\perp (\mathbf{A}_0^\perp \mathbf{u}_0)^\top, \mathbf{W}_{i,1} + \mathbf{b}^\perp (\mathbf{A}_1^\perp \mathbf{u}_1)^\top),$$

where $\mathbf{W}_{i,0}, \mathbf{W}_{i,1} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{2\times\ell}$, $\mathbf{u}_0 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell/2}$, $\mathbf{u}_1 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell/2-1}$.

Thus, we can re-write the output of $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ as:

$$([\mathbf{d} \cdot \widehat{\mathsf{RF}}_{i-1}(\mathsf{id})]_2, [\mathbf{k}_b + \mathbf{W}_{\mathsf{id}}^\top \widehat{\mathsf{RF}}_{i-1}(\mathsf{id}) \cdot \mathbf{d} + \mathbf{A}_{\mathsf{id}_i}^\perp \mathbf{u}_{\mathsf{id}_i} \cdot \widehat{\mathsf{RF}}_{i-1}(\mathsf{id}) (\mathbf{b}^\perp)^\top \mathbf{d}]_2).$$

Note that the vectors $\mathbf{u}_0$ and $\mathbf{u}_1$ do not appear in the public key or the challenge ciphertexts, since $\mathbf{a}_0^\top \mathbf{A}_0^\perp = \mathbf{a}_1^\top \mathbf{A}_1^\perp = \mathbf{0}$.

At this point, we use the DDH assumption in $\mathbb{G}_2$ to switch

$$([\widehat{\mathsf{RF}}_{i-1}(\mathsf{id})]_2, [\mathbf{u}_{\mathsf{id}_i} \cdot \widehat{\mathsf{RF}}_{i-1}(\mathsf{id})]_2)$$

to

$$([\widehat{\mathsf{RF}}_{i-1}(\mathsf{id})]_2, [\widetilde{\mathsf{RF}}_{i-1}^{(\mathsf{id}_i)}(\mathsf{id})]_2).$$

The output of $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ becomes:

$$([\mathbf{d} \cdot \widehat{\mathsf{RF}}_{i-1}(\mathsf{id})]_2, [\mathbf{k}_b + \mathbf{W}_{\mathsf{id}}^\top \widehat{\mathsf{RF}}_{i-1}(\mathsf{id}) \cdot \mathbf{d} + \mathbf{A}_{\mathsf{id}_i}^\perp \widetilde{\mathsf{RF}}_{i-1}^{(\mathsf{id}_i)}(\mathsf{id})(\mathbf{b}^\perp)^\top \mathbf{d}]_2).$$

Finally, we reverse the statistical change from $[\widehat{\mathsf{RF}}_{i-1}(\mathsf{id}) \cdot \mathbf{d}]_2$ to $[\mathbf{d}]_2$ in each user secret key, so that the output of $\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id})$ becomes:

$$([\mathbf{d}]_2, [\mathbf{k}_b + (\sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j})\mathbf{d} + (\mathbf{A}^\perp \mathsf{RF}_{i-1}(\mathsf{id}) + \mathbf{A}_{\mathsf{id}_i}^\perp \widetilde{\mathsf{RF}}_{i-1}^{(\mathsf{id}_i)}(\mathsf{id}))(\mathbf{b}^\perp)^\top \mathbf{d}]_2) =$$
$$([\mathbf{d}]_2, [\mathbf{k}_b + (\sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j})\mathbf{d} + (\mathbf{A}^\perp \mathsf{RF}_i(\mathsf{id})(\mathbf{b}^\perp)^\top \mathbf{d}]_2),$$

exactly as in game $\mathsf{G}_{2.i-1.2}$. Putting everything together, we obtain a PPT adversary $\mathcal{B}_i'$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_{2.i-1.1}) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_{2.i-1.2})| \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathbb{G}_2, \mathcal{B}_i'}(\lambda) + \frac{2Q_{\mathsf{sk}}}{p},$$

where Q_{sk} denotes the number of queries to $\mathsf{O}_{\mathsf{KeyGen}}$.

Summing up for all $i \in [\lambda]$, we obtain a PPT adversary $\mathcal{B}_2$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_2) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_3)| \leq 3\lambda \cdot \mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{B}_2}(\lambda) + \frac{2\lambda Q_{\mathsf{sk}}}{p}.$$

□

Property 3 (KDM security). First, as in the security proof of [**?**], we use the fact that the output of $\widetilde{\mathsf{Enc}}(\mathsf{pk}, \mathsf{sk}, [\mathbf{k}^\top \mathbf{w}]_T + [m]_T)$, which is of the form $([\mathbf{u}]_1, [\mathbf{k}^\top(\mathbf{u} + \mathbf{w})]_T + [m]_T$ with $[\mathbf{u}]_1 \leftarrow_{\mathsf{R}} \mathbb{G}_1^\ell$, is identically distributed to $([\mathbf{u} - \mathbf{w}]_1, [\mathbf{k}^\top \mathbf{u}]_T + [m]_T)$. That is, we can remove the dependence of the message on the key $\mathbf{k}$ via a statistical argument. At this point, the proof in [**?**] relies on the DDH assumption on $[\mathbf{a}]_1$. Namely, the ciphertexts are switched back to normal (as opposed to semi-functional), then a hybrid argument goes over each ciphertext one by one, switching it to semi-functional and using a statistical argument (the Left Over Hash lemma to extract the entropy from $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$ and masks the plaintext). However, we cannot use DDH on $[\mathbf{a}]_1$, since the normal component of the master secret key is of the form $\mathsf{msk}_{\mathsf{N}} := \frac{\mathbf{k}^\top \mathbf{a}}{\|\mathbf{a}\|_2^2} \cdot \mathbf{a}$. This value

Games $\mathsf{G}_{2.i-1}$, $\boxed{\mathsf{G}_{2.i-1.1}, \boxed{\mathsf{G}_{2.i-1.2}}}$ for $i \in [\lambda+1]$:

$b \leftarrow_{\mathsf{R}} \{0,1\}$, $\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$, $\ell := 4\lceil \log_2(p) \rceil$, $\mathbf{a} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$, $\mathbf{a}_0 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell$, $\boxed{\mathbf{a}_1 \leftarrow_{\mathsf{R}} \mathbb{Z}_p^\ell}$, $\mathbf{A}^\perp \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell \times (\ell-1)}$ s.t. $\mathbf{a}^\top \mathbf{A}^\perp = \mathbf{0}$, $\mathbf{b} \leftarrow_{\mathsf{R}} \mathsf{DDH}$, $\mathbf{b}^\perp \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$ s.t. $\mathbf{b}^\top \mathbf{b}^\perp = 0$, $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^\ell$.
For all $i \in [\lambda]$, $b \in \{0,1\}$, $\mathbf{W}_{i,b} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{2 \times \ell}$. $\mathsf{mpk} := (\mathcal{PG}, [\mathbf{a}]_1, \{[\mathbf{W}_{i,b}\mathbf{a}]_1, [\mathbf{W}_{i,b}^\top \mathbf{b}]_2\}_{i \in [\lambda], b \in \{0,1\}}, [\mathbf{k}^\top \mathbf{a}]_T)$
$b' \leftarrow_{\mathsf{R}} \mathcal{A}^{\mathsf{O}_{\mathsf{Enc}}(\cdot,\cdot), \mathsf{O}_{\mathsf{KeyGen}}(\cdot)}(\mathsf{mpk})$
Return 1 if $b' = b$ and identities queried to $\mathsf{O}_{\mathsf{Enc}}$ are distinct from identities queried to $\mathsf{O}_{\mathsf{KeyGen}}$.

$\mathsf{O}_{\mathsf{Enc}}(\mathsf{id} \in \{0,1\}^\ell, \mathbf{w} \in \mathbb{Z}_p^\ell, [m]_T \in \mathbb{G}_T)$:
$r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, $[\mathbf{c}]_1 := [\mathbf{a}_0 r]_1$, $\boxed{[\mathbf{c}]_1 := [\mathbf{a}_{\mathsf{id}_i} r]_1}$
$\mathbf{W}_{\mathsf{id}} := (\sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}) + \mathbf{b}^\perp (\mathbf{A}^\perp \mathsf{RF}_{i-1}(\mathsf{id}))^\top$
$\boxed{\mathbf{W}_{\mathsf{id}} := (\sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}) + \mathbf{b}^\perp (\mathbf{A}^\perp \mathsf{RF}_{i}(\mathsf{id}))^\top}$
$\mathsf{ct} := ([\mathbf{c}]_1, [\mathbf{W}_{\mathsf{id}}\mathbf{c}]_1, [\mathbf{k}^\top \mathbf{c} + \mathbf{k}^\top \mathbf{w} + m]_T)$

$\mathsf{O}_{\mathsf{KeyGen}}(\mathsf{id} \in \{0,1\}^\lambda)$:
$\mathbf{k}_0 := \mathbf{k}$, $\mathbf{k}_1 := \frac{\mathbf{k}^\top \mathbf{a}}{\|\mathbf{a}\|_2^2} \cdot \mathbf{a}$
$\mathbf{W}_{\mathsf{id}} := (\sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}) + \mathbf{b}^\perp (\mathbf{A}^\perp \mathsf{RF}_{i-1}(\mathsf{id}))^\top$
$\boxed{\mathbf{W}_{\mathsf{id}} := (\sum_{j \in [\lambda]} \mathbf{W}_{j,\mathsf{id}_j}) + \mathbf{b}^\perp (\mathbf{A}^\perp \mathsf{RF}_{i}(\mathsf{id}))^\top}$
$\mathbf{d} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^2$, return $\mathsf{sk}_{\mathsf{id}} := ([\mathbf{d}]_2, [\mathbf{k}_b + \mathbf{W}_{\mathsf{id}}^\top \mathbf{d}]_2)$.

Fig. 10. Games for the proof of Lemma 3. In each procedure, the components inside a solid (dotted) frame are only present in the games marked by a solid (dotted) frame. Here, for all $i \in [\lambda]$, $\mathsf{RF}_i : \{0,1\}^\lambda \to \mathbb{Z}_p^{\ell-1}$ denotes a random function that only reads the first i'th bits of its input, and that is computed on the fly.

is necessary to generate the user secret keys (see Property 2), and it is not clear how to generate $[\mathsf{msk}_{\mathsf{N}}]_T$ from $[\mathbf{a}]_1$, which prevents to use DDH with respect to $[\mathbf{a}]_1$. Instead, we switch the challenge ciphertexts from $([\mathbf{u}-\mathbf{w}]_1, [\mathbf{k}^\top \mathbf{u}]_T + [m]_T)$ to $([\mathbf{b}s-\mathbf{w}]_1, [\mathbf{k}^\top \mathbf{b}s]_T + [m]_T$, for $s \leftarrow_{\mathsf{R}} \mathbb{Z}_p$, which relies on the DDH assumption with respect to a public vector $[\mathbf{b}]_1 \leftarrow_{\mathsf{R}} \mathbb{G}_1^\ell$ that is independent of $\mathbf{a}$. The rest of the proof is similar to that [**?**]. It is given in Lemma 4.

Lemma 4 (Property 3, KDM security). *The PKE from Figure 6 satisfies Property 3. Namely, for any PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{KDM}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ is a negligible function of λ.*

Proof. The proof goes over a series of hybrid games, where for each game G, we denote by $\mathsf{Adv}_{\mathcal{A}}(\mathsf{G})$ the advantage of PPT adversary $\mathcal{A}$ in game G. We start

with G_0, which is the security game defined in Property 3. In that game, $\mathcal{A}$ receives $\mathsf{pk} := (\mathcal{PG}, [\mathbf{a}]_1, [\mathbf{k}^\top \mathbf{a}]_T)$ and $[\mathsf{msk}_\mathsf{N}]_T$. Recall that $\mathsf{msk} := [\mathbf{k}]_T$, with $\mathbf{k} := \mathsf{msk}_\mathsf{N} + \mathsf{msk}_\mathsf{SF}$, where msk_N and msk_SF are the projections of $\mathbf{k}$ onto $\mathbf{a}$ and $\mathbf{A}^\perp$, respectively; $\mathbf{a} \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$, and $\mathbf{A}^\perp \leftarrow_\mathsf{R} \mathbb{Z}_p^{\ell \times (\ell-1)}$ such that $\mathbf{a}^\top \mathbf{A}^\perp = \mathbf{0}$. For any $\mathbf{w} \in \mathbb{Z}_p^\ell$, $[m]_T \in \mathbb{G}_T$, the oracle $\mathsf{O}_\mathsf{Enc}(\mathbf{w}, [m]_T)$ sets $[m_0]_T := [m]_T$, $[m_1]_T \leftarrow_\mathsf{R} \mathbb{G}_T$, and returns $\widetilde{\mathsf{Enc}}(\mathsf{sk}, \mathsf{pk}, [\mathbf{k}^\top \mathbf{w}]_T + [m_b]_T)$, where $b \leftarrow_\mathsf{R} \{0,1\}$ is chosen by the experiment.

Game G_1. We switch the challenge ciphertexts from $\widetilde{\mathsf{Enc}}(\mathsf{sk}, \mathsf{pk}, [\mathbf{k}^\top \mathbf{w}]_T + [m_b]_T) := ([\mathbf{u}]_1, [\mathbf{k}^\top \mathbf{u}]_T + [\mathbf{k}^\top \mathbf{w} + m_b]_T)$ with $[\mathbf{u}]_1 \leftarrow_\mathsf{R} \mathbb{G}_1^\ell$ in game G_0 to $([\mathbf{u} - \mathbf{w}]_1, [\mathbf{k}^\top \mathbf{u}]_T + [m_b]_T)$ in game G_1. Doing so, we remove the dependence of the encrypted messages on $\mathbf{k}$. We show that the two games are identically distributed, so

$$\mathsf{Adv}_\mathcal{A}(\mathsf{G}_0) = \mathsf{Adv}_\mathcal{A}(\mathsf{G}_1).$$

We use the fact that for any $\mathbf{w} \in \mathbb{Z}_p$, the following distributions are identical:

$$\mathbf{u} \text{ and } \mathbf{u} - \mathbf{w},$$

where $\mathbf{u} \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$. The leftmost distribution corresponds to the game G_0, whereas the rightmost distribution corresponds to the game G_1.

Game G_2. We switch the challenge ciphertexts to $([\mathbf{b}s - \mathbf{w}]_1, [\mathbf{k}^\top \mathbf{b}s]_T + [m_b]_T)$ where $s \leftarrow_\mathsf{R} \mathbb{Z}_p$, and $\mathbf{b} \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$, independent of $\mathbf{a}$ used in the public key and in msk_N. Namely, we build a PPT adversary $\mathcal{B}$ such that:

$$|\mathsf{Adv}_\mathcal{A}(\mathsf{G}_1) - \mathsf{Adv}_\mathcal{A}(\mathsf{G}_2)| \leq \mathsf{Adv}^{\ell,Q\text{-}\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}}(\lambda).$$

By Lemma 1, the latter advantage is negligible by the DDH assumption in $\mathbb{G}_1$.

Upon receiving an (ℓ, Q)-fold DDH challenge $([\mathbf{b}]_1, \{[\mathbf{z}_i]_1\}_{i \in [Q]})$, $\mathcal{B}$ samples $b \leftarrow_\mathsf{R} \{0,1\}$, $\mathbf{a} \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$, $\mathbf{k} \leftarrow_\mathsf{R} \{0,1\}^\ell$, sets $\mathsf{pk} := ([\mathbf{a}]_1, [\mathbf{k}^\top \mathbf{a}]_T)$, $\mathsf{msk}_\mathsf{N} := \frac{\mathbf{k}^\top \mathbf{a}}{\|\mathbf{a}\|_2^2} \cdot \mathbf{a}$, and returns $(\mathsf{pk}, \mathsf{msk}_\mathsf{N})$ to $\mathcal{A}$. On the i'th query $\mathsf{O}_\mathsf{Enc}(\mathbf{w}, [m]_T)$, $\mathcal{B}$ computes $[m_0]_T := [m]_T$, $[m_1]_T \leftarrow_\mathsf{R} \mathbb{G}_T$, and returns $([\mathbf{z}_i - \mathbf{w}]_1, [\mathbf{k}^\top \mathbf{z}_i + m_b]_T)$ to $\mathcal{A}$.

Game G_3. We switch the challenge ciphertexts to $([\mathbf{b}s - \mathbf{w}]_1, [\gamma s]_T + [m_b]_T)$ where $s \leftarrow_\mathsf{R} \mathbb{Z}_p$, and $\mathbf{b} \leftarrow_\mathsf{R} \mathbb{Z}_p^\ell$, $\gamma \leftarrow_\mathsf{R} \mathbb{Z}_p$ independent of $\mathbf{a}$ used in the public key and in msk_N. We show that the games G_2 and G_3 are statistically close, using the left over hash lemma [**?**] recalled in Lemma 2, which implies that $(\mathbf{a}, \mathbf{b}, \mathbf{k}^\top \mathbf{a}, \mathbf{k}^\top \mathbf{b})$ is statistically close (within statistical distance $2^{-\lambda}$) from $(\mathbf{a}, \mathbf{b}, \mathbf{k}^\top \mathbf{a}, \gamma)$, where $\gamma \leftarrow_\mathsf{R} \mathbb{Z}_p$. The first distribution corresponds to the distribution of the game G_2, whereas the second distribution corresponds to the game G_3. Note that pk and msk_N can be computed from $(\mathbf{a}, \mathbf{k}^\top \mathbf{a})$. Thus, we have

$$|\mathsf{Adv}_\mathcal{A}(\mathsf{G}_2) - \mathsf{Adv}_\mathcal{A}(\mathsf{G}_3)| \leq 2^{-\lambda}.$$

Game G_4. We change all the messages in the challenge ciphertexts to uniformly random, regardless of the random bit $b \leftarrow_\mathsf{R} \{0,1\}$. Namely, in game G_4,

$\mathsf{O}_{\mathsf{Enc}}(\mathbf{w}, [m]_T)$, returns $([\mathbf{b}s]_1, [r]_T)$, where $[r]_T \leftarrow_{\mathsf{R}} \mathbb{G}_T$ and $s \leftarrow_{\mathsf{R}} \mathbb{Z}_p$ are sampled freshly for each query to $\mathsf{O}_{\mathsf{Enc}}$. Clearly:

$$\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_4) = 0.$$

To prove that game G_4 is computationally indistinguishable from G_3, we use the DDH assumption in $\mathbb{G}_1$ to switch $([s]_1, [\gamma s]_T)$ to $([s]_1, [r]_T)$. Namely, we build a PPT adversary $\mathcal{B}_3$ such that:

$$|\mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_3) - \mathsf{Adv}_{\mathcal{A}}(\mathsf{G}_4)| \leq \mathsf{Adv}^{1,Q_{\mathsf{Enc}}\text{-}\mathsf{DDH}}_{\mathbb{G}_1,\mathcal{B}_3}(\lambda),$$

where Q_{Enc} denotes the number of queries to $\mathsf{O}_{\mathsf{Enc}}$.

Upon receiving a $1, Q_{\mathsf{Enc}}$-fold DDH challenge $\{[s_i]_1, [z_i]_1\}_{i\in[Q_{\mathsf{Enc}}]})$, $\mathcal{B}_3$ samples $b \leftarrow_{\mathsf{R}} \{0,1\}$, $\mathbf{a}, \mathbf{b} \leftarrow_{\mathsf{R}} \mathbb{Z}_p^{\ell}$, $\mathbf{k} \leftarrow_{\mathsf{R}} \{0,1\}^{\ell}$, thanks to which it can compute mpk, $\mathsf{msk}_{\mathsf{N}}$, which it forwards to $\mathcal{A}$. On the i'th query of $\mathcal{A}$ to $\mathsf{O}_{\mathsf{Enc}}(\mathsf{id}, \mathbf{w}, [m]_T)$, $\mathcal{B}_3$ sets $[m_0]_T := [m]_T$, $[m_1]_T \leftarrow_{\mathsf{R}} \mathbb{G}_T$, and returns $([\mathbf{b}s_i]_1, [z_i]_T + [m_b]_T)$ to $\mathcal{A}$. When $[z_i]_1$ is of the form $[\gamma s_i]_1$, $\mathcal{B}_3$ simulates the game G_3, whereas it simulates the game G_4 when $[z_i]_1 \leftarrow_{\mathsf{R}} \mathbb{G}_1$. □