Better Security for Deterministic Public-Key Encryption: The Auxiliary-Input Setting

Zvika Brakerski[1] and Gil Segev[2]

[1] Department of Computer Science and Applied Mathematics
Weizmann Institute of Science, Rehovot 76100, Israel
zvika.brakerski@weizmann.ac.il
[2] Microsoft Research, Mountain View, CA 94043, USA
gil.segev@microsoft.com

Abstract. Deterministic public-key encryption, introduced by Bellare, Boldyreva, and O'Neill (CRYPTO '07), provides an alternative to randomized public-key encryption in various scenarios where the latter exhibits inherent drawbacks. A deterministic encryption algorithm, however, cannot satisfy any meaningful notion of security when the plaintext is distributed over a small set. Bellare et al. addressed this difficulty by requiring semantic security to hold only when the plaintext has high min-entropy from the adversary's point of view.

In many applications, however, an adversary may obtain auxiliary information that is related to the plaintext. Specifically, when deterministic encryption is used as a building block of a larger system, it is rather likely that plaintexts do not have high min-entropy from the adversary's point of view. In such cases, the framework of Bellare et al. might fall short from providing robust security guarantees.

We formalize a framework for studying the security of deterministic public-key encryption schemes with respect to auxiliary inputs. Given the trivial requirement that the plaintext should not be efficiently recoverable from the auxiliary input, we focus on *hard-to-invert* auxiliary inputs. Within this framework, we propose two schemes: the first is based on the decisional Diffie-Hellman (and, more generally, on the d-linear) assumption, and the second is based on a rather general class of subgroup indistinguishability assumptions (including, in particular, quadratic residuosity and Paillier's composite residuosity). Our schemes are secure with respect to any auxiliary input that is subexponentially hard to invert (assuming the *standard* hardness of the underlying computational assumptions). In addition, our first scheme is secure even in the multi-user setting where related plaintexts may be encrypted under multiple public keys. Constructing a scheme that is secure in the multi-user setting (even without considering auxiliary inputs) was identified by Bellare et al. as an important open problem.

1 Introduction

Public-key encryption is one of the most basic cryptographic tasks. A public-key encryption scheme consists of three algorithms: a key-generation algorithm that

produces a secret key and a corresponding public key, an encryption algorithm that uses the public key for mapping plaintexts into ciphertexts, and a decryption algorithm that uses the secret key for recovering plaintexts from ciphertexts. For modeling the security of public-key encryption schemes, the fundamental notion of *semantic security* was introduced in the seminal work of Goldwasser and Micali [20]. Semantic security asks that it should be infeasible to gain any effective information on the plaintext by seeing the ciphertext and the public key. More specifically, whatever can be computed efficiently from the ciphertext, the public key and possibly some auxiliary information, can essentially be computed efficiently from the public key and the auxiliary information alone.

Together with its rigorous, robust, and meaningful modeling of security, semantic security inherently carries the requirement for a randomized encryption algorithm. In some cases, however, a randomized encryption algorithm may suffer from various drawbacks. In terms of efficiency, ciphertexts are not length preserving (and might be significantly longer than their corresponding plaintexts), and are in general not efficiently searchable. These properties severely limit the deployment of public-key encryption schemes in applications involving, for example, massive data sets where the ciphertext expansion is crucial, or global deduplication-based storage systems where searches are highly frequent (e.g., [26]). In addition, in terms of security, the security guarantees provided by randomized public-key encryption, and by randomized cryptographic primitives in general, are typically highly dependant on the availability of true and fresh random bits (see, for example, [3] and the references therein).

Deterministic public-key encryption. For dealing with these kind of drawbacks, Bellare, Boldyreva, and O'Neill [2] initiated the study of *deterministic* public-key encryption schemes. These are public-key encryption schemes in which the encryption algorithm is deterministic[3]. In this setting, where full-fledged sematic security is out of reach, Bellare et al. put forward the goal of formalizing a notion of security that captures semantic security as much as possible. An immediate consequence of having a deterministic encryption algorithm, however, is that essentially no meaningful notion of security can be satisfied if the plaintext is distributed over a set of polynomial size. Indeed, in such a case an adversary who is given a public key pk and an encryption c of some plaintext m under the public key pk, can simply encrypt all possible plaintexts, compare each of them to the given ciphertext c, and thus recover the plaintext m.

Bellare et al. addressed this problem by requiring security to hold only when the plaintext is sampled from a distribution of high min-entropy. Subject to this restriction, they adapted semantic security to the setting of deterministic encryption: For any high-entropy plaintext distribution, whatever can be computed efficiently from the ciphertext and the public key, can also be computed efficiently from the public key alone. Constructions of deterministic public-key encryption schemes satisfying this and similar notions of security were proposed in the random oracle model by Bellare et al. [2], and then in the standard model

[3] Note that this is effectively a collection of injective trapdoor functions (assuming the decryption algorithm is deterministic as well).

by Bellare, Fischlin, O'Neill, and Ristenpart [4], by Boldyreva, Fehr, and O'Neill [5], and by O'Neill [23]. We refer the reader to Section 1.2 for an elaborated discussion of these constructions.

Security with respect to auxiliary information. In typical applications, a deterministic public-key encryption scheme is used as building block of a larger system. In such a setting, an adversary usually has additional information that it can use when trying to break the security of the scheme. This danger becomes even more critical when such additional information is related to the encrypted plaintext. In general, security with respect to auxiliary information is essential towards obtaining composable security (see, for example, [11] and the references therein). More closely related to our approach are the studies of security with respect to auxiliary information in the contexts of perfect one-way functions [10], program obfuscation [19], and leakage-resilient encryption [13, 12, 8].

For example, when using a deterministic public-key encryption scheme for enabling efficient searches on encrypted databases, as suggested by Bellare et al. [2], it is not unlikely that the same plaintext belongs to more than one database, and is therefore encrypted under several public keys; or that various statistics of the database are publicly available. A more acute example is when using a deterministic public-key encryption scheme for a key-encapsulation mechanism that "hedges against bad randomness" [3]. In such a case an adversary that observes the usage of the encapsulated key (say, as a key to a symmetric-key encryption scheme) may in fact obtain a huge amount of additional information on the encapsulated key.

In this light, the notion of security proposed by Bellare et al. [2] might fall short of capturing the likely case where auxiliary information is available. That is, although a plaintext may be sampled from a distribution with high min-entropy to begin with, it might still have no entropy, from the point of view of an adversary, in many realistic scenarios. We note that already in the setting of deterministic *symmetric-key* encryption of high-entropy messages, Dodis and Smith [14] observed that the main weakness of an approach that does not take into account auxiliary information, is the lack of composable security. It is thus a highly desirable task to model and to construct secure deterministic encryption schemes in the setting of auxiliary information, as a crucial and essential step towards obtaining more realistic security guarantees.

1.1 Our Contributions

In this paper we introduce a framework for modeling the security of deterministic public-key encryption schemes with respect to auxiliary inputs. Within this framework we propose constructions that are based on standard cryptographic assumptions in the standard model (i.e., without random oracles). Our framework is a generalization of the one formalized by Bellare et al. [2] (and further studied in [4, 5, 23]) to the auxiliary-input setting, in which an adversary possibly obtains additional information that is related to the encrypted plaintext, and might even fully determine the encrypted plaintext information theoretically.

Modeling auxiliary information. An immediate consequence of having a deterministic encryption algorithm is that no meaningful notion of security can be satisfied if the plaintext can be recovered from the adversary's auxiliary information (see Section 3 for a discussion of this inherent constraint[4]). Thus, we focus our attention on the case of *hard-to-invert* auxiliary inputs, where the source of hardness may be any combination of information-theoretic hardness (where the auxiliary-input function is many-to-one) and computational hardness (where the auxiliary input function is injective, but is hard to invert by efficient algorithms).

Notions of security. Following [2, 4, 5] we formalize three notions of security with respect to auxiliary inputs, and prove that all three are equivalent. The first is a simulation-based notion, capturing the intuitive meaning of semantic security: whatever can be computed efficiently given a public key, an encryption of a message, and *hard-to-invert auxiliary input*, can be computed efficiently given only the public key and the auxiliary input. The second is a comparison-based notion, which essentially serves as an intermediate notion towards an indistinguishability-based one that is somewhat easier to handle in proofs of security. The high-level approach of the equivalence proofs is motivated by those of [4, 5], but the existence of auxiliary inputs that may fully determine the encrypted messages introduces various difficulties that our techniques overcome.

Constructions. We propose two constructions in the standard model satisfying our notions of security. At a first glance, one might hope that the constructions proposed in [2, 4, 5, 23] can be naturally extended to the auxiliary-input setting by replacing the notion of statistical min-entropy with an appropriate notion of computational min-entropy. This, however, does not seem to be the case (at least without relying on random oracles), as these constructions seem to heavily rely on information-theoretic properties that might not have natural computational analogues[5].

Our first construction is based on the decisional Diffie-Hellman assumption, (and more generally, on any of the d-linear assumptions), and our second construction is based on a rather general class of subgroup indistinguishability assumptions as defined in [8] (including, in particular, the quadratic residuosity assumption, and Paillier's composite residuosity assumption [24]). The resulting schemes are secure with respect to any auxiliary input that is subexponentially hard to invert[6]. In addition, our first scheme is secure even in the multi-user setting where related messages may be encrypted under multiple public keys. In this setting we obtain security (with respect to auxiliary inputs) for any polynomial number of messages and users as long as the messages are related by invertible linear transformations. Constructing a scheme that is secure is the

[4] This is somewhat similar to the observation that security is impossible to achieve when the plaintext is distributed over a small set.

[5] A prime example is the generalized crooked leftover hash lemma [5], for which a computational analogue may seem somewhat challenging to devise.

[6] We emphasize that in this paper we rely on standard computational assumptions (i.e., d-linear or quadratic residuosity), and only the auxiliary inputs are assumed to have subexponential hardness.

multi-user setting (even without considering auxiliary inputs) was identified as an important open problem by Bellare et al. [2]. Finally, we note that this scheme also exhibits an interesting homomorphic property: it allows homomorphic additions and one multiplication, in the spirit of [6, 16]. This property may be found especially useful in light of the possible applications of deterministic public-key encryption schemes in database systems [2].

1.2 Related Work

Exploiting the entropy of messages to prove otherwise-impossible security was first proposed by Russell and Wang [25], followed by Dodis and Smith [14]. These works achieved information-theoretic security for symmetric-key encryption with short keys.

In the setting of public-key encryption, deterministic encryption for high min-entropy messages was proposed by Bellare, Boldyreva, and O'Neill [2] who formalized a definitional framework, which was later refined and extended by Bellare, Fischlin, O'Neill, and Ristenpart [4], by Boldyreva, Fehr, and O'Neill [5], and by O'Neill [23]. Bellare et at. [2] presented two constructions in the random oracle model: The first relies on any semantically-secure public-key encryption scheme; whereas the second relies on the RSA function (and is in fact length preserving). Constructions in the standard model (i.e., without random oracles), were then presented in [4, 5]. Bellare et al. [4] presented a construction based on trapdoor permutations, which is secure as long as the messages are (almost) uniformly distributed. Boldyreva et al. [5] presented a construction based on lossy trapdoor functions, which is secure as long as its n-bit messages have min-entropy at least n^ϵ for some constant $0 < \epsilon < 1$. These constructions, however, fall short in two interesting cases: In the multi-message setting, where arbitrarily related messages are encrypted under the same public key; and in the multi-user setting where the same message is encrypted under several (independently chosen) public keys. Recently, O'Neill [23] made a step towards addressing the former, by presenting a scheme that can securely encrypt any fixed number q of messages, but whose parameters depend polynomially on q. The latter case remained unexplored until this work.

Deterministic public-key encryption was used by Bellare et al. [3] who defined and constructed "hedged" public-key encryption schemes. These are schemes that are semantically secure in the standard sense, and maintain a meaningful and realistic notion of security even when "corrupt" randomness is used for the encryption, so long as the joint message-randomness pair has sufficient min-entropy. The definition of security in the latter case takes after that of deterministic public-key encryption.

The tools underlying our constructions in this paper are inspired by the line of research on "encryption in the presence of auxiliary input", initiated by Dodis, Kalai, and Lovett [13] in the context of symmetric-key encryption, and then extended in [12, 8] to public-key encryption. These works consider encryption schemes where the adversary may obtain a hard-to-invert function of the secret

key — extending the frameworks of "bounded leakage" [1] and "noisy leakage" [22].

1.3 Overview of Our Approach

In this section we provide a high-level overview of our approach and techniques. We begin with a brief description of the notions of security that we consider in the auxiliary-input setting, and then describe the main ideas underlying our two constructions. For simplicity, in what follows we consider the case where one message is encrypted under one public key, and refer the reader to the relevant sections for the more general case.

Defining security with respect to auxiliary inputs. Towards describing our notions of security, we first discuss our notion of hard-to-invert auxiliary inputs. We consider any auxiliary input $f(x)$ from which it is hard to recover the input x. The source of hardness may be any combination of information-theoretic hardness (where the function f is many-to-one), and computational hardness (where $f(x)$ fully determines x, but x is hard to recover by efficient algorithms). Informally, we say that a function f is ϵ-hard-to-invert with respect to a distribution $\mathcal{D}$, if for every efficient algorithm A it holds that $A(f(x)) = x$ with probability at most ϵ, over the choice of $x \leftarrow \mathcal{D}$ and the internal coin tosses of A.

As discussed in Section 1.1, we formalize three notions of security with respect to auxiliary inputs, and prove that all three are equivalent. For concreteness we focus here on the simulation-based definition, which captures the intuitive meaning of semantic security: Whatever can be computed efficiently given a public key, an encryption of a message, and *hard-to-invert auxiliary input*, can be computed efficiently given only the public key and the auxiliary input. A bit more formally, we say that a scheme is secure with respect to ϵ-hard-to-invert auxiliary inputs if for any probabilistic polynomial-time adversary A, and for any efficiently samplable plaintext distribution $\mathcal{M}$, there exists a probabilistic polynomial-time simulator S, such that for any efficiently computable function f that is ϵ-hard-to-invert with respect to $\mathcal{M}$, and for any function $g \in \{0,1\}^* \to \{0,1\}^*$, the probabilities of the events $A\left(pk, \mathsf{Enc}_{pk}(m), f(m)\right) = g(m)$ and $S\left(pk, f(m)\right) = g(m)$ are negligibly close, where $m \leftarrow \mathcal{M}$. We note that the functions f and g may be arbitrary related[7]. This is a generalization of the definitions considered in [2, 4, 5, 23].

The scheme of Boldyreva et al. [5]. Our starting point is the scheme of Boldyreva et al. [5] that is based on lossy trapdoor functions. This is in fact the only known construction in the standard model (i.e., without random oracles) that is secure for arbitrary plaintext distributions with high (but not nearly full) min-entropy. In their construction, the public key consists of a function h that is sampled from the injective mode of the collection of lossy trapdoor

[7] In fact, the "target" function g is allowed to take as input also the randomness that is used for sampling m, and any other public randomness – see Section 3.

functions, and a pair-wise independent permutation π. The secret key consists of the trapdoor for inverting h (we assume that π is efficiently invertible). The encryption of a message m is defined as $\mathsf{Enc}_{pk}(m) = h(\pi(m))$, and decryption is naturally defined.

In a high level, the proof of security in [5] considers the joint distribution of the public key and the ciphertext $(pk, \mathsf{Enc}_{pk}(m))$, and argues that it is computationally indistinguishable from a distribution that is independent of the plaintext m. This is done by considering a distribution of malformed public keys, that is computationally indistinguishable from the real distribution. Specifically, the injective function h is replaced with a lossy function $\widetilde{h}$ to obtain an indistinguishable public key $\widetilde{pk}$. The next step is to show that the ciphertext $\tilde{c} = \mathsf{Enc}_{\widetilde{pk}}(m)$ can be described by the following two-step process. First, an analogue of a strong extractor is applied to m (where the seed is the permutation π that lies in $\widetilde{pk}$) to obtain $v = \mathsf{ext}_{\widetilde{pk}}(m)$. Then, the output of the extractor is used to compute the ciphertext $\tilde{c} = g(\widetilde{pk}, v)$. From this point of view, it is evident that so long as the plaintext m is drawn from a distribution with high min-entropy, it holds that $v = \mathsf{ext}_{\widetilde{pk}}(m)$ is statistically close to a uniform distribution (over some domain). This holds even given the malformed public key, and does not depend on the distribution of m. This methodology of using an analog of a strong extractor relies on the *crooked leftover hash lemma* of Dodis and Smith [14], that enables to base the construction on any collection of lossy trapdoor functions.

Our constructions. In our setting, we wish to adapt this methodology to rely on computational hardness instead of min-entropy. However, there is currently no known analog of the crooked leftover hash lemma in the computational setting. This is an interesting open problem. We overcome this difficulty by relying of specific collections of lossy trapdoor functions, for which we are in fact able to extract pseudorandomness from computational hardness. We do this by replacing the strong extractor component with a hard-core function of the message (with respect to the auxiliary input). Specifically, our encryption algorithm (when using the malformed public key) can be interpreted as taking an inner product between our message m (viewed as a vector of bits) and a random vector a, where the resulting ciphertext depends only on $(a, \langle m, a \rangle)$. This is similar to the Goldreich-Levin hard-core predicate [18], except that the vector a is not binary and the inner product is performed over some large $\mathbb{Z}$-module and not over the binary field. We thus require the generalized Goldreich-Levin theorem of Dodis et al. [12] to obtain that even given the auxiliary input, the distributions $(a, \langle m, a \rangle)$ and (a, u) are computationally indistinguishable, where u is uniformly distributed and does not depend on the distribution of m.

To be more concrete, let us consider our DDH-based scheme (formally presented in Section 4) which is based on the lossy trapdoor functions of Freeman et al. [15]. The scheme is instantiated by a DDH-hard group $\mathbb{G}$ of prime order q that is generated by g. The message space is $\{0,1\}^n$ (where n is polynomial in the security parameter) and the public key is $g^{\mathbf{A}}$, for a random $n \times n$ matrix $\mathbf{A}$

over $\mathbb{Z}_q$. Encryption is done by computing $\mathsf{Enc}_{g^{\mathbf{A}}}(\mathbf{m}) = g^{\mathbf{A}\cdot\mathbf{m}}$ and decryption is performed using $sk = \mathbf{A}^{-1}$.[8]

For analyzing the security of the scheme, we consider the joint distribution of the public key, ciphertext and auxiliary input $(pk, \mathsf{Enc}_{pk}(\mathbf{m}), f(\mathbf{m})) = (g^{\mathbf{A}}, g^{\mathbf{A}\cdot\mathbf{m}}, f(\mathbf{m}))$. The malformed distribution $\widetilde{pk}$ is obtained by taking $\mathbf{A}$ to be a random rank-1 matrix (rather than completely random). DDH implies that pk and $\widetilde{pk}$ are computationally indistinguishabile. Such a low-rank matrix takes the form $\mathbf{A} = \mathbf{r}\cdot\mathbf{b}^T$, and therefore $\mathbf{A}\cdot\mathbf{m} = \mathbf{r}\cdot\mathbf{b}^T\cdot\mathbf{m}$, for random vectors $\mathbf{r}$ and $\mathbf{b}$. Thus, our ciphertext depends only on $(\mathbf{b}, \langle\mathbf{b},\mathbf{m}\rangle)$ which is indistinguishable from $(\mathbf{b}, u)$, for a uniformly random u, even given $f(\mathbf{m})$, by the generalized Goldreich-Levin theorem [12]. Our initial distribution is therefore indistinguishable from the distribution $(g^{\mathbf{r}\cdot\mathbf{b}^T}, g^{\mathbf{r}\cdot u}, f(\mathbf{m}))$ as required.

In the multi-user setting, we observe that any polynomial number of public keys $g^{\mathbf{A}_1}, \ldots, g^{\mathbf{A}_\ell}$ are computationally indistinguishable, by DDH, from having *joint rank*-1. Namely, in this case the distributions $(g^{\mathbf{A}_1}, \ldots, g^{\mathbf{A}_\ell})$ and $(g^{\mathbf{r}_1\cdot\mathbf{b}^T}, \ldots, g^{\mathbf{r}_\ell\cdot\mathbf{b}^T})$ are computationally indistinguishable, where the same vector $\mathbf{b}$ is used for all keys. Encrypting a message $\mathbf{m}$ under all such ℓ public keys results in a set of ciphertexts $(g^{\mathbf{r}_1\cdot\mathbf{b}^T\cdot\mathbf{m}}, \ldots, g^{\mathbf{r}_\ell\cdot\mathbf{b}^T\cdot\mathbf{m}})$, where all elements depend on $(\mathbf{b}, \langle\mathbf{b},\mathbf{m}\rangle)$. This enables to apply the above approach, and we show that it in fact extends to linearly-related messages.

Our second scheme (based on subgroup indistinguishability assumptions) is analyzed quite similarly. We rely on the lossy trapdoor functions of [21] and can again show that our public key distribution is indistinguishable from one over rank-1 matrices. However, the groups under consideration might be non-cyclic. This adds additional complications into the analysis. In addition, this scheme does not seem to allow a "joint rank" argument as above, and we leave it as an open problem to construct an analogous scheme that is secure in the multi-user setting.

Paper organization. The remainder of this paper is organized as follows. In Section 2 we formalize a general notion for hard-to-invert auxiliary inputs. In Section 3 we introduce a framework for modeling the security of deterministic public-key encryption schemes with respect to auxiliary inputs, consisting of three main notions of security. In Section 4 we present a construction based on the decisional Diffie-Hellman assumption (and, more generally, on any of the d-linear assumptions), and in Section 5 we present a construction based on subgroup indistinguishability assumptions.

Due to space limitations, not all results and proofs appear in this extended abstract. We refer the reader to the full version of this paper [9] for more details.

Notation. Throughout the paper we denote scalars in plain lowercase letters ($x \in \{0,1\}$). We use the term "vector" both in the algebraic sense, where it indicates an element in a vector space and denoted by bold lowercase letters ($\mathbf{x} \in$

[8] We overload the notation g^x to matrices as follows: for $\mathbf{X} \in \mathbb{Z}_q^{k\times n}$, we let $g^{\mathbf{X}} \in \mathbb{G}^{k\times n}$ denote the matrix defined as $(g^{\mathbf{X}})_{i,j} = g^{(\mathbf{X})_{i,j}}$.

$\{0,1\}^k$); and in the "combinatorial" sense, indicating an ordered set of elements (not necessarily having any algebraic properties) for which we use the notation $\vec{x}$. We denote a combinatorial vector whose elements are algebraic vectors by $\vec{\mathbf{x}}$, combinatorial vector of combinatorial vectors by $\vec{\vec{x}}$, and combinatorial vector of combinatorial vectors of algebraic vectors by $\vec{\vec{\mathbf{x}}}$. Matrices (always algebraic) are denoted in bold uppercase ($\mathbf{X} \in \{0,1\}^{k \times n}$). The $k \times k$ identity matrix is denoted $\mathbf{I}_k$. All vectors are column vectors by default, and a row vector is denoted by $\mathbf{x}^T$.

2 Hard-to-Invert Auxiliary Inputs

In this work we consider any auxiliary input $f(x)$ from which it is hard to recover the input x. The source of hardness may be any combination of information-theoretic hardness (where the function f is many-to-one) and computational hardness (where $f(x)$ fully determines x, but x is hard to recover by efficient algorithms). Informally, we say that a function f is ϵ-hard-to-invert with respect to a distribution $\mathcal{D}$, if for every efficient algorithm A it holds that $A(f(x)) = x$ with probability at most ϵ over the choice of $x \leftarrow \mathcal{D}$ and the internal coin tosses of A.

For our purposes, we formalize a slightly more general notion in which $\mathcal{D}$ is a distribution over vectors of inputs $\vec{x} = (x_1, \ldots, x_t)$, and for every $i \in \{1, \ldots, t\}$ it should be hard to efficiently recover x_i when given $f(\vec{x})$. In addition, we also consider a *blockwise* variant of this notion, in which it should be hard to efficiently recover x_i when given $(x_1, \ldots, x_{i-1}, f(\vec{x}))$.

Definition 2.1. *An efficiently computable function $\mathcal{F} = \{f_k\}_{k \in \mathbb{N}}$ is $\epsilon(k)$-hard-to-invert with respect to an efficiently samplable distribution $\mathcal{D} = \{\mathcal{D}_k\}_{k \in \mathbb{N}}$ over vectors of $t(k)$ inputs, if for every probabilistic polynomial-time algorithm A and for every $i \in \{1, \ldots, t(k)\}$ it holds that*

$$\Pr\left[A\left(1^k, f_k(\vec{x})\right) = x_i\right] \leq \epsilon(k) \ ,$$

for all sufficiently large k, where the probability is taken over the choice of $\vec{x} = (x_1, \ldots, x_{t(k)}) \leftarrow \mathcal{D}_k$, and over the internal coin tosses of A.

Definition 2.2. *An efficiently computable function $\mathcal{F} = \{f_k\}_{k \in \mathbb{N}}$ is $\epsilon(k)$-blockwise-hard-to-invert with respect to an efficiently samplable distribution $\mathcal{D} = \{\mathcal{D}_k\}_{k \in \mathbb{N}}$ over vectors of $t(k)$ inputs, if for every probabilistic polynomial-time algorithm A and for every $i \in \{1, \ldots, t(k)\}$ it holds that*

$$\Pr\left[A\left(1^k, x_1, \ldots, x_{i-1}, f_k(\vec{x})\right) = x_i\right] \leq \epsilon(k) \ ,$$

for all sufficiently large k, where the probability is taken over the choice of $\vec{x} = (x_1, \ldots, x_{t(k)}) \leftarrow \mathcal{D}_k$, and over the internal coin tosses of A.

Note that Definition 2.1 implies in particular that the distribution $\mathcal{D}$ is such that each x_i has min-entropy at least $\log(1/\epsilon(k))$. Furthermore, Definition 2.2 implies that the distribution $\mathcal{D}$ is a block source in which each block x_i has (average) min-entropy at least $\log(1/\epsilon(k))$ conditioned on the previous blocks $(x_1, \ldots, x_{i-1})$.

3 Modeling Security in the Auxiliary-Input Setting

In this section we present a framework for modeling the security of deterministic public-key encryption schemes with respect to auxiliary inputs. Our framework is obtained as a generalization of those considered in [2, 4, 5] to a setting in which the encrypted plaintexts may be fully determined by some auxiliary information that is available to the adversary. Following [2, 4, 5] we formalize three notions of security with respect to auxiliary inputs, and prove that all three are equivalent. The first is a simulation-based semantic security notion (PRIV-SSS), capturing the intuitive meaning of semantic security: whatever can be computed given an encryption of a message and *auxiliary input*, can also be computed given only the auxiliary input. The second is a comparison-based semantic-security notion (PRIV-CSS), which essentially serves as an intermediate notion towards an indistinguishability-based one (PRIV-IND) that is somewhat easier to handle in proofs of security.

In the remainder of this paper we use the following notation. For a deterministic public-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, a public key pk, and a vector of messages $\vec{m} = (m_1, \ldots, m_t)$ we denote by $\vec{\mathsf{Enc}}_{pk}(\vec{m})$ the vector $(\mathsf{Enc}_{pk}(m_1), \ldots, \mathsf{Enc}_{pk}(m_t))$. When considering a distribution $\mathcal{M}$ over vectors of messages $\vec{m} = (m_1, \ldots, m_t)$ all of which are encrypted under the same public key, then for the case of hard-to-invert auxiliary inputs we make in this paper the simplifying assumption that $m_i \neq m_j$ for every $i \neq j$ (a bit more formally, one should require that all distributions have identical equality patterns – see [2]). In the case of blockwise-hard-to-invert auxiliary inputs this assumption is not necessary. In addition, for simplicity we present our definitions for the case of hard-to-invert auxiliary inputs, and note that they naturally extend to the case of blockwise-hard-to-invert auxiliary inputs.

Definition 3.1 (Simulation-based security). *A deterministic public-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is* PRIV-SSS-secure with respect to ϵ-hard-to-invert auxiliary inputs *if for any probabilistic polynomial-time algorithm A and for any efficiently samplable distribution $\mathcal{M} = \{\mathcal{M}_k\}_{k\in\mathbb{N}}$, there exists a probabilistic polynomial-time algorithm S, such that for any efficiently computable function $\mathcal{F} = \{f_k\}_{k\in\mathbb{N}}$ that is ϵ-hard-to-invert with respect to $\mathcal{M}$, and for any function $g \in \{0,1\}^* \to \{0,1\}^*$, there exists a negligible function $\nu(k)$ such that*

$$\mathrm{Adv}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},S,\mathcal{F},g}(k) \stackrel{\mathsf{def}}{=} \left|\mathrm{Real}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k) - \mathrm{Ideal}^{\mathsf{PRIV-SSS}}_{\Pi,S,\mathcal{M},\mathcal{F},g}(k)\right| \leq \nu(k)$$

for all sufficiently large k, where

$$\begin{aligned}
\mathrm{Real}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k) &= \Pr\left[A\left(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}), f_k(\vec{m})\right) = g(\vec{m})\right] \\
\mathrm{Ideal}^{\mathsf{PRIV-SSS}}_{\Pi,S,\mathcal{M},\mathcal{F},g}(k) &= \Pr\left[S\left(1^k, pk, f_k(\vec{m})\right) = g(\vec{m})\right] \quad ,
\end{aligned}$$

and the probability is taken over the choices of $\vec{m} \leftarrow \mathcal{M}_k$, $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$, and over the internal coin tosses of A and S.

Definition 3.2 (Comparison-based security). *A deterministic public-key encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* PRIV-CSS-secure with respect to ϵ-hard-to-invert auxiliary inputs *if for any probabilistic polynomial-time algorithm* A*, for any efficiently samplable distribution* $\mathcal{M} = \{\mathcal{M}_k\}_{k\in\mathbb{N}}$*, for any efficiently computable function* $\mathcal{F} = \{f_k\}_{k\in\mathbb{N}}$ *that is* ϵ*-hard-to-invert with respect to* $\mathcal{M}$*, and for any function* $g \in \{0,1\}^* \to \{0,1\}^*$*, there exists a negligible function* $\nu(k)$ *such that*

$$\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k) \stackrel{\mathsf{def}}{=} \left|\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k,0) - \mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k,1)\right| \leq \nu(k)$$

for all sufficiently large k*, where*

$$\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k,b) = \Pr\left[A\left(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_b), f_k(\vec{m}_0)\right) = g(\vec{m}_0)\right] \ ,$$

and the probability is taken over the choices of $\vec{m}_0 \leftarrow \mathcal{M}_k$*,* $\vec{m}_1 \leftarrow \mathcal{M}_k$*,* $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$*, and over the internal coin tosses of* A*.*

Definition 3.3 (Indistinguishability-based security). *A deterministic public-key encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* PRIV-IND-secure with respect to ϵ-hard-to-invert auxiliary inputs *if for any probabilistic polynomial-time algorithm* A*, for any two efficiently samplable distributions* $\mathcal{M}_0 = \{\mathcal{M}_{0,k}\}_{k\in\mathbb{N}}$ *and* $\mathcal{M}_1 = \{\mathcal{M}_{1,k}\}_{k\in\mathbb{N}}$*, and for any efficiently computable function* $\mathcal{F} = \{f_k\}_{k\in\mathbb{N}}$ *that is* ϵ*-hard-to-invert with respect to both* $\mathcal{M}_0$ *and* $\mathcal{M}_1$*, there exists a negligible function* $\nu(k)$ *such that*

$$\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k) \stackrel{\mathsf{def}}{=} \left|\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k,0) - \mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k,1)\right| \leq \nu(k)$$

for all sufficiently large k*, where*

$$\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k,b) = \Pr\left[A\left(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_b), f_k(\vec{m}_0)\right) = 1\right] \ ,$$

and the probability is taken over the choices of $\vec{m}_0 \leftarrow \mathcal{M}_{0,k}$*,* $\vec{m}_1 \leftarrow \mathcal{M}_{1,k}$*,* $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$*, and over the internal coin tosses of* A*.*

The hard-to-invert requirement. We emphasize that in the setting of *deterministic* public-key encryption the requirement that the encrypted messages cannot be efficiently recovered from the auxiliary input is essential (unlike in the setting of *randomized* encryption, where the notion of semantic security takes into account any auxiliary input – see, for example, [17, Ch. 5]). This is easily observed using our indistinguishability-based formulation (Definition 3.3): an algorithm that on input $f_k(\vec{m}_0)$ (where $\vec{m}_0 = (m_{0,1}, \ldots, m_{0,t(k)})$) can recover one of the $m_{0,i}$ values can then encrypt this value under pk, compare the resulting ciphertext with the i-th component of $\vec{\mathsf{Enc}}_{pk}(\vec{m}_b)$, and thus learn the bit b.

Relation to previous notions. We note that any *constant function* is ϵ-hard-to-invert with respect to any message distribution of min-entropy at least

$\log(1/\epsilon)$. Thus, our notion of auxiliary-input security strictly generalizes previous security notions, in which auxiliary input is not considered, and the message distributions need to have sufficient min-entropy [2, 4, 5, 23].

Access to the public key. As observed by Bellare et al. [2] it is essential that the "target" function g does not take the public key as input. Specifically, with a deterministic encryption algorithm the ciphertext itself is a non-trivial information that it leaked about the plaintext, and can clearly be computed efficiently using the public key. We refer the reader to [2] for a more elaborated discussion.

The randomness of sampling. For our notions of security we in fact allow the auxiliary-input function f and the "target" function g to take as input not only the vector of message $\vec{m}$, but also the random string $r \in \{0,1\}^*$ that was used for sampling $\vec{m}$ from the distribution $\mathcal{D}_k$. When this aspect plays a significant role we explicitly include r as part of the input for f and g, and denote by $\vec{m} \leftarrow \mathcal{D}_k(r)$ the fact that $\vec{m}$ is sampled using the random string r. When this aspect does not play a significant role we omit it for ease of readability (in particular, we omitted it from the above definitions).

PRIV1: focusing on a single message. As in [5] we also consider the PRIV1-variants of our notion of security that focus on a single message (instead of vectors of any polynomial number of messages). In the full version [9] we also provide proof that security for a vector of messages with respect to a blockwise-hard-to-invert auxiliary input is in fact equivalent to security for a single message with respect to a hard-to-invert auxiliary input.

The multi-user setting. So far our notions of security considered vectors of messages that are encrypted under the same public key. Our definitions in this section naturally generalize to the multi-user setting, where there are multiple public keys, each of which is used for encrypting a vector of messages. Due to space limitations, we refer the reader to the full version [9] for this generalization.

An even stronger notion of security. Note that in Definition 3.3 the algorithm A is given as input the vector $\left(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_b), f_k(\vec{m}_0)\right)$, and that a seemingly stronger definition would even consider the vector

$$\left(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_b), \vec{\mathsf{Enc}}_{pk}(\vec{m}_{1-b}), f_k(\vec{m}_0), f_k(\vec{m}_1)\right)$$

as its input. As indicated by the equivalence of our three definitions, such a stronger variant is not needed for capturing the intuitive meaning of semantic security as in Definition 3.1. Nevertheless, our schemes in this paper in fact satisfy this stronger variant. We refer to this notion as *strong indistinguishability* (PRIV-sIND), formally defined as follows:

Definition 3.4. *A deterministic public-key encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* PRIV-sIND-secure with respect to ϵ-hard-to-invert auxiliary inputs

if for any probabilistic polynomial-time algorithm A, for any two efficiently samplable distributions $\mathcal{M}_0 = \{\mathcal{M}_{0,k}\}_{k\in\mathbb{N}}$ and $\mathcal{M}_1 = \{\mathcal{M}_{1,k}\}_{k\in\mathbb{N}}$, and for any efficiently computable function $\mathcal{F} = \{f_k\}_{k\in\mathbb{N}}$ that is ϵ-hard-to-invert with respect to both $\mathcal{M}_0$ and $\mathcal{M}_1$, there exists a negligible function $\nu(k)$ such that

$$\mathrm{Adv}^{\mathsf{PRIV-sIND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k) \stackrel{\mathsf{def}}{=} \left|\mathrm{Adv}^{\mathsf{PRIV-sIND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k,0) - \mathrm{Adv}^{\mathsf{PRIV-sIND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k,1)\right| \leq \nu(k)$$

for all sufficiently large k, where

$$\begin{aligned}&\mathrm{Adv}^{\mathsf{PRIV-sIND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k,b)\\ &\quad = \Pr\left[A\left(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_b), \vec{\mathsf{Enc}}_{pk}(\vec{m}_{1-b}), f_k(\vec{m}_0), f_k(\vec{m}_1)\right) = 1\right] \ ,\end{aligned}$$

and the probability is taken over the choices of $\vec{m}_0 \leftarrow \mathcal{M}_{0,k}$, $\vec{m}_1 \leftarrow \mathcal{M}_{1,k}$, $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$, and over the internal coin tosses of A.

4 A Scheme Based on the *d*-Linear Assumption

In this section we present our d-linear-based based deterministic encryption scheme and discuss its properties. We show that the d-linear-based lossy trapdoor function of Freeman et al. [15] is in fact a deterministic public-key encryption that is secure with respect to hard-to-invert auxiliary inputs.

The scheme $\Pi_{\mathbf{Lin}}$. Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^k, and outputs a triplet $(\mathbb{G}, q, g)$ where $\mathbb{G}$ is a group of prime order q that is generated by $g \in \mathbb{G}$, and q is a k-bit prime number. For describing the scheme we overload the notation g^x to matrices: for $\mathbf{X} \in \mathbb{M}^{k\times n}$, we let $g^{\mathbf{X}} \in \mathbb{G}^{k\times n}$ denote the matrix defined as $(g^{\mathbf{X}})_{i,j} = g^{(\mathbf{X})_{i,j}}$. The scheme is parameterized by the security parameter k and the message length $n = n(k)$.

- **Key generation.** The key-generation algorithm $\mathsf{KeyGen}(1^k)$ samples $(\mathbb{G}, q, g) \leftarrow \mathsf{GroupGen}(1^k)$, and a matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times n}$. It then outputs $pk = (\mathbb{G}, q, g, g^{\mathbf{A}})$ and $sk = \mathbf{A}^{-1}$ (note that $\mathbf{A}$ is invertible with all but a negligible probability).
- **Encryption.** The encryption algorithm $\mathsf{Enc}_{pk}(\mathbf{m})$, where $\mathbf{m} \in \{0,1\}^n \subseteq \mathbb{Z}_q^n$, outputs the ciphertext $g^{\mathbf{c}} = g^{\mathbf{A}\cdot\mathbf{m}}$.
- **Decryption.** The decryption algorithm $\mathsf{Dec}_{sk}(g^{\mathbf{c}})$, where $g^{\mathbf{c}} \in \mathbb{G}^n$, first computes $g^{\mathbf{m}} = g^{\mathbf{A}^{-1}\cdot\mathbf{c}}$. Then, note that if $\mathbf{m} \in \{0,1\}^n$ then it can be efficiently extracted from $g^{\mathbf{m}}$. In such case it outputs $\mathbf{m}$, and otherwise it outputs $\perp$.

Correctness follows immediately as in [15]. We prove the following theorem:

Theorem 4.1. *Let $d \in \mathbb{N}$ be some integer. Then under the d-linear assumption, for any constant $0 < \mu < 1$ and for any sufficiently large message length $n = n(k)$, the scheme Π_{Lin} is PRIV-IND-secure with respect to $2^{-n^{\mu}}$-blockwise-hard-to-invert auxiliary inputs.*

Due to space limitations, we only describe the main ideas underlying the security of the scheme. The full proof can be found in the full version [9]. For simplicity, we focus here on the case $d = 1$ (i.e., we rely on the DDH assumption). Given a distribution $\mathcal{M}$ over messages $\mathbf{m} \in \{0,1\}^n$, and an auxiliary-input function f that is sub-exponentially hard to invert with respect to $\mathcal{M}$, we argue that an encryption of a messages $\mathbf{m}$ sampled from the distribution $\mathcal{M}$ is computationally indistinguishable from being completely independent of the public key pk and the auxiliary input $f(\mathbf{m})$. More specifically, we prove that $(pk, \mathsf{Enc}_{pk}(\mathbf{m}), f(\mathbf{m})) \stackrel{c}{\approx} (pk, g^{\mathbf{u}}, f(\mathbf{m}))$, for a uniformly chosen vector $\mathbf{u}$. Transforming this into either one of our notions of security from Section 3 is rather standard.

Consider the joint distribution $(pk, \mathsf{Enc}_{pk}(\mathbf{m}), f(\mathbf{m})) = (g^{\mathbf{A}}, g^{\mathbf{A}\cdot\mathbf{m}}, f(\mathbf{m}))$ of the public key, the ciphertext, and the auxiliary input. The DDH assumption implies that replacing the uniformly chosen matrix $\mathbf{A}$ with a random matrix of rank 1 results in a computationally indistinguishable distribution. Such a low-rank matrix can be written as $\mathbf{A} = \mathbf{r} \cdot \mathbf{b}^T$, for random vectors $\mathbf{r}$ and $\mathbf{b}$, and therefore $\mathbf{A} \cdot \mathbf{m} = \mathbf{r} \cdot \mathbf{b}^T \cdot \mathbf{m}$. However $\mathbf{b}^T \cdot \mathbf{m} = \langle \mathbf{b}, \mathbf{m} \rangle$ is indistinguishable from the uniform distribution, even given $\mathbf{b}$ and $f(\mathbf{m})$, according to the generalized Goldreich-Levin theorem of [12]. Our initial distribution is thus indistinguishable from the distribution $(g^{\mathbf{r}\cdot\mathbf{b}^T}, g^{\mathbf{r}\cdot\alpha}, f(\mathbf{m}))$.

Now, notice that the matrix $[\mathbf{r} \cdot \mathbf{b}^T \| \mathbf{r} \cdot \alpha] \in \mathbb{Z}_q^{n\times(n+1)}$ is essentially a random matrix of rank 1. Relying on the DDH assumption once again, it can be replaced with a completely random matrix while preserving computational indistinguishability. This yields the distribution $(g^{\mathbf{A}}, g^{\mathbf{u}}, f(\mathbf{m}))$, where $\mathbf{A}$ and $\mathbf{u}$ are chosen uniformly at random.

Homomorphic properties. The scheme naturally exhibits homomorphic properties w.r.t. multiplication by a scalar or addition of two ciphertexts over $\mathbb{Z}_q^n$. This follows from "arithmetics in the exponent". We stress, however, that the output of such homomorphic operations will be decryptable if it lies in the message space of our scheme, $\{0,1\}^n$, which is a proper subset of the domain $\mathbb{Z}_q^n$ on which these operations are performed. More generally, decryption is possible as long as each entry of the encrypted plaintext vector belongs to a predetermined set of logarithmic size.

In addition, if the underlying group $\mathbb{G}$ is associated with a *bilinear map*, then our scheme enjoys an additional homomorphism w.r.t. *one* matrix multiplication. This is similar to the homomorphism style achieved in [6] and in [16]. We stress that in such case we base the security of the scheme on the d-linear assumption for $d \geq 2$ (as the 1-linear, i.e. DDH, cannot hold in such a group). Formally, let $\mathbb{G}$, q, and g be as in the parameters of our scheme, and let $\mathbb{G}_T$ be a (different) group of order q. A bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has the following properties. *Bilinearity:* for all $x, y \in \mathbb{G}$, $a, b \in \mathbb{Z}$ it holds that $e(x^a, y^b) = e(x, y)^{ab}$; *Non-degeneracy:* $e(g, g) \neq 1$. It follows that $g_T \stackrel{\mathsf{def}}{=} e(g, g)$ generates $\mathbb{G}_T$.

Homomorphic matrix multiplication, thus, is performed in our scheme as follows: Given two ciphertexts $g^{\mathbf{A}\mathbf{m}_1}$ and $g^{\mathbf{A}\mathbf{m}_2}$, one can compute $e(g, g)^{\mathbf{A}\mathbf{m}_1\mathbf{m}_2^T\mathbf{A}^T}$.

This ciphertext can be decrypted by multiplying by $\mathbf{A}^{-1}$ from the left (in the exponent) and $\mathbf{A}^{-T}$ from the right (again, in the exponent) to obtain $e(g,g)^{\mathbf{m}_1 \cdot \mathbf{m}_2^T}$. Since $\mathbf{m}_1$ and $\mathbf{m}_2$ are binary, $\mathbf{m}_1 \cdot \mathbf{m}_2^T$ is binary as well and can be extracted from the exponent.

The multi-user setting. We now show that Π_{Lin} is secure (with respect to auxiliary inputs) even in the multi-user setting, where related messages may be encrypted under multiple public keys. We allow any polynomial number of users, and for simplicity we assume that each public key encrypts one message. As in the single-user setting, this natural extends to the case where several messages are encrypted under each public key with blockwise-hard-to-invert auxiliary input. In addition, we require that the messages to be encrypted come from an *affine distribution*, a term we define below. Intuitively, this means that there are publicly known invertible linear relations (over $\mathbb{Z}_q^n$) between the messages.

Definition 4.2 (Affine message distributions). *Let $n = n(k)$ and $\ell = \ell(k)$ be integer functions of the security parameter, and let $\mathcal{M} = \{\mathcal{M}\}_k \subseteq (\{0,1\}^n)^\ell$ be a distribution ensemble.*[9] *Then $\mathcal{M}$ is* affine *if there exist invertible and efficiently computable (given k) matrices $\mathbf{V}_2, \ldots, \mathbf{V}_\ell \subseteq \mathbb{Z}_q^{n \times n}$ and vectors $\mathbf{w}_2, \ldots, \mathbf{w}_\ell \in \mathbb{Z}_q^n$, such that for all $(\mathbf{m}_1, \ldots, \mathbf{m}_\ell)$ in the support of $\mathcal{M}$ and for all $i \in \{2, \ldots, \ell\}$ it holds that $\mathbf{m}_i = \mathbf{V}_i \cdot \mathbf{m}_1 + \mathbf{w}_i$ (where arithmetics is over $\mathbb{Z}_q$).*

Note that we require that messages are taken over the space $\{0,1\}^n$, and arithmetics is over $\mathbb{Z}_q$. In particular, this captures the case of "broadcast encryption" where encrypting the same message under many public keys. Furthermore, this also captures XORing with a constant vector over the binary field, or permuting the coordinates of a binary vector (a tool used, e.g., in [7]). The result is formally stated below. For proof, see full version [9].

Theorem 4.3. *Let $d \in \mathbb{N}$ be some integer. Then under the d-linear assumption, for any constant $0 < \mu < 1$ and for any sufficiently large message length $n = n(k)$, the scheme Π_{Lin} is PRIV1-IND-MU-secure with respect to 2^{-n^μ}-hard-to-invert auxiliary inputs.*

5 A Scheme Based on Subgroup Indistinguishability Assumptions

In this section we present our second deterministic encryption scheme, which is based on a rather general class of subgroup indistinguishability. For concreteness we first describe the scheme based on the quadratic residuosity assumption, and then describe the more general case. We show that (a slight generalization of) the QR-based lossy trapdoor function of Hemenway and Ostrovsky [21] is in fact a deterministic public-key encryption scheme that is secure against sub-exponentially hard-to-invert auxiliary inputs.

[9] To be absolutely precise should write that $\mathcal{M}_k$ is a distributions over $((\{0,1\}^n)^t)^\ell$ for $t = 1$, but this space is trivially isomorphic to the one we consider.

The scheme $\Pi_{\mathbf{QR}}$. Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^k, and outputs an integer $N = PQ$, where P and Q are k-bit prime numbers, and $P \pmod 4 = Q \pmod 4 = 3$ (i.e., N is a Blum integer). In addition, recall that $y \leftarrow g^x$ denotes an application of an isomorphism transforming an element x in the module $\mathbb{M}_{\mathbb{QR}_N}$ into an element y in the group $\mathbb{QR}_N$ (since we will never express elements in the module explicitly, we do not care which isomorphism is used). We let $\hat{g}$ denote the isomorphism between the group $\mathbb{J}_N$ and the corresponding module, such that the generating set that corresponds to $\hat{g}$ is the same as that of g, appended with (-1). The scheme is parameterized by the security parameter k and the message length $n = n(k)$.

- **Key generation.** The key-generation algorithm $\mathsf{KeyGen}(1^k)$ samples $N \leftarrow \mathsf{GroupGen}(1^k)$, a vector $g^{\mathbf{w}^T} \leftarrow \mathbb{QR}_N^n$, and a vector $\mathbf{r} \leftarrow \left([N^2]\right)^n$. It then outputs $pk = (N, g^{\mathbf{w}^T}, (-1)^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T})$ and $sk = \mathbf{r}$.
  The matrix dot product above refers to element-wise multiplication:

$$\left((-1)^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T}\right)_{i,j} = \left((-1)^{\mathbf{I}_n}\right)_{i,j} \cdot \left(g^{\mathbf{r}\cdot\mathbf{w}^T}\right)_{i,j} .$$

  To be completely explicit, we emphasize that $pk \in \mathbb{N} \times \mathbb{J}_N^{1\times N} \times \mathbb{J}_N^{n\times n}$ and $sk \in \mathbb{N}^n$.
- **Encryption.** The encryption algorithm $\mathsf{Enc}_{pk}(\mathbf{m})$, where $pk = (N, \hat{g}^{\mathbf{w}^T}, \hat{g}^{\mathbf{T}})$ and $\mathbf{m} \in \{0,1\}^n$, outputs the ciphertext $c = (\hat{g}^{\mathbf{w}^T\cdot\mathbf{m}}, \hat{g}^{\mathbf{T}\cdot\mathbf{m}})$. We note that this computation can be performed efficiently and that $c \in \mathbb{J}_N \times \mathbb{J}_N^n$.
  For a legally generated public key $pk = (N, g^{\mathbf{w}^T}, (-1)^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T})$ and $sk = \mathbf{r}$, we get $c = (g^{\mathbf{w}^T\cdot\mathbf{m}}, (-1)^{\mathbf{m}} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T\cdot\mathbf{m}})$.
- **Decryption.** The decryption algorithm $\mathsf{Dec}_{sk}(c)$, where $c = (\hat{g}^v, \hat{g}^{\mathbf{y}})$, first computes $\hat{g}^{(\mathbf{y}-\mathbf{r}\cdot v)}$. If the output is of the form $(-1)^{\mathbf{m}}$, for $\mathbf{m} \in \{0,1\}^n$, then it outputs $\mathbf{m}$ and otherwise it outputs $\perp$.

Correctness follows immediately by definition. Security is stated below. The proof appears in the full version [9].

Theorem 5.1. *Under the quadratic residuosity assumption, for any constant $0 < \mu < 1$ and for any sufficiently large message length $n = n(k)$, the scheme Π_{QR} is PRIV-IND-secure with respect to 2^{-n^μ}-blockwise-hard-to-invert auxiliary inputs.*

Extension to general subgroup indistinguishability. As mentioned above, this construction can be extended to general subgroup indistinguishability assumptions [8] (these include, in particular, Paillier's composite residuosity assumption [24]). These assumptions are defined in a setting where $\mathbb{G}_U = \mathbb{G}_M \times \mathbb{G}_L$ is a product group such that $\mathbb{G}_U$ and $\mathbb{G}_L$ are computationally indistinguishable (there are of course additional requirements, we refer the reader to [8] for details). Specifically, quadratic residuosity fits into this setting by letting $\mathbb{G}_U = \mathbb{J}_N$,

$\mathbb{G}_L = \mathbb{QR}_N$, $\mathbb{G}_M = \{\pm 1\}$. To generalize our construction, we let $\mathbb{M}$ be the module that corresponds to $\mathbb{G}_L$ and replace (-1) with a generator h of $\mathbb{G}_M$. Namely, our keys become $pk = (g^{\mathbf{w}^T}, h^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T})$ and $sk = \mathbf{r}$; encryption of a message $\mathbf{m} \in \{0,1\}^n$ is done by computing $c = (g^{\mathbf{w}^T\cdot\mathbf{m}}, h^{\mathbf{m}} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T\cdot\mathbf{m}})$; and decryption of $c = (\hat{g}^v, \hat{g}^{\mathbf{y}})$ is done by computing $\hat{g}^{(\mathbf{y}-\mathbf{r}\cdot v)} = h^{\mathbf{m}}$ and extracting $\mathbf{m}$. The proof of security in this case is similar to that of the QR-based scheme.

References

1. A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *Proceedings of the 6th Theory of Cryptography Conference*, pages 474–495, 2009.
2. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *Advances in Cryptology – CRYPTO '07*, pages 535–552, 2007.
3. M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In *Advances in Cryptology – ASIACRYPT '09*, pages 232–249, 2009.
4. M. Bellare, M. Fischlin, A. O'Neill, and T. Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *Advances in Cryptology – CRYPTO '08*, pages 360–378, 2008.
5. A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Advances in Cryptology – CRYPTO '08*, pages 335–359, 2008.
6. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *Proceedings on the 2nd Theory of Cryptography Conference*, pages 325–341, 2005.
7. D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *Advances in Cryptology – CRYPTO '08*, pages 108–125, 2008.
8. Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *Advances in Cryptology – CRYPTO '10*, 2010.
9. Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. Cryptology ePrint Archive, Report 2011/209, 2011.
10. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology – CRYPTO '97*, pages 455–469, 1997.
11. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 136–145, 2001.
12. Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *Proceedings of the 7th Theory of Cryptography Conference*, pages 361–381, 2010.
13. Y. Dodis, Y. T. Kalai, and S. Lovett. On cryptography with auxiliary input. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 621–630, 2009.
14. Y. Dodis and A. Smith. Entropic security and the encryption of high entropy messages. In *Proceedings of the 2nd Theory of Cryptography Conference*, pages 556–577, 2005.

15. D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. In *Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography*, pages 279–295, 2010.
16. C. Gentry, S. Halevi, and V. Vaikuntanathan. A simple BGN-type cryptosystem from LWE. In *Advances in Cryptology – EUROCRYPT '10*, pages 506–522, 2010.
17. O. Goldreich. *Foundations of Cryptography – Volume 2: Basic Applications*. Cambridge University Press, 2004.
18. O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25–32, 1989.
19. S. Goldwasser and Y. T. Kalai. On the impossibility of obfuscation with auxiliary input. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 553–562, 2005.
20. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
21. B. Hemenway and R. Ostrovsky. Lossy trapdoor functions from smooth homomorphic hash proof systems. Electronic Colloquium on Computational Complexity, Report TR09-127, 2009.
22. M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology – CRYPTO '09*, pages 18–35, 2009.
23. A. O'Neill. Deterministic public-key encryption revisited. Cryptology ePrint Archive, Report 2010/533, 2010.
24. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT '99*, pages 223–238, 1999.
25. A. Russell and H. Wang. How to fool an unbounded adversary with a short key. *IEEE Transactions on Information Theory*, 52(3):1130–1140, 2006.
26. B. Zhu, K. Li, and R. H. Patterson. Avoiding the disk bottleneck in the data domain deduplication file system. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies*, pages 269–282, 2008.