J. Cryptol. (2014) 27: 210–247
DOI: 10.1007/s00145-012-9143-4

Journal of
CRYPTOLOGY

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

Zvika Brakerski[†] and Gil Segev[‡]
Stanford University, Stanford, CA 94305, USA
zvika@stanford.edu; segev@stanford.edu

Communicated by Reingold.

Received 19 April 2012
Online publication 11 January 2013

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 d-linear assumption for any $d \geq 1$ (including, in particular, the decisional Diffie–Hellman assumption), and the second is based on a rather general class of subgroup indistinguishability assumptions (including, in particular, the quadratic residuosity assumption and Paillier's composite residuosity assumption). 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.

[*] A preliminary version of this work appeared in *Advances in Cryptology, CRYPTO '11*, pp. 543–560, 2011.

[†] Most of the work was done while Z. Brakerski was a Ph.D. student at the Weizmann Institute of Science.

[‡] Most of the work was done while G. Segev was a Postdoctoral Researcher at Microsoft Research Silicon Valley.

Key words. Public-key encryption, deterministic encryption, auxiliary inputs, composable security.

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 [19]. 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 ratio is crucial, or global deduplication-based storage systems where searches are highly frequent (e.g. [29]). 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, [2] and the references therein).

Deterministic Public-Key Encryption For dealing with this kind of drawbacks, Bellare, Boldyreva, and O'Neill [3] initiated the study of *deterministic* public-key encryption schemes. These are public-key encryption schemes in which the encryption algorithm is deterministic.[1] 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,

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

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. [3], and then in the standard model by Bellare, Fischlin, O'Neill, and Ristenpart [4], by Boldyreva, Fehr, and O'Neill [5], by Fuller, O'Neill and Reyzin [15], by Mironov, Pandey, Reingold and Segev [22], and by Wee [27]. We refer the reader to Sects. 1.2 and 1.4 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, [10] 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 [9], program obfuscation [17], and leakage-resilient encryption [6, 11,12].

For example, when using a deterministic public-key encryption scheme for enabling efficient searches on encrypted databases, as suggested by Bellare et al. [3], 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" [2]. 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. [3] 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 [13] 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. [3] (and further studied in [4,5,15,22]) 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 Sect. 4 for a discussion of this inherent constraint[2]). 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 Refs. [3–5] 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 [3–5] 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 analogs.[3]

Our first construction is based on the d-linear assumption for any $d \geq 1$ (including, in particular, the decisional Diffie–Hellman assumption), and our second construction is based on a rather general class of subgroup indistinguishability assumptions as defined in [6] (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.[4] Moreover, 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)

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

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

[4] 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.

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 multi-user setting (even without considering auxiliary inputs) was identified as an important open problem by Bellare et al. [3].

Finally, we note that if we assume that the group under consideration is equipped with a bilinear map, then our first scheme exhibits an interesting homomorphic property: it allows homomorphic additions and one multiplication, in the spirit of Refs. [7,16]. This property may be found especially useful in light of the possible applications of deterministic public-key encryption schemes in database systems [3].

1.2. *Related Work*

Exploiting the entropy of messages to prove otherwise-impossible security was first proposed by Russell and Wang [26], followed by Dodis and Smith [13]. 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 [3] who formalized a definitional framework, which was later refined and extended by Bellare, Fischlin, O'Neill, and Ristenpart [4], and by Boldyreva, Fehr, and O'Neill [5]. Bellare et at. [3] 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 first 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. Fuller, O'Neill and Reyzin [15] 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. Additional progress in studying deterministic public-key encryption schemes was recently made by Mironov, Pandey, Reingold and Segev [22] who constructed such schemes with optimal incrementality: small changes in the plaintext translate into small changes in the corresponding ciphertext.

Deterministic public-key encryption was used by Bellare et al. [2] 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, Tauman Kalai, and Lovett [12] in the context of symmetric-key encryption, and then extended in [6,11] 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” [23].

Finally, we note that Wichs [28] has recently proved a strong impossibility result for deterministic public-key encryption, showing that the strongest notion of security cannot be achieved based on standard cryptographic assumptions while treating adversaries as black boxes. The strongest notion of security considers the encryption of plaintexts which may be arbitrarily correlated. His impossibility result does not apply to our constructions, where we do not allow arbitrary correlations (see Sect. 3 for our definition of *blockwise* hard-to-invert auxiliary inputs).

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 which follows the framework of Dodis, Tauman Kalai, and Lovett [12]. 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 Sect. 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 efficiently computable function $g \in \{0,1\}^* \to \{0,1\}^*$, the probabilities of the events $A(pk, \mathsf{Enc}_{pk}(m), f(m)) = g(m)$ and $S(pk, f(m)) = g(m)$ are negligibly close, where $m \leftarrow \mathcal{M}$. We note that the functions f and g may be arbitrary related.[5] This is a generalization of the definitions considered in [3–5,15,22].

The Boldyreva et al. [5] Scheme 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

[5] 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 Sect. 4.

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 functions, and a pairwise-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 analog 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 [13], that enables basing 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 pseudo-randomness 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 public 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 group (or, more accurately, over a $\mathbb{Z}$-module) and not over the binary field. We thus require the generalized Goldreich–Levin theorem of Dodis et al. [11] 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 Sect. 6) which is based on the lossy trapdoor functions of Freeman et al. [14]. 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 $\mathbb{Z}_q$.[6] Encryption is done by computing $\mathsf{Enc}_{g^{\mathbf{A}}}(\mathbf{m}) = g^{\mathbf{A}\cdot\mathbf{m}}$ and decryption is performed using $sk = \mathbf{A}^{-1}$ (note that such a matrix $\mathbf{A}$ is indeed invertible with high probability).

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 indistinguishable. 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 [11]. 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. Note that we use the generalized Goldreich–Levin theorem for extracting an element of $\mathbb{Z}_q$, and this requires the hardness of inverting the auxiliary input f to be roughly proportional to $1/q$, which can be made subexponential in the security parameter by choosing an appropriate message length (see Sect. 2.2 for more details).

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 applying the above approach, and we show that it in fact extends to linearly-related messages. More specifically, we show that our notions of security are satisfied even when any polynomial number of public keys are used for encrypting different messages, as long as there are publicly-known invertible linear relation between the messages.

Our second scheme (based on subgroup indistinguishability assumptions) is analyzed quite similarly. We rely on the lossy trapdoor functions of Hemenway and Ostrovsky [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 noncyclic. This adds additional complications to 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.[7]

1.4. *Open Problems and Subsequent Work*

Our work raises two natural open problems. The first problem is to construct deterministic public-key encryption schemes that are secure with respect to *any* hard-to-invert auxiliary input (while, of course, assuming the *standard* hardness of the underlying computational assumptions). In our constructions, the assumption that the auxiliary input is *subexponentially* hard to invert seems somewhat essential given that in our proofs

[6] 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 $(g^{\mathbf{X}})_{i,j} = g^{(\mathbf{X})_{i,j}}$.

[7] We remark that a similar issue came up in [6], and prevented them from achieving key-dependent message security for an unbounded number of keys.

of security we use a variant of the Goldreich–Levin theorem for producing pseudorandom strings whose length is polynomially related to the security parameter.

The second problem is to identify an even more general framework for designing deterministic public-key encryption schemes that are secure with respect to hard-to-invert auxiliary inputs. Such a framework can potentially unify our two constructions and lead to additional constructions based on other computational assumptions. Significant progress in this aspect was recently made by Wee [27], who introduced the notion of dual projective hashing and showed that it provides a simple construction of deterministic encryption schemes that are secure with respect to hard-to-invert auxiliary inputs. In particular, Wee's approach encompasses our two constructions and also provides a new construction based the learning with errors assumption.

1.5. *Paper Organization*

In Sect. 2 we introduce some notation and preliminary tools. In Sect. 3 we formalize a general notion for hard-to-invert auxiliary inputs that is considered in this paper. In Sect. 4 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 Sect. 5 we prove that these three notions are in fact equivalent. In Sect. 6 we present a construction based on the d-linear assumption, and in Sect. 7 we present a construction based on subgroup indistinguishability assumptions.

2. Preliminaries

For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. The *min-entropy* of a distribution X over a set $\mathcal{X}$ is defined as $\mathbf{H}_\infty(X) \overset{\mathsf{def}}{=} -\log(\max_{x\in\mathcal{X}} \Pr[X = x])$. The statistical distance (total variation distance) between X and Y is denoted $\mathrm{SD}(X, Y)$. A real function over the naturals is *negligible* if it vanishes faster than any inverse polynomial; we use $f(k) = \mathrm{negl}(k)$ to denote that f is a negligible function. Two distribution ensembles $\{X_k\}_{k\in\mathbb{N}}$, $\{Y_k\}_{k\in\mathbb{N}}$ are *statistically indistinguishable* if $\mathrm{SD}(X_k, Y_k) = \mathrm{negl}(k)$; we denote this by $X \overset{s}{\approx} Y$. They are *computationally indistinguishable* if for any polynomial time algorithm A, it holds that $|\Pr_{x\leftarrow X_k}[A(1^k, x) = 1] - \Pr_{y\leftarrow Y_k}[A(1^k, y) = 1]|$ is negligible; we denote this by $X \overset{c}{\approx} Y$.

All computational hardness in this work is stated with regard to *non-uniform adversaries*. However, adapting to the uniform setting is immediate and in most cases requires no changes at all.

We denote scalars in plain lowercase letters (e.g., $x \in \{0, 1\}$). 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 (e.g., $\mathbf{x} \in \{0, 1\}^k$); 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 (e.g., $\mathbf{X} \in \{0,1\}^{k \times n}$). 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 $\mathbf{x}^T$.

For a commutative multiplicative group $\mathbb{G}$, and an isomorphic $\mathbb{Z}$-module $\mathbb{M}$,[8] the isomorphism $\mathbb{M} \to \mathbb{G}$ is denoted by $y \leftarrow g^x$ where $y \in \mathbb{G}$ and $x \in \mathbb{M}$. In a sense, g can be thought of as a symbolic representation of a generating set of $\mathbb{G}$. If $\mathbb{G}$ is cyclic then this isomorphism corresponds to actual exponentiation, with g being a generator of $\mathbb{G}$. In such a case $\mathbb{M} = \mathbb{Z}_q$ where q is the order of $\mathbb{G}$. We overload the notation g^x to matrices as follows: 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 $(g^{\mathbf{X}})_{i,j} = g^{(\mathbf{X})_{i,j}}$.

2.1. *Computational Assumptions*

We now define the computational assumptions on which we based the security of our deterministic encryption schemes.

The Decisional Diffie–Hellman and d-Linear Assumptions 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. In this paper we rely on the following matrix form of the d-linear assumption due to Naor and Segev [23] (generalizing [8]). For $d = 1$ this variant is equivalent to the DDH assumption, and for $d > 1$ it is implied by the d-linear assumption (see [23]). We denote by $\mathrm{Rk}_i(\mathbb{Z}_q^{a \times b})$ the set of all $a \times b$ matrices over $\mathbb{Z}_q$ with rank i. The matrix form of the d-linear assumption is that for any integers a and b, and for any $d \le i < j \le \min\{a, b\}$, the distributions $\{(\mathbb{G}, q, g, g^{\mathbf{X}})\}_{\mathbf{X} \leftarrow \mathrm{Rk}_i(\mathbb{Z}_q^{a \times b}), k \in \mathbb{N}}$ and $\{(\mathbb{G}, q, g, g^{\mathbf{Y}})\}_{\mathbf{Y} \leftarrow \mathrm{Rk}_j(\mathbb{Z}_q^{a \times b}), k \in \mathbb{N}}$ are computationally indistinguishable, where $(\mathbb{G}, q, g) \leftarrow \mathsf{GroupGen}(1^k)$.

A rather useful implication of the matrix form of the d-linear assumption is that the distributions $\{(\mathbb{G}, q, g, g^{\mathbf{X}})\}_{\mathbf{X} \leftarrow \mathbb{Z}_q^{a \times b}, k \in \mathbb{N}}$ and $\{(\mathbb{G}, q, g, g^{\mathbf{R} \cdot \mathbf{S}})\}_{\mathbf{R} \leftarrow \mathbb{Z}_q^{a \times d}, \mathbf{S} \leftarrow \mathbb{Z}_q^{d \times b}, k \in \mathbb{N}}$ are computationally indistinguishable, where $(\mathbb{G}, q, g) \leftarrow \mathsf{GroupGen}(1^k)$.[9]

Subgroup Indistinguishability Assumptions We present the class of subgroup indistinguishability assumptions formalized by Brakerski and Goldwasser [6] together with its instantiations based on the quadratic residuosity and composite residuosity assumptions.

Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^k, and outputs a tuple $(\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L, h, T)$ where $\mathbb{G}_U = \mathbb{G}_M \times \mathbb{G}_L$ is a commutative multiplicative group, $\mathbb{G}_M$ is a cyclic group of order M that is generated by h, $\mathbb{G}_L$ is a group of order L (which is not necessarily cyclic), $\gcd(M, L) = 1$, $M \cdot L$ is a k-bit number, and $T \ge M \cdot L$. We require that there exist efficient algorithms for

[8] A $\mathbb{Z}$-module is identical to an additive abelian group, only that in addition to the $+$ operation, it formally allows operations of the form $k \cdot x$ for $k \in \mathbb{Z}$ and $x \in \mathbb{M}$ (the multiplication is defined as repeated addition). Note that such "multiplication by scalar" is undefined for groups.

[9] The equivalence follows by defining $\tilde{\mathbf{R}} \leftarrow \mathrm{Rk}_d(\mathbb{Z}_q^{a \times d})$ and $\tilde{\mathbf{S}} \leftarrow \mathrm{Rk}_d(\mathbb{Z}_q^{d \times b})$ and noticing that since q is super-polynomial then $(\mathbf{S}, \mathbf{R}) \overset{s}{\approx} (\tilde{\mathbf{S}}, \tilde{\mathbf{R}})$, and that $\tilde{\mathbf{R}} \cdot \tilde{\mathbf{S}}$ is distributed uniformly in $\mathrm{Rk}_d(\mathbb{Z}_q^{a \times b})$.

performing group operations in $\mathbb{G}_U$, and for sampling uniformly distributed elements from $\mathbb{G}_M$ and $\mathbb{G}_L$.

The subgroup indistinguishability assumption is that a uniformly sampled element from $\mathbb{G}_U$ is computationally indistinguishable from a uniformly sampled element from $\mathbb{G}_L$. More formally, the distributions $\{(\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L, h, T, x) : x \leftarrow \mathbb{G}_L\}_{k\in\mathbb{N}}$ and $\{(\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L, h, T, h \cdot x) : x \leftarrow \mathbb{G}_L\}_{k\in\mathbb{N}}$ are computationally indistinguishable, where $(\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L, h, T) \leftarrow \mathsf{GroupGen}(1^k)$.

For proving the security of our schemes we rely on the following lemma:

Lemma 2.1 ([6]). *Under the subgroup indistinguishability assumption, for any polynomial $n = n(k)$ it holds that*

$$\left\{\left(g^{\mathbf{w}^T}, h^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T}\right)\right\}_{k\in\mathbb{N}} \overset{c}{\approx} \left\{\left(g^{\mathbf{w}^T}, g^{\mathbf{r}\cdot\mathbf{w}^T}\right)\right\}_{k\in\mathbb{N}},$$

where $(\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L, h, T) \leftarrow \mathsf{GroupGen}(1^k)$, $g^{\mathbf{w}^T} \leftarrow \mathbb{G}_L^n$, and $\mathbf{r} \leftarrow [T^2]^n$.

For instantiating the subgroup indistinguishability assumption based on the quadratic residuosity and composite residuosity assumptions, we consider a modulus N of the form $N = pq$, where p and q are random $k/2$-bit odd primes (we do not require that p and q are "safe primes").

- **The Quadratic Residuosity Assumption**. Let $\mathbb{J}_N$ denote the set of elements in $\mathbb{Z}_N^*$ with Jacobi symbol $+1$, and $\mathbb{QR}_N$ denote the set of *quadratic residues* (squares) modulo N. Overloading notation, we denote by $\mathbb{J}_N$ and $\mathbb{QR}_N$ also the respective groups with the multiplication operation modulo N. The groups $\mathbb{J}_N$ and $\mathbb{QR}_N$ have orders $\varphi(N)/2$ and $\varphi(N)/4$, respectively, and we let $N' = \varphi(N)/4$. We require that N is a *Blum integer*, namely that $p, q = 3 \pmod 4$. In such a case it holds that $\gcd(2, N') = 1$ and that $(-1) \in \mathbb{J}_N \setminus \mathbb{QR}_N$.

  The quadratic residuosity assumption [19] is that a uniformly chosen quadratic residue is computationally indistinguishable from a uniformly chosen quadratic non-residue (with Jacobi symbol 1). It is obtained from the above subgroup indistinguishability assumption by setting $\mathbb{G}_U = \mathbb{J}_N$, $\mathbb{G}_M = \{\pm 1\}$, $\mathbb{G}_L = \mathbb{QR}_N$, $h = (-1)$, and $T = N \geq 2N'$.
- **The Composite Residuosity Assumption**. The composite residuosity assumption [24] is that a uniformly chosen element from $\mathbb{Z}_{N^2}^*$ is computationally indistinguishable from a uniformly chosen element from the subgroup of Nth residues $\{x^N : x \in \mathbb{Z}_{N^2}^*\}$. The group $\mathbb{Z}_{N^2}^*$ can be written as a product of the group generated by $1 + N$ (which has order N) and the group of Nth residues (which has order $\varphi(N)$). This assumption is obtained from the above subgroup indistinguishability assumption by setting $\mathbb{G}_U = \mathbb{Z}_{N^2}^*$, $\mathbb{G}_M = \{(1 + N)^i : i \in [N]\}$, $\mathbb{G}_L = \{x^N : x \in \mathbb{Z}_{N^2}^*\}$, $h = (1 + N)$, and $T = N^2$.

2.2. *Hard-Core Functions*

Hard-core functions play a central tool in our approach for reducing a hard search problem (the hardness of inverting the auxiliary input) into a decision problem (the hardness of distinguishing encryptions of messages sampled from different distributions).

We present two hard-core function theorems, both are extensions of the well-known Goldreich–Levin theorem [18]. Theorem 2.2 is essentially taken from [11], and Theorem 2.3 extends Theorem 2.2 to the case where the domain under consideration is not a field, and appears, in a slightly less general form, in [6].

Theorem 2.2 ([11, Theorem 1]). *There exists a uniform oracle machine B such that for all $n \in \mathbb{N}$, for any (possibly randomized) function $f : \{0,1\}^n \to \{0,1\}^*$, any distribution $\mathcal{D}$ over $\{0,1\}^n$, any (nontrivial) finite field $\mathbb{F} = \mathbb{F}_n$ and function A such that*

$$\left|\Pr_{x \leftarrow \mathcal{D}}\left[A\left(f(x), r, \langle r, x\rangle\right) = 1\right] - \Pr\left[A\left(f(x), r, \alpha\right) = 1\right]\right| \geq \epsilon,$$

where $x \leftarrow \mathcal{D} \subseteq \{0,1\}^n \subseteq \mathbb{F}^n$, $r \leftarrow \mathbb{F}^n$, $\alpha \leftarrow \mathbb{F}$ and the inner product is over $\mathbb{F}$, it holds that B^A runs in polynomial time and

$$\Pr\left[B^A\left(1^n, \lceil 1/\epsilon \rceil, f(x)\right) = x\right] \geq \frac{\epsilon^3}{512 \cdot n \cdot |\mathbb{F}|^2}.$$

Furthermore, B only needs to sample uniformly in $\mathbb{F}$ and to add two elements in $\mathbb{F}$, and does not use any other property of the field.

Theorem 2.3 (Implicit in [6]). *There exists a uniform oracle machine B such that for all $n \in \mathbb{N}$, for any (possibly randomized) function $f : \{0,1\}^n \to \{0,1\}^*$, any distribution $\mathcal{D}$ over $\{0,1\}^n$, any (nontrivial) $\mathbb{Z}$-module $\mathbb{M} = \mathbb{M}_n$ and function A such that*

$$\left|\Pr\left[A\left(f(x), r, \langle r, x\rangle\right) = 1\right] - \Pr\left[A\left(f(x), r, \alpha\right) = 1\right]\right| \geq \epsilon,$$

where $x \leftarrow \mathcal{D} \subseteq \{0,1\}^n \subseteq \mathbb{Z}^n$, $r \leftarrow \mathbb{M}^n$, $\alpha \leftarrow \mathbb{M}$ and the "inner product" is over the module, it holds that B^A runs in polynomial time and

$$\Pr\left[B^A\left(1^n, \lceil 1/\epsilon \rceil, f(x)\right) = x\right] \geq \frac{\epsilon}{8 \cdot |\mathbb{M}|^{1+\log(8n/\epsilon^2)}}.$$

Furthermore, B only needs to sample uniformly in $\mathbb{M}$ and to add two elements in $\mathbb{M}$, and does not use any other property of the module.

2.3. *Deterministic Public-Key Encryption Scheme*

A deterministic public-key encryption scheme over a message space ensemble $\mathcal{M} = \{\mathcal{M}_k\}$ is a triplet $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of polynomial-time algorithms with the following properties.

- The *key-generation* algorithm KeyGen is a randomized algorithm that takes the security parameter 1^k as input and outputs a key pair $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$ containing a secret key and a public key.
- The *encryption* algorithm Enc is a *deterministic* algorithm that takes as input a public key pk and a message $m \in \mathcal{M}_k$ (where k is the security parameter), and outputs a ciphertext $c = \mathsf{Enc}_{pk}(m)$.
- The *decryption* algorithm is a possibly randomized algorithm that takes as input a secret key sk and a ciphertext c and outputs a message $m' \leftarrow \mathsf{Dec}_{sk}(c)$ such that $m' \in \mathcal{M}_k$.

For simplicity in this paper we assume perfect decryption: For every $k \in \mathbb{N}$ and $m \in \mathcal{M}_k$ it holds that $\Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m] = 1$, where the probability is taken over the choice of $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$ (and also over the internal randomness of Dec if the latter algorithm is randomized). We note that although, in general, decryption algorithms may be randomized, all decryption algorithms considered in this paper are in fact deterministic.

3. 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, following the framework of Dodis, Tauman Kalai, and Lovett [12]. 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 3.1. 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

$$\Pr\bigl[A\bigl(1^k, f_k(\vec{x})\bigr) = x_i\bigr] \le \epsilon(k),$$

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 3.2. 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

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

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 any $\mathcal{F}$ which is ϵ-blockwise-hard-to-invert with respect to $\mathcal{D}$ is also ϵ-hard-to-invert with respect to $\mathcal{D}$, but the other direction does not always hold (for example, whenever all the x_i's are identical). Definition 3.1 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 3.2 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 $(x_1, \dots, x_{i-1})$.

4. 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 [3–5] to a setting in which the encrypted plaintexts may be fully determined by some auxiliary information that is available to the adversary. Following Refs. [3–5], 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 $\vec{m} = (m_1, \dots, m_t)$ we denote by $\vec{\mathsf{Enc}}_{pk}(\vec{m})$ the vector $(\mathsf{Enc}_{pk}(m_1), \dots, \mathsf{Enc}_{pk}(m_t))$. When considering a distribution $\mathcal{M}$ over vectors of messages $\vec{m} = (m_1, \dots, m_t)$ 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 [3]). 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 4.1 (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 efficiently computable function $g \in \{0,1\}^* \to \{0,1\}^*$, there exists a negligible function $\nu(k)$ such that

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

for all sufficiently large k, where

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

$$\mathrm{Ideal}^{\mathsf{PRIV-SSS}}_{\Pi,S,\mathcal{M},\mathcal{F},g}(k) = \Pr\left[S\left(1^k, f_k(\vec{m})\right) = g(\vec{m})\right],$$

and the probabilities are 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 4.2 (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 efficiently computable function $g \in \{0,1\}^* \to \{0,1\}^*$, there exists a negligible function $\nu(k)$ such that

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

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 4.3 (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

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

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, [20, Chap. 5]). This is easily observed using our indistinguishability-based formulation (Definition 4.3): 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 ith 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 all previous security notions for deterministic public-key encryption, in which auxiliary input is not considered, and the message distributions need to have sufficient min-entropy.

Access to the Public Key As observed by Bellare et al. [3] 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 nontrivial information that it leaked about the plaintext, and can clearly be computed efficiently using the public key. We refer the reader to ref. [3] 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).

PRIV-CSS for Balanced Predicates Whereas Definition 4.2 considers arbitrary efficiently computable functions $g : \{0,1\}^* \to \{0,1\}^*$, we note that it is in fact equivalent to its seemingly simpler variant that considers only efficiently computable *δ-balanced predicates* $g : \{0,1\}^* \to \{0,1\}$ (where we say that a predicate g is δ-balanced on a distribution $\mathcal{M}$ if $|\Pr[g(\vec{m}) = 1] - 1/2| \leq \delta$). Moreover, it even suffices to consider only the case $\delta = 1/4$. The proof of this fact is identical to the corresponding proof of Bellare et al. [4], as the aspect of having an auxiliary input does not play any role in this setting. We will rely on this fact in Sect. 5 when showing that PRIV-IND implies PRIV-CSS.

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 Sect. 5.5 we then prove 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.

An Even Stronger Notion of Security Note that in Definition 4.3 the algorithm A is given as the input vector $(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_b), f_k(\vec{m}_0))$, and that a seemingly stronger definition would even consider $(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_b), \vec{\mathsf{Enc}}_{pk}(\vec{m}_{1-b}), f_k(\vec{m}_0), f_k(\vec{m}_1))$ 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 4.1. 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 4.4. 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

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

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.

The Multi-User Setting So far our notions of security considered vectors of messages that are encrypted under the same public key. We now present a natural generalization to the multi-user setting, where there are multiple public keys, each of which is used for encrypting a vector of messages. For simplicity we focus here on an indistinguishability-based definition, and note that our equivalence proofs (see Sect. 5) easily extend to this more general setting.

For the following definition we fix integer functions $\ell(k)$ and $t(k)$ of the security parameter k, indicating the number of public keys and the number of messages encrypted under each key, respectively. In addition we use the notation $\vec{pk} = (pk_1, \dots, pk_{\ell(k)})$, $\vec{\vec{m}} = (\vec{m}_1, \dots, \vec{m}_{\ell(k)})$ (where each $\vec{m}_i$ is a vector of $t(k)$ messages), and $\vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{m}}) = (\vec{\mathsf{Enc}}_{pk_1}(\vec{m}_1), \dots, \vec{\mathsf{Enc}}_{pk_{\ell(k)}}(\vec{m}_{\ell(k)}))$. Note that in the following definition the adversary receives $\ell(k)$ public keys and $\ell(k) \cdot t(k)$ ciphertexts.

Definition 4.5. A deterministic public-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is *PRIV-IND-MU-secure setting 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

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

for all sufficiently large k, where

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

and the probability is taken over the choices of $\vec{\vec{m}}_0 \leftarrow \mathcal{M}_{0,k}$, $\vec{\vec{m}}_1 \leftarrow \mathcal{M}_{1,k}$, $(sk_i, pk_i) \leftarrow \mathsf{KeyGen}(1^k)$ for all $i \in \{1, \dots, \ell(k)\}$, and over the internal coin tosses of A.

5. Equivalences Between Our Notions of Security

In this section we prove that our simulation-based (PRIV-SSS), comparison-based (PRIV-CSS), and indistinguishability-based (PRIV-IND) notions of security are equivalent. The high-level approach of the equivalence proofs in this section is motivated by Refs. [3–5], but the existence of auxiliary inputs introduces various additional difficulties that our proofs overcome. Specifically, all proofs but one (PRIV-CSS $\Longrightarrow$ PRIV-IND) are quite natural extensions of the proofs given by refs. [3–5] to the auxiliary-input setting, and for the PRIV-CSS $\Longrightarrow$ PRIV-IND proof we also included a high-level overview explaining the main difference.

As discussed in Sect. 4, our comparison-based notion serves essentially as an intermediate notion between the two other notions: we first prove that PRIV-IND and PRIV-CSS are equivalent, and then we prove that PRIV-CSS and PRIV-SSS are equivalent. These proofs are presented in Sects. 5.1–5.4, stated for hard-to-invert auxiliary inputs, and we note that they immediately extend to blockwise-hard-to-invert auxiliary inputs. In Sect. 5.5 we then prove that for our notions of security, 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.

5.1. *PRIV-CSS* $\Longrightarrow$ *PRIV-IND*

The following lemma shows that any scheme which is secure according to the comparison-based definition (Definition 4.2) is also secure according to the indistinguishability-based one (Definition 4.3).

Lemma 5.1. *Let Π be a deterministic public-key encryption scheme. Then, for any probabilistic polynomial-time algorithm A, for any two efficiently samplable distributions $\mathcal{M}_0$ and $\mathcal{M}_1$, and for any efficiently computable function $\mathcal{F}$ that is ϵ-hard-to-invert with respect to both $\mathcal{M}_0$ and $\mathcal{M}_1$, there exist a probabilistic polynomial-time algorithm A', an efficiently samplable distribution $\mathcal{M}$, an efficiently computable function $\mathcal{F}'$ that is ϵ-hard-to-invert with respect to $\mathcal{M}$, and two efficiently computable functions $g, g' \in \{0,1\}^* \to \{0,1\}$, such that for any $k \in \mathbb{N}$,*

$$\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k) \le \mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M}_0,\mathcal{F},g}(k) + 2 \cdot \mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A',\mathcal{M},\mathcal{F}',g'}(k).$$

Before providing the formal proof of Lemma 5.1 we first provide a high-level overview of its structure. Given an adversary A, two distributions $\mathcal{M}_0$ and $\mathcal{M}_1$, and a function f that is hard-to-invert with respect to both $\mathcal{M}_0$ and $\mathcal{M}_1$, we would like to upper bound the advantage of A in distinguishing between the distributions $\mathcal{D}_0 = (pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_0), f(\vec{m}_0))_{\vec{m}_0 \leftarrow \mathcal{M}_0}$ and $\mathcal{D}_1 = (pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1), f(\vec{m}_0))_{\vec{m}_0 \leftarrow \mathcal{M}_0, \vec{m}_1 \leftarrow \mathcal{M}_1}$. The main idea underlying the proof is based on a hybrid argument by considering the intermediate distribution $\mathcal{D}' = (pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1), f(\vec{m}_0))_{\vec{m}_0, \vec{m}_1 \leftarrow \mathcal{M}_0}$. We first observe that the advantage of A in distinguishing between $\mathcal{D}_0$ and $\mathcal{D}'$ is exactly $\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M}_0,\mathcal{F},g}$, where g is the constant function that evaluates to 1 on all inputs. Next, for bounding the advantage of A in distinguishing between $\mathcal{D}'$ and $\mathcal{D}_1$, note that in both distributions the ciphertext and the auxiliary input that are given to A are independent (i.e., A is provided with auxiliary input for a message that is chosen independently than the one encrypted).

The proof in this case is essentially identical to that of Bellare et al. [4] by ignoring the auxiliary input.

Proof of Lemma 5.1. For any algorithm A, two distributions $\mathcal{M}_0$ and $\mathcal{M}_1$, and function $\mathcal{F}$, it holds that

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k) &= \big|\Pr_{\vec{m}_0 \leftarrow \mathcal{M}_0}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_0), f_k(\vec{m}_0)\big) = 1\big] \\
&\quad - \Pr_{\substack{\vec{m}_0 \leftarrow \mathcal{M}_0 \\ \vec{m}_1 \leftarrow \mathcal{M}_1}}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1), f_k(\vec{m}_0)\big) = 1\big]\big| \qquad (5.1) \\
&\le \big|\Pr_{\vec{m}_0 \leftarrow \mathcal{M}_0}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_0), f_k(\vec{m}_0)\big) = 1\big] \\
&\quad - \Pr_{\substack{\vec{m}_0 \leftarrow \mathcal{M}_0 \\ \vec{m}_1 \leftarrow \mathcal{M}_0}}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1), f_k(\vec{m}_0)\big) = 1\big]\big| \\
&\quad + \big|\Pr_{\substack{\vec{m}_0 \leftarrow \mathcal{M}_0 \\ \vec{m}_1 \leftarrow \mathcal{M}_0}}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1), f_k(\vec{m}_0)\big) = 1\big] \\
&\quad - \Pr_{\substack{\vec{m}_0 \leftarrow \mathcal{M}_0 \\ \vec{m}_1 \leftarrow \mathcal{M}_1}}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1), f_k(\vec{m}_0)\big) = 1\big]\big| \qquad (5.2) \\
&= \mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M}_0,\mathcal{F},g}(k) \\
&\quad + \big|\Pr_{\vec{m}_1 \leftarrow \mathcal{M}_0}\big[A'\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1)\big) = 1\big] \\
&\quad - \Pr_{\vec{m}_1 \leftarrow \mathcal{M}_1}\big[A'\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1)\big) = 1\big]\big|, \qquad (5.3)
\end{aligned}$$

where Eq. (5.1) follows by triangle inequality with $\Pr_{\substack{\vec{m}_0 \leftarrow \mathcal{M}_0 \\ \vec{m}_1 \leftarrow \mathcal{M}_0}}[A(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}_1), f_k(\vec{m}_0)) = 1]$ and where $g \equiv 1$ (i.e., g is the function that evaluates to 1 on all inputs), and A' is an algorithm that on input $(1^k, pk, \vec{c})$ samples $\vec{m}_0 \leftarrow \mathcal{M}_0$ and invokes A on input $(1^k, pk, \vec{c}, f_k(\vec{m}_0))$.

Note that the expressions in Eqs. (5.2) and (5.3) correspond to the advantage of A' in distinguishing between an encryption resulting from $\mathcal{M}_0$ and an encryption resulting from $\mathcal{M}_1$ without auxiliary input. For bounding this advantage, let $\mathcal{M}$ be the balanced convex combination of $\mathcal{M}_0$ and $\mathcal{M}_1$ (that is, $\mathcal{M}$ samples from $\mathcal{M}_b$ for a uniformly chosen $b \in \{0, 1\}$), and let g' be the indicator to whether $\mathcal{M}$ samples from $\mathcal{M}_0$ or $\mathcal{M}_1$. The assumption that $\mathcal{F}$ is ϵ-hard-to-invert with respect to both $\mathcal{M}_0$ and $\mathcal{M}_1$ implies in particular that by letting $\mathcal{F}'$ be any constant function we clearly have that $\mathcal{F}'$ is ϵ-hard-to-invert with respect $\mathcal{M}$. In addition, it holds that

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A',\mathcal{M},\mathcal{F}',g'}(k, 0) &= \frac{1}{2} \cdot \big(1 - \Pr_{\vec{m} \leftarrow \mathcal{M}_0}\big[A'\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m})\big) = 1\big]\big) \\
&\quad + \frac{1}{2} \cdot \Pr_{\vec{m} \leftarrow \mathcal{M}_1}\big[A'\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m})\big) = 1\big], \qquad (5.4)
\end{aligned}$$

and

$$\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A',\mathcal{M},\mathcal{F}',g'}(k, 1) = \Pr_{\substack{\vec{m} \leftarrow \mathcal{M} \\ \vec{m}' \leftarrow \mathcal{M}}}\big[A'\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}\big(\vec{m}'\big)\big) = g'(\vec{m})\big] = \frac{1}{2}. \qquad (5.5)$$

Combining Eqs. (5.4) and (5.5) implies that

$$\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A',\mathcal{M},\mathcal{F}',g'}(k) = \frac{1}{2} \cdot \bigl|\Pr_{\vec{m}\leftarrow\mathcal{M}_0}\bigl[A'\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m})\bigr) = 1\bigr] - \Pr_{\vec{m}\leftarrow\mathcal{M}_1}\bigl[A'\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m})\bigr) = 1\bigr]\bigr|,$$

and therefore $\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k) \le \mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M}_0,\mathcal{F},g}(k) + 2 \cdot \mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A',\mathcal{M},\mathcal{F}',g'}(k)$. □

5.2. *PRIV-IND* $\Longrightarrow$ *PRIV-CSS*

The following lemma shows that any scheme which is secure according to the indistinguishability-based definition (Definition 4.3) is also secure according to the comparison-based one (Definition 4.2). The proof is obtained quite naturally by extending the proof of Bellare et al. [4] to the auxiliary-input setting. As discussed in Sect. 4, recall that for the comparison-based definition it suffices to consider efficiently computable δ-balanced predicates $g : \{0,1\}^* \to \{0,1\}$ for $\delta = 1/4$.

Lemma 5.2. *Let Π be a deterministic public-key encryption scheme. Then, for any probabilistic polynomial-time algorithm A, for any efficiently samplable distribution $\mathcal{M}$, for any efficiently computable function $\mathcal{F}$ that is ϵ-hard-to-invert with respect to $\mathcal{M}$, for any efficiently computable predicate $g : \{0,1\}^* \to \{0,1\}$ that is δ-balanced on $\mathcal{M}$ (where $0 < \delta < 1/2$), and for any polynomial $p(k)$, there exist two efficiently samplable distributions $\mathcal{M}_0$ and $\mathcal{M}_1$ for which $\mathcal{F}$ is $(\frac{\epsilon(k)}{1/2-\delta} + (1/2+\delta)^{p(k)})$-hard-to-invert, such that for any $k \in \mathbb{N}$,*

$$\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k) \le \max\bigl\{\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M},\mathcal{F}}(k), \mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_1,\mathcal{M},\mathcal{F}}(k)\bigr\} + \left(\frac{1}{2}+\delta\right)^{p(k)}.$$

Proof. Given a distribution $\mathcal{M} = \{\mathcal{M}_k\}_{k\in\mathbb{N}}$ and a predicate $g : \{0,1\}^* \to \{0,1\}$ that is δ-balanced on $\mathcal{M}$, we let

$$\alpha_{0,k} = \Pr_{\vec{m}\leftarrow\mathcal{M}_k}\bigl[g(\vec{m}) = 0\bigr],$$
$$\alpha_{1,k} = \Pr_{\vec{m}\leftarrow\mathcal{M}_k}\bigl[g(\vec{m}) = 1\bigr],$$

and denote by $\mathcal{M}_k|_{g=0}$ and $\mathcal{M}_k|_{g=1}$ the conditional distributions of $\mathcal{M}_k$ given that $g(\mathcal{M}_k) = 0$ and $g(\mathcal{M}_k) = 1$, respectively. Note that these conditional distributions are not always efficiently samplable, and therefore we would like to approximate them by two efficiently samplable distributions to within a negligible statistical distance. For this purpose for each $b \in \{0,1\}$ we define the distribution $\mathcal{M}_b = \{\mathcal{M}_{b,k}\}_{k\in\mathbb{N}}$ as the convex combination

$$\mathcal{M}_{b,k} = \alpha^{p(k)}_{1-b,k} \cdot \mathcal{M}_k + \bigl(1 - \alpha^{p(k)}_{1-b,k}\bigr) \cdot \mathcal{M}_k|_{g=b}.$$

That is, the distribution $\mathcal{M}_{b,k}$ samples from $\mathcal{M}_k$ with probability $\alpha^{p(k)}_{1-b,k}$ and samples from $\mathcal{M}_k|_{g=b}$ with probability $(1-\alpha^{p(k)}_{1-b,k})$. The definition of $\mathcal{M}_{b,k}$ directly implies that the statistical distance between $\mathcal{M}_{b,k}$ and $\mathcal{M}_k|_{g=b}$ is at most $\alpha^{p(k)}_{1-b,k} \le (1/2+\delta)^{p(k)}$.

In addition, the distribution $\mathcal{M}_{b,k}$ is samplable by the following efficient algorithm: (1) sample $\vec{m}_1, \dots, \vec{m}_{p(k)+1} \leftarrow \mathcal{M}$, (2) if there exists an index $i \in \{1, \dots, p(k)\}$ such that $g(\vec{m}_i) = b$, then output $\vec{m}_i$ for the minimal such i, and otherwise output $\vec{m}_{p(k)+1}$. Finally, note that for any function $\mathcal{F} = \{f_k\}_{k \in \mathbb{N}}$ that is ϵ-hard-to-invert with respect to $\mathcal{M}$, for any efficient algorithm I, and for any $i \in \{1, \dots, t(k)\}$ it holds that

$$\begin{aligned}
\epsilon(k) &\ge \Pr_{\vec{x} \leftarrow \mathcal{M}_k}\bigl[I\bigl(1^k, f_k(\vec{x})\bigr) = x_i\bigr] \\
&\ge \alpha_{b,k} \cdot \Pr_{\vec{x} \leftarrow \mathcal{M}_k|_{g=b}}\bigl[I\bigl(1^k, f_k(\vec{x})\bigr) = x_i\bigr] \\
&\ge \alpha_{b,k} \cdot \bigl(\Pr_{\vec{x} \leftarrow \mathcal{M}_{b,k}}\bigl[I\bigl(1^k, f_k(\vec{x})\bigr) = x_i\bigr] - \mathrm{SD}(\mathcal{M}_k|_{g=b}, \mathcal{M}_{b,k})\bigr),
\end{aligned}$$

and therefore

$$\begin{aligned}
\Pr_{\vec{x} \leftarrow \mathcal{M}_{b,k}}\bigl[I\bigl(1^k, f_k(\vec{x})\bigr) = x_i\bigr] &\le \frac{\epsilon(k)}{\alpha_{b,k}} + \mathrm{SD}(\mathcal{M}_k|_{g=b}, \mathcal{M}_{b,k}) \\
&\le \frac{\epsilon(k)}{1/2 - \delta} + \left(\frac{1}{2} + \delta\right)^{p(k)}.
\end{aligned}$$

That is, $\mathcal{F}$ is $(\frac{\epsilon(k)}{1/2-\delta} + (\frac{1}{2} + \delta)^{p(k)})$-hard-to-invert with respect to $\mathcal{M}_0$ and $\mathcal{M}_1$. We conclude the proof by observing that

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k) &= \bigl|\Pr_{\vec{m} \leftarrow \mathcal{M}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}), f_k(\vec{m})\bigr) = g(\vec{m})\bigr] \\
&\quad - \Pr_{\substack{\vec{m} \leftarrow \mathcal{M} \\ \vec{m}' \leftarrow \mathcal{M}}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}\bigl(\vec{m}'\bigr), f_k(\vec{m})\bigr) = g(\vec{m})\bigr]\bigr| \\
&= \bigl|\alpha_{0,k} \cdot \Pr_{\vec{m} \leftarrow \mathcal{M}|_{g=0}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}), f_k(\vec{m})\bigr) = 0\bigr] \\
&\quad + \alpha_{1,k} \cdot \Pr_{\vec{m} \leftarrow \mathcal{M}|_{g=1}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}), f_k(\vec{m})\bigr) = 1\bigr] \\
&\quad - \alpha_{0,k} \cdot \Pr_{\substack{\vec{m} \leftarrow \mathcal{M}|_{g=0} \\ \vec{m}' \leftarrow \mathcal{M}}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}\bigl(\vec{m}'\bigr), f_k(\vec{m})\bigr) = 0\bigr] \\
&\quad - \alpha_{1,k} \cdot \Pr_{\substack{\vec{m} \leftarrow \mathcal{M}|_{g=1} \\ \vec{m}' \leftarrow \mathcal{M}}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}\bigl(\vec{m}'\bigr), f_k(\vec{m})\bigr) = 1\bigr]\bigr| \\
&\le \alpha_{0,k} \cdot \mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}|_{g=0},\mathcal{M},\mathcal{F}}(k) + \alpha_{1,k} \cdot \mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}|_{g=1},\mathcal{M},\mathcal{F}}(k) \\
&\le \alpha_{0,k} \cdot \left(\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M},\mathcal{F}}(k) + \left(\frac{1}{2} + \delta\right)^{p(k)}\right) \\
&\quad + \alpha_{1,k} \cdot \left(\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_1,\mathcal{M},\mathcal{F}}(k) + \left(\frac{1}{2} + \delta\right)^{p(k)}\right) \\
&\le \max\bigl\{\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M},\mathcal{F}}(k), \mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_1,\mathcal{M},\mathcal{F}}(k)\bigr\} + \left(\frac{1}{2} + \delta\right)^{p(k)}.
\end{aligned}$$

□

5.3. *PRIV-CSS* $\Longrightarrow$ *PRIV-SSS*

The following lemma shows that any scheme which is secure according to the comparison-based definition (Definition 4.2) is also secure according to the simulation-based one (Definition 4.1). The proof is essentially identical to that of Bellare et al. [4] as the auxiliary input can be easily incorporated into their analysis.

Lemma 5.3. *Let Π be a deterministic public-key encryption scheme. Then, for any probabilistic polynomial-time algorithm A and for any efficiently samplable distribution $\mathcal{M}$, there exists a probabilistic polynomial-time algorithm S, such that for any efficiently computable function $\mathcal{F}$ that is ϵ-hard-to-invert with respect to $\mathcal{M}$, for any efficiently computable function $g \in \{0,1\}^* \to \{0,1\}^*$, and for any $k \in \mathbb{N}$ it holds that*

$$\mathrm{Adv}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},S,\mathcal{F},g}(k) = \mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k).$$

Proof. Given an algorithm A and a distribution $\mathcal{M}$ as in the statement of the lemma, we define an algorithm S that on input $(1^k, y)$ samples $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$ and $\vec{m}' \leftarrow \mathcal{M}_k$, and then invokes A on the input $(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}'), y)$. Then, for any $\mathcal{F}$ and g it holds that

$$\begin{aligned}
\mathrm{Adv}^{\mathsf{PRIV-SSS}}_{\Pi,A,S,\mathcal{M},\mathcal{F},g}(k) &= \big|\Pr_{\vec{m}\leftarrow\mathcal{M}}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}), f_k(\vec{m})\big) = g(\vec{m})\big] \\
&\quad - \Pr_{\vec{m}\leftarrow\mathcal{M}}\big[S\big(1^k, f_k(\vec{m})\big) = g(\vec{m})\big]\big| \\
&= \big|\Pr_{\vec{m}\leftarrow\mathcal{M}}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}), f_k(\vec{m})\big) = g(\vec{m})\big] \\
&\quad - \Pr_{\substack{\vec{m}\leftarrow\mathcal{M}\\ \vec{m}'\leftarrow\mathcal{M}}}\big[A\big(1^k, pk, \vec{\mathsf{Enc}}_{pk}\big(\vec{m}'\big), f_k(\vec{m})\big) = g(\vec{m})\big]\big| \\
&= \mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k).
\end{aligned}$$

□

5.4. *PRIV-SSS* $\Longrightarrow$ *PRIV-CSS*

The following lemma shows that any scheme which is secure according to the simulation-based definition (Definition 4.1) is also secure according to the comparison-based one (Definition 4.2). The proof is essentially identical to that of Bellare et al. [4] as the auxiliary input can be easily incorporated into their analysis.

Lemma 5.4. *Let Π be a deterministic public-key encryption scheme. Then, for any probabilistic polynomial-time algorithm A, for any efficiently samplable distribution $\mathcal{M}$, for any efficiently computable function $\mathcal{F}$ that is ϵ-hard-to-invert with respect to $\mathcal{M}$, for any efficiently computable function $g : \{0,1\}^* \to \{0,1\}^*$, and for any probabilistic polynomial-time algorithm S, there exist an efficiently computable function $\mathcal{F}'$ that is ϵ-hard-to-invert with respect to $\mathcal{M}$, and an efficiently computable function $g' : \{0,1\}^* \to \{0,1\}^*$, such that for any $k \in \mathbb{N}$,*

$$\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k) \le \mathrm{Adv}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},S,\mathcal{F},g}(k) + \mathrm{Adv}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},S,\mathcal{F}',g'}(k).$$

Proof. For any algorithm A, distribution $\mathcal{M}$, functions $\mathcal{F}$ and g, and algorithm S as in the statement of the lemma, it holds that

$$\begin{align*}
\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k) &= \bigl|\Pr_{\vec{m}\leftarrow\mathcal{M}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}), f_k(\vec{m})\bigr) = g(\vec{m})\bigr] \\
&\quad - \Pr_{\substack{\vec{m}\leftarrow\mathcal{M}\\ \vec{m}'\leftarrow\mathcal{M}}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}\bigl(\vec{m}'\bigr), f_k(\vec{m})\bigr) = g(\vec{m})\bigr]\bigr| \\
&\le \bigl|\Pr_{\vec{m}\leftarrow\mathcal{M}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}(\vec{m}), f_k(\vec{m})\bigr) = g(\vec{m})\bigr] \tag{5.6}\\
&\quad - \Pr_{\vec{m}\leftarrow\mathcal{M}}\bigl[S\bigl(1^k, f_k(\vec{m})\bigr) = g(\vec{m})\bigr]\bigr| \tag{5.7}\\
&\quad + \bigl|\Pr_{\vec{m}\leftarrow\mathcal{M}}\bigl[S\bigl(1^k, f_k(\vec{m})\bigr) = g(\vec{m})\bigr] \tag{5.8}\\
&\quad - \Pr_{\substack{\vec{m}\leftarrow\mathcal{M}\\ \vec{m}'\leftarrow\mathcal{M}}} A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}\bigl(\vec{m}'\bigr), f_k(\vec{m})\bigr) = g(\vec{m})\bigr|. \tag{5.9}
\end{align*}$$

Note that the expression in Eqs. (5.6) and (5.7) is exactly $\mathrm{Adv}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},S,\mathcal{F},g}(k)$. For upper bounding the expression in Eqs. (5.8) and (5.9) we fix the vector of messages $\vec{m}$ in the support of $\mathcal{M}$ which maximizes this expression. That is, we define

$$\begin{aligned}
\vec{m}^* = \mathrm{argmax}_{\vec{m}} &\bigl|\Pr\bigl[S\bigl(1^k, f_k(\vec{m})\bigr) = g(\vec{m})\bigr] \\
&- \Pr_{\vec{m}'\leftarrow\mathcal{M}}\bigl[A\bigl(1^k, pk, \vec{\mathsf{Enc}}_{pk}\bigl(\vec{m}'\bigr), f_k(\vec{m})\bigr) = g(\vec{m})\bigr]\bigr|.
\end{aligned}$$

Then, we define the functions $\mathcal{F}' = \{f'_k\}_{k\in\mathbb{N}}$ and g' as the constant functions $f'_k(\vec{m}) := f_k(\vec{m}^*)$ and $g'(\vec{m}) := g(\vec{m}^*)$, respectively.[10] The function $\mathcal{F}'$ is clearly ϵ-hard-to-invert with respect to $\mathcal{M}$, and we obtain

$$\mathrm{Adv}^{\mathsf{PRIV-CSS}}_{\Pi,A,\mathcal{M},\mathcal{F},g}(k) \le \mathrm{Adv}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},S,\mathcal{F},g}(k) + \mathrm{Adv}^{\mathsf{PRIV-SSS}}_{\Pi,A,\mathcal{M},S,\mathcal{F}',g'}(k). \qquad \square$$

5.5. *PRIV1-IND* $\Longrightarrow$ *PRIV-IND*

We now prove that for the case of blockwise-hard-to-invert auxiliary inputs it is in fact sufficient to consider a single message (the other direction clearly follows by definition). Recall that PRIV1-SSS, PRIV1-CSS, and PRIV1-IND denote the notions of security when considering only distributions over a single message in Definitions 4.1, 4.2, and 4.3, respectively. Our proofs of equivalence in Sects. 5.1–5.4 hold for any polynomial number $t(k)$ of messages, and thus show that the notions PRIV1-SSS, PRIV1-CSS, and PRIV1-IND are equivalent. Therefore, it suffices to prove that any deterministic public-key encryption scheme that is PRIV1-IND-secure with respect to hard-to-invert auxiliary inputs is also PRIV-IND with respect to blockwise-hard-to-invert inputs. The proof is essentially identical to that of Boldyreva, Fehr, and O'Neill [5] as the auxiliary input can be easily incorporated into their analysis.

Lemma 5.5. *Let Π be a deterministic public-key encryption scheme. Then, for any probabilistic polynomial-time algorithm A, for any two efficiently samplable distributions $\mathcal{M}_0$ and $\mathcal{M}_1$ over vectors of polynomial length $t(k)$, and for any efficiently*

[10] We note that adapting the proof to the uniform setting is quite straightforward given that f'_k and g' are allowed to share a random string (as discussed in Sect. 4). Given such a shared string the message $\vec{m}^*$ can be sampled from $\mathcal{M}$ instead of being fixed in a non-uniform manner, and the exact same argument goes through.

computable function $\mathcal{F}$ that is $\epsilon(k)$-blockwise-hard-to-invert with respect to both $\mathcal{M}_0$ and $\mathcal{M}_1$, there exist probabilistic polynomial-time algorithms $\{B_b^{(i)}\}_{\substack{b\in\{0,1\}\\ i\in\{1,\ldots,t(k)\}}}$, efficiently samplable distributions $\{\mathcal{M}_b^{(i)}, \widetilde{\mathcal{M}}_b^{(i)}\}_{\substack{b\in\{0,1\}\\ i\in\{1,\ldots,t(k)\}}}$, and efficiently computable functions $\{\mathcal{F}_b^{(i)}\}_{\substack{b\in\{0,1\}\\ i\in\{1,\ldots,t(k)\}}}$ such that each $\mathcal{F}_b^{(i)}$ is $\epsilon(k)$-hard-to-invert with respect to both $\mathcal{M}_b^{(i)}$ and $\widetilde{\mathcal{M}}_b^{(i)}$, and for any $k \in \mathbb{N}$ it holds that

$$\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k) \le \sum_{\substack{b\in\{0,1\}\\ i\in\{1,\ldots,t(k)\}}} \mathrm{Adv}^{\mathsf{PRIV1-IND}}_{\Pi,B_b^{(i)},\mathcal{M}_b^{(i)},\widetilde{\mathcal{M}}_b^{(i)},\mathcal{F}_b^{(i)}}(k).$$

Proof. Let $t = t(k)$, $\vec{m}_0 = (m_{0,1}, \ldots, m_{0,t})$, $\vec{m}_1 = (m_{1,1}, \ldots, m_{1,t})$. Consider the uniform distribution on the message space of the scheme Π and let $\mathcal{U}$ be the distribution that outputs t such uniform messages: $\vec{u} = (u_1, \ldots, u_t)$. Then for any algorithm A, distributions $\mathcal{M}_0$ and $\mathcal{M}_1$, and function $\mathcal{F}$, as in the statement of the theorem, it holds that

$$\begin{aligned}
&\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{M}_1,\mathcal{F}}(k)\\
&\quad= \bigl|\Pr_{\vec{m}_0\leftarrow\mathcal{M}_0}\bigl[A\bigl(1^k, pk, \mathsf{Enc}_{pk}(m_{0,1}), \ldots, \mathsf{Enc}_{pk}(m_{0,t}), f_k(\vec{m}_0)\bigr) = 1\bigr]\\
&\qquad - \Pr_{\substack{\vec{m}_0\leftarrow\mathcal{M}_0\\ \vec{m}_1\leftarrow\mathcal{M}_1}}\bigl[A\bigl(1^k, pk, \mathsf{Enc}_{pk}(m_{1,1}), \ldots, \mathsf{Enc}_{pk}(m_{1,t}), f_k(\vec{m}_0)\bigr) = 1\bigr]\bigr|\\
&\quad\le \bigl|\Pr_{\vec{m}_0\leftarrow\mathcal{M}_0}\bigl[A\bigl(1^k, pk, \mathsf{Enc}_{pk}(m_{0,1}), \ldots, \mathsf{Enc}_{pk}(m_{0,t}), f_k(\vec{m}_0)\bigr) = 1\bigr]\\
&\qquad - \Pr_{\substack{\vec{m}_0\leftarrow\mathcal{M}_0\\ \vec{u}\leftarrow\mathcal{U}}}\bigl[A\bigl(1^k, pk, \mathsf{Enc}_{pk}(u_1), \ldots, \mathsf{Enc}_{pk}(u_t), f_k(\vec{m}_0)\bigr) = 1\bigr]\bigr|\\
&\qquad + \bigl|\Pr_{\substack{\vec{m}_0\leftarrow\mathcal{M}_0\\ \vec{u}\leftarrow\mathcal{U}}}\bigl[A\bigl(1^k, pk, \mathsf{Enc}_{pk}(u_1), \ldots, \mathsf{Enc}_{pk}(u_t), f_k(\vec{m}_0)\bigr) = 1\bigr]\\
&\qquad - \Pr_{\substack{\vec{m}_0\leftarrow\mathcal{M}_0\\ \vec{m}_1\leftarrow\mathcal{M}_1}}\bigl[A\bigl(1^k, pk, \mathsf{Enc}_{pk}(m_{1,1}), \ldots, \mathsf{Enc}_{pk}(m_{1,t}), f_k(\vec{m}_0)\bigr) = 1\bigr]\bigr|\\
&\quad= \mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{U},\mathcal{F}}(k) + \mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A',\mathcal{U},\mathcal{M}_1,\mathcal{F}'}(k),
\end{aligned}$$

where A' is an algorithm that on input $(1^k, pk, c_1, \ldots, c_t)$ samples $\vec{m}_0 \leftarrow \mathcal{M}_0$ and invokes A on input $(1^k, pk, c_1, \ldots, c_t, f_k(\vec{m}_0))$, and $\mathcal{F}'$ is any constant function. In the remainder of the proof we bound the terms $\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{U},\mathcal{F}}(k)$ and $\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A',\mathcal{U},\mathcal{M}_1,\mathcal{F}'}(k)$ separately.

We remark that while the use of the uniform distribution $\mathcal{U}$ might seem surprising at a first glance, it is required for a technical reason that will be pointed out when it becomes relevant.

Bounding $\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{U},\mathcal{F}}(k)$ For every $i \in \{1, \ldots, t+1\}$ denote by $\mathcal{D}_0^{(i)}$ the distribution that samples $\vec{m}_0 = (m_{0,1}, \ldots, m_{0,t}) \leftarrow \mathcal{M}_0$, $\vec{u} = (u_1, \ldots, u_t) \leftarrow \mathcal{U}$, and then outputs the vector of messages $(m_{0,1}, \ldots, m_{0,i-1}, u_i, \ldots, u_t)$. In addition, for every $i \in \{1, \ldots, t\}$ let $\mathcal{F}^{(i)} = \{f_k^{(i)}\}$ be the function that takes as input the randomness used for sampling from $\mathcal{D}_0^{(i)}$, and outputs $(m_{0,1}, \ldots, m_{0,i-1}, f_k(\vec{m}_0))$. Then $\mathcal{D}_0^{(1)} = \mathcal{U}$,

$\mathcal{D}_0^{(t+1)} = \mathcal{M}_0$, and it holds that

$$\begin{aligned}
&\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A,\mathcal{M}_0,\mathcal{U},\mathcal{F}}(k) \\
&\quad\le \sum_{i=1}^{t} \Big| \Pr_{\substack{\vec{m}_0 \leftarrow \mathcal{M}_0 \\ \vec{u} \leftarrow \mathcal{U}}} \big[A\big(1^k, pk, \mathsf{Enc}_{pk}(m_{0,1}), \ldots, \mathsf{Enc}_{pk}(m_{0,i-1}), \\
&\qquad\qquad\qquad \mathsf{Enc}_{pk}(u_i), \ldots, \mathsf{Enc}_{pk}(u_t), f_k(\vec{m}_0)\big) = 1 \big] \\
&\qquad - \Pr_{\substack{\vec{m}_0 \leftarrow \mathcal{M}_0 \\ \vec{u} \leftarrow \mathcal{U}}} \big[A\big(1^k, pk, \mathsf{Enc}_{pk}(m_{0,1}), \ldots, \mathsf{Enc}_{pk}(m_{0,i}), \\
&\qquad\qquad\qquad \mathsf{Enc}_{pk}(u_{i+1}), \ldots, \mathsf{Enc}_{pk}(u_t), f_k(\vec{m}_0)\big) = 1 \big] \Big| \\
&\quad= \sum_{i=1}^{t} \Big| \Pr_{\substack{\vec{m}' \leftarrow \mathcal{D}_0^{(i)}(r) \\ \vec{m}' = (m'_1, \ldots, m'_t)}} \big[B_0^{(i)}\big(1^k, pk, \mathsf{Enc}_{pk}\big(m'_i\big), f_k^{(i)}\big(\vec{m}', r\big)\big) = 1 \big] \\
&\qquad - \Pr_{\substack{\vec{m}' \leftarrow \mathcal{D}_0^{(i+1)}(r) \\ \vec{m}' = (m'_1, \ldots, m'_t)}} \big[B_0^{(i)}\big(1^k, pk, \mathsf{Enc}_{pk}\big(m'_i\big), f_k^{(i)}\big(\vec{m}', r\big)\big) = 1 \big] \Big|, \\
&\quad= \sum_{i=1}^{t} \mathrm{Adv}^{\mathsf{PRIV1-IND}}_{\Pi, B_0^{(i)}, \mathcal{M}_0^{(i)}, \widetilde{\mathcal{M}}_0^{(i)}, \mathcal{F}_0^{(i)}}(k),
\end{aligned}$$

where for every $i \in \{1, \ldots, t\}$:

- $\mathcal{M}_0^{(i)}$ is a distribution that samples from $\mathcal{D}_0^{(i)}$ and then outputs the ith component of the latter.
- $\widetilde{\mathcal{M}}_0^{(i)}$ is a distribution that samples from $\mathcal{D}_0^{(i+1)}$ and then outputs the ith component of the latter.
- $B_0^{(i)}$ is an algorithm that on input $(1^k, pk, c, \vec{y})$, where $\vec{y} = (m_1, \ldots, m_{i-1}, y)$, invokes A on input $(1^k, pk, \mathsf{Enc}_{pk}(m_1), \ldots, \mathsf{Enc}_{pk}(m_{i-1}), c, \mathsf{Enc}_{pk}(u_{i+1}), \ldots, \mathsf{Enc}_{pk}(u_t), y)$ for independently and uniformly chosen messages $u_{i+1}, \ldots, u_t$.

Let us now explain the necessity of using the uniform distribution $\mathcal{U}$, as opposed to using a hybrid distribution that consists of $\mathcal{M}_0$ and $\mathcal{M}_1$: Given $m_1, \ldots, m_{i-1}$ that are sampled from $\mathcal{M}_0$, and given that m_i is sampled from either $\mathcal{M}_0$ or $\mathcal{M}_1$, it is not clear how to sample $m_{i+1}, \ldots, m_t$ from the distribution $\mathcal{M}_1$ conditioned on the first i values $m_1, \ldots, m_i$. When we use a hybrid distribution that consists of $\mathcal{M}_0$ and the uniform distribution $\mathcal{U}$, however, sampling the values $u_{i+1}, \ldots, u_t$ from $\mathcal{U}$ is trivial as these values are uniformly distributed, independently of the first $i - 1$ elements that are sampled from $\mathcal{M}_0$ and independently of whether the ith element is sampled from $\mathcal{M}_0$ or from $\mathcal{U}$. We note that the technique of using the uniform distribution $\mathcal{U}$ in this proof dates back to the analogous proof of Boldyreva, Fehr, and O'Neill [5].

Finally, we observe that for every $i \in \{1, \ldots, t\}$ the function $\mathcal{F}^{(i)}$ is clearly $\epsilon(k)$-hard-to-invert with respect to the distribution $\mathcal{M}_0^{(i)}$: the distribution $\mathcal{M}_0^{(i)}$ outputs a uniformly distributed value u_i that is independent of the output $(m_{0,1}, \ldots, m_{0,i-1}, f_k(\vec{m}_0))$ of the function $\mathcal{F}^{(i)}$. In addition, the assumption that $\mathcal{F}$ is $\epsilon(k)$-blockwise-hard-to-invert with respect to $\mathcal{M}_0$ implies that for every $i \in \{1, \ldots, t\}$ the function $\mathcal{F}^{(i)}$ is $\epsilon(k)$-hard-to-invert also with respect to the distribution $\widetilde{\mathcal{M}}_0^{(i)}$: the distribution $\widetilde{\mathcal{M}}_0^{(i)}$ outputs the value $m_{0,i}$, whereas the output of the function $\mathcal{F}^{(i)}$ is $(m_{0,1}, \ldots, m_{0,i-1}, f_k(\vec{m}_0))$.

Bounding $\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A',\mathcal{U},\mathcal{M}_1,\mathcal{F}'}(k)$ This is a specific case of the previous one since no auxiliary input is available to the adversary (recall that $\mathcal{F}'$ is any constant function). Thus, the same analysis shows that there exist probabilistic polynomial-time algorithms $\{B_1^{(i)}\}_{i\in\{1,\ldots,t(k)\}}$, efficiently samplable distributions $\{\mathcal{M}_1^{(i)}, \widetilde{\mathcal{M}}_1^{(i)}\}_{i\in\{1,\ldots,t(k)\}}$, and efficiently computable functions $\{\mathcal{F}_1^{(i)}\}_{i\in\{1,\ldots,t(k)\}}$ such that each $\mathcal{F}_1^{(i)}$ is $\epsilon(k)$-hard-to-invert with respect to both $\mathcal{M}_1^{(i)}$ and $\widetilde{\mathcal{M}}_1^{(i)}$, and it holds that

$$\mathrm{Adv}^{\mathsf{PRIV-IND}}_{\Pi,A',\mathcal{U},\mathcal{M}_1,\mathcal{F}'}(k) \le \sum_{i=1}^{t} \mathrm{Adv}^{\mathsf{PRIV1-IND}}_{\Pi,B_1^{(i)},\mathcal{M}_1^{(i)},\widetilde{\mathcal{M}}_1^{(i)},\mathcal{F}_1^{(i)}}(k). \qquad \square$$

6. A Scheme Based on the d-Linear Assumption

In this section we present our d-linear-based deterministic encryption scheme. The scheme is presented in Sect. 6.1, where we also discuss its homomorphic properties. Security with respect to auxiliary inputs in proved in Sect. 6.2, and a generalization to the multi-user is proved in Sect. 6.3.

6.1. *The Scheme* Π_{Lin}

We show that the d-linear-based lossy trapdoor function of Freeman et al. [14] is in fact a deterministic public-key encryption that is secure with respect to hard-to-invert auxiliary inputs. We note that the lossy trapdoor function of Freeman et al. [14] is a generalization of the one of Peikert and Waters [25], which was shown by Boldyreva, Fehr and O'Neill [5] to be a secure deterministic public-key encryption scheme without auxiliary inputs.

The Scheme 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. The scheme is parameterized by the security parameter k and the message length $n = n(k)$.

- **Key generation**. The 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 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 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 a case it outputs $\mathbf{m}$, and otherwise it outputs $\perp$.

Correctness follows immediately as in [14].

Homomorphism 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 [7] and in [16]. We stress that in such a case we base the security of the scheme on the d-linear assumption for $d \geq 2$ (as the 1-linear, i.e. DDH, in general might not hold in such a group without introducing additional assumptions). 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 $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ 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 $g_T \overset{\text{def}}{=} e(g, g)$ 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.

6.2. *Proof of Security*

We now prove that the scheme Π_{Lin} is secure with respect to any blockwise-hard-to-invert auxiliary input with subexponential hardness. As shown in Sect. 5.5, it suffices to show that PRIV1-IND-security holds with respect to the same hardness. We prove the following theorem:

Theorem 6.1. *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 PRIV1-IND-secure with respect to 2^{-n^μ}-hard-to-invert auxiliary inputs.*

A corollary for blockwise-hard-to-invert auxiliary inputs then immediately follows.

Corollary 6.2. *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^μ}-blockwise-hard-to-invert auxiliary inputs.*

Before providing the formal proof of Theorem 6.1, we first describe the main ideas underlying the security of the scheme. 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 subexponentially 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 $(pk, \mathsf{Enc}_{pk}(\mathbf{m}), f(\mathbf{m})) \overset{c}{\approx} (pk, g^{\mathbf{u}}, f(\mathbf{m}))$, for a uniformly chosen vector $\mathbf{u}$. Transforming this into either one of our notions of security from Sect. 4 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. First, we replace the public key pk with a malformed, but computationally indistinguishable, public key $\widetilde{pk} \overset{c}{\approx} pk$: 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 from Sect. 2.2. 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.

The following lemma is the main technical ingredient in the proof of Theorem 6.1.

Lemma 6.3. *Let $d \in \mathbb{N}$ be some integer and let $n = n(k)$ be a polynomial. Let $\mathcal{D} = \{\mathcal{D}_k\}$ be a distribution ensemble over $\{0,1\}^n$ and let $\mathcal{F} = \{f_k : \{0,1\}^n \to \{0,1\}^*\}_{k\in\mathbb{N}}$ be a collection of ϵ-hard-to-invert functions with respect to $\mathcal{D}$, for $\epsilon \le \frac{\mathrm{negl}(k)}{q^{2d}}$. Then under the d-linear assumption,*

$$\left\{\left(g^{\mathbf{A}}, g^{\mathbf{Ax}}, f(\mathbf{x})\right)\right\}_{k\in\mathbb{N}} \overset{c}{\approx} \left\{\left(g^{\mathbf{A}}, g^{\mathbf{u}}, f(\mathbf{x})\right)\right\}_{k\in\mathbb{N}},$$

where $(\mathbb{G}, q, g) \leftarrow \mathsf{GroupGen}(1^k)$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times n}$, $\mathbf{x} \leftarrow \mathcal{D}_k$, *and* $\mathbf{u} \leftarrow \mathbb{Z}_q^n$.

Proof. We prove the lemma by a series of four hybrid distributions $H_0, \ldots, H_3$, where the hybrids H_0 and H_3 are the two distributions under consideration in the statement of the lemma. That is,

$$H_0 = \left(g^{\mathbf{A}}, g^{\mathbf{Ax}}, f(\mathbf{x})\right),$$
$$H_3 = \left(g^{\mathbf{A}}, g^{\mathbf{u}}, f(\mathbf{x})\right).$$

The intermediate hybrids H_1, H_2, and H_3 are defined as follows.

1. Hybrid H_1 is obtained from hybrid H_0 by replacing $g^{\mathbf{A}}$ with $g^{\mathbf{R}\cdot\mathbf{B}}$, where $\mathbf{R} \leftarrow \mathbb{Z}_q^{n\times d}$ and $\mathbf{B} \leftarrow \mathbb{Z}_q^{d\times n}$. That is,

$$H_1 = \left(g^{\mathbf{R}\cdot\mathbf{B}}, g^{\mathbf{R}\cdot\mathbf{Bx}}, f(\mathbf{x})\right),$$

and computational indistinguishability of H_0 and H_1 follows directly from the d-linear assumption.

2. Hybrid H_2 is obtained from hybrid H_1 by replacing $g^{\mathbf{R}\cdot\mathbf{B}\mathbf{x}}$ with $g^{\mathbf{R}\cdot\mathbf{v}}$, where $\mathbf{v} \leftarrow \mathbb{Z}_q^d$. That is,

   $$H_2 = \left(g^{\mathbf{R}\cdot\mathbf{B}}, g^{\mathbf{R}\cdot\mathbf{v}}, f(\mathbf{x})\right).$$

   Using Theorem 2.2 with the field $\mathbb{F} = \mathrm{GF}(q^d)$, we have

   $$\left(\mathbf{B}, \mathbf{B}\cdot\mathbf{x}, f(\mathbf{x})\right) \overset{c}{\approx} \left(\mathbf{B}, \mathbf{v}, f(\mathbf{x})\right),$$

   where $\mathbf{v} \leftarrow \mathbb{Z}_q^d$.

   To see this, we consider an efficient isomorphism between $\mathrm{GF}(q^d)$ and $\mathbb{Z}_q^d$: Thinking of $\mathrm{GF}(q^d)$ as an extension field over $\mathbb{Z}_q$, the elements of $\mathrm{GF}(q^d)$ correspond to degree-$(d-1)$ polynomials over $\mathbb{Z}_q$. The coefficient vector of such polynomial is exactly a vector in $\mathbb{Z}_q^d$.

   Since there is an (efficient) isomorphism between the columns of $\mathbf{B}$ and elements $\beta_1, \ldots, \beta_n \in \mathrm{GF}(q^d)$ such that $\mathbf{B}\cdot\mathbf{x} = \sum_{i\in[n]} x_i \cdot \beta_i = \langle \mathbf{x}, \boldsymbol{\beta}\rangle$, therefore H_1 and H_2 are computationally indistinguishable.
3. Hybrid H_3 is obtained from hybrid H_2 by replacing $g^{\mathbf{R}\cdot[\mathbf{B}\|\mathbf{v}]}$ with $g^{[\mathbf{A}\|\mathbf{u}]}$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times n}$ and $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. That is,

   $$H_3 = \left(g^{\mathbf{A}}, g^{\mathbf{u}}, f(\mathbf{x})\right),$$

   and computational indistinguishability of H_2 and H_3 follows by the d-linear assumption.

□

We can now go back to proving Theorem 6.1.

Proof of Theorem 6.1. Let $d \in \mathbb{N}$ be as in the theorem statement. Let $\mu > 0$ be some constant. Choosing $n(k) \geq k^{O(1/\mu)} = \mathrm{poly}(k)$ results in having

$$2^{-n^\mu} \leq \frac{\mathrm{negl}(k)}{q^{2d}}.$$

We denote the latter term by ϵ and note that if a function ensemble is 2^{-n^μ}-hard-to-invert then it is also ϵ-hard-to-invert.

Let $\mathcal{F} = \{f_k\}_k$ be ϵ-hard-to-invert with respect to distributions $\mathcal{M}_{0,k}$ and $\mathcal{M}_{1,k}$ over $\{0,1\}^n$, and recall that $pk = g^{\mathbf{A}}$ and $\mathsf{Enc}_{pk}(\mathbf{m}) = g^{\mathbf{A}\cdot\mathbf{m}}$ for any message $\mathbf{m}$. Letting $\mathbf{m}_0 \leftarrow \mathcal{M}_{0,k}$ and $\mathbf{m}_1 \leftarrow \mathcal{M}_{1,k}$, Lemma 6.3 asserts that there exists a distribution $\mathcal{U}$ that is independent of pk, $\mathcal{M}_{0,k}$, and $\mathcal{M}_{1,k}$, such that

$$\left(pk, \mathsf{Enc}_{pk}(\mathbf{m}_0), f_k(\mathbf{m}_0)\right) \overset{c}{\approx} \left(pk, \alpha, f_k(\mathbf{m}_0)\right),$$

$$\left(pk, \mathsf{Enc}_{pk}(\mathbf{m}_1), f_k(\mathbf{m}_1)\right) \overset{c}{\approx} \left(pk, \alpha, f_k(\mathbf{m}_1)\right),$$

where $\alpha \leftarrow \mathcal{U}$.[11] A straightforward hybrid argument implies that

$$\begin{aligned}&\big(pk, \mathsf{Enc}_{pk}(\mathbf{m}_0), \mathsf{Enc}_{pk}(\mathbf{m}_1), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big)\\ &\quad\overset{c}{\approx} \big(pk, \alpha_0, \mathsf{Enc}_{pk}(\mathbf{m}_1), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big)\\ &\quad\overset{c}{\approx} \big(pk, \alpha_0, \alpha_1, f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big),\end{aligned}$$

where $\alpha_0, \alpha_1 \leftarrow \mathcal{U}$. We now apply a symmetric argument to the above: Since α_0, α_1 are identically distributed, it follows that

$$\begin{aligned}&\big(pk, \mathsf{Enc}_{pk}(\mathbf{m}_0), \mathsf{Enc}_{pk}(\mathbf{m}_1), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big)\\ &\quad\overset{c}{\approx} \big(pk, \alpha_1, \alpha_0, f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big)\\ &\quad\overset{c}{\approx} \big(pk, \alpha_1, \mathsf{Enc}_{pk}(\mathbf{m}_0), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big)\\ &\quad\overset{c}{\approx} \big(pk, \mathsf{Enc}_{pk}(\mathbf{m}_1), \mathsf{Enc}_{pk}(\mathbf{m}_0), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big).\end{aligned}$$

We have that for the scheme Π_{Lin}:

$$\begin{aligned}&\big(pk, \mathsf{Enc}_{pk}(\mathbf{m}_0), \mathsf{Enc}_{pk}(\mathbf{m}_1), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big)\\ &\quad\overset{c}{\approx} \big(pk, \mathsf{Enc}_{pk}(\mathbf{m}_1), \mathsf{Enc}_{pk}(\mathbf{m}_0), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big),\end{aligned}$$

which implies PRIV1-sIND security and thus PRIV1-IND security. □

6.3. *The Multi-User Setting*

We now show that Π_{Lin} is secure with respect to auxiliary inputs even in the multi-user setting (see Definition 4.5). We allow any polynomial number of users, and for simplicity we assume that each public key encrypts one message (this corresponds to $\ell(k) = \mathrm{poly}(k)$ and $t(k) = 1$ in the discussion preceding Definition 4.5—which we denote by PRIV1-IND-MU). As in the single-user setting, this naturally 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 6.4 (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. 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 the arithmetic is over $\mathbb{Z}_q$).

[11] Recall that the distribution $\mathcal{U}$ is an abbreviation for $g^{\mathbf{u}}$, where $\mathbf{u} \leftarrow \mathbb{Z}_q^n$.

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 [8]). We can now state the multi-user security of Π_{Lin}.

Theorem 6.5. *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 $2^{-n^{\mu}}$-hard-to-invert auxiliary inputs.*

The idea behind the proof here is an extension of the one from Theorem 6.1. Consider a case of $\ell = \ell(k)$ users, each with their own public key $g^{\mathbf{A}_i}$. Consider encrypting linearly related messages $\mathbf{V}_i \cdot \mathbf{x}$ with these keys (for the sake of simplicity, we ignore the offsets $\mathbf{w}_i$ in this intuitive explanation). Then the sequence of keys and ciphertexts $(g^{\mathbf{A}_i}, g^{\mathbf{A}_i \cdot \mathbf{V}_i \cdot \mathbf{x}})$ can be written as $(g^{\mathbf{B}_i \cdot \mathbf{V}_i^{-1}}, g^{\mathbf{B}_i \cdot \mathbf{x}})$, for uniform $\mathbf{B}_i$'s. This can be transformed, similarly to Lemma 6.2 (omitting some technical details), into a distribution $(g^{\mathbf{B}_i \cdot \mathbf{V}_i^{-1}}, g^{\mathbf{u}_i})$, even given the auxiliary input. Thus, again, the ciphertexts of all users become independent of all other variables which enables proving security. We next prove a lemma that extends Lemma 6.3 and is used towards the formal proof, which follows.

Lemma 6.6. *Let $d \in \mathbb{N}$ be some integer and let $n = n(k)$, $\ell = \ell(k)$ be polynomials and let $\mathcal{M}$ be an affine distribution ensemble. Let $\mathcal{F} = \{f_k\}$ be a collection of $\epsilon = (\frac{\text{negl}(k)}{q^{2d}})$-hard-to-invert functions w.r.t. $\mathcal{M}$. Then for $\mathbf{A}_1, \dots, \mathbf{A}_\ell \leftarrow \mathbb{Z}_q^{n \times n}$, $\vec{\mathbf{m}} = (\mathbf{m}_1, \dots, \mathbf{m}_\ell) \leftarrow \mathcal{M}_k$, $\mathbf{u}_1, \dots, \mathbf{u}_\ell \leftarrow \mathbb{Z}_q^n$, it holds that*

$$\left(1^k, \left(g^{\mathbf{A}_1}, g^{\mathbf{A}_1 \cdot \mathbf{m}_1}\right), \dots, \left(g^{\mathbf{A}_\ell}, g^{\mathbf{A}_\ell \cdot \mathbf{m}_\ell}\right), f_k(\vec{\mathbf{m}})\right)$$
$$\stackrel{c}{\approx} \left(1^k, \left(g^{\mathbf{A}_1}, g^{\mathbf{u}_1}\right), \dots, \left(g^{\mathbf{A}_\ell}, g^{\mathbf{u}_\ell}\right), f_k(\vec{\mathbf{m}})\right).$$

Proof. We prove by a series of hybrids. The zeroth hybrid is, as expected,

$$H_0 \stackrel{\text{def}}{=} \left(1^k, \left(g^{\mathbf{A}_1}, g^{\mathbf{A}_1 \cdot \mathbf{m}_1}\right), \dots, \left(g^{\mathbf{A}_\ell}, g^{\mathbf{A}_\ell \cdot \mathbf{m}_\ell}\right), f_k(\vec{\mathbf{m}})\right).$$

As a first step, we will change our notation slightly. We define $\mathbf{x} \stackrel{\text{def}}{=} \mathbf{m}_1$, $\mathbf{V}_1 \stackrel{\text{def}}{=} \mathbf{I}_n$, $\mathbf{w}_1 = \mathbf{0}$. Namely, now for all $i \in [\ell]$ it holds that $\mathbf{m}_i = \mathbf{V}_i \cdot \mathbf{x} + \mathbf{w}_i$. We define $\mathcal{D}$ to be the marginal distribution of the first element in $\mathcal{M}$. Namely, $\mathbf{x}$ is distributed according to $\mathcal{D}$. The vector $\vec{\mathbf{m}}$ can be efficiently computed as a deterministic function of $\mathbf{x}$ (by computing all $\mathbf{V}_i$'s and $\mathbf{w}_i$'s and using them to compute the respective values). Thus we can consider $\mathcal{F}' = \{f'_k\}_k$ where $f'_k(\mathbf{x})$ first computes $\vec{\mathbf{m}}$ and then outputs $f_k(\vec{\mathbf{m}})$. The collection $\mathcal{F}'$ is ϵ-hard-to-invert on the distribution $\mathcal{D}$. Using this new notation, we have that

$$H_0 = \left(1^k, \left(\left(g^{\mathbf{A}_i}, g^{\mathbf{A}_i \cdot (\mathbf{V}_i \cdot \mathbf{x} + \mathbf{w}_i)}\right)\right)_{i \in [\ell]}, f'_k(\mathbf{x})\right).$$

Our hybrid argument proceeds from here.

1. In the hybrid H_1, we change the distribution of $\mathbf{A}_i$, for all $i \in [\ell]$, into $\mathbf{A}_i \overset{\mathsf{def}}{=} \mathbf{R}_i \cdot \mathbf{B} \cdot \mathbf{V}_i^{-1}$, where $\mathbf{R}_i \leftarrow \mathbb{Z}_q^{n\times n}$ and $\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times n}$. This distribution is statistically close to the original one (it would be identical if $\mathbf{B}$ was invertible, which happens with all but $O(1/q)$ probability). We get that $H_0 \overset{s}{\approx} H_1$ where

$$H_1 \overset{\mathsf{def}}{=} \left(1^k, \left(\left(g^{\mathbf{R}_i\cdot\mathbf{B}\cdot\mathbf{V}_i^{-1}}, g^{\mathbf{R}_i\cdot\mathbf{B}\cdot\mathbf{x}+\mathbf{R}_i\cdot\mathbf{B}\cdot\mathbf{V}_i^{-1}\cdot\mathbf{w}_i}\right)\right)_{i\in[\ell]}, f'_k(\mathbf{x})\right).$$

2. We now use Lemma 6.3 from above to argue that the ϵ-hardness of f' on the distribution $\mathcal{D}$ implies that

$$\left(g^{\mathbf{B}}, g^{\mathbf{B}\cdot\mathbf{x}}, f'_k(\mathbf{x})\right) \overset{c}{\approx} \left(g^{\mathbf{B}}, g^{\mathbf{u}}, f'_k(\mathbf{x})\right),$$

for $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. Plugging this into the previous hybrid, we get that $H_1 \overset{c}{\approx} H_2$ where

$$H_2 \overset{\mathsf{def}}{=} \left(1^k, \left(\left(g^{\mathbf{R}_i\cdot\mathbf{B}\cdot\mathbf{V}_i^{-1}}, g^{\mathbf{R}_i\cdot\mathbf{u}+\mathbf{R}_i\cdot\mathbf{B}\cdot\mathbf{V}_i^{-1}\cdot\mathbf{w}_i}\right)\right)_{i\in[\ell]}, f'_k(\mathbf{x})\right).$$

3. Denote $\mathbf{C} = [\mathbf{B} \| \mathbf{u}] \in \mathbb{Z}_q^{n\times(n+1)}$ and $\mathbf{R} = [\mathbf{R}_1^T \| \cdots \| \mathbf{R}_\ell^T] \in \mathbb{Z}_q^{n\times nr}$. Note that both matrices are uniform in their respective domains. For $n \geq d$, the d-linear assumption implies that $g^{\mathbf{R}^T\cdot\mathbf{C}}$ is computationally indistinguishable from uniform. Breaking the big matrix into blocks, we have that

$$\left(\left(g^{\mathbf{R}_i\cdot\mathbf{B}}, g^{\mathbf{R}_i\cdot\mathbf{u}}\right)\right)_{i\in[\ell]} \overset{c}{\approx} \left(\left(g^{\mathbf{S}_i}, g^{\mathbf{t}_i}\right)\right)_{i\in[\ell]},$$

for $\mathbf{S}_i \leftarrow \mathbb{Z}_q^{n\times n}$, $\mathbf{t}_i \leftarrow \mathbb{Z}_q^n$. Plugging this into H_2, we get that $H_2 \overset{c}{\approx} H_3$ where

$$H_2 \overset{\mathsf{def}}{=} \left(1^k, \left(\left(g^{\mathbf{S}_i\cdot\mathbf{V}_i^{-1}}, g^{\mathbf{t}_i+\mathbf{S}_i\cdot\mathbf{V}_i^{-1}\cdot\mathbf{w}_i}\right)\right)_{i\in[\ell]}, f'_k(\mathbf{x})\right).$$

However, noticing that $\mathbf{S}_i \cdot \mathbf{V}_i^{-1}$ is uniform in $\mathbb{Z}_q^{n\times n}$ and that $\mathbf{t}_i + \mathbf{S}_i \cdot \mathbf{V}_i^{-1} \cdot \mathbf{w}_i$ is uniform in $\mathbb{Z}_q^n$, and plugging back $f'_k(\mathbf{x}) = f_k(\vec{\mathbf{m}})$, we get that H_3 is identically distributed to

$$\left(1^k, \left(g^{\mathbf{A}_1}, g^{\mathbf{u}_1}\right), \ldots, \left(g^{\mathbf{A}_\ell}, g^{\mathbf{u}_\ell}\right), f_k(\vec{\mathbf{m}})\right),$$

and the result follows. □

Deriving Theorem 6.5 from Lemma 6.6 is now very similar to the derivation of Theorem 6.1 from Lemma 6.3.

Proof of Theorem 6.5. Let $d \in \mathbb{N}$ be as in the theorem statement. Let $0 < \mu < 1$ be some constant. Choosing $n(k) \geq k^{O(1/\mu)} = \text{poly}(k)$ results in having

$$2^{-n^\mu} \leq \frac{\text{negl}(k)}{q^{2d}}.$$

We denote the latter term by ϵ and note that if a function ensemble is 2^{-n^μ}-hard-to-invert then it is also ϵ-hard-to-invert.

Let $\mathcal{F} = \{f_k\}_{k \in \mathbb{N}}$ be a class of functions that are ϵ-hard-to-invert for affine distributions $\mathcal{M}_{0,k}, \mathcal{M}_{1,k} \subseteq ((\{0,1\}^n)^1)^\ell$, where $\epsilon \le \mathrm{negl}(k)/q^{2d}$ and $n(k), \ell(k)$ are polynomials. Letting $\vec{pk}$ denote a vector of length $\ell(k)$ of properly distributed public keys for Π_{Lin}, and letting $\vec{\vec{\mathbf{m}}}_0 \leftarrow \mathcal{M}_{0,k}$, $\vec{\vec{\mathbf{m}}}_1 \leftarrow \mathcal{M}_{1,k}$, Lemma 6.6 asserts that in there exists a distribution $\mathcal{U}'$ that is independent of $\vec{pk}, \mathcal{M}_0, \mathcal{M}_1$ such that

$$\bigl(\vec{pk}, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_0)\bigr) \overset{c}{\approx} \bigl(\vec{pk}, \alpha, f_k(\vec{\vec{\mathbf{m}}}_0)\bigr),$$

$$\bigl(\vec{pk}, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_1), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr) \overset{c}{\approx} \bigl(\vec{pk}, \alpha, f_k(\vec{\vec{\mathbf{m}}}_1)\bigr),$$

where $\alpha \leftarrow \mathcal{U}'$.[12]

It follows therefore that

$$\begin{aligned} &\bigl(\vec{pk}, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_0), \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_1), f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr) \\ &\quad \overset{c}{\approx} \bigl(\vec{pk}, \alpha_0, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_1), f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr) \\ &\quad \overset{c}{\approx} \bigl(\vec{pk}, \alpha_0, \alpha_1, f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr), \end{aligned}$$

where $\alpha_0, \alpha_1 \leftarrow \mathcal{U}'$. Since α_0, α_1 are identically distributed, it follows that

$$\begin{aligned} &\bigl(\vec{pk}, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_0), \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_1), f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr) \\ &\quad \overset{c}{\approx} \bigl(\vec{pk}, \alpha_1, \alpha_0, f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr) \\ &\quad \overset{c}{\approx} \bigl(\vec{pk}, \alpha_1, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr) \\ &\quad \overset{c}{\approx} \bigl(\vec{pk}, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_{\mathbf{1}}), \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr). \end{aligned}$$

We have that for the scheme Π_{Lin}:

$$\begin{aligned} &\bigl(\vec{pk}, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_0), \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_1), f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr) \\ &\quad \overset{c}{\approx} \bigl(\vec{pk}, \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_1), \vec{\mathsf{Enc}}_{\vec{pk}}(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_0), f_k(\vec{\vec{\mathbf{m}}}_1)\bigr), \end{aligned}$$

which implies that Definition 4.5 holds (in fact it implies a stronger multi-user security, in the spirit of Definition 4.4 for a single user). □

7. A Scheme Based on Subgroup Indistinguishability Assumptions

In this section we present our second deterministic public-key encryption scheme, which is based on a rather general class of subgroup indistinguishability assumptions (including, in particular, the quadratic residuosity assumption and Paillier's composite residuosity assumption—see Sect. 2.1). In Sect. 7.1 we describe the scheme, and in Sect. 7.2 we prove its security with respect to hard-to-invert auxiliary inputs.

[12] Recall that the distribution $\mathcal{U}'$ is an abbreviation for $(g^{\mathbf{u}_1}, \ldots, g^{\mathbf{u}_\ell})$, where $\mathbf{u}_i \leftarrow \mathbb{Z}_q^n$.

7.1. *The Scheme Π_{SGI}*

We show that a generalization of the QR-based lossy trapdoor function of Hemenway and Ostrovsky [21], which can be based on subgroup indistinguishability, is in fact a deterministic public-key encryption scheme that is secure against subexponentially hard-to-invert auxiliary inputs.

Let $\mathsf{GroupGen}$ be the generating algorithm from Sect. 2.1 which on input a security parameter 1^k outputs the tuple $\mathsf{params} = (\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L, h, T)$. We let $y \leftarrow g^x$ denote an application of an isomorphism transforming an element x in the module $\mathbb{M}_{\mathbb{G}_L}$ into an element y in the group $\mathbb{G}_L$ (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 $\mathbb{G}_U$ and the corresponding module, such that the generating set that corresponds to $\hat{g}$ is the same as that of g, appended with h. Our scheme is parameterized by the security parameter k and the message length $n = n(k)$.

- **Key Generation**. The algorithm $\mathsf{KeyGen}(1^k)$ samples $\mathsf{params} \leftarrow \mathsf{GroupGen}(1^k)$ (recall that $\mathsf{params} = (\mathbb{G}_U, \mathbb{G}_M, \mathbb{G}_L, h, T)$), a vector $g^{\mathbf{w}^T} \leftarrow {\mathbb{G}_L}^n$, and a vector $\mathbf{r} \leftarrow ([T^2])^n$. It then outputs $pk = (\mathsf{params}, g^{\mathbf{w}^T}, h^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T})$ and $sk = \mathbf{r}$.

  The matrix dot product above refers to element-wise multiplication:

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

  To be completely explicit, we emphasize that $pk \in \{0,1\}^* \times {\mathbb{G}_U}^{1\times n} \times {\mathbb{G}_U}^{n\times n}$ and $sk \in \mathbb{N}^n$.
- **Encryption**. The algorithm $\mathsf{Enc}_{pk}(\mathbf{m})$, where $pk = (\mathsf{params}, \hat{g}^{\mathbf{w}^T}, \hat{g}^{\mathbf{T}})$ 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 $c \in \mathbb{G}_U \times {\mathbb{G}_U}^n$.

  For a legally generated public key $pk = (\mathsf{params}, g^{\mathbf{w}^T}, h^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T})$ and $sk = \mathbf{r}$, we get $c = (g^{\mathbf{w}^T\cdot\mathbf{m}}, h^{\mathbf{m}} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T\cdot\mathbf{m}})$.
- **Decryption**. The 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 $h^{\mathbf{m}}$, for $\mathbf{m} \in \{0,1\}^n$, then it outputs $\mathbf{m}$ and otherwise it outputs $\perp$.

Correctness follows immediately by definition.

7.2. *Proof of Security*

We now prove that the scheme Π_{SGI} is secure with respect to any blockwise-hard-to-invert auxiliary input with subexponential hardness. As shown in Sect. 5.5, it suffices to show that PRIV1-IND-security holds with respect to the same hardness. We prove the following theorem:

Theorem 7.1. *Under the subgroup indistinguishability assumption, for any constant* $0 < \mu < 1$ *and for any sufficiently large message length* $n = n(k)$, *the scheme* Π_{SGI} *is PRIV*1*-IND-secure with respect to* $2^{-n^{\mu}}$*-hard-to-invert auxiliary inputs.*

Using Lemma 5.5, we have:

Corollary 7.2. *Under the subgroup indistinguishability assumption, for any constant* $0 < \mu < 1$ *and for any sufficiently large message length* $n = n(k)$*, the scheme* Π_{SGI} *is PRIV-IND-secure with respect to* $2^{-n^{\mu}}$*-blockwise-hard-to-invert auxiliary inputs.*

In the proof we consider the joint distribution of the public key, the ciphertext, and the auxiliary input $(pk, \mathsf{Enc}_{pk}(\mathbf{m}), f(\mathbf{m}))$. Similarly to Sect. 6, we show that this distribution is indistinguishable from one where the ciphertext does not depend on $\mathbf{m}$.

As a first step, we will modify the distribution of public keys. We consider a malformed, but computationally indistinguishable, public key $\widetilde{pk} \overset{c}{\approx} pk$. Specifically, while $pk = (\mathsf{params}, g^{\mathbf{w}^T}, h^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T})$, we switch to $\widetilde{pk} = (\mathsf{params}, g^{\mathbf{w}^T}, g^{\mathbf{r}\cdot\mathbf{w}^T})$, thus "losing" the h component of the public key. Subgroup indistinguishability guarantees that the two keys are indistinguishable.[13]

We thus have that $(pk, \mathsf{Enc}_{pk}(\mathbf{m}), f(\mathbf{m})) \overset{c}{\approx} (\widetilde{pk}, \mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}), f(\mathbf{m}))$, where $\mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}) = (g^{\mathbf{w}^T\cdot\mathbf{m}}, g^{\mathbf{r}\cdot\mathbf{w}^T\cdot\mathbf{m}})$. We see that the only dependence of the ciphertext in $\mathbf{m}$ is via $\langle\mathbf{w}, \mathbf{m}\rangle$. We are thus able to use a Goldreich–Levin like theorem to show that $(\mathbf{w}, \langle\mathbf{w}, \mathbf{m}\rangle, f(\mathbf{m})) \overset{c}{\approx} (\mathbf{w}, u, f(\mathbf{m}))$, for some u that is distributed independently of $\mathbf{m}$. We conclude, therefore, that $(\widetilde{pk}, \mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}), f(\mathbf{m})) \overset{c}{\approx} (\widetilde{pk}, (g^{u}, g^{\mathbf{r}\cdot u}), f(\mathbf{m}))$, granted that f is hard enough. Namely, we showed that the ciphertext gives no information on the message, beyond the auxiliary input. A formal proof follows.

Proof of Theorem 7.1. Let $\mu > 0$ be some constant, and recall that $L = |\mathbb{G}_L| \leq 2^{\mathrm{poly}(k)}$. Choosing $n(k) \geq k^{O(1/\mu)} = \mathrm{poly}(k)$ results in having

$$2^{-n^{\mu}} \leq \frac{\mathrm{negl}(k)}{L^{1+\log(8n/\mathrm{negl}(k))}}.$$

We denote the latter term by ϵ and note that if a function ensemble is $2^{-n^{\mu}}$-hard-to-invert then it is also ϵ-hard-to-invert.

Let $\mathcal{M}_0$ and $\mathcal{M}_1$ be distribution ensembles over $\{0,1\}^n$ and let $\mathcal{F} = \{f_k\}$ be a collection of $2^{-n^{\eta}}$-hard-to-invert (and thus also ϵ-hard-to-invert) functions with respect to both $\mathcal{M}_0$ and $\mathcal{M}_1$. In addition, let $pk \leftarrow (g^{\mathbf{w}^T}, h^{\mathbf{I}_n} \cdot g^{\mathbf{r}\cdot\mathbf{w}^T})$ be a properly distributed public key for Π_{SGI} and let $\mathbf{m}_0 \leftarrow \mathcal{M}_{0,k}$, $\mathbf{m}_1 \leftarrow \mathcal{M}_{1,k}$. The proof will follow by a series of hybrids. We begin with the distribution

$$H_0 = \bigl(pk, \mathsf{Enc}_{pk}(\mathbf{m}_0), \mathsf{Enc}_{pk}(\mathbf{m}_1), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\bigr).$$

Our first step is replacing our public key with $\widetilde{pk} \leftarrow (\mathsf{params}, g^{\mathbf{w}^T}, g^{\mathbf{r}\cdot\mathbf{w}^T})$. This distribution is computationally indistinguishable by Lemma 2.1. We thus get

$$H_1 = \bigl(\widetilde{pk}, \mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}_0), \mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}_1), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\bigr).$$

[13] We remark that while pk is a sample from the injective branch of a [21]-like lossy trapdoor function family, $\widetilde{pk}$ is a sample from the lossy branch.

Let us explicitly express $\mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}_0)$, $\mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}_1)$ according to the definition of our encryption algorithm. We have that

$$H_1 = \big(\big(g^{\mathbf{w}^T}, g^{\mathbf{r}\cdot\mathbf{w}^T}\big), \big(g^{\mathbf{w}^T\cdot\mathbf{m}_0}, g^{\mathbf{r}\cdot\mathbf{w}^T\cdot\mathbf{m}_0}\big), \big(g^{\mathbf{w}^T\cdot\mathbf{m}_1}, g^{\mathbf{r}\cdot\mathbf{w}^T\cdot\mathbf{m}_1}\big), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big).$$

We can now use our hard-core Theorem 2.3 (twice) to conclude that since f_k is ϵ-hard-to-invert, then

$$\big(\mathbf{w}^T, \mathbf{w}^T\cdot\mathbf{m}_0, \mathbf{w}^T\cdot\mathbf{m}_1, f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big) \tag{7.1}$$

$$\overset{c}{\approx} \big(\mathbf{w}^T, u_0, \mathbf{w}^T\cdot\mathbf{m}_1, f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big) \tag{7.2}$$

$$\overset{c}{\approx} \big(\mathbf{w}^T, u_0, u_1, f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big), \tag{7.3}$$

where $u_0, u_1 \leftarrow \mathbb{M}$ (the module that corresponds to the $\mathbb{G}_L$ group). Plugging this into our distribution, the next hybrid follows:

$$H_2 = \big(\big(g^{\mathbf{w}^T}, g^{\mathbf{r}\cdot\mathbf{w}^T}\big), \big(g^{u_0}, g^{\mathbf{r}\cdot u_0}\big), \big(g^{u_1}, g^{\mathbf{r}\cdot u_1}\big), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big).$$

However, since u_0 and u_1 are identically distributed, we can use Eqs. (7.1)–(7.3) when switching the roles of u_0 and u_1 to obtain the next hybrid, that is identical to H_1 except the change of rolls between $\mathbf{m}_0$ and $\mathbf{m}_1$.

$$H_3 = \big(\big(g^{\mathbf{w}^T}, g^{\mathbf{r}\cdot\mathbf{w}^T}\big), \big(g^{\mathbf{w}^T\cdot\mathbf{m}_1}, g^{\mathbf{r}\cdot\mathbf{w}^T\cdot\mathbf{m}_1}\big), \big(g^{\mathbf{w}^T\cdot\mathbf{m}_0}, g^{\mathbf{r}\cdot\mathbf{w}^T\cdot\mathbf{m}_0}\big), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big).$$

Writing the latter in the form of encryptions with a malformed public key, we have that

$$H_3 = \big(\widetilde{pk}, \mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}_1), \mathsf{Enc}_{\widetilde{pk}}(\mathbf{m}_0), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big).$$

Applying Lemma 2.1 again, to go back to the original public key yields the final hybrid

$$H_4 = \big(pk, \mathsf{Enc}_{pk}(\mathbf{m}_1), \mathsf{Enc}_{pk}(\mathbf{m}_0), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big).$$

Since $H_0 \overset{c}{\approx} H_4$, it follows that

$$\big(pk, \mathsf{Enc}_{pk}(\mathbf{m}_0), \mathsf{Enc}_{pk}(\mathbf{m}_1), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big)$$
$$\overset{c}{\approx} \big(pk, \mathsf{Enc}_{pk}(\mathbf{m}_1), \mathsf{Enc}_{pk}(\mathbf{m}_0), f_k(\mathbf{m}_0), f_k(\mathbf{m}_1)\big),$$

and the strongest PRIV1-sIND security follows. □

Acknowledgements

We thank Ilya Mironov, Omkant Pandey, and the anonymous reviewers for their comments.

References

[1] A. Akavia, S. Goldwasser, V. Vaikuntanathan, Simultaneous hardcore bits and cryptography against memory attacks, in *Proceedings of the 6th Theory of Cryptography Conference* (2009), pp. 474–495

[2] M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, S. Yilek, Hedged public-key encryption: how to protect against bad randomness, in *Advances in Cryptology, ASIACRYPT '09* (2009), pp. 232–249

[3] M. Bellare, A. Boldyreva, A. O'Neill, Deterministic and efficiently searchable encryption, in *Advances in Cryptology, CRYPTO '07* (2007), pp. 535–552

[4] M. Bellare, M. Fischlin, A. O'Neill, T. Ristenpart, Deterministic encryption: definitional equivalences and constructions without random oracles, in *Advances in Cryptology, CRYPTO '08* (2008), pp. 360–378

[5] A. Boldyreva, S. Fehr, A. O'Neill, On notions of security for deterministic encryption, and efficient constructions without random oracles, in *Advances in Cryptology, CRYPTO '08* (2008), pp. 335–359

[6] Z. Brakerski, S. Goldwasser, Circular and leakage resilient public-key encryption under subgroup indistinguishability (or: quadratic residuosity strikes back), in *Advances in Cryptology, CRYPTO '10* (2010)

[7] D. Boneh, E.-J. Goh, K. Nissim, Evaluating 2-DNF formulas on ciphertexts, in *Proceedings on the 2nd Theory of Cryptography Conference* (2005), pp. 325–341

[8] D. Boneh, S. Halevi, M. Hamburg, R. Ostrovsky, Circular-secure encryption from decision Diffie–Hellman, in *Advances in Cryptology, CRYPTO '08* (2008), pp. 108–125

[9] R. Canetti, Towards realizing random oracles: hash functions that hide all partial information, in *Advances in Cryptology, CRYPTO '97* (1997), pp. 455–469

[10] R. Canetti, Universally composable security: a new paradigm for cryptographic protocols, in *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science* (2001), pp. 136–145

[11] Y. Dodis, S. Goldwasser, Y. Tauman Kalai, C. Peikert, V. Vaikuntanathan, Public-key encryption schemes with auxiliary inputs, in *Proceedings of the 7th Theory of Cryptography Conference* (2010), pp. 361–381

[12] Y. Dodis, Y. Tauman Kalai, S. Lovett, On cryptography with auxiliary input, in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing* (2009), pp. 621–630

[13] Y. Dodis, A. Smith, Entropic security and the encryption of high entropy messages, in *Proceedings of the 2nd Theory of Cryptography Conference* (2005), pp. 556–577

[14] D.M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, 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* (2010), pp. 279–295

[15] B. Fuller, A. O'Neill, L. Reyzin, A unified approach to deterministic encryption: new constructions and a connection to computational entropy, in *Proceedings of the 9th Theory of Cryptography Conference* (2012), pp. 582–599

[16] C. Gentry, S. Halevi, V. Vaikuntanathan, A simple BGN-type cryptosystem from LWE, in *Advances in Cryptology, EUROCRYPT '10* (2010), pp. 506–522

[17] S. Goldwasser, Y. Tauman Kalai, On the impossibility of obfuscation with auxiliary input, in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science* (2005), pp. 553–562

[18] O. Goldreich, L.A. Levin, A hard-core predicate for all one-way functions, in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing* (1989), pp. 25–32

[19] S. Goldwasser, S. Micali, Probabilistic encryption. *J. Comput. Syst. Sci.* **28**(2), 270–299 (1984)

[20] O. Goldreich, *Foundations of Cryptography, Vol. 2: Basic Applications* (Cambridge University Press, Cambridge, 2004)

[21] B. Hemenway, R. Ostrovsky, Lossy trapdoor functions from smooth homomorphic hash proof systems. Report TR09-127, Electronic Colloquium on Computational Complexity (2009)

[22] I. Mironov, O. Pandey, O. Reingold, G. Segev, Incremental deterministic public-key encryption, in *Advances in Cryptology, EUROCRYPT '12*. Lecture Notes in Computer Science, vol. 7237 (Springer, Berlin, 2012), pp. 628–644

[23] M. Naor, G. Segev, Public-key cryptosystems resilient to key leakage, in *Advances in Cryptology, CRYPTO '09* (2009), pp. 18–35

[24] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in *Advances in Cryptology, EUROCRYPT '99* (1999), pp. 223–238

[25] C. Peikert, B. Waters, Lossy trapdoor functions and their applications. *SIAM J. Comput.* **40**(6), 1803–1844 (2011)

[26] A. Russell, H. Wang, How to fool an unbounded adversary with a short key. *IEEE Trans. Inf. Theory* **52**(3), 1130–1140 (2006)

[27] H. Wee, Dual projective hashing and its applications—lossy trapdoor functions and more, in *Advances in Cryptology, EUROCRYPT '12*. Lecture Notes in Computer Science, vol. 7237 (Springer, Berlin, 2012), pp. 246–262

[28] D. Wichs, Barriers in cryptography with weak, correlated and leaky sources. IACR Cryptology ePrint Archive, Report 2012/459 (2012)

[29] B. Zhu, K. Li, 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* (2008), pp. 269–282