J Cryptol (2018) 31:1012–1063
https://doi.org/10.1007/s00145-018-9287-y

Journal of
CRYPTOLOGY

Deterministic Public-Key Encryption for Adaptively-Chosen Plaintext Distributions*

Ananth Raghunathan†
Google, Mountain View, CA94043, USA
pseudorandom@google.com

Gil Segev‡
School of Computer Science and Engineering, Hebrew University of Jerusalem, 91904 Jerusalem, Israel
segev@cs.huji.ac.il

Salil Vadhan§
Center for Research on Computation and Society, School of Engineering and Applied Sciences, Harvard University, Cambridge, MA02138, USA
salil@seas.harvard.edu

Communicated by Rafail Ostrovsky.

Received 2 July 2013 / Revised 12 March 2018
Online publication 21 March 2018

Abstract. Bellare, Boldyreva, and O'Neill (CRYPTO '07) initiated the study of deterministic public-key encryption as an alternative in scenarios where randomized encryption has inherent drawbacks. The resulting line of research has so far guaranteed security only for adversarially chosen-plaintext distributions that are *independent* of the public key used by the scheme. In most scenarios, however, it is typically not realistic to assume that adversaries do not take the public key into account when attacking a scheme. We show that it is possible to guarantee meaningful security even for plaintext distributions that depend on the public key. We extend the previously proposed notions of security, allowing adversaries to *adaptively* choose plaintext distributions *after* seeing the public key, in an *interactive* manner. The only restrictions we make are that:

*A preliminary version of this work appeared in *Advances in Cryptology—EUROCRYPT '13*, pp. 93–110, 2013.

†Work done while the author was a Ph.D. student at Stanford University and an intern at Microsoft Research Silicon Valley.

‡Supported by the European Union's 7th Framework Program (FP7) via a Marie Curie Career Integration Grant (Grant No. 618094), by the European Union's Horizon 2020 Framework Program (H2020) via an ERC Grant (Grant No. 714253), by the Israel Science Foundation (Grant No. 483/13), by the Israeli Centers of Research Excellence (I-CORE) Program (Center No. 4/11), by the US-Israel Binational Science Foundation (Grant No. 2014632), and by a Google Faculty Research Award. Work done primarily as a Visiting Faculty at Stanford University and as a postdoctoral researcher at Microsoft Research Silicon Valley.

§Supported in part by NSF Grant CCF-1116616. Work done primarily while on leave as a Visiting Researcher at Microsoft Research Silicon Valley and as a Visiting Scholar at Stanford University.

(1) plaintext distributions are unpredictable (as is essential in deterministic public-key encryption), and (2) the number of plaintext distributions from which each adversary is allowed to adaptively choose is upper bounded by 2^p, where p can be any predetermined polynomial in the security parameter and plaintext length. For example, with $p = 0$ we capture plaintext distributions that are independent of the public key, and with $p = O(s \log s)$ we capture, in particular, all plaintext distributions that are samplable by circuits of size s. Within our framework we present both constructions in the random oracle model based on any public-key encryption scheme, and constructions in the standard model based on lossy trapdoor functions (thus, based on a variety of number-theoretic assumptions). Previously known constructions heavily relied on the independence between the plaintext distributions and the public key for the purposes of randomness extraction. In our setting, however, randomness extraction becomes significantly more challenging once the plaintext distributions and the public key are no longer independent. Our approach is inspired by research on randomness extraction from seed-dependent distributions. Underlying our approach is a new generalization of a method for such randomness extraction, originally introduced by Trevisan and Vadhan (FOCS '00) and Dodis (Ph.D. Thesis, MIT, '00).

Keywords. Public-key encryption, Deterministic encryption, Randomness extraction.

1. Introduction

Deterministic public-key encryption was introduced by Bellare et al. [3] as an alternative in scenarios where randomized encryption has inherent drawbacks. For example, ciphertexts that are produced by a randomized encryption algorithm are not length preserving (i.e., may be longer than their corresponding plaintexts) and are in general not efficient searchable—two properties that are problematic in many applications involving massive amounts of data. In addition, the security guarantees provided by randomized public-key encryption schemes are typically highly dependent on the assumption that fresh and essentially uniform random bits are available—which may not always be a valid assumption.

When using a deterministic encryption algorithm, however, the full-fledged notion of semantic security [15] is out of reach. In this light, Bellare et al. initiated the study of formalizing other strong and meaningful notions of security for deterministic public-key encryption, and quite a significant amount of work has been devoted to proposing various such notions and constructing schemes satisfying them [2–5,7,14,18,25]. Aiming to obtain as-strong-as-possible notions of security, this recent line of research has successfully shown that a natural variant of the notion of semantic security can be guaranteed even when using a deterministic encryption algorithm, as long as plaintexts are: (1) somewhat *unpredictable*, and (2) *independent* of the public key used by the scheme.

Plaintext unpredictability When using a deterministic encryption algorithm, essentially no meaningful notion of security can be satisfied when plaintexts are distributed over a small (e.g., polynomial-sized) set. 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,[1] compare each of them to the given ciphertext

[1] More generally, an adversary can encrypt all plaintexts that occur with at least some non-negligible probability.

c, and thus recover the plaintext m. Therefore, when formalizing a notion of security for deterministic public-key encryption, it is indeed essential to focus on security for unpredictable plaintext distributions.

Key-independent plaintext distributions Even when dealing with highly unpredictable plaintext distributions, some restrictions should be made on their relation to the public key. Consider, for example, the uniform distribution over plaintexts m subject to the restriction that the first bit of m and the first bit of $c = \mathsf{Enc}_{pk}(m)$ are equal.[2] More generally, by constructing plaintext distributions that depends on the public key, adversaries can use any *deterministic* encryption algorithm as a *subliminal channel* that leaks much more information on the plaintext than what any meaningful notion of security should allow.

This paper For preventing adversaries from exploiting deterministic encryption algorithms as subliminal channels, research on deterministic public-key encryption has so far guaranteed security only for plaintext distributions that are independent of the public key used by the scheme (which is not realistic, as an adversary can often influence the plaintext distribution after seeing the public key). In this paper, we ask whether or not this is essential. Namely, is it possible to formalize a meaningful notion of security that allows dependencies between plaintext distributions and keys?

1.1. *Our Contributions*

In this paper, we show that it is *not* essential to focus only on plaintexts distributions that are independent of the keys used by the scheme. We formalize and realize a new notion of security for deterministic public-key encryption, allowing adversaries to *adaptively* choose plaintext distributions *after* seeing the public key of the scheme, in an *interactive* manner. The only restriction we make is that the number of plaintext distributions from which each adversary is allowed to adaptively choose is upper bounded by $2^{p(\lambda)}$, where $p(\lambda)$ can be any predetermined polynomial in the security parameter λ. More specifically, we allow the message length $n = n(\lambda)$ to be any predetermined polynomial in the security parameter, and then allow $p = p(\lambda)$ to be any predetermined polynomial in both the security parameter and the message length. For simplicity, however, throughout this paper we refer to $p = p(\lambda)$ as a function of the security parameter instead of a function $p = p(\lambda, n(\lambda))$ of both the security parameter and the plaintext length.

We stress that the set of $2^{p(\lambda)}$ plaintext distributions can be different for each adversary. Intuitively, this bound says that the entire plaintext distribution (not just a single sample) contains at most $p(\lambda)$ bits of information about the public key. We view this as a natural first model for adaptively chosen-plaintext distributions, particularly in light of the impossibility of handling arbitrary dependencies (as sketched earlier), and hope that it will pave the way for more realistic models.

Our approach is a generalization of the security notions that have been proposed so far. For example, with $p(\lambda) \equiv 0$ we obtain the notion of security introduced by Bellare et al. [3], where the plaintext distribution chosen by the adversary is independent of the public key. As an additional example, with $p(\lambda) = O(s(\lambda) \log s(\lambda))$ we capture,

[2] Note that the support of this distribution will contain nearly half of all plaintexts with high probability.

in particular, all plaintext distributions that are samplable by boolean circuits of size at most $s(\lambda)$.

Within our framework we present both generic constructions in the random oracle model based on any public-key encryption scheme, and generic constructions in the standard model based on lossy trapdoor functions. Our constructions are inspired by the constructions of Bellare et al. [3] and of Boldyreva et al. [5]. These constructions rely on the independence between the plaintext distributions and the keys for the purposes of extracting randomness from the plaintext distributions. Randomness extraction becomes significantly more difficult once the plaintext distributions and the public keys are no longer independent. Challenges along somewhat similar lines arise in the context of deterministic randomness extraction, where one would like to construct seedless randomness extractors, or seeded randomness extractors for seed-dependent distributions. Indeed, underlying our approach is a new generalization of a method for deterministic extraction, originally introduced by Trevisan and Vadhan [22] and Dodis [10].

Finally, our approach naturally extends to the setting of "hedged" public-key encryption schemes, introduced by Bellare et al. [2]. In this setting, one would like to construct randomized schemes that are semantically secure in the standard sense, and maintain a meaningful and realistic notion of security even when "corrupt" randomness is used by the encryption algorithm. Our notions of adaptive security for deterministic public-key encryption give rise to analogous notions for hedged public-key encryption, and our constructions (when used within the framework of Bellare et al. [2][3]) yield the first adaptively secure hedged public-key encryption schemes.

1.2. *Related Work*

The formal study of deterministic public-key encryption was initiated by Bellare et al. [3], following research on symmetric-key encryption of high-entropy messages by Russell and Wang [21] and Dodis and Smith [12]. Bellare et al. formalized several notions of security, which were later refined and extended by Bellare et al. [4], and by Boldyreva et al. [5]. Bellare, Boldyreva, and O'Neill presented constructions in the random oracle model, and constructions in the standard model were first presented by Bellare, Boldyreva, and O'Neill, and additionally by Boldyreva, Fehr, and O'Neill. Brakerski and Segev [7] showed that the min-entropy requirement considered in all previous works on deterministic public-key encryption can be relaxed to consider hard-to-invert auxiliary inputs. Based on specific number-theoretic assumptions, they designed schemes that are secure in the more general auxiliary-input model, and their constructions were later unified by Wee [25]. Progress along similar lines was made by Fuller et al. [14], who presented a scheme that can securely encrypt a small predetermined number of plaintexts with arbitrary dependencies as long as each has high min-entropy. Additional progress in studying deterministic public-key encryption schemes was recently made by Mironov et al. [18] who constructed such schemes with optimal incrementality.

A step toward obtaining adaptive security for deterministic public-key encryption was made by Bellare et al. [2] who defined and constructed "hedged" public-key en-

[3] For example, as part of their generic "pad-then-deterministic" scheme, which deterministically encrypts the concatenation of the plaintext and the randomness.

cryption schemes (discussed in Sect. 1.1). Whereas the notions of security considered in [3–5,7,14,18,25] capture only "single-shot" adversaries (i.e., adversaries that challenge the given scheme with only one plaintext distribution), Bellare et al. [2] showed that it is possible to guarantee security even against "multi-shot" adversaries (i.e., adversaries that interactively challenge the scheme with plaintext distributions depending on previous ciphertexts that they received). In their notion of security, however, adversaries are not given access to the public key that is being attacked. In our work we consider the more general, and more typical, scenario where adversaries are given *direct access* to the public key being attacked (and are allowed to adaptively and interactively choose plaintext distributions depending on previous ciphertexts that they received).[4] As discussed in Sect. 1.1, our constructions yield the first adaptively secure hedged public-key encryption schemes.

1.3. *Overview of Our Approach*

In this section we provide a high-level overview of our notions of security and of the main ideas underlying our constructions. We focus here on our constructions in the standard model (i.e., without random oracles), as these emphasize more clearly the main challenges in designing encryption schemes satisfying our notions of security.

Our notions of security As discussed above, our notions of security for deterministic public-key encryption differ from the previously proposed ones by providing adversaries with *direct* access to the public key. Specifically, we formalize security via a game between an adversary and a "real-or-random" encryption oracle. First, a pair containing a public key and a secret key is produced using the key-generation algorithm of the scheme under consideration, and the adversary is given the public key. Then, the adversary adaptively interacts with the encryption oracle, where each query consists of a description of a plaintext distribution M. For simplicity, here we consider distributions over plaintexts, but in fact our notion allows distributions over blocks of plaintexts. The encryption oracle operates in one of two modes, "real" or "random," which is chosen uniformly at random at the beginning of the game. In the "real" mode, the encryption oracle samples a plaintext according to M, and the adversary is given its encryption under the public key. In the "random" mode, the encryption oracle samples a plaintext from the uniform distribution over the plaintext space, and the adversary is again given its encryption under the public key.[5]

The goal of the adversary in this game is to distinguish between the "real" mode and "random" mode with a non-negligible advantage, subject only to the requirement that for any such adversary there exists a set $\mathcal{X} = \mathcal{X}_\lambda$ of plaintext distributions such that:

[4] In fact, the approach of Bellare et al. [2] relies on encryption schemes in which ciphertexts reveal essentially no information on the corresponding public key. Therefore, even multi-shot adversaries learn essentially no information on the public key being attacked, and thus their "adaptive" choices of plaintext distributions are still independent of the public key. This approach does not seem to extend to our setting, where adversaries are given direct access to the public key.

[5] We note that the resulting notion of security is polynomially equivalent (via a standard hybrid argument) to an analogous "left" or "right" formulation in which the adversary specifies two plaintext distributions, and the encryption oracle uses either the left one of the right one.

1. $|\mathcal{X}| \leq 2^p$, where $p = p(\lambda)$ is any predetermined polynomial in the security parameter (the construction of the scheme can depend on the polynomial p).
2. The adversary queries the encryption oracle only with plaintext distributions in $\mathcal{X}$.
3. Each plaintext distribution in $\mathcal{X}$ has min-entropy at least k, where $k = k(\lambda)$ is a predetermined function of the security parameter.

In addition, we naturally extend the above game to capture chosen-ciphertext attacks, by allowing adversaries adaptive access to a decryption oracle (subject to the standard requirement of not querying the decryption oracle with any ciphertext that was produced by the encryption oracle).

We note that our security game is in fact almost identical to the standard "real-or-random" one for randomized public-key encryption. Specifically, unlike the previously proposed notions of security for deterministic public-key encryption, we provide the adversary with direct access to the public key and allow the adversary to adaptively interact with the encryption and decryption oracles *in any order*.[6]

Chosen-plaintext security in the standard model The starting point for our construction is the one of Boldyreva, Fehr, and O'Neill, which we now briefly describe. In their construction, the public key consists of a function f that is sampled from the injective mode of a collection of lossy trapdoor functions, and a permutation π sampled from a pairwise-independent collection of permutations (we refer the reader to Sect. 2 for the relevant definitions). The secret key consists of the trapdoor for inverting f (we require that π is efficiently invertible), and the encryption of a message m is defined as $\mathsf{Enc}_{pk}(m) = f(\pi(m))$ (decryption is naturally defined by first inverting f and then inverting π).

The proof of security consists of two steps. First, the security of the collection of lossy trapdoor functions allows one to replace the injective function f with a lossy function $\widetilde{f}$ (where lossy means that the size of $\widetilde{f}$'s image is significantly smaller than the size of its domain). Then, the crooked leftover hash lemma of Dodis and Smith [11] states that for any plaintext distribution M that has a certain amount of min-entropy, for a uniformly and independently chosen pairwise-independent permutation π it holds that the distributions $\widetilde{f}(\pi(M))$ and $\widetilde{f}(U)$ are statistically close (even given $\widetilde{f}$ and π), where U is the uniform distribution over plaintexts. That is, essentially no information on the plaintext is revealed.

When considering adversaries that can choose the plaintext distribution M after receiving the description of π, the scheme of Boldyreva et al. can still be proved secure by simple modifications to the above-described proof (specifically, applying the crooked leftover hash lemma to each plaintext distribution that the adversary may choose, and then applying a union bound over all such distributions). A close look into the parameters of the modified proof shows that the resulting scheme is secure as long as the adversary chooses a plaintext distribution from a set of size roughly $2^{O(\lambda)}$ such distributions. Recall, however, that we would like to offer adaptive security for any set of

[6] In contrast, due to requiring key-independent plaintext distributions, Bellare et al. [3] and Boldyreva et al. [5] allow chosen-ciphertext adversaries to query the decryption oracle *only after* they have queried the encryption oracle.

$2^{p(\lambda)}$ plaintext distributions, where $p(\lambda)$ may be any predetermined polynomial in the security parameter.

The main idea underlying our basic construction is to sample the permutation π from a collection of highly independent permutations. We prove that this modification results in a scheme that is secure according to our new notion of security by proving a *High-Moment* crooked leftover hash lemma for collections of permutations. Informally, we prove that for any lossy function $\widetilde{f}$, and for any set $\mathcal{X}$ of sources with a certain amount of min-entropy, with an overwhelming probability over the choice of a permutation π from a t-wise almost-independent collection of permutations (where t depends only logarithmically on the size of $\mathcal{X}$), for *every* $M \in \mathcal{X}$ it holds that $\widetilde{f}(\pi(M))$ and $\widetilde{f}(U)$ are statistically close. In particular, in such a setting the specific choice of $M \in \mathcal{X}$ can adaptively depend on the permutation π, and still the statistical distance is negligible.

As already noted, a high-moment generalization of the (standard) leftover hash lemma was given by Trevisan and Vadhan [22] and Dodis [10]. In addition, an analogous generalization of the crooked leftover hash lemma for collections of *functions* was implicitly given in the work of Kiltz et al. [17, Proof of Theorem 2]. Their generalization, however, does not seem to admit a direct translation to collections of *permutations*. A different high-moment generalization of the crooked leftover hash lemma was proved by Fuller et al. [14] for the purpose of extracting randomness from a small number of possibly correlated sources. This generalization does not allow seed-dependent sources, and therefore allows only non-adaptive adversaries.

The advantage of our *high-moment* generalization As shown by Trevisan and Vadhan [22], the main advantage in using high-moment variants of the leftover hash lemma over using the basic leftover hash lemma is the exponential improvement in the dependency of the required min-entropy on the size of the set of sources. Specifically, for obtaining security with respect to any set of 2^p plaintext distributions, in our proof of security we need to apply the (either basic or generalized) crooked leftover hash lemma together with a union bound over all 2^p distributions. For enabling a union bound over a set of 2^p distributions, the crooked leftover hash lemma would require all plaintext distributions to have min-entropy that is logarithmic in 2^p, whereas our high-moment generalization requires min-entropy that is *doubly-logarithmic* in 2^p. In both cases, the required min-entropy is also linear in $\log |\mathsf{Im}(f)|$ where $\tilde{f}$ is the lossy function that is used by the encryption scheme, in $\log T$ where T is the number of blocks when considering block-sources, and in $\log(1/\epsilon)$ where ϵ is the statistical security parameter.

One the one hand, this exponential improvement indeed comes at the cost of increasing the length of the "public" parameters (which, in our setting, correspond to the public key of the scheme). On the other hand, however, this exponential improvement enables us to guarantee security with respect to any set of 2^p distributions, where $p = p(\lambda, n(\lambda))$ may be *any predetermined polynomial* in the security parameter $\lambda \in \mathbb{N}$ and the plaintext length $n = n(\lambda)$, whereas the basic crooked leftover hash lemma would enable us to consider at most 2^n distributions (in fact, even less when taking constants into account as well as the security parameter). This means, for example, that our scheme can be set up to guarantee security against all plaintext distributions that can be sampled by circuits of size n^2, but using the basic crooked leftover hash lemma one would obtain security

only against circuits of size less than n (i.e., less than the length n of the plaintext that they output).

In addition, even when focusing on rather small sets of distributions, consider the case of dealing with $2^{n/2}$ distributions, where $|\mathsf{Im}(f)| = n^{\epsilon}$ (as provided by known constructions of lossy trapdoor functions). For these parameters, our approach requires all plaintext distributions to have min-entropy roughly n^{ϵ}, whereas the basic crooked leftover hash lemma would require min-entropy that is linear in n.

Chosen-ciphertext security in the standard model While in the setting of chosen-plaintext security our construction is a natural generalization of that of Boldyreva et al. [5] (given our high-moment generalization of the crooked leftover hash), this is not the case in the setting of chosen-ciphertext security. In this setting, the CCA-secure scheme of Boldyreva et al. relies more strongly on the assumption that the challenge plaintext distribution is independent of the public key of the scheme (not just in the context of the crooked leftover hash lemma as above)—an assumption that we do not make. Nevertheless, we show that some of the ideas underlying their approach can still be utilized to construct a scheme that is secure according to our notion of security.

The scheme of Boldyreva et al. follows the "all-but-one" simulation paradigm of Peikert and Waters [19] using all-but-one lossy trapdoor functions. These are tag-based functions, where one of the tags corresponds to a lossy function, and all other tags correspond to injective functions. As in the work of Peikert and Waters [19], the approach of Boldyreva et al. makes sure that the challenge plaintext corresponds to a lossy tag (and thus the challenge ciphertext reveals no information), while all other plaintexts correspond to injective tags (and a suitable simulator is able to properly simulate the decryption oracle). When dealing with a deterministic encryption algorithm, note that tags must be derived deterministically from the plaintext and the public key. The approach of Boldyreva et al. is based on first sampling the challenge plaintext m^*, and only then generating a public key for which m^* corresponds to a lossy tag, but all other plaintexts correspond to injective tags.

This approach fails in our setting, where adversaries specify the distribution of the challenge plaintext in an adaptive manner as a function of the public key. Thus, in our setting we must be able to generate a public key before the challenge plaintext is known. We note that a somewhat similar issue arises in the setting of identity-based encryption (IBE): "selective security" considers adversaries that specify the challenge identity in advance, whereas "full security" considers adversaries that can adaptively choose the challenge identity. One simple solution that was proposed in the IBE setting is to a-priori guess the challenge identity, and this solution naturally extends to our setting by guessing the tag corresponds to the challenge plaintext. This, however, requires sub-exponential hardness assumptions, which we aim to avoid.

Our approach is based on the one of Boneh and Boyen [1] (and on its refinement by Cash et al. [9] for converting a large class of selectively secure IBE schemes to fully secure ones,[7] combined with the idea of $\mathcal{R}$-lossiness due to Boyle et al. [8]. Specifically, we derive tags from plaintexts using an admissible hash function [1,9], and instead of

[7] We note that the work of Cash et al. [9] is based on ideas introduced by Boneh and Boyen [1] and Waters [24].

using all-but-one lossy trapdoor functions, we introduce the notion of $\mathcal{R}$-lossy trapdoor functions (which we generically construct based on lossy trapdoor functions).[8] This is a generalization of the notion of all-but-one lossy trapdoor functions, where the set of tags is partitioned into lossy tags and injective tags according to the relation $\mathcal{R}$. (In particular, there may be more than one lossy tag.) Combined with an admissible hash function, we are able to ensure that even with an adaptive adversary, with some non-negligible probability, the challenge plaintext corresponds to a lossy tag (and thus the challenge ciphertext reveals no information), while all other plaintexts correspond to injective tags (and a suitable simulator is able to properly simulate the decryption oracle). We show that such a guarantee enables us to prove the security of our scheme with respect to adaptive adversaries.

1.4. *Paper Organization*

The remainder of this paper is organized as follows. In Sect. 2 we introduce several basic definitions and tools. In Sect. 3 we formally define our new notions capturing adaptive security for deterministic public-key encryption. In Sect. 4 we present our high-moment generalization of the crooked leftover hash lemma, which we then use in Sect. 5 for constructing our basic adaptively secure scheme. In Sect. 6 we introduce and realize the notion of $\mathcal{R}$-lossy trapdoor functions, which we then use Sect. 7 for extending our basic construction to the setting of chosen-ciphertext attacks. Finally, in Sect. 8 we present generic constructions satisfying our notions of security in the random oracle model.

2. Preliminaries

For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$, and by U_n the uniform distribution over the set $\{0,1\}^n$. For a random variable X we denote by $x \leftarrow X$ the process of sampling a value x according to the distribution of X and by $\mathbb{E}[X]$ the expectation of the random variable X. Similarly, for a finite set S we denote by $x \leftarrow S$ the process of sampling a value x according to the uniform distribution over S. We denote by $\boldsymbol{X} = (X_1, \ldots, X_T)$ a joint distribution of T random variables, and by $\boldsymbol{x} = (x_1, \ldots, x_T)$ a sample drawn from $\boldsymbol{X}$. For two bit-strings x and y we denote by $x \| y$ their concatenation. A nonnegative function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* if it vanishes faster than any inverse polynomial.

In this paper we consider the uniform adversarial model (i.e., consider uniform probabilistic polynomial-time adversaries). We note that all of our results also apply to the nonuniform adversarial model (under nonuniform complexity assumptions).

The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. A *k-source* is a random variable X with $\mathbf{H}_\infty(X) \geq k$. A (T, k)-*source* is a random variable $\boldsymbol{X} = (X_1, \ldots, X_T)$ where each X_i is a k-source for every $i \in [T]$. A (T, k)-*block-source* is a random variable $\boldsymbol{X} = (X_1, \ldots, X_T)$ where for every $i \in [T]$ and $x_1, \ldots, x_{i-1}$ it holds that $\mathbf{H}_\infty(X_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) \geq k$.

[8] Boyle et al. [8] introduced the notion of $\mathcal{R}$-lossy public-key encryption, which can be viewed as a randomized variant of our notion of $\mathcal{R}$-lossy trapdoor functions.

The following standard lemma states that conditioning on random variable that obtains at most 2^v values can reduce the min-entropy of any other random variable by essentially at most v.

Lemma 2.1. (cf. [23, Lemma 6.30]) *Let (Z, X) be any two jointly distributed random variables such that $|\mathrm{Supp}(Z)| \leq 2^v$. Then, for any $\epsilon > 0$ it holds that*

$$\Pr_{z \leftarrow Z}\left[\mathbf{H}_\infty(X|Z=z) \geq \mathbf{H}_\infty(X) - v - \log(1/\epsilon)\right] \geq 1 - \epsilon.$$

The *statistical distance* between two random variables X and Y over a finite domain Ω is $\mathbf{SD}(X, Y) = \frac{1}{2}\sum_{\omega\in\Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. Two random variables X and Y are δ-*close* if $\mathbf{SD}(X, Y) \leq \delta$. Two distribution ensembles $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are *statistically indistinguishable* if it holds that $\mathbf{SD}(X_\lambda, Y_\lambda)$ is negligible in λ. They are *computationally indistinguishable* if for every probabilistic polynomial-time algorithm $\mathcal{A}$ it holds that

$$\left|\Pr_{x \leftarrow X_\lambda}\left[\mathcal{A}(1^\lambda, x) = 1\right] - \Pr_{y \leftarrow Y_\lambda}\left[\mathcal{A}(1^\lambda, y) = 1\right]\right|$$

is negligible in λ.

2.1. *t-Wise δ-Dependent Permutations*

A collection Π of permutations over $\{0,1\}^n$ is *t-wise δ-dependent* if for any distinct $x_1, \ldots, x_t \in \{0,1\}^n$ the distribution $(\pi(x_1), \ldots, \pi(x_t))$ where π is sampled uniformly from Π is δ-close in statistical distance to the distribution $(\pi^*(x_1), \ldots, \pi^*(x_t))$ where π^* is a truly random permutation. For our construction in the standard model we rely on an explicit construction of such a collection due to Kaplan et al. [16] that enjoys an asymptotically optimal description length (although we note that in fact any other construction can be used):

Theorem 2.2. [16] *For any integers n and $t \leq 2^n$, and for any $0 < \delta < 1$, there exists an explicit t-wise δ-dependent collection Π of permutations over $\{0,1\}^n$ where each permutation $\pi \in \Pi$ can be described using $O(nt + \log(1/\delta))$ bits, and is computable and invertible in time polynomial in n, t and $\log(1/\delta)$.*

For our purposes it would be quite convenient to rely on a t-wise δ-dependent collection of permutations Π in which the marginal distribution $\pi(x)$ is *perfectly uniform* (as opposed to just δ-close to uniform) for any $x \in \{0,1\}^n$ over the choice of $\pi \leftarrow \Pi$. Although this property is not essentially satisfied by any t-wise δ-dependent collection of permutations, it is straightforward to generically transform any such collection to having this additional property: Given a collection Π of permutations over $\{0,1\}^n$, consider the collection $\Pi' = \{\pi_y : (y, \pi) \in \{0,1\}^n \times \Pi\}$ defined as $\pi_y(x) = \pi(x) \oplus y$. It is easy to verify that: (1) if Π is t-wise δ-dependent collection then so is Π', and (2) for any $x \in \{0,1\}^n$ it holds that $\pi_y(x)$ is perfectly uniform over the choice of $\pi_y \leftarrow \Pi'$. Moreover, this simple generic transformation does not affect (in an asymptotic manner) the parameters stated in Theorem 2.2. From this point on in the paper, whenever we refer

to t-wise δ-dependent collections of permutations, we refer to collections that have this additional property.

2.2. *Admissible Hash Functions*

The concept of an *admissible hash function* was first defined by Boneh and Boyen [1] to convert a large class of selectively secure identity-based encryption scheme into a fully secure ones. In this paper we use such hash functions in a somewhat similar way as part of our construction of a CCA-secure deterministic public-key encryption scheme. The main idea of an admissible hash function is that it allows the reduction in the proof of security to secretly partition the message space into two subsets, which we will label as "lossy tags" and "injective tags," such that there is a noticeable probability that all of the messages in the adversary's decryption queries will correspond to injective tags, but the challenge ciphertext will correspond to a lossy tag. This is useful if the simulator can efficiently answer decryption queries with injective tags, while a challenge ciphertext with a lossy tag reveals essentially no information on the encrypted message. Our exposition and definition of admissible hash function follows that of Cash et al. [9].

For $K \in \{0, 1, \perp\}^{v(\lambda)}$, we define the "partitioning" function $P_K : \{0, 1\}^{v(\lambda)} \to \{\texttt{Lossy}, \texttt{Inj}\}$ which partitions the space $\{0, 1\}^{v(\lambda)}$ of tags in the following way:

$$P_K(y) := \begin{cases} \texttt{Lossy} & \text{if } \forall\, i \in \{1, \ldots, v(\lambda)\} \;:\; K_i = y_i \text{ or } K_i = \perp \\ \texttt{Inj} & \text{otherwise} \end{cases}$$

For any $u = u(\lambda) < v(\lambda)$, we let $\mathcal{K}_{u,\lambda}$ denote the uniform distribution over $\{0, 1, \perp\}^{v(\lambda)}$ *conditioned on* exactly u positions having $\perp$ values. (Note, if K is chosen from $\mathcal{K}_{u,\lambda}$, then the map $P_K(\cdot)$ defines exactly 2^u values as $\texttt{Lossy}$.) We would like to pick a distribution $\mathcal{K}_{u,\lambda}$ for choosing K so that, there is a noticeable probability for every set of tags $y_0, \ldots, y_q$, of y_0 being classified as "lossy" and all other tags "injective." Unfortunately, this cannot happen if we allow all tags. Instead, we will need to rely on a special hash function the maps messages x to tags y.

Definition 2.3. (*Admissible hash functions* [1,9]) Let $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ be a hash-function ensemble, where each $h \in \mathcal{H}_\lambda$ is a polynomial-time computable function $h : \{0, 1\}^{n(\lambda)} \to \{0, 1\}^{v(\lambda)}$. We say that $\mathcal{H}$ is an *admissible hash-function ensemble* if for every $\lambda \in \mathbb{N}$ and $h \in \mathcal{H}_\lambda$ there exists a efficiently recognizable set $\mathsf{Unlikely}_h \subseteq \bigcup_{q \in \mathbb{N}} \left(\{0, 1\}^{n(\lambda)}\right)^q$ of string-tuples such that the following two properties hold:

- For every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\nu(\lambda)$ satisfying

$$\Pr[(x_0, \ldots, x_q) \in \mathsf{Unlikely}_h] \leq \nu(\lambda),$$

  where $h \leftarrow \mathcal{H}_\lambda$ and $(x_0, \ldots, x_q) \leftarrow \mathcal{A}(1^\lambda, h)$.
- For every polynomial $q = q(\lambda)$ there is a polynomial $\Delta = \Delta(\lambda)$ and an efficiently computable $u = u(\lambda)$ such that, for every $h \in \mathcal{H}_\lambda$ and $(x_0, \ldots, x_q) \notin \mathsf{Unlikely}_h$ with $x_0 \notin \{x_1, \ldots, x_q\}$ we have:

$$\Pr_{K \leftarrow \mathcal{K}_{u,\lambda}} \left[P_K(h(x_0)) = \texttt{Lossy} \wedge P_K(h(x_1)) = \cdots = P_K(h(x_q)) = \texttt{Inj} \right] \geq \frac{1}{\Delta(\lambda)}.$$

The work of Boneh and Boyen [1] shows how to construct admissible hash functions from collision-resistant hash functions.

2.3. *Lossy Trapdoor Functions*

A collection of lossy trapdoor functions [19] consists of two families of functions. Functions in one family are injective and can be efficiently inverted using a trapdoor. Functions in the other family are "lossy," which means that the size of their image is significantly smaller than the size of their domain. The only security requirement is that a description of a randomly chosen function from the family of injective functions is computationally indistinguishable from a description of a randomly chosen function from the family of lossy functions.

Definition 2.4. (*Lossy trapdoor functions* [13,19]) Let $n : \mathbb{N} \to \mathbb{N}$ and $\ell : \mathbb{N} \to \mathbb{N}$ be nonnegative functions, and for any $\lambda \in \mathbb{N}$ let $n = n(\lambda)$ and $\ell = \ell(\lambda)$. A *collection of* (n, ℓ)*-lossy trapdoor functions* is a 4-tuple of probabilistic polynomial-time algorithms $(\mathsf{Gen}_0, \mathsf{Gen}_1, \mathsf{F}, \mathsf{F}^{-1})$ such that:

1. **Sampling a lossy function** $\mathsf{Gen}_0(1^\lambda)$ outputs a function index $\sigma \in \{0,1\}^*$.
2. **Sampling an injective function** $\mathsf{Gen}_1(1^\lambda)$ outputs a pair $(\sigma, \tau) \in \{0,1\}^* \times \{0,1\}^*$, where σ is a function index and τ is a trapdoor.
3. **Evaluation** Let $n = n(\lambda)$ and $\ell = \ell(\lambda)$. Then, for every function index σ produced by either Gen_0 or Gen_1, algorithm $\mathsf{F}(\sigma, \cdot)$ computes a function $f_\sigma : \{0,1\}^n \to \{0,1\}^*$ with one of the two following properties:
   - Lossy: If σ is produced by Gen_0, then the image of f_σ has size at most $2^{n-\ell}$.
   - Injective: If σ is produced by Gen_1, then the function f_σ is injective.
4. **Inversion of injective functions** For every pair (σ, τ) produced by Gen_1 and every $x \in \{0,1\}^n$, we have $\mathsf{F}^{-1}(\tau, \mathsf{F}(\sigma, x)) = x$.
5. **Security** The two ensembles $\left\{\sigma : \sigma \leftarrow \mathsf{Gen}_0(1^\lambda)\right\}_{\lambda \in \mathbb{N}}$ and $\left\{\sigma : (\sigma, \tau) \leftarrow \mathsf{Gen}_1(1^\lambda)\right\}_{\lambda \in \mathbb{N}}$ are computationally indistinguishable.

Constructions of lossy trapdoor functions were proposed based on a wide variety of number-theoretic assumptions and for a large range of parameters (see, for example, [13, 19] and the references therein). In particular, in terms of parameters, several constructions are known to offer $\ell = n - n^\epsilon$ for any fixed constant $0 < \epsilon < 1$ with $n = \mathrm{poly}(\lambda)$.

2.4. *Deterministic Public-Key Encryption*

A deterministic public-key encryption scheme 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 as input the security parameter 1^λ and outputs a key pair (sk, pk) consisting of a secret key sk and a public key pk.
- The encryption algorithm Enc is a *deterministic* algorithm that takes as input a public key pk and a message $m \in \{0,1\}^{n(\lambda)}$, 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 $m \in \{0,1\}^{n(\lambda)} \cup \{\perp\}$.

3. Formalizing Adaptive Security for Deterministic Public-Key Encryption

In this section we present a framework for modeling the security of deterministic public-key encryption schemes in an *adaptive* setting. As discussed in Sect. 1.3, we consider adversaries that *adaptively* choose plaintext distributions *after* seeing the public key of the scheme, in an *interactive* manner. The only restriction we make is that the *number* of plaintext distributions from which each adversary is allowed to choose is upper bounded by $2^{p(\lambda)}$, where $p(\lambda)$ can be any a priori given polynomial in the security parameter λ.

The security definitions that follow are parameterized by three parameters:

- $p = p(\lambda)$ denoting the 2^p bound on the number of allowed plaintext distributions.
- $T = T(\lambda)$ denoting the number of blocks in each plaintext distribution.
- $k = k(\lambda)$ denoting the min-entropy requirement.

Additionally, they are implicitly parameterized by bit-length $n = n(\lambda)$ of plaintexts. We begin by defining the "real-or-random" encryption oracle which we use to formalize security.

Definition 3.1. (*Real-or-random encryption oracle*) The real-or-random oracle RoR takes as input triplets of the form $(\mathsf{mode}, pk, \boldsymbol{M})$, where $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, pk is a public key, and $\boldsymbol{M} = (M_1, \ldots, M_T)$ is a circuit representing a joint distribution over T messages. If $\mathsf{mode} = \mathsf{real}$ then the oracle samples $(m_1, \ldots, m_T) \leftarrow \boldsymbol{M}$, and if $\mathsf{mode} = \mathsf{rand}$ then the oracle samples $(m_1, \ldots, m_T) \leftarrow U^T$ where U is the uniform distribution over the appropriate message space. It then outputs the vector of ciphertexts $(\mathsf{Enc}_{pk}(m_1), \ldots, \mathsf{Enc}_{pk}(m_T))$.

Following [3,5] we consider two classes of adversarially chosen message distributions $\boldsymbol{M} = (M_1, \ldots, M_T)$: The class of (T, k)-sources, where each M_i is assumed to be a k-source, and the more restrictive class of (T, k)-block-sources, where each M_i is assumed to be a k-source even given $M_1, \ldots, M_{i-1}$. (See Sect. 2 for formal definitions.) Our constructions in the random oracle model are secure with respect to (T, k)-sources, and our constructions in the standard model are secure with respect to (T, k)-block-sources. This gap was recently shown by Wichs [26] to be inherent to our techniques, and in fact to all the techniques that were so far used for designing deterministic public-key encryption schemes without random oracles [2,4,5,7,14,18,25]. Specifically, Wichs showed that no deterministic public-key encryption scheme can be proven secure for all (T, k)-sources using a black-box reduction to a "falsifiable" hardness assumption. (We refer the reader to [26] for more details on his notion of falsifiability.)

3.1. *Chosen-Plaintext Security*

The following two definitions capture the class of adversaries and security game that we consider in this paper.

Definition 3.2. (2^p*-bounded* (T, k)*-source adversary*) Let $\mathcal{A}$ be a probabilistic polynomial-time algorithm that is given as input a pair $(1^\lambda, pk)$ and oracle access to $\mathsf{RoR}(\mathsf{mode}, pk, \cdot)$ for some $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$. Then, $\mathcal{A}$ is a 2^p*-bounded* (T, k)*-source adversary* if for every $\lambda \in \mathbb{N}$ there exists a set $\mathcal{X} = \mathcal{X}_\lambda$ of polynomial-time samplable (T, k)-sources such that:

1. $|\mathcal{X}| \leq 2^p$.
2. For each of $\mathcal{A}$'s RoR queries $\boldsymbol{M}$ it holds that:
   - $\boldsymbol{M} \in \mathcal{X}$.
   - For all $(m_1, \ldots, m_T)$ in the support of $\boldsymbol{M}$ and for all distinct $i, j \in [T]$ it holds that $m_i \neq m_j$.

In addition, $\mathcal{A}$ is a *block-source* adversary if $\mathcal{X}$ is a set of (T, k)-block-sources.

Definition 3.3. (*Adaptive chosen-distribution attacks (ACD-CPA)*) A deterministic public-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is (p, T, k)*-ACD-CPA-secure* (resp. *block-wise* (p, T, k)*-ACD-CPA-secure*) if for any probabilistic polynomial-time 2^p-bounded (T, k)-source (resp. block-source) adversary $\mathcal{A}$, there exists a negligible function $\nu(k)$ such that

$$\mathbf{Adv}^{\text{ACD-CPA}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\Pi,\mathcal{A}}(\lambda) = 1\right]\right| \leq \nu(\lambda),$$

where for each $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{\mathsf{mode}}_{\Pi,\mathcal{A}}(\lambda)$ is defined as follows:

1. $(sk, pk) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. $b \leftarrow \mathcal{A}^{\mathsf{RoR}(\mathsf{mode}, pk, \cdot)}(1^\lambda, pk)$.
3. Output b.

In addition, such a scheme is (p, T, k)-ACD1-CPA-*secure* (resp. *block-wise* (p, T, k)-*ACD1-CPA-secure*) if the above holds for any probabilistic polynomial-time 2^p-bounded (T, k)-source (resp. block-source) adversary $\mathcal{A}$ that queries the RoR oracle at most once.

Our adaptive notion of security enables an immediate reduction in "multi-shot" adversaries to "single-shot" ones, as in the case of randomized public-key encryption. The following theorem follows via a standard hybrid argument.

Theorem 3.4. (Equivalence of ACD-CPA-security and ACD-CPA-security) *For any p, T, and k, a deterministic public-key encryption scheme* Π *is* (p, T, k)*-ACD-CPA-secure (resp. block-wise* (p, T, k)*-ACD-CPA-secure) if and only if it is* (p, T, k)*-ACD1-CPA-secure (resp. block-wise* (p, T, k)*-ACD1-CPA-secure).*

3.2. *Chosen-Ciphertext Security*

We now extend our notion of security to capture chosen-ciphertext adversaries. We note that, unlike Bellare et al. [3] and Boldyreva at el. [5] , we allow the adversary to adaptively interact with the encryption and decryption oracles *in any order*.

Definition 3.5. (*2^p-bounded (T, k)-source chosen-ciphertext adversary*) Let $\mathcal{A}$ be an algorithm that is given as input a pair $(1^\lambda, pk)$ and oracle access to two oracles: $\mathsf{RoR}(\mathsf{mode}, pk, \cdot)$ for some $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, and $\mathsf{Dec}(sk, \cdot)$. Then, $\mathcal{A}$ is a *2^p-bounded (T, k)-source chosen-ciphertext (resp. block-source) adversary* if:

1. $\mathcal{A}$ is a 2^p-bounded (T, k)-source (resp. block-source) adversary.
2. $\mathcal{A}$ never queries $\mathsf{Dec}(sk, \cdot)$ with any ciphertext c that was part of a previous output by the RoR oracle.

Definition 3.6. (*Adaptive chosen-distribution chosen-ciphertext attacks (ACD-CCA)*) A deterministic public-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is (p, T, k)-*ACD-CCA-secure* (resp. *block-wise* (p, T, k)-*ACD-CCA-secure*) if for every probabilistic polynomial-time 2^p-bounded (T, k)-source (resp. block-source) chosen-ciphertext adversary $\mathcal{A}$, there exists a negligible function $\nu(k)$ such that

$$\mathbf{Adv}^{\text{ACD-CCA}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{realCCA}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{randCCA}}_{\Pi,\mathcal{A}}(\lambda) = 1\right]\right| \leq \nu(\lambda),$$

where for each $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{\mathsf{modeCCA}}_{\Pi,\mathcal{A}}(\lambda)$ is defined as follows:

1. $(sk, pk) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. $b \leftarrow \mathcal{A}^{\mathsf{RoR}(\mathsf{mode}, pk, \cdot), \mathsf{Dec}(sk, \cdot)}(1^\lambda, pk)$.
3. Output b.

In addition, such a scheme is (p, T, k)-*ACD1-CCA-secure* (resp. *block-wise* (p, T, k)-*ACD1-CCA-secure*) if the above holds for any probabilistic polynomial-time 2^p-bounded (T, k)-source (resp. block-source) adversary $\mathcal{A}$ that queries the RoR oracle at most once. (Note that $\mathcal{A}$ may still query the decryption oracle many times.)

As in the case of chosen-plaintext security, a standard hybrid argument immediately reduce "multi-shot" adversaries to "single-shot" ones by exploiting the adaptive flavor of our security notion.

Theorem 3.7. (Equivalence of ACD-CCA security and ACD1-CCA-security) *For any p and k, a deterministic public-key encryption scheme Π is (p, T, k)-ACD-CCA-secure (resp. block-wise (p, T, k)-ACD-CCA-secure) if and only if it is (p, T, k)-ACD1-CCA-secure (resp. block-wise (p, T, k)-ACD1-CCA-secure).*

4. Deterministic Extraction via a High-Moment Crooked Leftover Hash Lemma

In this section we present a high-moment generalization of the crooked leftover hash lemma of Dodis and Smith [11]. Informally, the crooked leftover hash lemma states that for every lossy function f (where lossy means that the size of f's image is significantly smaller than the size of its domain), and for every random source X that has a certain amount of min-entropy, for a uniformly and independently chosen pairwise-independent permutation π it holds that the distributions $f(\pi(X))$ and $f(U)$ are statistically close (even given f and π), where U is the uniform distribution over the domain of f. In this paper, as discussed in Sect. 1.3, we consider a setting in which the distribution X may be adaptively chosen depending on π. In this setting, in general, the crooked leftover hash lemma no longer holds. Nevertheless, we show that a natural high-moment generalization of the crooked leftover hash lemma does hold in such a setting by applying a union bound over all possible choices. Our approach is based on that of Trevisan and Vadhan [22] and Dodis [10], who presented a similar generalization to the *standard* leftover hash lemma.

Specifically, we prove that for every lossy function f, and for every set $\mathcal{X}$ of random sources with a certain amount of min-entropy, with an overwhelming probability over the choice of a permutation π from a t-wise almost-independent collection of permutations (where t depends only logarithmically on the size of $\mathcal{X}$), for *every* $X \in \mathcal{X}$ it holds that $f(\pi(X))$ and $f(U)$ are statistically close. In particular, in such a setting the specific choice of $X \in \mathcal{X}$ can adaptively depend on the permutation π, and still the statistical distance is negligible.

We note that throughout this section, whenever our expressions for lower bounding the min-entropy k contain an additive constant factor (denoted by using the $\Theta(1)$ notation), this factor is a universal constant (which may differ from claim to claim).

4.1. *A High-Moment Crooked Leftover Hash Lemma*

Given a function f, we begin by considering a specific element y in the image of f and prove that for most permutations π the distributions $f(\pi(X))$ and $f(U)$ "hit" y with essentially the same probability.

As discussed in Sect. 2.1, recall that we find it convenient to rely (without loss of generality) on t-wise δ-dependent collections of permutations Π in which the marginal distribution $\pi(x)$ is *perfectly uniform* (as opposed to just δ-close to uniform) for any $x \in \{0,1\}^n$ over the choice of $\pi \leftarrow \Pi$.

Lemma 4.1. *Let $f : \{0,1\}^n \to \{0,1\}^{n'}$, and let Π be a t-wise δ-dependent collection of permutations over $\{0,1\}^n$, where $t \geq 8$ is even and $\delta \leq 2^{-nt}$. Then, for every $y \in \mathrm{Im}(f)$, every k-source X over $\{0,1\}^n$, and every $0 < \epsilon < 1$ such that*

$$k \geq \log|\mathrm{Im}(f)| + 2\log(1/\epsilon) + 2\log t + \Theta(1),$$

it holds that

$$\Pr_{\pi \leftarrow \Pi}\left[\left|\Pr_{x \leftarrow X}[f(\pi(x)) = y] - \frac{\left|f^{-1}(y)\right|}{2^n}\right| > \epsilon \cdot \max\left\{\frac{\left|f^{-1}(y)\right|}{2^n}, \frac{1}{|\mathrm{Im}(f)|}\right\}\right] \leq 2^{-t}. \tag{4.1}$$

Proof. For every $x \in \{0,1\}^n$ let $p_x = \Pr[X = x]$, and let $\mathbb{I}_{f(\pi(x))=y}$ be the indicator of the event in which $f(\pi(x)) = y$ (note that f and y are fixed). In addition, let $q_x = p_x \cdot \mathbb{I}_{f(\pi(x))=y}$ and $q = \sum_{x \in \{0,1\}^n} q_x = \Pr_{x \leftarrow X}[f(\pi(x)) = y]$.

Since X has min-entropy at least k, if for every $x \in \{0,1\}^n$ we let $Q_x = 2^k \cdot q_x = 2^k \cdot p_x \cdot \mathbb{I}_{f(\pi(x))=y}$, it holds that $Q_x \in [0,1]$. Let $Q = 2^k \cdot \sum_{x \in \{0,1\}^n} q_x$ and $\mu = \mathbb{E}[Q]$ (where the expectation is taken over the choice of π). For every $\pi \in \Pi$ it holds that

$$Q = 2^k \cdot \Pr_{x \leftarrow X}[f(\pi(x)) = y] \text{ and } \mathbb{E}[Q] = \mu = 2^k \cdot \frac{\left|f^{-1}(y)\right|}{2^n}.$$

Next, we define $\mu' \stackrel{\mathsf{def}}{=} \max\{\mu, 2^{k-\log|\mathrm{Im}(f)|}\}$. To bound the quantity in Eq. (4.1) (multiplying all terms by 2^k) we proceed as follows,

$$\begin{aligned}
&\Pr_{\pi \leftarrow \Pi}\left[\left|\Pr_{x \leftarrow X}[f(\pi(x)) = y] - \frac{\left|f^{-1}(y)\right|}{2^n}\right| > \epsilon \cdot \max\left\{\frac{\left|f^{-1}(y)\right|}{2^n}, \frac{1}{|\mathrm{Im}(f)|}\right\}\right] \\
&= \Pr_{\pi \leftarrow \Pi}\left[|Q - \mu| > \epsilon\mu'\right] \\
&= \Pr_{\pi \leftarrow \Pi}\left[(Q-\mu)^t > (\epsilon\mu')^t\right] \\
&\leq \frac{\mathbb{E}_{\pi \leftarrow \Pi}\left[(Q-\mu)^t\right]}{(\epsilon\mu')^t},
\end{aligned}$$

where the above inequalities use Markov's inequality and the fact that t is even. The following claim is proved in Sect. 4.3: □

Claim 4.2. *For Q and μ defined above it holds that*

$$\mathbb{E}_{\pi \leftarrow \Pi}\left[(Q-\mu)^t\right] \leq C_t \cdot (t\mu + t^2)^{t/2} + \delta \cdot 2^{nt},$$

for some small constant C_t (in fact, $C_t < 5$ for $t \geq 8$).

Claim 4.2 guarantees that

$$\begin{aligned}
\Pr_{\pi \leftarrow \Pi}\left[|Q - \mu| > \epsilon\mu'\right] &\leq C_t \cdot \left(\frac{t\mu + t^2}{\epsilon^2\mu'^2}\right)^{t/2} + \delta \cdot \left(\frac{2^n}{\epsilon\mu'}\right)^t \\
&\leq 2C_t \cdot \left(\frac{t\mu + t^2}{\epsilon^2\mu'^2}\right)^{t/2},
\end{aligned} \tag{4.2}$$

where the inequality derived in Eq. (4.2) uses the fact that $\delta \leq 2^{-nt}$ which implies that the dominant term is the first one. We now distinguish between two possible cases:

Case 1: $t \leq \mu$. In this case we have that

$$\Pr_{\pi \leftarrow \Pi}\left[|Q - \mu| > \epsilon\mu'\right] \leq 2C_t \cdot \left(\frac{2t\mu}{\epsilon^2\mu'^2}\right)^{t/2} \leq 2C_t \cdot \left(\frac{2t}{\epsilon^2\mu'}\right)^{t/2}.$$

Upon substituting for μ' and noting again that $\mu' \geq 2^{k-\log|\mathrm{Im}(f)|}$, we get:

$$\begin{aligned}
&\Pr_{\pi \leftarrow \Pi}\left[\left|\Pr_{x \leftarrow X}[f(\pi(x)) = y] - \frac{\left|f^{-1}(y)\right|}{2^n}\right| > \epsilon \cdot \max\left\{\frac{\left|f^{-1}(y)\right|}{2^n}, \frac{1}{\mathrm{Im}(f)|}\right\}\right] \\
&\quad \leq 2C_t \cdot \left(\frac{2t}{\epsilon^2 \cdot 2^{k-\log|\mathrm{Im}(f)|}}\right)^{t/2} \\
&\quad \leq 2C_t \cdot 2^{t/2\cdot(\log(2t)+2\log(1/\epsilon)+\log|\mathrm{Im}(f)|-k)} \\
&\quad \leq 2^{-t}.
\end{aligned}$$

Case 2: $t > \mu$. In this case we have that

$$\Pr_{\pi \leftarrow \Pi}\left[|Q - \mu| > \epsilon\mu'\right] \leq 2C_t \cdot \left(\frac{2t^2}{\epsilon^2\mu'^2}\right)^{t/2}.$$

Upon substituting for μ' and noting that $\mu' \geq 2^{k-\log|\mathrm{Im}(f)|}$, we get:

$$\begin{aligned}
&\Pr_{\pi \leftarrow \Pi}\left[\left|\Pr_{x \leftarrow X}[f(\pi(x)) = y] - \frac{\left|f^{-1}(y)\right|}{2^n}\right| > \epsilon \cdot \max\left\{\frac{\left|f^{-1}(y)\right|}{2^n}, \frac{1}{\mathrm{Im}(f)|}\right\}\right] \\
&\quad \leq 2C_t \cdot \left(\frac{2t^2}{\epsilon^2 \cdot 2^{2(k-\log|\mathrm{Im}(f)|)}}\right)^{t/2} \\
&\quad \leq 2C_t \cdot 2^{t/2\cdot\left(\log(2t^2)+2\log(1/\epsilon)+2\log|\mathrm{Im}(f)|-2k\right)} \\
&\quad \leq 2^{-t}.
\end{aligned}$$

□

The next lemma uses Lemma 4.1 to show that for most permutations π, not only that the distributions $f(\pi(X))$ and $f(U)$ are point-wise similar, but in fact they are statistically close.

Definition 4.3. A function $f : \{0,1\}^n \to \{0,1\}^{n'}$ is (n, ℓ)-*lossy* if $|\mathrm{Im}(f)| \leq 2^{n-\ell}$.

Lemma 4.4. *Let $f : \{0,1\}^n \to \{0,1\}^{n'}$ be (n, ℓ)-lossy, and let Π be a t-wise δ-dependent collection of permutations over $\{0,1\}^n$, where $t \geq 8$ is even and $\delta \leq 2^{-nt}$. Then, for every k-source X over $\{0,1\}^n$ and $0 < \epsilon < 1$ such that*

$$k \geq n - \ell + 2\log(1/\epsilon) + 2\log t + \Theta(1),$$

it holds that

$$\Pr_{\pi \leftarrow \Pi}[\mathbf{SD}\left(f(\pi(X)), f(U_n)\right) \leq \epsilon] \geq 1 - 2^{n-\ell-t},$$

where U_n is the uniform distribution over $\{0,1\}^n$.

Proof. From Lemma 4.1, for every k-source X there exists a set of permutations $\Pi_X \subseteq \Pi$ such that $\Pr_{\pi \leftarrow \Pi}[\pi \in \Pi_X] \geq 1 - 2^{-t} \cdot |\mathrm{Im}(f)| \geq 1 - 2^{n-\ell-t}$, and for every $\pi \in \Pi_X$ and $y \in \mathrm{Im}(f)$ it holds that

$$\left| \Pr_{x \leftarrow X}[f(\pi(x)) = y] - \frac{\left|f^{-1}(y)\right|}{2^n} \right| \leq \epsilon \cdot \max\left\{ \frac{\left|f^{-1}(y)\right|}{2^n}, \frac{1}{|\mathrm{Im}(f)|} \right\}.$$

The definition of statistical distance implies that for every $X \in \mathcal{X}$ and for every $\pi \in \Pi_X$

$$\begin{aligned}
\mathbf{SD}\left(f(\pi(X)), f(U_n)\right) &= \frac{1}{2} \sum_{y \in \mathrm{Im}(f)} |\Pr[f(\pi(X)) = y] - \Pr[f(U_n) = y]| \\
&= \frac{1}{2} \sum_{y \in \mathrm{Im}(f)} \left| \Pr_{x \leftarrow X}[f(\pi(x)) = y] - \frac{\left|f^{-1}(y)\right|}{2^n} \right| \\
&\leq \frac{1}{2} \sum_{y \in \mathrm{Im}(f)} \epsilon \cdot \frac{\left|f^{-1}(y)\right|}{2^n} + \frac{1}{2} \sum_{y \in \mathrm{Im}(f)} \frac{\epsilon}{|\mathrm{Im}(f)|} \qquad (4.3) \\
&\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon, \qquad (4.4)
\end{aligned}$$

where we use the fact that $\max(a,b) \leq a + b$ when $a, b \geq 0$ in Eq. (4.3) and the fact that $\sum_{y \in \mathrm{Im}(f)} |f^{-1}(y)| = 2^n$ in Eq. (4.4). □

4.2. *Generalization to Block-Sources*

We now extend Lemma 4.4 to block-sources by first deriving an *average-case* variant.

Lemma 4.5. *Let $f : \{0,1\}^n \to \{0,1\}^{n'}$ be (n,ℓ)-lossy, and let Π be a t-wise δ-dependent collection of permutations over $\{0,1\}^n$, where $t \geq 8$ is even and $\delta \leq 2^{-nt}$. Then, for every $0 < \epsilon < 1$ and for every jointly distributed random variables (X, Y) over $\{0,1\}^n \times \{0,1\}^m$ such that for every $y \in \{0,1\}^m$, $\mathbf{H}_\infty(X|Y = y) \geq k$ where*

$$k \geq n - \ell + 2\log(1/\epsilon) + 2\log t + \Theta(1),$$

it holds that

$$\Pr_{\pi \leftarrow \Pi}\left[\mathbf{SD}\Big(\left(f(\pi(X), Y)\right), \left(f(U_n), Y\right)\Big) \leq 2\epsilon\right] \geq 1 - \frac{2^{n-\ell-t}}{\epsilon},$$

where U_n is the uniform distribution over $\{0,1\}^n$.

Proof. For every permutation $\pi \in \Pi$ and for every $y \in \{0,1\}^m$, denote by $\mathsf{Bad}_\pi(y)$ the event in which

$$\mathbf{SD}\Big(\big(f(\pi(X|_{Y=y})),y\big),\big(f(U_n),y\big)\Big) > \epsilon.$$

As the distribution $X|_{Y=y}$ has min-entropy at least k for every $y \in \{0,1\}^m$, applying Lemma 4.4 with $|\mathcal{X}| = 1$, for every $y \in \{0,1\}^m$, we have that:

$$\Pr_{\pi \leftarrow \Pi}\big[\mathsf{Bad}_\pi(y)\big] < 2^{n-\ell-t}. \tag{4.5}$$

Applying Markov's inequality, it then following that for most permutations $\pi \in \Pi$ is holds that $\Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big] \le \epsilon$:

$$\begin{aligned}
\Pr_{\pi \leftarrow \Pi}\left[\Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big] > \epsilon\right] &\le \frac{\mathbb{E}_{\pi \leftarrow \Pi}\Big[\Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big]\Big]}{\epsilon} \\
&= \frac{1}{|\Pi|} \cdot \frac{\sum_{\pi \in \Pi} \cdot \Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big]}{\epsilon} \\
&= \frac{1}{|\Pi|} \cdot \frac{\sum_{\pi \in \Pi} \sum_{y \in \{0,1\}^m} \Pr[Y=y] \cdot \mathbb{I}_{\mathsf{Bad}_\pi(y)}}{\epsilon} \\
&= \frac{\Pr_{y \leftarrow Y}\big[\mathbb{E}_{\pi \leftarrow \Pi}\big[\mathsf{Bad}_\pi(y)\big]\big]}{\epsilon} \\
&\le \frac{2^{n-\ell-t}}{\epsilon}. \qquad \text{(from Eq.(4.5))}
\end{aligned} \tag{4.6}$$

Now, we bound the statistical distance between the distributions $(f(\pi(X)),Y)$ and $(f(U_n),Y)$ using Eq. (4.6) by partitioning the set Π of permutations into two disjoint subsets: Permutations π for which $\Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big] > \epsilon$, and permutations π for which $\Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big] \le \epsilon$. Specifically, it holds that

$$\begin{aligned}
&\Pr_{\pi \leftarrow \Pi}\big[\mathbf{SD}\big((f(\pi(X)),Y),(f(U_n),Y)\big) > 2\epsilon\big] \\
&\le \Pr_{\pi \leftarrow \Pi}\left[\Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big] > \epsilon\right] \\
&\quad + \Pr_{\pi \leftarrow \Pi}\left[\mathbf{SD}\left((f(\pi(X)),Y),(f(U_n),Y)\right) > 2\epsilon \;\middle|\; \Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big] \le \epsilon\right] \\
&\le \frac{2^{n-\ell-t}}{\epsilon} + 0.
\end{aligned}$$

To see that $\Pr_{\pi \leftarrow \Pi}\big[\mathbf{SD}\left((f(\pi(X)),Y),(f(U_n),Y)\right) > 2\epsilon \mid \Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big] \le \epsilon\big] = 0$, note that

$$\begin{aligned}
&\mathbf{SD}\left((f(\pi(X)),Y),(f(U_n),Y)\right) \\
&\quad \le \Pr_{y \leftarrow Y}\big[\mathsf{Bad}_\pi(y)\big] + \mathbf{SD}\Big(\big(f(\pi(X)),Y|_{\neg\mathsf{Bad}_\pi(y)}\big),\big(f(U_n),Y|_{\neg\mathsf{Bad}_\pi(y)}\big)\Big),
\end{aligned}$$

where each term is at most ϵ (from the conditioning event and the definition of $\mathsf{Bad}_\pi(y)$, respectively). This completes the proof of the lemma. □

We now use Lemma 4.5 and an inductive argument to show that applying $f \circ \pi$ allows us to deterministically extract from a set $\mathcal{X}$ of (T, k)-block-sources.

Theorem 4.6. *Let $f : \{0,1\}^n \to \{0,1\}^{n'}$ be (n, ℓ)-lossy, let Π be a t-wise δ-dependent collection of permutations over $\{0,1\}^n$ where $t = p + n - \ell + \log(T/\epsilon) + \log(T/\gamma) + 1$ and $\delta \leq 2^{-nt}$, and let $\mathcal{X}$ be a set of (T, k)-block-sources over $\{0,1\}^n$ such that $|\mathcal{X}| \leq 2^p$. Then, for every $0 < \epsilon < 1$ such that*

$$k \geq n - \ell + 2\log(1/\epsilon) + 2\log T + 2\log t + \Theta(1),$$

with probability at least $1 - \gamma$ over the choice of $\pi \in \Pi$, for every $\boldsymbol{X} = (X_1, \ldots, X_T) \in \mathcal{X}$ it holds that

$$\mathbf{SD}\left(\left(f\left(\pi\left(X_1\right)\right), \ldots, f\left(\pi\left(X_T\right)\right)\right), \left(f\left(U_n^{(1)}\right), \ldots, f\left(U_n^{(T)}\right)\right)\right) \leq \epsilon,$$

where $U_n^{(1)}, \ldots, U_n^{(T)}$ are T independent instances of the uniform distribution over $\{0,1\}^n$.

Proof. Fix a (T, k)-block-source $(X_1, \ldots, X_T) \in \mathcal{X}$. We prove the theorem using induction on the block index i of $(X_1, \ldots, X_T)$ starting with $i = T$ and ending with $i = 1$.

In particular, we show that for every $(X_1, \ldots, X_T) \in \mathcal{X}$ and every $i \in [T]$, it holds that

$$\begin{aligned} \Pr_{\pi \leftarrow \Pi}\Bigg[& \mathbf{SD}\Big(\left(X_1, \ldots, X_{i-1}, f\left(\pi\left(X_i\right)\right), \ldots, f\left(\pi\left(X_T\right)\right)\right), \\ & \left(X_1, \ldots, X_{i-1}, f\left(U_n^{(i)}\right), \ldots, f\left(U_n^{(T)}\right)\right)\Big) > \frac{\epsilon(T - i + 1)}{T}\Bigg] \\ & \leq \frac{2^{-p} \cdot \gamma(T - i + 1)}{T}. \end{aligned} \tag{4.7}$$

The base case when $i = T$ follows from Lemma 4.5 (by setting the distributions $X = X_T$, $Y = (X_1, \ldots, X_{T-1})$, and with $\epsilon/2T$ instead of ϵ) and noting that from the definition of a (T, k)-block-source, $\mathbf{H}_\infty(X|Y = y) \geq k$ as required.

We now assume that Eq. (4.7) holds for some $2 \leq i \leq T$ and prove that it holds for $i - 1$. From the triangle inequality, it holds that

$$\begin{aligned} \Pr_{\pi \leftarrow \Pi}\Bigg[& \mathbf{SD}\Big(\left(X_1, \ldots, X_{i-2}, f\left(\pi\left(X_{i-1}\right)\right), \ldots, f\left(\pi\left(X_T\right)\right)\right), \\ & \left(X_1, \ldots, X_{i-2}, f\left(U_n^{(i-1)}\right), \ldots, f\left(U_n^{(T)}\right)\right)\Big) > \frac{\epsilon(T - (i-1) + 1)}{T}\Bigg] \end{aligned}$$

$$\leq \Pr_{\pi \leftarrow \Pi}\Bigg[\mathbf{SD}\Big((X_1, \ldots, X_{i-2}, f(\pi(X_{i-1})), \ldots, f(\pi(X_T))),$$

$$\Big(X_1, \ldots, X_{i-2}, f(\pi(X_{i-1})), f\left(U_n^{(i)}\right), \ldots, f\left(U_n^{(T)}\right)\Big)\Big) > \frac{\epsilon(T-i+1)}{T}\Bigg]$$

$$+ \Pr_{\pi \leftarrow \Pi}\Bigg[\mathbf{SD}\Big(\Big(X_1, \ldots, X_{i-2}, f(\pi(X_{i-1})), f\left(U_n^{(i)}\right), \ldots, f\left(U_n^{(T)}\right)\Big) \tag{4.8}$$

$$\Big(X_1, \ldots, X_{i-2}, f\left(U_n^{(i-1)}\right), \ldots, f\left(U_n^{(T)}\right)\Big)\Big) > \frac{\epsilon}{T}\Bigg]. \tag{4.9}$$

$$\leq \frac{2^{-p} \cdot \gamma(T-i+1)}{T} + \frac{2^{-p} \cdot \gamma}{T} = \frac{2^{-p} \cdot \gamma(T-(i-1)+1)}{T}. \tag{4.10}$$

The two terms in Eq. (4.10) are derived as follows. The term in Eq. (4.8) is bounded by applying $f(\pi(\cdot))$ to X_{i-1} in Eq. (4.7) (i.e., by considering inductive step i) and noting that applying a (deterministic) function to any component cannot increase the statistical distance of two distributions. The term in Eq. (4.9) follows from Lemma (4.5) (by setting the distributions $X = X_{i-1}$, $Y = (X_1, \ldots, X_{i-2})$, and with $\epsilon/2T$ instead of ϵ) for our choice of parameter t and observing that the remaining components $f\left(U^i\right), \ldots, f\left(U^{(T)}\right)$ are sampled independently and identically in both distributions.

We complete the inductive argument in this manner. Now, setting $i = 1$ in Eq. (4.7) and applying a union bound over all 2^p possible (T, k)-block-sources in $\mathcal{X}$ completes the proof of the theorem. □

4.3. *Proof of Claim 4.2*

For every $x \in \{0,1\}^n$ define

$$W_x \stackrel{\text{def}}{=} 2^k \cdot p_x \cdot \mathbb{I}_{f(\pi^*(x))=y}, \tag{4.11}$$

where π^* is sampled uniformly at random from the set of all permutations over $\{0,1\}^n$, and let $W = \sum_{x \in \{0,1\}^n} W_x$. Note that the W_x's are defined in a similar manner to the Q_x's in Sect. 4, where the only difference is that here we consider the set of all permutations whereas in Sect. 4 we considered a t-wise δ-dependent collection of permutations. Note that

$$\mathbb{E}[(Q-\mu)^t] = \sum_{x_1,\ldots,x_t \in \{0,1\}^n} \mathbb{E}\left[\prod_{i=1}^{t}(Q_{x_i}-\mu)\right]$$

$$\leq \sum_{x_1,\ldots,x_t \in \{0,1\}^n} \mathbb{E}\left[\prod_{i=1}^{t}(W_{x_i}-\mu)\right] + \delta \cdot 2^{nt} \tag{4.12}$$

$$= \mathbb{E}\left[(W-\mu)^t\right] + \delta \cdot 2^{nt}, \tag{4.13}$$

where (4.12) follows from the definition of a t-wise δ-dependent collection of permutations.

If the W_x's were *independent* random variables, then we can trace the proof of [6, Lemma 2.3] to bound $\mathbb{E}[(W-\mu)^t]$ in (4.13) as done in [22, Prop. A.1]. However, the W_x's are not independent as they all share the same underlying permutation. Nevertheless, the main observation is that although the W_x's are not independent, for any integer $d \geq 1$, any $x_1, \ldots, x_d \in \{0,1\}^n$, and any integers $e_1, \ldots, e_d \geq 0$ it holds that

$$\mathbb{E}[W_{x_1}^{e_1} W_{x_2}^{e_2} \cdots W_{x_d}^{e_d}] \leq \mathbb{E}[W_{x_1}^{e_1}] \cdot \mathbb{E}[W_{x_2}^{e_2}] \cdots \mathbb{E}[W_{x_d}^{e_d}]. \tag{4.14}$$

This follows from the definition of W_x's and observing that the indicator variables $\mathbb{I}_{f(\pi^*(x))=y}$ have a higher probability of being 0 conditioned on the other indicator variables being 1 because π^* is a permutation. We use this in inequality (4.16) for deriving Lemma 4.7 below.

To bound the first term in (4.13), we first derive a variant of a lemma used in [20] and [6] applied to the W_x's. Lemma 4.7, stated and proved below, follows the proof outline of [6, Lemma A.5] closely but incorporates the inequality in Eq. (4.14).

Lemma 4.7. *Suppose that $W_{x_1}, \ldots, W_{x_{2^n}}$ are random variables as defined in Eq. (4.11) and let $W = \sum_{x \in \{0,1\}^n} W_x$. Then, for any $a \geq 0$ it holds that*

$$\Pr[|W - \mu| > a] < \max(2e^{-3a^2/8\mu}, e^{-2a/5}). \tag{4.15}$$

Proof. For some parameter γ, which will be optimized for later,

$$\begin{aligned}
\Pr[W - \mu > a] &\leq \frac{\mathbb{E}[e^{\gamma(W-\mu)}]}{e^{\gamma a}} = \frac{e^{-\mu\gamma}}{e^{\gamma a}} \cdot \sum_{i=1}^{\infty} \frac{\gamma^i}{i!} \mathbb{E}[W^i] \\
&= \frac{e^{-\mu\gamma}}{e^{\gamma a}} \cdot \sum_{i=1}^{\infty} \frac{\gamma^i}{i!} \sum_{x_1,\ldots,x_i} \mathbb{E}\left[\prod_{j=1}^{i} W_{x_j}\right] \\
&\leq \frac{e^{-\mu\gamma}}{e^{\gamma a}} \cdot \sum_{i=1}^{\infty} \frac{\gamma^i}{i!} \sum_{x_1,\ldots,x_i} \prod_{i=1}^{i} \mathbb{E}\left[W_{x_j}\right] \text{ (from Eq. (4.14))} \\
&\leq \prod_{x \in \{0,1\}^n} \mathbb{E}[e^{\gamma(W_x - \mu/2^n)}]/e^{\gamma a}.
\end{aligned} \tag{4.16}$$

From here on in, the proof is identical to the proof of [20, Lemma 2.2.9] and included here for completeness.

Let $\nu \stackrel{\text{def}}{=} \mu/2^n$. Now, by the convexity of the exponential function

$$\mathbb{E}[e^{\gamma(W_x - \nu)}] \leq (1-\nu)e^{-\gamma\nu} + \nu e^{\gamma(1-\nu)}.$$

Taking Taylor expansions and combining terms,

$$\begin{aligned}\mathbb{E}[e^{\gamma(W_x-\nu)}] &\le 1+\nu(1-\nu)\left(\frac{\gamma^2}{2!}+\left((1-\nu)^2-\nu^2\right)\frac{\gamma^3}{3!}+\left((1-\nu)^3+\nu^3\right)\frac{\nu^4}{4!}+\cdots\right)\\ &\le 1+\nu\left(\frac{\gamma^2}{2!}+\frac{|\gamma|^3}{3!}+\cdots\right)\\ &= 1+\nu\left(e^{|\gamma|}-1-|\gamma|\right)\\ &= 1+\nu\frac{\gamma^2}{2!}\left(\frac{e^{|\gamma|}-1-|\gamma|}{\gamma^2/2}\right).\end{aligned}$$

Restricting $|\gamma| < 4/5$ it follows

$$\mathbb{E}[e^{\gamma(W_x-\nu)}] \le 1+\nu\frac{2\gamma^2}{3} \le e^{2\nu\gamma^2/3},$$

which implies that $\mathbb{E}[e^{\gamma(W-\mu)}] \le e^{2\mu\gamma^2/3}$. Therefore,

$$\Pr[W-\mu > a] < e^{2\mu\gamma^2/3-\gamma a}.$$

The optimal value for γ in the above formula is $3a/4\mu$. But we must have that $|\gamma| < 4/5$, so we let $\gamma = \min(3a/4\mu, 4/5)$. For $a \le 16\mu/5$. $t = 3a/4\mu$, so

$$\Pr[W-\mu > a] < e^{3a^2/8\mu-3a^2/4\mu} = e^{-3a^2/8\mu}.$$

For $a \ge 16\mu/15$, $\gamma = 4/5$, so

$$\Pr[W-\mu > a] < e^{32\mu/75-4a/5} \le e^{2a/5-4a/5} = e^{-2a/5}.$$

Similarly, we note that

$$\Pr[W-\mu < -a] < \mathbb{E}[e^{\gamma(W-\mu)}]/e^{-\gamma a},$$

which is optimized by letting $\gamma = -3a/4\mu$, obtaining

$$\Pr[W-\mu < -a] < e^{-3a^2/8\mu}.$$

Note that we do not consider the case $a > 16\mu/15$ as $\Pr[W < 0] = 0$. Therefore, we have

$$\Pr\left[|W-\mu| > a\right] < \max(2e^{-3a^2/8\mu}, e^{-2a/5}).$$

□

Given Lemma 4.7, the proof of Claim 4.2 proceeds as follows.

$$\begin{aligned}\mathbb{E}[(W-\mu)^t] &= \int_0^\infty \Pr\left[|W-\mu| > x^{1/t}\right] dx \\ &\leq 2\int_0^\infty \exp\left(-\frac{3x^{2/t}}{8\mu}\right) \mathrm{d}x + \int_0^\infty \exp\left(-\frac{2x^{1/t}}{5}\right) \mathrm{d}x. \end{aligned} \tag{4.17}$$

Changing variables to $y = 3x^{2/t}/8\mu$ and then using the definition of the Gamma function and Stirling's approximation the first term in the sum in (4.17) can be bound by

$$\begin{aligned} 2 \cdot \frac{t}{2}\left(\frac{8\mu}{3}\right)^{t/2} \int_0^\infty y^{t/2-1} e^{-y} \mathrm{d}y &= 2 \cdot \left(\frac{8\mu}{3}\right)^{t/2} \cdot \frac{t}{2} \cdot \Gamma\left(\frac{t}{2}\right) \\ &= 2 \cdot \left(\frac{8\mu}{3}\right)^{t/2} \cdot (t/2)! \\ &< 2 \cdot \left(\frac{8\mu}{3}\right)^{t/2} \cdot e^{1/6t}\sqrt{\pi t}\left(\frac{t}{2e}\right)^{t/2} \\ &= 2e^{1/6t}\sqrt{\pi t}\left(\frac{4}{3e}\right)^{t/2} \cdot (t\mu)^{t/2}. \end{aligned} \tag{4.18}$$

Similarly with a change of variable $z = 2x^{1/t}/5$, the second term in (4.17) can be bounded by

$$e^{1/12t}\sqrt{2\pi t} \cdot \left(\frac{5}{2e}\right)^t \cdot t^t. \tag{4.19}$$

Putting together (4.17), (4.18), and (4.19) together to bound the first term in (4.13) and setting the constant $C_t = 2e^{1/6t}\sqrt{\pi t}(4/3e)^{t/2} + e^{1/12t}\sqrt{2\pi t}(5/2e)^t$ concludes the proof of Claim 4.2. □

5. Chosen-Plaintext Security based on Lossy Trapdoor Functions

In this section we present our basic construction of a public-key deterministic encryption scheme that is secure according to our notion of adaptive security. We refer the reader to Sect. 1.3 for a high-level description of the scheme, and of the main challenges and ideas underlying our approach. In what follows we formally describe the scheme, discuss the parameters that we obtain using known instantiations of its building blocks, and prove its security.

The scheme $\mathcal{DE}$ Let $n = n(\lambda)$, $\ell = \ell(\lambda)$, $t = t(\lambda)$ and $\delta = \delta(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. Let $(\mathsf{Gen}_0, \mathsf{Gen}_1, \mathsf{F}, \mathsf{F}^{-1})$ be a collection of (n, ℓ)-lossy trapdoor functions, and for every $\lambda \in \mathbb{N}$ let Π_λ be a t-wise δ-dependent collection

of permutations over $\{0,1\}^n$.[9] Our scheme $\mathcal{DE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is defined as follows:

- **Key generation** The key-generation algorithm KeyGen on input 1^λ samples $(\sigma, \tau) \leftarrow \mathsf{Gen}_1(1^\lambda)$ and $\pi \leftarrow \Pi_\lambda$. It then outputs $pk = (\sigma, \pi)$ and $sk = \tau$.
- **Encryption** The encryption algorithm Enc on input a public key $pk = (\sigma, \pi)$ and a message $m \in \{0,1\}^n$ outputs $c = \mathsf{F}(\sigma, \pi(m))$.
- **Decryption** The decryption algorithm Dec on input a secret key $sk = \tau$ and a ciphertext c outputs $m = \pi^{-1}\left(\mathsf{F}^{-1}(\tau, c)\right)$.

Theorem 5.1. *The scheme $\mathcal{DE}$ is block-wise (p, T, k)-ACD-CPA-secure for any $n = n(\lambda)$, $\ell = \ell(\lambda)$, $p = p(\lambda)$, and $T = T(\lambda)$ by setting $t = p + n - \ell + \log T + \omega(\log\lambda)$, $k = n - \ell + 2\log T + 2\log t + \omega(\log\lambda)$, and $\delta = 2^{-nt}$.*

Parameters Using existing constructions of lossy trapdoor functions (see Sect. 2.3), for any $n = n(\lambda)$ and for any constant $0 < \epsilon < 1$ we can instantiate our scheme with $\ell = n - n^\epsilon$. Therefore, for any $n = n(\lambda)$, $p = p(\lambda)$, and $T = T(\lambda)$, we obtain schemes with $t = p + n^\epsilon + \omega(\log\lambda)$, $k = n^\epsilon + \omega(\log\lambda)$, and $\delta = 2^{-nt}$.

Proof overview The proof of security consists of two steps. Let $\mathcal{X}$ be a set of at most 2^p plaintext distributions. First, the security of the collection of lossy trapdoor functions allows us to replace the injective function $f(\cdot) = \mathsf{F}(\sigma, \cdot)$ with a lossy function $\widetilde{f}(\cdot) = \mathsf{F}(\tilde{\sigma}, \cdot)$. Next, we use the high-moment crooked leftover hash lemma derived in Sect. 4 and show that with overwhelming probability over the choice of the permutation π, it holds that for *every* plaintext distribution $\boldsymbol{M} \in \mathcal{X}$, the two distributions $\widetilde{f}(\pi(\boldsymbol{M}))$ and $\widetilde{f}(\boldsymbol{U})$ are statistically close, even given the public key (i.e., $\tilde{\sigma}$ and π). Therefore, essentially no information on the plaintext is revealed—even when the specific choice of $\boldsymbol{M} \in \mathcal{X}$ may adaptively depend on pk. A second application of the security of the collection of lossy trapdoor functions allows us to switch back from the lossy function to an injective one, which exactly reflects the output of the real-or-random encryption oracle in the rand mode. We give a full proof of the theorem below.

Proof of Theorem 5.1 Using Theorem 3.4 it suffices to prove that $\mathcal{DE}$ is block-wise (p, T, k)-ACD1-CPA-secure. Let $\mathcal{A}$ be a 2^p-bounded (T, k)-block-source adversary that queries the oracle RoR at most once. In what follows, we describe four experiments, $\mathsf{Expt}_0, \ldots, \mathsf{Expt}_3$, and derive a series of claims relating them. We then combine these claims to bound the advantage of the adversary.

Experiment Expt_0 This is the experiment $\mathsf{Expt}^{\mathsf{real}}_{\mathcal{DE},\mathcal{A}}(\lambda)$ (recall Definition 3.3).

Experiment Expt_1 This experiment is obtained from Expt_0 by modifying the key-generation algorithm to sample a *lossy* function index $\tilde{\sigma}$ rather than an injective function index σ.

Claim 5.2. $\left|\Pr\left[\mathsf{Expt}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1(\lambda) = 1\right]\right|$ *is negligible in* λ.

[9] As discussed in Sect. 2.1, recall that we find it convenient to rely (without loss of generality) on t-wise δ-dependent collections of permutations Π in which the marginal distribution $\pi(x)$ is *perfectly uniform* (as opposed to just δ-close to uniform) for any $x \in \{0,1\}^n$ over the choice of $\pi \leftarrow \Pi$.

Proof. As $\mathcal{A}$ and RoR can be simulated in probabilistic polynomial time, the security of the collection of lossy trapdoor functions $(\mathsf{Gen}_0, \mathsf{Gen}_1, \mathsf{F}, \mathsf{F}^{-1})$ immediately implies Claim 5.2. Specifically, any efficient adversary $\mathcal{A}$ for which $|\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]|$ is non-negligible can be used to distinguish a randomly sampled injective key σ from a random sampled lossy key $\tilde{\sigma}$. □

Experiment Expt_2 This experiment is obtained from Expt_1 by running RoR in rand mode rather than in real mode (using a lossy function index $\tilde{\sigma}$ as in Expt_1).

Claim 5.3. $\left|\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]\right|$ *is negligible in* λ.

Proof. We fix a lossy key $\tilde{\sigma}$ and argue Claim 5.3 for any such lossy key. Note that $\mathcal{A}$'s view in Expt_1 is $\left(\tilde{\sigma}, \pi, f\left(\pi\left(X^{(1)}\right)\right), \ldots, f\left(\pi\left(X^{(T)}\right)\right)\right)$ where $\tilde{\sigma}$ is a fixed lossy key, $\pi \leftarrow \Pi$, $f(\cdot) \stackrel{\mathsf{def}}{=} \mathsf{F}(\tilde{\sigma}, \cdot)$, and $\boldsymbol{X} = \left(X^{(1)}, \ldots, X^{(T)}\right)$ is a (T, k)-block-source. Additionally, as $\mathcal{A}$ is 2^p-bounded, there is a set $\mathcal{X}$ of size at most 2^p such that $\boldsymbol{X} \in \mathcal{X}$.

Similarly, $\mathcal{A}$'s view in Expt_2 is $\left(\tilde{\sigma}, \pi, f\left(U_n^{(1)}\right), \ldots, f\left(U_n^{(T)}\right)\right)$, where $U_n^{(1)}, \ldots, U_n^{(T)}$ are T independent instances of the uniform distribution of $\{0,1\}^n$. Our choice of parameters enables us to apply Theorem 4.6 and obtain that with an overwhelming probability over the choice of $\pi \leftarrow \Pi$, for all such block-sources $\boldsymbol{X} = \left(X^{(1)}, \ldots, X^{(T)}\right) \in \mathcal{X}$ the distributions $\left(f\left(\pi\left(X^{(1)}\right)\right), \ldots, f\left(\pi\left(X^{(T)}\right)\right)\right)$ and $\left(f\left(U_n^{(1)}\right), \ldots, f\left(U_n^{(T)}\right)\right)$ are statistically close, and thus $|\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]|$ is negligible in the security parameter λ. □

Experiment Expt_3 This experiment is obtained from Expt_2 by modifying the key-generation algorithm to sample an *injective* function index σ rather than a lossy function index $\tilde{\sigma}$. That is, this is experiment $\mathsf{Expt}^{\mathsf{rand}}_{\mathcal{DE},\mathcal{A}}(\lambda)$ (recall Definition 3.3).

Claim 5.4. $\left|\Pr[\mathsf{Expt}_2(\lambda) = 1] - \Pr[\mathsf{Expt}_3(\lambda) = 1]\right|$ *is negligible in* λ.

Proof. This proof is identical to the proof of Claim 5.2. □

Completing the proof of Theorem 5.1. The definition of $\mathbf{Adv}^{\text{ACD-CPA}}_{\mathcal{DE},\mathcal{A}}(\lambda)$ implies that for any such adversary $\mathcal{A}$:

$$
\begin{aligned}
\mathbf{Adv}^{\text{ACD-CPA}}_{\mathcal{DE},\mathcal{A}}(\lambda) &\stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathcal{DE},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathcal{DE},\mathcal{A}}(\lambda) = 1\right]\right| \\
&= \left|\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_3(\lambda) = 1]\right| \\
&\leq \left|\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]\right| && (5.1) \\
&\quad + \left|\Pr[\mathsf{Expt}_1(\lambda) = 1] - \Pr[\mathsf{Expt}_2(\lambda) = 1]\right| && (5.2) \\
&\quad + \left|\Pr[\mathsf{Expt}_2(\lambda) = 1] - \Pr[\mathsf{Expt}_3(\lambda) = 1]\right|. && (5.3)
\end{aligned}
$$

Claims 5.2–5.4 state that the terms in Eqs. (5.1)–(5.3) are negligible, and this completes the proof of Theorem 5.1. □

6. $\mathcal{R}$-Lossy Trapdoor Functions

The notion of $\mathcal{R}$-lossy *public-key encryption schemes* was put forward by Boyle et al. [8], and here we define an analogous notion for *trapdoor functions*. Informally, an $\mathcal{R}$-lossy trapdoor function family is a collection of tagged functions where the set of possible tags is partitioned into two subsets: *injective* tags, and *lossy* tags. Functions evaluated with an injective tag can be efficiently inverted with a trapdoor (where all injective tags share the same trapdoor information). On the other hand, functions evaluated with a lossy tag lose information—the size of their image is significantly smaller than the size of their domain. The partitioning of the tags is defined by a binary relation $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$: the key-generation algorithm receives as input an *initialization value* $K \in \mathcal{K}$ and this partitions the set tags $\mathcal{T}$ so that $t \in \mathcal{T}$ is lossy if and only if $(K, t) \in \mathcal{R}$. More, formally, we require that the relation $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$ consists of a sequence of efficiently (in λ) recognizable sub-relations $\mathcal{R}_\lambda \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$. The only computational requirement of an $\mathcal{R}$-lossy trapdoor function family is that its description hides the initialization value K.

Definition 6.1. ($\mathcal{R}$-*lossy trapdoor functions*) Let $n : \mathbb{N} \to \mathbb{R}$ and $\ell : \mathbb{N} \to \mathbb{R}$ be nonnegative functions, and for any $\lambda \in \mathbb{N}$ let $n = n(\lambda)$ and $\ell = \ell(\lambda)$. Also, let $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$ be an efficiently computable binary relation. An $\mathcal{R}$-(n, ℓ)-*lossy trapdoor function family* is a triplet of probabilistic polynomial-time algorithms $\mathsf{RLTDF} = (\mathsf{Gen}_\mathcal{R}, \mathsf{G}, \mathsf{G}^{-1})$ such that:

1. **Key generation** For any initialization value $K \in \mathcal{K}_\lambda$, algorithm $\mathsf{Gen}_\mathcal{R}(1^\lambda, K)$ outputs a public index σ and a trapdoor τ.
2. **Evaluation** For any $K \in \mathcal{K}$, $(\sigma, \tau) \leftarrow \mathsf{Gen}_\mathcal{R}(1^\lambda, K)$, and any $t \in \mathcal{T}$, algorithm $\mathsf{G}(\sigma, t, \cdot)$ computes a function $f_{\sigma,t} : \{0,1\}^n \to \{0,1\}^*$ with one of the two following properties:
   - Lossy tags: If $(K, t) \in \mathcal{R}$, then the image of $f_{\sigma,t}$ has size at most $2^{n-\ell}$.
   - Injective tags: If $(K, t) \notin \mathcal{R}$, then the function $f_{\sigma,t}$ is injective.
3. **Inversion under injective tags** For any initialization value $K \in \mathcal{K}$ and tag $t \in \mathcal{T}$ such that $(K, t) \notin \mathcal{R}$, and for any input $x \in \{0,1\}^n$, we have $\mathsf{G}^{-1}(\tau, t, \mathsf{G}(\sigma, t, x)) = x$.
4. **Indistinguishability of initialization values** For every probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that

$$\mathbf{Adv}^{\mathcal{R}\text{-lossy}}_{\mathsf{RLTDF},\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{RLTDF},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{RLTDF},\mathcal{A}}(\lambda) = 1\right]\right| \leq \nu(\lambda),$$

where for each $b \in \{0,1\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{(b)}_{\mathsf{RLTDF},\mathcal{A}}(\lambda)$ is defined as follows:

(a) $(K_0, K_1, \mathsf{state}) \leftarrow \mathcal{A}(1^\lambda)$.
(b) $(\sigma, \tau) \leftarrow \mathsf{Gen}_\mathcal{R}(1^\lambda, K_b)$.
(c) $b' \leftarrow \mathcal{A}(1^\lambda, \sigma, \mathsf{state})$.
(d) Output b'.

6.1. *The Relation* $\mathcal{R}^{\mathrm{BM}}$

We are interested mainly in the bit-matching relation $\mathcal{R}^{\mathrm{BM}}$, as defined by Boyle et al. [8]. For every $\lambda \in \mathbb{N}$ let $\mathcal{K}_\lambda = \{0, 1, \perp\}^{v(\lambda)}$ and $\mathcal{T}_\lambda = \{0, 1\}^{v(\lambda)}$, and define $(K, t) \in \mathcal{R}^{\mathrm{BM}}_\lambda \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ if for every $i \in \{1, \ldots, v(\lambda)\}$ it holds that $K_i = t_i$ or $K_i = \perp$. That is, given some fixed initialization value K, the set of lossy tags t are exactly those whose bits match K in all positions i for which $K_i \neq \perp$.

In our construction of CCA-secure deterministic encryption schemes, the $\mathcal{R}^{\mathrm{BM}}$-lossy trapdoor functions will be used in combination with an *admissible hash function* (discussed in Sect. 2.2). An admissible hash function enables us to map messages to encryption tags such that, with high probability over an appropriate distribution of K, all decryption queries map to injective tags, while the challenge query maps to a lossy tag which loses information about the plaintext.

6.2. *Constructing* $\mathcal{R}^{\mathrm{BM}}$*-Lossy Trapdoor Functions*

We now present a generic construction of $\mathcal{R}^{\mathrm{BM}}$-lossy trapdoor functions based on any collection of lossy trapdoor functions. In turn, this implies that $\mathcal{R}^{\mathrm{BM}}$-lossy trapdoor functions can be based on a variety of number-theoretic assumptions.

Let $\mathsf{LTDF} = (\mathsf{Gen}_0, \mathsf{Gen}_1, \mathsf{F}, \mathsf{F}^{-1})$ be a collection of (n, ℓ)-lossy trapdoor functions. The key-generation algorithm of our collection of $\mathcal{R}^{\mathrm{BM}}$-lossy trapdoor functions samples $v(\lambda)$ pairs of keys from the collection LTDF. Each such pair is of one out of three possible types according to the symbols of the initialization value $K \in \{0, 1, \perp\}^{v(\lambda)}$. For every $i \in \{1, \ldots, v(\lambda)\}$, if $K_i = 0$ then the i-th pair consists of a lossy key and an injective key, if $K_i = 1$ then the i-th pair consists of an injective and a lossy key (i.e., the order is reversed), and if $K_i = \perp$ then i-th pair consists of two lossy keys. The evaluation algorithm given a tag $t \in \{0, 1\}^{v(\lambda)}$ and an input $x \in \{0, 1\}^n$ outputs the concatenation of the values obtained by evaluating one of the functions from each pair on x according to the corresponding bit of t. More formally, consider the following collection $\mathsf{RLTDF} = (\mathsf{Gen}_{\mathcal{R}^{\mathrm{BM}}}, \mathsf{G}, \mathsf{G}^{-1})$:

- **Key generation** On input 1^λ and an initialization value $K = K_1 \cdots K_{v(\lambda)} \in \{0, 1, \perp\}^{v(\lambda)}$, for every $1 \leq i \leq v(\lambda)$ algorithm $\mathsf{Gen}_{\mathcal{R}^{\mathrm{BM}}}$ produces a pair $((\sigma_{i,0}, \tau_{i,0}), (\sigma_{i,1}, \tau_{i,1}))$ as follows:
  - If $K_i = 0$ then it samples $\sigma_{i,0} \leftarrow \mathsf{Gen}_0(1^\lambda)$, $(\sigma_{i,1}, \tau_{i,1}) \leftarrow \mathsf{Gen}_1(1^\lambda)$, and sets $\tau_{i,0} = \perp$.
  - If $K_i = 1$ then it samples $(\sigma_{i,0}, \tau_{i,0}) \leftarrow \mathsf{Gen}_1(1^\lambda)$, and $\sigma_{i,1} \leftarrow \mathsf{Gen}_0(1^\lambda)$, and sets $\tau_{i,1} = \perp$.
  - If $K_i = \perp$ then it samples $\sigma_{i,0} \leftarrow \mathsf{Gen}_0(1^\lambda)$, $\sigma_{i,1} \leftarrow \mathsf{Gen}_0(1^\lambda)$, and sets $\tau_{i,0} = \tau_{i,1} = \perp$.

  It then outputs the pair (σ, τ) defined as

$$\sigma = \left(\left\{\left(\sigma_{i,0}, \sigma_{i,1}\right)\right\}_{i=1}^{v(\lambda)}\right)$$
$$\tau = \left(K, \left\{\left(\tau_{i,0}, \tau_{i,1}\right)\right\}_{i=1}^{v(\lambda)}\right)$$

- **Evaluation** On input a function index σ of the above form, a tag $t = t_1 \cdots t_{v(\lambda)} \in \{0,1\}^{v(\lambda)}$ and an input $x \in \{0,1\}^{n(\lambda)}$, algorithm G outputs

$$y = \left(\mathsf{F}_{\sigma_{1,t_1}}(x), \ldots, \mathsf{F}_{\sigma_{v(\lambda),t_{v(\lambda)}}}(x)\right)$$

- **Inversion** On input a trapdoor τ of the above form, a tag $t = t_1 \cdots t_{v(\lambda)} \in \{0,1\}^{v(\lambda)}$ and a value $y = (y_1, \ldots, y_{v(\lambda)})$, the inversion algorithm G^{-1} proceeds as follows. If $(K,t) \in \mathcal{R}^{\mathrm{BM}}$ (i.e., t is a lossy tag) then it outputs $\perp$. Otherwise (i.e., t is an injective tag), there exists an index $i \in \{1, \ldots, v(\lambda)\}$ such that $K_i \neq t_i$ and $K_i \neq \perp$, and therefore the pair $(\sigma_{i,t_i}, \tau_{i,t_i})$ corresponds to an injective function. In this case the inversion algorithm outputs $x = \mathsf{F}^{-1}(\tau_{i,t_i}, y_i)$.

Theorem 6.2. *For any $n = n(\lambda)$, $\ell = \ell(\lambda)$ and $v = v(\lambda)$, if $\mathsf{LTDF} = (\mathsf{Gen}_0, \mathsf{Gen}_1, \mathsf{F}, \mathsf{F}^{-1})$ is a collection of (n,ℓ)-lossy trapdoor functions, then $\mathsf{RLTDF} = (\mathsf{Gen}_{\mathcal{R}^{\mathrm{BM}}}, \mathsf{G}, \mathsf{G}^{-1})$ is a collection of $\mathcal{R}^{\mathrm{BM}}$-$(n, v\ell - (v-1)n)$-lossy trapdoor functions with v-bit tags.*

Proof. Indistinguishability of initialization values follows directly from the indistinguishability of lossy and injective keys of the underlying collection LTDF of lossy trapdoor functions via a straightforward hybrid argument. The correctness of the inversion algorithm under injective tags follows from the fact that for any injective tag t (i.e., $(K,t) \notin \mathcal{R}^{\mathrm{BM}}$) there exists an index $i \in \{1, \ldots, v(\lambda)\}$ such that $K_i \neq t_i$ and $K_i \neq \perp$, and therefore the pair $(\sigma_{i,t_i}, \tau_{i,t_i})$ corresponds to an injective function of LTDF. Lossiness of the function under lossy tags follows from the fact for any lossy tag t (i.e., $(K,t) \in \mathcal{R}^{\mathrm{BM}}$) and for any index $i \in \{1, \ldots, v(\lambda)\}$ it holds that σ_{i,t_i} corresponds to a lossy function of LTDF. Therefore, the possible number of output values for a lossy tag is at most $\left(2^{n-\ell}\right)^v = 2^{n-(v\ell-(v-1)n)}$. □

Parameters In our construction of a CCA-secure deterministic public-key encryption scheme in Sect. 7, v is the output length of an admissible hash function, which is n^ϵ for any constant $0 < \epsilon < 1$ [1]. Several of the known constructions of lossy trapdoor functions (see Sect. 2.3) offer $\ell = n - n^\epsilon$, and thus Theorem 6.2 guarantees that the possible number of output values for any lossy tag in our construction is at most $2^{n-(v\ell-(v-1)n)} = 2^{n^{2\epsilon}}$. That is, based on existing constructions of lossy trapdoor functions, for any constant $0 < \epsilon < 1$ Theorem 6.2 yields constructions of $\mathcal{R}^{\mathrm{BM}}$-$(n, n - n^{2\epsilon})$-lossy trapdoor functions with n^ϵ-bit tags.

7. Chosen-Ciphertext Security based on $\mathcal{R}$-Lossy Trapdoor Functions

In this section we present a construction of a public-key deterministic encryption scheme that is secure according to our notion of adaptive security even when adversaries can access a decryption oracle. As discussed in Sect. 1.3, our construction is inspired by that of Boldyreva et al. [5] combined with the approach of Boneh and Boyen [1] (and its refinement by Cash et al. [9]) for converting a large class of selectively secure IBE schemes to fully secure ones, and the notion of $\mathcal{R}$-lossy trapdoor functions that we

introduced in Sect. 6 following Boyle et al. [8]. In what follows we formally describe the scheme, discuss the parameters that we obtain using known instantiations of its building blocks, and prove its security.

The scheme $\mathcal{DE}_{\mathrm{CCA}}$ Let $n = n(\lambda)$, $\ell = \ell(\lambda)$, $v = v(\lambda)$, $t_1 = t_1(\lambda)$, $t_2 = t_2(\lambda)$, $\delta_1 = \delta_1(\lambda)$, and $\delta_2 = \delta_2(\lambda)$ be functions of the security parameter $\lambda \in \mathbb{N}$. Our construction relies on the following building blocks[10]:

1. A collection $\mathcal{H}_\lambda$ of admissible hash functions $h : \{0,1\}^n \to \{0,1\}^v$ for every $\lambda \in \mathbb{N}$.
2. A collection $(\mathsf{Gen}_0, \mathsf{Gen}_1, \mathsf{F}, \mathsf{F}^{-1})$ of (n, ℓ)-lossy trapdoor functions.
3. A collection $(\mathsf{Gen}_{\mathrm{BM}}, \mathsf{G}, \mathsf{G}^{-1})$ of $\mathcal{R}^{\mathrm{BM}}$-$(n, \ell)$-lossy trapdoor functions.
4. A t_1-wise δ_1-dependent collection $\Pi^{(1)}_\lambda$ of permutations over $\{0,1\}^n$ for every $\lambda \in \mathbb{N}$.
5. A t_2-wise δ_2-dependent collection $\Pi^{(2)}_\lambda$ of permutations over $\{0,1\}^n$ for every $\lambda \in \mathbb{N}$.

Our scheme $\mathcal{DE}_{\mathrm{CCA}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is defined as follows:

- **Key generation** The key-generation algorithm KeyGen on input 1^λ samples $h \leftarrow \mathcal{H}_\lambda$, $(\sigma_f, \tau_f) \leftarrow \mathsf{Gen}_1(1^\lambda)$, $K \leftarrow \mathcal{K}_\lambda$, $(\sigma_g, \tau_g) \leftarrow \mathsf{Gen}_{\mathrm{BM}}(1^\lambda, K)$, $\pi_1 \leftarrow \Pi^{(1)}_\lambda$, and $\pi_2 \leftarrow \Pi^{(2)}_\lambda$. Then, it outputs $pk = \big(h, \sigma_f, \sigma_g, \pi_1, \pi_2\big)$ and $sk = (\tau_f, \tau_g)$.
- **Encryption** The encryption algorithm Enc on input a public key $pk = (h, \sigma_f, \sigma_g, \pi_1, \pi_2)$ and a message $m \in \{0,1\}^n$ outputs

$$c = \Big(h(\pi_1(m)),\ \ \mathsf{F}\big(\sigma_f, \pi_2(m)\big),\ \ \mathsf{G}\big(\sigma_g, h(\pi_1(m)), \pi_2(m)\big)\Big).$$

- **Decryption** The decryption algorithm Dec on input a secret key $sk = (\tau_f, \tau_g)$ and a ciphertext tuple (c_h, c_f, c_g) first computes $m = \pi_2^{-1}\left(\mathsf{F}^{-1}(\tau_f, c_f)\right)$. Then, if $\mathsf{Enc}_{pk}(m) = (c_h, c_f, c_g)$ it outputs m, and otherwise it outputs $\perp$.

In other words, the decryption algorithm inverts c_f using the trapdoor τ_f, and outputs m if the ciphertext is well-formed.

Theorem 7.1. *The scheme $\mathcal{DE}_{\mathrm{CCA}}$ is block-wise (p, T, k)-ACD-CCA-secure for any $n = n(\lambda)$, $\ell = \ell(\lambda)$, $v = v(\lambda)$, $p = p(\lambda)$, and $T = T(\lambda)$ by setting*

$$\begin{aligned} t_1 &= p + (T-1)\cdot n + v + \omega(\log\lambda), & \delta_1 &= 2^{-nt_1},\\ t_2 &= p + (T-1)\cdot n + v + n - (2\ell - n) + \omega(\log\lambda), & \delta_2 &= 2^{-nt_2},\\ k &= \max\big(n - (2\ell - n), v\big) + 2\log t_2 + \omega(\log\lambda). \end{aligned}$$

Parameters Using existing constructions of admissible hash functions and lossy trapdoor functions (see Sects. 2.2 and 2.3, respectively), and using our construction of $\mathcal{R}^{\mathrm{BM}}$-lossy trapdoor functions (see Sect. 6), for any $n = n(\lambda)$ and for any constant $0 < \epsilon < 1$

[10] As discussed in Sect. 2.1, recall that we find it convenient to rely (without loss of generality) on t-wise δ-dependent collections of permutations Π in which the marginal distribution $\pi(x)$ is *perfectly uniform* (as opposed to just δ-close to uniform) for any $x \in \{0,1\}^n$ over the choice of $\pi \leftarrow \Pi$.

we can instantiate our scheme with $v = n^\epsilon$ and $\ell = n - n^\epsilon$. Therefore, for any $n = n(\lambda)$, $p = p(\lambda)$, and $T = T(\lambda)$, we obtain schemes with

$$t_1 = p + (T-1)\cdot n + n^\epsilon + \omega(\log\lambda), \qquad \delta_1 = 2^{-nt_1},$$
$$t_2 = p + (T-1)\cdot n + 3n^{2\epsilon} + \omega(\log\lambda), \qquad \delta_2 = 2^{-nt_2},$$
$$k = 2n^{2\epsilon} + \omega(\log\lambda).$$

Proof overview. On a high level, an encryption of a message m in our scheme consists of three ciphertext components. The first ciphertext component is a short tag $h(\pi_1(m))$, where h is an admissible hash function and π_1 is a permutation. Looking ahead, our high-moment crooked leftover hash lemma will enable us to argue that such a tag reveals essentially no information on m, as h is a compressing function. The second ciphertext component is $f(\pi_2(m))$, where f is an injective function sampled from a collection of lossy trapdoor functions, and π_2 is a permutation. The third ciphertext component is $g(h(\pi_1(m)), \pi_2(m))$ where g is sampled from a collection of $\mathcal{R}^{\mathrm{BM}}$-lossy trapdoor functions, and is evaluated on $\pi_2(m)$ using the tag $h(\pi_1(m))$. The role of the second and third components is to allow us to prove security using a generalization of the "all-but-one" simulation paradigm, as discussed in Sect. 1.3, to our setting of adaptive adversaries.

Specifically, in our proof of security, the combination of the admissible hash function and the $\mathcal{R}^{\mathrm{BM}}$-lossy trapdoor function enables us to generate a public key for which, with a non-negligible probability, all decryption queries correspond to injective tags for g, while the challenge ciphertext corresponds to a lossy tag for g—even when the challenge plaintext is not known in advance. This is done via a subtle artificial abort argument, similar to the one of Cash et al. [9]. Looking ahead, such a partitioning of the tags will enable us to simulate the decryption oracle for answering all decryption queries, and apply our high-moment crooked leftover hash lemma to argue that the second and third ciphertext components, $f(\pi_2(m))$ and $g(h(\pi_1(m)), \pi_2(m))$, reveal essentially no information on m. For applying our lemma, we observe that f can be replaced by a lossy function $\widetilde{f}$ (while answering decryption queries through the trapdoor for g—as all decryption queries correspond to injective tags for g), and that g is evaluated on $\pi_2(m)$ using a lossy tag $h(\pi_1(m))$.

Proof of Theorem 7.1. Using Theorem 3.7, it suffices to prove that $\mathcal{DE}_{\mathrm{CCA}}$ is block-wise (p, T, k)-ACD1-CCA-secure. Let $\mathcal{A}$ be a 2^p-bounded (T, k)-block-source chosen-ciphertext adversary that queries the real-or-random oracle RoR exactly once. We assume without loss of generality that $\mathcal{A}$ always makes q decryption queries for some polynomial $q = q(\lambda)$. We denote by $c^{(1)}, \ldots, c^{(q)}$ the random variables corresponding to these decryption queries, and by $\boldsymbol{c}^* = \left(c_1^*, \ldots, c_T^*\right)$ the vector of random variables corresponding to the challenge ciphertexts returned by the RoR oracle.

For every $i \in \{0, \ldots, T\}$ we define an experiment $\mathsf{Expt}^{(i)}$ that is obtained from the experiment $\mathsf{Expt}^{\mathsf{realCCA}}_{\mathcal{DE}_{\mathrm{CCA}},\mathcal{A}}$ by modifying the distribution of the challenge ciphertext. Recall that in the experiment $\mathsf{Expt}^{\mathsf{realCCA}}_{\mathcal{DE}_{\mathrm{CCA}},\mathcal{A}}$ the oracle RoR is given a block-source $\boldsymbol{M}$, samples $(m_1, \ldots, m_T) \leftarrow \boldsymbol{M}$, and outputs the challenge ciphertext $\left(\mathsf{Enc}_{pk}(m_1), \ldots, \mathsf{Enc}_{pk}(m_T)\right)$. In the experiment $\mathsf{Expt}^{(i)}$, the oracle RoR on input a block-source $\boldsymbol{M}$,

samples $(m_1, \ldots, m_T) \leftarrow \boldsymbol{M}$ and $(u_1, \ldots, u_T) \leftarrow (\{0,1\}^n)^T$, and outputs the challenge ciphertext $\left(\mathsf{Enc}_{pk}(m_1), \ldots, \mathsf{Enc}_{pk}(m_{T-i}), \mathsf{Enc}_{pk}(u_{T-i+1}), \mathsf{Enc}_{pk}(u_T)\right)$. That is, the first $T-i$ challenge messages are sampled according to $\boldsymbol{M}$, and the remaining messages are sampled independently and uniformly at random. Then, observe that $\mathsf{Expt}^{(0)} = \mathsf{Expt}^{\mathsf{realCCA}}_{\mathcal{DE}_{\mathrm{CCA}},\mathcal{A}}$ and $\mathsf{Expt}^{(T)} = \mathsf{Expt}^{\mathsf{randCCA}}_{\mathcal{DE}_{\mathrm{CCA}},\mathcal{A}}$. Therefore, it suffices to prove that for every $i \in \{0, \ldots, T-1\}$ the expression

$$\left|\Pr\left[\mathsf{Expt}^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(i+1)}(\lambda) = 1\right]\right| \tag{7.1}$$

is negligible in the security parameter λ. For the remainder of the proof we fix the value of i and focus on the experiments $\mathsf{Expt}^{(i)}$ and $\mathsf{Expt}^{(i+1)}$. We denote by $\mathsf{RoR}(i, pk, \cdot)$ and $\mathsf{RoR}(i+1, pk, \cdot)$ the encryption oracles of these two experiments, respectively, and observe that the only difference between them is the distribution of the challenge message m_{T-i}.

In what follows, for each $j \in \{i, i+1\}$ we describe seven experiments, $\mathsf{Expt}^{(j)}_0, \ldots, \mathsf{Expt}^{(j+1)}_6$, and derive a series of claims relating them. We then combine these claims to bound the expression in Eq. (7.1).

Experiment $\mathsf{Expt}^{(j)}_0$ This experiment is the experiment $\mathsf{Expt}^{(j)}$ as defined above.

Experiment $\mathsf{Expt}^{(j)}_1$ This experiment is obtained from $\mathsf{Expt}^{(j)}_0$ by outputting an independently and uniformly sampled bit whenever the $(T-i)$th challenge message and the messages corresponding to the decryption queries $c^{(1)}, \ldots, c^{(q)}$ define a "bad" sequence of inputs for the admissible hash function h (recall the efficiently recognizable set $\mathsf{Unlikely}_h$ from Definition 2.3).

Formally, let $x^* = \pi_1(m_{T-i})$ for $j = i$ and let $x^* = \pi_1(u_{T-i})$ for $j = i+1$. In addition, for any $\zeta \in [q]$, if $\mathsf{Dec}_{sk}(c^{(\zeta)}) \neq \perp$ then let $x_\zeta = \pi_1\left(\mathsf{Dec}_{sk}\left(c^{(\zeta)}\right)\right)$, and if $\mathsf{Dec}_{sk}\left(c^{(\zeta)}\right) = \perp$ then let x_ζ be an arbitrary value that is different from $x^*, x_1, \ldots, x_{\zeta-1}$. The experiment $\mathsf{Expt}^{(j)}_1$ is defined by running $\mathsf{Expt}^{(j)}_0$, and then outputting either an independently and uniformly sampled bit if $(x^*, x_1, \ldots, x_q) \in \mathsf{Unlikely}_h$, or the output of $\mathsf{Expt}^{(j)}_0$ if $(x^*, x_1, \ldots, x_q) \notin \mathsf{Unlikely}_h$.

Claim 7.2. *For each $j \in \{i, i+1\}$, it holds that*

$$\left|\Pr\left[\mathsf{Expt}^{(j)}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(j)}_1(\lambda) = 1\right]\right| \leq \mathrm{negl}(\lambda).$$

Proof. By the definition of admissible hash functions (see Definition 2.3), the probability that $(x^*, x_1, \ldots, x_q) \in \mathsf{Unlikely}_h$ is some negligible function $\nu(\lambda)$. Let $\mathtt{Bad}^{(j)}$ denote the event in which $(x^*, x_1, \ldots, x_q) \in \mathsf{Unlikely}_h$ in the experiment $\mathsf{Expt}^{(j)}_0$, then

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Expt}^{(j)}_1(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(j)}_0(\lambda) = 1\right]\right| \\
&\quad \leq \Pr\left[\neg\mathtt{Bad}^{(j)}\right] \\
&\qquad \cdot \left|\Pr\left[\mathsf{Expt}^{(j)}_1(\lambda) = 1 \,\middle|\, \neg\mathtt{Bad}^{(j)}\right] - \Pr\left[\mathsf{Expt}^{(j)}_0(\lambda) \,\middle|\, \neg\mathtt{Bad}^{(j)}\right] = 1\right|
\end{aligned}$$

$$
\begin{aligned}
&+ \Pr\left[\mathtt{Bad}^{(j)}\right] \cdot \left|\Pr\left[\mathsf{Expt}_1^{(j)}(\lambda) = 1 \,\middle|\, \mathtt{Bad}\right] - \Pr\left[\mathsf{Expt}_0^{(j)}(\lambda) \,\middle|\, \mathtt{Bad}^{(j)}\right] = 1\right| \\
&= \Pr\left[\neg\mathtt{Bad}^{(j)}\right] \cdot 0 + \Pr[\mathtt{Bad}] \cdot \left|\frac{1}{2} - \Pr\left[\mathsf{Expt}_0^{(j)}(\lambda) = 1\right]\right| \\
&\leq \frac{\Pr\left[\mathtt{Bad}^{(j)}\right]}{2} \\
&= \frac{\nu(\lambda)}{2},
\end{aligned}
$$

which is negligible as required. □

Experiment $\mathsf{Expt}_2^{(j)}$ This experiment is obtained from $\mathsf{Expt}_1^{(j)}$ by outputting the output of $\mathsf{Expt}_1^{(j)}$ with probability $1/\Delta$, and outputting an independent and uniform bit with probability $1 - 1/\Delta$, where $\Delta = \Delta(\lambda)$ is the polynomial corresponding to q from the definition of admissible hash functions (see Definition 2.3). The following claim follows in a straightforward manner.

Claim 7.3. *It holds that*

$$
\begin{aligned}
&\left|\Pr\left[\mathsf{Expt}_2^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2^{(i+1)}(\lambda) = 1\right]\right| = \frac{1}{\Delta} \\
&\cdot \left|\Pr\left[\mathsf{Expt}_1^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1^{(i+1)}(\lambda) = 1\right]\right|.
\end{aligned}
$$

Proof. For each $j \in \{i, i+1\}$ it holds that

$$
\Pr\left[\mathsf{Expt}_2^{(j)}(\lambda) = 1\right] = \frac{1}{\Delta} \cdot \Pr\left[\mathsf{Expt}_1^{(j)}(\lambda) = 1\right] + \left(1 - \frac{1}{\Delta}\right) \cdot \frac{1}{2}.
$$

□

Now, from the triangle inequality and Claim 7.2, we have the following series of inequalities.

$$
\begin{aligned}
&\left|\Pr\left[\mathsf{Expt}_0^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_0^{(i+1)}(\lambda) = 1\right]\right| \\
&\leq \left|\Pr\left[\mathsf{Expt}_0^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1^{(i)}(\lambda) = 1\right]\right| \\
&\quad + \left|\Pr\left[\mathsf{Expt}_1^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1^{(i+1)}(\lambda) = 1\right]\right| \\
&\quad + \left|\Pr\left[\mathsf{Expt}_1^{(i+1)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_0^{(i+1)}(\lambda) = 1\right]\right| \\
&\leq \left|\Pr\left[\mathsf{Expt}_0^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1^{(i)}(\lambda) = 1\right]\right| \\
&\quad + \Delta \cdot \left|\Pr\left[\mathsf{Expt}_2^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2^{(i+1)}(\lambda) = 1\right]\right| \\
&\quad + \left|\Pr\left[\mathsf{Expt}_1^{(i+1)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_0^{(i+1)}(\lambda) = 1\right]\right|,
\end{aligned}
$$

leading to the following corollary.

Corollary 7.4. *It holds that*

$$\left|\Pr\left[\mathsf{Expt}_0^{(i)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_0^{(i+1)}(\lambda)=1\right]\right| \leq \Delta\cdot\left|\Pr\left[\mathsf{Expt}_2^{(i)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_2^{(i+1)}(\lambda)=1\right]\right|+\mathrm{negl}(\lambda).$$

Experiment $\mathsf{Expt}_3^{(j)}$ This experiment is obtained from $\mathsf{Expt}_2^{(j)}$ by changing the abort condition. Specifically, at the end of experiment $\mathsf{Expt}_2^{(j)}$, we sample an independent initialization value K' (in addition to K that is used by the key-generation algorithm), and denote by $\mathsf{Partition}_{K',h}^{(j)}$ the event in which $P_{K'}(h(x^*))=\texttt{Lossy}$ and $P_{K'}(h(x_i))=\texttt{Inj}$ for any $\zeta\in[q]$ such that $\mathsf{Dec}_{sk}\left(c^{(\zeta)}\right)\neq\perp$, where $P_{K'}:\{0,1\}^v\to\{\texttt{Lossy},\texttt{Inj}\}$ is the partitioning function of the admissible hash function (recall that the values $x^*, x_1, \ldots, x_q$ were defined in $\mathsf{Expt}_1^{(j)}$).

We would like to replace the abort condition from experiment $\mathsf{Expt}_2^{(j)}$ (which is independent of the adversary's view) with one that depends on the event $\mathsf{Partition}_{K',h}^{(j)}$. Unfortunately, all we are guaranteed is that the event $\mathsf{Partition}_{K',h}^{(j)}$ occurs with probability that is *at least* $1/\Delta$ (assuming that $(x^*,x_1,\ldots,x_q)\notin\mathsf{Unlikely}_h$). Therefore, if $(x^*,x_1,\ldots,x_q)\notin\mathsf{Unlikely}_h$, we first approximate the value

$$p^{(j)}=\Pr_{K'\leftarrow\mathcal{K}_\lambda}\left[\mathsf{Partition}_{K',h}^{(j)}\mid(h(x^*),h(x_1),\ldots,h(x_q))\right]$$

by sampling a sufficient number of independent initialization keys $K''\leftarrow\mathcal{K}_\lambda$ and observing whether or not the event $\mathsf{Partition}_{K'',h}^{(j)}$ occurs (with respect to the fixed values $h(x^*), h(x_1), \ldots, h(x_q)$). For any polynomial S, Hoeffding's inequality yields that with $\lceil\lambda S\cdot\Delta\rceil$ samples we can obtain an approximation $\tilde{p}^{(j)}\geq(1/\Delta)$ of $p^{(j)}$ such that

$$\Pr\left[\left|p^{(j)}-\tilde{p}^{(j)}\right|\geq\frac{1}{\Delta\cdot S}\right]\leq\frac{1}{2^\lambda}. \tag{7.2}$$

Then, looking all the way back to experiment $\mathsf{Expt}_1^{(j)}$, the output of $\mathsf{Expt}_3^{(j)}$ is computed as follows:

1. If $(x^*,x_1,\ldots,x_q)\in\mathsf{Unlikely}_h$ or if the event $\mathsf{Partition}_{K',h}^{(j)}$ does not occur, then we output the output of $\mathsf{Expt}_1^{(j)}$.
2. If $(x^*,x_1,\ldots,x_q)\notin\mathsf{Unlikely}_h$ and the event $\mathsf{Partition}_{K',h}^{(j)}$ does occur, then we output the output of $\mathsf{Expt}_1^{(j)}$ with probability $1/(\Delta\tilde{p}^{(j)})$, and we "artificially" enforce an abort and output an independent and uniform bit with probability $1-1/(\Delta\tilde{p}^{(j)})$.

Claim 7.5. *For each $j\in\{i,i+1\}$ and for any polynomial $S=S(\lambda)$ it holds that*

$$\left|\Pr\left[\mathsf{Expt}_2^{(j)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_3^{(j)}(\lambda)=1\right]\right|\leq\frac{1}{\Delta S}+\frac{1}{2^\lambda}.$$

Proof. Denote by $\neg\mathtt{Abort}_2^{(j)}$ and $\neg\mathtt{Abort}_3^{(j)}$ the events in which the experiments $\mathsf{Expt}_2^{(j)}$ and $\mathsf{Expt}_3^{(j)}$ output the output of $\mathsf{Expt}_1^{(j)}$, respectively. Then,

$$\Pr\left[\neg\mathtt{Abort}_2^{(j)}\right] = \frac{1}{\Delta} \text{ and } \Pr\left[\neg\mathtt{Abort}_3^{(j)}\right] = p^{(j)} \cdot \frac{1}{\Delta \tilde{p}^{(j)}} = \frac{1}{\Delta} \cdot \frac{p^{(j)}}{\tilde{p}^{(j)}}.$$

Equation (7.2) implies that with probability at least $1 - 2^{-\lambda}$ it holds that

$$\left|\Pr\left[\neg\mathtt{Abort}_2^{(j)}\right] - \Pr\left[\neg\mathtt{Abort}_3^{(j)}\right]\right| = \frac{1}{\Delta} \cdot \left|\frac{\tilde{p}^{(j)} - p^{(j)}}{\tilde{p}^{(j)}}\right| \leq \frac{1}{\Delta^2 S \tilde{p}^{(j)}} \leq \frac{1}{\Delta S}. \tag{7.3}$$

As (7.3) holds for any $(x^*, x_1, \ldots, x_q)$ with probability at least $1 - 2^{-\lambda}$, we obtain that the statistical distance between the outputs of experiments $\mathsf{Expt}_2^{(j)}$ and $\mathsf{Expt}_3^{(j)}$ is at most $1/(\Delta S) + 2^{-\lambda}$. □

Now, from the triangle inequality and Claim 7.5, we get

$$\begin{aligned}
&\Delta \cdot \left|\Pr\left[\mathsf{Expt}_2^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2^{(i+1)}(\lambda) = 1\right]\right| \\
&\quad \leq \Delta \cdot \left|\Pr\left[\mathsf{Expt}_2^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3^{(i)}(\lambda) = 1\right]\right| \\
&\qquad + \Delta \cdot \left|\Pr\left[\mathsf{Expt}_3^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3^{(i+1)}(\lambda) = 1\right]\right| \\
&\qquad + \Delta \cdot \left|\Pr\left[\mathsf{Expt}_3^{(i+1)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2^{(i+1)}(\lambda) = 1\right]\right|.
\end{aligned}$$

This gives us the following corollary.

Corollary 7.6. *For any polynomial $S = S(\lambda)$ it holds that*

$$\begin{aligned}
&\Delta \cdot \left|\Pr\left[\mathsf{Expt}_2^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2^{(i+1)}(\lambda) = 1\right]\right| \\
&\quad \leq 2 \cdot \left(\frac{1}{S} + \frac{\Delta}{2^\lambda}\right) + \Delta \cdot \left|\Pr\left[\mathsf{Expt}_3^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3^{(i+1)}(\lambda) = 1\right]\right|.
\end{aligned}$$

Experiment $\mathsf{Expt}_4^{(j)}$ This experiment is obtained from $\mathsf{Expt}_3^{(j)}$ by replacing the event $\mathsf{Partition}_{K',h}$ with the event $\mathsf{Partition}_{K,h}$. That is, we do not sample a new initialization value K' for the partitioning, but rather consider the partition defined by the initialization value K used by the key-generation algorithm.

Claim 7.7. *For each $j \in \{i, i+1\}$ it holds that*

$$\left|\Pr\left[\mathsf{Expt}_3^{(j)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_4^{(j)}(\lambda) = 1\right]\right| \leq \mathrm{negl}(\lambda).$$

Proof. We observe that any adversary $\mathcal{A}$ for which the above difference is non-negligible can be used to distinguish initialization values of the $\mathcal{R}$-lossy trapdoor function family.

The distinguisher first chooses two keys $K, K' \leftarrow \mathcal{K}$ independently and uniformly at random. Then, upon receiving σ the public index sampled from one of the two ensembles $\{\sigma : (\sigma, \tau) \leftarrow \mathsf{Gen}_{\mathrm{BM}}(1^\lambda, K_\lambda)\}_{\lambda\in\mathbb{N}}$ or $\{\sigma : (\sigma, \tau) \leftarrow \mathsf{Gen}_{\mathrm{BM}}(1^\lambda, K'_\lambda)\}_{\lambda\in\mathbb{N}}$, the distinguisher proceeds to efficiently simulate $\mathcal{A}$ as follows: Sample two permutations and a lossy trapdoor function as in $\mathsf{Expt}_3^{(j)}$ but use $\sigma_g = \sigma$ (one of the two possible function indices returned by the $\mathcal{R}$-lossy challenge) to setup the public key pk. Then proceed to simulate $\mathsf{Expt}_3^{(j)}$ with the initialization value K.

If σ was sampled from the ensemble corresponding to K' then the adversary participates exactly in $\mathsf{Expt}_3^{(j)}$. However, if σ was sampled from the ensemble corresponding to K then the simulation proceeds exactly as in $\mathsf{Expt}_4^{(j)}$. □

Corollary 7.8. *It holds that*

$$\left|\Pr\left[\mathsf{Expt}_3^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3^{(i+1)}(\lambda) = 1\right]\right| \leq \left|\Pr\left[\mathsf{Expt}_4^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_4^{(i+1)}(\lambda) = 1\right]\right| + \mathrm{negl}(\lambda).$$

Experiment $\mathsf{Expt}_5^{(j)}$ This experiment is obtained from $\mathsf{Expt}_4^{(j)}$ by not taking into account the event $(x^*, x_1, \ldots, x_q) \in \mathsf{Unlikely}_h$ when computing the output of the experiment. Looking all the way back to experiment $\mathsf{Expt}_0^{(j)}$, the output of $\mathsf{Expt}_4^{(j)}$ is computed as follows:

1. If the event $\mathsf{Partition}_{K,h}^{(j)}$ does not occur, then we output an independent uniform bit.
2. If the event $\mathsf{Partition}_{K,h}^{(j)}$ does occur, then we output the output of $\mathsf{Expt}_0^{(j)}$ with probability $1/(\Delta\tilde{p}^{(j)})$, and we "artificially" enforce an abort and output an independent and uniform bit with probability $1 - 1/(\Delta\tilde{p}^{(j)})$.

Note that the event $(x^*, x_1, \ldots, x_q) \in \mathsf{Unlikely}_h$ has the same probability in the experiments $\mathsf{Expt}_4^{(j)}$ and $\mathsf{Expt}_5^{(j)}$, and that this probability is upper bounded by some negligible function $\nu(n)$ (see Claim 7.2). Therefore, for each $j \in \{i, i+1\}$ we have that $\left|\Pr\left[\mathsf{Expt}_4^{(j)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_5^{(j)}(\lambda) = 1\right]\right| \leq \mathrm{negl}(\lambda)$, implying the following corollary

Corollary 7.9. *It holds that*

$$\left|\Pr\left[\mathsf{Expt}_4^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_4^{(i+1)}(\lambda) = 1\right]\right| \leq \left|\Pr\left[\mathsf{Expt}_5^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_5^{(i+1)}(\lambda) = 1\right]\right| + \mathrm{negl}(\lambda).$$

Looking ahead, the modification of ignoring the (negligible probability) event $(x^*, x_1, \ldots, x_q) \in \mathsf{Unlikely}_h$ ensures that the abort conditions of experiments $\mathsf{Expt}_5^{(i)}$ and $\mathsf{Expt}_5^{(i+1)}$ are computed in an identical manner given K, h, and the challenge ciphertexts. Previously, the abort condition relied on x^* (which was defined as $\pi_1(m_{T-i})$ for $j = i$ and as $\pi_1(u_{T-i})$ for $j = i+1$), and now it relies on $h(x^*)$ which is given as part of

the challenge ciphertext (therefore, given the challenge ciphertexts, the abort condition is now completely independent of whether $j = i$ or $j = i + 1$).

Experiment $\mathsf{Expt}_6^{(j)}$ This experiment is obtained from $\mathsf{Expt}_5^{(j)}$ by changing the decryption oracle to decrypt using the trapdoor τ_g of the $\mathcal{R}$-lossy trapdoor function, instead of using the trapdoor τ_f of the lossy trapdoor function. Specifically, we define the oracle $\widetilde{\mathsf{Dec}}(sk, \cdot)$ that on input the ith decryption query $c^{(i)} = \left(c_h^{(i)}, c_f^{(i)}, c_g^{(i)}\right)$ computes $m = \pi_2^{-1}\left(\mathsf{G}^{-1}\left(\tau_g, c_g^{(i)}\right)\right)$, and checks whether the ciphertext components are well-formed. Note, however, that for a decryption query $c^{(i)}$ that corresponds to a lossy tag it is impossible to (efficiently) decrypt using τ_g. In this case the decryption oracle outputs $\perp$, and the output of the experiment is an independent and uniform bit.

Claim 7.10. *For each $j \in \{i, i+1\}$, we have* $\Pr\left[\mathsf{Expt}_5^{(j)}(\lambda) = 1\right] = \Pr\left[\mathsf{Expt}_6^{(j)}(\lambda) = 1\right]$.

Proof. Note that whenever the event $\mathsf{Partition}_{K,h}^{(j)}$ occurs then in particular all decryption queries which are well-formed correspond to injective tags and therefore can be decrypted using τ_g. Thus, conditioned on the event $\mathsf{Partition}_{K,h}^{(j)}$ (which as the exact same probability in $\mathsf{Expt}_5^{(j)}$ and $\mathsf{Expt}_6^{(j)}$) the oracles Dec and $\widetilde{\mathsf{Dec}}$ are identical from which the claim follows. □

Corollary 7.11. *It holds that*

$$\left|\Pr\left[\mathsf{Expt}_5^{(i)} = 1\right] - \Pr\left[\mathsf{Expt}_5^{(i+1)} = 1\right]\right| = \left|\Pr\left[\mathsf{Expt}_6^{(i)} = 1\right] - \Pr\left[\mathsf{Expt}_6^{(i+1)} = 1\right]\right|.$$

Experiment $\mathsf{Expt}_7^{(j)}$ This experiment is obtained from $\mathsf{Expt}_6^{(j)}$ by sampling the public key as follows: instead of an injective function σ_f, sample a lossy function $\tilde{\sigma}_f$. The rest of the experiment is identical to $\mathsf{Expt}_6^{(j)}$.

Claim 7.12. *For each $j \in \{i, i+1\}$ it holds that*

$$\left|\Pr\left[\mathsf{Expt}_6^{(j)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_7^{(j)}(\lambda) = 1\right]\right| \leq \mathrm{negl}(\lambda).$$

Proof. Observe that as σ_f is no longer used by the decryption oracle, and thus, replacing σ_f with $\tilde{\sigma}_f$ does not affect decryption queries. Therefore, any efficient adversary for which the claim is false can be used to distinguish a randomly sampled injective function σ_f from a randomly sampled lossy function $\tilde{\sigma}_f$. □

As a corollary, we get

Corollary 7.13. *It holds that*

$$\left|\Pr\left[\mathsf{Expt}_6^{(i)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_6^{(i+1)}(\lambda)=1\right]\right| \leq \left|\Pr\left[\mathsf{Expt}_7^{(i)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_7^{(i+1)}(\lambda)=1\right]\right|+\mathrm{negl}(\lambda).$$

The final claim we require is as follows.

Claim 7.14. *It holds that*

$$\left|\Pr\left[\mathsf{Expt}_7^{(i)}(\lambda)=1\right]-\Pr\left[\mathsf{Expt}_7^{(i+1)}(\lambda)=1\right]\right| \leq \mathrm{negl}(\lambda).$$

Proof. We prove the claim by upper bounding the statistical distance between the output distributions of $\mathsf{Expt}_7^{(i)}$ and $\mathsf{Expt}_7^{(i+1)}$. We observe that these output distributions can be computed by applying the *exact same* stochastic (and, very likely, inefficient) map to the joint distribution of the public key $\widetilde{pk}$ and the challenge ciphertext $\boldsymbol{c}^*$ in each experiment. The difference between the resulting distributions will follow from the difference between the challenge ciphertexts: In $\mathsf{Expt}_7^{(i)}$ the $(T-i)$th challenge message is m_{T-i}, whereas in $\mathsf{Expt}_7^{(i+1)}$ it is a uniform message u_{T-i}. This follows since, as discussed above, the modification of ignoring the (negligible probability) event $(x^*, x_1, \ldots, x_q) \in \mathsf{Unlikely}_h$ ensures that the abort conditions of experiments $\mathsf{Expt}_5^{(i)}$ and $\mathsf{Expt}_5^{(i+1)}$ are computed in an identical manner given K, h, and the challenge ciphertexts (and this continued to hold in remaining experiments). Previously, the abort condition relied on x^* (which was defined as $\pi_1(m_{T-i})$ for $j=i$ and as $\pi_1(u_{T-i})$ for $j=i+1$), and now it relies on $h(x^*)$ which is given as part of the challenge ciphertext (therefore, given the challenge ciphertexts, the abort condition is now completely independent of whether $j=i$ or $j=i+1$).

Therefore, it suffices to consider the statistical distance between the distribution $(\widetilde{pk}, \boldsymbol{c}^*)$ in the experiment $\mathsf{Expt}_7^{(i)}$ and the same distribution in the experiment $\mathsf{Expt}_7^{(i+1)}$ (since applying the same stochastic map to a pair of distributions cannot increase the statistical distance between them). Moreover, we prove that this statistical distance is negligible in the security parameter even when fixing all components of the public key $\widetilde{pk}$ other than the two permutations π_1 and π_2. Specifically, we prove that for any set $\mathcal{X}$ of at most 2^p (T,k)-block-sources, with an overwhelming probability over the choice of π_1 and π_2, for any $\boldsymbol{M} \in \mathcal{X}$, the distribution of the challenge ciphertext $\boldsymbol{c}^*$ resulting from $\boldsymbol{M}$ in $\mathsf{Expt}_7^{(i)}$ and the distribution of the challenge ciphertext $\boldsymbol{c}^*$ resulting from $\boldsymbol{M}$ in $\mathsf{Expt}_7^{(i+1)}$ lead these two experiments to statistically close outputs.

Recall that the challenge ciphertexts for $\mathsf{Expt}_7^{(j)}$ are of the form $\boldsymbol{c}^* = (c_1^*, \ldots, c_T^*)$ where the components $c_1^*, \ldots, c_{T-i-1}^*$ and $c_{T-i+1}^*, \ldots, c_T^*$ are identically distributed for $j \in \{i, i+1\}$. Moreover, in both experiments the components $c_{T-i+1}^*, \ldots, c_T^*$ are encryptions of independent and uniformly distributed messages. Therefore, it suffices to consider the distribution of c_{T-i}^* conditioned on $c_1^*, \ldots, c_{T-i-1}^*$ in each experiment. Recall from our definitions that,

$$c^*_{T-i} = \begin{cases} (c^*_h, c^*_f, c^*_g) \stackrel{\text{def}}{=} \Big(h(\pi_1(m_{T-i})), \mathsf{F}\big(\tilde{\sigma}_f, \pi_2(m_{T-i})\big), \mathsf{G}\big(\sigma_g, h(\pi_1(m_{T-i})), \pi_2(m_{T-i})\big)\Big) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } j = i, \\ (u^*_h, u^*_f, u^*_g) \stackrel{\text{def}}{=} \Big(h(\pi_1(u_{T-i})), \mathsf{F}\big(\tilde{\sigma}_f, \pi_2(u_{T-i})\big), \mathsf{G}\big(\sigma_g, h(\pi_1(u_{T-i})), \pi_2(u_{T-i})\big)\Big) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } j = i+1. \end{cases}$$

Denote by C^*_h, C^*_f, and C^*_g the random variables corresponding to c^*_h, c^*_f, and c^*_g, respectively, and similarly U^*_h, U^*_f, U^*_g corresponding to u^*_h, u^*_f, u^*_g where the probability is taken over the choice of π_1, π_2, m_{T-i}, and u_{T-i}. In what follows, we fix $m_1, \ldots, m_{T-i-1}$, and argue that the two distributions (C^*_h, C^*_f, C^*_g) and (U^*_h, U^*_f, U^*_g) conditioned on the first $T-i-1$ challenge messages $m_1, \ldots, m_{T-i-1}$ are statistically close.

We begin by focusing on the distributions $C^*_h = h(\pi_1(M_{T-i}))$ and $U^*_h = h(\pi_1(U_{T-i})$. Observe that $h : \{0,1\}^* \to \{0,1\}^v$ is an $(n, n-v)$-lossy function, and let Z denote the indicator of the event in which $M_1 = m_1, \ldots, M_{T-i-1} = m_{T-i-1}$. Consider the set $\mathcal{Z}$ defined as the set of distributions $M_{T-i}|_{Z=1}$ for all $\boldsymbol{M} = (M_1, \ldots, M_{T-i}, \ldots, M_T) \in \mathcal{X}$ and for all possible values of $m_1, \ldots, m_{T-i-1}$. Then, we have,

$$|\mathcal{Z}| \leq |\mathcal{X}| \cdot 2^{(T-i-1)n} \leq |\mathcal{X}| \cdot 2^{(T-1)n} \leq 2^{p+n(T-1)}.$$

Applying Theorem 4.6 (for $T = 1$) with our choice of parameters implies that with an overwhelming probability over the choice of $\pi_1 \leftarrow \Pi_1$ for any such M_{T-i} we have

$$\mathbf{SD}\left(h(\pi_1(M_{T-i}))|_{Z=1}, h(\pi_1(U_{T-i})|_{Z=1}\right) \leq 2^{-\omega(\log\lambda)}. \tag{7.4}$$

We now fix any $\pi_1 \in \Pi_1$ for which (7.4) holds. Consider now any possible value α_h that the random variables $h(\pi_1(M_{T-i}))$ and $h(\pi_1(U_{T-i})$ may obtain in the experiments $\mathsf{Expt}^{(i)}_7$ and $\mathsf{Expt}^{(i+1)}_7$, respectively. If α_h corresponds to an injective tag for G, then in particular the event $\mathsf{Partition}_{K,h}$ will not occur in either one of the experiments, and thus the output of both experiments is an independent and uniform bit. Moreover, (7.4) above implies that the probabilities of having an α_h that corresponds to injective tag in $\mathsf{Expt}^{(i)}_7$ and $\mathsf{Expt}^{(i+1)}_7$ are negligibly close. Therefore, it remains to show that for all but a negligible probability of the α_h's that correspond to lossy tags for G, the distributions $(C^*_f, C^*_g)|_{C^*_h=\alpha_h}$ and $(U^*_f, U^*_g)|_{U^*_h=\alpha_h}$ are statistically close (once again, conditioned on the first $T-i-1$ challenge messages $m_1, \ldots, m_{T-i-1}$ as before).

The following straightforward claim shows that the message distributions of the $(T-i)$th challenge message in $\mathsf{Expt}^{(i)}_7$ and $\mathsf{Expt}^{(i+1)}_7$ (denoted M_{T-i} and U_{T-i}, respectively), have sufficient entropy even when conditioned on α_h. □

Claim 7.15. *For any $\epsilon > 0$, with probability at least $1-\epsilon$ over the choice of $\alpha_h \leftarrow C^*_h$ conditioned on $P_K(\alpha_h) = \texttt{Lossy}$, it holds that*

$$\mathbf{H}_\infty\big(M_{T-i} \,\big|\, C^*_h = \alpha_h, M_1 = m_1, \ldots, M_{T-i-1} = m_{T-i-1}\big) \geq k - v - \log(1/\epsilon).$$

Similarly, for any $\epsilon > 0$, with probability at least $1 - \epsilon$ over the choice of $\alpha_h \leftarrow U_h^*$ conditioned on $P_K(\alpha_h) = \texttt{Lossy}$, it holds that

$$\mathbf{H}_\infty\left(U_{T-i} \,\middle|\, U_h^* = \alpha_h, M_1 = m_1, \ldots, M_{T-i-1} = m_{T-i-1}\right) \geq n - v - \log(1/\epsilon).$$

Proof. As the output length of h is v bits, the claim follows from applying Lemma 2.1 to the distribution $M_{T-i}|_{M_1=m_1,\ldots,M_{T-i-1}=m_{T-i-1}}$ (recall that $\boldsymbol{M}$ is a (T,k)-block-source) and to the uniform distribution U_{T-i}. □

Fix some $\epsilon = \omega(\log\lambda)$ and any α_h for which both parts of Claim 7.15 hold, and let $k' = k - v - \log(1/\epsilon)$. Then, since $P_K(\alpha_h) = \texttt{Lossy}$, we have that α_h corresponds to a lossy tag for G, and therefore for the function $f_h : \{0,1\}^n \to \{0,1\}^{n'}$ defined as $f_h(\cdot) = (\mathsf{F}(\tilde{\sigma}_f, \cdot), \mathsf{G}(\sigma_g, c_h^*, \cdot))$ it holds that $|\mathrm{Im}(f_h)| \leq 2^{2n-2\ell}$. Let Y denote the indicator of the event in which $C_h^* = \alpha_h, M_1 = m_1, \ldots, M_{T-i-1} = m_{T-i-1}$, and consider the set $\mathcal{Y}$ defined as the set of distributions $M_{T-i}|_{Y=1}$ for all $\boldsymbol{M} = (M_1, \ldots, M_{T-i}, \ldots, M_T) \in \mathcal{X}$ and for all possible values of $\alpha_h, m_1, \ldots, m_{T-i-1}$. Then, we have,

$$|\mathcal{Y}| \leq |\mathcal{X}| \cdot 2^v \cdot 2^{(T-i-1)n} \leq |\mathcal{X}| \cdot 2^{v+(T-1)n} \leq 2^{p+v+n(T-1)}.$$

Now, applying Theorem 4.6 (setting $T = 1$) with our choice of parameters implies that with an overwhelming probability over the choice of π_2, for any such M_{T-i} and Y we have

$$\mathbf{SD}\left(f_h(\pi_2(M_{T-i}))|_{Y=1},\, f_h(U_n)\right) \leq 2^{-\omega(\log\lambda)}.$$

An essentially identical argument holds for U_{T-i}, and from this it follows that

$$\mathbf{SD}\left(\left(C_f^*, C_g^*\right)\Big|_{C_h^*=\alpha_h}, \left(U_f^*, U_g^*\right)\Big|_{U_h^*=\alpha_h}\right) \leq \mathrm{negl}(\lambda) \tag{7.5}$$

for all but a negligible probability of the α_h's that correspond to lossy tags for G, as required. □

Completing the proof of Theorem 7.1 To complete the proof of the theorem, recollect that it suffices to bound the expression in (7.1). For any polynomial $S = S(\lambda)$, collecting negligible terms $\mathrm{negl}(\lambda)$, we have

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Expt}^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(i+1)}(\lambda) = 1\right]\right| \\
&\quad \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}_0^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_0^{(i+1)}(\lambda) = 1\right]\right| \\
&\quad \leq \Delta \cdot \left|\Pr\left[\mathsf{Expt}_2^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2^{(i+1)}(\lambda) = 1\right]\right| + \mathrm{negl}(\lambda) \quad \text{(from Cor. 7.4)} \\
&\quad \leq 2\left(\frac{1}{S} + \frac{\Delta}{2^\lambda}\right) + \Delta
\end{aligned}$$

$$
\begin{aligned}
&\cdot \left| \Pr\left[\mathsf{Expt}_3^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3^{(i+1)}(\lambda) = 1\right] \right| + \mathrm{negl}(\lambda) \quad \text{(from Cor. 7.6)} \\
&\leq \Delta \cdot \left| \Pr\left[\mathsf{Expt}_4^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_4^{(i+1)}(\lambda) = 1\right] \right| \\
&\quad + 2\left(\frac{1}{S} + \frac{\Delta}{2^\lambda}\right) + \mathrm{negl}(\lambda) \quad \text{(from Cor. 7.8)} \\
&\leq \Delta \cdot \left| \Pr\left[\mathsf{Expt}_5^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_5^{(i+1)}(\lambda) = 1\right] \right| \\
&\quad + 2\left(\frac{1}{S} + \frac{\Delta}{2^\lambda}\right) + \mathrm{negl}(\lambda) \quad \text{(from Cor. 7.9)} \\
&= \Delta \cdot \left| \Pr\left[\mathsf{Expt}_6^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_6^{(i+1)}(\lambda) = 1\right] \right| \\
&\quad + 2\left(\frac{1}{S} + \frac{\Delta}{2^\lambda}\right) + \mathrm{negl}(\lambda) \quad \text{(from Cor. 7.11)} \\
&\leq \Delta \cdot \left| \Pr\left[\mathsf{Expt}_7^{(i)}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_7^{(i+1)}(\lambda) = 1\right] \right| \\
&\quad + 2\left(\frac{1}{S} + \frac{\Delta}{2^\lambda}\right) + \mathrm{negl}(\lambda) \quad \text{(from Cor. 7.13)} \\
&\leq 2\left(\frac{1}{S} + \frac{\Delta}{2^\lambda}\right) + \mathrm{negl}(\lambda). \quad \text{(from Claim7.14)}
\end{aligned}
$$

As $\Delta = \Delta(\lambda)$ is some fixed polynomial, and the above holds for any polynomial $S = S(\lambda)$, this completes the proof of Theorem 7.1. □

8. Generic Constructions in the Random Oracle Model

In this section we present two generic constructions in the random oracle model based on any (randomized) public-key encryption scheme. In our first construction (Sect. 8.1), given any public-key encryption scheme, we modify its encryption algorithm Enc into a deterministic one Enc' as follows: Given a public key pk and a message m, the encryption algorithm Enc' first computes $r_m = H(m\|u)$, where H is a hash function modeled as a random oracle, and u is a uniformly chosen string of length roughly p bits that is part of the public key of the deterministic scheme. The encryption algorithm then outputs the ciphertext $\mathsf{Enc}_{pk}(m; r_m)$. This scheme was originally proposed by Bellare et al. [3], who proved its security with respect to adversarially chosen-plaintext distributions that are *independent* of the public key used by the scheme. We observe that by including in the public key a uniform value u of length roughly p bits, and then using it during the encryption process as described above, we obtain security against 2^p-bounded adversaries. The proof of security goes along the lines of the proof provided by Bellare et al. based on the following observation: For any challenge message m, as long as H is not queried on $m\|u$, either by the adversary $\mathcal{A}$ or by any of its 2^p possible plaintext distributions, then we can rely on the security of the underlying randomized

scheme. The proof of Bellare et al. considered a single plaintext distribution, whereas here we can apply a union bound over all 2^p such distributions due to the additional string u (whose length we set to $\ell = p + \omega(\log \lambda)$ for this purpose).

Our second construction (Sect. 8.2) considers a setting where adversaries adaptively query the real-or-random encryption oracle only with plaintexts distributions that are samplable using at most some predetermined number, $q = q(\lambda)$, of random oracle queries (and we do not require an upper bound, 2^p, on the number of plaintext distributions from which an adversary can choose). In this setting we show that the additive blowup in the length of the public key in our first construction can be avoided. Specifically, given any public-key encryption scheme, we modify its encryption algorithm Enc into a deterministic one Enc' as follows: Given a public key pk and a message m, the encryption algorithm Enc' first computes $r_m = \oplus_{i=1}^{q+1} H(m\|i)$, and then outputs the ciphertext $\mathsf{Enc}_{pk}(m; r_m)$.

8.1. *A Construction Secure Against 2^p-Bounded (T, k)-Source Adversaries*

The scheme Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a (randomized) public-key encryption scheme. We denote by $n = n(\lambda)$ and $\rho = \rho(\lambda)$ the bit-lengths of the messages and random strings that are given as input to the encryption algorithm Enc, respectively. In addition, let $H : \{0,1\}^* \to \{0,1\}^\rho$ be a hash function modeled as a random oracle. Our scheme $\Pi'_p = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ is parameterized by a polynomial $p = p(\lambda)$.

- **Key generation** On input the security parameter 1^λ the key-generation algorithm KeyGen' samples $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $u \leftarrow \{0,1\}^\ell$, where $\ell = \ell(\lambda) = p(\lambda) + \omega(\log \lambda)$. It then outputs $pk' = (pk, u)$ and $sk' = sk$.
- **Encryption** The encryption algorithm Enc' on input a public key $pk' = (pk, u)$ and a message m, computes $r_m = H(m\|u)$, and outputs $c = \mathsf{Enc}_{pk}(m; r_m)$.
- **Decryption** The decryption algorithm Dec' is identical to the underlying decryption algorithm Dec.

Theorem 8.1. *Let Π be a randomized public-key encryption scheme. Then, for any polynomials $p = p(\lambda)$, $T = T(\lambda)$, and for any $k = k(\lambda) = \omega(\log \lambda)$ the following hold:*

1. *If Π is* IND-CPA*-secure then Π'_p is (p, T, k)-ACD-CPA-secure.*
2. *If Π is* IND-CCA*-secure then Π'_p is (p, T, k)-ACD-CCA-secure.*

Proof overview The main idea underlying the proof of security is that, for any challenge message m, as long as H is not queried on $m\|u$, either by the adversary $\mathcal{A}$ or by its adaptively chosen-plaintext distribution $M \in \mathcal{X}$, then the adversary learns essentially no information on m (for simplicity we focus here on security for a single message m with a little abuse of notation, and refer the reader to the formal analysis below for the general case). We divide the set of random oracle queries made by $\mathcal{A}$ and M into queries made in either of the following three phases, and for each of these phases we argue that the query $m\|u$ appears with only a negligible probability:

- H-queries made by $\mathcal{A}$ *before* querying the real-or-random encryption oracle: As $\mathcal{A}$ runs in polynomial time, and $m \leftarrow M$ is sampled with a super-logarithmic min-entropy, then it is unlikely that $\mathcal{A}$ queries H with any input of the form $m\|*$.
- H-queries made by the challenge plaintext distribution M: The randomness, z, used by the real-or-random oracle when sampling from M is independent of u, and thus can be thought of as chosen before u. Therefore, for any set $\mathcal{X}$ of at most 2^p plaintext distributions, and for any $M \in \mathcal{X}$, the probability over the choice of $u \leftarrow \{0,1\}^\ell$ that $M(z)$ queries H with any input of the form $*\|u$ is at most $|M| \cdot 2^{-\ell}$ (where $|M|$ is an upper bound in the number of H-queries made by M). By setting $\ell = p + \omega(\log \lambda)$, a union bound over all 2^p possible such M's implies that with an overwhelming probability no such M queries H on an input of the form $*\|u$.
- H-queries made by $\mathcal{A}$ *after* querying the real-or-random encryption oracle: Assuming that H was not queried with $m\|u$ in either one of the two previous phases, then the value $r_m = H(m\|u)$ is independently and uniformly distributed from the adversary's point of view (subject to producing the challenge ciphertext). Thus, any adversary that queries H on $m\|u$ in this phase for the first time case be used to break the security of the underlying (randomized) encryption scheme.

Proof of Theorem 8.1. Using Theorems 3.4 and 3.7 it suffices to prove that Π'_p is (p, T, k)-ACD1-CPA-secure if Π is IND-CPA-secure and Π'_p is (p, T, k)-ACD1-CCA-secure if Π is IND-CCA-secure. Let $\mathcal{A}$ be a 2^p-bounded (T, k)-source adversary if Π is IND-CPA-secure and a 2^p-bounded (T, k)-source chosen-ciphertext adversary if Π is IND-CCA-secure. Note that in the random oracle model, the adversary $\mathcal{A}$ gets oracle access to H. Additionally, the (T, k)-source $\boldsymbol{M}$ with which $\mathcal{A}$ queries $\mathsf{RoR}(\mathsf{mode}, pk, \cdot)$ is samplable by a probabilistic polynomial-time algorithm that can query the random oracle H. The decryption algorithm $\mathsf{Dec}'_{sk}(\cdot)$ is identical to $\mathsf{Dec}_{sk}(\cdot)$ of the underlying scheme Π and therefore does not query the random oracle H.[11] In what follows, we describe four experiments, $\mathsf{Expt}_0, \ldots, \mathsf{Expt}_3$, and derive a series of claims relating them. We then combine these claims to bound the advantage of the adversary.

For our proof we define a variant $\widetilde{\mathsf{RoR}}$ of the oracle RoR, which uses true randomness for the encryption process instead of the value $r_m = H(m\|u)$. Specifically, on input $(\mathsf{mode}, pk, \boldsymbol{M})$ it samples $(m_1, \ldots, m_T)$ from either $\boldsymbol{M}$ if $\mathsf{mode} = \mathsf{real}$ or U^T if $\mathsf{mode} = \mathsf{rand}$, then samples $r_1, \ldots, r_T \leftarrow (\{0,1\}^\rho)^T$ independently and uniformly at random, and outputs $(\mathsf{Enc}_{pk}(m_1; r_1), \ldots, \mathsf{Enc}_{pk}(m_T; r_T))$.

Experiment Expt_0 This is the experiment $\mathsf{Expt}^{\mathsf{real}}_{\Pi',\mathcal{A}}(\lambda)$ (recall Definition 3.3) if Π is IND-CPA-secure or the experiment $\mathsf{Expt}^{\mathsf{realCCA}}_{\Pi',\mathcal{A}}(\lambda)$ (recall Definition 3.6) if Π is IND-CCA secure.

Experiment Expt_1 This experiment is obtained from Expt_0 by replacing $\mathsf{RoR}(\mathsf{real}, \cdot, \cdot)$ with $\widetilde{\mathsf{RoR}}(\mathsf{real}, \cdot, \cdot)$.

Claim 8.2. $\left|\Pr\left[\mathsf{Expt}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1(\lambda) = 1\right]\right|$ *is negligible in* λ.

[11] We do allow the underlying scheme Π to rely on a random oracle and in this case, we assume that H is independent of the random oracle used by Π (e.g., uses a different prefix).

Proof. Let $(m_1, \ldots, m_T)$ denote the messages sampled from $\boldsymbol{M}$. For every $j \in [T]$ we denote by $\mathtt{Bad}_j$ the event in which H is queried on the point $m_j \| u$. Note that for every $j \in [T]$, as long as the event $\mathtt{Bad}_j$ does not occur, then the value $r_{m_j} = H(m_j \| u)$ is uniformly distributed and independent of the adversary's view. Thus, the oracles $\mathsf{RoR}^H(\mathsf{real}, pk, \cdot)$ and $\widetilde{\mathsf{RoR}}^H(\mathsf{real}, pk, \cdot)$ are identical as long as the event $\mathtt{Bad} = \cup_{j=1}^{T} \mathtt{Bad}_j$ does not occur. This implies that

$$\left| \Pr\left[\mathsf{Expt}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1(\lambda) = 1\right] \right| \leq \Pr[\mathtt{Bad}].$$

To calculate Pr[Bad], we divide the random oracle queries during the experiment Expt_0 and Expt_1 into the following three (disjoint) phases.

Phase I: Random oracle queries that are made by $\mathcal{A}$ *before* querying RoR^H $(\mathsf{real}, pk, \cdot)$ or $\widetilde{\mathsf{RoR}}^H$ $(\mathsf{real}, pk, \cdot)$. During this phase the two experiments are identical. As $\mathcal{A}$ is a probabilistic polynomial-time algorithm, it queries H only a polynomial number of times. Noting that for each $j \in [T]$, the random variable corresponding to m_j has min-entropy at least $k(\lambda)$, the probability over the choice of m_j that m_j appears in even one of the H-queries that were made by $\mathcal{A}$ before m_j is sampled is at most $\mathrm{poly}(\lambda) \cdot 2^{-k(\lambda)}$, which is negligible since $k(\lambda) = \omega(\log \lambda)$. Thus, the probability that the event Bad occurs in phase I is negligible in either one of Expt_0 or Expt_1.

Phase II: Random oracle queries that are made by the challenge distribution $\boldsymbol{M}$. We model the probabilistic polynomial-time algorithm that samples from $\boldsymbol{M}$ as taking a single input a sufficiently long random string $z \in \{0,1\}^*$. Then, for any $\boldsymbol{M} \in \mathcal{X}$ and randomness z, the probability over the choice of $u \leftarrow \{0,1\}^\ell$ that $\boldsymbol{M}(z)$ queries H on an input of the form $(*\|u)$ is at most $\mathrm{poly}(\lambda) \cdot 2^{-|u|}$, where $\mathrm{poly}(\lambda)$ is an upper bound on the number of oracle queries made by any $\boldsymbol{M} \in \mathcal{X}$. Therefore, for any randomness z, a union bound over all $\boldsymbol{M} \in \mathcal{X}$ implies that

$$\Pr_{u \leftarrow \{0,1\}^\ell}\left[\exists \boldsymbol{M} \in \mathcal{X} \text{ s.t. } \boldsymbol{M}(z) \text{ queries } H \text{ on some } (*\|u)\right] \leq \mathrm{poly}(\lambda) \cdot 2^{-|u|} \cdot |\mathcal{X}|.$$

From our choice of ℓ, the probability therefore that Bad occurs for the first time in phase II is negligible in either Expt_0 or Expt_1.

Phase III: Random oracle queries that are made by $\mathcal{A}$ *after* querying RoR^H $(\mathsf{real}, pk, \cdot)$ or $\widetilde{\mathsf{RoR}}^H$ $(\mathsf{real}, pk, \cdot)$. Assuming that the event Bad did not occur during phases I and II, then during this phase Expt_0 and Expt_1 are identical until the event Bad occurs.[12] Therefore, it suffices to consider Expt_1.

In Expt_1, however, the security of the underlying encryption scheme yields that the view of the adversary $\mathcal{A}$ is computationally indistinguishable from being independent of $(m_1, \ldots, m_T)$. Specifically, since the oracle $\widetilde{\mathsf{RoR}}$ uses true randomness when encrypting

[12] This is observed by noticing that if the event Bad did not occur during phases I and II, then for every $j \in [T]$ H was not queried on $(m_j \| u)$. Thus, if for every $j \in [T]$ we set the value of the random oracle H at the point $(m_j \| u)$ to r_j (where r_j is the random string used by $\widetilde{\mathsf{RoR}}$ to encrypt m_j) it holds that the two executions are in fact identical (until, clearly, the event Bad occurs for the first time).

$m_1, \ldots, m_T$, the security of the underlying encryption scheme enables us to replace the output of this oracle by T random encryptions of 0, and the probability of the event Bad will change by only a negligible additive factor. In this case, since $\mathcal{A}$ queries the random oracle only a polynomial number of times, and since each m_j has min-entropy at least $k(\lambda) = \omega(\log \lambda)$, there is only a negligible probability that some m_j would appear in even one of the H-queries that were made by $\mathcal{A}$. Thus, the probability that the event Bad occurs for the first time in phase III is negligible in either one of Expt_0 or Expt_1. □

Experiment $\mathbf{Expt_2}$. This experiment is obtained from Expt_1 by replacing $\widetilde{\mathsf{RoR}}(\mathsf{real}, \cdot, \cdot)$ with $\widetilde{\mathsf{RoR}}(\mathsf{rand}, \cdot, \cdot)$.

Claim 8.3. $\left|\Pr\left[\mathsf{Expt}_1(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2(\lambda) = 1\right]\right|$ *is negligible in* λ.

Proof. This follows from the security of the underlying scheme Π. Observe that as $\widetilde{\mathsf{RoR}}$ implements Π using true randomness for the encryption process. Therefore, any adversary $\mathcal{A}$ for which $|\Pr\left[\mathsf{Expt}_1(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2(\lambda) = 1\right]|$ is non-negligible can be used to break the IND-CPA security or the IND-CCA security of Π. □

Experiment $\mathbf{Expt_3}$. This experiment is obtained from Expt_2 by replacing $\widetilde{\mathsf{RoR}}(\mathsf{rand}, \cdot, \cdot)$ with $\mathsf{RoR}(\mathsf{rand}, \cdot, \cdot)$. That is, this is experiment $\mathsf{Expt}^{\mathsf{rand}}_{\Pi',\mathcal{A}}(\lambda)$ (recall Definition 3.3) if Π is IND-CPA-secure, experiment $\mathsf{Expt}^{\mathsf{randCCA}}_{\Pi',\mathcal{A}}(\lambda)$ (recall Definition 3.6) if Π is IND-CCA-secure.

Claim 8.4. $\left|\Pr\left[\mathsf{Expt}_2(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3(\lambda) = 1\right]\right|$ *is negligible in* λ.

Proof. The proof of this claim follows in an identical manner to the proof of Claim 8.2 noting that the distribution U^T from which challenge messages are sampled has min-entropy at least k in each coordinate. □

Completing the proof of Theorem 8.1. Let $(\mathrm{ATK}, \mathsf{mode}_1, \mathsf{mode}_2) = (\text{ACD1-CPA}, \mathsf{real}, \mathsf{rand})$ if Π is IND-CPA-secure, and let $(\mathrm{ATK}, \mathsf{mode}_1, \mathsf{mode}_2) = (\text{ACD1-CCA}, \mathsf{realCCA}, \mathsf{randCCA})$ if Π is IND-CCA-secure. Then, the definition of $\mathbf{Adv}^{\mathrm{ATK}}_{\Pi'_p,\mathcal{A}}(\lambda)$ implies that for any 2^p-bounded (T, k)-source adversary $\mathcal{A}$ it holds that

$$
\begin{aligned}
\mathbf{Adv}^{\mathrm{ATK}}_{\Pi'_p,\mathcal{A}}(\lambda) &\stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{mode}_1}_{\Pi'_p,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{mode}_2}_{\Pi'_p,\mathcal{A}}(\lambda) = 1\right]\right| \\
&= \left|\Pr\left[\mathsf{Expt}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3(\lambda) = 1\right]\right| \\
&\le \left|\Pr\left[\mathsf{Expt}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1(\lambda) = 1\right]\right| \qquad (8.1) \\
&\quad + \left|\Pr\left[\mathsf{Expt}_1(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2(\lambda) = 1\right]\right| \qquad (8.2) \\
&\quad + \left|\Pr\left[\mathsf{Expt}_2(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3(\lambda) = 1\right]\right|. \qquad (8.3)
\end{aligned}
$$

Claims 8.2–8.4 state that the terms in Eqs. (8.1)–(8.3) are negligible, and this completes the proof of Theorem 8.1. □

8.2. *A Construction Secure Against q-Query (T, k)-Source Adversaries*

We define the notion of a q-query adversary, and extend our notions of adaptive security to such adversaries. Our definitions, in addition to the parameters T denoting the number of blocks and $k = k(\lambda)$ denoting the min-entropy requirement are parameterized by a new parameter $q = q(\lambda)$ that denotes an upper bound on the number of queries to the random oracle required for sampling from $\boldsymbol{M}$. Unlike the definitions in Sect. 3, we do not need a bound 2^p on the set of allowed message distributions. As before, they are implicitly parameterized by bit-length $n = n(\lambda)$ of plaintext blocks.

Definition 8.5. (*q-query (T, k)-source adversary*) Let $\mathcal{A}$ be a probabilistic polynomial-time algorithm that is given as input a pair $(1^\lambda, pk)$ and oracle access to $\mathsf{RoR}(\mathsf{mode}, pk, \cdot)$ for some $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$. Then, $\mathcal{A}$ is a *q-query (T, k)-source adversary* if for each of $\mathcal{A}$'s RoR queries $\boldsymbol{M}$ it holds that:

- $\boldsymbol{M}$ is a (T, k)-source that is samplable by a polynomial-size circuit using at most q queries to the random oracle.
- For any $(m_1, \ldots, m_T)$ in the support of $\boldsymbol{M}$ it holds that $m_i \neq m_j$ for any distinct $i, j \in [T]$.

In addition, $\mathcal{A}$ is a *block-source* adversary if each such $\boldsymbol{M}$ is a (T, k)-block-source.

Definition 8.6. (Adaptive chosen-distribution attacks (ACD-CPA)) A deterministic public-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is (q, T, k)-*ACD-CPA-secure* (resp. *block-wise* (s, k)-*ACD-CPA-secure*) if for any probabilistic polynomial-time q-query (T, k)-source (resp. block-source) adversary $\mathcal{A}$, there exists a negligible function $\nu(k)$ such that

$$\mathbf{Adv}^{\text{ACD-CPA}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\Pi,\mathcal{A}}(\lambda) = 1\right]\right| \leq \nu(\lambda),$$

where for each $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{\mathsf{mode}}_{\Pi,\mathcal{A}}(\lambda)$ is identical to the one in Definition 3.3.

In addition, such a scheme is (q, T, k)-*ACD1-CPA-secure* (resp. *block-wise* (q, T, k)-*ACD1-CPA-secure*) if the above holds for any probabilistic polynomial-time q-query (T, k)-source (resp. block-source) adversary $\mathcal{A}$ that queries RoR at most once.

Definition 8.7. (*q-query (T, k)-source chosen-ciphertext adversary*) Let $\mathcal{A}$ be an algorithm that is given as input a pair $(1^\lambda, pk)$ and oracle access to two oracles: $\mathsf{RoR}(\mathsf{mode}, pk, \cdot)$ for some $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, and $\mathsf{Dec}(sk, \cdot)$. Then, $\mathcal{A}$ is an *q-query (T, k)-source chosen-ciphertext (resp. block-source) adversary* if the following two conditions hold:

1. $\mathcal{A}$ is an q-query (T, k)-source (resp. block-source) adversary.
2. $\mathcal{A}$ does not query $\mathsf{Dec}(sk, \cdot)$ with any ciphertext c that was part of a previous output by the RoR oracle.

Definition 8.8. (*Adaptive chosen-distribution chosen-ciphertext attacks*) A deterministic public-key encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is (q, T, k)-*ACD-CCA-*

secure (resp. *block-wise* (q, T, k)*-ACD-CCA-secure*) if for any probabilistic polynomial-time q-query (T, k)-source (resp. block-source) chosen-ciphertext adversary $\mathcal{A}$, there exists a negligible function $\nu(k)$ such that

$$\mathbf{Adv}^{\text{ACD-CCA}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{realCCA}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{randCCA}}_{\Pi,\mathcal{A}}(\lambda) = 1\right]\right| \leq \nu(\lambda),$$

where for each $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{\mathsf{modeCCA}}_{\Pi,\mathcal{A}}(\lambda)$ is identical to the one defined in Definition 3.6.

We are now ready to describe a scheme with short public keys.

The scheme Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a (randomized) public-key encryption scheme. We denote by $n = n(\lambda)$ and $\rho = \rho(\lambda)$ the bit-lengths of the messages and random strings that are given as input to the encryption algorithm Enc, respectively. In addition, let $H : \{0,1\}^* \to \{0,1\}^\rho$ be a hash function modeled as a random oracle. Our scheme $\Pi'_q = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ is parameterized by an upper bound $q = q(\lambda)$ on the number of random oracle queries made by the plaintext distributions.

- **Key generation** The key-generation algorithm KeyGen' is identical to the underlying key-generation algorithm KeyGen.
- **Encryption** The encryption algorithm Enc' on input a public key pk and a message m computes $r_m = \oplus_{i=1}^{q+1} H(m\|i)$, and outputs $c = \mathsf{Enc}_{pk}(m; r_m)$.
- **Decryption** The decryption algorithm Dec' is identical to the underlying decryption algorithm Dec.

Theorem 8.9. *Let* Π *be a randomized public-key encryption scheme. Then, for any polynomials* $q = q(\lambda)$, $T = T(\lambda)$, *and for any* $k = k(\lambda) = \omega(\log \lambda)$ *the following hold:*

1. *If* Π *is* IND-CPA*-secure then* Π'_q *is* (q, T, k)*-ACD-CPA-secure.*
2. *If* Π *is* IND-CCA*-secure then* Π'_q *is* (q, T, k)*-ACD-CCA-secure.*

Proof overview The main idea underlying the proof of security is similar to that underlying the proof of Theorem 8.1: For any challenge message m, as long as H is not queried on the $q + 1$ points $m\|1, \ldots, m\|q + 1$, either by the adversary $\mathcal{A}$ or by its adaptively chosen-plaintext distribution $M \in \mathcal{X}$, then the adversary learns essentially no information on m (for simplicity we focus here on security for a single message m, and refer the reader to the formal analysis below for the general case). As in the proof of Theorem 8.1, we divide the set of random oracle queries made by $\mathcal{A}$ and M into queries made in either of the following three phases, and for each of these phases we argue that the random oracle is queried with $q + 1$ queries $m\|1, \ldots, m\|q + 1$ only a negligible probability:

- H-queries made by $\mathcal{A}$ *before* querying the real-or-random encryption oracle: As $\mathcal{A}$ runs in polynomial time, and $m \leftarrow M$ is sampled with a super-logarithmic min-entropy, then it is unlikely that $\mathcal{A}$ queries H with any input of the form $m\|*$.
- H-queries made by the challenge plaintext distribution M: The distribution is q-query bounded, and therefore there is at least one index $j \in [q + 1]$ such that $m\|j$ is not queried by the circuit that samples the distribution.

- H-queries made by $\mathcal{A}$ *after* querying the real-or-random encryption oracle: Assuming that H was not queried with each of $m\|j$ for $j \in [q+1]$ in either one of the two previous phases, then the value $r_m = \oplus_{i=1}^{q+1} H(m\|i)$ is independently and uniformly distributed from the adversary's point of view (subject to producing the challenge ciphertext). Thus, any adversary that queries H on each of $m\|j$ in this phase for the first time case be used to break the security of the underlying (randomized) encryption scheme.

Proof of Theorem 8.9. Using Theorems 3.4 and 3.7 slightly modified to accommodate q-query adversaries, it suffices to prove that Π'_q is (q, T, k)-ACD1-CPA-secure if Π is IND-CPA-secure and Π'_q is (q, T, k)-ACD1-CCA-secure if Π is IND-CCA-secure. As mentioned above, the following proof is similar to that of Theorem 8.1 and follows essentially the same structure and reasoning.

Let $\mathcal{A}$ be a (q, T, k)-ACD1-CPA adversary if Π is IND-CPA-secure and a (q, T, k)-ACD1-CCA adversary if Π is IND-CCA-secure. In the random oracle model, the adversary $\mathcal{A}$ gets oracle access to H. Additionally, from the definition of a q-query adversary, the (T, k)-source $\boldsymbol{M}$ with which $\mathcal{A}$ queries $\mathsf{RoR}(\mathsf{mode}, pk, \cdot)$ is samplable by *oracle* circuit (that is, the circuit is allowed to contain H-gates) with at most q oracle gates. Note that the decryption algorithm $\mathsf{Dec}'_{sk}(\cdot)$ is identical to $\mathsf{Dec}_{sk}(\cdot)$ of the underlying scheme Π and therefore does not query the random oracle H.[13] In what follows, we describe four experiments, $\mathsf{Expt}_0, \ldots, \mathsf{Expt}_3$, and derive a series of claims relating them. We then combine these claims to bound the advantage of the adversary.

For our proof we define a variant $\widetilde{\mathsf{RoR}}$ of the oracle RoR, which uses true randomness for the encryption process instead of the value $r_m = \oplus_{i=1}^{q+1} H(m\|i)$. Specifically, on input $(\mathsf{mode}, pk, \boldsymbol{M})$ it first samples $(m_1, \ldots, m_T)$ from either $\boldsymbol{M}$ if $\mathsf{mode} = \mathsf{real}$ or U^T if $\mathsf{mode} = \mathsf{rand}$, then samples $r_1, \ldots, r_T \leftarrow (\{0,1\}^\rho)^T$ independently and uniformly at random, and outputs

$$\big(\mathsf{Enc}_{pk}(m_1; r_1), \ldots, \mathsf{Enc}_{pk}(m_T; r_T)\big).$$

Experiment Expt_0 This is the experiment $\mathsf{Expt}^{\mathsf{real}}_{\Pi',\mathcal{A}}(\lambda)$ (recall Definition 3.3) if Π is IND-CPA-secure or the experiment $\mathsf{Expt}^{\mathsf{realCCA}}_{\Pi',\mathcal{A}}(\lambda)$ (recall Definition 3.6) if Π is IND-CCA secure.

Experiment Expt_1 This experiment is obtained from Expt_0 by replacing $\mathsf{RoR}(\mathsf{real}, \cdot, \cdot)$ with $\widetilde{\mathsf{RoR}}(\mathsf{real}, \cdot, \cdot)$.

Claim 8.10. $\big|\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_1(\lambda) = 1]\big|$ *is negligible in* λ.

Proof. Let $(m_1, \ldots, m_T)$ denote the messages sampled from $\boldsymbol{M}$. For every $j \in [T]$ we denote by $\mathtt{Bad}_j$ the event in which H is queried on *each* of the $q+1$ points $(m_j\|1), \ldots, (m_j\|q+1)$. Note that for every $j \in [T]$, as long as the event $\mathtt{Bad}_j$ does not

[13] We do allow the underlying scheme Π to rely on a random oracle and in this case, we assume that H is independent of the random oracle used by Π (e.g., uses a different prefix).

occur, then the value $r_{m_j} = \oplus_{i=1}^{q+1} H(m_j \| i)$ is uniformly distributed and independent of the adversary's view. Thus, the oracles $\mathsf{RoR}^H(\mathsf{real}, pk, \cdot)$ and $\widetilde{\mathsf{RoR}}^H(\mathsf{real}, pk, \cdot)$ are identical as long as the event $\mathtt{Bad} = \cup_{j=1}^{T} \mathtt{Bad}_j$ does not occur. This implies that $\left|\Pr\left[\mathsf{Expt}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1(\lambda) = 1\right]\right| \leq \Pr[\mathtt{Bad}]$.

We divide the random oracle queries during the experiment Expt_0 and Expt_1 into the following three (disjoint) phases.

Phase I: Random oracle queries that are made by $\mathcal{A}$ *before* querying RoR^H (real, pk, $\cdot$) or $\widetilde{\mathsf{RoR}}^H$ (real, pk, $\cdot$). During this phase the two experiments are identical. As $\mathcal{A}$ is a probabilistic polynomial-time algorithm, it queries H only a polynomial number of times. Noting that for each $j \in [T]$, the random variable corresponding to m_j has min-entropy at least $k(\lambda)$, the probability over the choice of m_j that m_j appears in even one of the H-queries that were made by $\mathcal{A}$ before m_j is sampled is at most $\mathrm{poly}(\lambda) \cdot 2^{-k(\lambda)}$, which is negligible since $k(\lambda) = \omega(\log \lambda)$. Thus, the probability that the event $\mathtt{Bad}$ occurs in phase I is negligible in either one of Expt_0 or Expt_1.

Phase II: Random oracle queries that are made by $\boldsymbol{M}$. As $(m_1, \ldots, m_T) \leftarrow \boldsymbol{M}$ is chosen by a q-query adversary, the number of H-queries in this phase is at most q. Therefore, for any $j \in [T]$ assuming that m_j does not appear in an H-query in phase I, there always exists at least one index $i \in [q+1]$ such that H is not queried on $(m_j \| i)$. Thus, the probability that the event $\mathtt{Bad}$ occurs for the first time in phase II is negligible in either one of Expt_0 or Expt_1.

Phase III: Random oracle queries that are made by $\mathcal{A}$ *after* querying RoR^H (real, pk, $\cdot$) or $\widetilde{\mathsf{RoR}}^H$ (real, pk, $\cdot$). Assuming that the event $\mathtt{Bad}$ did not occur during phases I and II, then during this phase Expt_0 or Expt_1 are identical until the event $\mathtt{Bad}$ occurs.[14] Therefore, it suffices to consider Expt_1.

In Expt_1, however, the security of the underlying encryption scheme yields that the view of the adversary $\mathcal{A}$ is computationally indistinguishable from being independent of $(m_1, \ldots, m_T)$. Specifically, since the oracle $\widetilde{\mathsf{RoR}}$ uses true randomness when encrypting $m_1, \ldots, m_T$, the security of the underlying encryption scheme enables us to replace the output of this oracle by T random encryptions of 0, and the probability of the event $\mathtt{Bad}$ will change by only a negligible additive factor. In this case, since $\mathcal{A}$ queries the random oracle only a polynomial number of times, and since each m_j has min-entropy at least $k(\lambda) = \omega(\log \lambda)$, there is only a negligible probability that some m_j would appear in even one of the H-queries that were made by $\mathcal{A}$. Thus, the probability that the event $\mathtt{Bad}$ occurs for the first time in phase III is negligible in either one of Expt_0 or Expt_1. □

[14] This is observed by noticing that if the event $\mathtt{Bad}$ did not occur during phases I and II, then for every $j \in [T]$ there exists at least one index $i = i(j) \in [q+1]$ such that H was not queried on $(m_j \| i)$. Thus, if for every $j \in [T]$ we set the value of the random oracle H at the point $(m_j \| i)$ to $r_j \oplus_{t \in [q+1] \setminus \{i\}} H(m_j \| t)$ (where r_j is the random string used by $\widetilde{\mathsf{RoR}}$ to encrypt m_j) it holds that the two executions are in fact identical (until, clearly, the event $\mathtt{Bad}$ occurs for the first time).

Experiment $\mathbf{Expt_2}$ This experiment is obtained from Expt_1 by replacing $\widetilde{\mathsf{RoR}}(\mathsf{real}, \cdot, \cdot)$ with $\widetilde{\mathsf{RoR}}(\mathsf{rand}, \cdot, \cdot)$.

Claim 8.11. $\left|\Pr\left[\mathsf{Expt}_1(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2(\lambda) = 1\right]\right|$ *is negligible in* λ.

Proof. This follows from the security of the underlying scheme Π. Observe that as $\widetilde{\mathsf{RoR}}$ implements Π using true randomness for the encryption process. Therefore, any adversary $\mathcal{A}$ for which $|\Pr\left[\mathsf{Expt}_1(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2(\lambda) = 1\right]|$ is non-negligible can be used to break the IND-CPA security or the IND-CCA security of Π. □

Experiment $\mathbf{Expt_3}$ This experiment is obtained from Expt_2 by replacing $\widetilde{\mathsf{RoR}}(\mathsf{rand}, \cdot, \cdot)$ with $\mathsf{RoR}(\mathsf{rand}, \cdot, \cdot)$. That is, this is experiment $\mathsf{Expt}^{\mathsf{rand}}_{\Pi',\mathcal{A}}(\lambda)$ (recall Definition 3.3) if Π is IND-CPA-secure, experiment $\mathsf{Expt}^{\mathsf{randCCA}}_{\Pi',\mathcal{A}}(\lambda)$ (recall Definition 3.6) if Π is IND-CCA-secure.

Claim 8.12. $\left|\Pr\left[\mathsf{Expt}_2(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3(\lambda) = 1\right]\right|$ *is negligible in* λ.

Proof. The proof of this claim follows in an identical manner to the proof of Claim 8.10 noting that the distribution U^T from which challenge messages are sampled has min-entropy at least k in each coordinate. □

Completing the proof of Theorem 8.9 Let $(\mathsf{ATK}, \mathsf{mode}_1, \mathsf{mode}_2) = (\mathsf{ACD1\text{-}CPA}, \mathsf{real}, \mathsf{rand})$ if Π is IND-CPA-secure, and let $(\mathsf{ATK}, \mathsf{mode}_1, \mathsf{mode}_2) = (\mathsf{ACD1\text{-}CCA}, \mathsf{realCCA}, \mathsf{randCCA})$ if Π is IND-CCA-secure. Then, the definition of $\mathbf{Adv}^{\mathsf{ATK}}_{\Pi'_q,\mathcal{A}}(\lambda)$ implies that for any q-query (T, k)-source adversary $\mathcal{A}$ it holds that

$$
\begin{aligned}
\mathbf{Adv}^{\mathsf{ATK}}_{\Pi'_q,\mathcal{A}}(\lambda) &\stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{mode}_1}_{\Pi'_q,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{mode}_2}_{\Pi'_q,\mathcal{A}}(\lambda) = 1\right]\right| \\
&= \left|\Pr\left[\mathsf{Expt}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3(\lambda) = 1\right]\right| \\
&\le \left|\Pr\left[\mathsf{Expt}_0(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_1(\lambda) = 1\right]\right| \qquad (8.4) \\
&\quad + \left|\Pr\left[\mathsf{Expt}_1(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_2(\lambda) = 1\right]\right| \qquad (8.5) \\
&\quad + \left|\Pr\left[\mathsf{Expt}_2(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}_3(\lambda) = 1\right]\right|. \qquad (8.6)
\end{aligned}
$$

Claims 8.10–8.12 state that the terms in Eqs. (8.4)–(8.6) are negligible, and this completes the proof of Theorem 8.9. □

Acknowledgements

We thank David Xiao and Damien Vergnaud for a discussion regarding the parameters stated in Theorem 7.1, and the anonymous referees for their many useful comments.

References

[1] D. Boneh, X. Boyen, Secure identity based encryption without random oracles, in *Advances in Cryptology—CRYPTO '04*, 2004, pp. 443–459

[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] M. Bellare, J. Rompel, Randomness-efficient oblivious sampling, in *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, 1994, pp. 276–287

[7] Z. Brakerski, G. Segev, Better security for deterministic public-key encryption: the auxiliary-input setting, in *Advances in Cryptology—CRYPTO '11*, 2011, pp. 543–560

[8] E. Boyle, G. Segev, D. Wichs, Fully leakage-resilient signatures, in *Advances in Cryptology—EUROCRYPT '11*, 2011, pp 89–108

[9] D. Cash, D. Hofheinz, E. Kiltz, C. Peikert, Bonsai trees, or how to delegate a lattice basis, in *Advances in Cryptology—EUROCRYPT '10*, 2010, pp. 523–552

[10] Y. Dodis, *Exposure-Resilient Cryptography*. PhD thesis, MIT, 2000

[11] Y. Dodis, A. Smith, Correcting errors without leaking partial information, in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, 2005, pp. 654–663

[12] 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

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

[14] 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

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

[16] E. Kaplan, M. Naor, O. Reingold, Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica* **55**(1), 113–133 (2009)

[17] E. Kiltz, A. O'Neill, A. Smith, Instantiability of RSA-OAEP under chosen-plaintext attack, in *Advances in Cryptology—CRYPTO '10*, 2010, pp. 295–313

[18] I. Mironov, O. Pandey, O. Reingold, G. Segev, Incremental deterministic public-key encryption, in *Advances in Cryptology—EUROCRYPT '12*, 2012, pp. 628–644

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

[20] J. Rompel, *Techniques for computing with low-independence randomness*. PhD thesis, Massachusetts Institute of Technology, 1990

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

[22] L. Trevisan, S.P. Vadhan, Extracting randomness from samplable distributions, in *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000, pp. 32–42

[23] S. Vadhan, Pseudorandomness (draft survey). http://people.seas.harvard.edu/~salil/pseudorandomness/, 2012

[24] B. Waters, Efficient identity-based encryption without random oracles, in *Advances in Cryptology—EUROCRYPT '05*, 2005, pp. 114–127

[25] H. Wee, Dual projective hashing and its applications—lossy trapdoor functions and more, in *Advances in Cryptology—EUROCRYPT '12*, 2012, pp. 246–262

[26] D. Wichs, Barriers in cryptography with weak, correlated and leaky sources, in *Proceedings of the 4th Innovations in Theoretical Computer Science Conference*, 2013