Identity-Based Encryption with Master Key-Dependent Message Security and Applications

David Galindo[1], Javier Herranz[2], and Jorge Villar[2]

[1] University of Luxembourg
e-mail: david.galindo@uni.lu
[2] Universitat Politècnica de Catalunya, Dept. Matemàtica Aplicada IV (Spain)
e-mail: {jherranz,jvillar}@ma4.upc.edu

Abstract. We introduce the concept of identity-based encryption (IBE) with master key-dependent chosen-plaintext (mKDM-sID-CPA) security. These are IBE schemes that remain secure even after the adversary sees encryptions, under some initially selected identities, of functions of the master secret key(s). We then propose a generic construction of chosen-ciphertext secure key-dependent encryption (KDM-CCA) schemes in the public key setting starting from mKDM-sID-CPA secure IBE schemes. This is reminiscent to the celebrated work by Canetti, Halevi and Katz (Eurocrypt 2004) on the traditional key-oblivious setting. Previously only one generic construction of KDM-CCA secure public key schemes was known, due to Camenisch, Chandran and Shoup (Eurocrypt 2009), and it required non-interactive zero knowledge proofs (NIZKs). Our transformation shows that NIZKs are not intrinsic to KDM-CCA public key encryption. Additionally, we are able to instantiate our new concept under the Rank assumption on pairing groups and for affine functions of the secret keys. The scheme builds on previous work by Boneh, Halevi, Hamburg and Ostrovsky (Crypto 2008). Our concrete schemes are only able to provide security against a *bounded* number of encryption queries, which is enough in some practical scenarios. As a corollary we obtain a KDM-CCA secure public key encryption scheme, in the standard model, whose security reduction to a static assumption is independent of the number of challenge queries. As an independent contribution, we give new and better reductions between the Rank problem (previously named as Matrix DDH problem) and the Decisional Linear and the Decisional 3-Party Diffie-Hellman problems.

1 Introduction

Until recently public key encryption (PKE) schemes were only required to provide confidentiality against adversaries that see encryptions of plaintexts that depend solely on public information. That is, it was assumed (and even advocated) that an encryption scheme would never be used to encrypt its own decryption key. This requirement is certainly reasonable for many applications, but it has been challenged both by practical and foundational reasons [1, 14]. The paradigmatic case is the scenario of *circular encryptions*, where for $n \geq 2$ public/secret key pairs $(pk_1, sk_1), \ldots, (pk_n, sk_n)$, the adversary is given the ciphertexts $Enc_{pk_1}(sk_2), Enc_{pk_2}(sk_3), \ldots, Enc_{pk_n}(sk_1)$, and still semantic security shall hold. Thus a dedicated stronger security notion called *key-dependent message* security has emerged in the last few years [5]. Roughly speaking, it is required that semantic security holds even if the adversary sees encryptions of plaintexts that depend on the decryption keys. For the motivation, applications and history of key-dependent message security we refer to the excellent survey by Teranishi, Malkin and Yung [23].

The first breakthrough was due to Boneh, Halevi, Hamburg and Ostrovsky (BHHO) [9], who proposed a public key encryption scheme with indistinguishability against key-dependent chosen-plaintext attacks (KDM-CPA) in the standard model under the Decisional Diffie-Hellman assumption for affine functions of the secret key. Shortly after Applebaum, Cash, Peikert, and Sahai [3] proposed an efficient KDM-CPA secure scheme for affine functions under the Learning Parity with Noise assumption. Brakerski and Goldwasser [11] extended the BHHO scheme to a suite of KDM-CPA schemes secure under subgroup indistinguishability assumptions.

Camenisch, Chandran and Shoup [13] proposed a generic construction of chosen-ciphertext secure key-dependent encryption (KDM-CCA) schemes in the public key setting, that requires in particular a KDM-CPA secure scheme and specialized non-interactive zero knowledge proofs (NIZKs). By applying their transformation to (a variation of) the BHHO scheme, they obtained a KDM-CCA secure scheme under the Decision Linear assumption on pairing groups. This was the only generic construction of KDM-CCA secure public key encryption schemes in the standard model before our work. Concurrently to our work, Hofheinz [20]

has proposed a PKE scheme with KDM-CCA security in the standard model with compact ciphertexts; his construction is direct and does not use key-dependent IBE.

1.1 Our Contribution

We initiate here the study of identity-based encryption (IBE) schemes secure against key dependent messages[1]. This has a double interest, since IBE is of interest by itself [28] and because of its numerous applications [8]. In IBE there are two types of secret keys, on the one hand a master secret key SK_i corresponding to the master public key PK_i; on the other hand the secret keys $sk[id]$ belonging to individual users id. This potentially gives rise to two levels of key-dependent message security, depending on whether the adversary is allowed to ask for encryptions of functions of the master-keys or the user-keys. We choose here to deal only with master key-dependent messages (mKDM security). A concurrent work [2] deals with the user key-dependent message security. The first reason is that this allows us to update mKDM-sID-CPA. Secondly, master key-dependence seems harder to achieve than user-key dependence, and that in some cases master key-dependent security implies a restricted form of user-key dependent security "for free" (see Section 4.1 for the case of our scheme).

Informally, we say that an IBE scheme has master key-dependent indistinguishability against selective-identity and chosen plaintext attacks (mKDM-sID-CPA security for short) if no adversary is able to distinguish between encryptions of a particular message $\mathfrak{m}$ and encryptions of some functions of a set of master secret keys, under a certain set identities chosen by the adversary ahead of time. We are able to give an instantiation of a mKDM-sID-CPA secure IBE in the standard model, under the Rank assumption over bilinear groups. The Rank assumption states that it is difficult to distinguish whether an $n \times n$ matrix has rank r_1 or r_2, where $2 \leq r_1 < r_2 \leq n$. As an additional contribution, which may be of independent interest, we give a new reduction between the Rank problem and both the Decisional Linear and the Decisional 3-Party Diffie-Hellman problems. Our new reduction improves that of [26] from a linear to a logarithmic factor and can be used to improve the reduction from the Rank assumption to the Decisional Diffie-Hellman problem given in [9] in a similar fashion.

One of the most well-known applications of IBE in the theory of cryptography is the CHK generic construction of chosen-ciphertext secure public key encryption out of chosen-plaintext secure identity-based encryption. We show that the same transformation can be applied to the KDM setting, resulting in KDM-CCA secure public key encryption out of mKDM-sID-CPA secure identity-based encryption. Thus we show a practical generic construction for key-dependent chosen-ciphertext security that dispenses with the need of NIZKs from [13]. In other words, we show that NIZKs are not inherent to KDM-CCA public key encryption. Plugging our concrete IBE scheme(s) into the Canetti-Halevi-Katz transformation gives rise to KDM-CCA secure encryption scheme(s) with security based on either the Decisional Linear assumption or the Decisional 3-Party Diffie-Hellman assumption. Surprisingly, it turns out that if we use in our instantiations a strongly unforgeable one-time signature scheme with tight security in the multi-user setting (i.e. more than one public keys are considered in the security definition [24]), we obtain KDM-CCA secure schemes whose security reduction loss factor does not depend on the number of encryption queries. Let us point out that the Camenish *et al.* KDM-CCA scheme incurs in a loss factor in its reduction that depends linearly on the total number of encryption queries. Our result (partially) solves an open problem posed by Bellare, Boldyreva and Micali [4] regarding chosen-ciphertext secure encryption schemes in the multi-user setting. One drawback of our chosen-ciphertext secure schemes is that the public key size depends on the number of encryption queries per public key (but importantly ciphertext-size does not); in other words, we were only able to prove security against a *bounded* number of encryption queries per public key. However, this is enough in some practical scenarios such as that of circular security, where the number of encryption queries per public key equals the binary length of a secret key. Previously Hofheinz and Unruh [21] proposed KDM-CPA symmetric encryption schemes secure against a bounded number of encryption queries but with some limitations, e.g. messages length must be smaller than secret key length. Such restrictions are overcome here. We stress here that the concurrent construction of uKDM-sID-CPA secure IBE in [2] has a similar drawback: therein, the size of the

[1] See [2] for a concurrent and independent work on the same topic.

master public key, the user secret keys and the ciphertext depend on the parameter n, which is the maximum number of user secret keys involved in an encryption query.

1.2 Organization

In Section 2 we introduce most of the notation to be used throughout the paper, as well as the hardness assumptions we will work with. Additionally we recall previous KDM security notions for public key encryption. In Section 3 we define master key-dependent indistinguishability against selective-identity and chosen-plaintext attacks for identity-based encryption. We prove then that the celebrated CHK transformation from passively-secure IBE to chosen-ciphertext PKE also holds in the KDM setting. Section 4 contains the bulk of our contribution, namely an instantiation of identity-based encryption with key-dependent security in the standard model under the Decisional Linear assumption or the Decisional 3-Party Diffie-Hellman assumption. Actually two schemes are presented: the first one is simpler to reason with and its security reduction is easier to understand; the second one improves the efficiency by a factor 2λ, where λ is the security parameter. We end in Section 5 by outlining future research directions.

2 Preliminaries

In this section we list some of the notation, probability results and computational assumptions that will be used afterwards.

2.1 Some Probability and Linear Algebra

A distribution of probability $\mathcal{D}$ defined on a set $\mathcal{X}$ is ϵ-uniform if $\sum_{x\in\mathcal{X}} \left|\mathcal{D}(x) - \frac{1}{|\mathcal{X}|}\right| \leq \epsilon$.
A family of hash functions $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$ is 2-universal if $\forall x_1, x_2 \in \mathcal{X}, x_1 \neq x_2$, we have $\Pr_{h\in\mathcal{H}}[h(x_1) = h(x_2)] \leq \frac{1}{|\mathcal{Y}|}$.

Lemma 1. ***(Simplified left-over hash lemma, [9]).*** *Let $\mathcal{H}$ be a 2-universal hash family from a set $\mathcal{X}$ to a set $\mathcal{Y}$. Then the distribution $(h, h(x))$, where $h \in_{\mathrm{R}} \mathcal{H}$ and $x \in_{\mathrm{R}} \mathcal{X}$, is $\sqrt{\frac{|\mathcal{Y}|}{4|\mathcal{X}|}}$-uniform on $\mathcal{H} \times \mathcal{X}$.*

Corollary 1. *Let $\mathbf{W} \in_{\mathrm{R}} {\mathbb{Z}_q}^{2\times\ell;2}$ and $\boldsymbol{s} \in_{\mathrm{R}} \{0,1\}^\ell$. Then the distribution $(\mathbf{W}, \mathbf{W}s)$ is $\frac{1}{q}$-uniform on ${\mathbb{Z}_q}^{2\times\ell;2} \times \mathbb{Z}_q^2$, provided $\ell \geq 4\log q$.*

Proof. We consider $\mathcal{X} = \{0,1\}^\ell$ and $\mathcal{Y} = \mathbb{Z}_q^2$. For any matrix $\mathbf{W} \in {\mathbb{Z}_q}^{2\times\ell;2}$, we consider the hash function $h_{\mathbf{W}} : \mathcal{X} \to \mathcal{Y}$ defined by $h_{\mathbf{W}}(\boldsymbol{s}) = \mathbf{W}\boldsymbol{s}$. It is quite easy to check that the family of hash functions $\mathcal{H} = \{h_{\mathbf{W}} : \mathcal{X} \to \mathcal{Y}\}_{\mathbf{W}\in{\mathbb{Z}_q}^{\ell_1\times\ell_2;r}}$ is 2-universal.

Applying Lemma 1 to $\mathcal{X}, \mathcal{Y}, \mathcal{H}$, we obtain that the distribution $(\mathbf{W}, \mathbf{W}\boldsymbol{s})$ is $\sqrt{\frac{q^2}{4\cdot 2^\ell}}$-uniform on ${\mathbb{Z}_q}^{2\times\ell;2} \times \mathbb{Z}_q^2$. Since $\ell \geq 4\log q$, we have $\sqrt{\frac{q^2}{4\cdot 2^\ell}} \leq \frac{1}{2q} < \frac{1}{q}$, as desired. □

In the security proofs of this paper we will use the following technical results, some of them arising from basic linear algebra. For convenience we will use the notation $\mathbf{A} \oplus \mathbf{B}$ for block matrix concatenation:

$$\mathbf{A} \oplus \mathbf{B} = \left(\begin{array}{c|c} \mathbf{A} & 0 \\ \hline 0 & \mathbf{B} \end{array}\right)$$

In addition, we will denote $\mathbf{I}_\ell$ and $0_{\ell_1\times\ell_2}$ for the neutral element in $\mathrm{GL}_\ell(\mathbb{Z}_q)$ and the null matrix in ${\mathbb{Z}_q}^{\ell_1\times\ell_2}$, respectively. The shorthand $0_\ell = 0_{\ell\times\ell}$ will also be used. Given a matrix $\mathbf{A} \in {\mathbb{Z}_q}^{\ell_1\times\ell_2}$, the transpose of $\mathbf{A}$ is denoted as $\mathbf{A}^\top$, and the vector subspace spanned by the columns of $\mathbf{A}$ is denoted as $\mathrm{Span}(\mathbf{A}) \subseteq \mathbb{Z}_q^{\ell_1}$, which dimension equals $\mathrm{rank}(\mathbf{A})$. The orthogonal subspace of a vector subspace $V \subset \mathbb{Z}_q^\ell$ of dimension r is denoted as $V^\perp$, and its dimension is $\ell - r$. Notice that as we are working on positive characteristic fields, the intersection $V \cap V^\perp$ can be nontrivial.

Lemma 2. *The statistical distance of the two probability distributions* $\mathbf{A}_0 \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell}$ *and* $\mathbf{A}_1 \in_{\mathrm{R}} GL_\ell(\mathbb{Z}_q)$ *is upper bounded by* $1 - q^{-\ell^2}\,|GL_\ell(\mathbb{Z}_q)| < 1/(q-1)$.

Lemma 3. *The following three natural group actions are transitive:*[2]

1. *the left-action of* $GL_{\ell_1}(\mathbb{Z}_q)$ *on* ${\mathbb{Z}_q}^{\ell_1\times\ell_2;\ell_2}$, *for* $\ell_1 \geq \ell_2$, *defined by* $\mathbf{A} \mapsto \mathbf{UA}$, *where* $\mathbf{U} \in GL_{\ell_1}(\mathbb{Z}_q)$ *and* $\mathbf{A} \in {\mathbb{Z}_q}^{\ell_1\times\ell_2;\ell_2}$,
2. *the right-action of* $GL_{\ell_2}(\mathbb{Z}_q)$ *on* ${\mathbb{Z}_q}^{\ell_1\times\ell_2;\ell_1}$, *for* $\ell_1 \leq \ell_2$, *defined by* $\mathbf{A} \mapsto \mathbf{AV}$, *where* $\mathbf{V} \in GL_{\ell_2}(\mathbb{Z}_q)$ *and* $\mathbf{A} \in {\mathbb{Z}_q}^{\ell_1\times\ell_2;\ell_1}$,
3. *the left-right-action of* $GL_{\ell_1}(\mathbb{Z}_q) \times GL_{\ell_2}(\mathbb{Z}_q)$ *on* ${\mathbb{Z}_q}^{\ell_1\times\ell_2;r}$, *defined by* $\mathbf{A} \mapsto \mathbf{UAV}$, *where* $\mathbf{U} \in GL_{\ell_1}(\mathbb{Z}_q)$, $\mathbf{V} \in GL_{\ell_2}(\mathbb{Z}_q)$ *and* $\mathbf{A} \in {\mathbb{Z}_q}^{\ell_1\times\ell_2;r}$.

Lemma 4 (Rank Decomposition). *Given any matrix* $\mathbf{A} \in {\mathbb{Z}_q}^{\ell_1\times\ell_2;r}$, *there exist matrices* $\mathbf{L} \in {\mathbb{Z}_q}^{\ell_1\times r;r}$ *and* $\mathbf{R} \in {\mathbb{Z}_q}^{r\times\ell_2;r}$ *such that* $\mathbf{A} = \mathbf{LR}$.

Lemma 5. *Given two matrices* $\mathbf{A}, \mathbf{B} \in {\mathbb{Z}_q}^{\ell_1\times\ell_2}$ *such that Span*$(\mathbf{A})$ = *Span*$(\mathbf{B})$, *there exists* $\mathbf{C} \in GL_{\ell_2}(\mathbb{Z}_q)$ *such that* $\mathbf{B} = \mathbf{AC}$.

2.2 Bilinear Pairings, Matrices and Hardness Assumptions

Let $\mathcal{G}$ be a multiplicative group of prime order q admitting a bilinear pairing. That is, let $\mathcal{G}_T$ be a multiplicative group of prime order q and let $e(\cdot,\cdot) : \mathcal{G} \times \mathcal{G} \to \mathcal{G}_T$ an efficiently computable bilinear map. We will denote as $g_T = e(g,g)$ the generator of $\mathcal{G}_T$ induced by g a given generator of $\mathcal{G}$. Note that, due to the bilinear properties of the pairing, for any two integers $a, b \in \mathbb{Z}_q$ we have $g_T^{ab} = e(g^a, g^b) = e(g^a, g)^b = e(g^b, g)^a$.

These operations extend to vectors and matrices in a natural way. Let ${\mathbb{Z}_q}^{\ell_1\times\ell_2}$ denote the set of all $\ell_1 \times \ell_2$ matrices and ${\mathbb{Z}_q}^{\ell_1\times\ell_2;r}$ the matrices with rank r. In the special case of invertible matrices we will write $\mathrm{GL}_\ell(\mathbb{Z}_q) = {\mathbb{Z}_q}^{\ell\times\ell;\ell}$. Let $\mathcal{G}^{\ell_1\times\ell_2}$ and ${\mathcal{G}_T}^{\ell_1\times\ell_2}$ denote the set of all $\ell_1 \times \ell_2$ matrices over $\mathcal{G}$ and $\mathcal{G}_T$ respectively. Therefore, for any two matrices $\mathbf{A} \in {\mathbb{Z}_q}^{\ell_1\times\ell_2}$ and $\mathbf{B} \in {\mathbb{Z}_q}^{\ell_2\times\ell_3}$, we have $g^{\mathbf{AB}} = (g^{\mathbf{A}})^{\mathbf{B}} \in \mathcal{G}^{\ell_1\times\ell_3}$. Here, if the (i,j) component of matrix A is denoted as $a_{i,j} \in \mathbb{Z}_q$, then $g^{\mathbf{A}}$ denotes the matrix obtained, component-wise, by computing the values $g^{a_{i,j}}$. Note that $g^{\mathbf{AB}}$ can be easily computed from $g^{\mathbf{A}}$ and B. Again, we can naturally extend these definitions to matrices and bilinear pairings: if $\mathbf{A} \in {\mathbb{Z}_q}^{\ell_1\times\ell_2}$ and $\mathbf{B} \in {\mathbb{Z}_q}^{\ell_2\times\ell_3}$, then $e(g^{\mathbf{A}}, g^{\mathbf{B}}) = g_T^{\mathbf{AB}}$. Once again, $g_T^{\mathbf{AB}}$ can be computed from $g_T^{\mathbf{A}}$ and B. Furthermore, if $\mathbf{C} \in {\mathbb{Z}_q}^{\ell_3\times\ell_4}$, then it holds $g_T^{\mathbf{ABC}} = e(g^{\mathbf{AB}}, g^{\mathbf{C}}) = e(g^{\mathbf{A}}, g^{\mathbf{BC}}) \in {\mathcal{G}_T}^{\ell_1\times\ell_4}$.

The security of our schemes can be reduced to the hardness of the Decisional Linear (DLin) problem [7] or the Decisional 3-Party Diffie-Hellman (D3DH) problem [22, 10, 19] in the group $\mathcal{G}$.

The DLin problem consists in distinguishing between the distributions $(g, g^x, g^y, g^z, g^t, g^{(x^{-1}z+y^{-1}t)}) \in \mathcal{G}^6$ and $(g, g^x, g^y, g^z, g^t, g^u) \in \mathcal{G}^6$, where g is a generator of $\mathcal{G}$ and $x, y, z, t, u \in_{\mathrm{R}} \mathbb{Z}_q$ are chosen independently and at random. The problem is formally defined through the following two experiments between a challenger and a solver $\mathcal{A}_{\mathbf{DLin}}$. Experiment $\mathbf{ExpDLin}^b_{\mathcal{A}_{\mathbf{DLin}}}(\mathcal{G})$ is defined as follows, for $b = 0, 1$.

1. The challenger chooses a generator g of $\mathcal{G}$ and random $x, y, z, t, u \in_{\mathrm{R}} \mathbb{Z}_q$ independently and uniformly distributed.
   In Experiment $b = 0$, the challenger sends $(g, g^x, g^y, g^z, g^t, g^{(x^{-1}z+y^{-1}t)}) \in \mathcal{G}^6$ to $\mathcal{A}_{\mathbf{DLin}}$.
   In Experiment $b = 1$, it sends $(g, g^x, g^y, g^z, g^t, g^u) \in \mathcal{G}^6$ to $\mathcal{A}_{\mathbf{DLin}}$.
2. The solver $\mathcal{A}_{\mathbf{DLin}}$ outputs a bit $b' \in \{0, 1\}$.

Let us denote as Ω_b the event that $\mathcal{A}_{\mathbf{DLin}}$ outputs $b' = 1$ in Experiment $\mathbf{ExpDLin}^b_{\mathcal{A}_{\mathbf{DLin}}}(\mathcal{G})$. Let $\mathbf{AdvDLin}_{\mathcal{A}_{\mathbf{DLin}}}(\mathcal{G}) = |\Pr[\Omega_0] - \Pr[\Omega_1]|$. We can then define $\mathbf{AdvDLin}(\mathcal{G}; t) = \max_{\mathcal{A}_{\mathbf{DLin}}} \{\mathbf{AdvDLin}_{\mathcal{A}_{\mathbf{DLin}}}(\mathcal{G})\}$, where the maximum is taken over adversaries $\mathcal{A}_{\mathbf{DLin}}$ running in time at most t.

[2] The action of a group G on a set A is transitive if for any $a, b \in A$ there exists $g \in G$ such that $b = g \cdot a$. As a consequence, if $g \in_{\mathrm{R}} G$ then for any $a \in A$, $g \cdot a$ is uniform in A.

Definition 1. *The* Decisional Linear assumption *in* $\mathcal{G}$ *states that* $\mathbf{AdvDLin}(\mathcal{G};t)$ *is negligible in* $\lambda = \log|\mathcal{G}|$ *for any value of* t *that is polynomial in* λ.

For a group $\mathcal{G}$ with prime order $q > 2^\lambda$ and a generator g of $\mathcal{G}$, the Decisional 3-Party Diffie-Hellman (D3DH) problem [22, 10, 19] consists in distinguishing between the distributions $(g, g^x, g^y, g^z, g^{xyz}) \in \mathcal{G}^5$ and $(g, g^x, g^y, g^z, g^t) \in \mathcal{G}^5$, where $x, y, z, t \in_{\mathrm{R}} \mathbb{Z}_q$ are chosen independently at random. The problem is formally defined through the following two experiments between a challenger and a solver $\mathcal{A}_{\mathbf{D3DH}}$. Experiment $\mathbf{ExpD3DH}^b_{\mathcal{A}_{\mathbf{D3DH}}}(\mathcal{G})$ is defined as follows, for $b = 0, 1$.

1. The challenger chooses random $x, y, z, t \in_{\mathrm{R}} \mathbb{Z}_q$ independently and uniformly distributed.
   In Experiment $b = 0$, the challenger sends the tuple $(g, g^x, g^y, g^z, g^{xyz}) \in \mathcal{G}^5$ to $\mathcal{A}_{\mathbf{D3DH}}$.
   In Experiment $b = 1$, the challenger sends the tuple $(g, g^x, g^y, g^z, g^t) \in \mathcal{G}^5$ to $\mathcal{A}_{\mathbf{D3DH}}$.
2. The solver $\mathcal{A}_{\mathbf{D3DH}}$ outputs a bit $b' \in \{0, 1\}$.

Let Ω_b be the event that $\mathcal{A}_{\mathbf{D3DH}}$ outputs $b' = 1$ in $\mathbf{ExpD3DH}^b_{\mathcal{A}_{\mathbf{D3DH}}}(\mathcal{G})$. Let $\mathbf{AdvD3DH}_{\mathcal{A}_{\mathbf{D3DH}}}(\mathcal{G}) = |\Pr[\Omega_0] - \Pr[\Omega_1]|$ and let $\mathbf{AdvD3DH}(\mathcal{G}, t) = \max_{\mathcal{A}_{\mathbf{D3DH}}} \{\mathbf{AdvD3DH}_{\mathcal{A}_{\mathbf{D3DH}}}(\mathcal{G})\}$, where the maximum is taken over adversaries $\mathcal{A}_{\mathbf{D3DH}}$ running in time at most t.

Definition 2. *The* Decisional 3-Party Diffie-Hellman assumption *in a group* $\mathcal{G}$ *states that* $\mathbf{AdvD3DH}(\mathcal{G}, t)$ *is negligible in* $\lambda = \log|\mathcal{G}|$ *for any value of* t *that is polynomial in* λ.

2.3 The Rank Problem

We consider an assumption related to matrices. Given a (multiplicative) cyclic group $\mathcal{G}$ of prime order q, the $\mathbf{Rank}(\mathcal{G}, \ell_1, \ell_2, r, s)$ problem informally consists of distinguishing if a given matrix in $\mathbb{Z}_q^{\ell_1 \times \ell_2}$ has rank r or has rank s for given integers $r \neq s$, when the matrix is hidden in the exponent of a generator g of $\mathcal{G}$. The problem is formally defined through the following two experiments between a challenger and a distinguisher $\mathcal{A}_{\mathbf{Rank}}$. For $b = 0, 1$, experiment $\mathbf{ExpRank}^b_{\mathcal{A}_{\mathbf{Rank}}}(\mathcal{G}, \ell_1, \ell_2, r, s)$ is defined as follows.

1. In Experiment $b = 0$, the challenger chooses $\mathbf{M} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell_1 \times \ell_2; r}$ and sends $g^{\mathbf{M}}$ to $\mathcal{A}_{\mathbf{Rank}}$.
   In Experiment $b = 1$, it chooses $\mathbf{M} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell_1 \times \ell_2; s}$ and sends $g^{\mathbf{M}}$ to $\mathcal{A}_{\mathbf{Rank}}$.
2. The solver $\mathcal{A}_{\mathbf{Rank}}$ outputs a bit $b' \in \{0, 1\}$.

Let us denote as Ω_b the event that $\mathcal{A}_{\mathbf{Rank}}$ outputs $b' = 1$ in Experiment $\mathbf{ExpRank}^b_{\mathcal{A}_{\mathbf{Rank}}}(\mathcal{G}, \ell_1, \ell_2, r, s)$. For any such adversary $\mathcal{A}_{\mathbf{Rank}}$ let

$$\mathbf{AdvRank}_{\mathcal{A}_{\mathbf{Rank}}}(\mathcal{G}, \ell_1, \ell_2, r, s) = |\Pr[\Omega_0] - \Pr[\Omega_1]|$$

We can then define

$$\mathbf{AdvRank}(\mathcal{G}, \ell_1, \ell_2, r, s; t) = \max_{\mathcal{A}_{\mathbf{Rank}}} \{\mathbf{AdvRank}_{\mathcal{A}_{\mathbf{Rank}}}(\mathcal{G}, \ell_1, \ell_2, r, s)\},$$

where the maximum is taken over adversaries $\mathcal{A}_{\mathbf{Rank}}$ running in time at most t.

Definition 3. *The* $\mathbf{Rank}(\mathcal{G}, \ell_1, \ell_2, r, s)$ *assumption in a group* $\mathcal{G}$ *states that* $\mathbf{AdvRank}(\mathcal{G}, \ell_1, \ell_2, r, s; t)$ *is negligible in* $\lambda = \log|\mathcal{G}|$ *for any value of* t *that is polynomial in* λ.

The Rank assumption appeared in recent papers under the names Matrix-DDH [9] and Matrix d-Linear [26]. Therein, it was already proved that the Rank problem is harder than the Decisional Linear problem. However, the reduction given in the next proposition substantially improves the reductions previously given. Namely, the loss factor is no longer linear but logarithmic in the rank.

Proposition 1. *For any ℓ_1, ℓ_2, r, s such that $2 \le s < r \le \min(\ell_1, \ell_2)$ we have*

$$\begin{aligned}\mathbf{AdvRank}(\mathcal{G}, \ell_1, \ell_2, r, s; t) &\le \left\lceil \tfrac{\log(3r)-\log(3s-2)}{\log 3-\log 2} \right\rceil \mathbf{AdvDLin}(\mathcal{G}; t') \\ &\le \lceil 1.71(\log_2 r - \log_2(s-1)) \rceil \, \mathbf{AdvDLin}(\mathcal{G}; t'),\end{aligned}$$

where $t' = t + \mathcal{O}(\ell_1\ell_2(\ell_1+\ell_2))$, taking the cost of an exponentiation in $\mathcal{G}$ as one time unit.

Before proving the proposition, we note that the $\mathbf{Rank}(\mathcal{G}, \ell_1, \ell_2, r, s)$ problem is random self-reducible, because given $\mathbf{M}_0 \in \mathbb{Z}_q^{\ell_1 \times \ell_2; k}$, for random $\mathbf{L} \in_{\mathrm{R}} \mathrm{GL}_{\ell_1}(\mathbb{Z}_q)$ and $\mathbf{R} \in_{\mathrm{R}} \mathrm{GL}_{\ell_2}(\mathbb{Z}_q)$ the product $\mathbf{L}\mathbf{M}_0\mathbf{R}$ is uniformly distributed in $\mathbb{Z}_q^{\ell_1 \times \ell_2; k}$. For the actual proof of Proposition 1, we use the following result.

Lemma 6. *Any distinguisher for $\mathbf{Rank}(\mathcal{G}, \ell_1, \ell_2, k-\delta, k)$, $\ell_1, \ell_2 \ge 3$, $k \ge 3$, $1 \le \delta \le \lfloor \frac{k}{3} \rfloor$ can be converted into a distinguisher for the Decisional Linear* (DLin) *problem, with the same advantage and running essentially within the same time.*

Proof. Given the DLin instance $(g, g^x, g^y, g^z, g^t, g^u)$ the DLin distinguisher builds the $\ell_1 \times \ell_2$ matrix

$$\mathbf{M} = \underbrace{\begin{pmatrix} x & 0 & 1 \\ 0 & y & t \\ z & 1 & u \end{pmatrix} \oplus \cdots \oplus \begin{pmatrix} x & 0 & 1 \\ 0 & y & t \\ z & 1 & u \end{pmatrix}}_{\delta \text{ times}} \oplus I_{k-3\delta} \oplus 0_{(\ell_1-k)\times(\ell_2-k)}$$

and submits the randomized matrix $g^{\mathbf{LMR}}$ to the $\mathbf{Rank}(\mathcal{G}, \ell_1, \ell_2, k-\delta, k)$ distinguisher, where $\mathbf{L} \in_{\mathrm{R}} \mathrm{GL}_{\ell_1}(\mathbb{Z}_q)$ and $\mathbf{R} \in_{\mathrm{R}} \mathrm{GL}_{\ell_2}(\mathbb{Z}_q)$. Notice that if $u = x^{-1}z + y^{-1}t \mod q$ then the resulting matrix is a random matrix in $\mathcal{G}^{\ell_1 \times \ell_2; k-\delta}$. Otherwise, it is a random matrix in $\mathcal{G}^{\ell_1 \times \ell_2; k}$. □

We can now apply a hybrid argument to prove Proposition 1. Let us consider the sequence of integers $\{r_i\}$ defined by the recurrence $r_0 = s$ and $r_{i+1} = \lfloor \frac{3r_i}{2} \rfloor$, and let k be the smallest index such that $r_k \ge r$. Then define a sequence of random matrices $\{\mathbf{M}_i\}$, where $\mathbf{M}_i \in_{\mathrm{R}} \mathbb{Z}_q^{\ell_1 \times \ell_2; r_i}$ for $i = 0, \ldots, k-1$, and $\mathbf{M}_k \in_{\mathrm{R}} \mathbb{Z}_q^{\ell_1 \times \ell_2; r}$. For any distinguisher $\mathcal{A}_{\mathbf{Rank}}$ with running time upper bounded by t, let $p_i = \Pr[1 \leftarrow \mathcal{A}_{\mathbf{Rank}}(g^{\mathbf{M}_i})]$. By Lemma 6, we have that for $i = 0, \ldots, k-2$

$$\begin{aligned}|p_{i+1} - p_i| &= \mathbf{AdvRank}_{\mathcal{A}_{\mathbf{Rank}}}(\mathcal{G}, \ell_1, \ell_2, r_{i+1}, r_i) \le \mathbf{AdvDLin}(\mathcal{G}; t'), \\ |p_k - p_{k-1}| &= \mathbf{AdvRank}_{\mathcal{A}_{\mathbf{Rank}}}(\mathcal{G}, \ell_1, \ell_2, r, r_{k-1}) \le \mathbf{AdvDLin}(\mathcal{G}; t')\end{aligned}$$

Therefore, $\mathbf{AdvRank}_{\mathcal{A}_{\mathbf{Rank}}}(\mathcal{G}, \ell_1, \ell_2, r, s) = |p_k - p_0| \le$
$|p_1 - p_0| + \ldots + |p_k - p_{k-1}| \le k \cdot \mathbf{AdvDLin}(\mathcal{G}; t')$.
On the other hand, since $\lfloor \frac{3x}{2} \rfloor \ge \frac{3x-1}{2}$ then $r_k \ge \left(\frac{3}{2}\right)^k \left(s - \frac{2}{3}\right)$, which implies that $k \le \frac{\log(3r)-\log(3s-2)}{\log 3-\log 2}$. □

Similarly, we can prove that the D3DH problem is easier than the Rank problem.

Proposition 2. *For any ℓ_1, ℓ_2, r, s such that $2 \le s < r \le \min(\ell_1, \ell_2)$*

$$\begin{aligned}\mathbf{AdvRank}(\mathcal{G}, \ell_1, \ell_2, r, s; t) &\le \left\lceil \frac{\log(3r) - \log(3s-2)}{\log 3 - \log 2} \right\rceil \mathbf{AdvD3DH}(\mathcal{G}; t') \le \\ &\le \lceil 1.71(\log_2 r - \log_2(s-1)) \rceil \, \mathbf{AdvD3DH}(\mathcal{G}; t')\end{aligned}$$

Proof. The proof only differs from the proof of Proposition 1 in the 3×3 blocks built from a problem instance, in the proof of Lemma 6. Indeed, given the D3DH instance (g, g^x, g^y, g^z, g^t) the matrix

$$\begin{pmatrix} x & -1 & 0 \\ 0 & y & 1 \\ t & 0 & z \end{pmatrix}$$

has rank 2 or 3 depending on whether $t = xyz \mod q$. □

As a consequence of the results in this section, any appearance of **AdvDLin** in the security results of this paper can be safely replaced with **AdvD3DH**.

2.4 KDM Secure Encryption

A public key encryption scheme Π supporting ciphertexts consists of four probabilistic polynomial algorithms, $\Pi = (\Pi.\mathsf{Stp}, \Pi.\mathsf{KG}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$[3]. The setup protocol $\Pi.\mathsf{Stp}$ takes as input a security parameter λ and outputs some public information pms, including plaintext space $\mathcal{M}$ and secret key space $\mathcal{S}$. The security parameter λ is included in the string pms, which is implicitly an input to the remaining algorithms. The key generation protocol $\Pi.\mathsf{KG}_{\mathsf{pms}}$ on input the empty string ε outputs a pair of secret and public keys, (sk, pk), where the secret key sk belongs to the set $\mathcal{S}$ of possible secret keys. The encryption protocol takes as input a public key pk and a message $m \in \mathcal{M}$ and outputs a ciphertext $C = \Pi.\mathsf{Enc}_{\mathsf{pms}}(pk, m)$. Finally, the decryption protocol takes as input secret key sk and a ciphertext C, and outputs $\widetilde{m} = \Pi.\mathsf{Dec}_{\mathsf{pms}}(sk, C)$, where $\widetilde{m} \in \mathcal{M} \cup \{\perp\}$. The correctness property requires that $\Pi.\mathsf{Dec}_{\mathsf{pms}}(sk, \Pi.\mathsf{Enc}_{\mathsf{pms}}(pk, m)) = m$, for any message $m \in \mathcal{M}$ and parameters pms generated by $\Pi.\mathsf{Stp}$ and any pair (sk, pk) generated by $\Pi.\mathsf{KG}_{\mathsf{pms}}$.

Informally, security with respect to key dependent messages under chosen plaintext attacks (KDM-CPA) requires that an adversary is not able to distinguish between encryptions of a particular message $\mathtt{m}$ and encryptions of some functions (chosen by the adversary from a specific set of functions $\mathcal{F}$) of a set of secret keys. In the case of security with respect to key dependent messages under chosen ciphertext attacks (KDM-CCA), the adversary is given additional access to a decryption oracle that he can query for ciphertexts of his choice, as long as these ciphertexts are different to those the adversary has to distinghish.

Unlike standard security definitions for PKE schemes [18, 27, 17], KDM security notions have been defined from the very beginning [14, 5, 13] in the multi-user setting, namely they involve in general n public keys with $n \geq 1$. For concrete security concerns, in the following definitions two integer parameters $n, q_e \geq 1$ are given as input to the security game, representing respectively the number of users in the system and the maximum number of encryption queries per user allowed to the adversary. We shall see how these two parameters influence security reductions of our and previous KDM secure schemes.

To formalize this notion, we follow the definitions in [13, 23]. Let $n, q_e \geq 1$ be integers and let $\mathcal{F} = \{f : \mathcal{S}^n \to \mathcal{M}\}$ be a finite set of efficiently computable functions. KDM-CPA security of a public key encryption scheme Π is defined with respect to the set of functions $\mathcal{F}$ through the following two experiments between a challenger and an adversary $\mathcal{A}_\Pi$. Let $\mathtt{m} \in \mathcal{M}$ be a fixed message.

Experiment $\mathbf{ExpKDM\text{-}CCA}^{b,\Pi}_{\mathcal{A}_\Pi}(\lambda, n, q_e)$ is defined as follows, for $b = 0, 1$.

1. **Initialization.** The challenger runs $\mathsf{pms} \leftarrow \Pi.\mathsf{Stp}(\lambda)$ and then runs n times $(sk_i, pk_i) \leftarrow \Pi.\mathsf{KG}_{\mathsf{pms}}$ to produce n pairs $(sk_1, pk_1), \ldots, (sk_n, pk_n)$. The public keys $(pk_1, \ldots, pk_n)$ and pms are sent to $\mathcal{A}_\Pi$. A list L_{quer} is initially set to empty.
2. **Queries.** The adversary $\mathcal{A}_\Pi$ can adaptively make two types of queries to the challenger.
   (a) **Encryption queries.** For each $1 \leq i \leq n$ the adversary $\mathcal{A}_\Pi$ can make up to q_e encryption queries of the form (i, f) with $f \in \mathcal{F}$. The challenger computes $m = f(sk_1, \ldots, sk_n) \in \mathcal{M}$, and then sets $C = \Pi.\mathsf{Enc}_{\mathsf{pms}}(pk_i, m)$ in Experiment $b = 0$, and sets $C = \Pi.\mathsf{Enc}_{\mathsf{pms}}(pk_i, \mathtt{m})$ in Experiment $b = 1$. The resulting ciphertext C is sent to $\mathcal{A}_\Pi$ and the tuple (i, C) is added to the list L_{quer}.
   (b) **Decryption queries.** $\mathcal{A}_\Pi$ can make a decryption query of the form (i, C), as long as $(i, C) \notin L_{\mathsf{quer}}$. The challenger sends back to $\mathcal{A}_\Pi$ the output $\Pi.\mathsf{Dec}_{\mathsf{pms}}(sk_i, C)$.
3. **Final guess.** The adversary $\mathcal{A}_\Pi$ outputs a bit $b' \in \{0, 1\}$.

Let us denote as Ω_b the event that $\mathcal{A}_\Pi$ outputs $b' = 1$ in Experiment $\mathbf{ExpKDM\text{-}CCA}^{b,\Pi}_{\mathcal{A}_\Pi}(\lambda, n, q_e)$. For any adversary $\mathcal{A}_\Pi$ as above let $\mathbf{AdvKDM\text{-}CCA}^{\Pi}_{\mathcal{A}_\Pi}(\lambda, n, q_e) = |\Pr[\Omega_0] - \Pr[\Omega_1]|$. For any t, n, q_e we define the advantage function of the scheme Π for key-dependent message security against chosen-ciphertext attacks (KDM-CCA) as $\mathbf{AdvKDM\text{-}CCA}(\Pi, \lambda, n, q_e; t) = \max_{\mathcal{A}_\Pi} \left\{\mathbf{AdvKDM\text{-}CCA}^{\Pi}_{\mathcal{A}_\Pi}(\lambda, n, q_e)\right\}$, where the maximum is over adversaries $\mathcal{A}_\Pi$ with time-complexity t and making no more than q_e encryption queries for each $1 \leq i \leq n$.

[3] Notice that the algorithm $\Pi.\mathsf{Stp}$ was only implicitly considered in [5, 9, 13]. Such an algorithm is needed to ensure that all users in the system share message and secret key spaces, and it is included in the latest work [23].

Definition 4. *A public key encryption scheme Π is polynomially-secure against key dependent chosen-ciphertext attacks with respect to the set of functions $\mathcal{F}$ if* $\mathbf{AdvKDM\text{-}CCA}(\Pi, \lambda, n, q_e; t)$ *is negligible in λ for all polynomial values of t, n, q_e.*

Analogously we can define *indistinguishability against key dependent chosen-plaintext attacks*, denoted KDM-CPA security. The resulting definition is obtained by disallowing any decryption queries in the KDM-CCA experiment. In both cases, we refer to *security against single encryption queries* when $q_e = 1$, which means that the adversary can make several encryption queries but each one for a different public key.

In this work we consider $\mathcal{F}$ to be the set of *affine functions*. This contains as particular cases constant functions (which lead to the notion of IND-CCA security in the multi-user setting [4]) and projections $f_i(sk_1, \ldots, sk_n) = sk_i$, for $1 \leq i \leq n$. An encryption scheme which is KDM-CCA-secure with respect to a set of functions containing projections achieves *clique security*, which in particular captures circular security.

3 From mKDM-sID-CPA Secure IBE to KDM-CCA Secure PKE

In this section we extend the Canetti-Halevi-Katz transformation [16] to build IND-CCA encryption schemes to the key-dependent messages setting. We start by recalling the syntactic definition and security properties of one-time signatures. A (one-time) signature scheme $\Theta = (\Theta.\mathsf{Stp}, \Theta.\mathsf{KG}, \Theta.\mathsf{Sign}, \Theta.\mathsf{Vfy})$ consists of four probabilistic polynomial time protocols. $\mathsf{pms}_\Theta \leftarrow \Theta.\mathsf{Stp}(1^\lambda)$ is the setup protocol, which produces some common public parameters (that will be an implicit input for the rest of protocols) for a given security parameter. $(sk_\Theta, vk_\Theta) \leftarrow \Theta.\mathsf{KG}()$ is the key generation protocol, which outputs a secret signing key sk_Θ and a public verification key vk_Θ. The signing protocol $\theta \leftarrow \Theta.\mathsf{Sign}(sk_\Theta, m)$ takes as input the signing key and a message m, and outputs a signature θ. Finally, the verification protocol $\{1, 0\} \leftarrow \Theta.\mathsf{Vfy}(vk_\Theta, m, \theta)$ takes as input the verification key, a message and a signature, and outputs 1 if the signature is valid, or 0 otherwise.

Regarding security, we consider an adversary F_Θ in the multi-user setting, with N users. F_Θ first receives N verification keys $\{vk_\Theta^{(i)}\}_{1 \leq i \leq N}$ obtained from running $\Theta.\mathsf{Stp}(1^\lambda) \rightarrow \mathsf{pms}_\Theta$ once and then running N times the protocol $\Theta.\mathsf{KG}() \rightarrow (sk_\Theta^{(i)}, vk_\Theta^{(i)})$, for $i = 1, \ldots, N$. The adversary can make at most one signature query of the form (i, m_i), for each $i = 1, \ldots, N$, for messages m_i of his choice, obtaining as answer valid signatures $\Theta.\mathsf{Sign}(sk_\Theta^{(i)}, m_i) \rightarrow \theta_i$. Finally F_Θ outputs a tuple $(i^\star, m^\star, \theta^\star)$. We say that the adversary F_Θ succeeds if $\Theta.\mathsf{Vfy}(vk_\Theta^{(i^\star)}, m^\star, \theta^\star) \rightarrow 1$ and $(m^\star, \theta^\star) \neq (m_{i^\star}, \theta_{i^\star})$.

We denote $\mathcal{F}_\Theta$'s success probability in the above game as $\mathbf{AdvOTS}^\Theta_{\mathcal{F}_\Theta}(\lambda, N)$. The signature scheme Θ is *one-time strongly unforgeable* if $\mathbf{AdvOTS}^\Theta_{\mathcal{F}_\Theta}(\lambda, N)$ is a negligible function of the security parameter $\lambda \in \mathbb{N}$, for any polynomial-time attacker $\mathcal{F}_\Theta$ against Θ and any polynomial value of N. A good candidate is the scheme proposed by Mohassel [25], whose security is tightly related to the hardness of the Discrete Logarithm problem, in the standard model. Although the proof by Mohassel is in the single-user setting, it can be easily adapted to the multi-user setting, by using the self-reducibility properties of the Discrete Logarithm problem.

3.1 mKDM-sID-CPA Identity-Based Encryption

An identity-based encryption scheme Γ consists of five probabilistic polynomial algorithms, $\Gamma = (\Gamma.\mathsf{Stp}, \Gamma.\mathsf{Mkg}, \Gamma.\mathsf{Ukg}, \Gamma.\mathsf{Enc}, \Gamma.\mathsf{Dec})$. The setup protocol, $\Gamma.\mathsf{Stp}$ takes as input a security parameter λ and outputs some system-wide parameters ibp to be shared by all the master authorities in the system. In particular, ibp includes the description of the sets of admissible identities, plaintexts and ciphertexts, $\mathcal{I}$, $\mathcal{M}$, $\mathcal{C}$ respectively. The string ibp is an implicit input to the remaining algorithms. $\Gamma.\mathsf{Mkg}_{\mathsf{ibp}}$ on input the empty string outputs (PK, SK), where PK is the master public key and SK is the master secret key. The user's key generation protocol, $\Gamma.\mathsf{Ukg}_{\mathsf{ibp}}$, on input the master secret key SK and an identity id, outputs the user's decryption key $sk[id]$. The encryption algorithm $\Gamma.\mathsf{Enc}_{\mathsf{ibp}}$ takes as input PK, an admissible identity id and a plaintext m and outputs a ciphertext $c = \Gamma.\mathsf{Enc}_{\mathsf{ibp}}(PK, id, m)$. Finally, the decryption protocol takes as input a decryption key $sk[id]$ and an admissible ciphertext c and outputs $\widetilde{m}$, where $\widetilde{m}$ is an admissible plaintext or the reject

symbol $\perp$. The correctness property requires that $\Gamma.\mathsf{Dec}_{\mathsf{ibp}}(\Gamma.\mathsf{Ukg}(SK, id), \Gamma.\mathsf{Enc}_{\mathsf{ibp}}(PK, id, m)) = m$, for any identity $id \in \mathcal{I}$, message $m \in \mathcal{M}$, parameters ibp generated by $\Gamma.\mathsf{Stp}(1^k)$ and any pair (PK, SK) generated by $\Gamma.\mathsf{Mkg}_{\mathsf{ibp}}()$.

Informally, we say that an IBE scheme has master key-dependent indistinguishability against selective-identity and chosen plaintext attacks (mKDM-sID-CPA security, for short) if no adversary is able to distinguish between encryptions of a particular message $\mathfrak{m}$ and encryptions of some functions (chosen by the adversary from a specific set of functions $\mathcal{F}$) of a set of master secret keys.

To formalize this notion, we extend the definitions in [15, 9, 13]. Let $n, q_e \geq 1$ be integers and let $\mathcal{F} = \{f : \mathcal{T}^n \to \mathcal{M}\}$ be a finite set of efficiently computable functions, where $\mathcal{T}$ is the set of master secret keys and $\mathcal{M}$ the set of admissible plaintexts. mKDM-sID-CPA security is defined with respect to the set of functions $\mathcal{F}$ through the following two experiments between a challenger and an adversary $\mathcal{A}_\Gamma$. Let $\mathfrak{m} \in \mathcal{M}$ be a fixed message.

Experiment $\mathbf{ExpKDM\text{-}sID\text{-}CPA}^{b,\Gamma}_{\mathcal{A}_\Gamma}(\lambda, n, q_e)$ is defined as follows, for $b = 0, 1$.

1. **Setup.** The challenger runs $\mathsf{ibp} \leftarrow \Gamma.\mathsf{Stp}(\lambda)$. The adversary $\mathcal{A}_\Gamma$ on input ibp outputs a tuple $\mathcal{I}^\star$ of $n \cdot q_e$ identities $\mathcal{I}^\star = (id_1^1, \ldots, id_1^{q_e}, id_2^1, \ldots, id_2^{q_e}, \ldots, id_n^1, \ldots, id_n^{q_e})$.
2. **Initialization.** The challenger runs n times $\Gamma.\mathsf{Mkg}_{\mathsf{ibp}}$ to obtain n pairs $(PK_1, SK_1), \ldots, (PK_n, SK_n)$. The master public keys $(PK_1, \ldots, PK_n)$ are sent to $\mathcal{A}_\Gamma$.
3. **Queries.** The adversary $\mathcal{A}_\Gamma$ can adaptively make two types of queries to the challenger:
   (a) **Encryption Queries.** For every index i such that $1 \leq i \leq n$, a counter j is maintained, with initial value $j \leftarrow 1$. $\mathcal{A}_\Gamma$ can make encryption queries of the form (i, f), where $f \in \mathcal{F}$. The challenger computes $m = f(SK_1, \ldots, SK_n) \in \mathcal{M}$, and then sets $c = \Gamma.\mathsf{Enc}_{\mathsf{ibp}}(PK_i, id_i^j, m)$ in Experiment $b = 0$, and sets $c = \Gamma.\mathsf{Enc}_{\mathsf{ibp}}(PK_i, id_i^j, \mathfrak{m})$ in Experiment $b = 1$, where j is the current counter value. After the ciphertext c is sent to $\mathcal{A}_\Gamma$, the counter is updated as $j \leftarrow j + 1$. $\mathcal{A}_\Gamma$ can make up to q_e encryption queries per index i.
   (b) **Private key Queries.** $\mathcal{A}_\Gamma$ can make users' private key queries of the form (i, id), where $1 \leq i \leq n$ and $id \neq id_i^j$ for all $j \in \{1, \ldots, q_e\}$. The challenger computes $sk_i[id] = \Gamma.\mathsf{Ukg}_{\mathsf{ibp}}(SK_i, id)$ and gives it back to $\mathcal{A}_\Gamma$.
4. **Final guess.** The adversary $\mathcal{A}_\Gamma$ outputs a bit $b' \in \{0, 1\}$.

Let us denote as Ω_b the event that $\mathcal{A}_\Gamma$ outputs $b' = 1$ in Experiment $\mathbf{ExpKDM\text{-}sID\text{-}CPA}^{b,\Gamma}_{\mathcal{A}_\Gamma}(\lambda, n, q_e)$. For any adversary $\mathcal{A}_\Gamma$ as above let $\mathbf{AdvKDM\text{-}sID\text{-}CPA}^{\Gamma}_{\mathcal{A}_\Gamma}(\lambda, n, q_e) = |\Pr[\Omega_0] - \Pr[\Omega_1]|$. For any t, n, q_e we define the advantage function of the scheme Γ against selective-identity and key-dependent chosen plaintext attacks (mKDM-sID-CPA) as $\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma, \lambda, n, q_e; t) = \max_{\mathcal{A}_\Gamma} \left\{\mathbf{AdvKDM\text{-}sID\text{-}CPA}^{\Gamma}_{\mathcal{A}_\Gamma}(\lambda, n, q_e)\right\}$, where the maximum is taken over adversaries $\mathcal{A}_\Gamma$ with time-complexity t.

Definition 5. *An identity-based encryption scheme Γ is polynomially-secure against selective-identity and master key-dependent chosen plaintext attacks* (mKDM-sID-CPA) *with respect to the set of functions $\mathcal{F}$ if* $\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma, \lambda, n, q_e; t)$ *is negligible in λ for polynomial values of n, t, q_e.*

For technical reasons we will additionally consider a restriction of the mKDM-sID-CPA security notion in which the adversary is given only one master key (*i.e.*, $n = 1$) and it can only select a single identity (*i.e.*, $id_1^1 = \cdots = id_1^{q_e}$). We will refer to this notion as mKDM-ssID-CPA security, from single-selective ID, as only one identity can be selected, while the adversary can still make up to q_e encryption queries.

In the rest of the paper, we will sometimes use the explicit notation (n, q_e)-mKDM-sID-CPA security, meaning that the adversary is given n master public keys and it can ask up to q_e encryption queries per master public key. Similarly q_e-mKDM-ssID-CPA security means that the adversary is allowed to make up to q_e encryption queries in the mKDM-ssID-CPA experiment. mKDM-sID-CPA security for single encryption queries is obtained when $q_e = 1$.

3.2 Canetti-Halevi-Katz Transformation in the KDM Setting

Let $\Gamma = (\Gamma.\mathsf{Stp}, \Gamma.\mathsf{Mkg}, \Gamma.\mathsf{Ukg}, \Gamma.\mathsf{Enc}, \Gamma.\mathsf{Dec})$ be an IBE scheme and let $\Theta = (\Theta.\mathsf{KG}, \Theta.\mathsf{Sign}, \Theta.\mathsf{Vfy})$ be a one-time signature scheme. We use the well-knownCanetti-Halevi-Katz transformation [16] to construct from these two primitives a public-key encryption scheme $\Pi = (\Pi.\mathsf{Stp}, \Pi.\mathsf{KG}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$, as follows:

$\Pi.\mathsf{Stp}(1^\lambda)$*:* run $\mathsf{ibp} \leftarrow \Gamma.\mathsf{Stp}(1^\lambda)$ and $\mathsf{pms}_\Theta \leftarrow \Theta.\mathsf{Stp}(1^\lambda)$. Let us recall that ibp contains in particular the description of the admissible identities, plaintexts and ciphertexts for the scheme Γ. We assume that verification keys output by Θ lie in the identities space of Γ. Otherwise choose an encoding mapping the space of verification keys inside the space of admissible identities for Γ. Define the output of the setup protocol as $\mathsf{pms} = (\mathsf{ibp}, \mathsf{pms}_\Theta)$.

$\Pi.\mathsf{KG}_{\mathsf{pms}}()$*:* parse $\mathsf{pms} = (\mathsf{ibp}, \mathsf{pms}_\Theta)$, run $(PK, SK) \leftarrow \Gamma.\mathsf{Mkg}_{\mathsf{ibp}}()$ and define the secret key as $sk = SK$ and the public key as $pk = PK$.

$\Pi.\mathsf{Enc}_{\mathsf{pms}}(pk, m)$*:* to encrypt a plaintext $m \in \mathcal{M}$ for a receiver with public key pk, parse $\mathsf{pms} = (\mathsf{ibp}, \mathsf{pms}_\Theta)$ and proceed as follows. Run $(sk_\Theta, vk_\Theta) \leftarrow \Theta.\mathsf{KG}()$ and set $id = vk_\Theta$; run $c \leftarrow \Gamma.\mathsf{Enc}_{\mathsf{ibp}}(pk, id, m)$; run $\theta \leftarrow \Theta.\mathsf{Sign}(sk_\Theta, c)$. The final ciphertext output by the algorithm is $C = (vk_\Theta, c, \theta)$.

$\Pi.\mathsf{Dec}_{\mathsf{pms}}(sk, C)$*:* parse $\mathsf{pms} = (\mathsf{ibp}, \Theta)$ and $C = (vk_\Theta, c, \theta)$. First of all, run $\Theta.\mathsf{Vfy}(vk_\Theta, c, \theta)$. If the output bit is 0, then stop and output $\perp$. Otherwise, set $id = vk_\Theta$ and run $sk[id] \leftarrow \Gamma.\mathsf{Ukg}_{\mathsf{ibp}}(sk, id)$ and output the result of running $\Gamma.\mathsf{Dec}_{\mathsf{ibp}}(sk[id], c)$.

Theorem 1. *If Γ enjoys* mKDM-sID-CPA *security with respect to a set of functions $\mathcal{F}$ and the signature scheme Θ is one-time strongly unforgeable, then the constructed public-key encryption scheme Π enjoys* KDM-CCA *security with respect to the same set of functions $\mathcal{F}$.*

Proof. Let us assume, to the contrary, that Π does not enjoy KDM-CCA security with respect to the set of functions $\mathcal{F}$. This means that there exists a successful adversary $\mathcal{A}_\Pi$ in the KDM-CCA security game described in Section 2.4. Let us use the existence of such an adversary $\mathcal{A}_\Pi$ to contruct either a successful adversary $\mathcal{A}_\Gamma$ against the mKDM-sID-CPA security of the scheme Γ or a successful forger $\mathcal{F}_\Theta$ against the one-time unforgeability of the signature scheme Θ. This will contradict the assumption that both Γ and Θ are secure, and so will conclude the proof.

The adversary $\mathcal{A}_\Gamma$ against the identity-based encryption scheme Γ is initialized with input ibp, where $\mathsf{ibp} \leftarrow \Gamma.\mathsf{Stp}(1^k)$ has been executed by the challenger. After that, $\mathcal{A}_\Gamma$ runs $\mathsf{pms}_\Theta \leftarrow \Theta.\mathsf{Stp}(1^\lambda)$ once and then runs $n \cdot q_e$ times the key generation protocol $\Theta.\mathsf{KG}(1^k)$, obtaining $n \cdot q_e$ key pairs $\{(sk_\Theta^{(i,j)}, vk_\Theta^{(i,j)})\}_{1 \le i \le n, 1 \le j \le q_e}$. Now $\mathcal{A}_\Gamma$ defines $id_i^j = vk_\Theta^{(i,j)}$, for $i = 1, \ldots, n$ and $j = 1, \ldots, q_e$, to form and output the set $\mathcal{I}^\star$ of $n \cdot q_e$ identities. Now the challenger sends to $\mathcal{A}_\Gamma$ a list of n master public keys $(PK_1, \ldots, PK_n)$.

$\mathcal{A}_\Gamma$ can then define $\mathsf{pms} = (\mathsf{ibp}, \mathsf{pms}_\Theta)$ and initialize the adversary $\mathcal{A}_\Pi$ against the encryption scheme Π by giving pms and $(pk_1, \ldots, pk_n)$ to him, where $pk_i = PK_i$ for $i = 1, \ldots, n$. The adversary $\mathcal{A}_\Pi$ is allowed to make encryption and decryption queries, that $\mathcal{A}_\Gamma$ answers in the following way. The list L_{quer} is initially set to empty.

(a) **Encryption queries.** For each index $i \in \{1, \ldots, n\}$, $\mathcal{A}_\Pi$ can make up to q_e encryption queries of the form (i, f), with $f \in \mathcal{F}$. Let us consider the j-th query of this form, where $j \in \{1, \ldots, q_e\}$. Then $\mathcal{A}_\Gamma$ makes an encryption query (i, f) to his encryption oracle, and receives as answer a ciphertext c, computed for master public key $PK_i = pk_i$ and identity id_i^j. After that, $\mathcal{A}_\Gamma$ runs $\theta \leftarrow \Theta.\mathsf{Sign}(sk_\Theta^{(i,j)}, c)$, defines $C = (vk_\Theta^{(i,j)}, c, \theta)$ and sends C to $\mathcal{A}_\Pi$. The tuple (i, C) is added to the list L_{quer}.

(b) **Decryption queries.** $\mathcal{A}_\Pi$ can make decryption queries of the form (i, C), as long as $(i, C) \notin L_{\mathsf{quer}}$. Let us parse $C = (vk_\Theta, c, \theta)$. First of all, $\mathcal{A}_\Gamma$ runs $\Theta.\mathsf{Vfy}(vk_\Theta, c, \theta)$. If the output is 0, then $\mathcal{A}_\Gamma$ sends back $\perp$ to $\mathcal{A}_\Pi$. Otherwise, we distinguish two cases:

   (i) $vk_\Theta \neq vk_\Theta^{(i,j)}$ for all $j \in \{1, \ldots, q_e\}$. In this case, $\mathcal{A}_\Gamma$ can define $id = vk_\Theta$ and make the private key query (i, id), receiving as answer the value $sk_i[id] \leftarrow \Gamma.\mathsf{Ukg}_{\mathsf{ibp}}(SK_i, id)$. Finally, $\mathcal{A}_\Gamma$ sends to $\mathcal{A}_\Pi$ the result of running $\Gamma.\mathsf{Dec}_{\mathsf{ibp}}(sk_i[id], c)$.

(ii) $vk_{\Theta} = vk_{\Theta}^{(i,j)}$ for some $j \in \{1, \ldots, q_e\}$. In this case, $\mathcal{A}_{\Gamma}$ aborts. Let us denote as δ the probability that $\mathcal{A}_{\Gamma}$ aborts due to one such decryption query.

At the end of the experiment, adversary $\mathcal{A}_{\Pi}$ outputs a bit b'. The adversary $\mathcal{A}_{\Gamma}$ that we are constructing outputs the same bit b'.

It is easy to see that, provided $\mathcal{A}_{\Gamma}$ does not abort during the experiment, the advantage of $\mathcal{A}_{\Gamma}$ in the experiment $\mathbf{ExpKDM\text{-}sID\text{-}CPA}^{b,\Gamma}_{\mathcal{A}_{\Gamma}}(\lambda, n, q_e)$ is exactly the same as the advantage of $\mathcal{A}_{\Pi}$ in the experiment $\mathbf{ExpKDM\text{-}CCA}^{b,\Pi}_{\mathcal{A}_{\Pi}}(\lambda, n, q_e)$. Therefore, the only difference between these two advantages is given by the probability δ that $\mathcal{A}_{\Gamma}$ aborts.

However, it is possible to construct a forger $\mathcal{F}_{\Theta}$ against the one-time unforgeability of the signature scheme Θ, in a scenario with $N = n \cdot q_e$ users, whose success probability would be $\geq \delta$. The idea is to run $\mathcal{A}_{\Gamma}$ by considering as the identities id_i^j the verification keys $vk_{\Theta}^{(i,j)}$ that $\mathcal{F}_{\Theta}$ receives as input in his unforgeability game (in the multi-user setting). The n key pairs of Γ are generated by $\mathcal{F}_{\Theta}$ and therefore it can compute any function $f \in \mathcal{F}$ of the master secret keys. The step where $\mathcal{A}_{\Gamma}$ needs the corresponding secret keys $sk_{\Theta}^{(i,j)}$ to answer encryption queries is replaced with a (one-time) query by $\mathcal{F}_{\Theta}$ to his signing oracle for secret key $sk_{\Theta}^{(i,j)}$. If $\mathcal{A}_{\Gamma}$ receives a valid decryption query (i, C) where $C = (vk_{\Theta}, c, \theta)$ with $vk_{\Theta} = vk_{\Theta}^{(i,j)}$ for some $i \in \{1, \ldots, n\}$ and some $j \in \{1, \ldots, q_e\}$, and such that $(i, C) \notin L_{\mathsf{quer}}$, this means that θ is a valid and fresh signature, under verification key $vk_{\Theta}^{(i,j)}$, for a message c. In other words, $\mathcal{F}_{\Theta}$ succeeds with probability δ.

Putting all the pieces together, we conclude that

$$\mathbf{AdvKDM\text{-}CCA}^{\Pi}_{\mathcal{A}_{\Pi}}(\lambda, n, q_e) \leq \mathbf{AdvKDM\text{-}sID\text{-}CPA}^{\Gamma}_{\mathcal{A}_{\Gamma}}(\lambda, n, q_e) + \mathbf{AdvOTS}^{\Theta}_{\mathcal{F}_{\Theta}}(\lambda, n \cdot q_e) \quad \square$$

4 New Bounded mKDM-sID-CPA Secure IBE Schemes

In this section we propose some identity-based encryption scheme enjoying mKDM-sID-CPA security. We start with a basic scheme Γ_0 that achieves (n, q_e)-mKDM-sID-CPA security. It is inspired by some IBE techniques [6] and the KDM-CPA techniques in [9]. After that, we explain the modifications that can be applied to Γ_0 to improve its efficiency (for example, the size of the master public keys). The final resulting scheme, Γ_2, can be used to implement the transformation of Section 3.2, in combination with any one-time signature scheme, leading to new KDM-CCA secure public-key encryption schemes. In all the schemes in this section, the size of the master public keys depends linearly on the number q_e of allowed encryption queries; that is, these schemes achieve mKDM-sID-CPA security only against *bounded* adversaries.

4.1 The Basic mKDM-sID-CPA Secure Scheme, Γ_0

Let us introduce a glossary for notation valid hereafter. The letter $\mathbf{S}$ denotes a matrix in ${\mathbb{Z}_q}^{\ell \times \ell}$ and the letter $\mathbf{\Psi}$ denotes a matrix in ${\mathbb{Z}_q}^{(\ell+1)\times \ell}$; $\boldsymbol{t}$ denotes a column vector in $\mathbb{Z}_q^{\ell}$; $\boldsymbol{c}, \boldsymbol{d}$ respectively denote row and column vectors in $\mathcal{G}^{\ell}$; finally $\boldsymbol{r}$ denotes a row vector in $\mathbb{Z}_q^{\ell+1}$. For a specific value of q_e the scheme $\Gamma_0(q_e)$ works as follows.

Setup, $\Gamma_0.\mathsf{Stp}(1^{\lambda})$**:** a pairing group $(\mathcal{G}, \mathcal{G}_T, e(\cdot,\cdot))$ of prime order q, where q is λ-bits long, and a generators $g \in \mathcal{G}, g_T = e(g,g) \in \mathcal{G}_T$ are chosen. A second security parameter $\ell > 4\lambda$ is also considered. Therefore, we define $\mathsf{ibp} = (\lambda, \ell, q, \mathcal{G}, g, \mathcal{G}_T, g_T, e(\cdot,\cdot))$.

Master key generation, $\Gamma_0.\mathsf{Mkg}_{\mathsf{ibp}}()$**:** firstly, take $\mathbf{\Psi} \in_{\mathrm{R}} {\mathbb{Z}_q}^{(\ell+1)\times \ell;\ell}$, $\widetilde{\mathbf{S}} \in_{\mathrm{R}} \mathrm{GL}_{\ell}(\mathbb{Z}_q)$ and a binary (column) vector $\boldsymbol{x} \in_{\mathrm{R}} \{0,1\}^{\ell}$, and compute $g_T^{-\mathbf{\Psi}\boldsymbol{x}} \in \mathcal{G}_T^{\ell}$. Then two random functions are built in the following way. Take random matrices $\mathbf{S}_0, \ldots, \mathbf{S}_{q_e-1} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell \times \ell}$ and $\mathbf{S}_{q_e} \in_{\mathrm{R}} \mathrm{GL}_{\ell}(\mathbb{Z}_q)$, which define the function $\mathbf{F}(\cdot)$ acting on the identity space $\mathbb{Z}_q$ as $\mathbf{F}(id) = \sum_{j=0}^{q_e} id^j \mathbf{S}_j \in \mathbb{Z}_q^{\ell \times \ell}$. Then define $\widetilde{\mathbf{F}}(id) = \mathbf{\Psi}\mathbf{F}(id)\widetilde{\mathbf{S}}^{-1} = \sum_{j=0}^{q_e} id^j \widetilde{\mathbf{\Psi}}_j \in \mathbb{Z}_q^{(\ell+1)\times \ell}$, where $\widetilde{\mathbf{\Psi}}_j = \mathbf{\Psi}\mathbf{S}_j\widetilde{\mathbf{S}}^{-1}$, for $j = 0, \ldots, q_e$. Therefore $\widetilde{\mathbf{F}}(id)\widetilde{\mathbf{S}} = \mathbf{\Psi}\mathbf{F}(id)$ for any $id \in \mathbb{Z}_q$. The public and master

secret keys are then $PK = (g^{\boldsymbol{\Psi}}, g^{\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\boldsymbol{\Psi}\boldsymbol{x}})$ and $SK = g_T^{\boldsymbol{x}}$, where the description of the functions $g^{\mathbf{F}(\cdot)}$ and $g^{\widetilde{\mathbf{F}}(\cdot)}$ are their (matrix) coefficients $g^{\mathbf{S}_0}, \ldots, g^{\mathbf{S}_{q_e}}$ and $g^{\widetilde{\boldsymbol{\Psi}}_0}, \ldots, g^{\widetilde{\boldsymbol{\Psi}}_{q_e}}$, respectively.

User key generation, $\Gamma_0.\mathsf{Ukg}_{\mathsf{ibp}}(SK, id)$**:** for an identity $id \in \mathbb{Z}_q$ the secret key $sk[id] = (g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}) \in \mathcal{G}^\ell \times \mathcal{G}^\ell$ is generated as $g^{\boldsymbol{d}_1} = g^{\boldsymbol{x}} \cdot g^{\mathbf{F}(id)\boldsymbol{t}}$ and $g^{\boldsymbol{d}_2} = g^{\widetilde{\mathbf{S}}\boldsymbol{t}}$, where $\boldsymbol{t} \in_{\mathrm{R}} \mathbb{Z}_q^\ell$ and $g^{\boldsymbol{x}}$ is computed component by component from $SK = g_T^{\boldsymbol{x}}$. The user can verify the validity of the secret key by checking the equation $g_T^{-\boldsymbol{\Psi}\boldsymbol{x}} \cdot e(g^{\boldsymbol{\Psi}}, g^{\boldsymbol{d}_1}) = e(g^{\mathbf{t}\mathbf{F}(id)}, g^{\boldsymbol{d}_2})$.

Encryption, $\Gamma_0.\mathsf{Enc}_{\mathsf{ibp}}(PK, id, m)$**:** to encrypt a message $m \in \mathcal{G}_T$ for an identity $id \in \mathbb{Z}_q$ and master public key PK, a row vector $\boldsymbol{r} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell+1}$ is chosen and the ciphertext $C = (g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c) \in \mathcal{G}^\ell \times \mathcal{G}^\ell \times \mathcal{G}_T$ is computed as $g^{\boldsymbol{c}_1} = g^{\boldsymbol{r}\boldsymbol{\Psi}}$, $g^{\boldsymbol{c}_2} = g^{\boldsymbol{r}\mathbf{t}\mathbf{F}(id)}$ and $c = m \cdot g_T^{-\boldsymbol{r}\boldsymbol{\Psi}\boldsymbol{x}}$. The ciphertext fulfils the equation $e(g^{\boldsymbol{c}_1}, g^{\mathbf{F}(id)}) = e(g^{\boldsymbol{c}_2}, g^{\widetilde{\mathbf{S}}})$, so its consistency with respect to identity id can be publicly verified.

Decryption, $\Gamma_0.\mathsf{Dec}_{\mathsf{ibp}}(sk[id], C)$**:** let $C = (g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ be a ciphertext for an identity id. The user who owns $sk[id] = (g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$ recovers $m = c \cdot e(g^{\boldsymbol{c}_1}, g^{\boldsymbol{d}_1})/e(g^{\boldsymbol{c}_2}, g^{\boldsymbol{d}_2})$.

Some Intuition. Notice that a direct modification of the Boneh-Boyen IBE scheme [6] following the structure of the Boneh et al. KDM-CPA scheme [9] leads to an insecure scheme, due to the fact that the master secret key $\boldsymbol{x} \in \{0,1\}^\ell$ can be trivially recovered from its representation $g_T^{\boldsymbol{x}}$. In fact, in the direct translation the user-keys would have been $sk[id] = (g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$ with $g^{\boldsymbol{d}_1} = g^{\boldsymbol{x}} g^{\mathbf{F}(id)\boldsymbol{t}}$ and $g^{\boldsymbol{d_2}} = g^{\boldsymbol{t}}$, where $\boldsymbol{t} \in_{\mathrm{R}} \mathbb{Z}_q^\ell$ (ciphertexts would be changed accordingly). In such a case, an adversary that obtains a single user-key $sk[id]$ can compute $e(g^{\boldsymbol{d}_1}, g) = g_T^{\boldsymbol{x}} \cdot e(g^{\mathbf{F}(id)}, g^{\boldsymbol{t}})$ on the one hand, and $e(g^{\mathbf{F}(id)}, g^{\boldsymbol{d}_2}) = e(g^{\mathbf{F}(id)}, g^{\boldsymbol{t}})$ on the other hand. The adversary thus recovers $g_T^{\boldsymbol{x}}$, which leads to the recovery of master secret key, since $\boldsymbol{x} \in \{0,1\}^\ell$. For this reason we are forced to "hide" $\boldsymbol{t}$ even more, by multiplying it with the matrix $\widetilde{\mathbf{S}} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$. This makes scheme description and security proofs more intricate, for example because some care must be taken regarding the invertibility and the probability distribution of such matrices $\widetilde{\mathbf{S}} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$, when master public keys are rerandomized.

The definition of the function $\mathbf{F}(id)$ as a polynomial with degree q_e is necessary in order to be able to reply q_e encryption queries. Finally, the fact that the coefficients of this polynomial are $\ell \times \ell$ matrices will be improved to $\ell \times 2$ matrices in the improved scheme Γ_2, in Section 4.3. The dimension ℓ comes from the ideas in the KDM-CPA scheme in [9]. The dimension 2 comes from the fact that we are using (symmetric) pairing groups $\mathcal{G}$ and the security of our scheme relies on the hardness of the Rank problem in $\mathcal{G}$, which is easy if the rank is 1.

Affine functions. Let us define the set of affine functions $\mathcal{F} = \{f : \mathcal{T}^n \to \mathcal{G}_T\}$, where $\mathcal{T}$ is the set of master secret keys. Let $SK_1, \ldots, SK_n \in \mathcal{G}_T^\ell$ be n secret keys generated by $\Gamma_0.\mathsf{Mkg}_{\mathsf{ibp}}()$. Following the notation in [9], for every $n\ell$-vector $\boldsymbol{u} = (u_i)$ over Z_q, every $n\ell$-vector $\boldsymbol{s} \in \mathcal{G}_T^{n\ell}$ and every scalar $H \in \mathcal{G}_T$, let $f_{\boldsymbol{u},H}(\boldsymbol{s}) = H \cdot \prod_{i=1,\ldots,n\ell} s_i^{u_i} \in \mathcal{G}_T$. Then, $\mathcal{F} = \{f_{\boldsymbol{u},H} : \mathcal{G}_T^{n\ell} \to \mathcal{G}\}_{\boldsymbol{u} \in \mathbb{Z}_q^{n\ell}, H \in \mathcal{G}_T}$.

Additionally, since the algorithm $\Gamma_0.\mathsf{Ukg}_{\mathsf{ibp}}(SK, id)$ can be seen as an affine function from $\mathcal{G}^\ell$ to $\mathcal{G}^{2\ell}$, we obtain uKDM-sID-CPA security [2] with respect to the set of affine functions from $\mathcal{G}^{2n\ell}$ to $\mathcal{G}_T$. Alas, this is is only a restricted form of uKDM-sID-CPA security, since in particular we can not encrypt the j-th selection function $(sk[id_1], \ldots, sk[id_n]) \mapsto sk[id_j]$, as $sk[id_j] \in \mathcal{G}^{2\ell}$.

Additional Properties of Γ_0. As in [9] some extra properties of the scheme will be used in the security proof. Here we will consider the notion of *half-ciphertext* as a ciphertext $(g^{\boldsymbol{c}_1}, \emptyset, c)$ that is not ID-enabled but still can be decrypted directly from the master key.

Master Decryption. A half-ciphertext $(g^{\boldsymbol{c}_1}, \emptyset, c)$ can be directly decrypted from $msk = g_T^{\boldsymbol{x}}$ by running $m = \mathbf{master\text{-}dec}(g^{\boldsymbol{c}_1}, \emptyset, c)\, msk, \boldsymbol{c}, \widetilde{\boldsymbol{c}}, e) = c \cdot e(g^{\boldsymbol{c}_1}, g^{\boldsymbol{x}})$, where $\boldsymbol{x}$ is extracted component by component from the master secret key $g_T^{\boldsymbol{x}}$.

MASTER PUBLIC KEY RERANDOMIZATION. The master public key $PK = (g^{\mathbf{\Psi}}, g^{\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\mathbf{Psi}\boldsymbol{x}})$ can be rerandomized into another random master public key $PK' = (g^{\mathbf{\Psi}'}, g^{\mathbf{F}'(\cdot)}, g^{\widetilde{\mathbf{F}}'(\cdot)}, g^{\widetilde{\mathbf{S}}'}, (g_T^{-\mathbf{\Psi}\boldsymbol{x}})')$ for the same master secret key SK, with the same probability distribution as the output of the key generation algorithm conditioned to SK. Indeed one can define $g^{\widetilde{\mathbf{S}}'} = g^{\widetilde{\mathbf{S}}\mathbf{R}}$, $g^{\mathbf{\Psi}'} = g^{\mathbf{L}\mathbf{\Psi}}$, $g^{\mathbf{F}'(\cdot)} = g^{\mathbf{F}(\cdot)\mathbf{R}} g^{\mathbf{Q}(\cdot)\,\widetilde{S}\,\mathbf{R}}$, $g^{\widetilde{\mathbf{F}}'(\cdot)} = g^{\mathbf{L}\,\widetilde{F}(\cdot)} g^{\mathbf{\Psi}\mathbf{Q}(\cdot)}$ and $(g_T^{-\mathbf{\Psi}\boldsymbol{x}})' = g_T^{-\mathbf{L}\mathbf{\Psi}\boldsymbol{x}}$, for a random (matrix) polynomial $\mathbf{Q}(\cdot) = \sum_{j=0}^{q_e} Q_j id^j$, where $\mathbf{Q}_0, \ldots, \mathbf{Q}_{q_e} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell\times\ell}$, and random matrices $\mathbf{L} \in_{\mathrm{R}} \mathrm{GL}_{\ell+1}(\mathbb{Z}_q)$ and $\mathbf{R} \in_{\mathrm{R}} \mathrm{GL}_{\ell}(\mathbb{Z}_q)$. Notice that $\mathbf{S}'_j = (\mathbf{S}_j + \mathbf{Q}_j\widetilde{\mathbf{S}})\mathbf{R}$ and $\widetilde{\mathbf{\Psi}'}_j = \mathbf{L}(\widetilde{\mathbf{\Psi}}_j + \mathbf{\Psi}\mathbf{Q}_j)$, for $j = 0, \ldots, q_e$. With overwhelming probability $1 - 1/(q-1)$, the resulting $\mathbf{S}'_{q_e}$ is an invertible matrix[4], and PK' is a valid public key. We will write it as $PK' = \mathbf{PK\text{-}rand}(PK; \mathbf{L}, \mathbf{R}, \mathbf{Q}(\cdot))$. Any half-ciphertext valid for PK is also valid for PK', and both decrypt to the same plaintext. Furthermore, any ciphertext $(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ valid for a message m, identity id and master public key PK can be converted into another one, $\mathbf{PK\text{-}rand\text{-}ciph}((g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c); \mathbf{L}, \mathbf{R}, \mathbf{Q}(\cdot)) = (g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2} g^{\boldsymbol{c}_1\mathbf{Q}(id)}, c)$, valid for m, id and PK'. Similarly, any user secret key $sk[id] = (g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$ generated with PK can be transformed into $\mathbf{PK\text{-}rand\text{-}user}(g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}; \mathbf{L}, \mathbf{R}, \mathbf{Q}(\cdot)) = (g^{\boldsymbol{d}_1} g^{\mathbf{Q}(id)\boldsymbol{d}_2}, g^{\boldsymbol{d}_2})$ which is valid for PK'.

CIPHERTEXT RERANDOMIZATION. A ciphertext $(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ for an identity id and master public key PK can be easily rerandomized into another ciphertext $(g^{\boldsymbol{c}'_1}, g^{\boldsymbol{c}'_2}, c') = \mathbf{enc\text{-}rand}(PK, id, g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c; \boldsymbol{r})$ of the same message, identity and public key as $g^{\boldsymbol{c}'_1} = g^{\boldsymbol{r}\mathbf{\Psi}} g^{\boldsymbol{c}_1}$, $g^{\boldsymbol{c}'_2} = g^{\boldsymbol{r}\mathbf{t}\mathbf{F}(id)} g^{\boldsymbol{c}_2}$ and $c' = g_T^{-\boldsymbol{r}\mathbf{\Psi}\boldsymbol{x}} c$, for a random $\boldsymbol{r} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell+1}$.

SECRET KEY ENCRYPTION. The i-th bit x_i of the master secret key can be encrypted (without knowing it) for an identity id by rerandomizing the self-referring half-ciphertext $\mathbf{self\text{-}ref}(i) = (g^{\boldsymbol{c}_1}, \emptyset, 1_{\mathcal{G}_T})$ such that $g^{\boldsymbol{c}_1} = (1_{\mathcal{G}}, \ldots, g, \ldots, 1_{\mathcal{G}})$, where the only element different from $1_{\mathcal{G}} = g^0$ is in the i-th position of $g^{\boldsymbol{c}_1}$.

CIPHERTEXT HOMOMORPHISM. Since, for a fixed identit,y decryption (both user and master) is a linear map between $\mathbb{Z}_q$-vector spaces, ciphertexts can be combined to obtain a new ciphertext of a linear combination of the corresponding plaintexts. Indeed, given k encryptions $C_i = (g^{\boldsymbol{c}_{1,i}}, g^{\boldsymbol{c}_{2,i}}, c_i)$ of plaintexts $m_i \in \mathcal{G}_T$, $i = 1, \ldots, k$, under the same master public key and identity, the ciphertext

$$C = \left(\prod_{i=1}^{k} (g^{\boldsymbol{c}_{1,i}})^{\lambda_i} \;,\; \prod_{i=1}^{k} (g^{\boldsymbol{c}_{2,i}})^{\lambda_i} \;,\; \prod_{i=1}^{k} c_i^{\lambda_i} \right)$$

is an encryption of $m = \prod_{i=1}^{k} m_i^{\lambda_i}$ under the same master key and identity, for any scalars $\lambda_i \in \mathbb{Z}_q$. The same can be applied to half-ciphertexts.

REENCRYPTION. Assume two different master secret keys $SK = g_T^{\boldsymbol{x}}$ and $SK' = g_T^{\boldsymbol{x}'}$ are related by a known bijective affine transformation, given by an invertible matrix $\mathbf{M} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$ and a column shift vector $\boldsymbol{\mu} \in \mathbb{Z}_q^\ell$ by $\boldsymbol{x}' = \mathbf{M}\boldsymbol{x} + \boldsymbol{\mu}$. Then a ciphertext $(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ for an identity id and master public key $PK = (g^{\mathbf{\Psi}}, g^{\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\mathbf{\Psi}\boldsymbol{x}})$ for SK SK can be reencrypted into another ciphertext $\mathbf{reencr\text{-}ciph}(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c; \mathbf{M}, \boldsymbol{\mu}) = (g^{\boldsymbol{c}_1\mathbf{M}^{-1}}, g^{\boldsymbol{c}_2}, c \cdot e(g^{\boldsymbol{c}_1}, g^{-\mathbf{M}^{-1}\boldsymbol{\mu}}))$ for the same identity and plaintext, and for a new master public key $PK' = \mathbf{reencr\text{-}PK}(PK; \mathbf{M}, \boldsymbol{\mu}) = (g^{\mathbf{\Psi}\mathbf{M}^{-1}}, g^{\mathbf{M}\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\mathbf{\Psi}\boldsymbol{x}} \cdot e(g^{\mathbf{\Psi}}, g^{-\mathbf{M}^{-1}\boldsymbol{\mu}}))$ corresponding to SK'. The same applies to half-ciphertexts. Moreover, any extracted user key $(g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$ for SK, PK and identity id can be converted into another valid user key $\mathbf{reencr\text{-}user}((g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}); \mathbf{M}, \boldsymbol{\mu}) = (g^{\mathbf{M}\boldsymbol{d}_1 + \boldsymbol{\mu}}, g^{\boldsymbol{d}_2})$ for SK', PK' and the same identity.

Observe that for every binary vector $\boldsymbol{a} \in \{0,1\}^\ell$ there exists a bijective affine transformation that maps any binary vector $\boldsymbol{x}$ into the componentwise XOR, $\boldsymbol{a} \oplus \boldsymbol{x}$, and it is given by a diagonal matrix $\mathbf{M}$ with diagonal $1 - 2\boldsymbol{a}$ and a shift vector $\boldsymbol{\mu} = \boldsymbol{a}$. Since XOR-ing is an involution, the direct and inverse transformations are equal. Hence $\mathbf{M}^{-1} = \mathbf{M}$ and $-\mathbf{M}^{-1}\boldsymbol{\mu} = \boldsymbol{\mu}$, and we can write $\mathbf{reencrXOR\text{-}ciph}(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c; \mathbf{M}, \boldsymbol{\mu}) = (g^{\boldsymbol{c}_1\mathbf{M}}, g^{\boldsymbol{c}_2}, c \cdot e(g^{\boldsymbol{c}_1}, g^{\boldsymbol{\mu}}))$ and $\mathbf{reencrXOR\text{-}PK}(PK; \mathbf{M}, \boldsymbol{\mu}) = (g^{\mathbf{\Psi}\mathbf{M}}, g^{\mathbf{M}\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\mathbf{\Psi}\boldsymbol{x}} \cdot e(g^{\mathbf{\Psi}}, g^{\boldsymbol{\mu}}))$[5]. It is

[4] Indeed, for fixed invertible matrices $\mathbf{R}$, $\mathbf{S}_{q_e}$ and $\widetilde{\mathbf{S}}$, the map $\mathbf{Q}_{q_e} \mapsto (\mathbf{S}_{q_e} + \mathbf{Q}_{q_e}\widetilde{\mathbf{S}})\mathbf{R}$ is a bijection. Therefore $\mathbf{S}'_{q_e}$ is a uniformly random square matrix, and it is invertible with probability $q^{-\ell^2}\,|\mathrm{GL}_\ell(\mathbb{Z}_q)| > 1 - 1/(q-1)$.

[5] **reencrXOR-user** is defined exactly as **reencr-user**.

easy to see that the composition of two affine transformations $(\mathbf{M}_1, \boldsymbol{\mu}_1)$, $(\mathbf{M}_2, \boldsymbol{\mu}_2)$ corresponding to the XOR masks is the affine transformation $(\mathbf{M}_1\mathbf{M}_2, \boldsymbol{\mu}_1 \oplus \boldsymbol{\mu}_2)$.

ID-Enabling of Half-Ciphertexts. Any half-ciphertext $(g^{\boldsymbol{c}_1}, \emptyset, c)$ can be completed into a ciphertext for the same master public key, message and identity id, if some extra information $\mathbf{W}$ is given, through $\textbf{enable-half}(PK, id, \mathbf{W}, g^{\boldsymbol{c}_1}, \emptyset, c) = (g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$. Indeed, if the matrix $\mathbf{W} \in {\mathbb{Z}_q}^{\ell\times\ell}$ such that $\mathbf{F}(id) = \mathbf{W}\widetilde{\mathbf{S}}$ is given, then $g^{\boldsymbol{c}_2} = g^{\boldsymbol{c}_1\mathbf{W}}$.

ID-Transform. Given a polynomial transformation of the identities, $P(id) = \sum_{i=0}^{k} p_i id^i$ where $p_0, \ldots, p_k \in \mathbb{Z}_q$ and $p_k \neq 0$, a master public key PK for the scheme $\Gamma_0(q_e)$ can be transformed into another one PK' for the scaled scheme $\Gamma_0(kq_e)$ so that the secret key remains unchanged, and any ciphertext or user key valid for the identity $P(id)$ under PK in $\Gamma_0(q_e)$ is still valid for the identity id under PK' in $\Gamma_0(kq_e)$. Indeed, $PK' = \textbf{id-transf}(PK; P) = (g^{\boldsymbol{\Psi}}, g^{(\mathbf{F}\circ P)(\cdot)}, g^{(\widetilde{\mathbf{F}}\circ P)(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\boldsymbol{\Psi}\boldsymbol{x}})$, where the (matrix) coefficients of the compositions $g^{(\mathbf{F}\circ P)(\cdot)}$ and $g^{(\widetilde{\mathbf{F}}\circ P)(\cdot)}$ can be easily computed from the coefficients $g^{\mathbf{S}_j}$, $g^{\widetilde{\boldsymbol{\Psi}}_j}$ and p_i. Notice that the resulting public key does not follow the right probability distribution and it must be rerandomized with **PK-rand**.

4.2 (n, q_e)-mKDM-sID-CPA Security of Γ_0

In this section we prove that our basic scheme, $\Gamma_0(q_e)$, is mKDM-sID-CPA secure with respect to the set of affine functions $\mathcal{F}$. The proof consists of two steps: firstly we prove that $\Gamma_0(1)$ achieves mKDM-ssID-CPA security under the Decisional Linear assumption, and then we prove that nq_e-mKDM-ssID-CPA security of $\Gamma_0(1)$ implies (n, q_e)-mKDM-sID-CPA security in the case of $\Gamma_0(q_e)$. We stress here that in all the security reductions of the paper, the involved execution times satisfy the tight relations $t' \leq t + \mathcal{O}(n \cdot q_e \cdot \ell^3)$ and $t'' \leq t + \mathcal{O}(n \cdot q_e \cdot \ell^3)$, taking the cost of a scalar multiplication in $\mathcal{G}$ as one time unit. We omit these relations and the technical details from here on, for simplicity, and just write t, t', t''.

Theorem 2. *$\Gamma_0(1)$ is q_e-mKDM-ssID-CPA secure under the assumption that the Decisional Linear problem is hard. In particular,* $\textbf{AdvKDM-ssID-CPA}(\Gamma_0(1), \lambda, \ell, q_e; t) \leq 4\,(\lceil 1.71 \log_2 \ell \rceil + 1)\,\textbf{AdvDLin}(\mathcal{G}; t') + 8 \cdot 2^{-\lambda}$.

We use in the proof of Theorem 2 the following technical result about the indistinguishability of different distributions of matrices, all of them related to the set of solutions of the matrix equation $AY = XB$ for fixed A and B.

Lemma 7. *Given a pairing group $(\mathcal{G}, \mathcal{G}_T, e(\cdot,\cdot))$ of prime order q, where q is λ-bits long, a generator g of $\mathcal{G}$ and the parameter ℓ, and assuming the Rank problem is hard, then the following probability distributions of the variables $(g^{\boldsymbol{\Psi}}, g^{\mathbf{S}}, g^{\widetilde{\boldsymbol{\Psi}}}, g^{\widetilde{\mathbf{S}}}, g^{\boldsymbol{z}})$, where $(\boldsymbol{\Psi}, \mathbf{S}, \widetilde{\boldsymbol{\Psi}}, \widetilde{\mathbf{S}}, \boldsymbol{z}) \in {\mathbb{Z}_q}^{(\ell+1)\times\ell} \times {\mathbb{Z}_q}^{\ell\times\ell} \times {\mathbb{Z}_q}^{(\ell+1)\times\ell} \times {\mathbb{Z}_q}^{\ell\times\ell} \times \mathbb{Z}_q^{\ell}$ given as uniform distributions under the restrictions given below are polynomially indistinguishable.*

D_0: $\boldsymbol{\Psi}\mathbf{S} = \widetilde{\boldsymbol{\Psi}}\widetilde{\mathbf{S}}$; $\mathit{rank}(\boldsymbol{\Psi}) = \mathit{rank}(\widetilde{\boldsymbol{\Psi}}) = \mathit{rank}(\mathbf{S}) = \mathit{rank}(\widetilde{\mathbf{S}}) = \ell$; $\exists \boldsymbol{x} \in \{0,1\}^{\ell}$, $\boldsymbol{z} = \widetilde{\mathbf{S}}\mathbf{S}^{-1}\boldsymbol{x}$.
D_5: $\boldsymbol{\Psi}\mathbf{S} = \widetilde{\boldsymbol{\Psi}}\widetilde{\mathbf{S}}$; $\mathit{rank}(\boldsymbol{\Psi}) = \mathit{rank}(\widetilde{\boldsymbol{\Psi}}) = \ell$; $\mathit{rank}(\mathbf{S}) = \mathit{rank}(\widetilde{\mathbf{S}}) = \ell - 1$; $\mathit{Span}(\boldsymbol{\Psi}) \neq \mathit{Span}(\widetilde{\boldsymbol{\Psi}})$.

Proof. Consider the following intermediate probability distributions of the variables $(g^{\boldsymbol{\Psi}}, g^{\mathbf{S}}, g^{\widetilde{\boldsymbol{\Psi}}}, g^{\widetilde{\mathbf{S}}}, g^{\boldsymbol{z}})$ given as uniform distributions under the following restrictions.

D_1: $\boldsymbol{\Psi}\mathbf{S} = \widetilde{\boldsymbol{\Psi}}\widetilde{\mathbf{S}}$; $\mathrm{rank}(\boldsymbol{\Psi}) = \mathrm{rank}(\widetilde{\mathbf{S}}) = 2$; $\mathrm{rank}(\mathbf{S}) = \mathrm{rank}(\widetilde{\boldsymbol{\Psi}}) = \ell$; $\exists \boldsymbol{x} \in \{0,1\}^{\ell}$, $\boldsymbol{z} = \widetilde{\mathbf{S}}\mathbf{S}^{-1}\boldsymbol{x}$.
D_2: $\boldsymbol{\Psi}\mathbf{S} = \widetilde{\boldsymbol{\Psi}}\widetilde{\mathbf{S}}$; $\mathrm{rank}(\boldsymbol{\Psi}) = \mathrm{rank}(\widetilde{\mathbf{S}}) = 2$; $\mathrm{rank}(\mathbf{S}) = \mathrm{rank}(\widetilde{\boldsymbol{\Psi}}) = \ell$.
D_3: $\boldsymbol{\Psi}\mathbf{S} = \widetilde{\boldsymbol{\Psi}}\widetilde{\mathbf{S}}$; $\mathrm{rank}(\boldsymbol{\Psi}) = \mathrm{rank}(\widetilde{\boldsymbol{\Psi}}) = \mathrm{rank}(\mathbf{S}) = \mathrm{rank}(\widetilde{\mathbf{S}}) = \ell$.
D_4: $\boldsymbol{\Psi}\mathbf{S} = \widetilde{\boldsymbol{\Psi}}\widetilde{\mathbf{S}}$; $\mathrm{rank}(\boldsymbol{\Psi}) = \mathrm{rank}(\widetilde{\boldsymbol{\Psi}}) = \ell$; $\mathrm{rank}(\mathbf{S}) = \mathrm{rank}(\widetilde{\mathbf{S}}) = \ell - 1$; $\mathrm{Span}(\Psi) = \mathrm{Span}(\widetilde{\Psi})$.

We will show that every probability distribution $D_0, \ldots, D_5$ is indistinguishable from the previous one. For any PPT algorithm $\mathcal{A}$ let us call Ω_i to the event that $\mathcal{A}(X) = 1$ where X is sampled from D_i, and let $\mathbf{Adv}_{D_i,D_j} = |\Pr[\Omega_i] - \Pr[\Omega_j]|$.

Distributions D_0 to D_3 share the property $\mathbf{S} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$. Therefore, there exists a unique $\mathbf{T} \in {\mathbb{Z}_q}^{\ell\times\ell}$ such that $\widetilde{\mathbf{S}} = \mathbf{TS}$ and $\mathbf{\Psi} = \widetilde{\mathbf{\Psi}}\mathbf{T}$. Moreover, the same happens with D_4, but for a different reason. Indeed, as $\mathbf{\Psi}$ and $\widetilde{\mathbf{\Psi}}$ are full-rank matrices such that $\mathrm{Span}(\mathbf{\Psi}) = \mathrm{Span}(\widetilde{\mathbf{\Psi}})$, then by Lemma 5 there also exists a unique $\mathbf{T} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$ such that $\mathbf{\Psi} = \widetilde{\mathbf{\Psi}}\mathbf{T}$. Therefore, $\widetilde{\mathbf{\Psi}}\widetilde{\mathbf{S}} = \mathbf{\Psi}\mathbf{S} = \widetilde{\mathbf{\Psi}}\mathbf{TS}$, which implies $\widetilde{\mathbf{S}} = \mathbf{TS}$ because left-multiplication by $\widetilde{\mathbf{\Psi}}$ is an injective map.

However, $\mathrm{rank}(\mathbf{T}) = \ell$ in distributions D_0 and D_3, while $\mathrm{rank}(\mathbf{T}) = 2$ in D_1 and D_2. Therefore, any distinguisher of D_0 and D_1 can be used to solve the $\mathbf{Rank}(\mathcal{G}, \ell, \ell, 2, \ell)$ problem with the same advantage and essentially the same running time, since given $g^{\mathbf{T}}$, where $\mathbf{T} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell;r}$ for either $r = 2$ or $r = \ell$, one can choose random $\mathbf{S} \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$, $\widetilde{\mathbf{\Psi}} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell;\ell}$ and $\boldsymbol{x} \in_{\mathrm{R}} \{0,1\}^\ell$ and complete the tuple $(g^{\widetilde{\mathbf{\Psi}}\mathbf{T}}, g^{\mathbf{S}}, g^{\widetilde{\mathbf{\Psi}}}, g^{\mathbf{TS}}, g^{\mathbf{T}\boldsymbol{x}})$ which is distributed exactly as D_0 when $r = \ell$ and D_1 when $r = 2$. The same argument applies to D_2 and D_3 by simply taking $\boldsymbol{z} \in_{\mathrm{R}} \mathbb{Z}_q^\ell$. Therefore,

$$\mathbf{Adv}_{D_0,D_1} \le \mathbf{AdvRank}(\mathcal{G}, \ell, \ell, 2, \ell), \qquad \mathbf{Adv}_{D_2,D_3} \le \mathbf{AdvRank}(\mathcal{G}, \ell, \ell, 2, \ell)$$

Moreover, any distinguisher of D_3 and D_4 can be used to solve the $\mathbf{Rank}(\mathcal{G}, \ell, \ell, \ell-1, \ell)$ problem in a similar way. This time we start with $g^{\mathbf{S}}$, where $\mathbf{S} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell;r}$ for either $r = \ell - 1$ or $r = \ell$, and we take $\mathbf{T} \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$, $\widetilde{\mathbf{\Psi}} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell;\ell}$ and $\boldsymbol{z} \in_{\mathrm{R}} \mathbb{Z}_q^\ell$ to complete the tuple $(g^{\widetilde{\mathbf{\Psi}}\mathbf{T}}, g^{\mathbf{S}}, g^{\widetilde{\mathbf{\Psi}}}, g^{\mathbf{TS}}, g^{\boldsymbol{z}})$, which is distributed exactly as D_3 when $r = \ell$ and D_4 when $r = \ell - 1$. Therefore,

$$\mathbf{Adv}_{D_3,D_4} \le \mathbf{AdvRank}(\mathcal{G}, \ell, \ell, \ell-1, \ell)$$

To compare distributions D_1 and D_2 we will use the Leftover Hashing Lemma (Lemma 1). According to its corollary given in Appendix 2.1 the probability distribution of $(\mathbf{W}, \mathbf{W}\boldsymbol{x})$ for $\mathbf{W} \in_{\mathrm{R}} {\mathbb{Z}_q}^{2\times\ell;2}$ and $\boldsymbol{x} \in_{\mathrm{R}} \{0,1\}^\ell$ is $1/q$-close to the uniform distribution in ${\mathbb{Z}_q}^{2\times\ell;2} \times \mathbb{Z}_q^\ell$, as $\ell > 4\log q$. Since in distributions D_1 and D_2, $\mathbf{T} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell;2}$, then by Lemma 4 for each $\mathbf{T}$ there exists $\mathbf{V} \in {\mathbb{Z}_q}^{\ell\times2;2}$ and $\mathbf{W} \in {\mathbb{Z}_q}^{2\times\ell;2}$ such that $\mathbf{T} = \mathbf{VW}$. Moreover, if $\mathbf{V}$ and $\mathbf{W}$ are uniformly distributed, so is $\mathbf{T} = \mathbf{VW}$. Therefore, from $\mathbf{W}$ and $\boldsymbol{x}$ one can build the tuple $(g^{\widetilde{\mathbf{\Psi}}\mathbf{VW}}, g^{\mathbf{S}}, g^{\widetilde{\mathbf{\Psi}}}, g^{\mathbf{VW}\boldsymbol{x}})$ distributed as D_1, if $\mathbf{V} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times2;2}$, $\mathbf{S} \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$ and $\widetilde{\mathbf{\Psi}} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell;\ell}$, but it follows distributions D_2 when replacing $\mathbf{VW}\boldsymbol{x}$ by a vector $\boldsymbol{z} \in_{\mathrm{R}} \mathbb{Z}_q^\ell$. Thus,

$$\mathbf{Adv}_{D_1,D_2} \le 1/q$$

Remark 1. Actually, taking $\ell > 3\log q$ leads to $\mathbf{Adv}_{D_1,D_2} \le 1/\sqrt{q}$ which is enough because this is the complexity of the generic baby-step giant-step algorithm that solves the discrete logarithm problem and hence any interesting problem in the group $\mathcal{G}$. In addition, the matrix $\mathbf{T}$ (or $\mathbf{W}$) is computationally hidden in the tuple and there is some hope that a computational indistinguishability result exists that would avoid the use of the Leftover Hashing Lemma and then decrease the value of ℓ.

Finally, the indistinguishability of D_4 and D_5 requires a more elaborated reduction. From a matrix $g^{\mathbf{M}}$ such that $\mathbf{M} \in_{\mathrm{R}} {\mathbb{Z}_q}^{(\ell+1)\times(\ell+1);r}$ with either $r = \ell$ or $r = \ell + 1$, a distinguisher chooses $\mathbf{W} \in_{\mathrm{R}} {\mathbb{Z}_q}^{(\ell-1)\times\ell;\ell-1}$, $\mathbf{R}, \widetilde{\mathbf{R}} \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$ and $\boldsymbol{z} \in \mathbb{Z}_q^\ell$, and completes the tuple $(g^{\mathbf{\Psi}}, g^{\mathbf{S}}, g^{\widetilde{\mathbf{\Psi}}}, g^{\widetilde{\mathbf{S}}}, g^{\boldsymbol{z}})$ as follows. It parses $g^{\mathbf{M}} = g^{(\mathbf{M}'|\boldsymbol{m}|\widetilde{\boldsymbol{m}})}$ where $\boldsymbol{m}, \widetilde{\boldsymbol{m}}$ are the last two columns of $\mathbf{M}$, and similarly $\mathbf{R} = (\mathbf{R}' \mid \boldsymbol{r})$ and $\widetilde{\mathbf{R}} = \left(\widetilde{\mathbf{R}}' \mid \widetilde{\boldsymbol{r}}\right)$. Then it takes $g^{\mathbf{\Psi}} = g^{(\mathbf{M}'|\boldsymbol{m})\mathbf{R}^{-1}}$, $g^{\widetilde{\mathbf{\Psi}}} = g^{(M'|\widetilde{\boldsymbol{m}})\widetilde{\mathbf{R}}^{-1}}$, $\mathbf{S} = \mathbf{R}'\mathbf{W}$ and $\widetilde{\mathbf{S}} = \widetilde{\mathbf{R}}'\mathbf{W}$.

If $\mathrm{rank}(\mathbf{M}) = \ell + 1$ then it is easy to see that $\widetilde{\mathbf{\Psi}}$ and $\mathbf{\Psi}$ are uniformly distributed matrices in ${\mathbb{Z}_q}^{(\ell+1)\times\ell;\ell}$ such that $\mathrm{Span}(\widetilde{\mathbf{\Psi}}) \neq \mathrm{Span}(\mathbf{\Psi})$. Indeed $\mathrm{Span}(\mathbf{\Psi}) \cap \mathrm{Span}(\widetilde{\mathbf{\Psi}}) = \mathrm{Span}(\mathbf{M}')$, while $\mathrm{Span}(\mathbf{\Psi}) = \mathrm{Span}\left(\mathbf{M}' \,\middle|\, \boldsymbol{m}\right)$ and $\mathrm{Span}(\widetilde{\mathbf{\Psi}}) = \mathrm{Span}\left(\mathbf{M}' \mid \widetilde{\boldsymbol{m}}\right)$. Is is easy to see that the equation $\mathbf{\Psi}\mathbf{S} = \widetilde{\mathbf{\Psi}}\widetilde{\mathbf{S}}$ implies that $\mathbf{R}^{-1}\mathbf{S}$ and $\widetilde{\mathbf{R}}^{-1}\widetilde{\mathbf{S}}$ are the same matrix with its last row set to $\mathbf{0}$. Then the above expressions for $\mathbf{S}$ and $\widetilde{\mathbf{S}}$ follow, and the probability distribution of the tuple is exactly D_5.

On the other hand, assume for a while $\mathrm{rank}(\mathbf{M}) = \mathrm{rank}\left(\mathbf{M}' \mid \boldsymbol{m}\right) = \mathrm{rank}\left(\mathbf{M}' \mid \widetilde{\boldsymbol{m}}\right) = \ell$. Therefore, $\mathrm{Span}(\mathbf{\Psi}) = \mathrm{Span}\left(\mathbf{M}' \mid \boldsymbol{m}\right) = \mathrm{Span}\left(\mathbf{M}' \mid \widetilde{\boldsymbol{m}}\right) = \mathrm{Span}(\widetilde{\mathbf{\Psi}})$ and hence by Lemma 5 there exists a (unique) matrix $\mathbf{T} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$ such that $\mathbf{\Psi} = \widetilde{\mathbf{\Psi}}\mathbf{T}$. Then, equation $\mathbf{\Psi}\mathbf{S} = \widetilde{\mathbf{\Psi}}\widetilde{\mathbf{S}}$ implies $\widetilde{\mathbf{S}} = \mathbf{TS}$, so $\widetilde{\mathbf{S}}$ is completely determined from $\mathbf{\Psi}$, $\widetilde{\mathbf{\Psi}}$ and $\mathbf{S}$. Observe that now the tuple exactly follows distribution D_4.

However, $\mathrm{rank}\,(\mathbf{M}' \mid \boldsymbol{m}) = \mathrm{rank}\,(\mathbf{M}' \mid \widetilde{\boldsymbol{m}}) = \ell$ does not hold for all matrices in ${\mathbb{Z}_q}^{(\ell+1)\times(\ell+1);\ell}$. Indeed, by a simple counting argument[6], it happens to a random $\mathbf{M} \in_{\mathrm{R}} {\mathbb{Z}_q}^{(\ell+1)\times(\ell+1);\ell}$ with probability at least $1 - 2/q$. Therefore,

$$\mathbf{Adv}_{D_4,D_5} \leq \mathbf{AdvRank}(\mathcal{G}, \ell+1, \ell+1, \ell, \ell+1) + 2/q$$

and summing up

$$\begin{aligned}\mathbf{Adv}_{D_0,D_5} \leq 2\mathbf{AdvRank}(\mathcal{G}, \ell, \ell, 2, \ell) + \mathbf{AdvRank}(\mathcal{G}, \ell, \ell, \ell-1, \ell) + \\ + \mathbf{AdvRank}(\mathcal{G}, \ell+1, \ell+1, \ell, \ell+1) + 3/q\end{aligned}$$

By using Proposition 1 we can write

$$\mathbf{Adv}_{D_0,D_5} \leq \left(2\left\lceil 1.71 \log_2 \ell \right\rceil + 2\right) \mathbf{AdvDLin}(\mathcal{G}) + 3/q$$

□

Let us now proceed with the actual proof of Theorem 2. The proof is structured as a sequence of games played by a challenger $\mathcal{C}$ and an q_e-mKDM-ssID-CPA adversary $\mathcal{A}$, ranging from perfect simulation to perfect hiding of the target plaintexts. Games 0-b, $b \in \{0,1\}$, are exactly the two experiments $\mathbf{ExpKDM\text{-}ssID\text{-}CPA}_{\mathcal{A}}^{b,\Gamma_0(1)}(\lambda, \ell, q_e)$. Let us denote as $\Omega_{i,b}$ the event that $\mathcal{A}$ outputs $b' = 1$ in Game i-b, and let $\mathbf{Adv}_i = |\Pr[\Omega_{i,0}] - \Pr[\Omega_{i,1}]|$. Notice that in the scheme $\Gamma_0(1)$, q_e is not explicitly used and $\mathbf{F}(id) = \mathbf{S}_0 + id\mathbf{S}_1$ and $\widetilde{\mathbf{F}}(id) = \widetilde{\mathbf{\Psi}}_0 + id\widetilde{\mathbf{\Psi}}_1$.

Game 1-b: $\mathcal{C}$ receives as input the parameters $\mathsf{ibp} = (\lambda, \ell, q, \mathcal{G}, g, \mathcal{G}_T, g_T, e(\cdot,\cdot))$ and some input parameters $D = (g^{\mathbf{\Psi}}, g^{\mathbf{S}_1}, g^{\widetilde{\mathbf{\Psi}}_1}, g^{\widetilde{\mathbf{S}}}, g^{\boldsymbol{z}})$ whose probability distribution is the uniform distribution in the subset of ${\mathbb{Z}_q}^{(\ell+1)\times\ell} \times {\mathbb{Z}_q}^{\ell\times\ell} \times {\mathbb{Z}_q}^{(\ell+1)\times\ell} \times {\mathbb{Z}_q}^{\ell\times\ell} \times \mathbb{Z}_q^{\ell}$ defined by the restrictions $\mathbf{\Psi}\mathbf{S}_1 = \widetilde{\mathbf{\Psi}}_1\widetilde{\mathbf{S}}$, $\mathrm{rank}(\widetilde{\mathbf{\Psi}}_1) = \mathrm{rank}(\mathbf{\Psi}) = \ell$, $\mathrm{rank}(\mathbf{S}_1) = \mathrm{rank}(\widetilde{\mathbf{S}}) = \ell$ and $\mathbf{S}_1\widetilde{\mathbf{S}}^{-1}\boldsymbol{z} \in \{0,1\}^{\ell}$. Let $\mathfrak{m} = 1_{\mathcal{G}_T}$ be a fixed message. For a bit $b \in \{0,1\}$ the description of $\mathcal{C}$ follows.

1. **Setup.** $\mathcal{C}$ parses $\mathsf{ibp} = (\lambda, \ell, q, \mathcal{G}, g, \mathcal{G}_T, g_T, e(\cdot,\cdot))$ and $D = (g^{\mathbf{\Psi}}, g^{\mathbf{S}_1}, g^{\widetilde{\mathbf{\Psi}}_1}, g^{\widetilde{\mathbf{S}}}, g^{\boldsymbol{z}})$ as defined above, and sends ibp to the adversary $\mathcal{A}$. Then $\mathcal{S}$ receives the identity $id^\star$ selected by $\mathcal{A}$.
2. **Initialization.** $\mathcal{C}$ generates and sends to $\mathcal{A}$ a master public key $PK = (g^{\mathbf{\Psi}}, g^{\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\mathbf{\Psi}\boldsymbol{x}})$, without knowing the corresponding secret key $SK = g_T^{\boldsymbol{x}} \in {\mathcal{G}_T}^{\ell}$, as follows. $g^{\mathbf{S}_0} = g^{\mathbf{W}\widetilde{\mathbf{S}} - id^\star\mathbf{S}_1}$, $g^{\widetilde{\mathbf{\Psi}}_0} = g^{\mathbf{\Psi}\mathbf{W} - id^\star\widetilde{\mathbf{\Psi}}_1}$, for a random $\mathbf{W} \in {\mathbb{Z}_q}^{\ell\times\ell}$, and $g_T^{-\mathbf{\Psi}\boldsymbol{x}} = 1/e(g^{\widetilde{\mathbf{\Psi}}_1}, g^{\boldsymbol{z}})$. As usually, $g^{\mathbf{F}(id)} = g^{\mathbf{S}_0 + id\mathbf{S}_1}$ and $g^{\widetilde{\mathbf{F}}(id)} = g^{\widetilde{\mathbf{\Psi}}_0 + id\widetilde{\mathbf{\Psi}}_1}$. Observe that $\mathbf{F}(id^\star) = \mathbf{W}\widetilde{\mathbf{S}}$ and $\widetilde{\mathbf{F}}(id^\star) = \mathbf{\Psi}\mathbf{W}$.
3. **Encryption Queries.** When $\mathcal{A}$ asks an encryption query f, where $f \in \mathcal{F}$, if $b = 0$, $\mathcal{C}$ proceeds as follows. Assume $f(SK) = f(g_T^{\boldsymbol{x}}) = g_T^{\lambda_0} \cdot \left(g_T^{x_1}\right)^{\lambda_1} \cdot \ldots \cdot \left(g_T^{x_\ell}\right)^{\lambda_\ell}$. The challenger $\mathcal{C}$ computes the half-encryptions of all secret key bits under PK, $C_j = \mathbf{self\text{-}ref}(j)$ for all $j = 1, \ldots, \ell$ and the half-encryption of g_T, $C_0 = (g^{\mathbf{0}}, \emptyset, g_T)$. Now, $\mathcal{C}$ combines all half-encryptions, component by component, to obtain a half-encryption of $f(SK)$ as (informally) $C = C_0^{\lambda_0} \cdot C_1^{\lambda_1} \cdot \ldots \cdot C_\ell^{\lambda_\ell}$. Finally, $\mathcal{C}$ outputs $\mathbf{enc\text{-}rand}(PK, id^\star, \mathbf{enable\text{-}half}(pk, id^\star, \mathbf{W}, C); \boldsymbol{r})$, for a random $\boldsymbol{r} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell+1}$. Otherwise, if $b = 1$ $\mathcal{C}$ outputs $\mathbf{Enc}(PK, id^\star, g_T; \boldsymbol{r})$, for a random $\boldsymbol{r} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell+1}$.

Remark 2. It is easy to see that (when $b = 0$) the resulting ciphertext is $C = (g^{\boldsymbol{\lambda} + \boldsymbol{r}\mathbf{\Psi}}, g^{(\boldsymbol{\lambda} + \boldsymbol{r}\mathbf{\Psi})\mathbf{W}}, g_T^{\lambda_0} \cdot g_T^{-\boldsymbol{r}\mathbf{\Psi}\boldsymbol{x}})$, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_\ell)$.

[6] $\left|{\mathbb{Z}_q}^{(\ell+1)\times(\ell+1);\ell}\right| = \frac{q^{\ell+1}-1}{q-1}\left|{\mathbb{Z}_q}^{(\ell+1)\times\ell;\ell}\right|$ (being the first factor the number of subspaces of dimension ℓ in $\mathbb{Z}_q^{\ell+1}$, and the second factor the number of generating sets of a given subspace with cardinality $\ell+1$) while the above condition holds for a number of matrices equal to $(q^\ell - q^{\ell-1})\left|{\mathbb{Z}_q}^{(\ell+1)\times\ell;\ell}\right|$ (where the first factor is the number of choices of $\widetilde{\boldsymbol{m}}$ for each choice of $(\mathbf{M}' \mid \boldsymbol{m})$). Thus the probability is $\frac{q^{\ell-1}(q-1)^2}{q^{\ell+1}-1} \geq 1 - \frac{2}{q}$.

4. **Private key Queries.** When $\mathcal{A}$ asks a user's private key query id, where $id \neq id^\star$ then $\mathcal{C}$ answers with $(g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$, computed as

$$g^{\boldsymbol{d}_1} = g^{\mathbf{F}(id)\boldsymbol{t}} \cdot g^{-\frac{1}{id-id^\star}\mathbf{W}\boldsymbol{z}} \qquad\qquad g^{\boldsymbol{d}_2} = g^{\widetilde{\mathbf{S}}\boldsymbol{t}} \cdot g^{-\frac{1}{id-id^\star}\boldsymbol{z}_{\mathcal{G}}}$$

where $\boldsymbol{t} \in_{\mathrm{R}} \mathbb{Z}_q^\ell$.
5. **Final guess.** $\mathcal{C}$ outputs the bit $b' \in \{0,1\}$ sent by $\mathcal{A}$.

It is straightforward to check that Game 0-b and Game 1-b are identical. Therefore,

$$\mathbf{Adv}_0 = \mathbf{Adv}_1$$

Game 2-b: This game only differs from Game 1-b in the probability distribution of the inputs of $\mathcal{C}$. Now the restrictions are $\boldsymbol{\Psi}\mathbf{S}_1 = \widetilde{\boldsymbol{\Psi}}_1\widetilde{\mathbf{S}}$, $\mathrm{rank}(\widetilde{\boldsymbol{\Psi}}_1) = \mathrm{rank}(\boldsymbol{\Psi}) = \ell$, $\mathrm{rank}(\mathbf{S}_1) = \mathrm{rank}(\widetilde{\mathbf{S}}) = \ell - 1$ and $\mathrm{Span}(\boldsymbol{\Psi}) \neq \mathrm{Span}(\widetilde{\boldsymbol{\Psi}}_1)$, which implies $\mathrm{Span}(\boldsymbol{\Psi}) \cap \mathrm{Span}(\widetilde{\boldsymbol{\Psi}}_1) = \mathrm{Span}(\boldsymbol{\Psi}\mathbf{S}_1) = \mathrm{Span}(\widetilde{\boldsymbol{\Psi}}_1\widetilde{\mathbf{S}})$. There is no restriction on $\boldsymbol{z}$. As the inputs of $\mathcal{C}$ in both games match the distributions in Lemma 7,

$$|\mathbf{Adv}_2 - \mathbf{Adv}_1| \leq 4\left(\lceil 1.71 \log_2 \ell \rceil + 1\right) \mathbf{AdvDLin}(\mathcal{G}) + 6/q$$

On the other hand we will see that $\widetilde{\boldsymbol{\Psi}}_1\boldsymbol{z} \notin \mathrm{Span}(\boldsymbol{\Psi})$ with probability $1 - 1/q$, and that this is enough to perfectly hide the plaintexts contained in the ciphertexts returned by the encryption oracle to the adversary.

Given a nonzero vector $\boldsymbol{u} \in \ker(\boldsymbol{\Psi}^\top)$ and any subspace V such that $\mathbb{Z}_q^{\ell+1} = \ker(\boldsymbol{\Psi}^\top) \oplus V$, any (column) vector $\boldsymbol{r} \in \mathbb{Z}_q^{\ell+1}$ can be uniquely written as $\boldsymbol{r} = \boldsymbol{v} + \mu\boldsymbol{u}$, where $\boldsymbol{v} \in V$ and $\mu \in \mathbb{Z}_q$. Actually, if $\boldsymbol{v}$ and μ are chosen uniformly at random, then $\boldsymbol{r}$ is uniformly distributed in $\mathbb{Z}_q^{\ell+1}$. On the other hand, each ciphertext returned by the encryption oracle can be written as $(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_1\mathbf{W}}, c)$ where $\boldsymbol{c}_1 = \boldsymbol{\lambda}_{\mathcal{G}}^\top + \boldsymbol{r}^\top\boldsymbol{\Psi}$ and $c = g_T^{\lambda_0} \cdot g_T^{-\boldsymbol{r}^\top\boldsymbol{\Psi}\boldsymbol{x}} = g_T^{\lambda_0 - \boldsymbol{r}^\top\widetilde{\boldsymbol{\Psi}}_1\boldsymbol{z}}$. But $\boldsymbol{r}^\top\boldsymbol{\Psi} = \boldsymbol{v}^\top\boldsymbol{\Psi} + \mu\boldsymbol{u}^\top\boldsymbol{\Psi}$ and $\boldsymbol{u}^\top\boldsymbol{\Psi} = (\boldsymbol{\Psi}^\top\boldsymbol{u})^\top = \mathbf{0}$. Thus, the first two components of the ciphertext do not depend on μ. However, $\boldsymbol{r}^\top\widetilde{\boldsymbol{\Psi}}_1\boldsymbol{z} = \boldsymbol{v}^\top\widetilde{\boldsymbol{\Psi}}_1\boldsymbol{z} + \mu\boldsymbol{u}^\top\widetilde{\boldsymbol{\Psi}}_1\boldsymbol{z}$, and $\boldsymbol{u}^\top\widetilde{\boldsymbol{\Psi}}_1\boldsymbol{z} \neq \mathbf{0}$ with probability $1 - 1/q$, which implies that c does depend on μ. Indeed, $\boldsymbol{v}^\top\widetilde{\boldsymbol{\Psi}}_1 = \mathbf{0}$ would imply $\ker(\boldsymbol{\Psi}^\top) = \ker(\widetilde{\boldsymbol{\Psi}}_1^\top)$, which contradicts $\mathrm{Span}(\boldsymbol{\Psi}) \neq \mathrm{Span}(\widetilde{\boldsymbol{\Psi}}_1)$[7]. Moreover, $\boldsymbol{u}^\top\widetilde{\boldsymbol{\Psi}}_1\boldsymbol{z} = \mathbf{0}$ can only happen if $\boldsymbol{z}$ lies on a specific $(\ell-1)$-dimensional subspace of $\mathbb{Z}_q^\ell$, which occurs with probability $1/q$.

Game 3-b: The only difference from Game 2-b is that the third component e of the challenge ciphertexts are taken at random in $\mathcal{G}_T$. From the above explanation,

$$|\mathbf{Adv}_2 - \mathbf{Adv}_3| \leq 2/q$$

as every ciphertext in Game 2-b uses its own random value μ. Obviously in Game 3-b all plaintexts are perfectly hidden to the adversary and

$$\mathbf{Adv}_3 = 0$$

Summing all up

$$\mathbf{AdvKDM\text{-}ssID\text{-}CPA}_{\mathcal{A}}^{\Gamma_0(1)}(\lambda, \ell, q_e; t) = \mathbf{Adv}_0 = \mathbf{Adv}_1 \leq$$

$$\leq \mathbf{Adv}_2 + 4\left(\lceil 1.71 \log_2 \ell \rceil + 1\right) \mathbf{AdvDLin}(\mathcal{G}; t') + 6/q \leq$$

$$\leq 4\left(\lceil 1.71 \log_2 \ell \rceil + 1\right) \mathbf{AdvDLin}(\mathcal{G}; t') + 8 \cdot 2^{-\lambda}$$

This completes the proof of Theorem 2.

[7] As $\mathrm{Span}(\boldsymbol{\Psi})$ is the orthogonal subspace $(\ker(\boldsymbol{\Psi}^\top))^\perp$, and similarly $\mathrm{Span}(\widetilde{\boldsymbol{\Psi}}_1) = (\ker(\widetilde{\boldsymbol{\Psi}}_1^\top))^\perp$, then if the two kernels are equal so are $\mathrm{Span}(\boldsymbol{\Psi})$ and $\mathrm{Span}(\widetilde{\boldsymbol{\Psi}}_1)$.

Theorem 3. *If $\Gamma_0(1)$ is nq_e-mKDM-ssID-CPA secure then $\Gamma_0(q_e)$ is (n, q_e)-mKDM-sID-CPA secure. In particular,* $\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma_0(1), \lambda, \ell, n, q_e; t) \leq 2n \cdot 2^{-\lambda} + \mathbf{AdvKDM\text{-}ssID\text{-}CPA}(\Gamma_0(q_e), \lambda, \ell, nq_e; t')$.

Proof. Let $\mathcal{A}_n$ be an adversary against (n, q_e)-mKDM-sID-CPA security of $\Gamma_0(q_e)$. We show how to build another adversary $\mathcal{A}_1$ against nq_e-mKDM-ssID-CPA security of $\Gamma_0(1)$, which uses $\mathcal{A}_n$ as a subroutine and has essentially the same advantage and running time.

Let Game 0-b, $b \in \{0,1\}$, be the experiment $\mathbf{ExpKDM\text{-}sID\text{-}CPA}^{b,\Gamma_0(q_e)}_{\mathcal{A}_n}(\lambda, \ell, n, q_e)$ played by $\mathcal{A}_n$ and a challenger $\mathcal{C}_n$. Let us denote as $\Omega_{i,b}$ the event that $\mathcal{A}_n$ outputs $b' = 1$ in Game i-b, and let $\mathbf{Adv}_i = |\Pr[\Omega_{i,0}] - \Pr[\Omega_{i,1}]|$. Notice that an inequality $|\Pr[\Omega_{i,b}] - \Pr[\Omega_{i+1,b}]| < \epsilon$ implies $|\mathbf{Adv}_i - \mathbf{Adv}_{i+1}| < 2\epsilon$.

In Game 1-b the public and secret keys are generated in a completely different way. The challenger $\mathcal{C}_n$ is divided into two entities: A challenger $\mathcal{C}_1$ playing the experiment $\mathbf{ExpKDM\text{-}ssID\text{-}CPA}^{b,\Gamma_0(1)}_{\mathcal{A}_1}(\lambda, \ell, nq_e)$ and a simulator $\mathcal{S}$ which adapts the experiment to $\mathcal{A}_n$. The simulator $\mathcal{S}$ works as follows:

1. **Setup.** $\mathcal{S}$ parses $\mathsf{ibp} = (\lambda, \ell, q, \mathcal{G}, g, \mathcal{G}_T, g_T, e(\cdot,\cdot))$ received from $\mathcal{C}_1$. Then $\mathcal{S}$ forwards ibp to $\mathcal{A}_n$, selects an identity $id^\star$ and sends it to $\mathcal{C}_1$. Eventually, $\mathcal{S}$ receives from $\mathcal{A}_n$ nq_e selected identities $\mathcal{I}^\star = (id_1^1, \ldots, id_n^{q_e})$.
2. **Initialization.** After receiving a master public key $PK = (g^{\boldsymbol{\Psi}}, g^{\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\boldsymbol{\Psi}\boldsymbol{x}})$ from $\mathcal{C}_1$, $\mathcal{S}$ generates n independent master public keys $PK_1, \ldots, PK_n$, without knowing the corresponding secret keys, as follows.
   For each $k = 1, \ldots, n$, $\mathcal{S}$ chooses a XOR mask $\boldsymbol{a}_k \in_{\mathrm{R}} \{0,1\}^\ell$. Let $(\mathbf{M}_k, \boldsymbol{\mu}_k)$ be the corresponding affine transformation. The k-th master public key is set to

$$PK_k = \mathbf{PK\text{-}rand}(\mathbf{id\text{-}transf}(\mathbf{reencrXOR\text{-}PK}(PK; \mathbf{M}_k, \boldsymbol{\mu}_k); P^{(k)}); \mathbf{L}_k, \mathbf{R}_k, \mathbf{Q}^{(k)}(\cdot))$$

   where for any $k = 1, \ldots, n$, $\mathbf{L}_k \in_{\mathrm{R}} \mathrm{GL}_{\ell+1}(\mathbb{Z}_q)$, $\mathbf{R}_k \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$, $\mathbf{Q}^{(k)}(id) = \sum_{j=0}^{q_e} \mathbf{Q}_j^{(k)} id^j$ is a random (matrix) polynomial of degree at most q_e, as described in **PK-rand**, and $P^{(k)}(id) = id^\star + \prod_{j=1}^{q_e}(id - id_k^j)$. With probability at least $1 - n/(q-1)$, all PK_k are valid master public keys for unknown random secret keys, related in a known way. Now $\mathcal{S}$ stores $\mathbf{M}_k, \boldsymbol{\mu}_k, \mathbf{L}_k, \mathbf{R}_k, \mathbf{Q}^{(k)}, P^{(k)}$ and sends all PK_k to $\mathcal{A}_n$.

   Remark 3. One can show that the public key elements are given by the equations

$$g^{\boldsymbol{\Psi}^{(k)}} = g^{\mathbf{L}_k \boldsymbol{\Psi} \mathbf{M}_k}, \qquad g^{\widetilde{\mathbf{S}}^{(k)}} = g^{\widetilde{\mathbf{S}} R_k}, \qquad \left(g_T^{-\boldsymbol{\Psi}\boldsymbol{x}}\right)^{(k)} = g_T^{-\mathbf{L}_k \boldsymbol{\Psi}\boldsymbol{x}} \cdot e(g^{\boldsymbol{\Psi}}, g^{\boldsymbol{\mu}_k})^{\mathbf{L}_k}$$
$$g^{\mathbf{F}^{(k)}(id)} = g^{\left(\mathbf{M}_k \mathbf{F}(P^{(k)}(id)) + \mathbf{Q}^{(k)}(id)\widetilde{\mathbf{S}}\right)\mathbf{R}_k}$$
$$g^{\widetilde{\mathbf{F}}^{(k)}(id)} = g^{\mathbf{L}_k\left(\widetilde{\mathbf{F}}(P^{(k)}(id)) + \boldsymbol{\Psi}\mathbf{M}_k\mathbf{Q}^{(k)}(id)\right)}$$

   Moreover, if $(g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$ is a valid secret key for PK and $id^{(k)} = P^{(k)}(id)$ then $(g^{\mathbf{M}_k \boldsymbol{d}_1 + \boldsymbol{\mu}_k} \cdot g^{\mathbf{Q}^{(k)}(id)\boldsymbol{d}_2}, g^{\boldsymbol{d}_2})$ is a valid secret secret key for PK_k and id, and a ciphertext $(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ valid for PK and $id^{(k)} = P^{(k)}(id)$ decrypts to the same plaintext as $(g^{\boldsymbol{c}_1 \mathbf{M}_k}, g^{\boldsymbol{c}_2} \cdot g^{\boldsymbol{c}_1 \mathbf{M}_k \mathbf{Q}^{(k)}(id)}, c \cdot e(g^{\boldsymbol{c}_1}, g^{\boldsymbol{\mu}_k}))$ for PK_k and id. On the other hand, observe that $P^{(k)}(id_k^j) = id^\star$, for all $k = 1, \ldots, n$ and $j = 1, \ldots, q_e$.
3. **Encryption Queries.** When $\mathcal{A}_n$ asks an encryption query (k, f), where $f \in \mathcal{F}$, $\mathcal{S}$ proceeds as follows. Let us denote $SK_k = g_T^{\boldsymbol{x}^{(k)}}$ for $k = 1, \ldots, n$. Assume $f(SK_1, \ldots, SK_n) = g_T^{\lambda_0} \cdot \left(g_T^{x_1^{(1)}}\right)^{\lambda_{11}} \cdot \ldots \cdot \left(g_T^{x_\ell^{(n)}}\right)^{\lambda_{n\ell}}$. Since $\boldsymbol{x}^{(k)} = \boldsymbol{a}_k \oplus \boldsymbol{x}$, then one can write $x_j^{(k)} = a_{kj} \oplus x_j = a_{kj} + (1 - 2a_{kj})x_j$. Therefore $\mathcal{S}$ can efficiently compile f as an affine function $\widetilde{f}$ of $\boldsymbol{x}$. Now $\mathcal{S}$ queries its own encryption oracle on $(k, \widetilde{f})$, obtaining an encryption $(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ under PK and identity $id^\star$, that is also valid for PK_k and $id_k^{j_k}$, where j_k counts the queries made for k. So $\mathcal{S}$ simply has to reencrypt this ciphertext under PK_k and identity $id_k^{j_k}$ and send the resulting

$$\mathbf{PK\text{-}rand\text{-}ciph}(\mathbf{reencrXOR\text{-}ciph}(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c; \mathbf{M}_k, \boldsymbol{\mu}_k); \mathbf{L}_k, \mathbf{R}_k, \mathbf{Q}^{(k)}(\cdot))$$

4. **Private key Queries.** When $\mathcal{A}_n$ asks a user's private key query (k, id), where $id \notin \{id_k^1, \ldots, id_k^{q_e}\}$, $\mathcal{S}$ forwards the query to its own oracle, asking a user's private key for identity $P^{(k)}(id)$, which is different from $id^\star$. If the oracle answer is $(g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$, then $\mathcal{S}$ computes and sends $\mathbf{PK\text{-}rand\text{-}user}(\mathbf{reencr\text{-}user}(g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}; \mathbf{M}_k, \boldsymbol{\mu}_k); \mathbf{L}_k, \mathbf{R}_k, \mathbf{Q}^{(k)}(\cdot))$
5. **Final guess.** Output the bit $b' \in \{0,1\}$ sent by $\mathcal{A}_n$.

The simulation is perfect except when the keys are not correctly generated (because of **PK-rand**), which happens with negligible probability at most $n/(q-1)$. Therefore, $|\mathbf{Adv}_0 - \mathbf{Adv}_1| \leq 2n/(q-1)$ and $\mathbf{AdvKDM\text{-}sID\text{-}CPA}_{\mathcal{A}_n}^{\Gamma_0(q_e)}(\lambda, \ell, n, q_e) = \mathbf{Adv}_0 \leq 2n/(q-1) + \mathbf{Adv}_1 \leq 2n \cdot 2^{-\lambda} + \mathbf{AdvKDM\text{-}ssID\text{-}CPA}_{\mathcal{A}_1}^{\Gamma_0(1)}(\lambda, \ell, nq_e)$ □

Corollary 2. *From Theorems 2 and 3, we obtain*

$$\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma_0(q_e), \lambda, \ell, n, q_e; t) \leq 2(n+4)2^{-\lambda} + 4\left(\lceil 1.71 \log_2 \ell \rceil + 1\right) \mathbf{AdvDLin}(\mathcal{G}; t')$$

Note that the loss factor in the reduction is constant with respect to the number n of master keys and the number q_e of allowed queries. The factor only grows logarithmically on the security parameter ℓ. One disadvantage of our schemes is that the size of the public key is proportional to q_e. When the CHK transformation is applied to our IBE schemes together with Mohassel's one-time signature scheme [25], the resulting public key schemes achieve KDM-CCA security, with a reduction loss factor that does not depend on n nor q_e. In contrast, the loss factor in the security reduction for the KDM-CCA secure scheme in [13] is linear in the number q_e of encryption queries.

4.3 Improving Efficiency, Γ_2

In this section, the size of the master public key of $\Gamma_0(q_e)$ is (drastically) reduced without any noticeable effect in the security of the scheme. The resulting scheme $\Gamma_2(q_e)$ is described below.

$\Gamma_2.\mathsf{Stp}(1^\lambda)$*:* identical to $\Gamma_0.\mathsf{Stp}(1^\lambda)$, which leads to $\mathsf{ibp} = (\lambda, \ell, q, \mathcal{G}, g, \mathcal{G}_T, g_T, e(\cdot,\cdot))$, where still $\ell > 4\lambda$.

$\Gamma_2.\mathsf{Mkg}_{\mathsf{ibp}}()$*:* $PK = (g^{\boldsymbol{\Psi}}, g^{\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\mathbf{Psi}\boldsymbol{x}})$ and $SK = g_T^{\boldsymbol{x}}$, where $\boldsymbol{x} \in_{\mathrm{R}} \{0,1\}^\ell$, $\boldsymbol{\Psi} \in_{\mathrm{R}} \mathbb{Z}_q^{2\times \ell;2}$, $\widetilde{\mathbf{S}} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell \times 2;2}$, $\mathbf{F}(id) = \mathbf{T}(id)\widetilde{\mathbf{S}} \in \mathbb{Z}_q^{\ell\times 2}$, $\widetilde{\mathbf{F}}(id) = \boldsymbol{\Psi}\mathbf{T}(id) \in \mathbb{Z}_q^{2\times\ell}$ for a random (matrix) polynomial $\mathbf{T}(id) = \sum_{j=0}^{q_e} \mathbf{T}_j id^j$, with $\mathbf{T}_0, \ldots, \mathbf{T}_{q_e-1} \in_{\mathrm{R}} \mathbb{Z}_q^{\ell\times\ell}$ and $\mathbf{T}_{q_e} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$. Clearly, $\boldsymbol{\Psi}\mathbf{F}(id) =$
$mathbftF(id)\widetilde{\mathbf{S}}$ holds again.

$\Gamma_2.\mathsf{Ukg}_{\mathsf{ibp}}(SK, id)$*:* for an identity id the secret key $sk[id] = (g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2}) \in \mathcal{G}^\ell \times \mathcal{G}^\ell$ is generated as $g^{\boldsymbol{d}_1} = g^{\boldsymbol{x}} \cdot g^{\mathbf{F}(id)\boldsymbol{t}}$ and $g^{\boldsymbol{d}_2} = g^{\widetilde{\mathbf{S}}\boldsymbol{t}}$, where $\boldsymbol{t} \in_{\mathrm{R}} \mathbb{Z}_q^2$ and $g^{\boldsymbol{x}}$ is computed component by component from $SK = g_T^{\boldsymbol{x}}$. The user can verify the validity of the secret key by checking the equation $g_T^{-\boldsymbol{\Psi}\boldsymbol{x}} \cdot e(g^{\boldsymbol{\Psi}}, g^{\boldsymbol{d}_1}) = e(g^{\mathbf{t}\mathbf{F}(id)}, g^{\boldsymbol{d}_2})$.

$\Gamma_2.\mathsf{Enc}_{\mathsf{ibp}}(PK, id, m)$*:* to encrypt a message $m \in \mathcal{G}_T$ for an identity id and master public key PK, a row vector $\boldsymbol{r} \in_{\mathrm{R}} \mathbb{Z}_q^2$ is chosen and the ciphertext $C = (g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c) \in \mathcal{G}^\ell \times \mathcal{G}^\ell \times \mathcal{G}_T$ is computed as $g^{\boldsymbol{c}_1} = g^{\boldsymbol{r}\boldsymbol{\Psi}}$, $g^{\boldsymbol{c}_2} = g^{\boldsymbol{r}\mathbf{t}\mathbf{F}(id)}$ and $c = m \cdot g_T^{-\boldsymbol{r}\boldsymbol{\Psi}\boldsymbol{x}}$. The ciphertext fulfils the equation $e(g^{\boldsymbol{c}_1}, g^{\mathbf{F}(id)}) = e(g^{\boldsymbol{c}_2}, g^{\widetilde{\mathbf{S}}})$.

$\Gamma_2.\mathsf{Dec}_{\mathsf{ibp}}(sk[id], c)$*:* let $C = (g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ be a ciphertext for an identity id. The user who owns $sk[id] = (g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$ recovers $m = c \cdot e(g^{\boldsymbol{c}_1}, g^{\boldsymbol{d}_1})/e(g^{\boldsymbol{c}_2}, g^{\boldsymbol{d}_2})$.

The main differences with respect to Γ_0 are in $\Gamma_2.\mathsf{Mkg}_{\mathsf{ibp}}()$, because now the (matrix) polynomials $\mathbf{F}(\cdot), \widetilde{\mathbf{F}}(\cdot)$ are generated in a slightly different way, and all matrices in PK have now 2ℓ elements, instead of ℓ^2 or $\ell^2 + \ell$ elements in Γ_0. This means the length of PK is reduced by a factor $\ell/2 > 2\lambda$.

Theorem 4. *Under the assumption that the Rank problem is hard, if $\Gamma_0(q_e)$ is (n, q_e)-mKDM-sID-CPA secure then Γ_2 is also (n, q_e)-mKDM-sID-CPA secure. In particular,* $\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma_2(q_e), \lambda, \ell, n, q_e; t) \leq 4\lceil 1.71 \log_2(\ell+1)\rceil\ \mathbf{AdvDLin}(\mathcal{G}; t') + 4n \cdot 2^{-\lambda} + \mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma_0(q_e), \lambda, \ell, n, q_e; t'')$

To prove this theorem, we define a new scheme Γ_1, which can be seen as an intermediate step between Γ_0 and Γ_2. With respect to Γ_0, the number of rows of both $\mathbf{\Psi}$ and the (matrix) coefficients of $\widetilde{\mathbf{F}}(\cdot)$ is reduced from $\ell+1$ to 2, and the size of $g_T^{-\mathbf{Psi}\boldsymbol{x}}$ is reduced accordingly. The main difference with respect to Γ_2 is that the number of columns of both $\widetilde{\mathbf{S}}$ and the (matrix) coefficients of $\mathbf{F}(\cdot)$ is reduced from ℓ in Γ_1 to 2 in Γ_2.

$\Gamma_1.\mathsf{Stp}(1^\lambda)$*:* identical to $\Gamma_0.\mathsf{Stp}(1^\lambda)$.

$\Gamma_1.\mathsf{Mkg}_{\mathsf{ibp}}()$*:* as in $\Gamma_0(q_e)$, $PK = (g^{\mathbf{\Psi}}, g^{\mathbf{F}(\cdot)}, g^{\widetilde{\mathbf{F}}(\cdot)}, g^{\widetilde{\mathbf{S}}}, g_T^{-\mathbf{Psi}\boldsymbol{x}})$ and $SK = g_T^{\boldsymbol{x}}$, where $\widetilde{\mathbf{S}} \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$, $\boldsymbol{x} \in_{\mathrm{R}} \{0,1\}^\ell$ and the (matrix) coefficients of $\mathbf{F}(\cdot)$ are $\mathbf{S}_0, \ldots, \mathbf{S}_{q_e-1} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell}$, $\mathbf{S}_{q_e} \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$. But now $\mathbf{\Psi} \in_{\mathrm{R}} {\mathbb{Z}_q}^{2\times\ell;2}$, $g_T^{-\mathbf{Psi}\boldsymbol{x}} \in {\mathcal{G}_T}^2$ and $\widetilde{\mathbf{F}}(\cdot)$ is computed accordingly. Namely, $\widetilde{\mathbf{\Psi}}_j = \mathbf{\Psi}\mathbf{S}_j\widetilde{\mathbf{S}}^{-1} \in {\mathbb{Z}_q}^{2\times\ell}$ for $j = 0, \ldots, q_e$, so that $\mathbf{\Psi}\mathbf{F}(id) = \widetilde{\mathbf{F}}(id)\widetilde{\mathbf{S}}$.

$\Gamma_1.\mathsf{Ukg}_{\mathsf{ibp}}(SK, id)$*:* identical to $\Gamma_0.\mathsf{Ukg}_{\mathsf{ibp}}(SK, id)$.

$\Gamma_1.\mathsf{Enc}_{\mathsf{ibp}}(PK, id, m)$*:* to encrypt a message $m \in \mathcal{G}_T$ for an identity $id \in \mathbb{Z}_q$, a row vector $\boldsymbol{r} \in_{\mathrm{R}} \mathbb{Z}_q^2$ is chosen and the ciphertext $C = (g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c) \in \mathcal{G}^\ell \times \mathcal{G}^\ell \times \mathcal{G}_T$ is computed as $g^{\boldsymbol{c}_1} = g^{\boldsymbol{r}\mathbf{\Psi}}$, $g^{\boldsymbol{c}_2} = g^{\boldsymbol{r}\mathbf{t}\mathbf{F}(id)}$ and $c = m \cdot g_T^{-\boldsymbol{r}\mathbf{\Psi}\boldsymbol{x}}$. The ciphertext fulfils the equation $e(g^{\boldsymbol{c}_1}, g^{\mathbf{F}(id)}) = e(g^{\boldsymbol{c}_2}, g^{\widetilde{\mathbf{S}}})$, so its consistency with respect to identity id can be publicly verified.

$\Gamma_1.\mathsf{Dec}_{\mathsf{ibp}}(sk[id], c)$*:* identical to $\Gamma_0.\mathsf{Dec}_{\mathsf{ibp}}(sk[id], c)$.

The proof of Theorem 4 directly follows from Propositions 3 and 4 below.

Proposition 3. *Under the assumption that the Rank problem is hard, if $\Gamma_0(q_e)$ is (n, q_e)-mKDM-sID-CPA secure then $\Gamma_1(q_e)$ is also (n, q_e)-mKDM-sID-CPA secure. In particular,*

$$\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma_1(q_e), \lambda, \ell, n, q_e; t) \leq 2\left\lceil 1.71 \log_2(\ell+1)\right\rceil \mathbf{AdvDLin}(\mathcal{G}; t') + 2n \cdot 2^{-\lambda} +$$

$$+\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma_0(q_e), \lambda, \ell, n, q_e; t'')$$

Proof. The proof is also structured as a sequence of games played by a challenger $\mathcal{C}_1$ and an (n, q_e)-mKDM-sID-CPA adversary $\mathcal{A}_1$ against $\Gamma_1(q_e)$. Games 0-b, $b \in \{0,1\}$, are exactly the two (n, q_e)-mKDM-sID-CPA security games defined in section 3.1. Let us denote as $\Omega_{i,b}$ the event that $\mathcal{A}_1$ outputs $b' = 1$ in Game i-b, and let $\mathbf{Adv}_i = |\Pr[\Omega_{i,0}] - \Pr[\Omega_{i,1}]|$. We will show in the last game a (n, q_e)-mKDM-sID-CPA adversary $\mathcal{A}_0$ against $\Gamma_0(q_e)$ with essentially the same advantage and running time.

Game 1-b: The challenger $\mathcal{C}_1$ generates the n independent master keys $PK_1, \ldots, PK_n$ and $SK_1, \ldots, SK_n$ in a different but almost equivalent way. $\mathcal{C}_1$ generates n independent master public and secret keys for $\Gamma_0(q_e)$ instead of $\Gamma_1(q_e)$. Then, it picks some random matrices $\mathbf{H} \in_{\mathrm{R}} {\mathbb{Z}_q}^{(\ell+1)\times(\ell+1);2}$ and $\mathbf{L}_1, \ldots, \mathbf{L}_n \in_{\mathrm{R}} {\mathbb{Z}_q}^{2\times(\ell+1);2}$, and computes the master public keys for $\Gamma_1(q_e)$ by just replacing $g^{\mathbf{\Psi}^{(k)}}$ and $g^{\widetilde{\mathbf{\Psi}}_j^{(k)}}$ respectively with $g^{\mathbf{L}_k\mathbf{H}\mathbf{\Psi}^{(k)}}$ and $g^{\mathbf{L}_k\mathbf{H}\widetilde{\mathbf{\Psi}}_j^{(k)}}$, for $j = 0, \ldots, q_e$. Finally, $\left(g_T^{-\mathbf{Psi}\boldsymbol{x}}\right)^{(k)} = e(g^{-\mathbf{\Psi}^{(k)}}, g^{\boldsymbol{x}^{(k)}})$ is replaced with $e(g^{-\mathbf{L}_k\mathbf{H}\mathbf{\Psi}^{(k)}}, g^{\boldsymbol{x}^{(k)}})$. Observe that $\mathcal{C}_1$ needs to know $g^{\mathbf{\Psi}^{(k)}}$ and the coefficients $g^{\widetilde{\mathbf{\Psi}}_j^{(k)}}$ (which it learns from the key generation of $\Gamma_0(q_e)$). User key queries are handled as in Game 0-b, but the way target ciphertexts are computed is changed. Namely, each ciphertext $(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ is computed as $g^{\boldsymbol{c}_1} = g^{\boldsymbol{r}'\mathbf{H}\mathbf{\Psi}^{(k)}}$, $g^{\boldsymbol{c}_2} = g^{\boldsymbol{r}'\mathbf{H}\left(\sum_{j=0}^{q_e} mathbftPsi_j^{(k)} id^j\right)}$ and $c = m \cdot g_T^{-\boldsymbol{r}'\mathbf{H}\mathbf{Psi}^{(k)}\boldsymbol{x}^{(k)}}$, where now $\boldsymbol{r}' \in_{\mathrm{R}} \mathbb{Z}_q^{\ell+1}$.

Let us see that with overwhelming probability the generated keys are valid keys for $\Gamma_1(q_e)$ and with the same distribution as in Game 0-b. Actually, it suffices to check whether $g^{\mathbf{L}_k\mathbf{H}\mathbf{\Psi}^{(k)}}$ has the right distribution, since the other elements in the master public key are generated exactly the same way as in Game 0-b.

Observe that $\mathrm{rank}(\mathbf{L}_k\mathbf{H}\boldsymbol{\Psi}^{(k)}) = 2$ with probability at least $1 - 1/(q-1)$,[8] and the randomness in $\mathbf{L}_k$ and $\boldsymbol{\Psi}^{(k)}$ guarantees the uniformity of the resulting matrix in ${\mathbb{Z}_q}^{2\times\ell;2}$, and its independence of the other elements in the master public keys. Indeed, by Lemma 3, $\mathbf{L}_k$ could be computed from a fixed $\mathbf{L} \in {\mathbb{Z}_q}^{2\times(\ell+1);2}$ as $\mathbf{L}_k = \mathbf{U}_k\mathbf{L}\mathbf{V}_k$, where $\mathbf{U}_k \in_{\mathrm{R}} \mathrm{GL}_2(\mathbb{Z}_q)$ and $\mathbf{V}_k \in_{\mathrm{R}} \mathrm{GL}_{\ell+1}(\mathbb{Z}_q)$. And similarly $\boldsymbol{\Psi}^{(k)} = \mathbf{W}_k\boldsymbol{\Psi}\mathbf{X}_k$, where $\boldsymbol{\Psi} \in {\mathbb{Z}_q}^{\ell+1\times\ell;\ell}$, $\mathbf{W}_k \in_{\mathrm{R}} \mathrm{GL}_{\ell+1}(\mathbb{Z}_q)$ and $\mathbf{X}_k \in_{\mathrm{R}} \mathrm{GL}_{\ell}(\mathbb{Z}_q)$. Therefore, for each choice of $\mathbf{V}_k$, $\mathbf{L}$, $\mathbf{H}$, $\mathbf{W}_k$, $\boldsymbol{\Psi}$ such that $\mathrm{rank}(\mathbf{L}\mathbf{V}_k\mathbf{H}\mathbf{W}_k\boldsymbol{\Psi}) = 2$, due to the randomness in $\mathbf{U}_k$ and $\mathbf{X}_k$, $\mathbf{L}_k\mathbf{H}\boldsymbol{\Psi}^{(k)}$ is uniformly distributed in ${\mathbb{Z}_q}^{2\times\ell;2}$.

Furthermore, $\mathrm{rank}(\mathbf{L}_k\mathbf{H}) = \mathrm{rank}(\mathbf{H}) = 2$ implies $\mathrm{Span}(\mathbf{H}^\top\mathbf{L}_k^\top) = \mathrm{Span}(\mathbf{H}^\top)$. Therefore, the probability distributions of $\boldsymbol{r}\mathbf{L}_k\mathbf{H}$ for $\boldsymbol{r} \in_{\mathrm{R}} \mathbb{Z}_q^2$ and $\boldsymbol{r}'\mathbf{H}$ for $\boldsymbol{r}' \in_{\mathrm{R}} \mathbb{Z}_q^{\ell+1}$ are the same, and the simulation of the encryption oracle is perfect (assuming $\mathrm{rank}(\mathbf{L}_k\mathbf{H}\boldsymbol{\Psi}^{(k)}) = 2$ as above). Then, $|\mathbf{Adv}_0 - \mathbf{Adv}_1| \le 2n/(q-1)$.

Game 2-b: The same as Game 1-b but now $\mathbf{H} \in \mathrm{GL}_{\ell+1}(\mathbb{Z}_q)$. From Games 1-$b$ and 2-b one can build a distinguisher for $\mathbf{Rank}(\mathcal{G}, \ell+1, \ell+1, 2, \ell+1)$ problem. Therefore,

$$|\mathbf{Adv}_1 - \mathbf{Adv}_2| \le 2\left\lceil 1.71\log_2(\ell+1)\right\rceil \mathbf{AdvDLin}(\mathcal{G})$$

Game 3-b: The same as Game 2-b but now $\mathbf{H} = \mathbf{I}_{\ell+1}$ (that is, $\mathbf{H}$ is removed). However, Games 2-b and 3-b are identical, since the distribution of keys and ciphertexts remain unchanged. Indeed, the public key elements are now computed as $g^{\mathbf{L}_k\boldsymbol{\Psi}^{(k)}}$, $g^{\mathbf{L}_k\widetilde{\boldsymbol{\Psi}}_j^{(k)}}$ and $e(g^{-\mathbf{L}_k\boldsymbol{\Psi}^{(k)}}, g^{\boldsymbol{x}^{(k)}})$. The target ciphertexts $(g^{\boldsymbol{c}_1}, g^{\boldsymbol{c}_2}, c)$ are computed exactly as in $\Gamma_0(q_e)$. In other words, $g^{\boldsymbol{c}_1} = g^{\boldsymbol{r}'\boldsymbol{\Psi}^{(k)}}$, $g^{\boldsymbol{c}_2} = g^{\boldsymbol{r}'\,\widetilde{F}^{(k)}(id)}$ and $c = m \cdot g_T^{-\boldsymbol{r}'\mathbf{Psi}^{(k)}\boldsymbol{x}^{(k)}}$, where $\boldsymbol{r}' \in_{\mathrm{R}} \mathbb{Z}_q^{\ell+1}$. Thus, stepping from Game 2-b to Game 3-b is just replacing $\mathbf{L}_k\mathbf{H}$ with $\mathbf{L}_k$ and $\boldsymbol{r}'\mathbf{H}$ with $\boldsymbol{r}'$. Therefore, $\mathbf{Adv}_2 = \mathbf{Adv}_3$.

Game 4-b: In this game, a challenger $\mathcal{C}$ plays with an adversary $\mathcal{A}_0$ against the scheme $\Gamma_0(q_e)$, which uses $\mathcal{A}_1$ as a subroutine. Basically, $\mathcal{A}_0$ picks random matrices $\mathbf{L}_1, \ldots, \mathbf{L}_n \in_{\mathrm{R}} {\mathbb{Z}_q}^{2\times(\ell+1);2}$ and forwards all the messages in either direction, except that the master public key elements $g^{\boldsymbol{\Psi}^{(k)}}$, $g^{\widetilde{\mathbf{F}}^{(k)}}$ and $\left(g_T^{-\mathbf{Psi}\boldsymbol{x}}\right)^{(k)}$ are respectively replaced with $g^{\mathbf{L}_k\boldsymbol{\Psi}^{(k)}}$, $g^{\mathbf{L}_k\widetilde{\mathbf{F}}^{(k)}}$ and $\left(g_T^{-\mathbf{L}_k\mathbf{Psi}\boldsymbol{x}}\right)^{(k)}$. Notice that $\mathcal{A}_0$ does not use any secret key, and the user key and encryption queries by $\mathcal{A}_1$ are directly answered by the oracles given to $\mathcal{A}_0$. Therefore,

$$\mathbf{Adv}_4 = \mathbf{AdvKDM\text{-}sID\text{-}CPA}_{\mathcal{A}_0}^{\Gamma_0(q_e)}(\lambda, \ell, n, q_e)$$

Since Games 3-b and 4-b are also identical, $\mathbf{Adv}_3 = \mathbf{Adv}_4$. and

$$\mathbf{AdvKDM\text{-}sID\text{-}CPA}_{\mathcal{A}_1}^{\Gamma_1(q_e)}(\lambda, \ell, n, q_e) = \mathbf{Adv}_0 \le 2\left\lceil 1.71\log_2(\ell+1)\right\rceil \mathbf{AdvDLin} + 2n/(q-1) + \mathbf{Adv}\mathcal{A}_0 \le$$

$$= 2\left\lceil 1.71\log_2(\ell+1)\right\rceil \mathbf{AdvDLin}(\mathcal{G}, t') + 2n\cdot 2^{-\lambda} + \mathbf{AdvKDM\text{-}sID\text{-}CPA}_{\mathcal{A}_0}^{\Gamma_0(q_e)}(\lambda, \ell, n, q_e)$$

□

The following proposition (and its proof) verbosely follows the previous one.

Proposition 4. *Under the assumption that the Rank problem is hard, if $\Gamma_1(q_e)$ is (n, q_e)-mKDM-sID-CPA secure then $\Gamma_2(q_e)$ is also (n, q_e)-mKDM-sID-CPA secure. In particular,*

$$\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma_2(q_e), \lambda, \ell, n, q_e; t) \le 2\left\lceil 1.71\log_2\ell\right\rceil \mathbf{AdvDLin}(\mathcal{G}; t') + 2n\cdot 2^{-\lambda} +$$

$$+\mathbf{AdvKDM\text{-}sID\text{-}CPA}(\Gamma_1(q_e), \lambda, \ell, n, q_e; t'')$$

[8] $\mathrm{rank}(\mathbf{L}_k\mathbf{H}\boldsymbol{\Psi}^{(k)}) = 2$ if and only if $\mathrm{Span}(\mathbf{L}_k^\top) \cap \ker(\mathbf{H}^\top) = \{\mathbf{0}\}$ and $\mathrm{Span}(\mathbf{H}^\top) \cap \ker((\boldsymbol{\Psi}^{(k)})^\top) = \{\mathbf{0}\}$, which are independent outcomes. By a counting argument, the probability of the first outcome is $\frac{q^{2\ell-2}(q^2-1)(q^2-q)}{(q^{\ell+1}-1)(q^{\ell+1}-q)} > 1 - q^{-1} - q^{-2}$, while the second is $\frac{q^{\ell+1}-q^2}{q^{\ell+1}-1} > 1 - q^{-(\ell-1)}$. Thus, if $\ell \ge 4$ the product is greater than $1 - \frac{1}{q-1}$.

Proof. The proof is also structured as a sequence of games played by a challenger $\mathcal{C}_2$ and an (n, q_e)-mKDM-sID-CPA adversary $\mathcal{A}_2$ against $\Gamma_2(q_e)$. Games 0-b, $b \in \{0,1\}$, are exactly the two (n, q_e)-mKDM-sID-CPA security games defined in section 3.1. Let us denote as $\Omega_{i,b}$ the event that $\mathcal{A}_2$ outputs $b' = 1$ in Game i-b, and let $\mathbf{Adv}_i = |\Pr[\Omega_{i,0}] - \Pr[\Omega_{i,1}]|$. We will show in the last game a (n, q_e)-mKDM-sID-CPA adversary $\mathcal{A}_1$ against $\Gamma_1(q_e)$ with essentially the same advantage and running time.

Game 1-b: The challenger $\mathcal{C}_2$ generates the n independent master keys $PK_1, \ldots, PK_n$ and $SK_1, \ldots, SK_n$ in a different but almost equivalent way. $\mathcal{C}_2$ generates n independent master public and secret keys for $\Gamma_1(q_e)$ instead of $\Gamma_2(q_e)$. Then, it picks some random matrices $\mathbf{H} \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times\ell;2}$ and $\mathbf{R}_1, \ldots, \mathbf{R}_n \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times 2;2}$, and computes the public keys for Γ_2 by just replacing the elements $g^{\widetilde{\mathbf{S}}^{(k)}}$ and $g^{\mathbf{S}_j^{(k)}}$ respectively with $g^{\widetilde{\mathbf{S}}^{(k)}\mathbf{HR}_k}$ and $g^{\mathbf{S}_j^{(k)}\mathbf{HR}_k}$, for $j = 0, \ldots, q_e$. Observe that $\mathcal{C}_2$ needs to know $g^{\widetilde{\mathbf{S}}^{(k)}}$ and all the coefficient matrices $g^{\mathbf{S}_j^{(k)}}$ (which it learns from the key generation of $\Gamma_1(q_e)$). Encryption queries are handled as in Game 0-b, but the way user keys are computed is changed. Namely, user key $(g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$ for identity id and master public key PK_k is computed as $g^{\boldsymbol{d}_1} = g^{\boldsymbol{x}^{(k)}} \cdot g^{\left(\sum_{j=0}^{q_e} \mathbf{S}_j^{(k)} id^j\right)\mathbf{H}\boldsymbol{t}'}$ and $g^{\boldsymbol{d}_2} = g^{\widetilde{\mathbf{S}}^{(k)}\mathbf{H}\boldsymbol{t}'}$, where now $\boldsymbol{t}' \in_{\mathrm{R}} \mathbb{Z}_q^\ell$. Still $\mathcal{C}_2$ needs the master secret keys $g_T^{\boldsymbol{x}^{(k)}}$.

Let us see that with overwhelming probability the generated keys are valid keys for $\Gamma_2(q_e)$ and with the same distribution as in Game 0-b. Actually, it suffices to check whether $g^{\widetilde{\mathbf{S}}^{(k)}\mathbf{HR}_k}$ has the right distribution, since the other elements in the master public keys are generated in an equivalent way as in Game 0-b. Indeed, for every $j = 0, \ldots, q_e - 1$ one can define $\mathbf{T}_j{}^{(k)} = \mathbf{S}_j{}^{(k)}(\widetilde{\mathbf{S}}^{(k)})^{-1} \in {\mathbb{Z}_q}^{\ell\times\ell}$, and $\mathbf{T}_{q_e}{}^{(k)} = \mathbf{S}_{q_e}{}^{(k)}(\widetilde{\mathbf{S}}^{(k)})^{-1} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$, which are independent and uniformly distributed. Therefore, $\widetilde{\boldsymbol{\Psi}}_j{}^{(k)} = \boldsymbol{\Psi}^{(k)}\mathbf{T}_j{}^{(k)}$ and $\mathbf{S}_j^{(k)}\mathbf{HR}_k = \mathbf{T}_j^{(k)}\widetilde{\mathbf{S}}^{(k)}\mathbf{HR}_k$, for all j.

Observe that $\mathrm{rank}(\widetilde{\mathbf{S}}^{(k)}\mathbf{HR}_k) = 2$ with probability at least $1 - 1/(q-1)$,[9] and the randomness in $\mathbf{R}_k$ and $\widetilde{\mathbf{S}}^{(k)}$ guarantees the uniformity of the resulting matrix in ${\mathbb{Z}_q}^{\ell\times 2;2}$, and its independence of the other elements in the master public keys. Indeed, from Lemma 3 $\mathbf{R}_k$ could be computed from a fixed $\mathbf{R} \in {\mathbb{Z}_q}^{\ell\times 2;2}$ as $\mathbf{R}_k = \mathbf{U}_k\mathbf{R}\mathbf{V}_k$, where $\mathbf{U}_k \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$ and $\mathbf{V}_k \in_{\mathrm{R}} \mathrm{GL}_2(\mathbb{Z}_q)$. And similarly $\widetilde{\mathbf{S}}^{(k)} = \mathbf{W}_k\widetilde{\mathbf{S}}$, where $\widetilde{\mathbf{S}}, \mathbf{W}_k \in_{\mathrm{R}} \mathrm{GL}_\ell(\mathbb{Z}_q)$. Therefore, for each choice of $\mathbf{U}_k$, $\mathbf{R}$, $\mathbf{H}$, and $\widetilde{\mathbf{S}}$ such that $\mathrm{rank}(\widetilde{\mathbf{S}}\mathbf{H}\mathbf{U}_k\mathbf{R}) = 2$, due to the randomness in $\mathbf{V}_k$ and $\mathbf{W}_k$, $\widetilde{\mathbf{S}}^{(k)}\mathbf{HR}_k$ is uniformly distributed in ${\mathbb{Z}_q}^{\ell\times 2;2}$.

Furthermore, $\mathrm{rank}(\mathbf{HR}_k) = \mathrm{rank}(\mathbf{H}) = 2$ implies $\mathrm{Span}(\mathbf{HR}_k) = \mathrm{Span}(\mathbf{H})$. Therefore, the probability distributions of $\mathbf{HR}_k\boldsymbol{t}$ for $\boldsymbol{t} \in_{\mathrm{R}} \mathbb{Z}_q^2$ and $\mathbf{H}\boldsymbol{t}'$ for $\boldsymbol{t}' \in_{\mathrm{R}} \mathbb{Z}_q^\ell$ are the same, and the simulation of the user key generation oracle is perfect (assuming $\mathrm{rank}(\widetilde{\mathbf{S}}^{(k)}\mathbf{HR}_k) = 2$ as above). Then, $|\mathbf{Adv}_0 - \mathbf{Adv}_1| \leq 2n/(q-1)$.

Game 2-b: The same as Game 1-b but now $\mathbf{H} \in \mathrm{GL}_\ell(\mathbb{Z}_q)$. From Games 1-$b$ and 2-b one can build a distinguisher for the $\mathbf{Rank}(\mathcal{G}, \ell, \ell, 2, \ell)$ problem. Therefore,

$$|\mathbf{Adv}_1 - \mathbf{Adv}_2| \leq 2\left\lceil 1.71 \log_2 \ell \right\rceil \mathbf{AdvDLin}(\mathcal{G})$$

Game 3-b: The same as Game 2-b but now $\mathbf{H} = \mathbf{I}_\ell$ (that is, $\mathbf{H}$ is removed). However, Games 2-b and 3-b are identical, since the distribution of keys and ciphertexts remain unchanged. Indeed, the public key elements are now computed as $g^{\widetilde{\mathbf{S}}^{(k)}\mathbf{R}_k}$ and $g^{\mathbf{F}^{(k)}\mathbf{R}_k}$, and user secret keys $(g^{\boldsymbol{d}_1}, g^{\boldsymbol{d}_2})$ are computed exactly as in $\Gamma_1(q_e)$; namely, $g^{\boldsymbol{d}_1} = g^{\boldsymbol{x}^{(k)}} \cdot g^{\mathbf{F}^{(k)}(id)\boldsymbol{t}'}$ and $g^{\boldsymbol{d}_2} = g^{\widetilde{\mathbf{S}}^{(k)}\boldsymbol{t}'}$, with $\boldsymbol{t}' \in_{\mathrm{R}} \mathbb{Z}_q^\ell$. Thus, stepping from Game 2-b to Game 3-b is just replacing $\mathbf{HR}_k$ by $\mathbf{R}_k$ and $\mathbf{H}\boldsymbol{t}'$ by $\boldsymbol{t}'$. Therefore, $\mathbf{Adv}_2 = \mathbf{Adv}_3$.

Game 4-b: In this game, a challenger $\mathcal{C}_1$ plays with an adversary $\mathcal{A}_1$ against the scheme $\Gamma_1(q_e)$, which uses $\mathcal{A}_2$ as a subroutine. Basically, $\mathcal{A}_1$ picks random matrices $\mathbf{R}_1, \ldots, \mathbf{R}_k \in_{\mathrm{R}} {\mathbb{Z}_q}^{\ell\times 2;2}$ and forwards all the messages in either direction, except that the master public key elements $g^{\widetilde{\mathbf{S}}^{(k)}}$ and $g^{\mathbf{F}^{(k)}(\cdot)}$ are respectively replaced with

[9] $\mathrm{rank}(\widetilde{\mathbf{S}}^{(k)}\mathbf{HR}_k) = \mathrm{rank}(\mathbf{HR}_k)$ since $\widetilde{\mathbf{S}}^{(k)}$ is invertible, and $\mathrm{rank}(\mathbf{HR}_k) = 2$ if and only if $\mathrm{Span}(\mathbf{R}_k) \cap \ker(\mathbf{H}) = \{\mathbf{0}\}$. By a counting argument, the probability of that outcome is $\frac{q^{2\ell-4}(q^2-1)(q^2-q)}{(q^\ell-1)(q^\ell-q)} > 1 - q^{-1} - q^{-2} > 1 - \frac{1}{q-1}$.

$g^{\widetilde{\mathbf{S}}^{(k)}\mathbf{R}_k}$ and $g^{\mathbf{F}^{(k)}(\cdot)\mathbf{R}_k}$. Notice that $\mathcal{A}_1$ does not use any secret key, and the user key and encryption queries by $\mathcal{A}_2$ are directly answered by the oracles given to $\mathcal{A}_1$. Therefore,

$$\mathbf{Adv}_4 = \mathbf{AdvKDM\text{-}sID\text{-}CPA}_{\mathcal{A}_1}^{\Gamma_1(q_e)}(\lambda, \ell, n, q_e)$$

Since Games 3-b and 4-b are also identical, $\mathbf{Adv}_3 = \mathbf{Adv}_4$. and

$$\mathbf{AdvKDM\text{-}sID\text{-}CPA}_{\mathcal{A}_2}^{\Gamma_2(q_e)}(\lambda, \ell, n, q_e) = \mathbf{Adv}_0 \leq 2\left\lceil 1.71 \log_2 \ell \right\rceil \mathbf{AdvDLin}(\mathcal{G}) + 2n/(q-1) + \mathbf{Adv}\mathcal{A}_1 \leq$$

$$= 2\left\lceil 1.71 \log_2 \ell \right\rceil \mathbf{AdvDLin}(\mathcal{G}) + 2n \cdot 2^{-\lambda} + \mathbf{AdvKDM\text{-}sID\text{-}CPA}_{\mathcal{A}_1(q_e)}^{\Gamma_1}(\lambda, \ell, n, q_e)$$

□

5 Open problems

We enumerate some problems for future work. Probably the most prominent is to build mKDM-sID-CPA secure IBE schemes where the master public key and ciphertext sizes do not depend on the number of challenge queries q_e. Alternatively, user-key dependent chosen plaintext secure IBE schemes where the master public key, ciphertext and user key sizes do not depend on the number of challenge queries n is also an interesting research direction [2].

Finally finding an efficient IND-CCA encryption scheme under a static assumption whose security reduction does not depend on the number of challenge queries remains an open question.

References

1. Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.
2. Jacob Alperin-Sheriff and Chris Peikert. Circular and kdm security for identity-based encryption. In *PKC 2012*. To appear, available at www.cc.gatech.edu/grads/j/jmas6/pubs/kdm-ibe.pdf.
3. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *LNCS*, pages 595–618. Springer, 2009.
4. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *LNCS*, pages 259–274. Springer, 2000.
5. John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *LNCS*, pages 62–75. Springer, 2002.
6. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In Cachin and Camenisch [12], pages 223–238.
7. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
8. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.
9. Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In David Wagner, editor, *CRYPTO*, volume 5157 of *LNCS*, pages 108–125. Springer, 2008.
10. Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *LNCS*, pages 573–592. Springer, 2006.
11. Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Tal Rabin, editor, *CRYPTO*, volume 6223 of *LNCS*, pages 1–20. Springer, 2010.
12. Christian Cachin and Jan Camenisch, editors. *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *LNCS*. Springer, 2004.
13. Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *LNCS*, pages 351–368. Springer, 2009.
14. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *LNCS*, pages 93–118. Springer, 2001.

15. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *LNCS*, pages 255–271. Springer, 2003.
16. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Cachin and Camenisch [12], pages 207–222.
17. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *STOC*, pages 542–552. ACM, 1991.
18. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
19. Matthew Green and Susan Hohenberger. Practical adaptive oblivious transfer from simple assumptions. In Yuval Ishai, editor, *TCC*, volume 6597 of *LNCS*, pages 347–363. Springer, 2011.
20. Dennis Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. Cryptology ePrint Archive, Report 2012/150, 2012. http://eprint.iacr.org/.
21. Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *LNCS*, pages 108–126. Springer, 2008.
22. Fabien Laguillaumie, Pascal Paillier, and Damien Vergnaud. Universally convertible directed signatures. In Bimal K. Roy, editor, *ASIACRYPT*, volume 3788 of *LNCS*, pages 682–701. Springer, 2005.
23. Tal Malkin, Isamu Teranishi, and Moti Yung. Efficient circuit-size independent public key encryption with kdm security. In Kenneth G. Paterson, editor, *EUROCRYPT*, volume 6632 of *LNCS*, pages 507–526. Springer, 2011.
24. Alfred Menezes and Nigel Smart. Security of signature schemes in a multi-user setting. *Des. Codes Cryptography*, 33:261–274, November 2004.
25. Payman Mohassel. One-time signatures and chameleon hash functions. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *LNCS*, pages 302–319. Springer, 2010.
26. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *LNCS*, pages 18–35. Springer, 2009.
27. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *LNCS*, pages 433–444. Springer, 1991.
28. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.