An extended abstract of this article appears in the proceedings of CT-RSA 2019. This is the full version.

Lossy Trapdoor Permutations with Improved Lossiness

Benedikt Auerbach[1], Eike Kiltz[2], Bertram Poettering[3], and Stefan Schoenen[4]

[1] Horst-Görtz Institute for IT Security, Ruhr-University Bochum, Germany
benedikt.auerbach@rub.de

[2] Horst-Görtz Institute for IT Security, Ruhr-University Bochum, Germany
eike.kiltz@rub.de, ORCID iD 0000-0003-1178-048X

[3] Royal Holloway, University of London, UK
bertram.poettering@rhul.ac.uk, ORCID iD 0000-0001-6525-5141

[4] paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen, Germany
stefan.schoenen@paluno.uni-due.de

Abstract. Lossy trapdoor functions (Peikert and Waters, STOC 2008 and SIAM J. Computing 2011) imply, via black-box transformations, a number of interesting cryptographic primitives, including chosen-ciphertext secure public-key encryption. Kiltz, O'Neill, and Smith (CRYPTO 2010) showed that the RSA trapdoor permutation is lossy under the Phi-hiding assumption, but syntactically it is not a lossy trapdoor function since it acts on $\mathbb{Z}_N$ and not on strings. Using a domain extension technique by Freeman et al. (PKC 2010 and J. Cryptology 2013) it can be extended to a lossy trapdoor permutation, but with considerably reduced lossiness.

In this work we give new constructions of lossy trapdoor permutations from the Phi-hiding assumption, the quadratic residuosity assumption, and the decisional composite residuosity assumption, all with improved lossiness. Furthermore, we propose the first all-but-one lossy trapdoor permutation from the Phi-hiding assumption. A technical vehicle used for achieving this is a novel transform that converts trapdoor functions with index-dependent domain to trapdoor functions with fixed domain.

1 Introduction

Lossy Trapdoor Functions. Lossy trapdoor functions (LTFs) are like classic (one-way) trapdoor functions but with strengthened security properties. Instances of an LTF can be created in two computationally indistinguishable ways: An instance generated with the standard key-generation algorithm describes an injective function that can be efficiently inverted using the trapdoor; and an instance generated with the lossy key-generation algorithm describes a "lossy" function, meaning its range is considerably smaller than its domain. The *lossiness factor* $L \geq 1$, defined as the ratio of the cardinalities of domain and range, measures the LTF's quality.[5] The larger the lossiness factor, the better the cryptographic properties of the LTF. In case the non-lossy instances define permutations, we will refer to the whole object as a lossy trapdoor permutation (LTP).

Lossy trapdoor functions were introduced by Peikert and Waters [23,24] who showed that they imply (via black-box constructions) fundamental cryptographic primitives such as classic trapdoor functions, collision-resistant hash functions, oblivious transfer, and chosen-ciphertext secure public-key encryption. Furthermore, LTFs have found various other applications, including deterministic public-key encryption [7], OAEP-based public-key encryption [18], "hedged" public-key encryption for protecting against bad randomness [2,4], security against selective opening attacks [5], efficient non-interactive string commitments [21], threshold encryption [28], correlated-product secure trapdoor functions [26], adaptive trapdoor functions [17], and many others.

LTFs with index-dependent domains. In the original definition by Peikert and Waters, all instances of an LTF are defined over the same fixed domain $\{0,1\}^k$. That is, the domain is independent of the specific index output by the key-generation algorithm ('index' is used synonym with the public key describing the instance). Subsequently, LTFs were generalized to *LTFs with index-dependent domains* [12] where the domain may depend on the function's index. To illustrate index-dependent domains, consider the well-known RSA trapdoor permutation $f_{\mathrm{RSA}} \colon \mathbb{Z}_N \to \mathbb{Z}_N;\ x \mapsto x^e \bmod N$. Its index consists of a modulus $N = pq$ (of fixed bit-length k) and an exponent e; its domain is $\mathbb{Z}_N$, hence it is index-dependent. For $e \leq 2^{k/4}$, permutation f_{RSA} was proved to be lossy [18] with lossiness factor $L = e$ under the Phi-hiding assumption [9].[6] Similarly, constructions of trapdoor functions based on quadratic residuosity or Paillier's assumption yield LTPs with index-dependent domains [11,12].

As pointed out in [12], LTFs with index-dependent domains do not seem to be sufficient for constructing correlated-product secure trapdoor functions [26] or chosen-ciphertext secure public-key encryption [23]. The difficulty is that in these applications a fixed value has to be evaluated on many independently generated instances of the trapdoor function. It is therefore crucial that the domains are the same for all these instances. Furthermore, most constructions of deterministic encryption schemes (e.g., [7,3,8,19,25]) assume message distributions that do not depend on the public key and hence cannot be constructed from LTFs with index-dependent domains. Fortunately, however, LTFs with index-dependent domains turn out to be sufficient for many other applications.

In [12, Section 3.2], a general domain-extension technique was (implicitly) proposed to transform an LTF $f \colon \mathbb{Z}_N \to \mathbb{Z}_N$ with index-dependent domain $\mathbb{Z}_N$ (with $2^{k-1} \leq N < 2^k$) into an

[5] The original definition of lossy trapdoor functions [23,24] measures lossiness on a logarithmic scale. That is, $\ell := \log_2(L)$ is the lossiness of the LTF and L is the lossiness factor (which we use in this work).

[6] In brief, the Phi-hiding assumption states that (N, e), where $N = pq$ and $e \nmid \varphi(N)$, is computationally indistinguishable from (N, e), where $N = pq$ and $e \mid \varphi(N)$. The Phi-hiding assumption is conjectured to hold for $e \leq N^{1/4-\epsilon}$ and does not hold for $e > N^{1/4}$ (due to a Coppersmith-like attack). If $e \mid \varphi(N)$, then $f_{\mathrm{RSA}}(x) = x^e \bmod N$ is roughly an e-to-1 function.

LTF $f_{\mathrm{de}}\colon \{0,1\}^k \to \{0,1\}^k$ with index-independent domain $\{0,1\}^k$ by defining

$$f_{\mathrm{de}}(x) := \begin{cases} f(x) & 0 \le x < N \\ x & N \le x < 2^k \end{cases} . \tag{1}$$

However, this transform does bad in preserving lossiness, in particular in the case where N is close to 2^{k-1}. Indeed, if the lossiness factor of f is L then the lossiness factor of f_{de} is about $L_{\mathrm{de}} = 2 \cdot L/(L+1) < 2$. Note that such a small lossiness factor does not even imply one-wayness, i.e., the resulting LTF is, taken by itself, essentially useless. (Based on a result by Mol and Yilek [20] it can still be used to build IND-CCA secure encryption, but with considerably worse efficiency.) In [12, Section 4.4] also an alternative domain-extension technique was sketched that can be used to construct an LTF f_{de} with index-independent domain $\{0,1\}^{k+\log(L)}$ and lossiness factor $L_{\mathrm{de}} \approx L$. Here, every evaluation of f_{de} requires $\log(L)$ many applications of f. For interesting values of L this is again prohibitively inefficient.

All-but-one Lossy Trapdoor Functions. All-but-one lossy trapdoor functions (ABO-LTFs) are a generalization of LTFs. An ABO-LTF is associated with a set $\mathcal{B}r$ of branches. The corresponding generator algorithm is invoked on input a target branch $br^* \in \mathcal{B}r$ and outputs a trapdoor and a family of functions $(f_{br})_{br \in \mathcal{B}r}$ with the property that f_{br} is injective for all $br \neq br^*$ (and can be inverted using the trapdoor), but function f_{br^*} is lossy. Moreover, the lossy branch is hidden (computationally) by the description of the function family. ABO-LTFs with just two branches are equivalent to LTFs, and, similarly to LTFs, ABO-LTFs can have index-independent or index-dependent domains. Using the techniques of Peikert and Waters [23] an ABO-LTF with exponentially large branch set can be constructed from any LTF, but the latter is required to have a sufficiently large lossiness factor L. (This transformation also works for LTFs with index-dependent domains.) Many of the mentioned applications of LTFs require in fact ABO-LTFs.

Known LTFs and ABO-LTFs. Roughly speaking, cryptographic assumptions are typically rooted in one out of three different environments: over cyclic groups, over lattices, or over RSA moduli. Over cyclic groups as well as over lattices, constructions of LTFs and ABO-LTFs are known [23]. They have index-independent domain and can be instantiated to have an arbitrarily large lossiness factor L. In the RSA setting, the situation is different.[7] There are constructions known from the quadratic residuosity assumption [12], Paillier's decisional composite residuosity assumption [12], and from the Phi-hiding assumption [9,18] (for a fourth one, see below). All constructions have index-dependent domains (the transform sketched above fixes this, but the results are essentially useless due to the small lossiness factor). Unfortunately, for the constructions based on the Phi-hiding assumption and the quadratic residuosity assumption the lossiness factor cannot be made arbitrarily large and, in particular, it is not sufficient to construct efficient ABO-TDFs. However, both an index-independent LTF and an ABO-LTF based on the decisional composite residuosity assumption are known [11].

As it is quite general, we describe in more detail the technique from [23] for building LTFs. Starting with an additively homomorphic encryption scheme, function indices correspond with element-wise encryptions of the identity matrix. The range of the construction consists of vectors of ciphertexts. If ElGamal encryption is used to instantiate the encryption scheme one obtains an LTF with security based on DDH. Constructions of LTFs and ABO-LTFs in the same spirit, but that achieve smaller index sizes and output lengths, are proposed in [16,6]. Using a generalization of the Goldwasser–Micali homomorphic encryption scheme [13] allows

[7] When we say an LTF is "RSA-based" we mean it is defined in respect to some composite number $N = pq$ where p, q are primes. This shall not suggest its security relies on the RSA assumption (the hardness of computing e-th roots).

this construction, in contrast to processing the LTF input bit-by-bit, to consider input values sequences of numbers of some fixed bit-length. The construction's security is based not only on the DDH assumption but also on the quadratic residuosity assumption for a restricted class of RSA moduli and an additional non-standard assumption, which can be removed by making further restrictions on the modulus.

While the described constructions from [23,16,6] achieve high lossiness factors, a common disadvantage is that their indices are ciphertext matrices and the function ranges are ciphertext vectors, and thus quite large. Further, [16,6] require strong hardness assumptions in a quite restricted RSA setting.

As shown in [29], collision-resistant hash functions, CPA- and CCA-secure public-key encryption, and deterministic encryption can be constructed from adversary-dependent lossy trapdoor functions and ABO-LTFs, a variant of LTFs and ABO-LTFs with relaxed security conditions. The authors give index-independent constructions of these primitives from the factoring assumption for semi-smooth RSA moduli. The proposed instantiations achieve high lossiness factors and have compact indices and ciphertexts of roughly the size of an RSA modulus.

1.1 Our Results

In this work we propose a new general domain-extension transformation that can be used to transform index-dependent LTPs into index-independent LTPs without sacrificing much lossiness. Concretely, our transformation decreases the lossiness factor by at most by a factor of 2. For the special cases of the LTP based on the Phi-hiding assumption and the LTP from [12] based on the quadratic residuosity assumption, a more refined analysis even shows that the lossiness factor effectively stays invariant. That is, ultimately we construct an LTP with index-independent domain $\{0,1\}^k$ and lossiness factor as large as $L = 2^{k/4}$ from the Phi-hiding assumption, and an LTP with index-independent domain $\{0,1\}^k$ and lossiness factor 2 from the quadratic residuosity assumption. In comparison, the index-independent variants obtained via the transform implicitly given in [12] would result in lossiness factors of 2 and 4/3 respectively. Applying our transformation to the index-dependent LTF and ABO-LTF based on the decisional composite residuosity assumption yields index-independent variants with slightly larger domain and lossiness factor than the direct index-independent constructions of [12]. Finally we construct the first ABO-LTP from (a variant of) the Phi-hiding assumption. We highlight that in particular our Phi-hiding based construction has particularly compact indices (of the size of an RSA modulus) and range elements.

Domain extension for LTFs with index-dependent domains. We explain our domain extension technique for the special case of a LTF $f\colon \mathbb{Z}_N \to \mathbb{Z}_N$ with index-dependent domain $\mathbb{Z}_N$ (with $2^{k-1} \le N < 2^k$). We use a two-round construction in the spirit of Hayashi, Okamoto and Tanaka [14], who used a similar construction to extend the domain of the RSA one-way permutation. We define the function

$$f'_{\mathrm{de}}\colon \{0,1\}^k \to \{0,1\}^k, \quad f'_{\mathrm{de}}(x) := f_{\mathrm{de}}(\pi(f_{\mathrm{de}}(x))), \tag{2}$$

where f_{de} is defined in (1) and permutation $\pi\colon \{0,1\}^k \to \{0,1\}^k$ is given as $\pi(x) = x - (N - 1) \bmod 2^k$. The intuition of this construction is that the LTF f is applied to every $x \in \{0,1\}^k$ at least once. Indeed, if f is one-way, then f'_{de} defined in (2) is one-way [14]. Our first main result states that if f is a LTF with index-dependent domain and lossiness factor L, then f'_{de} is a LTF with index-independent domain $\{0,1\}^k$ and lossiness factor $L'_{\mathrm{de}} = L/2$.

In the case of the RSA-based LTF f_{RSA} we can even prove that the lossiness factor of f'_{de} is completely preserved, i.e. $L'_{\mathrm{de}} = L$. Under the Phi-hiding assumption this gives us a LTP with index-independent domain and lossiness factor as large as $k^{1/4}$. We also show how to obtain index-independent LTPs from the quadratic residuosity and the decisional composite residuosity assumption, which have a larger lossiness factor than the constructions of [11].

An ABO-LTP in the RSA setting. Our second main result is the construction of an ABO-LTP with index-dependent domain from the Phi-hiding assumption. Our generic domain extension technique also works for ABO-LTFs, so it can be transformed into an ABO-LTP with index-independent domain $\{0,1\}^k$.

Our construction essentially follows [17, Section 5.2] who construct an adaptive trapdoor function from the instance-independent RSA assumption, a decisional version of the RSA assumption. It makes use of a new primitive that we call *prime family generator* (PFG), an abstraction that may be of independent interest. An instance of a PFG indicates a fixed sequence of (distinct) primes $e_1, \ldots, e_{2^n}$ of some specified bit-length $l \geq n/2$. A specific programmability feature allows embedding any given prime at any given position, where the position remains hidden (computationally) from the instance. We give an information-theoretic construction of a PFG that is based on work by Cachin, Micali, and Stadler [9]. A PFG instance consists of l^2 bits and we leave it as an open problem to construct a (computationally secure) PFG with improved parameters, for example by using the PRF-based construction as implicitly in the work of Hohenberger and Waters [15].

Given a PFG we define our new RSA-setting based ABO-LTP for a branch $br \in \{0,1\}^n$ as

$$f_{br} \colon \mathbb{Z}_N \to \mathbb{Z}_N; \quad f_{br}(x) := x^{e_{br}} \ ,$$

where e_{br} is the br-th prime of the PFG prime sequence. To prove the ABO-LTF security property we first use the Phi-hiding assumption to change the distribution of the RSA modulus N to satisfy $e^* \mid \varphi_N$, for some random prime e^*. Next, we use the PFG's programmability feature to make sure that $e_{br^*} = e^*$, meaning the function $f_{br}(\cdot)$ is injective if $br \neq br^*$ and e^*-to-1 if $br = br^*$.[8]

Applications. Our constructions of index-independent LTFs and LTPs over domain $\{0,1\}^k$ (and our techniques to build them) are mostly of theoretical interest with potential future applications. Whereas with our current knowledge we are not able to present a killer application, let us still discuss possible minor applications. Most importantly, correlated-product secure trapdoor functions [26] and IND-CCA secure public-key encryption [23] can be constructed from index-independent LTFs over domain $\{0,1\}^k$. Both require the lossiness factor L to be larger than $2^{k/2}$, whereas our construction based on the Phi-hiding assumption cannot go beyond $L = 2^{k/4}$. One can still apply the amplification result by Mol and Yilek [20] to build IND-CCA secure encryption. The efficiency loss will be smaller than with the previous constructions from the Phi-hiding assumption (having lossiness factor $L \approx 2$).

2 Preliminaries

2.1 Notation

If $a, b \in \mathbb{N}, a < b$, we use notations $[a \mathinner{..} b] = \{a, \ldots, b\}$, $[b] = [1 \mathinner{..} b]$, $[\![a \mathinner{..} b]\!] = [a \mathinner{..} (b-1)]$, and $[\![b]\!] = [0 \mathinner{..} (b-1)]$. We say $m \in \mathbb{N}$ is an l-bit number if $m \in [\![2^{l-1} \mathinner{..} 2^l]\!]$. For any set $M \subseteq \mathbb{N}$ we denote with $M_l := M \cap [\![2^{l-1} \mathinner{..} 2^l]\!]$ its subset of l-bit elements. We write $\{0,1\}^l$ for the set of strings of length l and denote the bit-wise exclusive-or operation of same-length strings with $\oplus$. For all $l \in \mathbb{N}$ we assume a canonic bijection $\# \colon [\![2^l]\!] \to \{0,1\}^l$ and correspondingly denote with

[8] In fact this requires a slightly strengthened variant of the Phi-hiding assumption where for a larger set $\mathcal{E}$ it is known that precisely one element $e \in \mathcal{E}$ is a divisor of φ_N. We call this the *unique-divisor Phi-hiding assumption*, see Section 2.3.

$\#x$ the interpretation of an element x of $[\![2^l]\!]$ as a string in $\{0,1\}^l$. The support of a randomized algorithm A on input x, i.e., the set of values it outputs with non-zero probability, is denoted by $[\mathrm{A}(x)]$. We annotate a disjoint union with $\uplus$.

2.2 (All-but-one) lossy trapdoor permutations

We recall the concepts of lossy trapdoor functions and all-but-one lossy trapdoor functions as introduced by Peikert and Waters [23]. More precisely, we slightly deviate from their formalizations by restricting attention to permutations, supporting index-dependent domains [12], and considering permutations that are not perfectly correct.

Lossy trapdoor permutations. Let $\mathcal{X}$ be a domain, $\mathcal{Id}$ a universe of function indices, and for each index $id \in \mathcal{Id}$ let $\mathcal{X}(id) \subseteq \mathcal{X}$ be a specific (sub)domain. A *lossy trapdoor permutation* (LTP) for $\mathcal{X}, \mathcal{Id}$ then consists of a trapdoor space $\mathcal{Td}$ and three efficient algorithms $\mathrm{F} = (\mathrm{FGen}, \mathrm{FEv}, \mathrm{FInv})$ for which the following hold: Algorithm

$$\{0,1\} \to \mathrm{FGen} \to_{\$} \mathcal{Id} \times (\mathcal{Td} \uplus \{\perp\})$$

is a randomized instance generator. Its input $b \in \{0,1\}$ specifies whether the generated instance is *injective* ($b = 1$) or *lossy* ($b = 0$). We require $[\mathrm{FGen}(1)] \subseteq \mathcal{Id} \times \mathcal{Td}$ and $[\mathrm{FGen}(0)] \subseteq \mathcal{Id} \times \{\perp\}$. In injective mode, if $(id, td) \in [\mathrm{FGen}(1)]$, we refer to td as the trapdoor corresponding to id. Algorithms

$$\mathcal{Id} \times \mathcal{X} \to \mathrm{FEv} \to \mathcal{X} \qquad \text{and} \qquad \mathcal{Td} \times \mathcal{X} \to \mathrm{FInv} \to \mathcal{X}$$

are the *evaluation* and *inversion* algorithms, respectively. We require that $\mathrm{FEv}(id, x) \in \mathcal{X}(id)$ for all $id \in \mathcal{Id}$ and $x \in \mathcal{X}(id)$. For correctness we further require that in injective mode the mapping $\mathcal{X}(id) \to \mathcal{X}(id)$ induced by FEv can be effectively inverted on (almost) all values if the trapdoor is known. Formally, we say that F is $(1 - \epsilon_1)$-correct if

$$\Pr[(id, td) \leftarrow_{\$} \mathrm{FGen}(1), x \leftarrow_{\$} \mathcal{X}(id), y \leftarrow \mathrm{FEv}(id, x) : \mathrm{FInv}(td, y) \neq x] \leq \epsilon_1 \ .$$

This means that for $\epsilon_1 > 0$ the function implemented by $\mathrm{FEv}(id, \cdot)$ might technically not be a permutation. For security we require (a) that FEv lose information in lossy mode, and (b) that injective mode and lossy mode be indistinguishable. Concerning (a), we say the LTP is *L-lossy* if for all $(id, \perp) \in [\mathrm{FGen}(0)]$ we have $|\mathrm{FEv}(id, \mathcal{X}(id))| \leq |\mathcal{X}(id)|/L$.[9] Concerning (b), we say the LTP is *(τ, ϵ_2)-indistinguishable* if for all τ-time distinguishers $\mathcal{D}$ we have

$$\left| \begin{array}{l} \Pr[(id, td) \leftarrow_{\$} \mathrm{FGen}(1) : \mathcal{D}(id) \Rightarrow 1] \\ \quad - \Pr[(id, \perp) \leftarrow_{\$} \mathrm{FGen}(0) : \mathcal{D}(id) \Rightarrow 1] \end{array} \right| \leq \epsilon_2 \ .$$

All-but-one lossy trapdoor permutations. All-but-one LTPs are a generalization of LTPs where in addition to the universe of function indices there is a universe of branches; function FEv is lossy for one branch and injective for all others. In particular, a (regular) LTP is equivalent to an all-but-one LTP if the branch space consists of precisely two elements.

Let $\mathcal{Br}$ be a branch space, $\mathcal{X}$ a domain, $\mathcal{Id}$ a universe of function indices, and for each index $id \in \mathcal{Id}$ let $\mathcal{X}(id) \subseteq \mathcal{X}$ be a specific (sub)domain. An *all-but-one lossy trapdoor permutation*

[9] According to our definition, L-lossiness indicates that the size of the lossy image is by a factor L smaller than the domain. The original definition by Peikert and Waters indicates the same quantity on a logarithmic scale, i.e., they report $\log_2(L)$ instead of L.

(ABO-LTP) for $\mathcal{B}r, \mathcal{X}, \mathcal{I}d$ then consists of a trapdoor space $\mathcal{T}d$ and three efficient algorithms $\mathrm{A} = (\mathrm{FGen}, \mathrm{FEv}, \mathrm{FInv})$ for which the following hold: Algorithm

$$\mathcal{B}r \to \mathrm{FGen} \to_{\$} \mathcal{I}d \times \mathcal{T}d$$

is a randomized instance generator such that $(id, td) \leftarrow_{\$} \mathrm{FGen}(br)$, for a branch br, generates a function index id with trapdoor td. Similarly as for LTPs, algorithms

$$\mathcal{B}r \times \mathcal{I}d \times \mathcal{X} \to \mathrm{FEv} \to \mathcal{X} \qquad \text{and} \qquad \mathcal{B}r \times \mathcal{T}d \times \mathcal{X} \to \mathrm{FInv} \to \mathcal{X}$$

are the evaluation and inversion algorithms. We require that for all $br, br^* \in \mathcal{B}r$ and $(id, td) \in [\mathrm{FGen}(br^*)]$ and $x \in \mathcal{X}(id)$, if $y = \mathrm{FEv}(br, id, x)$ then $y \in \mathcal{X}(id)$. We further require that the mappings $\mathcal{X}(id) \to \mathcal{X}(id)$ induced by FEv on all branches with exception of br^* can be effectively inverted (on almost all values) if the trapdoor is known. Formally, we say that A is $(1 - \epsilon_1)$-correct if for all $br, br^* \in \mathcal{B}r$, $br \neq br^*$, we have

$$\Pr\left[(id, td) \leftarrow_{\$} \mathrm{FGen}(br^*), x \leftarrow_{\$} \mathcal{X}(id) : \mathrm{FInv}\left(br, td, \mathrm{FEv}(br, id, x)\right) \neq x\right] \leq \epsilon_1 \ .$$

For security we require that FEv lose information on its *lossy branch*, i.e., the branch br^* the instance was generated for. Further, it shall be unfeasible to identify the lossy branch. Concretely, we say the ABO-LTP is *L-lossy* if for all $br^* \in \mathcal{B}r$ and $(id, td) \in [\mathrm{FGen}(br^*)]$ we have $|\mathrm{FEv}(br^*, id, \mathcal{X}(id))| \leq |\mathcal{X}(id)|/L$, and we say it is *$(\tau, \epsilon_2)$-indistinguishable* if for all $br_0, br_1 \in \mathcal{B}r$ and all τ-time distinguishers $\mathcal{D}$ (that may depend on br_0, br_1) we have

$$\left|\begin{array}{l}\Pr[(id, td) \leftarrow_{\$} \mathrm{FGen}(br_0) : \mathcal{D}(id) \Rightarrow 1] \\ \quad - \Pr[(id, td) \leftarrow_{\$} \mathrm{FGen}(br_1) : \mathcal{D}(id) \Rightarrow 1]\end{array}\right| \leq \epsilon_2 \ .$$

Index-dependent vs. index-independent LTPs/ABO-LTPs. In the above definition of LTPs, the domain $\mathcal{X}(id) \subseteq \mathcal{X}$ on which $\mathrm{FEv}(id, \cdot)$ operates may depend on function index id. We say the LTP is index-independent if this restriction does not exist, i.e., if $\mathcal{X}(id) = \mathcal{X}$ for all id. For ABO-LTPs we say correspondingly. In later sections we show how to generically transform an index-dependent trapdoor permutation into an index-independent one.

2.3 Number theoretic assumptions

For $a, b \in \mathbb{N}, a \neq 0$, we write $a \mid b$ if a divides b, i.e., if there exists $d \in \mathbb{N}$ s.t. $b = da$. We further write $a \mid_1 b$ if a divides b exactly once, i.e., if $a \mid b \wedge a^2 \nmid b$. The greatest common divisor of a, b is denoted $\gcd(a, b)$. We denote the set of prime numbers with $\mathcal{P}$. Recall from Section 2.1 that $\mathbb{N}_l$ and $\mathcal{P}_l$ denote the sets of l-bit natural and prime numbers, respectively.

If k is an even number, a product $N = pq$ is a k-bit *RSA modulus* if $N \in \mathbb{N}_k$, $p, q \in \mathcal{P}_{k/2}$, and $p \neq q$. The order of the multiplicative group $\mathbb{Z}_N^*$ is $\varphi_N := \varphi(N) = (p-1)(q-1)$. We denote the space of k-bit RSA moduli with $\mathcal{RSA}_k$. If we want to restrict attention to k-bit RSA moduli that fulfill a specific condition C, we write $\mathcal{RSA}_k[C]$. The set of k-bit Blum integers, i.e., RSA moduli where the prime factors satisfy $p \equiv q \equiv 3 \bmod 4$, is denoted by $\mathcal{BRSA}_k := \mathcal{RSA}_k[p \equiv q \equiv 3 \bmod 4]$.

Phi-hiding assumption. In standard RSA encryption, public exponent e is chosen constraint to $e \nmid \varphi_N$ so that the mapping $x \mapsto x^e$ is a bijection. Some applications in addition use exponents $e \mid_1 \varphi_N$ and require that it be hard, given (N, e), to decide whether $e \mid_1 \varphi_N$ or $e \nmid \varphi_N$. Roughly, the Phi-hiding assumption [9,18] for a set of primes $\mathcal{E}$ says that $N \in \mathcal{RSA}_k$ can be generated such that for uniformly picked $e \in \mathcal{E}$ the cases $N \in \mathcal{RSA}_k[e \nmid \varphi_N]$ and $N \in \mathcal{RSA}_k[e \mid_1 \varphi_N]$ are computationally indistinguishable. Formally, we say that the (τ, ϵ)-*Phi-hiding assumption* holds for $(k, \mathcal{E})$ if for all τ-time adversaries $\mathcal{D}$ we have

$$\left| \begin{array}{l} \Pr[e \leftarrow_\$ \mathcal{E}; (N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[e \mid_1 \varphi_N] : \mathcal{D}(N, e) \Rightarrow 1] \\ \quad - \Pr[e \leftarrow_\$ \mathcal{E}; (N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[e \nmid \varphi_N] : \mathcal{D}(N, e) \Rightarrow 1] \end{array} \right| \leq \epsilon \ .$$

In the probability expressions we write $(N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[C]$ for an algorithm that generates a k-bit RSA modulus satisfying condition C, and also outputs $\varphi_N = |\mathbb{Z}_N^*|$.

In this paper we also need a variant of this assumption: An added restriction is that precisely one $e \in \mathcal{E}$ shall be a divisor of φ_N, and, as before, if e divides φ_N then at most once.[10] This is expressed by condition

$$C(\mathcal{E}, \varphi_N, e) \quad :\Longleftrightarrow \quad e \mid \varphi_N \wedge \gcd(\mathcal{E}, \varphi_N/e) = 1 \ ,$$

where the gcd term encodes that φ_N/e is relative prime to all elements of $\mathcal{E}$; this in particular implies $e \mid_1 \varphi_N$. We say the *unique-divisor* (τ, ϵ)-*Phi-hiding assumption* holds for $(k, \mathcal{E})$ if for all τ-time adversaries $\mathcal{D}$ we have

$$\left| \begin{array}{l} \Pr[e_0 \leftarrow_\$ \mathcal{E}; (N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[C(\mathcal{E}, \varphi_N, e_0)] : \mathcal{D}(N, e_0) \Rightarrow 1] \\ \quad - \Pr[e_0, e_1 \leftarrow_\$ \mathcal{E}; (N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[C(\mathcal{E}, \varphi_N, e_0)] : \mathcal{D}(N, e_1) \Rightarrow 1] \end{array} \right| \leq \epsilon \ .$$

Quadratic residuosity assumption. Roughly, the quadratic residuosity assumption says that it is hard to distinguish quadratic residues modulo a Blum integer from quadratic non-residues that have positive Jacobi symbol.

Formally, for all $N \in \mathbb{N}$ denote with $\mathcal{QR}_N \subseteq \mathbb{Z}_N^*$ the set of quadratic residues modulo N and with $\mathcal{J}_N \subseteq \mathbb{Z}_N^*$ the set of numbers with positive Jacobi symbol. (In particular we have $\mathcal{QR}_N \subseteq \mathcal{J}_N$.) We say that the (τ, ϵ)-*quadratic residuosity assumption* holds for k if for all τ-time adversaries $\mathcal{D}$ we have

$$\left| \begin{array}{l} \Pr[(N, p, q) \leftarrow_\$ \mathcal{BRSA}_k, x \leftarrow_\$ \mathcal{QR}_N : \mathcal{D}(N, x) \Rightarrow 1] \\ \quad - \Pr[(N, p, q) \leftarrow_\$ \mathcal{BRSA}_k, x \leftarrow_\$ \mathcal{J}_N \setminus \mathcal{QR}_N : \mathcal{D}(N, x) \Rightarrow 1] \end{array} \right| \leq \epsilon \ .$$

In the probability expressions we write $(N, p, q) \leftarrow_\$ \mathcal{BRSA}_k$ for an algorithm that generates a k-bit Blum integer and also outputs its prime factors. Note that sampling elements of $\mathcal{QR}_N$ and $\mathcal{J}_N \setminus \mathcal{QR}_N$ can be done efficiently if these factors are known.

Decisional composite residuosity assumption. Roughly, the decisional composite residuosity assumption [22] says that given an RSA modulus N it is hard to distinguish random elements of $\mathbb{Z}_{N^2}^*$ from Nth powers in $\mathbb{Z}_{N^2}^*$. For our application we additionally demand that the RSA modulus N be sufficiently close to the next power of 2. This is expressed by the condition

$$C(N, s, k) \quad :\Longleftrightarrow \quad 2^{k-1/(s+1)} < N < 2^k \ ,$$

where $s \in \mathbb{N}$ such that $s < 2^{k/2-1}$. Note that the upper bound always holds for RSA moduli sampled from $\mathcal{RSA}_k$. We say that the (τ, ϵ)-*decisional composite residuosity assumption* holds for (s, k) if for all τ-time adversaries $\mathcal{D}$ we have

$$\left| \begin{array}{l} \Pr[N \leftarrow_\$ \mathcal{RSA}_k[C(N, s, k)], y \leftarrow_\$ \mathbb{Z}_{N^2}^*, x \leftarrow y^N \bmod N^2 : \mathcal{D}(N, x) \Rightarrow 1] \\ \quad - \Pr[N \leftarrow_\$ \mathcal{RSA}_k[C(N, s, k)], x \leftarrow_\$ \mathbb{Z}_{N^2}^* : \mathcal{D}(N, x) \Rightarrow 1] \end{array} \right| \leq \epsilon \ .$$

In the probability expressions we write $N \leftarrow_\$ \mathcal{RSA}_k[C]$ for an algorithm that generates a k-bit RSA modulus satisfying condition C.

[10] While this assumption is stronger than the standard Phi-hiding assumption, we conjecture that it is rather mild (possibly in the same way as the strengthened Quadratic Residuosity assumption from [16] that is specialized towards defining the 2^k-th Power Residue symbol).

3 From index-dependence to index-independence

Many natural constructions of lossy trapdoor permutations are index-dependent, i.e., for each index id the function $\mathrm{FEv}(id, \cdot)$ operates on an individual set $\mathcal{X}(id) \subseteq \mathcal{X}$. However, for applications it might be necessary that there is only one domain: $\mathcal{X}(id) = \mathcal{X}$ for all id. In this section we show how to convert index-dependent LTPs into index-independent LTPs. Some transforms of this type have been proposed before. For instance, [12] implicitly uses the somewhat trivial approach of leaving elements in $\mathcal{X} \setminus \mathcal{X}(id)$ untouched (i.e., elements in $\mathcal{X}(id)$ are processed with the LTP, the others are passed through without modification). As discussed in the introduction, the performance of this conversion is generally rather poor: In the worst case, if $|\mathcal{X}(id)| \ll |\mathcal{X}|$, lossiness is bounded by $L = 1$.

The two-round construction we study below was first proposed in [14], however in a different context. There, the goal was to extend the domain of the RSA trapdoor permutation, while aspects of lossiness were not studied. Further, our exposition is more generic. The idea behind the transformation is to ensure that FEv is applied to every point of $\mathcal{X}$ at least once. In both rounds the points of $\mathcal{X}(id)$ are permuted with FEv while the remaining points of $\mathcal{X}$ stay unchanged. To achieve the property stated above, after the first round a permutation π_{id} is used to move into $\mathcal{X}(id)$ all those points that have not yet been touched by FEv.

Let $\mathrm{F} = (\mathrm{FGen}, \mathrm{FEv}, \mathrm{FInv})$ be a LTP with domain $\mathcal{X}$ and index space $\mathcal{I}d$. Assume F has index-dependent domains. For all $id \in \mathcal{I}d$ write $\overline{\mathcal{X}(id)} = \mathcal{X} \setminus \mathcal{X}(id)$ and let $\pi_{id} \colon \mathcal{X} \to \mathcal{X}$ be an efficiently computable and efficiently invertible permutation satisfying $\pi_{id}(\overline{\mathcal{X}(id)}) \subseteq \mathcal{X}(id)$ or, equivalently, $\pi_{id}^{-1}(\overline{\mathcal{X}(id)}) \subseteq \mathcal{X}(id)$. (Note that such a π_{id} can exist only if $|\mathcal{X}(id)| \geq |\mathcal{X}|/2$ for all id.) From F and $(\pi_{id})_{id \in \mathcal{I}d}$ we construct a LTP $\mathrm{F_{ii}} = (\mathrm{FGen_{ii}}, \mathrm{FEv_{ii}}, \mathrm{FInv_{ii}})$ with index-independent domain $\mathcal{X}$, i.e., $\mathcal{X}(id) = \mathcal{X}$ for all id. The algorithms are specified in Figure 1 and illustrated in Figure 2. The analysis is in Lemma 1.

```
$\mathrm{FGen_{ii}}(b)$
$(id, td) \leftarrow_{\$} \mathrm{FGen}(b)$
Return $(id, td)$
```

```
$\mathrm{FEv_{ii}}(id, x)$
If $x \in \mathcal{X}(id)$:
    $x \leftarrow \mathrm{FEv}(id, x)$
$y \leftarrow \pi_{id}(x)$
If $y \in \mathcal{X}(id)$:
    $y \leftarrow \mathrm{FEv}(id, y)$
Return $y$
```

```
$\mathrm{FInv_{ii}}(td, y)$
If $y \in \mathcal{X}(id)$:
    $y \leftarrow \mathrm{FInv}(td, y)$
$x \leftarrow \pi_{id}^{-1}(y)$
If $x \in \mathcal{X}(id)$:
    $x \leftarrow \mathrm{FInv}(td, x)$
Return $x$
```

Fig. 1. Transformation of index-dependent LTP into index-independent LTP. To make algorithm $\mathrm{FInv_{ii}}$ well-defined we assume implicitly that trapdoor td contains a copy of function index id. A visualization of the construction is in Figure 2.

Lemma 1. *Let* F *be a* $(1 - \epsilon_1)$*-correct,* (τ, ϵ_2)*-indistinguishable* L*-lossy trapdoor permutation with index-dependent domain. Furthermore, let* $(\pi_{id})_{id \in \mathcal{I}d}$ *be a family of permutations on* $\mathcal{X}$ *as described. Then* $\mathrm{F_{ii}}$ *is an* $(1 - 2\epsilon_1)$*-correct,* (τ, ϵ_2)*-indistinguishable* $L/2$*-lossy trapdoor permutation with index-independent domain* $\mathcal{X}$*. In particular, if* F *is* 1*-correct, then so is* $\mathrm{F_{ii}}$*.*

Proof. As $\mathrm{F_{ii}}$ uses the same instance generator as F, it inherits to be (τ, ϵ_2)-indistinguishable. We now prove the bound on $\mathrm{F_{ii}}$'s correctness. Let $(id, td) \in [\mathrm{FGen_{ii}}(1)]$ and let $\psi \colon \mathcal{X} \to \mathcal{X}$ be the function with $\psi|_{\mathcal{X}(id)} = \mathrm{FEv}(id, \cdot)$ and $\psi|_{\overline{\mathcal{X}(id)}} = \mathrm{ID}$ (the identity function). Furthermore denote by $\psi_{\mathrm{inv}} \colon \mathcal{X} \to \mathcal{X}$ the function with $\psi_{\mathrm{inv}}|_{\mathcal{X}(id)} = \mathrm{FInv}(td, \cdot)$ and $\psi_{\mathrm{inv}}|_{\overline{\mathcal{X}(id)}} = \mathrm{ID}$ and let $\mathcal{I}_{id} \subseteq \mathcal{X}(id) \subseteq \mathcal{X}$ be the set on which ψ_{inv} fails to invert ψ. By construction $\mathrm{FEv_{ii}}(id, \cdot)$ computes the function $\psi^* := \psi \circ \pi_{id} \circ \psi$ and $\mathrm{FInv_{ii}}(td, \cdot)$ computes the function $\psi^*_{\mathrm{inv}} := \psi_{\mathrm{inv}} \circ \pi_{id}^{-1} \circ \psi_{\mathrm{inv}}$. Note that ψ^*_{inv} only fails to invert ψ^* on some point x if either $x \in \mathcal{I}_{id}$, or if the value y assigned

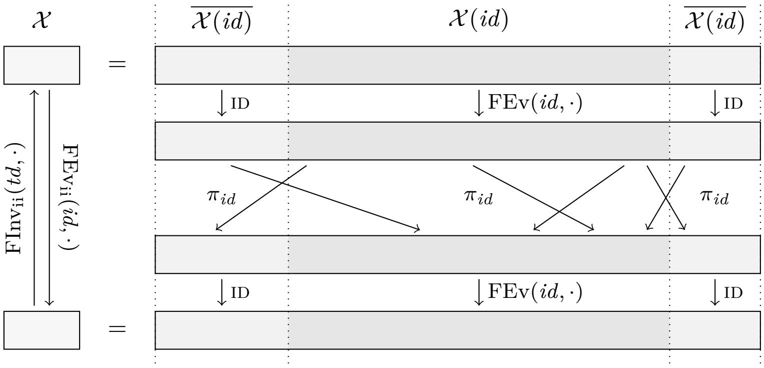

$\mathcal{X}$
$\overline{\mathcal{X}(id)}$
$\mathcal{X}(id)$
$\overline{\mathcal{X}(id)}$
=
ID
$\mathrm{FEv}(id,\cdot)$
ID
π_{id}
π_{id}
π_{id}
ID
$\mathrm{FEv}(id,\cdot)$
ID
=
$\mathrm{FInv}_{\mathrm{ii}}(td,\cdot)$
$\mathrm{FEv}_{\mathrm{ii}}(id,\cdot)$

Fig. 2. Working principle of transformation of index-dependent LTP into index-independent LTP. The corresponding algorithms are in Figure 1. Note that π_{id} is chosen such that every point in $\mathcal{X}$ is permuted by FEv at least once.

in line 04 (of Figure 1) lies in $\mathcal{I}_{id}$. Since $\psi|_{\mathcal{X}\setminus\mathcal{I}_{id}}$ is injective and π_{id} is a permutation, there exist at most $|\mathcal{I}_{id}|$ points with the latter property that are not already elements of $\mathcal{I}_{id}$. Hence $\mathrm{FInv}_{\mathrm{ii}}$ fails to invert $\mathrm{FEv}_{\mathrm{ii}}$ on at most $2|\mathcal{I}_{id}|$ points of $\mathcal{X}$. Summing up for every $(\tilde{id},\tilde{td}) \in [\mathrm{FGen}_{\mathrm{ii}}(1)]$ we have

$$\begin{aligned}&\Pr[(id,td) \leftarrow_{\$} \mathrm{FGen}_{\mathrm{ii}}(1), x \leftarrow_{\$} \mathcal{X}(id), y \leftarrow \mathrm{FEv}_{\mathrm{ii}}(id,x) : \mathrm{FInv}_{\mathrm{ii}}(td,y) \neq x \mid (id,td) = (\tilde{id},\tilde{td})]\\ &\quad\leq 2\Pr[(id,td) \leftarrow_{\$} \mathrm{FGen}(1), x \leftarrow_{\$} \mathcal{X}(id), y \leftarrow \mathrm{FEv}(id,x) : \mathrm{FInv}(td,y) \neq x \mid (id,td) = (\tilde{id},\tilde{td})]\end{aligned}$$

Since both F_{ii} and F use the same index generation algorithm, the bound on F_{ii}'s correctness follows from an application of the law of total probability and F's $(1-\epsilon_1)$-correctness.

We conclude by proving the bound on the lossiness. Assume $id \in \mathcal{I}d$ is created in lossy mode. Since every $y \in \mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \mathcal{X}(id)$ by lines 05 and 06 (of Figure 1) is an element of $\mathrm{FEv}(id,\mathcal{X}(id))$ and id is a lossy index, we obtain

$$|\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \mathcal{X}(id)| \leq |\mathcal{X}(id)|/L \ .$$

Now consider $y \in \mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \overline{\mathcal{X}(id)}$ and let $x \in \mathcal{X}$ with $\mathrm{FEv}_{\mathrm{ii}}(id,x) = y$. Since $y \notin \mathcal{X}(id)$, it is the direct output of π_{id} in the computation of $\mathrm{FEv}_{\mathrm{ii}}(id,x)$, i.e., assigned in line 04 and not overwritten in line 06. Furthermore, since $\pi_{id}^{-1}(\overline{\mathcal{X}(id)}) \subseteq \mathcal{X}(id)$, we have that $\pi_{id}^{-1}(y) \in \mathcal{X}(id)$. Hence, by lines 02 and 03 we obtain $\pi_{id}^{-1}(y) \in \mathrm{FEv}(id,\mathcal{X}(id))$. By the lossiness of id there exist at most $|\mathcal{X}(id)|/L$ values y with this property. Summing up,

$$\begin{aligned}|\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X})| &= |\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \mathcal{X}(id)| + |\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \overline{\mathcal{X}(id)}|\\ &\leq |\mathcal{X}(id)|/L + |\mathcal{X}(id)|/L\\ &\leq 2|\mathcal{X}|/L \ .\end{aligned}$$

This completes the proof. □

Analogously to the construction in Figure 1 it is possible to transform an index-dependent ABO-LTP A = (FGen, FEv, FInv) into an index-independent ABO-LTP $\mathrm{A}_{\mathrm{ii}} = (\mathrm{FGen}, \mathrm{FEv}_{\mathrm{ii}}, \mathrm{FInv}_{\mathrm{ii}})$. Note that A_{ii} uses the same instance generator as A. Algorithms $\mathrm{FEv}_{\mathrm{ii}}$ and $\mathrm{FInv}_{\mathrm{ii}}$ work as their counterparts for LTPs defined in Figure 1, the only difference being the use of the additional input br to evaluate FEv and FInv. We obtain the following.

Lemma 2. *Let* A *be a $(1-\epsilon_1)$-correct, (τ,ϵ_2)-indistinguishable L-lossy ABO-LTP with index-dependent domain. Further, let $(\pi_{id})_{id\in\mathcal{I}d}$ be a family of permutations on $\mathcal{X}$ as described. Then* A_{ii} *is an $(1-2\epsilon_1)$-correct, (τ,ϵ_2)-indistinguishable $L/2$-lossy ABO-LTP with index-independent domain $\mathcal{X}$. In particular, if* A *is 1-correct, then so is* A_{ii}.

4 Lossy trapdoor permutations from Phi-hiding assumption

Fix an RSA modulus N and let $e \ll \varphi_N$ be prime. We say e is *injective* for N if $e \nmid \varphi_N$ and that it is *lossy* for N if $e \mid_1 \varphi_N$. In the injective case the mapping $E\colon \mathbb{Z}_N \to \mathbb{Z}_N;\ x \mapsto x^e$ is inverted by $D\colon y \mapsto y^d$, where d is such that $ed = 1 \bmod \varphi_N$. In the lossy case, the restriction $E|_{\mathbb{Z}_N^*}$ of E to domain $\mathbb{Z}_N^*$ is e-to-1, i.e., we have $|E(\mathbb{Z}_N^*)|/|\mathbb{Z}_N^*| = 1/e$. The Phi-hiding assumption from Section 2.3 then precisely says that it is hard to decide whether a candidate exponent e is injective or lossy for N.

We propose two LTPs in the RSA setting, both with security based on the Phi-hiding assumption. The first construction is quite natural but has index-dependent domains. The second construction is the index-independent analogue of the first, obtained via the transformation from Section 3. Here, our contribution is establishing a better bound on the lossiness than is possible with the generic result. (Our arguments are based on structures specific to the RSA setting.)

4.1 Index-dependent domain LTP from Phi-hiding assumption

Let k be an even number indicating a desired bit length of RSA moduli. Let $\mathcal{E}$ be a distribution of prime numbers such that the (τ, ϵ)-Phi-hiding assumption holds for $(k, \mathcal{E})$. Consider the constructions of LTPs $\mathrm{F} = (\mathrm{FGen}, \mathrm{FEv}, \mathrm{FInv})$ and $\mathrm{F}^* = (\mathrm{FGen}, \mathrm{FEv}^*, \mathrm{FInv})$ given by the algorithms in Figure 3. Observe that condition $e \mid_1 \varphi_N$ in line 02 implies that no element of $\mathcal{E}$ can be longer than $k/2$ bits. Further, to protect from known attacks it is necessary that $\max \mathcal{E} \leq 2^{k/4}$.

The working principle of F is as follows: Function indices id correspond with RSA parameters (N, e). The domain corresponding to index id is $\mathcal{X}(id) = \mathbb{Z}_N$. In injective mode, (N, e) are chosen such that e is invertible modulo φ_N, i.e., such that a corresponding decryption exponent d exists. The FEv and FInv algorithms, in this case, are the standard RSA mappings $x \mapsto x^e$ and $y \mapsto y^d$ (lines 10 and 13). In lossy mode, e is a divisor of φ_N. In this case, mapping $x \mapsto x^e$ is e-to-1 for elements in $\mathbb{Z}_N^*$. The resulting overall lossiness (i.e., for full $\mathbb{Z}_N$) is analyzed in Lemma 3.

We next discuss F^*. This variant achieves better lossiness by building on the fact that given an element of $\mathbb{Z}_N \setminus \mathbb{Z}_N^*$ it is possible to effectively determine whether the function index (N, e) is injective or lossy. In the first case FEv^* uses the standard RSA map; in the second case elements in $\mathbb{Z}_N \setminus \mathbb{Z}_N^*$ are detected and explicitly mapped to 0. The identification of lossy indices and elements in $\mathbb{Z}_N \setminus \mathbb{Z}_N^*$ is handled in lines 17–21. Observe that the condition in line 17 can be checked efficiently.

We analyze constructions F and F^* in Lemma 3.While the second LTP is more complicated to implement, the achieved lossiness bound is easier to work with.

```
FGen($b$)
00 $e \leftarrow_\$ \mathcal{E}$
01 If $b = 0$:    (lossy mode)
02     $(N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[e \mid_1 \varphi_N]$
03     $id \leftarrow (N, e)$; $td \leftarrow \bot$
04 If $b = 1$:    (injective mode)
05     $(N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[e \nmid \varphi_N]$
06     $d \leftarrow e^{-1} \bmod \varphi_N$
07     $id \leftarrow (N, e)$; $td \leftarrow (N, d)$
08 Return $(id, td)$
```

```
FEv($id, x$)
09 $(N, e) \leftarrow id$
10 $y \leftarrow x^e \bmod N$
11 Return $y$

FInv($td, y$)
12 $(N, d) \leftarrow td$
13 $x \leftarrow y^d \bmod N$
14 Return $x$
```

```
FEv*($id, x$)
15 $(N, e) \leftarrow id$
16 If $x = 0$: Return 0
17 If $x \notin \mathbb{Z}_N^*$:
18     $p \leftarrow \gcd(x, N)$
19     $q \leftarrow N/p$
20     $\varphi_N \leftarrow (p-1)(q-1)$
21     If $e \mid \varphi_N$: Return 0
22 $y \leftarrow x^e \bmod N$
23 Return $y$
```

Fig. 3. LTPs F and F^* from Phi-hiding assumption (with index-dependent domains).

Lemma 3. *If for $(k,\mathcal{E})$ the (τ,ϵ)-Phi-hiding assumption holds and $L \leq \min \mathcal{E}$ is a lower bound on the elements in the support of $\mathcal{E}$, LTP* F *is a* 1*-correct,* (τ,ϵ)*-indistinguishable* $(1/L + 2^{-k/2+3})^{-1}$*-lossy trapdoor function. Furthermore, LTP* F^* *is a* 1*-correct* (τ,ϵ)*-indistinguishable* L*-lossy trapdoor function. Both LTPs have index-dependent domain.*

Proof. That F and F^* have (τ,ϵ)-indistinguishable injective and lossy modes follows directly from the definition of the Phi-hiding assumption for $(k,\mathcal{E})$. The 1-correctness of F is clear. To see that also F^* is 1-correct observe that line 16 of Figure 3 together with the fact that the condition of line 21 is never met (in injective mode) ensure that also $\mathrm{FEv}^*(id,\cdot)$ implements precisely the (bijective) RSA exponentiation mapping.

We proceed with showing the bound on F's lossiness. Let (N,e) be a lossy index and p,q denote the prime divisors of N. As pointed out above, $E|_{\mathbb{Z}_N^*}$ is e-to-1. Since $|\mathbb{Z}_N^*| = \varphi_N$,

$$|E(\mathbb{Z}_N)| = |E(\mathbb{Z}_N^*)| + |E(\mathbb{Z}_N \setminus \mathbb{Z}_N^*)| \leq |E(\mathbb{Z}_N^*)| + |\mathbb{Z}_N \setminus \mathbb{Z}_N^*| = \frac{\varphi_N}{e} + p + q - 1 < \frac{N}{e} + p + q \ .$$

We assume w.l.o.g. that $p < \sqrt{N} < q$. Since both p and q are $k/2$-bit primes, we have $q < 2p < 2\sqrt{N}$. Hence $p + q \leq 3\sqrt{N}$, which yields $(p+q)/N \leq 3/\sqrt{N} \leq 2^{-k/2+3}$. Summing up,

$$|\mathrm{FEv}(id,\mathbb{Z}_N)| < N(1/e + 2^{-k/2+3}) \leq |\mathbb{Z}_N| \left(1/L + 2^{-k/2+3}\right) \ ,$$

which establishes the bound.

We next assess the lossiness of F^*. Let $(id,\perp) \leftarrow_\$ \mathrm{FGen}(0)$ with $id = (N,e)$. Then, when restricted to $\mathbb{Z}_N^*$, function $\mathrm{FEv}^*(id,\cdot)$ computes $\mathbb{Z}_N^* \to \mathbb{Z}_N^*; x \mapsto x^e$ which is e-to-1. Further, all values $x \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$ are mapped to 0: For $x = 0$ this follows by line 16, and for $x \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$ with $x \neq 0$, lines 18, 19, 20 recover p, q, φ_N (p and q not necessarily in this order), so that it can be confirmed in line 21 that the LTP is operated in lossy mode and value 0 can be output. Summing up,

$$|\mathrm{FEv}^*(id,\mathbb{Z}_N)| = \varphi(N)/e + 1 \leq \varphi(N)/e + (p+q-1)/e = N/e \leq |\mathbb{Z}_N|/L \ .$$

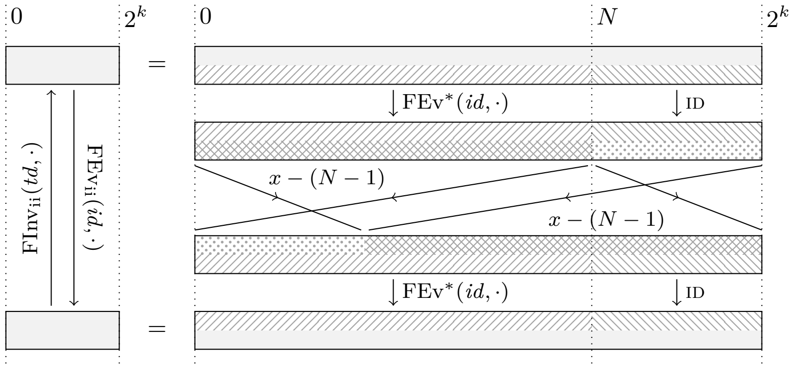
□

4.2 Index-independent domain LTP from Phi-hiding assumption

The LTP F^* from Section 4.1 has index-dependent domains: for function index $id = (N,e)$, algorithm $\mathrm{FEv}^*(id,\cdot)$ operates on domain $\mathcal{X}(id) = \mathbb{Z}_N$. By construction we have $N \in [\![2^{k-1} \mathinner{..} 2^k]\!]$ and thus $\mathcal{X}(id) \subseteq \mathcal{X}$ for $\mathcal{X} = [\![2^k]\!]$. To obtain an LTP $\mathrm{F_{ii}}$ with index-independent domain $[\![2^k]\!]$ we can apply to F^* the generic transform of Section 3. By Lemma 1, assuming appropriately chosen permutations (π_{id}), if F^* is L-lossy, then $\mathrm{F_{ii}}$ is $L/2$-lossy. The contribution of the current section is to show that for a specifically defined family (π_{id}) using direct (non-generic) arguments this result can be strengthened: If F^* is L-lossy, then also $\mathrm{F_{ii}}$ is L-lossy. In other words, there is no price to pay for switching from index-dependent domains to index-independent domains. (This holds for the lossiness; computation time might double.)

As a first step we identify a family (π_{id}) of permutations on $\mathcal{X}$ that suits the conditions of the transform from Section 3, namely $\pi_{id}(\overline{\mathcal{X}(id)}) \subseteq \mathcal{X}(id)$ for all $id \in \mathcal{I}d$. Hence let $\mathcal{X} = [\![2^k]\!]$ and $(N,e) = id \in \mathcal{I}d$, where $N \in [\![2^{k-1} \mathinner{..} 2^k]\!]$. We define $\pi_{id}\colon \mathcal{X} \to \mathcal{X}; x \mapsto x - (N-1) \bmod 2^k$. Then π_{id} is a permutation on $\mathcal{X}$ and we have $N \leq x < 2^k \Rightarrow 1 \leq \pi_{id}(x) \leq 2^k - N < N$ (the last inequality follows from $2^{k-1} \leq N < 2^k$); this establishes $\pi_{id}(\overline{\mathcal{X}(id)}) \subseteq \mathcal{X}(id)$. We illustrate the transform from Figure 2 in conjunction with this family of bijections (π_{id}) in Figure 4. In the following, we first state the generic result obtained by applying Lemma 1 to this setup. We then give the one established directly.

Corollary 1. *Let $\mathcal{E}$ be a prime distribution and $L \leq \min \mathcal{E}$. Let $\mathrm{F}^* = (\mathrm{FGen}, \mathrm{FEv}^*, \mathrm{FInv})$ be the L-lossy LTP defined in Figure 3, $(\pi_{id})_{id \in \mathcal{I}d}$ the permutation family defined above, and F_{ii} the conversion of F^* via Figure 1. If for $(k, \mathcal{E})$ the (τ, ϵ)-Phi-hiding assumption holds, then F_{ii} is a 1-correct, (τ, ϵ)-indistinguishable $L/2$-lossy trapdoor function with index-independent domain $\mathcal{X} = [\![2^k]\!]$.*



0
2^k
0
N
2^k
=
$\mathrm{FEv}^*(id, \cdot)$
ID
$x - (N-1)$
$x - (N-1)$
$\mathrm{FEv}^*(id, \cdot)$
ID
=
$\mathrm{FInv}_{\mathrm{ii}}(td, \cdot)$
$\mathrm{FEv}_{\mathrm{ii}}(id, \cdot)$

Fig. 4. Illustration of Phi-hiding based LTP F_{ii} with index-independent domain.

Lemma 4. *Let $\mathcal{E}$ be a prime distribution and $L \leq \min \mathcal{E}$. Let $\mathrm{F}^* = (\mathrm{FGen}, \mathrm{FEv}^*, \mathrm{FInv})$ be the L-lossy LTF defined in Figure 3, $(\pi_{id})_{id \in \mathcal{I}d}$ the permutation family defined above, and F_{ii} the conversion of F^* via Figure 1. If for $(k, \mathcal{E})$ the (τ, ϵ)-Phi-hiding assumption holds, then F_{ii} is a 1-correct, (τ, ϵ)-indistinguishable $L/(1 + 2^{-k/2})$-lossy trapdoor function with index-independent domain $\mathcal{X} = [\![2^k]\!]$.*

Proof. The (τ, ϵ)-indistinguishability of injective and lossy mode follows with Lemma 1. We show F_{ii}'s improved lossiness by bounding, for any lossy index $id = (N, e)$, the quantity $|\mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X})|/|\mathcal{X}|$. We partition $\mathcal{X}$ into the sets $\mathcal{X}(id)$ and $\overline{\mathcal{X}(id)}$, obtaining

$$|\mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X})| = |\mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X}) \cap \mathcal{X}(id)| + |\mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X}) \cap \overline{\mathcal{X}(id)}| \ .$$

As in the proof of Lemma 1 we have

(a) $|\mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X}) \cap \mathcal{X}(id)| \leq |\mathcal{X}(id)|/L = N/L$.

(b) For all $y \in \mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X}) \cap \overline{\mathcal{X}(id)}$ there exists $x \in \mathcal{X}(id)$ such that $\pi_{id}^{-1}(y) = \mathrm{FEv}^*(id, x)$.

Note that by our choice of π_{id} we have $\pi_{id}^{-1}(\overline{\mathcal{X}(id)}) = [2N - 1 - 2^k \mathinner{..} N - 1]$. Hence by (b) we have $|\mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X}) \cap \overline{\mathcal{X}(id)}| = |\mathrm{FEv}^*(id, \mathcal{X}(id)) \cap [2N - 1 - 2^k \mathinner{..} N - 1]|$. Let p, q be the prime factors of N. W.l.o.g. we may assume $e \mid_1 (p-1)$ and $e \nmid (q-1)$. We claim the following.

Claim.

$$|\mathrm{FEv}^*(id, \mathcal{X}(id)) \cap [2N - 1 - 2^k \mathinner{..} N - 1]| \leq (2^k - N + p)/L$$

Combining the claim with (a) we obtain

$$|\mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X})| \leq N/L + (2^k - N + p)/L = (2^k + p)/L \leq (2^k + 2^{k/2})/L \ .$$

Dividing by $|\mathcal{X}| = 2^k$ yields the lemma.

It remains to prove the claim. Let $E\colon \mathbb{Z}_N^* \to \mathbb{Z}_N^*; x \mapsto x^e$ and $E_p\colon \mathbb{Z}_p^* \to \mathbb{Z}_p^*; x \mapsto x^e$. Since $e \mid_1 p$, the map E_p is e-to-1. Hence $e \geq L$ implies $|E_p(\mathbb{Z}_p^*)|/|\mathbb{Z}_p^*| \leq 1/L$. Let $y \in \mathbb{Z}_N^*$ be arbitrary. Calculating modulo p we obtain that $y \in E(\mathbb{Z}_N^*)$ implies $(y \bmod p) \in E_p(\mathbb{Z}_p^*)$. Note that for any $x \in \mathbb{Z}_p^*$ there are at most $\lceil (2^k - N)/p \rceil$ many numbers $y \in [2N-1-2^k \mathinner{..} N-1]$ satisfying $x \equiv y \bmod p$. Furthermore, $|E_p(\mathbb{Z}_p^*)| < p/L$. Using that $\mathrm{FEv}^*(id, x) = 0$ for all $x \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$ we obtain

$$\begin{aligned}
|\mathrm{FEv}^*(id, \mathcal{X}(id)) \cap [2N-1-2^k \mathinner{..} N-1]| &= |E(\mathbb{Z}_N^*) \cap [2N-1-2^k \mathinner{..} N-1]| \\
&= \sum_{x \in E_p(\mathbb{Z}_p^*)} |\{y \in [2N-1-2^k \mathinner{..} N-1] \mid y \equiv x \bmod p\}| \\
&< p/L \cdot \lceil (2^k - N)/p \rceil \\
&\leq (2^k - N + p)/L \ .
\end{aligned}$$

This is the statement of the claim. □

5 Lossy trapdoor permutations from Quadratic residuosity assumption

In this section we recall the index-dependent lossy trapdoor function F of [11] based on the quadratic residuosity assumption and show how the transform of Section 3 can be used to obtain an index-independent variant F_{ii}. Since F has a lossiness factor of 2, using the generic bound is of no use in this case. However, by exploiting the algebraic structure of the construction we are able to establish that F_{ii} has essentially the same lossiness factor as F. This improves on the index-independent variant given in [11], which achieves a lossiness factor of 4/3.

5.1 Index-dependent domain LTP from Quadratic residuosity assumption

Let p, q be primes of bit length $k/2$ satisfying $p \equiv 3 \bmod 4$ and $q \equiv 3 \bmod 4$. Consider the functions $j_N\colon \mathbb{Z} \to \{0,1\}$ and $h_N\colon \mathbb{Z} \to \{0,1\}$ defined by

$$j_N(x) = \begin{cases} 0, & \text{if } x \in \mathcal{J}_N \cup (\mathbb{Z}_N \setminus \mathbb{Z}_N^*) \\ 1, & \text{if } x \in \mathbb{Z}_N^* \setminus \mathcal{J}_N \end{cases}$$

$$h_N(x) = \begin{cases} 0, & \text{if } x \leq N/2 \\ 1, & \text{if } x > N/2 \end{cases} .$$

Note that both j_N and h_N can be efficiently computed given N. Let $d_j, d_h \in \{0,1\}$. Then —as pointed out in [11]— for each $y \in \mathcal{QR}_N$ exactly one of the four solutions of the equation $x^2 = y \bmod N$ satisfies $j_N(x) = d_j$ and $h_N(x) = d_h$. We denote this square root of y by R_{d_j,d_h}. Furthermore for every $y \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$ with $y \in \mathcal{QR}_p \vee y \in \mathcal{QR}_q$ the equation $y = x^2 \bmod N$ has exactly two solutions —one being the negative of the other. Hence both solutions satisfy $j_N(x) = 0$ and for $d_h \in \{0,1\}$ exactly one of the solutions satisfies $h_N(x) = d_h$. Analogous to the situation above we denote this solution by R_{0,d_h}. In [11] the authors construct a lossy trapdoor permutation with index-dependent domain $\mathbb{Z}_N$. The LTP's algorithms are depicted in Figure 5. The idea of the construction is to map elements $x \in \mathbb{Z}_N$ to x^2, which is afterwards multiplied by some appropriately chosen group elements, which allow to reconstruct x^2 as well as both $j_N(x) =: d_j$ and $h_N(x) =: d_h$. Then the LTF can be inverted by computing R_{d_j,d_h}.

Lemma 5 ([11]). *Let* $\mathrm{F} = (\mathrm{FGen}, \mathrm{FEv}, \mathrm{FInv})$ *the LTP of Figure 5. If the* (τ, ϵ)*-Quadratic residuosity assumption holds for* k*, then* F *is an* (τ', ϵ)*-indistinguishable 2-lossy trapdoor function with index-dependent domain* $\mathcal{X}((N,r,s)) = \mathbb{Z}_N \subseteq [\![2^k]\!] = \mathcal{X}$*, where* $\tau' \approx \tau$*.*

```
FGen($b$)
$(N,p,q) \leftarrow_{\$} \mathcal{BRSA}_k$
$r \leftarrow_{\$} \mathbb{Z}_N^* \setminus \mathcal{J}_N$
If $b = 0$:    (lossy mode)
    $s \leftarrow_{\$} \mathcal{QR}_N$
    $id \leftarrow (N,r,s)$; $td \leftarrow \bot$
If $b = 1$:    (injective mode)
    $s \leftarrow_{\$} \mathcal{J}_N \setminus \mathcal{QR}_N$
    $id \leftarrow (N,r,s)$
    $td \leftarrow (N,p,q,r,s)$
Return $(id, td)$
```

```
FEv($id, x$)
$(N,r,s) \leftarrow id$
$d_j \leftarrow j_N(x)$
$d_h \leftarrow h_N(x)$
$y \leftarrow x^2 r^\iota s^\tau$
Return $y$
```

```
FInv($td, y$)
If $y = 0$: return 0
$(N,p,q,r,s) \leftarrow td$
$d_j \leftarrow j_N(y)$
$y' \leftarrow y r^{-d_j}$
If $y' \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$:
    If $y' \notin \mathcal{QR}_p \wedge y' \notin \mathcal{QR}_q$:
        $d_h \leftarrow 1$
    Else: $d_h \leftarrow 0$
Elseif $y' \in \mathcal{QR}_N$: $d_h \leftarrow 0$
Else: $d_h \leftarrow 1$
$y'' \leftarrow y' s^{-d_h}$
$x \leftarrow R_{d_j,d_h}(y'')$
Return $x$
```

Fig. 5. LTP F from Quadratic residuosity assumption (with index-dependent domains).

5.2 Index-independent domain LTP from Quadratic residuosity assumption

In [11] the authors propose to modify the LTP F of Section 5.1 in the following way to obtain an LTP F_{ii}^* with index-independent domain $[\![2^k]\!]$. This is done by letting $\mathrm{F}_{ii}^*|_{\mathcal{X}(id)}(id,\cdot) := \mathrm{F}(id,\cdot)$ and $\mathrm{F}_{ii}^*|_{\overline{\mathcal{X}(id)}}(id,\cdot) := \textsc{id}$. The resulting LTP F_{ii}^* is a 4/3-lossy trapdoor function. In this section we show that using our transformation of Section 3 with an appropriate permutation yields an index-independent LTP based on the quadratic residuosity assumption having essentially the same lossiness factor as the underlying LTP F.

To be able to use our transformation we need a family of permutations on $\mathcal{X}$ that suits the conditions of Section 3. Since —as in the construction based on the Phi-hiding assumption— the index-dependent domain $\mathcal{X}(id)$ for some $id = (N,r,s)$ is $\mathbb{Z}_N$, we are able to use the same family of permutations. Hence for $id = (N,r,s) \in \mathcal{I}d$ define $\pi_{id} \colon \mathcal{X} \to \mathcal{X}; x \mapsto x - (N-1) \bmod 2^k$. Then π_{id} is a permutation on $\mathcal{X}$ and as in Section 4.2 we obtain $\pi_{id}(\overline{\mathcal{X}(id)}) \subseteq \mathcal{X}(id)$.

Note that applying Lemma 1 would only yield a bound of $2/2 = 1$ on the lossiness factor of the transformed LTP, which is of no use. However, we are able to establish a desirable result directly using techniques similar to the ones used in the proof of Lemma 4.

Lemma 6. *Let* $\mathrm{F} = (\mathrm{FGen}, \mathrm{FEv}, \mathrm{FInv})$ *be the* 1*-correct,* 2*-lossy LTP defined in Figure 5,* $(\pi_{id})_{id \in \mathcal{I}d}$ *the permutation family defined above, and* F_{ii} *the transformation of* F *via Figure 1. If the* (τ,ϵ)*-Quadratic residuosity assumption holds for* k*,* F_{ii} *is a* 1*-correct* (τ',ϵ)*-indistinguishable* $2/(1+2^{-k/2})$*-lossy trapdoor function with index-independent domain* $\mathcal{X} = [\![2^k]\!]$*, where* $\tau' \approx \tau$*.*

Proof. Correctness and (τ',ϵ)-indistinguishability of injective mode and lossy mode follows with Lemma 1. We prove the bound on F_{ii}'s lossiness by bounding, for any lossy index $id = (N,r,s)$, the quantity $|\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X})|/|\mathcal{X}|$. We partition $\mathcal{X}$ into the sets $\mathcal{X}(id)$ and $\overline{\mathcal{X}(id)}$, obtaining

$$|\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X})| = |\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \mathcal{X}(id)| + |\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \overline{\mathcal{X}(id)}| \ .$$

As in the proof of Lemma 1 we have

(a) $|\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \mathcal{X}(id)| \leq |\mathcal{X}(id)|/2 = N/2$.

(b) For all $y \in \mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \overline{\mathcal{X}(id)}$ there exists $x \in \mathcal{X}(id)$ such that $\pi_{id}^{-1}(y) = \mathrm{FEv}(id,x)$.

Note that by our choice of π_{id} we have $\pi_{id}^{-1}(\overline{\mathcal{X}(id)}) = [2N-1-2^k \mathinner{..} N-1]$. Hence by (b) we have $|\mathrm{FEv}_{\mathrm{ii}}(id,\mathcal{X}) \cap \overline{\mathcal{X}(id)}| = |\mathrm{FEv}(id,\mathcal{X}(id)) \cap [2N-1-2^k \mathinner{..} N-1]|$. Let p,q be the prime factors of N and let $r_p := r \bmod p$ and $r_q := r \bmod q$. By definition we have that $r \in \mathbb{Z}_N^* \setminus \mathcal{J}_N$. This implies that either $r_p \in \mathcal{QR}_p$ or $r_q \in \mathcal{QR}_q$. W.l.o.g. we assume $r_p \in \mathcal{QR}_p$. We claim the following.

Claim.

$$|\mathrm{FEv}^*(id, \mathcal{X}(id)) \cap [2N-1-2^k \,..\, N-1]| \le (2^k - N + p + 1)/2$$

Combining the claim with (a) we obtain

$$|\mathrm{FEv}_{\mathrm{ii}}(id, \mathcal{X})| \le N/2 + (2^k - N + p + 1)/2 = (2^k + p + 1)/2 \le (2^k + 2^{k/2})/2 \ .$$

Dividing by $|\mathcal{X}| = 2^k$ yields the lemma.

It remains to prove the claim. Let $y \in \mathrm{FEv}(\mathbb{Z}_N)$ and denote $y \bmod p$ by y_p. Then by definition of FEv there exists $x \in \mathbb{Z}_N^*$ such that $y = x^2 r^{j(x)} s^{h(x)}$. Since both r and s are squares modulo p this implies that either $y_p \in \mathcal{QR}_p$ or $y_p = 0$. Hence y_p takes one of at most $(p+1)/2$ different values. Note that for any $x \in \mathbb{Z}_p$ there are at most $\lceil (2^k - N)/p \rceil$ many numbers $y \in [2N-1-2^k \,..\, N-1]$ satisfying $x \equiv y \bmod p$. We obtain

$$\begin{aligned}
&|\mathrm{FEv}^*(id, \mathcal{X}(id)) \cap [2N-1-2^k \,..\, N-1]| \\
\le &\sum_{x \in \mathcal{QR}_p \cup \{0\}} |\{y \in [2N-1-2^k \,..\, N-1] \mid y \equiv x \bmod p\}| \\
< &(p+1)/2 \cdot \lceil (2^k - N)/p \rceil \\
\le &(2^k - N + p + 1)/2 \ .
\end{aligned}$$

This is the statement of the claim. □

6 Lossy trapdoor permutations from decisional composite residuosity assumption

In this section we recall the lossy trapdoor function with index-dependent domain introduced in [11] based on the Damgård–Jurik encryption scheme [10] and show how to obtain an index-independent variant using the transformation of Section 3. The resulting index-independent LTP achieves a slightly larger lossiness factor than the index-independent variant given in [11].

6.1 Index-dependent domain LTP from decisional composite residuosity assumption

In [11] the authors give the construction of a LTP with index-dependent domains based on the decisional composite residuosity assumption. In this section we give a slightly modified version of this result, aiming for easier applicability of our transform of Section 3. While [11] work with domain $\mathcal{X}(id) = \mathbb{Z}_{N^s} \times \mathbb{Z}_N^*$ for some RSA modulus N, we expand the domains to $\mathbb{Z}_{N^{s+1}}$, which we identify via a bijection with $\mathbb{Z}_{N^s} \times \mathbb{Z}_N \supseteq \mathbb{Z}_{N^s} \times \mathbb{Z}_N^*$. The main working principle of the construction is the following result.

Theorem 1 ([10]). *Let $k \in \mathbb{N}$ be even and $N \in [\mathcal{RSA}_k]$. Furthermore let $s \in \mathbb{N}$ such that $s < 2^{k/2-1}$. Then the map*

$$\Psi_{N,s} : \begin{cases} \mathbb{Z}_{N^s} \times \mathbb{Z}_N^* & \to \mathbb{Z}_{N^{s+1}}^* \\ (x_1, x_2) & \mapsto (1+N)^{x_1} x_2^{N^s} \end{cases}$$

is an efficiently computable isomorphism of abelian groups. Furthermore given $\lambda(N) = lcm(p-1, q-1)$ the map $\Psi_{N,s}$ can be efficiently inverted.

Let $N \in \mathbb{N}$. We define a bijection ϕ between $\mathbb{Z}_{N^s} \times \mathbb{Z}_N$ and $\mathbb{Z}_N^{s+1}$ by mapping (x_1, x_2) to $x_1 + x_2 N^s$. Consider the LTP defined in Figure 6. Then the following holds.

```
FGen($b$)
00 $(N, \lambda_N) \leftarrow_\$ \mathcal{RSA}_k[C(N,s,k)]$
01 $r \leftarrow_\$ \mathbb{Z}_N^*$
02 If $b = 0$:   (lossy mode)
03    $c \leftarrow r^{N^s} \bmod N^{s+1}$
04    $td \leftarrow \perp$
05 If $b = 1$:   (injective mode)
06    $c \leftarrow (1+N)r^{N^s} \bmod N^{s+1}$
07    $td \leftarrow (\lambda_N, r, id)$
08 $id \leftarrow (N, c)$
09 Return $(id, td)$

FEv($id, x$)
10 $(N, c) \leftarrow id$
11 $(x_1, x_2) \leftarrow \phi^{-1}(x)$
12 If $x_2 \notin \mathbb{Z}_N^*$:
13    return 0
14 $y \leftarrow c^{x_1} x_2^{N^s} \bmod N^{s+1}$
15 Return $y$

FInv($td, y$)
16 $(\lambda_N, r, id) \leftarrow td$
17 $(N, c) \leftarrow id$
18 If $y \notin \mathbb{Z}_{N^{s+1}}^*$:
19    return 0
20 $(x_1, z) \leftarrow \Psi_s^{-1}(y)$
21 $x_2 \leftarrow zr^{-x_1}$
22 $x \leftarrow \phi(x_1, x_2)$
23 Return $x$
```

Fig. 6. LTP F from decisional composite residuosity assumption (with index-dependent domains). The expression $C(N, s, k)$ denotes the condition that $2^{k-1/(s+1)} < N < 2^k$.We write $(N, \lambda_N) \leftarrow_\$ \mathcal{RSA}_k[C]$ for an algorithm that generates a k-bit RSA modulus satisfying condition C and also outputs the value $\lambda_N = \mathrm{lcm}(p-1, q-1)$ of the Carmichael function evaluated at position N.

Theorem 2 ([11]). *Let $k \in \mathbb{N}$ be even and $s \in \mathbb{N}$ such that $s < 2^{k/2-1}$. Consider the LTP* F *defined in Figure 6. If the (τ, ϵ, s)-decisional composite residuosity assumption holds for k, then* F *is a $(1 - 2^{-k/2+3})$-correct, (τ', ϵ)-indistinguishable $2^{(k-1)s-1}$-lossy trapdoor function with index-dependent domain $\mathcal{X}((N, c)) = \mathbb{Z}_{N^{s+1}} \subseteq \llbracket 2^{k(s+1)} \rrbracket = \mathcal{X}$, where $\tau' \approx \tau$.*

Proof. A proof of the (τ', ϵ)-indistinguishability and the bound on F's lossiness may be found in [11]. Let $(id, td) \in [\mathrm{FGen}(1)]$, where $id = (N, c)$. Further let $x \in \mathcal{X}(id)$ and $(x_1, x_2) = \phi(x)$. By Theorem 1 FInv fails to invert FEv on x only if $x_2 \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$. For uniformly chosen x the probability of this occurring is bounded by $|\mathbb{Z}_N \setminus \mathbb{Z}_N^*|/|\mathbb{Z}_N| = (p + q - 1)/N < 2^{-k/2+3}$, where the inequality was shown in the proof of Lemma 3. This establishes $(1 - 2^{k/2})$-correctness. □

In [11] the authors extend their composite residuosity based LTP construction to an ABO-LTP A. A has branching set $\mathcal{B}r = \llbracket 2^{k/2-1} \rrbracket$, and index-dependent domains $\mathcal{X}(id) = \mathbb{Z}_{N^{s+1}}^*$, where N is an RSA modulus that is part of id. Analogously to our construction from above $\mathcal{X}(id)$ can be extended to $\mathbb{Z}_{N^{s+1}}$.

6.2 Index-independent domain LTP from decisional composite residuosity assumption

In this section we show how the LTP F of Section 6.1 can be transformed into an LTP $\mathrm{F_{ii}}$ with index-independent domain $\mathcal{X} = \llbracket 2^{k(s+1)} \rrbracket$ using the technique of Section 3. To be able to apply the transform of Section 3 we first construct an appropriate family of permutations on $\llbracket 2^{k(s+1)} \rrbracket$. Hence, for $id = (N, c) \in \mathcal{I}d$ let $\pi_{id} : \llbracket 2^{k(s+1)} \rrbracket \to \llbracket 2^{k(s+1)} \rrbracket; x \mapsto x - N^{(s+1)} \bmod 2^{k(s+1)}$. Since according to line 00 of Figure 6 the modulus N is chosen such that $2^{k(s+1)-1} \leq N^{(s+1)} < 2^{k(s+1)}$, we have that $x - N^{(s+1)} \in \llbracket N^{(s+1)} \rrbracket \bmod 2^{k(s+1)}$ holds for every $x \in \llbracket 2^{k(s+1)} \rrbracket \setminus \llbracket N^{(s+1)} \rrbracket = \llbracket N^{(s+1)} \mathinner{..} 2^{k(s+1)} \rrbracket$. This implies $\pi_{id}(\overline{\mathcal{X}(id)}) \subseteq \mathcal{X}(id)$. Hence the family of permutations $(\pi_{id})_{id \in \mathcal{I}d}$ has the necessary property to apply Lemma 1 and we obtain the following.

Corollary 2. *Let $k \in \mathbb{N}$ be even and $s \in \mathbb{N}$ such that $s < 2^{k/2-1}$. Furthermore let* F *be the LTP defined in Figure 6 and* $\mathrm{F_{ii}}$ *the LTP transformation of* F *via Figure 1 with respect to the family of permutations $(\pi_{id})_{id \in \mathcal{I}d}$ defined above. If the (τ, ϵ)-decisional composite residuosity assumption holds for k, then* $\mathrm{F_{ii}}$ *is a $(1 - 2^{-k/2+3})$-correct, (τ', ϵ)-indistinguishable $2^{(k-1)s-2}$-lossy trapdoor permutation with index-independent domain $\mathcal{X} = \llbracket 2^{k(s+1)} \rrbracket$, where $\tau' \approx \tau$.*

This result carries over to the composite residuosity based ABO-LTP with index-dependent domains of [11] described in the previous section.

Comparison to the construction of [11]. In [11] the authors propose a different way to modify the LTP F of Figure 6 to obtain an index-independent variant. This is done by restricting the domain of F to $[\![2^{(k-1)s+k/2-1}]\!]$, which is a subset of $\mathcal{X}(id) = [\![N^{k(s+1)}]\!]$ for every $id = (N, c) \in \mathcal{I}d$. The resulting index-independent LTP $\mathrm{F}^*_{\mathrm{ii}}$ has a lossiness factor of $2^{(k-1)s-k/2-1}$. Since $\mathrm{F}^*_{\mathrm{ii}}$'s has domain $[\![2^{(k-1)s+k/2-1}]\!]$ and range $[\![2^{k(s+1)}]\!]$ it is a lossy trapdoor *function*, while our transform applied to F yields a lossy trapdoor *permutation* F_{ii} working over a larger domain, having slightly better lossiness. However, this comes at the cost of potentially having to evaluate the underlying LTP F twice per evaluation of F_{ii}.

7 Prime family generators

In Section 8 we construct all-but-one lossy trapdoor permutations from the unique divisor Phi-hiding assumption. As a building block we use prime family generators, a tool that deterministically derives prime numbers from a randomly picked seed. While this concept already appeared in [9], we need a variant of the tool with different functionality and security properties. Below, we first define syntax and functionality of prime family generators, and then give a construction based on polynomial evaluation.

Let $\mathcal{Q} \subseteq \mathcal{P}$ be a finite set of prime numbers and let $L \leq |\mathcal{Q}|$. For $(\mathcal{Q}, L)$, any instance of a *prime family generator* (PFG) indicates a sequence of distinct primes $q_1, \ldots, q_L \in \mathcal{Q}$. A specific programmability feature allows for embedding any given prime at any given position. Formally, an (ϵ_1, ϵ_2)-PFG for $(\mathcal{Q}, L)$ consists of a seed space $\mathcal{S}d$ and three algorithms PGen, PGet, PProg such that

$$\mathrm{PGen} \to_\$ \mathcal{S}d \quad \text{and} \quad \mathcal{S}d \times [L] \to \mathrm{PGet} \to \mathcal{Q} \quad \text{and} \quad [L] \times \mathcal{Q} \to \mathrm{PProg} \to_\$ \mathcal{S}d \ .$$

For functionality we demand (a) programmability: for all $i \in [L]$ we require

$$\Pr[q \leftarrow_\$ \mathcal{Q}; sd \leftarrow_\$ \mathrm{PProg}(i, q) : \mathrm{PGet}(sd, i) \neq q] \leq \epsilon_1 \ .$$

(b) distinctness of outputs: for all $i \in [L]$ we require

$$\Pr[sd \leftarrow_\$ \mathrm{PGen} : \exists j \in [L], i \neq j : \mathrm{PGet}(sd, i) = \mathrm{PGet}(sd, j)] \leq \epsilon_2 \ .$$

For security we require perfectly indistinguishable programmability: We demand that for all $i \in [L]$ and every distinguisher $\mathcal{D}$ (running in arbitrary time) we have

$$\left| \begin{array}{l} \Pr[sd \leftarrow_\$ \mathrm{PGen} : \mathcal{D}(sd) \Rightarrow 1] \\ \quad - \Pr[q \leftarrow_\$ \mathcal{Q}; sd \leftarrow_\$ \mathrm{PProg}(i, q) : \mathcal{D}(sd) \Rightarrow 1] \end{array} \right| = 0 \ .$$

7.1 Construction based on polynomial evaluation

The PFG we construct here outputs (l-bit) primes from $\mathcal{Q} = \mathcal{P}_l$. While the construction is similar to one by [9], their PFG would also output primes shorter than l bits. Further, our analysis of probabilities is different, for being tailored towards our application: the construction of ABO-LTPs.

Concretely, for a set of chosen parameters $l, n, d, \lambda \in \mathbb{N}$ we construct a $(2^{-(\lambda+1)}, 2^{-\lambda})$-PFG for $\mathcal{Q} = \mathcal{P}_l$ and $L = 2^n$. The construction is based on a family $\{F_{sd}\}$ of d-wise independent hash functions and, roughly, works as follows (see Figure 7). The PFG's seed space $\mathcal{S}d$ is equal to $\{F_{sd}\}$'s key space. For $sd \in \mathcal{S}d$ and $i \in [2^n]$, natural numbers are generated by evaluating F_{sd} at up to $d/2$ distinct points. $\mathrm{PGet}(sd, i)$'s output is the first prime found. Since numbers

```
PGen
$sd \leftarrow_{\$} \{0,1\}^{d(l-1)}$
Return $sd$

PGet($sd, i$)
For $j \leftarrow 1$ to $d/2$:
    $q \leftarrow F_{sd}(\#i \| \#j)$
    If $q + 2^{l-1} \in \mathcal{P}_l$:
        Return $q + 2^{l-1}$
Return $\perp$

PProg($i, q$)
$(a_1, \ldots, a_d) \leftarrow_{\$} [\![ 2^{l-1} \mathinner{..} 2^l ]\!]^d$
Find smallest $j$ with $a_j \in \mathcal{P}_l$
    $a_j \leftarrow q$
$sd \leftarrow \mathrm{FindC}(i, a_1 - 2^{l-1}, \ldots, a_d - 2^{l-1})$
Return $sd$
```

Fig. 7. PFG based on polynomial evaluation

of bit length l are tested for primality, the prime number theorem guarantees that PGen will succeed in finding a prime on average after roughly l attempts. Furthermore, if d is chosen large enough finding a prime in this way will succeed except with some negligible error probability. Concretely, we instantiate $\{F_{sd}\}$ with polynomial evaluation of degree d over the field $\mathrm{GF}(2^{l-1})$. Programming a prime q into a particular point i is done by sampling a sequence of d-many values a_j in the image of F_{sd}. Then —if existent— the first prime in this sequence is replaced by q. By polynomial interpolation it is possible to find a seed sd such that F_{sd} evaluated at the j'th point equals a_j. The technical challenge is to prove that if q was a uniformly distributed prime then then resulting seed sd has the correct distribution and, furthermore, with high probability satisfies $q = \mathrm{PGet}(sd, i)$.

We now specify the construction in detail. We start by imposing necessary restrictions on its parameters. Let $l, n, d, \lambda \in \mathbb{N}$ with d even and $l \geq 25$. We require

$$n \leq l - \lambda - \log_2(l) - 2 \tag{3}$$

$$2l(\lambda + 1)/\log_2(e) \leq d < 2^{l-1-n} \tag{4}$$

where e is Euler's number. The first inequality ensures that the probability of two primes sampled uniformly from $\mathcal{P}_l$ colliding is small, the second inequality makes sure that d on one hand is large enough that PGet finds a prime with high probability and on the other hand small enough, that numbers smaller than d can be encoded with few bits. Note that for $l = \mathcal{O}(\lambda)$ and $n = l/2$ equation (3) will typically be fulfilled and results in d being of order $\mathcal{O}(\lambda^2)$. The family of hash functions used in our construction is defined as follows. For $sd \in \mathrm{GF}(2^{l-1})^d$ let

$$F_{sd} \colon \{0,1\}^{l-1} \to [\![2^{l-1}]\!];\ x \mapsto \textstyle\sum_{k=0}^{d-1} sd_k x^k \ .$$

Here x is interpreted as element of $\mathrm{GF}(2^{l-1})$. Note that the function family $(F_{sd})_{sd \in \mathcal{S}d}$ is a d-wise independent hash function [27]. Finally, we define an algorithm FindC as follows. FindC receives as input a tuple $(i, a_1, \ldots, a_d)$, where $i \in [\![2^n]\!]$ and $a_1, \ldots, a_d \in [\![2^{l-1}]\!]$. It then uses Lagrange interpolation to find $sd_0, \ldots, sd_{d-1} \in \mathrm{GF}(2^{l-1})$ such that $F_{sd}(\#i \| \#j) = a_j$ for all $j \in [d]$, where $sd := (sd_0, \ldots, sd_{d-1})$ (see Section 2.1 for the # notation). Here we assume $\#j \in \{0,1\}^{l-1-n}$, which is possible since by equation 4 we have $j \leq d < 2^{l-1-n}$. FindC's output is sd. Note that for every $i \in [\![2^n]\!]$ the function implemented by $\mathrm{FindC}(i, \cdot)$ is a bijection between $[\![2^{l-1}]\!]^d$ and $\mathcal{S}d$. The description of the PFG P may be found in Figure 7.

Note that in the definition we formally do not allow PGet to return elements that are not in $\mathcal{Q}$. However, P returns $\perp$ if after d tests no prime has been found. This issue could be solved by letting PGet return some fixed prime $q \in \mathcal{Q}$ in this case. We obtain the following result.

Theorem 3. *Let $l, n, d, \lambda \in \mathbb{N}$ as defined above. Then $\mathrm{P} = (\mathrm{PGen}, \mathrm{PGet}, \mathrm{PProg})$ as defined in Figure 7 is a $(2^{-(\lambda+1)}, 2^{-\lambda})$-PFG for $(\mathcal{P}_l, 2^n)$ with seed space $\mathcal{S}d = \mathrm{GF}(2^{l-1})^d$.*

The theorem is due to the following three propositions establishing indistinguishable programmability, distinctness of outputs and programmability respectively.

Proposition 1. *Let $i \in [2^n]$ and $q \leftarrow_\$ \mathcal{P}_l$. Then* PProg *on input (i, q) returns a uniformly distributed $sd \in \mathcal{S}d$.*

Proposition 2. *Let $i \in [2^n]$ and $sd \leftarrow_\$$* PGen*. For $j \in [2^n]$ let $p_j \leftarrow \mathrm{PGet}(sd, j)$. Then*

$$\Pr[\exists j \in [2^n], j \neq i : p_i = \bot \vee p_i = p_j] \leq 2^{-\lambda} \ .$$

Proposition 3. *Let $i \in [\![2^i]\!]$ and $q \leftarrow_\$ \mathcal{P}_l$. Then*

$$\Pr[sd \leftarrow_\$ \mathrm{PProg}(i, q); q' \leftarrow \mathrm{PGet}(sd, i) : q \neq q'] \leq 2^{-(\lambda+1)} \ .$$

Proof (of Theorem 3). Proposition 1 shows that for all $i \in [\![2^n]\!]$ and all distinguishers $\mathcal{D}$

$$|\Pr[sd \leftarrow_\$ \mathrm{PGen} : \mathcal{D}(sd) \Rightarrow 1] - \Pr[q \leftarrow_\$ \mathcal{P}_l; sd \leftarrow_\$ \mathrm{PProg}(i, q) : \mathcal{D}(sd) \Rightarrow 1| = 0 \ ,$$

since in both cases sd has the same distribution. Furthermore, by Propositions 2 and 3 the PFG has distinctness of output error and programmability error bounded by $2^{-\lambda}$ and $2^{-(\lambda+1)}$ respectively. □

Proof (of Proposition 1). The values $a_1, \ldots, a_d$ sampled in line 07 of Figure 7 are independent and uniformly distributed on the set $[\![2^{l-1} \mathinner{..} 2^l]\!]$. Assume that $\{a_1, \ldots, a_d\}$ contains a prime and let j' denote the smallest number such that $a_{j'}$ is prime. Then $a_{j'}$ is uniformly distributed on $\mathcal{P}_l$. Hence replacing $a_{j'}$ with the uniformly sampled $q \leftarrow_\$ \mathcal{P}_l$, as done in line 09, does not change the distribution of the a_j. If $\{a_1, \ldots, a_d\}$ does not contain a prime, all a_j remain unchanged. Summing up, in line 10 all values a_j are independent and uniformly distributed on $[\![2^{l-1} \mathinner{..} 2^l]\!]$. Hence $(a_1 - 2^{l-1}, \ldots, a_{d-1} - 2^{l-1})$ is uniformly distributed on $[\![2^{l-1}]\!]^d$. Since $\mathrm{FindC}(i, \cdot)$ computes a bijection, the resulting sd is uniformly distributed on $\mathrm{GF}(2^{l-1})^d$.

□

Proof (of Proposition 2). Let $i \in [2^n]$. Note that $p_i \neq \bot$ by line 04 of Figure 7 implies that p_i is a l-bit prime. The probability of a uniformly distributed element z of $[\![2^{l-1} \mathinner{..} 2^l]\!]$ being prime can be lower bounded by $\Pr[z \in \mathcal{P}_l] \geq 1/l$. (For $l \geq 25$ this is easily derived from [1, Corollary 1].) Furthermore, $(1 - 1/l)^l \leq e^{-1}$. During one execution of PGet up to $d/2$ distinct expressions of the form $F_{sd}(\#i \| \#j)$ are checked on primality. Since F_{sd} is a d-wise independent hash function, we obtain

$$\Pr[p_i = \bot] \leq (1 - 1/l)^{d/2} \leq e^{-d/(2l)} = 2^{-\log_2(e)d/(2l)} \leq 2^{-(\lambda+1)} \ ,$$

where the last inequality is due to equation 4. Let $j \in [2^n]$ such that $j \neq i$ and let $\mathit{\Pi}$ denote the event that $p_i \neq \bot$. Assume $\mathit{\Pi}$ occurs. We have $|\mathcal{P}_l| \geq 2^{l-1}/l$. (For $l \geq 25$ this is easily derived from [1, Corollary 1].) Since F_{sd} is a d-wise independent hash function and both p_i, p_j were computed using at most $d/2$ evaluations of F_{sd} at different points, p_i is uniformly distributed on $\mathcal{P}_l$ and independent of p_j, which yields $\Pr[p_i = p_j \mid \mathit{\Pi}] \leq |\mathcal{P}_l|^{-1} \leq 2^{-l+1+\log_2(l)}$. By the union bound we obtain

$$\Pr[\exists j \in [2^n], j \neq i : p_i = p_j \mid \mathit{\Pi}] \leq 2^{n-l+1+\log_2(l)} \leq 2^{-(\lambda+1)} \ ,$$

where the last inequality is due to equation 3. Summing up

$$\begin{aligned}\Pr[\exists j \in [2^n], j \neq i : p_i = \bot \vee p_i = p_j] &\leq \Pr[p_i = \bot] + \Pr[\exists j \in [2^n], j \neq i : p_i = p_j \mid \mathit{\Pi}] \\ &\leq 2^{-(\lambda+1)} + 2^{-(\lambda+1)} \\ &\leq 2^{-\lambda} \ ,\end{aligned}$$

which establishes the proposition. □

```
$\mathrm{FGen}(br^*)$
$sd \leftarrow_\$ \mathrm{PGen}$
$e^* \leftarrow_\$ \mathrm{PGet}(sd, br^*)$
$(N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[C(e^*, \varphi_N)]$
$id \leftarrow (N, sd)$
$td \leftarrow (N, sd, \varphi_N)$
Return $(id, td)$
```

```
$\mathrm{FEv}(br, id, x)$
If $x = 0$: Return 0
$(N, sd) \leftarrow id$
$e \leftarrow \mathrm{PGet}(sd, br)$
If $x \notin \mathbb{Z}_N^*$:
    $p \leftarrow \gcd(x, N)$
    $q \leftarrow N/p$
    $\varphi_N \leftarrow (p-1)(q-1)$
    If $e \mid \varphi_N$: Return 0
$y \leftarrow x^e \bmod N$
Return $y$
```

```
$\mathrm{FInv}(br, td, y)$
$(N, sd, \varphi_N) \leftarrow td$
$e \leftarrow \mathrm{PGet}(sd, br)$
If $\gcd(e, \varphi_N) \neq 1$:
    Return $\perp$
$d \leftarrow e^{-1} \bmod \varphi_N$
$x \leftarrow y^d \bmod N$
Return $x$
```

Fig. 8. ABO from Phi-hiding assumption. $C(e^*, \varphi_N)$ denotes the condition defined in Section 2.3.

Proof (of Proposition 3). Let $sd \leftarrow_\$ \mathrm{PProg}(i, q)$. Assume that the subset $\{a_1, \ldots, a_{d/2}\}$ of values sampled in line 07 of Figure 7 contains a prime. In this case the first prime $a_{j'}$ in the sequence is replaced by q. Then in line 10 a value sd is chosen such that $F_{sd}(\#i \| \#j') + 2^{l-1} = q$ and $F_{sd}(\#i \| \#j) + 2^{l-1} = a_j$ for $j \neq j'$. Since a_j' is the smallest prime in the sequence, $\mathrm{PGet}(sd, i)$ in this case returns q. Hence

$$\begin{aligned}&\Pr[sd \leftarrow_\$ \mathrm{PProg}(i, q); q' \leftarrow \mathrm{PGet}(sd, i) : q \neq q'] \\ =&\Pr[sd \leftarrow_\$ \mathrm{PProg}(i, q); q' \leftarrow \mathrm{PGet}(sd, i) : q' = \perp]\ .\end{aligned}$$

Since $q \leftarrow_\$ \mathcal{P}_l$, we know by Proposition 1 that sd is uniformly distributed on $\mathcal{S}d$. Hence as in the proof of Proposition 2 we have $\Pr[sd \leftarrow_\$ \mathrm{PProg}(i, q); q' \leftarrow \mathrm{PGet}(sd, i) : q' = \perp] \leq 2^{-\log_2(e)d/(2l)} \leq 2^{-(\lambda+1)}$, where the last inequality is due to equation 4. □

8 An ABO-LTP with index-independent domain from unique-divisor Phi-hiding

We use a prime family generator (for instance the one defined in Section 7) to construct an ABO-LTP with index-independent domain, which can be shown secure under the unique-divisor Phi-hiding assumption. The construction resembles [17, Section 5.2] who build an adaptive trapdoor function. As a starting point we first specify an ABO-LTP A having index-dependent domains. Using the transform from Section 3, A can be made index-independent. Due to the result of Lemma 4 the transformed ABO-LTP has essentially the same lossiness factor as A.

8.1 An index-dependent ABO-LTP from unique-divisor Phi-hiding

Let $n, l \in \mathbb{N}$ and P a prime family generator for $(\mathcal{P}_l, 2^n)$. Consider the ABO-LTP defined in Figure 8. The construction has branch space $\mathcal{B}r = [2^n]$. Indices for lossy branch $br^* \in [2^n]$ consist of a seed sd and an RSA modulus N sampled such that the br^*-th prime e^* in the sequence determined by sd is the only prime in $\mathcal{P}_l$ that divides φ_N. To compute $\mathrm{FEv}(br, id, x)$ for some $x \in \mathcal{X}(id) = \mathbb{Z}_N$, first the prime e corresponding to br is retrieved using PGet. Then the image of x is computed as in the Phi-hiding based LTF construction of Section 4 using index (N, e). By definition e^* is the only prime in $\mathcal{P}_l$ that divides φ_N. Hence, for every branch in $\mathcal{B}r \setminus \{br^*\}$ the construction defines a permutation, which can be efficiently inverted given trapdoor $td = \varphi_N$. We obtain the following result.

Lemma 7. *Let $n, l \in \mathbb{N}$ and let* $\mathrm{P} = (\mathrm{PGen}, \mathrm{PGet}, \mathrm{PProg})$ *be a (ϵ_1, ϵ_2)-PFG for $(\mathcal{P}_l, 2^n)$. Consider* $\mathrm{A} = (\mathrm{FGen}, \mathrm{FEv}, \mathrm{FInv})$ *as defined in Figure 8. If the unique-divisor (τ, ϵ)-Phi-hiding assumption holds for $(k, \mathcal{P}_l)$, A is a $(1-\epsilon_2)$-correct, L-lossy, $(\tau', 2(\epsilon+\epsilon_1)/(1-\epsilon_1))$-indistinguishable*

```
$\mathcal{D}'_{br_0,br_1}(e^*,N)$
$d \leftarrow_\$ \{0,1\}$
$sd \leftarrow_\$ \mathrm{PProg}(br_d, e^*)$
$id \leftarrow (N, sd)$
$d' \leftarrow_\$ \mathcal{D}_{br_0,br_1}(id)$
If $d = d'$: Return 1
Else: Return 0
```

Fig. 9. Adversary for the proof of Lemma 7

ABO-LTP with index-dependent domain $\mathcal{X} = [\![2^k]\!]$, *where* $L = 2^{l-1}$ *and* $\tau' \approx \tau$. *Furthermore* A *has branching set* $\mathcal{B}r = [2^n]$.

Proof. We start by establishing the bound on A's correctness. Let $br^* \in \mathcal{B}r$ and $id = (N, sd)$, where $(id, td) \leftarrow_\$ \mathrm{FGen}(br^*)$. Since according to line 02 of Figure 8 the RSA modulus N is sampled such that condition $C(e^*, \varphi_N)$ holds, the only prime of bit length l that divides φ_N is $e^* \leftarrow \mathrm{PGet}(sd, br^*)$. Let $br \in \mathcal{B}r$ and $e \leftarrow \mathrm{PGet}(sd, br)$. If $e^* \neq e$ then $\gcd(e, \varphi_N) = 1$ and by Lemma 3 the function implemented by $\mathrm{FEv}(br, id, \cdot)$ is injective and inverted by $\mathrm{FInv}(br, td, \cdot)$. Therefore an inversion error may only occur if $e = e^*$. Hence by distinctness of outputs the probability that there exists a branch $br \neq br^*$ such that $\mathrm{FInv}(br, td, \cdot)$ does not invert $\mathrm{FEv}(br, id, \cdot)$ on all points of $\mathcal{X}(id)$ is bounded by ϵ_2. This establishes $(1-\epsilon_2)$-correctness.

Concerning the bound on A's lossiness note that N is sampled such that $e^* \mid_1 \varphi_N$. Furthermore all primes in $\mathcal{P}_l$ are greater than 2^{l-1}. Now the bound follows as in Lemma 3.

It remains to prove that the construction is $(\tau', 2(\epsilon + \epsilon_1)/(1 - \epsilon_1))$-indistinguishable. For $br_0, br_1 \in \mathcal{B}r$ let $\mathcal{D}_{br_0,br_1}$ be an adversary against the indistinguishability of lossy branches. Furthermore denote its advantage in distinguishing function indexes sampled with respect to br_0 from indexes sampled with respect to br_1 by

$$\mathbf{Adv}^{\text{abo-ind}}(\mathcal{D}_{br_0,br_1}) := \left| \begin{array}{l} \Pr[(id, td) \leftarrow_\$ \mathrm{FGen}(br_0) : \mathcal{D}_{br_0,br_1}(id) \Rightarrow 1] \\ \quad - \Pr[(id, td) \leftarrow_\$ \mathrm{FGen}(br_1) : \mathcal{D}_{br_0,br_1}(id) \Rightarrow 1] \end{array} \right| .$$

Consider adversary $\mathcal{D}'_{br_0,br_1}$of Figure 9, which expects input as in the unique divisor Phi-hiding game. Note that its running time is essentially the running time of $\mathcal{D}_{br_0,br_1}$ plus some minor bookkeeping. It receives as input (e^*, N), samples bit d and seed $sd \leftarrow_\$ \mathrm{PProg}(br_d, e^*)$, sets $id \leftarrow (N, sd)$ and runs $\mathcal{D}_{br_0,br_1}$ on input id. If the bit d' returned by $\mathcal{D}_{br_0,br_1}$ equals d it returns 1, else it returns 0. We now analyze $\mathcal{D}'_{br_0,br_1}$'s advantage in the unique divisor Phi-hiding game, which we denote by

$$\mathbf{Adv}^{\text{phi-hid}}_{br_0,br_1}(\mathcal{D}'_{br_0,br_1}) :=$$
$$\left| \begin{array}{l} \Pr[e_0 \leftarrow_\$ \mathcal{E}; (N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[C(\mathcal{E}, \varphi_N, e_0)] : \mathcal{D}'_{br_0,br_1}(N, e_0) \Rightarrow 1] \\ \quad - \Pr[e_0, e_1 \leftarrow_\$ \mathcal{E}; (N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[C(\mathcal{E}, \varphi_N, e_0)] : \mathcal{D}'_{br_0,br_1}(N, e_1) \Rightarrow 1] \end{array} \right| .$$

Let D_0 denote the event $\{e_0 \leftarrow_\$ \mathcal{E}; (N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[C(\mathcal{E}, \varphi_N, e_0)] : \mathcal{D}'_{br_0,br_1}(N, e_0) \Rightarrow 1\}$ and D_1 the event $\{e_0, e_1 \leftarrow_\$ \mathcal{E}; (N, \varphi_N) \leftarrow_\$ \mathcal{RSA}_k[C(\mathcal{E}, \varphi_N, e_0)] : \mathcal{D}'_{br_0,br_1}(N, e_1) \Rightarrow 1\}$ and furthermore denote by Π the event that during an execution of $\mathcal{D}'_{br_0,br_1}(E^*, N)$ a programming error occurs, i.e. that $e^* \neq e \leftarrow \mathrm{PGet}(sd, br^*)$. We rewrite $\mathcal{D}'_{br_0,br_1}$'s advantage as

$$\begin{aligned} \mathbf{Adv}^{\text{phi-hid}}_{br_0,br_1}(\mathcal{D}'_{br_0,br_1}) &= |(\Pr[D_0 \mid \neg\Pi] - \Pr[D_1 \mid \neg\Pi]) \Pr[\neg\Pi] + (\Pr[D_0 \mid \Pi] - \Pr[D_1 \mid \Pi]) \Pr[\Pi]| \\ &\geq (1 - \epsilon_1)|\Pr[D_0 \mid \neg\Pi] - \Pr[D_1 \mid \neg\Pi]| - \epsilon_1 \ , \end{aligned} \tag{5}$$

where the inequality follows from P's ϵ_1-programmability. We now analyze $|\Pr[D_0 \mid \neg\Pi] - \Pr[D_1 \mid \neg\Pi]|$.

First consider $\Pr[D_1 \mid \neg\Pi]$. In this case $\mathcal{D}'_{br_0,br_1}$'s input is (e^*, N), where e^* is independent of N. In this case $id = (N, sd)$ is equally distributed for both $d = 0$ and $d = 1$. Hence d is independent of $\mathcal{D}_{br_0,br_1}$'s input $id = (N, sd)$, which implies that in this case the bit d' returned by $\mathcal{D}_{br_0,br_1}$ equals d with probability 1/2. We obtain

$$\Pr[D_1 \mid \neg\Pi] = 1/2 \ . \tag{6}$$

Now consider $\Pr[D_0 \mid \neg\Pi]$. In this case $\mathcal{D}'_{br_0,br_1}$'s input is (e^*, N), where $e^* \mid_1 \varphi_N$. Since we conditioned on $\neg\Pi$ and since P has perfectly indistinguishable programmability, if $d = 0$ the index id set up by $\mathcal{D}'_{br_0,br_1}$ in line 02 is distributed as the output of $\mathrm{FGen}(br_0)$. On the other hand, if $d = 1$ then id is distributed as $\mathrm{FGen}(br_1)$. This yields

$$\begin{aligned}\Pr[D_0 \mid \neg\Pi] &= 1/2\left(\Pr[d = d' \mid d = 1, \neg\Pi] + \Pr[d = d' \mid d = 0, \neg\Pi]\right)\\ &= 1/2\left(\Pr[\mathcal{D}_{br_0,br_1}(id) \Rightarrow 1 \mid d = 1, \neg\Pi] + 1 - \Pr[\mathcal{D}_{br_0,br_1}(id) \Rightarrow 1 \mid d = 0, \neg\Pi]\right)\\ &= \pm 1/2\mathbf{Adv}^{\text{abo-ind}}(\mathcal{D}_{br_0,br_1}) + 1/2 \ . \end{aligned} \tag{7}$$

Combining equations 5, 6 and 7 yields

$$\begin{aligned}\mathbf{Adv}^{\text{phi-hid}}_{br_0,br_1}(\mathcal{D}'_{br_0,br_1}) &\geq (1 - \epsilon_1)|\pm\mathbf{Adv}^{\text{abo-ind}}(\mathcal{D}_{br_0,br_1})/2 + 1/2 - 1/2| - \epsilon_1\\ &= \frac{(1 - \epsilon_1)}{2}\mathbf{Adv}^{\text{abo-ind}}(\mathcal{D}_{br_0,br_1}) - \epsilon_1\end{aligned}$$

Now the claim follows from the unique divisor (τ, ϵ)-Phi-hiding assumption. □

8.2 An index-independent ABO-LTP from unique-divisor Phi-hiding

Using the technique from Section 3 it is possible to transform A into an index-independent ABO-LTP $\mathrm{A_{ii}}$. Using the improved bound on the lossiness from Lemma 4 we obtain the following.

Corollary 3. *Let $n, l, k \in \mathbb{N}$ and $\mathrm{A} = (\mathrm{FGen}, \mathrm{FEv}, \mathrm{FInv})$ be the ABO defined in Figure 8. Further, for $(N, sd) = id \in \mathcal{I}d$ let π_{id} the permutation*

$$\pi_{id} \colon [\![2^k]\!] \to [\![2^k]\!]; x \mapsto (x - N + 1) \bmod N \ .$$

Let $\mathrm{A_{ii}}$ be the conversion of A via Figure 1. If the unique-divisor (τ, ϵ)-Phi-hiding assumption holds for $(k, \mathcal{P}_l)$, $\mathrm{A_{ii}}$ is a $(1-2\epsilon_2)$-correct, L-lossy, $(\tau', 2(\epsilon+\epsilon_1)/(1-\epsilon_1)))$-indistinguishable index-independent ABO-LTP with domain $[\![2^k]\!]$ and branching set $[2^n]$, where $L = 2^{l-1}/(1 + 2^{-k/2})$ and $\tau' \approx \tau$.

Acknowledgments

Benedikt Auerbach was supported in part by the NRW Research Training Group SecHuman and by ERC Project ERCC (FP7/615074). Bertram Poettering conducted part of the research at Ruhr University Bochum, supported by ERC Project ERCC (FP7/615074). Eike Kiltz was supported in part by ERC Project ERCC (FP7/615074) and by DFG SPP 1736 Big Data.

References

1. Barkley Rosser, J., Schoenfeld, L.: Approximate formulas for some functions of prime numbers. Ill. J. Math 6, 64–94 (1962)
2. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged public-key encryption: How to protect against bad randomness. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 232–249. Springer, Heidelberg (Dec 2009)
3. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: Definitional equivalences and constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (Aug 2008)
4. Bellare, M., Hoang, V.T.: Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 627–656. Springer, Heidelberg (Apr 2015)
5. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (Apr 2009)
6. Benhamouda, F., Herranz, J., Joye, M., Libert, B.: Efficient cryptosystems from 2^k-th power residue symbols. Journal of Cryptology 30(2), 519–549 (Apr 2017)
7. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (Aug 2008)
8. Brakerski, Z., Segev, G.: Better security for deterministic public-key encryption: The auxiliary-input setting. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 543–560. Springer, Heidelberg (Aug 2011)
9. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (May 1999)
10. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (Feb 2001)
11. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (May 2010)
12. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. Journal of Cryptology 26(1), 39–74 (Jan 2013)
13. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
14. Hayashi, R., Okamoto, T., Tanaka, K.: An RSA family of trap-door permutations with a common domain and its applications. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 291–304. Springer, Heidelberg (Mar 2004)
15. Hohenberger, S., Waters, B.: Short and stateless signatures from the RSA assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 654–670. Springer, Heidelberg (Aug 2009)
16. Joye, M., Libert, B.: Efficient cryptosystems from 2^k-th power residue symbols. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 76–92. Springer, Heidelberg (May 2013)
17. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (May / Jun 2010)
18. Kiltz, E., O'Neill, A., Smith, A.: Instantiability of RSA-OAEP under chosen-plaintext attack. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 295–313. Springer, Heidelberg (Aug 2010)
19. Mironov, I., Pandey, O., Reingold, O., Segev, G.: Incremental deterministic public-key encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 628–644. Springer, Heidelberg (Apr 2012)
20. Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 296–311. Springer, Heidelberg (May 2010)
21. Nishimaki, R., Fujisaki, E., Tanaka, K.: Efficient non-interactive universally composable string-commitment schemes. In: Pieprzyk, J., Zhang, F. (eds.) ProvSec 2009. LNCS, vol. 5848, pp. 3–18. Springer, Heidelberg (Nov 2009)
22. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (May 1999)
23. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 187–196. ACM Press (May 2008)
24. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. SIAM J. Comput. 40(6), 1803–1844 (2011), http://dx.doi.org/10.1137/080733954

25. Raghunathan, A., Segev, G., Vadhan, S.P.: Deterministic public-key encryption for adaptively chosen plaintext distributions. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 93–110. Springer, Heidelberg (May 2013)
26. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (Mar 2009)
27. Shoup, V.: A Computational Introduction to Number Theory and Algebra. Cambridge University Press (2005)
28. Xie, X., Xue, R., Zhang, R.: Efficient threshold encryption from lossy trapdoor functions. In: Yang, B.Y. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011. pp. 163–178. Springer, Heidelberg (Nov / Dec 2011)
29. Yamakawa, T., Yamada, S., Hanaoka, G., Kunihiro, N.: Adversary-dependent lossy trapdoor function from hardness of factoring semi-smooth RSA subgroup moduli. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 3–32. Springer, Heidelberg (Aug 2016)