Public-Key Encryption with Simulation-Based Selective-Opening Security and Compact Ciphertexts

Dennis Hofheinz[1]*, Tibor Jager[2], and Andy Rupp[1]

[1] Karlsruhe Institute of Technology, Germany
{dennis.hofheinz,andy.rupp}@kit.edu
[2] Ruhr-University Bochum, Germany
tibor.jager@rub.de

Abstract. In a selective-opening (SO) attack on an encryption scheme, an adversary A gets a number of ciphertexts (with possibly related plaintexts), and can then adaptively select a subset of those ciphertexts. The selected ciphertexts are then opened for A (which means that A gets to see the plaintexts and the corresponding encryption random coins), and A tries to break the security of the unopened ciphertexts.

Two main flavors of SO security notions exist: indistinguishability-based (IND-SO) and simulation-based (SIM-SO) ones. Whereas IND-SO security allows for simple and efficient instantiations, its usefulness in larger constructions is somewhat limited, since it is restricted to special types of plaintext distributions. On the other hand, SIM-SO security does not suffer from this restriction, but turns out to be significantly harder to achieve. In fact, all known SIM-SO secure encryption schemes either require $\mathbf{O}(|m|)$ group elements in the ciphertext to encrypt $|m|$-bit plaintexts, or use specific algebraic properties available in the DCR setting.

In this work, we present the first SIM-SO secure PKE schemes in the discrete-log setting with compact ciphertexts (whose size is $\mathbf{O}(1)$ group elements plus plaintext size). The SIM-SO security of our constructions can be based on, e.g., the k-linear assumption for any k.

Technically, our schemes extend previous IND-SO secure schemes by the property that simulated ciphertexts can be *efficiently* opened to arbitrary plaintexts. We do so by encrypting the plaintext in a bitwise fashion, but such that each encrypted bit leads only to a single ciphertext bit (plus $\mathbf{O}(1)$ group elements that can be shared across many bit encryptions). Our approach leads to rather large public keys (of $\mathbf{O}(|m|^2)$ group elements), but we also show how this public key size can be reduced (to $\mathbf{O}(|m|)$ group elements) in pairing-friendly groups.

Keywords: Public-key encryption, selective-opening security, lossy encryption, matrix assumptions.

* Supported by DFG grants GZ HO 4534-2/2 and 4534/4-1.

1 Introduction

Selective-opening (SO) attacks. A selective-opening (SO) attack on an encryption scheme models the adaptive corruption of multiple senders. More formally, an SO adversary A first receives many ciphertexts $c_1, \ldots, c_n$ for respective plaintexts $m_1, \ldots, m_n$ that are jointly sampled (and may thus be related). A may then ask for the opening of an arbitrary subset of the c_i.[3] Finally, A is asked to break the security of the unopened ciphertexts.

Different flavors of SO security notions. Note that it is not entirely clear what "breaking the security of the unopened ciphertexts" should mean. For instance, since the plaintexts are related, it is possible that *all* plaintexts (including those from unopened ciphertexts) can be efficiently computed from the opened plaintexts. Furthermore, to achieve greater generality, usually the joint *distribution* from which the $m_1, \ldots, m_n$ are sampled is adversarially chosen, so A may already have some a-priori (partial or even full) knowledge about the unopened m_i.

Hence, two different flavors of SO security have developed: simulation-based (SIM-SO [10, 2]) and indistinguishability-based (IND-SO [2, 7]) security. Intuitively, SIM-SO security requires that the output of A above can be simulated by a simulator that sees only the opened m_i (and no ciphertexts at all). In particular, all information A can extract about the unopened m_i can also be generated by a simulator from the opened m_i alone.

On the other hand, IND-SO security requires that the unopened plaintexts look indistinguishable from independently sampled plaintexts. Because the plaintexts may be related, this independent sampling must be conditioned on the already opened plaintexts to avoid trivial attacks. Hence, if, e.g., the opened plaintexts already fully determine all plaintexts, conditional sampling will lead to the originally encrypted plaintexts, and IND-SO security is trivially achieved.

As a consequence, the IND-SO experiment itself is only efficient for plaintext distributions that are "efficiently (conditionally) re-samplable" in the above sense. In fact, usually IND-SO security is only considered for such plaintext distributions [2, 18, 19], which limits its applicability to scenarios with such distributions; there is no known encryption scheme that is IND-SO secure against arbitrary (i.e., only efficiently *samplable*) plaintext distributions.

The difficulty of achieving simulation-based SO security. Hence, from an application point of view, SIM-SO security is the preferable notion of SO security. Unfortunately, while IND-SO security (restricted to efficiently re-samplable plaintext distributions and in the chosen-plaintext case) is already achieved by any lossy encryption scheme [2, 29], SIM-SO security seems much harder to obtain. For instance, [1] show (under mild computational assumptions) that there are encryption schemes that are IND-CPA but not SIM-SO secure. Furthermore, known constructions of SIM-SO secure encryption schemes follow dedicated (and

[3] In this paper, we consider sender corruptions, in which case the opening of a c_i consists of the plaintext m_i and the encryption random coins used to construct c_i.

somewhat nonstandard) design strategies [2, 12, 3, 18, 19, 22, 15]. As a result, all known SIM-SO secure schemes fall into one of the following two categories:

Large ciphertexts. The SIM-SO secure schemes from [2, 12, 3, 22] have ciphertexts of $\mathbf{O}(|m|)$ group elements, where $|m|$ is the bitsize of the plaintext.

DCR-based. The schemes from [18, 19, 15][4] have more compact ciphertexts, but are limited to the decisional composite residuosity (DCR) setting [28, 9] (and rely on its specific algebraic features).

Below, when explaining our technical approach, we will also comment on the technical obstacles that need to be overcome for SIM-SO security.

Our results. In this work, we offer the first SIM-SO secure encryption schemes with compact ciphertexts in the discrete-log setting. Specifically, ciphertexts in our scheme carry $\mathbf{O}(1)$ group elements (plus $|m|$ bits, where $|m|$ is the plaintext bitsize), and SIM-SO security can be proved under any matrix assumption [11] (thus, in particular under, e.g., the k-linear assumption for any $k \geq 1$). Our construction is simple, works in the standard model, and does not require pairings.

The price we pay for these features is a rather large public key size (of $\mathbf{O}(|m|^2)$ group elements, and computationally expensive encryption and decryption procedures. Specifically, our encryption proceeds bitwise, and requires $\mathbf{O}(|m|)$ exponentiations for each message bit. (Alternatively, the operation needed to encrypt one bit could also be viewed as one multi-exponentiation with respect to $\mathbf{O}(|m|)$ fixed bases. So there is room for some small improvements in runtime by a constant factor, e.g., using interleaving multi-exponentiation [23].) Concerning the key size, we show how a technique of [6] can be used to at least compress the public key to $\mathbf{O}(|m|)$ group elements by using a pairing. Still, in particular in light of the relatively inefficient encryption and decryption in our scheme, we view our result mainly as a feasibility result.

In the following, we give a brief overview over our approach.

Our starting point. Our starting point is the lossy (and thus IND-SO secure) PKE scheme of [25] (see also [17, 29, 2]). In this scheme, public keys and ciphertexts are of the form

$$pk = (g, g^x, g^y, g^z) \qquad c = (u, v) = (g^{r+sx}, g^{ry+sz} \cdot m) \tag{1}$$

for random exponents x, y, r, s, for $z = xy$, and a plaintext m. Note that if we switch z to an independently random value (however with $z \neq xy$), then encryption becomes lossy: ciphertexts are tuples of random group elements, independently of m. Furthermore, such a switch can be justified with the decisional Diffie-Hellman (DDH) assumption.

Efficient openability. In order to achieve SIM-SO security, we additionally require a property called “efficient openability” of ciphertexts [2, 12]. In a nutshell, efficient openability requires that ciphertexts generated under lossy public keys can be opened to arbitrary messages with a special trapdoor. (Note that such an arbitrary opening is always possible inefficiently in the lossy case.)

[4] In fact, [18] also offers a scheme with large ciphertexts in the discrete-log setting.

Scheme	Security	Assumption	$\vert pk\vert$	$\vert m\vert$	$\vert c\vert - \vert m\vert$
BHY09 [2]	IND-SO-CPA	DDH	$2 \times \vert G\vert$	$\vert G\vert$	$\vert G\vert$
BHY09 [2]	SIM-SO-CPA	QR	$1 \times \vert N\vert$	n	$n(\vert N\vert - 1)$
BY12 [4][5]	SIM-SO-CPA	DDH	$2 \times \vert G\vert$	1	$2\vert G\vert - 1$
FHKW10 [12]	SIM-SO-CPA	TDOWP	TDOWP-pk	1	$\vert \mathrm{img}\vert - 1$
FHKW10 [12]	SIM-SO-CCA	DDH	$2 \times \vert G\vert$	poly(λ)	$2\vert m\vert\vert G\vert + \vert m\vert\lambda$
HLOV11 [18]	SIM-SO-CPA	DCR	$2 \times \vert N\vert$	$\vert N\vert$	$\vert N\vert$
Ours	SIM-SO-CPA	DDH	$(\vert m\vert + 1)^2$	poly(λ)	$1 \times \vert G\vert$
Ours	SIM-SO-CPA	DLIN	$(\vert m\vert + 2)^2$	poly(λ)	$2 \times \vert G\vert$
Ours	SIM-SO-CPA	k-linear	$(\vert m\vert + k)^2$	poly(λ)	$k \times \vert G\vert$
Ours	SIM-SO-CPA	BDDH	$\vert m\vert \cdot (4\vert G\vert + \vert G_T\vert)$	poly(λ)	$1 \times \vert G_T\vert$

Table 1: Comparison of our construction with other SO-secure PKE schemes. (We omit schemes that do not achieve SIM-SO-CPA security in any more efficient way than the ones mentioned, e.g., because they focus on CCA security [18, 19, 15] or on the IBE setting [3].) $|G|$ denotes the description (bit-)size of elements of a group in the discrete-log setting, and $|G|$ and $|G_T|$ denote the corresponding sizes in a pairing-friendly setting with source group G and target group G_T. λ denotes the security parameter. The entry poly(λ) in the $|m|$ column means that the message size is not restricted and might be set arbitrarily (and especially independent of the group size). QR denotes the quadratic residuosity assumption, DCR denotes Paillier's decisional composite residuosity assumption, and $|N|$ denotes the length of a suitable composite number (determining the modulus) for such schemes. TDOWP denotes an arbitrary trapdoor one-way permutation, and $|\mathrm{img}|$ denotes the (bit-)size of elements in the corresponding image. $|c| - |m|$ denotes the ciphertext overhead (i.e., the bitlength of the ciphertext minus the plaintext bitlength).

We note that efficient openability implies SIM-SO security [2]. In fact, all mentioned SIM-SO secure schemes achieve (a suitable variant of) efficient openability.[6] Unfortunately, this strong property is not achieved easily. For instance, consider the PKE scheme from (1) (with lossy public keys, i.e., with $z \neq xy$). In order to open a given ciphertext $c = (u, v)$ as an encryption of an externally given plaintext m, a simulator would have to supply random coins (r, s) satisfying $r + sx = \mathrm{dlog}_g(u)$ and $ry + sz = \mathrm{dlog}_g(v) - \mathrm{dlog}_g(m)$. Hence, the ability to open to arbitrary m implies the ability to compute discrete logarithms (which would seem to require special trapdoors in standard discrete-log groups).[7]

A bitwise scheme. Our first observation is that the situation changes if only bits (or messages from a small domain) are encrypted. Concretely, consider the following slightly modified scheme that encrypts only bits:

$$pk = (g, g^x, g^y, g^z) \qquad c = (u, v) = (g^{r+sx}, H(g^{ry+sz}) \oplus m) \tag{2}$$

where x, y, z, r, s are as before, H is a universal hash function that maps group elements to bits, and $m \in \{0, 1\}$. This scheme allows for an efficient opening

[6] However, it should also be noted that neither efficient openability nor lossiness (in the sense of [29, 2]) may be necessary for SIM-SO security (see [27] for the lossiness case). Still, our construction is easiest to explain by following this path.

[7] One reason why the DCR settings seems much more suitable for SO security is that certain DCR subgroups allow to easily compute discrete logarithms. Put differently: in DCR-based encryption schemes [28, 9, 18], both plaintexts and encryption random coins are exponents. Hence, encryption random coins can be computed from plaintexts (as required for a SIM-SO simulation) much more easily.

operation (if $z \neq xy$). Namely, to open a ciphertext $c = (u, v)$ (as in (2)) to a message m, using as trapdoor x, y, z, r, s, simply sample r', s' randomly subject to $r' + s'x = r + sx$ until $H(g^{r'y+s'z}) \oplus m = v$. (On average, it takes 2 such samplings until suitable r', s' are found.) This scheme can be generalized to messages $m \in \{0,1\}^{\mathbf{O}(\log(\lambda))}$ (where λ denotes the security parameter), using hash functions with output length $|m|$, at the cost of a less efficient opening algorithm. In the following, however, we will focus on the bitwise processing of messages for simplicitly.

The scheme from (2) hence achieves efficient openability (and thus SIM-SO security), but suffers from a small message space. Of course, its message space can be expanded by concatenating several ciphertexts, which would however increase the ciphertext size to $\mathbf{O}(|m|)$ group elements.

Compressing ciphertexts. Hence, instead of concatenating ciphertexts, we reuse the value of u across several bit encryptions. Doing so naively (e.g., by setting $u = g^{r+sx}$ and $v_i = H(g_i^{ry+sz}) \oplus m_i$ for different generators g_i) would however interfere with our efficient opening strategy. Specifically, it is not obvious how to efficiently sample r', s' as above that would lead to $H(g_i^{r'y+s'z}) \oplus m_i = v_i$ for all i simultaneously.

We resolve this issue by adding more encryption random coins (and thus more "degrees of freedom" for our efficient opening procedure). That is, we set

$$\begin{aligned} pk &= (g, (g^{x_i}, g^{y_j})_{j=1}^{\mu}, (g^{z_{i,j}})_{i,j=1}^{\mu}) \\ c &= (u, (v_i)_{i=1}^{\mu}) = (g^{r+\sum_{j=1}^{\mu} s_j x_j}, (H(g^{ry_j+\sum_{j=1}^{\mu} s_j z_{i,j}}) \oplus m_i)_{i=1}^{\mu}) \end{aligned} \tag{3}$$

for random exponents $x_i, y_j, z_{i,j}, r, s_j$ and $z_{i,j} = x_i y_j$, and an μ-bit plaintext $m = (m_i)_{i=1}^{\mu}$. Since $z_{i,j} = x_i y_j$, knowledge of all x_i, y_j allows to decrypt. However, switching to random $z_{i,j} \neq x_i y_j$ (which can be justified with the DDH assumption) implies that encryption becomes lossy (as with (1) and (2)).

Moreover, in case $z_{i,j} \neq x_i y_j$, a ciphertext $c = (u, (v_i)_{i=1}^{\mu})$ can be efficiently opened as follows. First, select "target exponents" $t_1, \ldots, t_\mu$ randomly subject to $H(g^{t_i}) \oplus m_i = v_i$ for all i. (The t_i can be sampled individually, one after the other, and so this step requires 2μ samplings on average.) Next, solve the system that consists of the linear equations $r'y_j + \sum_{j=1}^{\mu} s'_j z_{i,j} = t_i$ (with $1 \leq i \leq \mu$) and $r' + \sum_{j=1}^{\mu} s'_j x_j = r + \sum_{j=1}^{\mu} s_j x_j$ for the variables r', s'_j. (Since the $z_{i,j} \neq x_i y_j$ are random, this system is solvable using linear algebra with high probability.) Finally, output $(r', (s'_j)_{j=1}^{\mu})$ as the desired random coins that open c to m.

Extensions and open problems. Inside, we also show how to generalize this idea to weaker assumptions than DDH (in the same spirit in which [14] generalize the DDH-based lossy trapdoor function of [30]). In particular, we obtain constructions based on any Matrix Diffie-Hellman (MDDH) assumption [11] (at the price of somewhat larger ciphertexts, but whose overhead is still independent of $|m|$, and somewhat larger public keys), including the k-linear assumption [20, 31]. Furthermore, we show how to compress the public key of our scheme from $\mathbf{O}(|m|^2)$ to $\mathbf{O}(|m|)$ group elements using a pairing-based technique used to compress the public key of lossy trapdoor functions [6].

In this work, we focus on chosen-plaintext (CPA) security. One interesting open problem is to extend our techniques to the chosen-ciphertext (CCA) setting to obtain a SIM-SO-CCA secure scheme with compact ciphertexts in the discrete-log regime. Besides, of course a further compression of the public key in our schemes or an improvement in computational efficiency would be desirable.

Relation to a scheme of Bellare and Yilek. Our "bitwise" scheme from (2) above is very similar to a scheme of Bellare and Yilek (from Section 5.4 of the September 23, 2012 update of [4]). (We thank one TCC reviewer for pointing us to that scheme, which we were not aware of previously.) The main difference is that we use the use the term $H(g^{ry+sz}) \oplus m$ to hide the message, whereas Bellare and Yilek use $g^{ry+sz}) \cdot g^m$. This entails (conceptually not very significant) differences in the respective opening algorithms. However, the more important difference in these schemes is that our scheme from (2) only has one group element (plus one hidden message bit) in the ciphertext, while Bellare and Yilek use a whole group element to hide a one-bit message. Hence, our main trick above (namely, to modify and then reuse the first ciphertext element g^{ry+sz} for many bit encryptions) would not lead to compact ciphertexts when applied to the scheme of Bellare and Yilek.

SO security against corrupted receivers, and relation to non-committing encryption. Traditionally, SO security models a setting in which only senders are corrupted (and thus, an opening only reveals the corresponding encryption random coins). However, some works (e.g., [1, 21]) *additionally* consider SO security against corrupted receivers (in which case there are many public keys, and an opening consists of the respective secret key). In this setting, strong impossibility results hold [1], which provide a fixed upper limit the number of secure encryptions under any given public key. The arising technical problems are *commitment problems*, and are very related to the inherent problems of non-committing encryption (NCE, [8]). Indeed, NCE schemes can be seen as encryption schemes that are SO secure both against corrupted senders and corrupted receivers.

In contrast, the more commonly considered notion of SO security against corrupted senders (which we also consider here) allows for more efficient schemes, that in particular tolerate an arbitrary number of encryptions and corruptions. The price to pay here is of course that only corruptions of senders (but not of receivers) are considered.

Roadmap. After fixing some notation and basic definitions in Section 2, we introduce our construction of lossy encryption with efficient weak opening in Section 3. The construction is generic and relies on what we call a matrix rank assumption. In Section 4, we then instantiate those assumptions with the family of MDDH assumptions from [11] (and thus in particular with the k-linear assumption). Finally, Section 5 presents a matrix rank assumption with a linear-size representation which is implied by the BDDH assumption in pairing groups. This results in a scheme with a public key size that is linear in $|m|$.

2 Preliminaries

Notation. Throughout the paper, $\lambda \in \mathbb{N}$ denotes the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For a probabilistic algorithm A, we denote with $\mathcal{R}_A$ the space of A's random coins. $y \leftarrow A(x; R)$ denotes the process of running A on input x and with randomness $R \leftarrow \mathcal{R}_A$, and assigning y the result. We write $y \leftarrow A(x)$ for $y \leftarrow A(x; R)$ with uniform R. If A's running time is polynomial in λ, then A is called probabilistic polynomial-time (PPT). We call a positive function η negligible if for every polynomial p there exists λ_0 such that for all $\lambda \geq \lambda_0$ holds $\eta(\lambda) \leq \frac{1}{p(\lambda)}$. We call η overwhelming if $\eta(\lambda) \geq 1 - \nu(\lambda)$, where ν is a negligible function. The statistical distance between two random variables X and Y over a finite common domain D is defined by $\Delta(X, Y) = \frac{1}{2} \sum_{z \in D} |\Pr[X = z] - \Pr[Y = z]|$. We say that two families $X = (X_\lambda)_{\lambda \in \mathbb{N}}$ and $Y = (Y_\lambda)_{\lambda \in \mathbb{N}}$ of random variables are statistically close or statistically indistinguishable, denoted by $X \approx_s Y$, if $\Delta(X_\lambda, Y_\lambda)$ is negligible in λ.

2.1 Groups and Matrix Assumptions

Prime-order k-linear group generators. We use the following formal definition of a k-linear prime-order group generator for our constructions.

Remark 1. We stress that *our constructions do not require multilinear maps* in the sense of [16]. We rather want to capture both single-group settings and bilinear group settings in one unified definition, because this will be helpful in the sequel for the exposition of results that apply to both settings. Hence, one should have $k = 1$ or $k = 2$ in mind in the following definition.

Definition 1. *A* prime-order k-linear group generator *is a PPT algorithm $\mathcal{G}_k$ that on input of a security parameter 1^λ outputs a tuple of the form*

$$\mathcal{MG}_k := (k, G_1, \ldots, G_k, G_{k+1}, g_1, \ldots, g_k, e, p) \leftarrow \mathcal{G}_k(1^\lambda)$$

where $G_1, \ldots, G_{k+1}$ are descriptions of cyclic groups of prime order p, $\log p = \Theta(\lambda)$, g_i is a generator of G_i for $1 \leq i \leq k$, and $e \colon G_1 \times \ldots \times G_k \to G_{k+1}$ is a map which satisfies the following properties:

- k-linearity: *For all $a_1 \in G_1, \ldots, a_k \in G_k$, $\alpha \in \mathbb{Z}_p$, and $i \in \{1, \ldots, k\}$ we have $e(a_1, \ldots, a_{i-1}, \alpha a_i, a_{i+1}, \ldots, a_k) = \alpha e(a_1, \ldots, a_k)$.*
- Non-degeneracy: *$g_{k+1} := e(g_1, \ldots, g_k)$ generates G_{k+1}.*

If $G_1 = \ldots = G_k$, we call $\mathcal{G}_k$ a symmetric *k-linear group generator.*

Note that Definition 1 captures both ordinary single group generators and symmetric bilinear group generators:

- In the single-group setting, $\mathcal{G}_1(1^\lambda)$ would output $\mathcal{MG}_1 := (1, G_1, G_2, g_1, e, p)$, where $G_1 = G_2$ and $e : G_1 \to G_2$ is the identity mapping.

- In the symmetric bilinear group setting, $\mathcal{G}_2(1^\lambda)$ would output $\mathcal{MG}_2 := (1, G_1, G_2, G_3, g_1, g_2, e, p)$, where $G_1 = G_2$ and $g_1 = g_2$ and $e : G_1 \times G_2 \to G_3$ is a pairing.

Implicit Representation. Following [11], we introduce the notion of implicit representations. Let G_i be a cyclic group of order p generated by g_i. Then by $[a]_i := g_i^a$ we denote the *implicit representation* of $a \in \mathbb{Z}_p$ in G_i. More generally, we also define such representations for vectors $\vec{b} \in \mathbb{Z}_p^n$ by $[\vec{b}]_i := ([b_j]_i)_j \in G_i^n$ and for matrices $\mathbf{A} = (a_{j,k})_{j,k} \in \mathbb{Z}_p^{n \times \ell}$ by $[\mathbf{A}]_i := ([a_{j,k}]_i)_{j,k} \in G_i^{n \times \ell}$.

Matrix-vector operations in implicit representation. If a matrix $[\mathbf{A}] = [(a_{i,j})_{i,j}] \in G^{n \times \ell}$ is known "in the exponent", and a vector $\vec{u} = (u_i)_i \in \mathbb{Z}_p^\ell$ is known "in clear", then the product $[\mathbf{A} \cdot \vec{u}] \in G^n$ can be efficiently computed as $[(v_i)_i]$ for $[v_i] = \prod_{j=1}^{\ell} [a_{i,j}]^{u_j}$. Similarly, $[\mathbf{A} \cdot \mathbf{B}] \in G^{n \times k}$ can be computed given $[\mathbf{A}] = [(a_{i,j})_{i,j}] \in G^{n \times \ell}$ and $\mathbf{B} \in \mathbb{Z}_p^{\ell \times k}$. If only $[\mathbf{A}]_1$ and $[\mathbf{B}]_2$ are known (i.e., only "in the exponent") and a bilinear map $e : G_1 \times G_2 \to G_3$ is given, we can still compute the matrix product $[\mathbf{A} \cdot \mathbf{B}]_3$ in the target group G_3, as $[(c_{i,j})_{i,j}]_3$ for $[c_{i,j}]_3 = \prod_{t=1}^{\ell} e([a_{i,t}]_1, [b_{t,j}]_2)$.

Matrix distributions and MDDH assumptions. For instantiating our construction we will make use of matrix distributions and the Matrix Diffie-Hellman assumption family as introduced in [11].

Let $n, \ell \in \mathbb{N}$, $n > \ell$. We call $\mathcal{D}_{n,\ell}$ a *matrix distribution* if it outputs (in probabilistic polynomial time and with overwhelming probability in $\log(p)$) matrices $\mathbf{A} \in \mathbb{Z}_p^{n \times \ell}$ of full rank ℓ. We define $\mathcal{D}_\ell := \mathcal{D}_{\ell+1,\ell}$.

Definition 2. *We say that the* $\mathcal{D}_{n,\ell}$-Matrix Diffie-Hellman assumption*, or just* $\mathcal{D}_{n,\ell}$*-MDDH assumption for short, holds in* G_i *and relative to the* k*-linear group generator* $\mathcal{G}_k$*, if for all PPT adversaries* D*, we have that*

$$\mathbf{Adv}_{\mathcal{D}_{n,\ell},\mathcal{G}_k}(D) = |\Pr[D(\mathcal{MG}_k, [\mathbf{A}]_i, [\mathbf{A}\vec{w}]_i) = 1] - \Pr[D(\mathcal{MG}_k, [\mathbf{A}]_i, [\vec{u}]_i) = 1]|$$

is negligible, where the probability is taken over the output

$$\mathcal{MG}_k = (k, G_1, \dots, G_k, G_{k+1}, g_1, \dots, g_k, e, p) \leftarrow \mathcal{G}_k(1^\lambda),$$

$\mathbf{A} \leftarrow \mathcal{D}_{n,\ell}$*,* $\vec{w} \leftarrow \mathbb{Z}_p^\ell$*,* $\vec{u} \leftarrow \mathbb{Z}_p^n$ *and the coin tosses of the adversary* D*.*

In particular, we will refer to the following examples of matrix distributions, all for $n = \ell + 1$:

$$\mathcal{SC}_\ell : \mathbf{A} = \begin{pmatrix} s & 0 & \dots & 0 & 0 \\ 1 & s & \dots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & \dots & 1 & s \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \; \mathcal{L}_\ell : \mathbf{A} = \begin{pmatrix} s_1 & 0 & 0 & \dots & 0 \\ 0 & s_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & s_\ell \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix}, \; \mathcal{U}_\ell : \mathbf{A} \leftarrow \mathbb{Z}_p^{(\ell+1) \times \ell},$$

where $s, s_i \leftarrow \mathbb{Z}_p$. The $\mathcal{SC}_\ell$ assumption, introduced in [11], is the *ℓ-symmetric cascade assumption* (ℓ-SCasc). The $\mathcal{L}_\ell$ assumption is actually the well-known *ℓ-linear assumption* (ℓ-Lin, [5]) in matrix language (DDH equals 1-Lin), and the

$\mathcal{U}_\ell$ assumption is the *ℓ-uniform assumption*. Moreover, ℓ-SCasc, ℓ-Lin, and the ℓ-uniform assumption hold in the generic group model [32] relative to a k-linear group generator if $k \leq \ell$ [11].

The circulant matrix assumption

$$\mathcal{C}_{\ell+d,\ell} : \mathbf{A} = \begin{pmatrix} s_1 & & & 0 \\ \vdots & s_1 & & \\ & \vdots & \ddots & \\ s_d & & & \\ 1 & s_d & & s_1 \\ & 1 & \ddots & \vdots \\ & & \ddots & s_d \\ 0 & & & 1 \end{pmatrix},$$

has very recently been proposed in [24] as a $\mathcal{D}_{n,\ell}$-MDDH assumption with optimal representation size among all assumptions with $n > \ell + 1$. This assumption has been shown to hold in the ℓ-linear generic group model [24]. More generally, we can also define the $\mathcal{U}_{n,\ell}$ assumption for arbitrary $n > \ell$. Note that the $\mathcal{U}_{n,\ell}$ assumption is the weakest MDDH assumption (with the worst representation size) and implied by any other $\mathcal{D}_{n,\ell}$ assumption [11]. In particular ℓ-Lin implies the ℓ-uniform assumption as shown by Freeman [13].

Bilinear Decisional Diffie-Hellman. We will make use of the bilinear decisional Diffie-Hellman (BDDH) assumption for our construction with linear-size public keys.

Definition 3. *Let $\mathcal{MG}_2 := (2, G_1, G_2, G_3, g_1, g_2, e, p) \leftarrow \mathcal{G}_2(1^\lambda)$, where $\mathcal{G}_2$ is a symmetric bilinear group generator (i.e., $G_1 = G_2$ and $g_1 = g_2$), and let $a, b, c \leftarrow \mathbb{Z}_p$, $b \leftarrow \{0,1\}$, $T_0 := abc$ and $T_1 \leftarrow \mathbb{Z}_p$. We say that the bilinear decisional Diffie-Hellman (BDDH) assumption holds relative to $\mathcal{G}_2$, if*

$$\mathsf{Adv}^{\mathsf{bddh}}_{B,\mathcal{G}_2}(1^\lambda) := \left| \begin{array}{l} \quad\Pr\left[1 \leftarrow B(1^\lambda, \mathcal{MG}_2, [(a,b,c)]_1, [T_0]_3)\right] \\ - \Pr\left[1 \leftarrow B(1^\lambda, \mathcal{MG}_2, [(a,b,c)]_1, [T_1]_3)\right] \end{array} \right|$$

is a negligible function for all PPT adversaries B.

2.2 Selective-Opening Secure Encryption

Public-Key Encryption. A public-key encryption (PKE) scheme PKE with message space $\mathcal{M}$ consists of three PPT algorithms $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$. The key generation algorithm $\mathsf{Gen}(1^\lambda)$ outputs a public key pk and a secret key sk. Encryption algorithm $\mathsf{Enc}(pk, m)$ takes pk and a message $m \in \mathcal{M}$, and outputs a ciphertext c. Decryption algorithm $\mathsf{Dec}(sk, c)$ takes sk and a ciphertext c, and outputs a message m. For correctness, we want $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$ for all $m \in \mathcal{M}$ and all $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$.

```
**Experiment** $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}cpa\text{-}real}}_{\mathsf{PKE},A,\mathcal{T},n}(\lambda)$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
  $\mathfrak{D}_{\mathsf{so}} \leftarrow A(\mathtt{dist}, pk)$
  $(m_i)_{i\in[n]} \leftarrow \mathfrak{D}_{\mathsf{so}}$
  $(R_i)_{i\in[n]} \leftarrow (\mathcal{R}_{\mathsf{Enc}})^n$
  $(c_i)_{i\in[n]} := (\mathsf{Enc}(pk, m_i; R_i))_{i\in[n]}$
  $\mathcal{I} \leftarrow A(\mathtt{sel}, (c_i)_{i\in[n]})$
  $out_A \leftarrow A(\mathtt{out}, (m_i)_{i\in\mathcal{I}}, (R_i)_{i\in\mathcal{I}})$
  return $\mathcal{T}(\mathfrak{D}_{\mathsf{so}}, (m_i)_{i\in[n]}, out_A)$
```

```
**Experiment** $\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}cpa\text{-}ideal}}_{\mathsf{PKE},S,\mathcal{T},n}(\lambda)$
  $\mathfrak{D}_{\mathsf{so}} \leftarrow S(\mathtt{dist})$
  $(m_i)_{i\in[n]} \leftarrow \mathfrak{D}_{\mathsf{so}}$
  $\mathcal{I} \leftarrow S(\mathtt{sel}, 1^{|m_i|})$
  $out_A \leftarrow S(\mathtt{out}, (m_i)_{i\in\mathcal{I}})$
  return $\mathcal{T}(\mathfrak{D}_{\mathsf{so}}, (m_i)_{i\in[n]}, out_A)$
```

Fig. 1: SIM-SO-CPA security experiments.

Simulation-based selective opening security. We use the definition of SO-security against chosen-plaintext attacks of Fehr *et al.* [12], which refines the definition of [2, 4] (by letting the adversary choose the message distribution).

Definition 4 (Simulation-based security against selective opening attacks). *For a PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, a polynomially bounded function* $n = n(\lambda) > 0$*, a function* $\mathcal{T}$ *and a stateful PPT adversary* A*, consider the experiments in Figure 1. We call* PKE SIM-SO-CPA *secure if for any PPT adversary* A *and PPT function* $\mathcal{T}$ *there is a stateful PPT simulator* S *such that*

$$\mathsf{Adv}^{\mathsf{sim\text{-}so\text{-}cpa}}_{\mathsf{PKE},A}(\lambda) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}cpa\text{-}real}}_{\mathsf{PKE},A,\mathcal{T},n}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{sim\text{-}so\text{-}cpa\text{-}ideal}}_{\mathsf{PKE},S,\mathcal{T},n}(\lambda) = 1\right] \right|$$

is negligible. As usual, we require that the distribution $\mathfrak{D}_{\mathsf{so}}$ *that* A *outputs is encoded as a circuit. Since* A *is PPT, this enforces efficient samplability of* $\mathfrak{D}_{\mathsf{so}}$*.*

2.3 Selective Opening Security from Lossy Encryption

In [2, 4], Bellare et al. show that any lossy encryption scheme where ciphertexts can be *efficiently* opened to arbitrary messages is indeed SIM-SO-CPA secure. The following definition essentially repeats the definition of lossy encryption with efficient opening from [4] with one small change: the Opener algorithm may receive an additional input, the random coins used to generate the ciphertext (that should now be opened to a different message). We call a scheme satisfying this definition, a lossy encryption scheme with efficient weak opening.

Definition 5 (Lossy encryption with efficient weak opening). *A lossy encryption scheme with efficient weak opening and message space* $\mathcal{M}$ *is a tuple of PPT algorithms* $\mathsf{LPKE} = (\mathsf{Gen}, \mathsf{LGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that*

- $\mathsf{Gen}(1^\lambda)$ *takes as input the security parameter* 1^λ *and outputs a keypair* (pk, sk)*. We call* pk *a real or injective public key.*
- $\mathsf{LGen}(1^\lambda)$ *takes as input the security parameter* 1^λ *and outputs a keypair* (pk, sk)*. We call* pk *a lossy public key.*

- $\mathsf{Enc}(pk, m)$ *takes as input a (real or lossy) public key* pk *and a message* $m \in \mathcal{M}$ *and outputs a ciphertext* c
- $\mathsf{Dec}(sk, c)$ *takes as input a secret key* sk *and a ciphertext* c *and outputs either a message* $m \in \mathcal{M}$ *or* $\perp$ *in case of a failure.*

Additionally, LPKE *needs to satisfy the following properties:*

1. Correctness for real keys: *For all* $\lambda \in \mathbb{N}$*,* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$*, messages* $m \in \mathcal{M}$*, and ciphertexts* $c \leftarrow \mathsf{Enc}(pk, m)$*, it always holds that* $m \leftarrow \mathsf{Dec}(sk, c)$*.*
2. Indistinguishability of real keys from lossy keys: *For any PPT algorithm* D *it holds that the advantage*
$$\mathsf{Adv}^{\mathsf{ind\text{-}lossy\text{-}key}}_{\mathsf{LPKE},D}(\lambda) := \left| \begin{array}{l} \quad\Pr[1 \leftarrow D(1^\lambda, pk) \mid (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)] \\ - \Pr[1 \leftarrow D(1^\lambda, pk) \mid (pk, sk) \leftarrow \mathsf{LGen}(1^\lambda)] \end{array} \right|$$
is negligible in λ*.*
3. Lossiness of encryption with lossy keys: *Let* $\lambda \in \mathbb{N}$*. For any* $(pk, sk) \leftarrow \mathsf{LGen}(1^\lambda)$ *and distinct messages* $m_0 \neq m_1 \in \mathcal{M}$*, holds that*
$$(sk, \mathsf{Enc}(pk, m_0)) \approx_s (sk, \mathsf{Enc}(pk, m_1))$$
4. Efficient weak openability: *Let* $\mathcal{R}_{\mathsf{Enc}}$ *denote the space of random coins for encryption. There exists a PPT algorithm* Opener *such that for any two messages* $m_0, m_1 \in \mathcal{M}$*, the probability that* Opener *on input of a lossy public and secret key* $(pk, sk) \leftarrow \mathsf{LGen}(1^\lambda)$*, a ciphertext* $c \leftarrow \mathsf{Enc}(pk, m_0; r')$*, where* $r' \leftarrow \mathcal{R}_{\mathsf{Enc}}$*, the corresponding random coins* r'*, and a message* m_1*, outputs uniform random coins* r *from* $\{r \in \mathcal{R}_{\mathsf{Enc}} \mid \mathsf{Enc}(pk, m_1; r) = c\}$ *is overwhelming.*

Despite our small changes with respect to the definition of lossy encryption and SIM-SO-CPA compared to the definitions in [4], the following theorem still follows from the corresponding proof in [4]: It does not matter for the proof if the message distribution is some arbitrary but fixed distribution (where we quantify over all efficiently samplable distributions) or if it is the output of the adversary after seeing the (lossy) public key. Moreover, the simulator which uses the Opener algorithm knows the encryption randomness of the (dummy) ciphertexts (that should be opened differently) as it has generated these ciphertexts itself.[8]

Theorem 1 ([2, 4]). *If* LPKE *is a lossy encryption scheme with efficient weak opening then* LPKE *is SIM-SO-CPA secure.*

3 Lossy Encryption from Matrix Rank Assumptions

First, we would like to stress that although we use k-linear group generators $\mathcal{G}_k$ in the following definitions and constructions for generality, the existence of

[8] Note that the los-ind2 adversary C in the proof of Theorem 5.2 in [4] is unbounded and thus may find the appropriate encryption randomness required for our Opener algorithm itself.

k-linear maps for $k > 2$ is not required to instantiate our constructions. For the instantiations based on MDDH assumptions (Section 4), an ordinary group generator $\mathcal{G}_1$ or bilinear group generator $\mathcal{G}_2$ can be assumed (where the pairing is not used for encryption). For the instantiation based on the BDDH assumption (Section 5), a bilinear group generator $\mathcal{G}_2$ is required where the pairing is needed in the encryption routine. Hence, for the remainder of this paper, it might be best to have $k = 1$ or $k = 2$ in mind.

In the following, we show how to build efficient lossy encryption with efficient weak opening for multiple bits from rank problems. Roughly speaking, this problem asks to distinguish a $n \times n$ matrix of rank $\ell < n$ chosen according to some (not necessarily uniform) distribution from a matrix of full rank n chosen according to some (not necessarily uniform) distribution, where both matrices are given in implicit representation. The following definition captures rank assumptions and additionally allows the considered matrices to be given in some "compressed form" (which, e.g., can be decompressed efficiently using a pairing).

Definition 6. *Let $\mathcal{MG}_k := (k, G_1, \ldots, G_{k+1}, g_1, \ldots, g_k, e, p) \leftarrow \mathcal{G}_k(1^\lambda)$ be a k-linear group generator. A (n, ℓ)-indistinguishable matrix constructor* MCon *for G_i, where $1 \leq i \leq k+1$, is a tuple* $\mathsf{MCon} = (\mathsf{SetupNFR}, \mathsf{SetupFR}, \mathsf{Constr})$ *of PPT algorithms with the following properties.*

Setup of non-full rank matrix description. $\mathsf{SetupNFR}(\mathcal{MG}_k)$ *returns a matrix $\mathbf{A} \in \mathbb{Z}_p^{n \times n}$ of rank ℓ, where we assume that $\mathbf{A}$'s first ℓ rows are linearly independent, as well as a (compact) description $mat \in \{0,1\}^*$ of the implicit representation $[\mathbf{A}]_i$ of $\mathbf{A}$.*

Setup of full rank matrix description. $\mathsf{SetupFR}(\mathcal{MG}_k)$ *returns a matrix $\mathbf{A} \in \mathbb{Z}_p^{n \times n}$ of rank n as well as a (compact) description $mat \in \{0,1\}^*$ of the implicit representation $[\mathbf{A}]_i$ of $\mathbf{A}$.*[9]

Reconstruction of matrix from matrix description. $\mathsf{Constr}(\mathcal{MG}_k, mat)$ *returns $[\mathbf{A}]_i \in G_i^{n \times n}$ on input of a matrix description mat.*

Correctness. MCon *is called* correct *relative to $\mathcal{G}_k$ if for all $\lambda \in \mathbb{N}$, $\mathcal{MG}_k := (k, G_1, \ldots, G_{k+1}, g_1, \ldots, g_k, e, p) \leftarrow \mathcal{G}_k(1^\lambda)$, and $(\mathbf{A}, mat_{\mathbf{A}}) \leftarrow \mathsf{SetupNFR}(\mathcal{MG}_k)$, $(\mathbf{B}, mat_{\mathbf{B}}) \leftarrow \mathsf{SetupFR}(\mathcal{MG}_k)$, the matrices $\mathbf{A}$ and $\mathbf{B}$ are of rank ℓ and of rank n with probability 1, respectively, and $[\mathbf{A}]_i \leftarrow \mathsf{Constr}(\mathcal{MG}_k, mat_{\mathbf{A}})$ and $[\mathbf{B}]_i \leftarrow \mathsf{Constr}(\mathcal{MG}_k, mat_{\mathbf{B}})$.*

Security. MCon *is called* secure *relative to $\mathcal{G}_k$, if for all PPT algorithms A and for $\mathcal{MG}_k \leftarrow \mathcal{G}_k(1^\lambda)$, $(\mathbf{A}, mat) \leftarrow \mathsf{SetupNFR}(\mathcal{MG}_k)$, and $(\mathbf{A}', mat') \leftarrow \mathsf{SetupFR}(\mathcal{MG}_k)$ holds that the advantage*

$$\mathsf{Adv}^{\mathsf{ind\text{-}matrix\text{-}rank}}_{\mathsf{MCon},A}(1^\lambda) := \left|\Pr[1 \leftarrow A(1^\lambda, \mathcal{MG}_k, mat)] - \Pr[1 \leftarrow A(1^\lambda, \mathcal{MG}_k, mat')]\right|$$

is negligible in λ.

[9] This description mat can always be set to $[\mathbf{A}]_i$. In some cases (e.g., in case of the ℓ-linear distribution), however, $[\mathbf{A}]_i$ has more structure can be represented with fewer group elements, see also [11].

Construction of the LPKE scheme with efficient weak opening. Apart from an (n, ℓ)-indistinguishable matrix constructor for G_i, we additionally need a hash function $H : G_i \to \{0,1\}$ such that $H(a)$, for uniformly random $a \leftarrow G_i$, is statistically indistinguishable from the uniform distribution on $\{0,1\}$. By writing $H(\vec{b})$, where $\vec{b}$ is a vector of group elements from G_i, we refer to the componentwise application of the hash function, which results in a (bit-)vector of hash values of the same length as $\vec{b}$.

Based on these ingredients, we can define a lossy encryption scheme with efficient weak opening $\mathsf{LPKE} = (\mathsf{Gen}, \mathsf{LGen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\{0,1\}^{n-\ell}$ and ciphertexts consisting of ℓ group elements and $n - \ell$ bits. Note that the parameter ℓ reflects the strength of the assumption we are willing to make, the smaller ℓ, the stronger the underlying assumption. For instance, the assumption that random rank ℓ matrices are indistinguishable from random full rank matrices is implied by the assumption that random rank $\ell - 1$ matrices are indistinguishable from random full rank matrices. (Furthermore, rank ℓ vs. n indistinguishability is implied by the ℓ-linear assumption.) Hence, to make ciphertexts as compact as possible, one would choose $\ell = 1$ and could, e.g., base security on the 1-linear assumption which equals DDH.

The idea underlying encryption (with a real key) in our construction is as follows: a message bit is encrypted using the hash of a randomized linear dependent row vector of $\mathbf{A}$ given in implicit representation. Additionally, the linear independent row vectors of $\mathbf{A}$ are randomized the same way and given in implicit representation as part of the ciphertext. Decryption then boils down to recomputing the (implicit representation of the) linear dependent vector from the (implicit representations of the) linear independent vectors. As all row vectors are randomized the same way (which is a linear operation), the dependencies are not changed by the randomization. The details of LPKE are given below.

- $\mathsf{Gen}(1^\lambda)$ runs the group generator $\mathcal{MG}_k := (k, G_1, \ldots, G_{k+1}, g_1, \ldots, g_k, e, p) \leftarrow \mathcal{G}_k(1^\lambda)$ as well as $(\mathbf{A}, \mathit{mat}) \leftarrow \mathsf{SetupNFR}(\mathcal{MG}_k)$ to choose a matrix of rank ℓ. Let $\mathbf{A}_0$ denote the first ℓ rows of $\mathbf{A}$ and $\mathbf{A}_1$ the remaining $n - \ell$ rows. Then it computes a matrix $\mathbf{T} \in \mathbb{Z}_p^{(n-\ell)\times\ell}$ satisfying

  $$\mathbf{T}\mathbf{A}_0 = \mathbf{A}_1 \tag{4}$$

  As the rows of $\mathbf{A}_1$ linearly depend on the rows of $\mathbf{A}_0$, $\mathbf{T}$ always exists and can be computed efficiently (e.g., using Gaussian Elimination). The algorithm returns $\mathit{pk} := (\mathcal{MG}_k, \mathit{mat})$ and $\mathit{sk} := (\mathcal{MG}_k, \mathbf{T})$.
- $\mathsf{LGen}(1^\lambda)$ runs the group generator $\mathcal{MG}_k := (k, G_1, \ldots, G_{k+1}, g_1, \ldots, g_k, e, p) \leftarrow \mathcal{G}_k(1^\lambda)$ as well as $(\mathbf{A}, \mathit{mat}) \leftarrow \mathsf{SetupFR}(\mathcal{MG}_k)$ to choose a matrix of rank n. The algorithm returns $\mathit{pk} := (\mathcal{MG}_k, \mathit{mat})$ and $\mathit{sk} := (\mathcal{MG}_k, \mathbf{A})$.
- $\mathsf{Enc}(\mathit{pk}, \vec{m})$ reconstructs the matrix $[\mathbf{A}]_i \leftarrow \mathsf{Constr}(\mathcal{MG}_k, \mathit{mat})$. Let $[\mathbf{A}_0]_i$ denote the first ℓ rows of $[\mathbf{A}]_i$ and $[\mathbf{A}_1]_i$ the remaining $n - \ell$ rows. Then it chooses $\vec{w} \leftarrow \mathbb{Z}_p^n$, computes

  $$\begin{aligned} [\vec{c}_0]_i &:= [\mathbf{A}_0\vec{w}]_i \\ \vec{c}_1 &:= H([\mathbf{A}_1\vec{w}]_i) \oplus \vec{m} \end{aligned} \tag{5}$$

(using exponentiations with the entries of $\vec{w}$), and returns ciphertext $c := ([\vec{c}_0]_i, \vec{c}_1) \in G_i^\ell \times \{0,1\}^{n-\ell}$.

– $\mathsf{Dec}(sk, c)$ recomputes $\vec{m}$ as $\vec{m} := H([\mathbf{T}\vec{c}_0]_i) \oplus \vec{c}_1$.

We show that LPKE indeed satisfies the four properties of a lossy encryption scheme with efficient weak opening.

Theorem 2. *If* MCon *is secure and the output of* H *statistically indistinguishable from uniform for random input then* LPKE *is a lossy encryption scheme with efficient weak opening.*

Proof.

Correctness for real keys. Given a real public key $pk := (\mathcal{MG}_k, mat)$ and secret key $sk := (\mathcal{MG}_k, \mathbf{T})$ returned by $\mathsf{Gen}(1^\lambda)$ as well as a ciphertext $c := ([\vec{c}_0]_i, \vec{c}_1)$, correctness of decryption follows from the equation

$$\begin{aligned} H([\mathbf{T}\vec{c}_0]_i) \oplus \vec{c}_1 &= H([\mathbf{T}\vec{c}_0]_i) \oplus H([\mathbf{A}_1\vec{w}]_i) \oplus \vec{m} \\ &= H([\mathbf{T}\mathbf{A}_0\vec{w}]_i) \oplus H([\mathbf{A}_1\vec{w}]_i) \oplus \vec{m} \\ &= H([\mathbf{A}_1\vec{w}]_i) \oplus H([\mathbf{A}_1\vec{w}]_i) \oplus \vec{m} \end{aligned} \tag{6}$$

Indistinguishability of real keys from lossy keys. It follows from the security of MCon that a real public key $(\mathcal{MG}_k, mat)$ generated by $\mathsf{Gen}(1^\lambda)$ is indistinguishable from a lossy one $(\mathcal{MG}_k, mat')$ generated by $\mathsf{LGen}(1^\lambda)$.

Lossiness of encryption with lossy keys. Consider the matrix $[\mathbf{A}]_i \leftarrow \mathsf{Constr}(\mathcal{MG}_k, mat)$, where mat is computed by $\mathsf{LGen}(1^\lambda)$. This matrix has full rank, so the linear map defined by $\mathbf{A}$ as $\vec{w} \mapsto \mathbf{A}\vec{w}$ is bijective. Thus, for uniformly random $\vec{w}$, $[\vec{c}_0]_i = [\mathbf{A}_0\vec{w}]_i$ is uniformly random over G_i^ℓ and $[\mathbf{A}_1\vec{w}]_i$ is uniformly random over $G_i^{n-\ell}$ (even when $\mathbf{A}$ is given).

Now, since by assumption the output of H is statistically close to uniform for uniformly random input, $H([\mathbf{A}_1\vec{w}]_i) \oplus \vec{m}$ will also be statistically close to uniform over $\{0,1\}^{n-\ell}$ for any string $\vec{m}$.

Hence, for uniformly random $\vec{w} \leftarrow \mathbb{Z}_p^n$, the distributions of

$$(\mathbf{A}, ([\mathbf{A}_0\vec{w}]_i, H([\mathbf{A}_1\vec{w}]_i) \oplus \vec{m})) \qquad \text{and} \qquad (\mathbf{A}, ([\mathbf{A}_0\vec{w}]_i, H([\mathbf{A}_1\vec{w}]_i) \oplus \vec{m}'))$$

are statistically close for any two distinct message vectors $\vec{m} \neq \vec{m}' \in \{0,1\}^{n-\ell}$.

Efficient weak openability. Let a lossy keypair $(pk = (\mathcal{MG}_k, mat), sk = (\mathcal{MG}_k, \mathbf{A})) \leftarrow \mathsf{LGen}(1^\lambda)$, message vector $\vec{m}$, a ciphertext $c := ([\vec{c}_0]_i, \vec{c}_1) \leftarrow \mathsf{Enc}(pk, \vec{m}'; \vec{w}')$, as well as the corresponding encryption randomness $\vec{w}'$ be given. Then Opener should efficiently determine some encryption randomness $\vec{w}$ such that $\mathsf{Enc}(pk, \vec{m}; \vec{w}) = ([\vec{c}_0]_i, \vec{c}_1)$. This can be done by setting up a linear system of equations in the exponent

$$\mathbf{A}\vec{w} = \vec{b}, \tag{7}$$

where the right-hand side vector

$$\vec{b} = \begin{pmatrix} \vec{b}_0 \\ \vec{b}_1 \end{pmatrix} \tag{8}$$

satisfies $\vec{b}_0 = \vec{c}_0$ and $H([\vec{b}_1]_i) \oplus \vec{c}_1 = \vec{m}$.

First, Opener can easily determine $\vec{b}_0 := \vec{c}_0 \in \mathbb{Z}_p^\ell$, i.e., the discrete logarithms of $[\vec{c}_0]_i$ to the base g_i, by computing $\mathbf{A}\vec{w}'$. Second, it can efficiently find a vector $\vec{b}_1 \in \mathbb{Z}_p^{n-\ell}$ satisfying $H([\vec{b}_1]_i) \oplus \vec{c}_1 = \vec{m}$ by randomly guessing one component of $\vec{b}_1$ after another and verifying the equation for this component. As the output of H is close to uniform for random input, this will require about $2(n-\ell)$ steps. After that, Opener can solve the system of linear equations from Equation 7 by multiplying with the inverse of $\mathbf{A}$ as this matrix is of full rank.

It is not hard to see that the determined randomness $\vec{w}$ has the correct distribution, i.e., $\vec{w}$ is uniformly chosen from

$$\mathsf{Coins}(\vec{m}, c) := \{\vec{w} \in \mathbb{Z}_p^n \mid \mathsf{Enc}(pk, \vec{m}; \vec{w}) = c\} \tag{9}$$

Note that each $\vec{w} \in \mathsf{Coins}(\vec{m}, c)$ uniquely determines a right-hand side $\vec{b}$ in (7), i.e., a vector from

$$\mathsf{KENCs}(\vec{m}, c) := \left\{ \vec{b} = \begin{pmatrix} \vec{b}_0 \\ \vec{b}_1 \end{pmatrix} \;\middle|\; \vec{b}_0 = \vec{c}_0 \wedge H([\vec{b}_1]_i) \oplus \vec{c}_1 = \vec{m} \right\} \tag{10}$$

Hence, to uniformly sample $\vec{w}$ from $\mathsf{Coins}(\vec{m}, c)$ it suffices to uniformly sample $\vec{b}$ from $\mathsf{KENCs}(\vec{m}, c)$ and invert the bijective mapping by computing $\mathbf{A}^{-1}\vec{b}$. This is exactly what Opener does.

4 From MDDH Assumptions To Matrix Rank Assumptions

We have seen in Section 3 that in order to build an $(n-\ell)$-bit LPKE scheme with efficient weak opening, it suffices to define a secure (n, ℓ)-indistinguishable matrix constructor. In the following, we first show that such a constructor is generically given by any $\mathcal{D}_{n,\ell}$-MDDH assumption (including DDH, ℓ-Lin, ℓ-SCasc, (n, ℓ)-circulant matrix assumption, etc.). Then, we consider the size of the public key when using different members of MDDH assumption family.

Generic construction from MDDH assumptions. Let $\mathcal{G}_k$ be a k-linear group generator and $\mathcal{MG}_k := (k, G_1, \ldots, G_{k+1}, g_1, \ldots, g_k, e, p) \leftarrow \mathcal{G}_k(1^\lambda)$. Furthermore, let $\mathcal{D}_{n,\ell}$ be a matrix distribution over $\mathbb{Z}_p^{n \times \ell}$, where $n > \ell$. We assume that the first ℓ rows of an output of $\mathcal{D}_{n,\ell}$ forms a regular matrix with overwhelming probability. A (n, ℓ)-indistinguishable matrix constructor $\mathsf{MCon}_{\mathcal{D}_{n,\ell}\text{-}\mathsf{MDDH}}$ for G_i can then be defined based on $\mathcal{D}_{n,\ell}$-MDDH as follows:

- $\mathsf{SetupNFR}(\mathcal{MG}_k)$ samples a matrix $\mathbf{A}' \leftarrow \mathcal{D}_{n,\ell}$ of rank ℓ according to the given matrix distribution. If $\mathbf{A}'$ is not of rank ℓ the sampling is repeated. (Note that since $\mathcal{D}_{n,\ell}$ outputs full rank matrices with overwhelming probability this case should virtually never happen.) Furthermore, a random matrix $\mathbf{R} \leftarrow \mathbb{Z}_p^{\ell \times (n-\ell)}$ is sampled. Then it computes $\mathbf{A} := \mathbf{A}'(\mathbf{I}_\ell || \mathbf{R}) = (\mathbf{A}' || \mathbf{A}'\mathbf{R})$, where $\mathbf{I}_\ell$ is the $\ell \times \ell$ identity matrix, and returns $(\mathbf{A}, [\mathbf{A}]_i)$.

- $\mathsf{SetupFR}(\mathcal{MG}_k)$ samples a matrix $\mathbf{A}' \leftarrow \mathcal{D}_{n,\ell}$ of rank ℓ (if the rank of $\mathbf{A}'$ is smaller sampling is repeated). After that, random matrices $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times(n-\ell)}$ are sampled until $\mathbf{A} := (\mathbf{A}'||\mathbf{U})$ is of full rank n. (Note that $\mathbf{A}$ will be of rank n with overwhelming probability of at least $1 - \frac{n-\ell}{p^{n-\ell}}$ for uniform $\mathbf{U}$.) It then returns $(\mathbf{A}, [\mathbf{A}]_i)$.
- $\mathsf{Constr}(\mathcal{MG}_k, mat)$ returns mat (as the matrix is not compressed).

Remark 2. Consider the matrix $\mathbf{A}' \leftarrow \mathcal{D}_{n,\ell}$ generated during $\mathsf{SetupNFR}(\mathcal{MG}_k)$. Let $\mathbf{A}'_0$ denote the first ℓ rows of $\mathbf{A}'$ and $\mathbf{A}'_1$ the last $n-\ell$ rows of $\mathbf{A}'$. Then the transformation matrix $\mathbf{T}$ from Equation 4, which is used as the secret key, can be set to $\mathbf{T} := \mathbf{A}'_1(\mathbf{A}'_0)^{-1}$. Correctness follows from

$$\begin{aligned}\mathbf{T}\mathbf{A}_0 &= \mathbf{A}'_1(\mathbf{A}'_0)^{-1}\mathbf{A}_0\\ &= \mathbf{A}'_1(\mathbf{A}'_0)^{-1}\mathbf{A}'_0(\mathbf{I}_\ell||\mathbf{R})\\ &= \mathbf{A}'_1(\mathbf{I}_\ell||\mathbf{R})\\ &= \mathbf{A}_1\end{aligned} \tag{11}$$

Correctness. Consider $(\mathbf{A}, mat_{\mathbf{A}}) \leftarrow \mathsf{SetupNFR}(\mathcal{MG}_k)$ and $(\mathbf{B}, mat_{\mathbf{B}}) \leftarrow \mathsf{SetupFR}(\mathcal{MG}_k)$. Obviously, $\mathbf{A} = (\mathbf{A}'||\mathbf{A}'\mathbf{R})$ will be of rank ℓ as this is the case for $\mathbf{A}'$. Similarly, $\mathbf{B} := (\mathbf{B}'||\mathbf{U})$ will be of rank n by construction. Furthermore, clearly, it holds that $[\mathbf{A}]_i \leftarrow \mathsf{Constr}(\mathcal{MG}_k, mat_{\mathbf{A}})$ and $[\mathbf{B}]_i \leftarrow \mathsf{Constr}(\mathcal{MG}_k, mat_{\mathbf{B}})$.

Security. As for security we show

Lemma 1. *If the $\mathcal{D}_{n,\ell}$-MDDH assumption holds relative to $\mathcal{G}_k$, then the scheme* $\mathsf{MCon}_{\mathcal{D}_{n,\ell}\text{-}\mathsf{MDDH}}$ *is secure.*

Proof. First note that the distribution of $\mathbf{A}$ returned by $\mathsf{SetupNFR}$ and the distribution of $\mathbf{B}$ returned by $\mathsf{SetupFR}$ are statistically indistinguishable from the distribution of $(\mathbf{A}'||\mathbf{A}'\mathbf{R})$ and $(\mathbf{A}'||\mathbf{U})$, respectively, where $\mathbf{A}' \leftarrow \mathcal{D}_{n,\ell}$, $\mathbf{R} \leftarrow \mathbb{Z}_p^{\ell\times(n-\ell)}$, and $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times(n-\ell)}$.

Then considering the latter distributions, the lemma immediately follows from the $\mathcal{D}_{n,\ell}$-Matrix Diffie-Hellman assumption and its random self-reducibility. More concretely, the $\mathcal{D}_{n,\ell}$-MDDH assumption demands that for all PPT adversaries D holds that

$$|\Pr[D(\mathcal{MG}_k, [\mathbf{A}']_i, [\mathbf{A}'\vec{r}]_i) = 1] - \Pr[D(\mathcal{MG}_k, [\mathbf{A}']_i, [\vec{u}]_i) = 1]|$$

is negligible, where $\mathcal{MG}_k \leftarrow \mathcal{G}_k(1^\lambda)$, $\mathbf{A}' \leftarrow \mathcal{D}_{n,\ell}$, $\vec{r} \leftarrow \mathbb{Z}_p^\ell$ and $\vec{u} \leftarrow \mathbb{Z}_p^n$. Hence, $[\mathbf{A}'||\mathbf{A}'\vec{r}]_i$ is computationally indistinguishable from $[\mathbf{A}'||\vec{u}]_i$. As any matrix assumption is random self-reducible (Lemma 1 in [11]), it follows that

$$|\Pr[D(\mathcal{MG}_k, [\mathbf{A}']_i, [\mathbf{A}'\mathbf{R}]_i) = 1] - \Pr[D(\mathcal{MG}_k, [\mathbf{A}']_i, [\mathbf{U}]_i) = 1]|$$

is negligible, where $\mathbf{R} \leftarrow \mathbb{Z}_p^{\ell\times(n-\ell)}$ and $\mathbf{U} \leftarrow \mathbb{Z}_p^{n\times(n-\ell)}$. Thus, $[\mathbf{A}'||\mathbf{A}'\mathbf{R}]_i$ is computationally indistinguishable from $[\mathbf{A}'||\mathbf{U}]_i$.

Concrete instantiations. Let us now consider what we get from different members of the MDDH assumption family.

1-*bit LPKE from standard assumptions.* From standard assumptions like DDH and ℓ-Lin, we can immediately obtain a one bit lossy encryption scheme by means of the corresponding indistinguishable matrix constructor. More precisely, for ℓ-Lin we would consider the $\mathcal{L}_{\ell+1,\ell}$ matrix distribution which samples $(\ell+1) \times \ell$ matrices of the form

$$\mathbf{A}' = \begin{pmatrix} s_1 & 0 & 0 & \dots & 0 \\ 0 & s_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & s_\ell \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix} \tag{12}$$

Hence, this results in a public key of the form

$$[\mathbf{A}]_i = \left[\begin{pmatrix} s_1 & 0 & 0 & \dots & 0 & s_1 r_1 \\ 0 & s_2 & 0 & \dots & 0 & \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & s_\ell & s_\ell r_\ell \\ 1 & 1 & 1 & \dots & 1 & r_1 + \dots + r_\ell \end{pmatrix} \right]_i , \tag{13}$$

where $r_i \leftarrow \mathbb{Z}_p$, which can be represented using $2(\ell+1)$ group elements.

Multi-bit LPKE from standard assumptions. Note that the number of bits we can encrypt equals the number of linearly dependent row vectors of $\mathbf{A} \in \mathbb{Z}_p^{n \times n}$, i.e., $n-\ell$. Thus, if we had a distribution $\mathcal{D}_{n,\ell}$ that yields matrices with more than one linearly dependent vector, i.e., $n > \ell+1$, our construction would be able to encrypt more than one bit. Hence, we could obtain a scheme for multiple bits from a standard assumption by finding a $\mathcal{D}_{n,\ell}$-MDDH assumption with $n > \ell+1$ which is implied by this standard assumption. For instance, the ℓ-Lin assumption implies $\mathcal{U}_{n,\ell}$-MDDH for arbitrary n, where $\mathcal{U}_{n,\ell}$ samples uniform $n \times \ell$ matrices of rank ℓ (this follows from Lemma A.1 in [26]). Hence, from DDH, for example, we can get a scheme for $(n-1)$-bit messages with arbitrary $n \in \mathbb{N}$ by means of the uniform distribution $\mathcal{U}_{n,1}$ which samples a matrix of the form

$$\mathbf{A}' = \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} \tag{14}$$

and, thus, yields a public key of the form

$$[\mathbf{A}]_i = \left[\begin{pmatrix} s_1 & s_1 r_1 & \dots & s_1 r_{n-1} \\ \vdots & & & \vdots \\ s_n & s_n r_1 & \dots & s_n r_{n-1} \end{pmatrix} \right]_i , \tag{15}$$

where $r_i \leftarrow \mathbb{Z}_p$. Note that the resulting scheme is essentially the DDH-based scheme sketched in the introduction (with the minor difference that s_n is set to 1 instead of being uniformly chosen).

It is interesting to observe that ℓ-Lin is a family of assumptions which (at least in the generic group model) become strictly weaker as ℓ grows and that we can get an LPKE scheme for messages of arbitrary size for each member of this family (by means of $\mathcal{U}_{n,\ell}$).

On the downside, if make the detour to the $\mathcal{U}_{n,\ell}$ distribution (instead of directly building on ℓ-Lin), the public key will consist of n^2 group elements to represent $[\mathbf{A}]_i$. Alternatively, we can take a more direct approach and extend a (standard) $\mathcal{D}_{\ell+1,\ell}$-MDDH assumption (like ℓ-Lin) to the $\mathcal{D}_{n,\ell}$-MDDH assumption, where the first $\ell+1$ rows of $\mathbf{A}' \leftarrow \mathcal{D}_{n,\ell}$ are sampled as by $\mathcal{D}_{\ell+1,\ell}$ and the remaining $n-\ell-1$ are sampled uniformly. In this case, $\mathcal{D}_{n,\ell}$-MDDH is implied by $\mathcal{D}_{\ell+1,\ell}$-MDDH [24]. The representation of $[\mathbf{A}]_i$ will consist of $E+(n-\ell-1)\ell+n(n-\ell)$ group elements to encrypt $n-\ell$ bits, where E is the number of elements required to represent a matrix sampled by the $\mathcal{D}_{\ell+1,\ell}$ distribution (e.g., 1 for ℓ-SCasc).

Multi-bit LPKE from a new $\mathcal{D}_{n,\ell}$-MDDH assumption. A $\mathcal{D}_{n,\ell}$-MDDH for $n > \ell+1$ with an optimal representation size has recently been proposed in [24]. The circulant matrix distribution $\mathcal{C}_{\ell+d,\ell}$ outputs matrices $\mathbf{A}' \in \mathbb{Z}_p^{(\ell+d)\times\ell}$ which can be represented using d group elements. The assumption has been shown to hold in the ℓ-linear generic group model [24]. Plugging this distribution into our scheme, we obtain a public key consisting of $d+(\ell+d)d$ group elements (representing $[\mathbf{A}]_i$) to encrypt d bits.

5 From the BDDH Assumption To a Compact Matrix Rank Assumption

In this section, we show how to leverage the lossy trapdoor function construction of Boyen and Waters [6] to obtain a $(n,1)$-indistinguishable matrix constructor $\mathsf{MCon}_{\mathsf{BDDH}}$ with a linear-size matrix description *mat*. This translates to an $(n-1)$-bit lossy encryption scheme featuring a linear public key size. (Note that the size of the secret key is also linear.)

Essentially, the idea is to generate the quadratic number of group elements in the matrix from a linear number of group elements, by applying a bilinear map. A technical hurdle is to do this in a way such that matrices computed in this way have either rank 1 or full rank, in a computationally indistinguishable way. Here we apply the "linear equations" technique of Boyen and Waters, which enables an algorithm to re-compute the full matrix by applying the bilinear map, *except for the diagonal.* The diagonal entries of the matrix are given additionally in the matrix description *mat*, and set-up such that the resulting matrix has either rank 1 or full rank. Interestingly, the *lossy* trapdoor function of Boyen and Waters corresponds to our *injective* encryption scheme, and vice versa.

Let $\mathcal{MG}_2 := (2, G_1, G_2, G_3, g_1, g_2, g_3, e, p) \leftarrow \mathcal{G}_2(1^\lambda)$, where $G_1 = G_2$ and $g_1 = g_2$, be a symmetric bilinear group generator. Then a $(n,1)$-indistinguishable matrix constructor $\mathsf{MCon}_{\mathsf{BDDH}}$ for G_1 can be defined as follows:

- $\mathsf{SetupNFR}(\mathcal{MG}_2)$ samples two uniformly random elements $h, k \leftarrow \mathbb{Z}_p^*$, and two exponent vectors $\vec{r} = (r_1, \ldots, r_n)^\top \leftarrow (\mathbb{Z}_p^*)^n$ and $\vec{u} = (u_1, \ldots, u_n)^\top \leftarrow (\mathbb{Z}_p^*)^n$. Then it sets $\mathbf{A} := (a_{i,j}) \in (\mathbb{Z}_p^*)^{n\times n}$ with $a_{i,j} := hr_iu_j$. Furthermore, it computes
    - $[\vec{s}]_1 := [(s_1, \ldots, s_n)^\top]_1 \in G_1^n$ where $s_i := (hi+k)r_i$

- $[\vec{v}]_1 := [(v_1, \ldots, v_n)^\top]_1 \in G_1^n$ where $v_j := (hj + k)u_j$
- $[\vec{d}]_3 := [(d_1, \ldots, d_n)^\top]_3 \in G_3^n$ where $d_i := hr_iu_i$

and sets $mat := ([\vec{r}]_1, [\vec{s}]_1, [\vec{u}]_1, [\vec{v}]_1, [\vec{d}]_3)$. It returns $(\mathbf{A}, mat)$.

– $\mathsf{SetupFR}(\mathcal{MG}_2)$ samples elements $h, k \leftarrow \mathbb{Z}_p^*$ and vectors $\vec{r}, \vec{u} \leftarrow (\mathbb{Z}_p^*)^n$ the same way as SetupNFR. It sets $\mathbf{A} := (a_{i,j}) \in \mathbb{Z}_p^{n\times n}$ with $a_{i,j} := hr_iu_j$ for $i \neq j$ and $a_{i,i} := hr_iu_i + 1$. Accordingly, $[\vec{s}]_1$ and $[\vec{v}]_1$ are defined as in SetupNFR but d_i is set to $d_i := hr_iu_i + 1$. It sets $mat := ([\vec{r}]_1, [\vec{s}]_1, [\vec{u}]_1, [\vec{v}]_1, [\vec{d}]_3)$ and returns $(\mathbf{A}, mat)$.

– $\mathsf{Constr}(\mathcal{MG}_2, mat)$ computes $[\mathbf{A}]_3 := ([a_{i,j}]_3)_{i,j}$ for $1 \leq i, j \leq n$ as follows:

- For $i \neq j$, it uses the pairing to compute

$$[a_{i,j}]_3 := e([r_i]_1, [v_j]_1)^{1/(j-i)} e([u_j]_1, [s_i]_1)^{-1/(j-i)} = [(r_i \cdot v_j - u_j \cdot s_i)/(j-i)]_3$$

- For $i = j$ it sets $[a_{i,i}]_3 := [d_i]_3$

Remark 3. The transformation matrix $\mathbf{T}$ from Equation 4 can be set to $\mathbf{T} := (r_2/r_1, \ldots, r_n/r_1)^\top$.

Correctness. Consider $(\mathbf{A}, mat_{\mathbf{A}}) \leftarrow \mathsf{SetupNFR}(\mathcal{MG}_2)$ and $(\mathbf{B}, mat_{\mathbf{B}}) \leftarrow \mathsf{SetupFR}(\mathcal{MG}_2)$. Let $\mathbf{A}_0$ be the first row of $\mathbf{A}$ and $\mathbf{A}_1$ be the remaining $n-1$ rows. It is easy to see that $\mathbf{TA}_0 = \mathbf{A}_1$, where $\mathbf{T}$ is defined as described above. Moreover, $\mathbf{A}$ cannot be the zero-matrix, because h and all r_i and u_j are non-zero. So $\mathbf{A}$ is of rank 1.

Note also that by construction of SetupFR we have $\mathbf{B} = \mathbf{A} + \mathbf{I}_n$, where $\mathbf{A}$ has rank 1 (as above) and $\mathbf{I}_n$ is the $(n \times n)$-identity matrix. Thus, since $\mathbf{A}$ has rank 1, $\mathbf{B}$ is row-equivalent to $\mathbf{I}_n$, which is equivalent to $\mathbf{B}$ having full rank.

To see that for $[\mathbf{A}']_3 := \mathsf{Constr}(\mathcal{MG}_2, mat_{\mathbf{A}})$ and $[\mathbf{B}']_3 := \mathsf{Constr}(\mathcal{MG}_2, mat_{\mathbf{B}})$ we have $[\mathbf{A}']_3 = [\mathbf{A}]_3$ and $[\mathbf{B}']_3 = [\mathbf{B}]_3$, first observe that the diagonal entries are correct, since $[a'_{i,i}]_3 = hr_iu_i$ and $[b'_{i,i}]_3 = hr_iu_i + 1$. Moreover, in either case we have for $i \neq j$ that

$$\begin{aligned}
[a'_{i,j}]_3 = [b'_{i,j}]_3 &= [(r_iv_j - u_js_i)/(j-i)]_3 \\
&= [(r_i(hj+k)u_j - u_j(hi+k)r_i)/(j-i)]_3 \\
&= [(hr_iu_jj + kr_iu_j - hr_iu_ji - kr_iu_j)/(j-i)]_3 \\
&= [hr_iu_j(j-i)/(j-i)]_3 \\
&= [hr_iu_j]_3
\end{aligned} \tag{16}$$

Security. Following [6], we prove security under the bilinear decisional Diffie-Hellman assumption (cf. Definition 3). However, to simplify the security proof of $\mathsf{MCon}_{\mathsf{BDDH}}$, we first define the following slightly modified BDDH* assumption, which is implied by the standard BDDH assumption from Definition 3 by a straightforward reduction.

Definition 7. *Let $\mathcal{MG}_2 := (2, G_1, G_2, G_3, g_1, g_2, e, p) \leftarrow \mathcal{G}_2(1^\lambda)$, $a, b, c \leftarrow \mathbb{Z}_p^*$, $b \leftarrow \{0,1\}$, $T_0 := abc$ and $T_1 := abc + 1$. We say that the BDDH* assumption* *holds relative to $\mathcal{G}_2$, if*

$$\mathsf{Adv}^{\mathsf{bddh*}}_{B,\mathcal{G}_2}(1^\lambda) := \left|\begin{array}{l} \quad\Pr\left[1 \leftarrow B(1^\lambda, \mathcal{MG}_2, [(a,b,c)]_1, [T_0]_3)\right] \\ -\Pr\left[1 \leftarrow B(1^\lambda, \mathcal{MG}_2, [(a,b,c)]_1, [T_1]_3)\right] \end{array}\right|$$

is a negligible function for all PPT adversaries B.

Remark 4. A straightforward reduction allows to show that $\mathsf{Adv}^{\mathsf{bddh*}}_{B,\mathcal{G}_2}(1^\lambda) \le 2 \cdot \mathsf{Adv}^{\mathsf{bddh}}_{B,\mathcal{G}_2}(1^\lambda)$ for all PPT algorithms B.

Theorem 3. *If the BDDH* assumption holds relative to $\mathcal{G}_2$, then* $\mathsf{MCon}_{\mathsf{BDDH}}$ *is secure.*

Proof. We will show that one can construct an adversary B against the BDDH* assumption from each adversary A against MCon such that

$$\mathsf{Adv}^{\mathsf{ind\text{-}matrix\text{-}rank}}_{\mathsf{MCon},A}(1^\lambda) \le n \cdot \mathsf{Adv}^{\mathsf{bddh*}}_{B,\mathcal{G}_2}(1^\lambda) \tag{17}$$

To this end, we describe a hybrid argument which consists of $n+1$ hybrid games $H_0, \dots, H_n$. In Hybrid H_δ, $\delta \in \{0, \dots, n\}$, we run A on input $\mathit{mat} := ([\vec{r}]_1, [\vec{s}]_1, [\vec{u}]_1, [\vec{v}]_1, [\vec{d}]_3)$, where all values are computed exactly as in $\mathsf{SetupNFR}$, except that

$$d_i := \begin{cases} hr_iu_i + 1 & \text{for } i < \delta \\ hr_iu_i & \text{for } i \ge \delta \end{cases}$$

Note that the input mat of A in H_0 is identically distributed to the the matrix descriptions computed by $(\mathbf{A}, \mathit{mat}) \leftarrow \mathsf{SetupNFR}(\mathcal{MG}_2)$. In H_n, A receives a matrix description mat which is distributed exactly as a matrix description computed by $(\mathbf{A}, \mathit{mat}) \leftarrow \mathsf{SetupFR}(\mathcal{MG}_2)$.

Let X_δ denote the event that A outputs "1" in Hybrid H_δ. We show that for each $\delta \in \{1, \dots, n\}$ we can construct an adversary B such that

$$\mathsf{Adv}^{\mathsf{bddh*}}_{B,\mathcal{G}_2} \ge |\Pr[X_{\delta-1}] - \Pr[X_\delta]| \tag{18}$$

which proves (17). B receives as input a BDDH*-instance $(\mathcal{MG}_2, [(a,b,c)]_1, [T])$. It creates $\mathit{mat} = ([\vec{r}]_1, [\vec{s}]_1, [\vec{u}]_1, [\vec{v}]_1, [\vec{d}]_3)$ as follows.

- $[\vec{r}]_1 := [(r_1, \dots, r_n)^\top]_1$, where $[r_\delta]_1 := [a]_1$ and $r_i \leftarrow \mathbb{Z}_p^*$ for all $i \in \{1, \dots, n\}$ with $i \ne \delta$
- $[\vec{u}]_1 := [(u_1, \dots, u_n)^\top]_1$, where $[u_\delta]_1 := [b]_1$ and $u_i \leftarrow \mathbb{Z}_p^*$ for all $i \in \{1, \dots, n\}$ with $i \ne \delta$
- $[h]_1 := [c]_1$ and $[k]_1 := [-h\delta + y]$ for $y \leftarrow \mathbb{Z}_p \setminus \{h\delta\}$
- $[\vec{s}]_1 := [(s_1, \dots, s_n)^\top]_1$, where $[s_i]_1 = [(hi+k)r_i]_1 = [(h(i-\delta)+y)r_i]_1$. Note that all the $[s_i]_1$ can efficiently be computed by B, due to the above setup of $[h]_1$, $[k]_1$, $[\vec{r}]_1$.
- $[\vec{v}]_1 := [(v_1, \dots, v_n)^\top]_1$, where $[v_j]_1 = [(hj+k)u_j]_1 = [(h(j-\delta)+y)u_j]_1$. As above, all the $[v_i]_1$ can efficiently be computed by B, due to the setup of $[h]_1$, $[k]_1$, $[\vec{u}]_1$.

Finally, B sets $[\vec{d}]_3 := [(d_1, \ldots, d_n)^\top]_3$, where

$$[d_i]_3 = \begin{cases} [hr_i u_i + 1]_3 & \text{for } i < \delta \\ [T]_3 & \text{for } i = \delta \\ [hr_i u_i]_3 & \text{for } i > \delta \end{cases}$$

Then it runs A on input $(\mathcal{MG}_2, mat)$ and outputs whatever A outputs.

Note that if $[T]_3 = [abc]_3 = [hr_\delta u_\delta]_3$, then the view of A when interacting with B is identical to its view in hybrid $H_{\delta-1}$. Thus, the probability that A outputs "1" in this case is equal to $\Pr[X_{\delta-1}]$. If $[T]_3 = [abc+1]_3 = [hr_\delta u_\delta + 1]_3$, then it is identical to H_δ, so that the the probability that A outputs "1" in this case is equal to $\Pr[X_\delta]$. This yields (18) and thus concludes the proof.

Shortcut evaluation. We remark that it is possible to reduce the number of pairing computations required to compute $[\mathbf{A}\vec{w}]_3$ for $\vec{w} \in \mathbb{Z}_p^n$, given mat. In the naïve approach sketched above, one first has to recompute $[\mathbf{A}]_3$ from mat, which requires $\mathbf{O}(n^2)$ pairing evaluations, and then $[\mathbf{A}]_3\vec{w}$.

Following the "shortcut evaluation" approach described in [6], we note that the number of pairing evaluations can be reduced to $2n = \mathbf{O}(n)$, by computing $([z_1]_3, \ldots, [z_n]_3)$ from $mat = ([\vec{r}]_1, [\vec{s}]_1, [\vec{u}]_1, [\vec{v}]_1, [\vec{d}]_3)$ and $\vec{w} \in \mathbb{Z}_p^n$ as

$$[z_j]_3 := \frac{\left[\sum_{i \neq j} \frac{w_i r_i}{j-i}\right]_1 \cdot [v_j]_1}{\left[\sum_{i \neq j} \frac{w_i u_i}{j-i}\right]_1 \cdot [s_j]_1} + [w_j d_j]_3$$

Indeed, as shown by Boyen and Waters [6], it is easy to verify that

$$[z_j]_3 = \left[\sum_{i=1}^{n} r_i u_i h w_i\right]_3$$

for all $j \in \{1, \ldots, n\}$, and thus it holds that $([z_1]_3, \ldots, [z_n]_3)^\top = [\mathbf{A}\vec{w}]_3$. Note that this "shortcut evaluation" takes only two pairing evaluations for each $j \in \{1, \ldots, n\}$, which amounts to only $2n$ pairing evaluations in total.

References

[1] Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard security does not imply security against selective-opening. In: Proc. EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 645–662. Springer (2012)

[2] Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Proc. EUROCRYPT 2009. Lecture Notes in Computer Science, vol. 5479, pp. 1–35. Springer (2009)

[3] Bellare, M., Waters, B., Yilek, S.: Identity-based encryption secure against selective opening attack. In: Proc. TCC 2011. Lecture Notes in Computer Science, vol. 6597, pp. 235–252. Springer (2011)

[4] Bellare, M., Yilek, S.: Encryption schemes secure under selective opening attack. Cryptology ePrint Archive, Report 2009/101 (2009), http://eprint.iacr.org/2009/101

[5] Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Proc. CRYPTO 2004. Lecture Notes in Computer Science, vol. 3152, pp. 41–55. Springer (2004)

[6] Boyen, X., Waters, B.: Shrinking the keys of discrete-log-type lossy trapdoor functions. In: Proc. ACNS 2010. Lecture Notes in Computer Science, vol. 6123, pp. 35–52 (2010)

[7] Böhl, F., Hofheinz, D., Kraschewski, D.: On definitions of selective opening security. In: Proc. Public Key Cryptography 2012. Lecture Notes in Computer Science, vol. 7293, pp. 522–539. Springer (2012)

[8] Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multiparty computation. In: Proc. STOC 1996. pp. 639–648. ACM (1996)

[9] Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Proc. Public Key Cryptography 2001. Lecture Notes in Computer Science, vol. 1992, pp. 119–136. Springer (2001)

[10] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: Proc. FOCS 1999. pp. 523–534. IEEE Computer Society (1999)

[11] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for diffie-hellman assumptions. In: Proc. CRYPTO (2) 2013. Lecture Notes in Computer Science, vol. 8043, pp. 129–147. Springer (2013)

[12] Fehr, S., Hofheinz, D., Kiltz, E., Wee, H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Proc. EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 381–402. Springer (2010)

[13] Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Proc. EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 44–61. Springer (2010)

[14] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. J. Cryptology 26(1), 39–74 (2013)

[15] Fujisaki, E.: All-but-many encryption - a new framework for fully-equipped uc commitments. In: Proc. ASIACRYPT (2) 2014. Lecture Notes in Computer Science, vol. 8874, pp. 426–447. Springer (2014)

[16] Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Proc. EUROCRYPT 2013. Lecture Notes in Computer Science, vol. 7881, pp. 1–17. Springer (2013)

[17] Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Proc. EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 415–432. Springer (2008)

[18] Hemenway, B., Libert, B., Ostrovsky, R., Vergnaud, D.: Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In: Proc. ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 70–88. Springer (2011)

[19] Hofheinz, D.: All-but-many lossy trapdoor functions. In: Proc. EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 209–227. Springer (2012)

[20] Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Proc. CRYPTO 2007. Lecture Notes in Computer Science, vol. 4622, pp. 553–571. Springer (2007)

[21] Hofheinz, D., Rao, V., Wichs, D.: Standard security does not imply indistinguishability under selective opening. IACR Cryptology ePrint Archive 2015, 792 (2015), `http://eprint.iacr.org/2015/792`

[22] Huang, Z., Liu, S., Qin, B.: Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In: Proc. Public Key Cryptography 2013. Lecture Notes in Computer Science, vol. 7778, pp. 369–385. Springer (2013)

[23] Möller, B.: Algorithms for multi-exponentiation. In: Vaudenay, S., Youssef, A.M. (eds.) Selected Areas in Cryptography, 8th Annual International Workshop, SAC 2001 Toronto, Ontario, Canada, August 16-17, 2001, Revised Papers. Lecture Notes in Computer Science, vol. 2259, pp. 165–180. Springer (2001), `http://dx.doi.org/10.1007/3-540-45537-X_13`

[24] Morillo, P., Ràfols, C., Villar, J.L.: Matrix computational assumptions in multilinear groups. Cryptology ePrint Archive, Report 2015/353 (2015), `http://eprint.iacr.org/`

[25] Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Proc. SODA 2001. pp. 448–457. ACM/SIAM (2001)

[26] Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Proc. CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 18–35. Springer (2009)

[27] Ostrovsky, R., Rao, V., Scafuro, A., Visconti, I.: Revisiting lower and upper bounds for selective decommitments. In: Proc. TCC 2013. pp. 559–578 (2013)

[28] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proc. EUROCRYPT 1999. Lecture Notes in Computer Science, vol. 1592, pp. 223–238. Springer (1999)

[29] Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Proc. CRYPTO 2008. Lecture Notes in Computer Science, vol. 5157, pp. 554–571. Springer (2008)

[30] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proc. STOC 2008. pp. 187–196. ACM (2008)

[31] Shacham, H.: A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. IACR ePrint Archive, report 2007/74 (2007), `http://eprint.iacr.org/2007/74`

[32] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Proc. EUROCRYPT 1997. Lecture Notes in Computer Science, vol. 1233, pp. 256–266. Springer (1997)