Domain Extension of Public Random Functions: Beyond the Birthday Barrier*

Ueli Maurer and Stefano Tessaro

Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland
{maurer,tessaros}@inf.ethz.ch

Abstract. A *public* random function is a random function that is accessible by all parties, including the adversary. For example, a (public) random oracle is a public random function $\{0,1\}^* \to \{0,1\}^n$. The natural problem of constructing a public random oracle from a public random function $\{0,1\}^m \to \{0,1\}^n$ (for some $m > n$) was first considered at Crypto 2005 by Coron et al. who proved the security of variants of the Merkle-Damgård construction against adversaries issuing up to $\mathcal{O}(2^{n/2})$ queries to the construction and to the underlying compression function. This bound is less than the square root of $n2^m$, the number of random bits contained in the underlying random function.

In this paper, we investigate domain extenders for public random functions approaching optimal security. In particular, for all $\epsilon \in (0,1)$ and all functions m and ℓ (polynomial in n), we provide a construction $\mathbf{C}_{\epsilon,m,\ell}(\cdot)$ which extends a public random function $\mathbf{R} : \{0,1\}^n \to \{0,1\}^n$ to a function $\mathbf{C}_{\epsilon,m,\ell}(\mathbf{R}) : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ with time-complexity polynomial in n and $1/\epsilon$ and which is secure against adversaries which make up to $\Theta(2^{n(1-\epsilon)})$ queries. A central tool for achieving high security are special classes of unbalanced bipartite expander graphs with small degree. The achievability of practical (as opposed to complexity-theoretic) efficiency is proved by a non-constructive existence proof.

Combined with the iterated constructions of Coron et al., our result leads to the first iterated construction of a hash function $\{0,1\}^* \to \{0,1\}^n$ from a component function $\{0,1\}^n \to \{0,1\}^n$ that withstands all recently proposed generic attacks against iterated hash functions, like Joux's multi-collision attack, Kelsey and Schneier's second-preimage attack, and Kelsey and Kohno's herding attacks.

1 Introduction

1.1 Secret vs. Public Random Functions

Primitives that provide some form of randomness are of central importance in cryptography, both as a primitive assumed to be given (e.g. a secret key), and as a primitive constructed from a weaker one to "behave like" a certain ideal random primitive (e.g. a random function), according to some security notion.

* This research was partially supported by the Swiss National Science Foundation (SNF), project no. 200020-113700/1.

A. Menezes (Ed.): CRYPTO 2007, LNCS 4622, pp. 187–204, 2007.

An adversary may have different types of access to a random primitive. The two extreme cases are that the adversary has *no access* and that he has *complete access*[1] to it. For example, the adversary is assumed to have no access to a secret key, and a pseudo-random function (PRF) is a (computationally-secure) realization from such a secret key of a secret random function to which the adversary has no access. In contrast, a (public) random oracle, as used in the so-called random-oracle model [7], is a function $\{0,1\}^* \to \{0,1\}^n$ to which the adversary has *complete* access, like the legitimate parties. Similarly, a *public parameter* (e.g. the parameter selecting a hash function from a class) is a finite random string to which the adversary has complete access. It is natural to also consider finite-domain public random functions.

In this paper we are interested in such *public* random primitives and reductions among them. The question whether (and how) a certain primitive can be securely realized from another primitive is substantially more complex in the public setting, compared to the secret setting, and even the security notion is more involved. For example, while the CBC-construction can be seen as the secure realization of a secret random function $\{0,1\}^* \to \{0,1\}^n$ from a secret random function $\{0,1\}^n \to \{0,1\}^n$ [5,19], the same statement is false if public functions (accessible to the adversary) are considered. Another famous example of a reduction problem for public primitives is the realization of a (public) random oracle from a public parameter. This was shown to be impossible [8,21].

1.2 Domain Extension and the Birthday Barrier

A random primitive (both secret or public) can be characterized by the number of random bits it contains. An ℓ-bit key is a string (or table) containing ℓ random bits, a random function $\{0,1\}^m \to \{0,1\}^n$ corresponds to a table of $n2^m$ random bits which can be accessed efficiently, and a random oracle $\{0,1\}^* \to \{0,1\}^n$ corresponds to a countably infinite table of random bits.[2] Of course, a random table of N bits can be interpreted as a random function $\{0,1\}^m \to \{0,1\}^n$ for any m and n with $n2^m \leq N$. For example, n can be doubled at the apparently minor expense of reducing m by 1.

An important topic in cryptography is the secure expansion of such a table, considered as an ideal system. This is referred to as *domain extension*, say from $\{0,1\}^m$ to $\{0,1\}^{2m}$ (or to $\{0,1\}^*$), which corresponds to an exponential (or even infinite) blow-up of the table size. (In contrast, increasing the range, say from $\{0,1\}^n$ to $\{0,1\}^{2n}$, corresponds to merely a doubling of the table size.)

[1] Side-channel attack analyses, where part of the secret key is assumed to leak, are examples of intermediate scenarios.

[2] Each bit can be accessed in time logarithmic in its position in the table, which is optimal since the specification of the position requires logarithmically many bits. In this paper we only consider such random primitives where the bits can be accessed efficiently, but there are also more complicated primitives, like an ideal cipher, which on one hand has a special permutation structure and also allows on the other hand a special additional type of access, namely inverse queries.

In [21] a generalization of indistinguishability to systems with public access, called *indifferentiability*, was proposed. Like for indistinguishability, there is a computational and a stronger, information-theoretic, version of indifferentiability. This general notion allows to discuss the secure realization of a *public* random primitive from another public random primitive. In [21] also a simple general framework was proposed, based on entropy arguments, for proving impossibility results like that of [8]. One can easily show that not even a single-bit extension of a public parameter, from ℓ to $\ell+1$ bits, is possible, let alone to an exponentially large table (corresponding to a public random function $\{0,1\}^m \rightarrow \{0,1\}^n$) or even to an infinite table (corresponding to the impossibility of realizing a random oracle [8,21]).

However, the situation is different if one starts from a public random function (as opposed to just a public random string). Coron et al. [11] considered the problem of constructing a random oracle $\{0,1\}^* \rightarrow \{0,1\}^n$ from a public random function $\{0,1\}^m \rightarrow \{0,1\}^n$ (where $m > n$) and showed that a modified Merkle-Damgård construction [24,12] works, with information-theoretic security (i.e., indifferentiability) up to about $\mathcal{O}(2^{n/2})$ queries. This bound, only the square root of $\mathcal{O}(2^n)$, is usually called the "birthday barrier". The term "birthday" is used because the birthday paradox applies (as soon as two different inputs to the function occur which produce the same output, security is lost) and the term "barrier" is used because breaking it is non-trivial if at all possible.

For *secret* random functions, many constructions in the literature, also those based on universal hashing [9,26] and the CBC-construction [5,19], suffer from the birthday problem, and hence several researchers [1,4,19] considered the problem of achieving security beyond the birthday barrier. The goal of this paper is to solve the corresponding problem for public random functions. Namely, we want to achieve essentially maximal security, i.e., up to $\Theta(2^{n(1-\epsilon)})$ queries for any $\epsilon > 0$ (where the construction may depend on ϵ). Like for other problems (see e.g. [13]), going from the "secret case" to the "public case" appears to involve substantial new construction elements and analysis techniques.

1.3 Significance of Domain Extension for Public Random Functions

The domain extension problem for public random functions has important implications for the design of cryptographic functions, in addition to being of general theoretical interest. We also refer to [11] for a discussion of the significance of this problem.

Cryptographic functions with arbitrary input-length are of crucial importance in cryptography. Desirable properties for such functions are collision-resistance, second-preimage resistance, multi-collision resistance, being pseudo-random, or being a secure MAC, etc. A general paradigm for constructing a cryptographic function $\{0,1\}^* \rightarrow \{0,1\}^n$, both in the secret and the public case, is to make use of a component function $\mathbf{F} : \{0,1\}^m \rightarrow \{0,1\}^n$ and to embed it into an iterated construction $\mathbf{C}(\cdot)$ (e.g. the CBC or the Merkle-Damgård construction), resulting in the overall function $\mathbf{C}(\mathbf{F}) : \{0,1\}^* \rightarrow \{0,1\}^n$.

It is important to be able to separate the reasoning about the component function $\mathbf{F}$ and about the construction $\mathbf{C}(\cdot)$. Typically, $\mathbf{F}$ is simply assumed to have some property, like being collision-resistant, second-preimage resistant, a secure MAC, etc. In contrast, the construction $\mathbf{C}(\cdot)$ is (or should be!) designed in a way that one can *prove* certain properties.

There are two types of such proofs for $\mathbf{C}(\cdot)$. The first type is a complexity-theoretic reduction proof showing that if there exists an adversary breaking a certain property of $\mathbf{C}(\mathbf{F})$, then there exists a comparably efficient adversary breaking a property (the same or a different one) of $\mathbf{F}$. For example, using such an argument one can prove that the Merkle-Damgård [24,12] construction is collision-resistant if the component function is. Similarly, one can prove that the CBC construction is a PRF if the component function is [5], or that certain constructions [2,22] are secure MACs if the component function is.

A second type of proof, which is the subject of [11] and of this paper, is the proof that if $\mathbf{F}$ is a public random function, then so is $\mathbf{C}(\mathbf{F})$, up to a certain number B of queries. Such a proof implies the absence of a generic (black-box) attack against $\mathbf{C}(\mathbf{F})$, i.e., an attack which does not exploit specific properties of $\mathbf{F}$, but uses it merely as a black-box.[3] Such a generic proof is not an ultimate security proof for $\mathbf{C}(\mathbf{F})$, but it proves that the construction $\mathbf{C}(\cdot)$ itself has no weakness. A main advantage of such a proof is that it applies to *every* cryptographic property of interest (which a random function has), not just to specific properties like collision-resistance.

The number B of queries up to which security is guaranteed is a crucial parameter of such a proof, especially in view of several surprises of the past years regarding weaknesses of iterated constructions. Joux [15] showed that the security of the Merkle-Damgård construction (with compression function with n-bit output) against finding multi-collisions is not much higher than the security against normal collision attacks, namely the birthday barrier $\mathcal{O}(2^{n/2})$, which is surprising because for a random function, finding an r-multi-collision requires $\Theta(2^{\frac{r-1}{r}n})$ queries. Joux's attack has been generalized to a wider class of constructions [14]. Other attacks in a similar spirit against iterated constructions are the second-preimage attack by Kelsey and Schneier [17], and herding attacks [16]. One possibility to overcome these issues is to rely on a compression function with input domain much larger than the size of the output of the construction (cf. for example the constructions in [18] and the double block-length construction of [10]), but this does not seem to be the best possible approach, both from a theoretical and from a practical viewpoint, as explained below.

A proof, like that of [11], for a construction $\mathbf{C}(\cdot)$ of a public random function, implies that $\mathbf{C}(\cdot)$ is secure against all possible attacks, up to the bound B on the number of queries stated in the proof. Since the bound in [11] is the birthday barrier, this implies nothing (beyond the birthday barrier) for attacks that require more queries, like the attacks of [15,17,10] mentioned above, and indeed the constructions of [11] also suffer from the same attacks.

[3] This is analogous to security proofs in the generic group model [27,20] which show that no attack exists that does not exploit the particular representation of group elements.

The bound B is also of importance since it determines the input and output sizes of $\mathbf{F}$. For example, because collision-resistance is a property that can hold only up to $2^{n/2}$ queries (due to the birthday paradox), n must be chosen twice as large as one might expect to be feasible in a naïve security analysis. Moreover, since the function must be compressing to be useful in a construction $\mathbf{C}(\cdot)$, the input size m must be larger than the output size n. However, if collision-resistance is not required, but instead for example second-preimage resistance, then the input size m of $\mathbf{F}$ can potentially be smaller or, turning the argument around, security for a given m can be much higher.

The input size m of $\mathbf{F}$ is relevant for two more reasons. First, if one considers the (perhaps not very realistic) possibility of finding a random function in Nature (say, by scanning the surface of the moon or by appropriately accessing the WWW), then m is a crucial parameter since the table size $n2^m$ is exponential in m. Second, for a given maximal computing time for $\mathbf{F}$, the difficulty of designing a concrete cryptographic function $\mathbf{F} : \{0,1\}^m \to \{0,1\}^n$ that is supposed to "look random" increases significantly if m is large. This can be seen as follows. Such a function $\mathbf{F}$ for large m could be modified in many different ways to reduce m to $m' < m$ (e.g. set $m - m'$ input bits to 0 or to any fixed value, or repeat an input of size m' until a block of length m is filled, etc.), and for each of these modifications it would still have to be secure.[4] Hence simply designing a new function with doubled m is not a very reasonable solution for the birthday barrier problem. Rather, one should find a construction that doubles (or multiplies) the input size but at the same time preserves the security almost optimally.

1.4 Contributions and Outline of This Paper

The main contribution of this paper is a construction paradigm for breaking the birthday barrier for domain extension of public random functions. More precisely, in Section 3 we prove that for every $\epsilon \in (0,1)$, m and ℓ, there exists an efficient construction $\mathbf{C}_{\epsilon,m,\ell}(\cdot)$ which extends a public random function $\{0,1\}^n \to \{0,1\}^n$ to a public random function $\{0,1\}^m \to \{0,1\}^\ell$, and which guarantees security for up to $\Theta(2^{n(1-\epsilon)})$ queries.

A central tool in our approach is a new combinatorial object, which we call an *input-restricting function family*. Section 4 discusses constructions of such families from highly-unbalanced bipartite expander graphs. While current expander constructions only allow our paradigm to be efficient in a complexity-theoretic sense (i.e. polynomial-time), an existence proof shows that very efficient constructions exist which would be of real practical interest if such graphs could be made explicit. We hope this paper provides additional motivation to investigate explicit constructions of unbalanced bipartite expanders for parameters ranges which have not received much attention so far.

Finally, our techniques allow to use a public random function $\{0,1\}^n \to \{0,1\}^n$ to construct a compression function with sufficiently large domain and

[4] This argument applies even though we know that a public random function is not securely realizable from a public random parameter.

range and to plug it into the construction of [11] to achieve the first iterated construction of a public random oracle $\{0,1\}^* \to \{0,1\}^n$ from a public random function $\{0,1\}^n \to \{0,1\}^n$ with security above the birthday barrier. We discuss this in Section 5.

2 Preliminaries

2.1 Notation and Probabilities

Throughout this paper, calligraphic letters (e.g. $\mathcal{U}$) denote sets. A k-tuple is denoted as $u^k = [u_1, \ldots, u_k]$, and the set of k-tuples of elements of $\mathcal{U}$ is denoted as $\mathcal{U}^k$. We use capital letters (e.g. U) to name random variables, whereas their concrete values are often denoted by the corresponding lower-case letters (e.g. u). Also, we write P_U for the probability distribution of U, and we use the shorthand $\mathsf{P}_U(u)$ for $\mathsf{P}(U = u)$. Given random variables U and V, as well as events $\mathcal{A}$ and $\mathcal{B}$, $\mathsf{P}_{U\mathcal{A}|V\mathcal{B}}$ denotes the corresponding conditional probability distribution, which is interpreted as a function $\mathcal{U} \times \mathcal{V} \to \mathbb{R}_{\geq 0}$, where the value $\mathsf{P}_{U\mathcal{A}|V\mathcal{B}}(u, v)$ is well-defined for all $u \in \mathcal{U}$ and $v \in \mathcal{V}$ such that $\mathsf{P}_{V\mathcal{B}}(v) > 0$ (and undefined otherwise). Two probability distributions P_U and $\mathsf{P}_{U'}$ on the same set $\mathcal{U}$ are equal, denoted $\mathsf{P}_U = \mathsf{P}_{U'}$, if $\mathsf{P}_U(u) = \mathsf{P}_{U'}(u)$ for all $u \in \mathcal{U}$. Also, for conditional probability distributions, equality holds if it holds for all inputs for which *both* are defined. We often need to deal with distinct random experiments where equally-named random variables and/or events appear. To avoid confusion, we add superscripts to probability distributions (e.g. $\mathsf{P}^{\mathcal{E}}_{U|V}(u, v)$) to make the random experiment explicit. Finally, we denote by $s \| s'$ the concatenation of two binary strings $s, s' \in \{0,1\}^*$.

2.2 Indistinguishability of Random Systems

In this section, we review basic definitions and facts from the framework of *random systems* of [19]. A random system is the abstraction of the input-output behavior of any discrete system.

Definition 1. *An* $(\mathcal{X}, \mathcal{Y})$-random system $\mathbf{F}$ *is a (generally infinite) sequence of conditional probability distributions*[5] $\mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}$ *for all* $i \geq 1$*. Two random systems* $\mathbf{F}$ *and* $\mathbf{G}$ *are* equivalent*, denoted* $\mathbf{F} \equiv \mathbf{G}$*, if* $\mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}} = \mathsf{p}^{\mathbf{G}}_{Y_i|X^iY^{i-1}}$ *for all* $i \geq 1$.

The system is described by the conditional probabilities $\mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}(y_i, x^i, y^{i-1})$ (for $i \geq 1$) of obtaining the output $y_i \in \mathcal{Y}$ on query $x_i \in \mathcal{X}$ given the previous $i-1$ queries $x^{i-1} = [x_1, \ldots, x_{i-1}] \in \mathcal{X}^{i-1}$ and their corresponding outputs $y^{i-1} = [y_1, \ldots, y_{i-1}] \in \mathcal{Y}^{i-1}$. An example of a random system that we consider in the following is a *random function* $\mathbf{R} : \{0,1\}^m \to \{0,1\}^n$, which

[5] We use a lower-case p to stress the fact that these conditional distributions by themselves do not define a random experiment.

returns for every distinct input value $x \in \{0,1\}^m$ an independent and uniformly-distributed n-bit value. Moreover, a *random oracle* $\mathbf{O} : \{0,1\}^* \to \{0,1\}^n$ is a random function taking inputs of arbitrary length.

A *distinguisher* $\mathbf{D}$ for an $(\mathcal{X},\mathcal{Y})$-random system is a $(\mathcal{Y},\mathcal{X})$-random system which is one query ahead, i.e. it is defined by the conditional probability distributions $\mathsf{p}^{\mathbf{D}}_{X_i|X^{i-1}Y^{i-1}}$ for all $i \geq 1$. In particular, $\mathsf{p}^{\mathbf{D}}_{X_1}$ is the probability distribution of the first value queried by $\mathbf{D}$. Finally, the distinguisher outputs a bit after a certain number (say k) of queries depending on the transcript (X^k, Y^k). For an $(\mathcal{X},\mathcal{Y})$-random system $\mathbf{F}$ and a distinguisher $\mathbf{D}$, we denote by $\mathbf{D} \circ \mathbf{F}$ the random experiment[6] where $\mathbf{D}$ interacts with $\mathbf{F}$. Furthermore, given an additional $(\mathcal{X},\mathcal{Y})$-random system $\mathbf{G}$, the *distinguishing advantage* of $\mathbf{D}$ in distinguishing systems $\mathbf{F}$ and $\mathbf{G}$ is defined as $\Delta^{\mathbf{D}}(\mathbf{F},\mathbf{G}) := \left|\mathsf{P}^{\mathbf{D}\circ\mathbf{F}}(1) - \mathsf{P}^{\mathbf{D}\circ\mathbf{G}}(1)\right|$, where $\mathsf{P}^{\mathbf{D}\circ\mathbf{F}}(1)$ and $\mathsf{P}^{\mathbf{D}\circ\mathbf{G}}(1)$ denote the probabilities that $\mathbf{D}$ outputs 1 after its k queries when interacting with $\mathbf{F}$ and $\mathbf{G}$, respectively.

We are interested in considering an internal *monotone condition* defined on a random system $\mathbf{F}$. Such a condition is initially true, and once it fails, it cannot become true any more. In particular, a *system* $\mathbf{F}^{\mathcal{A}}$ *with a monotone condition* $\mathcal{A}$ is an $(\mathcal{X},\mathcal{Y} \times \{0,1\})$-random system, where the additional output bit indicates whether the condition $\mathcal{A}$ holds after the i'th query has been answered. In general, we characterize such a condition by a sequence of events $\mathcal{A} = A_0, A_1, \ldots$, where A_0 always holds, and A_i holds if the condition holds after query i. The condition *fails* at query i if $A_{i-1} \wedge \overline{A_i}$ occurs. For a system with a monotone condition $\mathbf{F}^{\mathcal{A}}$, we write $\mathbf{F}$ for the system where the additional output bit is ignored. Generally, we are interested in considering the behavior of systems only as long as a certain monotone condition holds: Given two systems $\mathbf{F}^{\mathcal{A}}$ and $\mathbf{G}^{\mathcal{B}}$ with monotone conditions $\mathcal{A}$ and $\mathcal{B}$, respectively, they are *equivalent*, denoted $\mathbf{F}^{\mathcal{A}} \equiv \mathbf{G}^{\mathcal{B}}$, if $\mathsf{p}^{\mathbf{F}}_{A_iY_i|X^iY^{i-1}A_{i-1}} = \mathsf{p}^{\mathbf{G}}_{B_iY_i|X^iY^{i-1}B_{i-1}}$ holds for all $i \geq 1$.

The probability that a distinguisher $\mathbf{D}$ issuing k queries makes a monotone condition $\mathcal{A}$ fail in the random experiment $\mathbf{D} \circ \mathbf{F}$ is defined as $\nu^{\mathbf{D}}(\mathbf{F}^{\mathcal{A}}) := \mathsf{P}^{\mathbf{D}\circ\mathbf{F}}_{\overline{A_k}}$. The following lemma from [19] relates this probability with the distinguishing advantage.

Lemma 1. *If* $\mathbf{F}^{\mathcal{A}} \equiv \mathbf{G}^{\mathcal{B}}$ *holds, then* $\Delta^{\mathbf{D}}(\mathbf{F},\mathbf{G}) \leq \nu^{\mathbf{D}}(\mathbf{F}^{\mathcal{A}}) = \nu^{\mathbf{D}}(\mathbf{G}^{\mathcal{B}})$ *for all distinguishers* $\mathbf{D}$*.*

One can use a random system $\mathbf{F}$ as a component of a larger system: In particular, we are interested in *constructions* $\mathbf{C}(\cdot)$ such that the resulting random system $\mathbf{C}(\mathbf{F})$ invokes $\mathbf{F}$ as a subsystem. (Note that $\mathbf{C}(\cdot)$ itself is not a random system, while $\mathbf{C}(\mathbf{F})$ is a random system.)

Finally, we remark that in general when we mention that a construction (or a distinguisher) is *efficient* we mean that there exists a probabilistic interactive Turing machine implementing the same input-output behavior and with polynomial running time (in the understood security parameter).

[6] In particular, in this random experiment, the joint distribution $\mathsf{P}^{\mathbf{D}\circ\mathbf{F}}_{X^kY^k}$ is well-defined as $\prod_{i=1}^{k} \mathsf{p}^{\mathbf{D}}_{X_i|X^{i-1}Y^{i-1}} \cdot \mathsf{p}^{\mathbf{F}}_{Y_i|X^iY^{i-1}}$.

2.3 Indifferentiability, Reductions, and Public Random Primitives

The notion of *indifferentiability* [21] naturally extends the concept of indistinguishability to systems with a *public* and a *private* interface[7] adopting a simulation-based approach. The public interface can be used by all parties, including the adversary, whereas the legitimate parties have exclusive access to the private interface. Generally, we denote such a system as an ordered pair $\mathbf{F} = [\mathbf{F}_{\text{pub}}, \mathbf{F}_{\text{priv}}]$. Furthermore, given constructions $\mathbf{S}(\cdot)$ and $\mathbf{C}(\cdot)$ leaving, respectively, private and public queries unmodified, we simply write $\mathbf{S}(\mathbf{F}) = [\mathbf{S}(\mathbf{F}_{\text{pub}}), \mathbf{F}_{\text{priv}}]$ and $\mathbf{C}(\mathbf{F}) = [\mathbf{F}_{\text{pub}}, \mathbf{C}(\mathbf{F}_{\text{priv}})]$.

Public random primitives are a special case of such systems. A *public random function* (*puRF*) $\mathbf{R} : \{0,1\}^m \to \{0,1\}^n$ is a system with a public and a private interface which behaves as the *same* random function at *both* interfaces.[8] In particular, both interfaces answer consistently. Furthermore, a *public random oracle (puRO)* $\mathbf{O} : \{0,1\}^* \to \{0,1\}^n$ is a public random function which takes inputs of arbitrary bit-length.

The following definition refines the notion of (information-theoretic) indifferentiability from [21] to deal with concrete parameters.

Definition 2. *Let* $\alpha : \mathbb{N} \to \mathbb{R}_{\geq 0}$ *and* $\sigma : \mathbb{N} \to \mathbb{N}$ *be functions. A system* $\mathbf{F}$ *is* (α, σ)-indifferentiable *from* $\mathbf{G}$, *denoted* $\mathbf{F} \overset{\alpha,\sigma}{\sqsubset} \mathbf{G}$, *if there exists a simulator* $\mathbf{S}$ *such that* $\Delta^{\mathbf{D}}([\mathbf{F}_{pub}, \mathbf{F}_{priv}], [\mathbf{S}(\mathbf{G}_{pub}), \mathbf{G}_{priv}]) \leq \alpha(k)$ *for all distinguishers* $\mathbf{D}$ *making at most* k *queries, and* $\mathbf{S}$ *makes at most* $\sigma(k)$ *queries to* $\mathbf{G}_{pub}$ *when interacting with* $\mathbf{D}$.

The purpose of the simulator is to mimic $\mathbf{F}_{\text{pub}}$ by querying $\mathbf{G}_{\text{pub}}$, but without seeing the queries made to $\mathbf{G}_{\text{priv}}$. Indifferentiability directly implies a notion of reducibility.

Definition 3. *A system* $\mathbf{G}$ *is* (α, σ)-reducible *to a system* $\mathbf{F}$ *if there exists an* efficient*,* deterministic*, and* stateless *construction* $\mathbf{C}(\cdot)$ *such that* $[\mathbf{F}_{pub}, \mathbf{C}(\mathbf{F}_{priv})] \overset{\alpha,\sigma}{\sqsubset} \mathbf{G}$. *The construction* $\mathbf{C}(\cdot)$ *is called an* (α, σ)-reduction.

Note that if a random primitive is (α, σ)-reducible to a further random primitive with an N-bit table, then $\alpha(k) \geq \frac{1}{2}$ for all $k > N$, and hence security can only be achieved with respect to distinguishers issuing at most N queries. (We refer the reader to the full version [23] for a proof.) The following lemma states that reducibility is transitive. We omit its simple proof.

Lemma 2. *Let* $\mathbf{E}, \mathbf{F}$, *and* $\mathbf{G}$ *be systems. If* $\mathbf{C}(\cdot)$ *is a* (α, σ)*-reduction of* $\mathbf{F}$ *to* $\mathbf{E}$, *and* $\mathbf{C}'(\cdot)$ *is an* (α', σ') *reduction of* $\mathbf{G}$ *to* $\mathbf{F}$ *that makes at most* $k_{\mathbf{C}'}(k)$ *queries to* $\mathbf{F}_{priv}$ *when queried* k *times, then* $\mathbf{C}'(\mathbf{C}(\cdot))$ *is an* $(\overline{\alpha}, \overline{\sigma})$*-reduction of* $\mathbf{G}$ *to* $\mathbf{E}$, *where* $\overline{\alpha}(k) = \alpha(k + k_{\mathbf{C}'}(k)) + \alpha'(k + \sigma(k))$ *and* $\overline{\sigma}(k) = \sigma'(\sigma(k))$.

[7] Formally, this can be seen as a random system with a single interface and two types of queries.

[8] For this reason, we generally write both $\mathbf{R}_{\text{pub}}$ and $\mathbf{R}_{\text{priv}}$ as $\mathbf{R}$.

The computational variant of indifferentiability is obtained by requiring $\mathbf{S}$ to be efficient and the advantage $\Delta^{\mathbf{D}}([\mathbf{F}_{\text{pub}}, \mathbf{F}_{\text{priv}}], [\mathbf{S}(\mathbf{G}_{\text{pub}}), \mathbf{G}_{\text{priv}}])$ to be negligible for all efficient distinguishers $\mathbf{D}$. A *computational reduction* is defined accordingly. In the information theoretic case, it is sometimes desirable to prove that the simulator is efficient when queried by an efficient distinguisher, as this then implies the corresponding complexity-theoretic statement. We refer the reader to [21,11] for the implications of computational indifferentiability.

In contrast, as long as we are only interested in excluding generic attacks against security properties of a random function, the running time of the simulator is irrelevant. If $\mathbf{C}(\cdot)$ is an (α, σ)-reduction of a puRO $\mathbf{O} : \{0,1\}^* \to \{0,1\}^n$ (or of a puRF $\mathbf{R}' : \{0,1\}^m \to \{0,1\}^\ell$) to a puRF $\mathbf{R} : \{0,1\}^n \to \{0,1\}^n$, then $\mathbf{C}(\mathbf{R})$ inherits *all* the security properties of the truly-random oracle $\mathbf{O}$ (or of $\mathbf{R}'$), as long as the number of queries keeps $\alpha(k)$ small: Any adversary A making k queries (to both $\mathbf{R}$ and $\mathbf{C}(\mathbf{R})$) and breaking some property of $\mathbf{C}(\mathbf{R})$ with probability $\pi(k)$ can be transformed (combining it with the simulator) into an adversary A' making at most $k + \sigma(k)$ queries to $\mathbf{O}$ and breaking the same property for $\mathbf{O}$ with probability at least $\pi(k) - \alpha(k)$, and if no such A' can exist, then also no adversary A exists. The actual running time of A' is irrelevant, as the security of a random function (or oracle) with respect to a certain property is determined by the number of queries of the adversary, and not by its running time. For example, if $\sigma(k) = \Theta(k)$, then, given a random element $s \in \{0,1\}^m$, no adversary can find a second preimage $s' \in \{0,1\}^m$ with $s' \neq s$ and $\mathbf{C}(\mathbf{R})(s) = \mathbf{C}(\mathbf{R})(s')$ with probability higher than $\Theta(k \cdot 2^{-n}) + \alpha(k)$.

3 Beyond-Birthday Domain Extension for Public Random Functions

3.1 The Construction

We first discuss at an abstract level the main construction of this paper (represented in Figure 1), which implements a function mapping m-bit strings to ℓ-bit strings from $r + t$ independent puRF's $\mathbf{F}_1, \ldots, \mathbf{F}_r : \{0,1\}^n \to \{0,1\}^{t\rho n}$ and $\mathbf{G}_1, \ldots, \mathbf{G}_t : \{0,1\}^n \to \{0,1\}^\ell$ (for given parameters r, t, and ρ). Let $E_1, \ldots, E_r : \{0,1\}^m \to \{0,1\}^n$ be efficiently-computable functions (to be instantiated below). On input $s \in \{0,1\}^m$, the construction operates in three stages:

1. The values $\mathbf{F}_p(E_p(s)) = \mathbf{F}_p^{(1)}(E_p(s)) \| \cdots \| \mathbf{F}_p^{(t)}(E_p(s)) \in \{0,1\}^{t\rho n}$ are computed for all $p = 1, \ldots, r$, where $\mathbf{F}_p^{(q)}(E_p(s)) \in \{0,1\}^{\rho n}$ for all $q = 1, \ldots, t$;
2. The value $w(s) = w^{(1)}(s) \| \cdots \| w^{(t)}(s)$ is computed, where $w^{(q)}(s)$ equals (for all $q = 1, \ldots, t$) the first n bits of the product $\bigodot_{p=1}^{r} \mathbf{F}_p^{(q)}(E_p(s))$, and $\odot$ denotes multiplication in $GF(2^{\rho n})$ with ρn-bit strings interpreted as elements of the finite field $GF(2^{\rho n})$;
3. Finally, the value $\bigoplus_{q=1}^{t} \mathbf{G}_q(w^{(q)}(s))$ is output.

Our approach relies on the observation that if for each new query to the construction with input $s \in \{0,1\}^m$ there exists an index $q \in \{1, \ldots, t\}$ for which $\mathbf{G}_q$

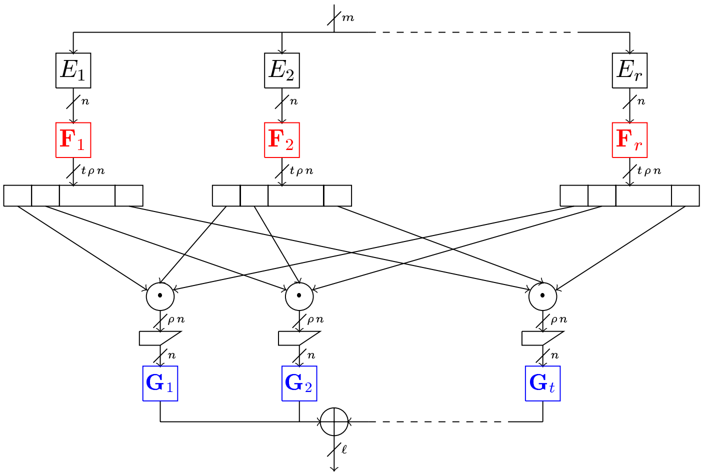

Fig. 1. Main construction, where $\mathbf{F}_1, \ldots, \mathbf{F}_r$ and $\mathbf{G}_1, \ldots, \mathbf{G}_t$ are independent puRF's and $E_1, \ldots, E_r : \{0,1\}^m \to \{0,1\}^n$ are efficiently-computable functions

has not been queried yet at the value $w^{(q)}(s)$, either directly at its public interface or by the construction at the private interface, the resulting output value is uniformly distributed and independent from all previously-returned values. This resembles the approach taken to extend the domain of (secret) random functions [1,4,19]. However, we stress that the role of the first two stages (including the functions $E_1, \ldots, E_r$) is crucial here: Not only they have to guarantee that such an index q always exists, but they must also permit simulation of the puRF's $\mathbf{F}_1, \ldots, \mathbf{F}_r$ and $\mathbf{G}_1, \ldots, \mathbf{G}_t$ given only access to the public interface of an (ideal) puRF $\mathbf{R} : \{0,1\}^m \to \{0,1\}^\ell$, without seeing the queries made to the private interface of $\mathbf{R}$. Also, the probability that the simulation fails must be small enough to allow security beyond the birthday barrier.

3.2 Input-Restricting Functions

For every $s \in \{0,1\}^m$ one can always learn the value $w(s)$ by querying the public interfaces of $\mathbf{F}_1, \ldots, \mathbf{F}_r$ with appropriate inputs $E_1(s), \ldots, E_p(s)$, respectively. For every such s, the sum $\bigoplus_{q=1}^{t} \mathbf{G}_q(w^{(q)}(s))$ equals the output of the construction on input s. The simulator must ensure that its answers for queries to the functions $\mathbf{G}_1, \ldots, \mathbf{G}_t$ are consistent with these constraints. However, if $E_1, \ldots, E_r$ allow a relatively small number of queries to the functions $\mathbf{F}_1, \ldots, \mathbf{F}_t$ to reveal a too large number of values $w(s)$, then the simulator possibly fails to satisfy all constraints. For example, the *Benes* construction [1] adopts an approach similar to the one of our construction, but suffers from this problem and its security in the setting of puRF's is inherently bounded by the birthday barrier. (We provide a concrete attack in the full version [23].) To overcome this problem, we introduce the following combinatorial notion.

Definition 4. *Let* $\epsilon \in (0,1)$ *, and let* $m > n$*. A family* $\mathcal{E}$ *of functions* $E_1, \ldots, E_r : \{0,1\}^m \to \{0,1\}^n$ *is called* (m, δ, ϵ)-input restricting *if it satisfies the following two properties:*

Injective. *For all* $s \neq s' \in \{0,1\}^m$*, there exists* $p \in \{1, \ldots, r\}$ *such that* $E_p(s) \neq E_p(s')$*.*

Input-Restricting. *For all subsets* $\mathcal{U}_1, \ldots, \mathcal{U}_r \subseteq \{0,1\}^n$ *such that* $|\mathcal{U}_1| + \cdots + |\mathcal{U}_r| \leq 2^{n(1-\epsilon)}$*, we have*

$$\left|\{s \in \{0,1\}^m \mid E_p(s) \in \mathcal{U}_p \textit{ for all } p = 1, \ldots, r\}\right| \leq \delta \cdot (|\mathcal{U}_1| + \cdots + |\mathcal{U}_r|) .$$

It is easy to see that $\delta \geq 1/r$ must hold. Furthermore, we need $r \cdot n \geq m$ for the family to be injective. When talking about efficiency, we can naturally extend the notion to asymptotic families $\mathcal{E} = \{\mathcal{E}_n\}_{n \in \mathbb{N}}$ of function families by letting m, δ, ϵ, and r be functions of n, and $\mathcal{E}_n = \{E_1^n, \ldots, E_{r(n)}^n\}$, with $E_p^n : \{0,1\}^{m(n)} \to \{0,1\}^n$. In particular, note that we allow the size of the family to grow with the security parameter. The family $\mathcal{E}_n$ is called *explicit* if $r = r(n)$ is polynomial in n and if there exists a (uniform) polynomial-time (in n) algorithm E that outputs $E_p^n(s) \in \{0,1\}^n$ on input $n \in \mathbb{N}$, $s \in \{0,1\}^{m(n)}$, and $p \in \{1, \ldots, r(n)\}$. The family is additionally called *invertible* if there exists an algorithm which on input the sets $\mathcal{U}_1, \ldots, \mathcal{U}_r \subseteq \{0,1\}^n$ and n returns the set of all $s \in \{0,1\}^m$ for which $E_p(s) \in \mathcal{U}_p$ for all $p = 1, \ldots, r$ in time polynomial in $|\mathcal{U}_1| + \cdots + |\mathcal{U}_r|$ and in n. We will not, however, stress the asymptotic point of view in the following, as long as it is clear from the context that the statements can be also formalized in this sense.

We postpone the discussion of the existence of explicit function families to Section 4, where we construct (for all constants ϵ) explicit families of (m, δ, ϵ)-input-restricting functions for all polynomials m and sufficiently-small δ using highly unbalanced expander graphs with polynomial-degree.

3.3 Main Result

Let $\epsilon \in (0,1)$. The concrete construction $\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot)$ is obtained from the description in Section 3.1 by instantiating the functions $E_1, \ldots, E_r$ with an explicit family $\mathcal{E} = \{E_1, \ldots, E_r\}$ of (m, δ, ϵ)-input restricting functions with n-bit output. Also, we let $\rho := \left\lceil \frac{m}{n} + 2 - \epsilon \right\rceil$ and $t := \lceil 2/\epsilon - 1 \rceil$. Note that underlying $r + t$ puRF's can be seen as a single puRF $\mathbf{R}' : \{0,1\}^{n+\phi(n)} \to \{0,1\}^n$, where $\phi(n) = \lceil \log(r \cdot t\rho + t\ell/n) \rceil$. If m, ℓ, and $1/\epsilon$ are polynomial in n, then in particular $\phi(n) = \mathcal{O}(\log n)$. Also, it is easy to see that $\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot)$ is efficient, as long as the function family $\mathcal{E}$ is explicit. The following is the main theorem of this paper and it is proved in the next section.

Theorem 1. *The construction* $\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot)$ *is an* (α, σ)*-reduction of the puRF* $\mathbf{R} : \{0,1\}^m \to \{0,1\}^\ell$ *to the puRF's* $\mathbf{F}_1, \ldots, \mathbf{F}_r : \{0,1\}^n \to \{0,1\}^{t \cdot \rho n}$ *and* $\mathbf{G}_1, \ldots, \mathbf{G}_t : \{0,1\}^n \to \{0,1\}^\ell$*, where for all* $k \leq 2^{n(1-\epsilon)} - r$*,*

$$\alpha(k) \leq 2r^t(\delta+1)^{t+1} \cdot k^{t+2} \cdot 2^{-nt} + \frac{1}{2}t(\delta+1) \cdot k \cdot (k + 2r + 1) \cdot 2^{m-\rho n}$$

and $\sigma(k) \leq \delta(n) \cdot k$. If the family $\mathcal{E}$ is invertible, the simulator runs in time polynomial in k and n, and in particular $\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot)$ is also a computational reduction.

We remark the following two important consequences of Theorem 1. First, if ϵ is constant and r, δ polynomial in n, the above advantage $\alpha(k)$ is negligible for all parameters k up to $k = 2^{n(1-\epsilon)} - r$. In particular, choosing $\epsilon < \frac{1}{2}$ leads to security beyond the birthday barrier,[9] and we are going to provide input-restricting families of functions with appropriate parameters in Section 4. Second, the result can be used to extend the domain of a puRF $\mathbf{R}' : \{0,1\}^n \to \{0,1\}^n$ with security up to $2^{n(1-\mu)}$ queries: One chooses any $\epsilon < \mu$ and n' maximal such that $n' + \phi(n') \leq n$, and interprets the function $\mathbf{R}'$ as a puRF $\{0,1\}^{n'+\phi(n')} \to \{0,1\}^{n'}$ by dropping approximately $\phi(n')$ bits of the output. The above advantage is still negligible for all $k \leq 2^{n'(1-\epsilon)} - r$, and hence for all $k \leq 2^{n(1-\mu)}$ for n large enough, since $n - n' = o(n)$.

3.4 Proof of Theorem 1

We prove that there exists a simulator $\mathbf{S}$ such that $\Delta^{\mathbf{D}}(\mathbf{H}_1, \mathbf{H}_2)$ is bounded by the above expression for all distinguishers $\mathbf{D}$ making at most $k \leq 2^{n(1-\epsilon)} - r$ queries, where for notational convenience $\mathbf{H}_1$ and $\mathbf{H}_2$ are defined as

$$\begin{aligned}\mathbf{H}_1 &:= [\mathbf{F}_1, \ldots, \mathbf{F}_r, \mathbf{G}_1, \ldots, \mathbf{G}_t, \mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\mathbf{F}_1, \ldots, \mathbf{F}_r, \mathbf{G}_1, \ldots, \mathbf{G}_t)]\\ \mathbf{H}_2 &:= [\mathbf{S}(\mathbf{R}), \mathbf{R}].\end{aligned}$$

There are three types of queries to the systems $\mathbf{H}_1$ and $\mathbf{H}_2$: The first two types are $\mathbf{F}$-*queries*, denoted (F, p, u) for $p \in \{1, \ldots, r\}$ and $u \in \{0,1\}^n$, and $\mathbf{G}$-*queries*, denoted (G, q, v), for $v \in \{0,1\}^n$ and $q \in \{1, \ldots, t\}$. In $\mathbf{H}_1$, a query (F, p, u) returns the value $\mathbf{F}_p(u)$ and a query (G, q, v) returns the value $\mathbf{G}_q(v)$, while in $\mathbf{H}_2$ both query-types are answered by the simulator $\mathbf{S}$. The third type of queries, called $\mathbf{R}$-*queries*, are denoted (R, s) for $s \in \{0,1\}^m$ and are answered by the construction $\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot)$ in $\mathbf{H}_1$, and by the private interface of the random function $\mathbf{R}$ in $\mathbf{H}_2$. Given the first i queries $x^i = [x_1, \ldots, x_i]$, where $x_j \in \{(\mathrm{F}, p, u), (\mathrm{G}, q, v), (\mathrm{R}, s)\}$ for all $j = 1, \ldots, i$, we define for all indices p and q the sets $\mathcal{F}_{p,i}$ and $\mathcal{G}_{q,i}$ that contain, respectively, all values $u \in \{0,1\}^n$ for which a query (F, p, u) and all $v \in \{0,1\}^n$ for which a query (G, q, v) appears in x^i. Also, we let $\mathcal{R}_i$ be the set of values $s \in \{0,1\}^m$ for which a query (R, s) appears in x^i, and we let $\mathcal{S}_i$ consist of all the values $s \in \{0,1\}^m$ such that $E_p(s) \in \mathcal{F}_{p,i}$ for all $p = 1, \ldots, r$. Furthermore, let $\Delta\mathcal{S}_i := \mathcal{S}_i \backslash \mathcal{S}_{i-1}$. Notice that the set $\mathcal{S}_i$ contains all inputs for which the values returned by the first i queries allow to compute the value $w(s)$. Clearly, $|\mathcal{S}_i| = \sum_{j=1}^{i} |\Delta\mathcal{S}_j| \leq \delta \cdot i$ for all $i \leq 2^{n(1-\epsilon)}$, since the family $\mathcal{E}$ is input-restricting. For $s \in \mathcal{S}_i$, we define $w(s) = w^{(1)}(s) \| \cdots \| w^{(t)}(s)$ as in the description of $\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot)$ according to the answers of the first queries, and for a set $\mathcal{S} \subseteq \mathcal{S}_i$ we use the shorthand $w^{(q)}(\mathcal{S}) := \{w^{(q)}(s) \mid s \in \mathcal{S}\}$.

[9] Note that ϵ could even be some function going (slowly) towards zero, even though this may require setting t differently.

The simulator $\mathbf{S}$ defines the function tables of $\mathbf{F}_1, \ldots, \mathbf{F}_r$ and of $\mathbf{G}_1, \ldots, \mathbf{G}_t$ *dynamically.* That is, all values $\mathbf{F}_p(u)$ and $\mathbf{G}_q(v)$ are initially *undefined* for all $u, v \in \{0,1\}^n$ and indices p and q. Upon processing a new $\mathbf{F}$-query $x_i = (\mathrm{F}, p, u)$, the simulator sets the value $\mathbf{F}_p(u)$ to a fresh random value and computes the set $\Delta\mathcal{S}_i$: The simulator knows this set, as it processes all $\mathbf{F}$-queries. For each $s \in \Delta\mathcal{S}_i$, the equality $\bigoplus_{q=1}^{t} \mathbf{G}_q(w^{(q)}(s)) = \mathbf{R}(s)$ must be satisfied, and hence $\mathbf{S}$ tries to satisfy these constraints by appropriately setting the values of the functions $\mathbf{G}_1, \ldots, \mathbf{G}_t$. More precisely, it looks for an ordering of $\Delta\mathcal{S}_i = \{s_1, \ldots, s_{|\Delta\mathcal{S}_i|}\}$ with the property that for all $j = 1, \ldots, |\Delta\mathcal{S}_i|$ there exists $q_j \in \{1, \ldots, t\}$ such that $w^{(q_j)}(s_j) \notin \{w^{(q_j)}(s_1), \ldots, w^{(q_j)}(s_{j-1})\} \cup \mathcal{G}_{q,i-1}$, and sets $\mathbf{G}_{q_j}(w^{(q_j)}(s_j)) := \mathbf{R}(s_j) \oplus \bigoplus_{q \neq q_j} \mathbf{G}_q(w^{(q)}(s_j))$ for $j = 1, \ldots, |\Delta\mathcal{S}_i|$, where each undefined value in the sums is set to an independent random value. A query to the public interface of $\mathbf{R}$ is issued in order to learn $\mathbf{R}(s_j)$. If no such ordering exists, then the simulator aborts.[10] Finally, the value $\mathbf{F}_p(u)$ is returned. For a query $x_i = (\mathrm{G}, q, v)$, the simulator returns $\mathbf{G}_q(v)$, defining it to a random value if undefined. In the full version of this paper [23], we provide a detailed pseudo-code description of the simulator $\mathbf{S}$. The number of $\mathbf{R}$-queries made by the simulator after $i \leq 2^{n(1-\epsilon)}$ queries is $|\mathcal{S}_i| \leq \delta \cdot i$. Also, as long as the family $\mathcal{E}$ is invertible and an appropriate ordering can be efficiently found, its running time is efficient in k and n. In fact, we show that with very high probability *any* ordering can be used.

Without loss of generality, it is convenient to advance the generation of the random functions $\mathbf{F}_1, \ldots, \mathbf{F}_r$ to the initialization phase, that is, their *entire* function tables are generated once uniformly at random in both $\mathbf{H}_1$ and $\mathbf{H}_2$. Subsequently, all queries (F, p, u) are answered according to the initial choice. In particular, this means that in $\mathbf{H}_2$ the simulator $\mathbf{S}$ uses the value $\mathbf{F}_p(u)$ already defined instead of generating a new fresh random value. It is clear that the behavior of both systems is unchanged. This also allows us to define the value $w(s) = w^{(1)}(s) \| \cdots \| w^{(t)}(s)$ for *all* $s \in \{0,1\}^m$ and each such value induces a constraint, namely the answer of an $\mathbf{R}$-query (R, s) must equal $\bigoplus_{q=1}^{t} \mathbf{G}_q(w^{(q)}(s))$. Such a constraint remains hidden until $s \in \Delta\mathcal{S}_i$ from some i, and in this case the simulator attempts to fill the function tables of $\mathbf{G}_1, \ldots, \mathbf{G}_t$ consistently. To avoid possible problems, we have to account for two things captured by the two following monotone conditions which we define on both $\mathbf{H}_1$ and $\mathbf{H}_2$:

(a) The monotone condition $\mathcal{A} = A_0, A_1, \ldots$ fails at query i if there exists an $s \in \Delta\mathcal{S}_i$ such that $w^{(q)}(s) \in w^{(q)}(\mathcal{S}_i \setminus \{s\}) \cup \mathcal{G}_{q,i-1}$ for all $q = 1, \ldots, t$.

(b) The monotone condition $\mathcal{B} = B_0, B_1, \ldots$ fails at query i if there exists $s \in \mathcal{R}_i \setminus \mathcal{S}_i$ such that $w^{(q)}(s) \in w^{(q)}(\mathcal{S}_i \cup \mathcal{R}_i \setminus \{s\}) \cup \mathcal{G}_{q,i}$ for all $q = 1, \ldots, t$.

As long as $\mathcal{A}$ does not fail, the simulator never aborts. This in particular implies that $\mathbf{R}$-queries (R, s) for $s \in \mathcal{S}_i$ in $\mathbf{H}_2$ are consistent with $\mathbf{G}$-queries answered by the simulator. However, all $\mathbf{R}$-queries (R, s) for $s \notin \mathcal{S}_i$ are answered independently and uniformly at random in $\mathbf{H}_2$, and $\mathcal{B}$ ensures that this happens in $\mathbf{H}_1$

[10] Note that there is no need to formalize the exact meaning of abortion, since whenever the simulator fails to find such an ordering, then the distinguisher is assumed to win.

as well. In the full version [23], we prove the following lemma, which formalizes this argument and states that as long as neither $\mathcal{A}$ nor $\mathcal{B}$ fail, then $\mathbf{H}_1$ and $\mathbf{H}_2$ behave identically.

Lemma 3. $\mathbf{H}_1^{\mathcal{A}\wedge\mathcal{B}} \equiv \mathbf{H}_2^{\mathcal{A}\wedge\mathcal{B}}$.

To provide some intuition as to why the probability that a distinguisher $\mathbf{D}$ makes $\mathcal{A} \wedge \mathcal{B}$ fail is small, let us assume first that for any two distinct $s, s' \in \{0,1\}^m$ (such that at least one of them is not in $\mathcal{S}_i$) and for all $q = 1, \ldots, t$, the probability (conditioned on the answers to the previous queries) that $w^{(q)}(s) = w^{(q)}(s')$ is bounded by some small value φ (say $\varphi \approx 2^{-n}$). In order to upper bound the probability of $\mathcal{A}$ failing after query i, combining the union bound with the above assumption we see that $\mathsf{P}(w^{(q)}(s) \in w^{(q)}(\mathcal{S}_i \setminus \{s\}) \cup \mathcal{G}_{q,i-1}) \leq |w^{(q)}(\mathcal{S}_i \setminus \{s\}) \cup \mathcal{G}_{q,i-1}| \cdot \varphi \leq (\delta+1) \cdot i \cdot \varphi$ for all $s \in \Delta\mathcal{S}_i$, since $\mathcal{E}$ is input-restricting. Furthermore, for all distinct $q, q' \in \{1, \ldots, t\}$ and $s, s' \in \{0,1\}^n$ (possibly $s = s'$), the structure of the first two stages of $\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot)$ ensures that the values $w^{(q)}(s)$ and $w^{(q')}(s')$ are statistically independent, and hence

$$\mathsf{P}(\forall q : w^{(q)}(s) \in w^{(q)}(\mathcal{S}_i \setminus \{s\}) \cup \mathcal{G}_{q,i-1}) \leq (\delta+1)^t \cdot i^t \cdot \varphi^t.$$

Therefore, the probability $\mathsf{p}^{\mathbf{H}_1}_{A_i|X^iY^{i-1}A_{i-1}}(x^i, y^{i-1}) = \mathsf{p}^{\mathbf{H}_2}_{A_i|X^iY^{i-1}A_{i-1}}(x^i, y^{i-1})$ that there exists an $s \in \Delta\mathcal{S}_i$ making $\mathcal{A}$ fail after query i is bounded by $|\Delta\mathcal{S}_i| \cdot (\delta+1)^t \cdot i^t \cdot \varphi^t$, where $|\Delta\mathcal{S}_i|$ is small for all $i \leq 2^{n(1-\epsilon)}$.

Nevertheless, turning this intuition into a formal proof (and extending it to the monotone condition $\mathcal{B}$) requires additional care. The probability that $w^{(q)}(s)$ equals $w^{(q)}(s')$ happens to be small only with overwhelming probability (taken over the answers to the previous queries): This fact follows from the use of multiplication in $GF(2^{\rho n})$ and the choice of a sufficiently large parameter ρ.

In particular, a complete proof of the following lemma appears in the full version of this paper [23].

Lemma 4. *For all distinguishers* $\mathbf{D}$ *making at most* $k \leq 2^{n(1-\epsilon)} - r$ *queries we have* $\nu^{\mathbf{D}}(\mathbf{H}_1^{\mathcal{A}\wedge\mathcal{B}}) = \nu^{\mathbf{D}}(\mathbf{H}_2^{\mathcal{A}\wedge\mathcal{B}}) \leq 2r^t(\delta+1)^{t+1} \cdot k^{t+2} \cdot 2^{-nt} + \frac{1}{2}t(\delta+1) \cdot k \cdot (k + 2r + 1) \cdot 2^{m-\rho n}$.

By combining Lemmas 3 and 4, Theorem 1 follows making use of Lemma 1.

4 Existence of Input-Restricting Function Families

In this section, we prove the existence of input-restricting function families according to Definition 4, and we study their relationship to *highly unbalanced* bipartite expander graphs. First, we recall the following definition.

Definition 5. *A bipartite graph* $G = (V_1, V_2, E)$ *is* (K, γ)-expanding *if* $|\Gamma(X)| \geq \gamma \cdot |X|$ *for all subsets* $X \subset V_1$ *such that* $|X| \leq K$, *where* $\Gamma(X) \subseteq V_2$ *is the set of neighbors of* X. *Furthermore, such a graph has* left-degree D *if the degree of all* $v \in V_1$ *is bounded by* D.

A family of graphs $G = (V_1, V_2, E)$ with $V_1 := \{0,1\}^{m(n)}$, $V_2 := \{0,1\}^n$ (parameterized by the security parameter n) with left-degree $D = D(n)$ is called *explicit* if there exists a (uniform) algorithm which, on input 1^n, $v \in \{0,1\}^{m(n)}$ and $i \in \{1, \ldots, D(n)\}$ outputs the i'th neighbor of v in time polynomial in n. (The ordering of the neighbors is arbitrary.)

Given a bipartite graph $G = (V_1, V_2, E)$ with $V_1 = \{0,1\}^m$, $V_2 = \{0,1\}^n$, and left-degree D, we construct the family of functions $\mathcal{E} = \{E_1, \ldots, E_r\}$, where $r = D + \lceil m/n \rceil$, and the functions $E_1, \ldots, E_D : \{0,1\}^m \to \{0,1\}^n$ are such that $E_p(s)$ is the p'th neighbor of s in G for all $p = 1, \ldots, D$. Furthermore, the functions $E_{D+1}, \ldots, E_{D+\lceil m/n \rceil}$ are defined as $E_{D+p}(s) := s^{(p)}$ for $p = 1, \ldots, \lceil m/n \rceil$, where extra zeros are appended to s to make its length a multiple of n. Clearly, this family is injective. Furthermore, it turns out that good expanding properties for G imply that the family $\mathcal{E}$ is input-restricting. We refer the reader to the full version [23] for a proof of the following lemma.

Lemma 5. *Let $m \geq n$. Assume that there exists an explicit family of bipartite (K, γ)-expander graphs $G = (V_1, V_2, E)$ with polynomially-bounded left-degree D where $V_1 = \{0,1\}^m$ and $V_2 = \{0,1\}^n$. Then, for all $\epsilon > 0$ such that $\epsilon > 1 - \frac{\log(K\gamma)}{n}$ for n large enough, there exists an explicit (m, δ, ϵ)-input-restricting family of functions with $\delta = \gamma^{-1}$ and cardinality $r := D + \lceil m/n \rceil$. Furthermore, if $\lceil m/n \rceil$ is constant, then the family is invertible.*

For example, if a family exists with $K = 2^{n(1-\eta)}$ and constant expansion factor $\gamma > 1$, then $1 - \frac{\log K\gamma}{n} = \eta - o(1)$, and hence the family is (m, γ^{-1}, η)-input restricting. It remains to be shown that an explicit family of unbalanced expander graphs with sufficiently small (i.e. polynomially-bounded) left-degree exists. Much work in this area has been devoted to *lossless* unbalanced expanders, i.e., with $\gamma \approx D$, but the best known constructions (cf. e.g. [28,25]) for this case lead to either super-polynomial degree or a much too small bound K for our choice of parameters. However, we are satisfied even if the expansion factor is much smaller than the left-degree, as long as the latter stays small, and it is possible to obtain such graphs by appropriately composing known constructions. We discuss the following result in the full version of this paper [23].[11]

Theorem 2. *For all polynomials γ and constants $\eta \in (0,1)$, and all functions m (polynomially-bounded in n), there exists an explicit family of expander graphs $G = (V_1, V_2, E)$ with $V_1 = \{0,1\}^m$, $V_2 = \{0,1\}^n$ which is $(2^{n(1-\eta)}, \gamma)$-expanding and has left-degree polynomially-bounded in n.*

Note that these techniques even allow to obtain slightly stronger results, for instance allowing η to be a moderately vanishing function. Combining this with Lemma 5 we see that for all constants $\epsilon \in (0,1)$ there exist explicit (m, δ, ϵ)-input-restricting families with δ^{-1} polynomial in n. However, by dropping the explicitness requirement, families with much better parameters exist. In particular, the following result is a simple application of the probabilistic method.

[11] Also note that a very similar result appears in unpublished work by Baltz et al.[3].

Lemma 6. *Let K and γ be arbitrary such that $K \cdot \gamma \leq 2^n$, and let m be such that $m \geq n$. There exists a graph $G = (V_1, V_2, E)$ where $V_1 = \{0,1\}^m$ and $V_2 = \{0,1\}^n$ which is (K, γ)-expanding and with left-degree $D = \left\lceil \frac{1+\gamma \log e + m}{n - \log(K\gamma)} + \gamma \right\rceil$.*

For example, setting $m = \ell = 2n$, $\gamma = 1$ and $K = 2^{n(1-\epsilon)}$, we obtain left-degree $D = 1 + \frac{2}{\epsilon} + (\log e + 1)/(\epsilon \cdot n)$. For $\epsilon = \frac{1}{4}$ and $n = 128$, this leads to a family of size 12 by Lemma 5. Furthermore in this case $t = 7$ and $\rho = 4$, and all these values do not grow with n. (And a similar reasoning applies to all constants $\epsilon > 0$.) With these parameters, the construction is of practical interest, as it only relies on the design of a secure component function $\{0,1\}^n \to \{0,1\}^n$ which may be very efficient. We hope this motivates further research on de-randomizing families of unbalanced expander graphs for a wider range of parameters.

5 Constructing Public Random Oracles

We first review a slightly generalized version of the *prefix-free Merkle-Damgård* construction [11]. Let n be the given output size, and let $\ell \geq n$. We are given both a compression function $f : \{0,1\}^{b+\ell} \to \{0,1\}^\ell$ and a *prefix-free padding scheme*, that is, a mapping $\mathsf{pad} : \{0,1\}^* \to \left(\{0,1\}^b\right)^+$ such that $\mathsf{pad}(s)$ is not a prefix of $\mathsf{pad}(s')$ for all distinct $s, s' \in \{0,1\}^*$. The *prefix-free Merkle-Damgård construction* $\mathbf{pfMD}_{b,\ell,n}(f)$ proceeds as follows. On input $s \in \{0,1\}^*$, it computes $s_1 \| \cdots \| s_l = \mathsf{pad}(s)$ (with $s_i \in \{0,1\}^b$) and the chaining values $v_i := f(s_i, v_{i-1})$ for all $1 \leq i \leq l$, where v_0 is set to some initialization vector $IV \in \{0,1\}^\ell$. Finally, the construction outputs the first n bits of v_l. The following theorem easily[12] follows from Theorem 2 in [11].

Theorem 3. *Let $\mathbf{F} : \{0,1\}^{\ell+b} \to \{0,1\}^\ell$ be a puRF and let $\mathbf{O} : \{0,1\}^* \to \{0,1\}^n$ be a puRO. The construction $\mathbf{pfMD}_{b,\ell,n}(\cdot)$ is an (α', σ')-reduction of $\mathbf{O}$ to $\mathbf{F}$ with $\alpha'(k) = \mathcal{O}((l_{\max} \cdot k)^2 \cdot 2^{-\ell})$ and $\sigma'(k) = k$, where $l_{\max}$ is the maximal length (of the padding) of a message input to the construction.*

We note that there exists a trade-off between the number of queries and the length of the queries to the construction.[13] This issue is inevitable in all iterated constructions. We take now $\ell, b > 0$ as in the above explanation, and some $\epsilon > 0$. We set $m := \ell + b$, and we let $\mathcal{E}$ be an explicit (m, δ, ϵ)-input restricting family of functions. If given only a compression function $\mathbf{R}' : \{0,1\}^{n+\phi(n)} \to \{0,1\}^n$ (for $\phi(n)$ defined as in Section 3.3), we obtain a construction $\mathbf{pfMD}_{b,\ell,n}(\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot))$ which replaces calls to the compression functions by calls to the construction $\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot)$. We obtain the following theorem using Lemma 2.

[12] The only difference with respect to the original result is that we allow the chaining value to be larger than the output value, i.e. $\ell > n$.

[13] A possible distinguishing strategy would consist of doing few very long queries, instead of many queries, and security is guaranteed only as long as $l_{\max} \cdot k < 2^{\ell/2}$.

Theorem 4. *The construction* $\mathbf{pfMD}_{b,\ell,n}(\mathbf{C}^{\mathcal{E}}_{\epsilon,m,\ell}(\cdot))$ *is an* $(\overline{\alpha},\overline{\sigma})$*-reduction of a puRO* $\mathbf{O} : \{0,1\}^* \rightarrow \{0,1\}^n$ *to* $\mathbf{R}'$*, where* $\overline{\alpha}(k) = \alpha((l_{\max}+1)k) + \alpha'((\delta+1)k)$ *and* $\overline{\sigma}(k) = \delta \cdot k$*, with* α *and* α' *as in Theorems 1 and 3, respectively.*

Setting $\ell > 2n(1-\epsilon)$ leads to security for all distinguishers such that $l_{\max} \cdot k \leq \Theta(2^{n(1-\epsilon)})$. We finally note that our approach also works with all other known constructions of a public random oracle from a public compression function, as for example the constructions of [6,10], or other constructions discussed in [11].

Setting ϵ small enough provides high levels of security for properties like preimage resistance, second preimage resistance, multicollision resistance, or CTFP preimage resistance [16], and also excludes the existence of attacks for these properties (up to the obtained bound), that is, even with respect to adversaries which perform enough queries to find collisions for the component function $f : \{0,1\}^n \rightarrow \{0,1\}^n$.

References

1. Aiello, W., Venkatesan, R.: Foiling birthday attacks in length-doubling transformations. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 307–320. Springer, Heidelberg (1996)
2. An, J.H., Bellare, M.: Constructing VIL-MACs from FIL-MACs: Message authentication under weakened assumptions. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 252–269. Springer, Heidelberg (1999)
3. Baltz, A., Jäger, G., Srivastav, A., Ta-Shma, A.: An explicit construction of sparse asymmetric connectors. Manuscript (2003)
4. Bellare, M., Goldreich, O., Krawczyk, H.: Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 270–287. Springer, Heidelberg (1999)
5. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences 61(3), 362–399 (2000)
6. Bellare, M., Ristenpart, T.: Multi-property-preserving hash domain extension and the EMD transform. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 299–314. Springer, Heidelberg (2006)
7. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS '93: Proceedings of the 1st ACM conference on Computer and Communications Security, pp. 62–73. ACM Press, New York (1993)
8. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. Journal of the ACM 51(4), 557–594 (2004)
9. Carter, J.L., Wegman, M.N.: Universal classes of hash functions. Journal of Computer and System Sciences 18(2), 143–154 (1979)
10. Chang, D., Lee, S., Nandi, M., Yung, M.: Indifferentiable security analysis of popular hash functions with prefix-free padding. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 283–298. Springer, Heidelberg (2006)
11. Coron, J.-S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle–Damgård revisited: How to construct a hash function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005)
12. Damgård, I.B.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1989)

13. Dodis, Y., Puniya, P.: On the relation between the ideal cipher and the random oracle models. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 184–206. Springer, Heidelberg (2006)
14. Hoch, J.J., Shamir, A.: Breaking the ICE — finding multicollisions in iterated concatenated and expanded (ICE) hash functions. In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 179–194. Springer, Heidelberg (2006)
15. Joux, A.: Multicollisions in iterated hash functions. Application to cascaded constructions. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 306–316. Springer, Heidelberg (2004)
16. Kelsey, J., Kohno, T.: Herding hash functions and the Nostradamus attack. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 183–200. Springer, Heidelberg (2006)
17. Kelsey, J., Schneier, B.: Second preimages on n-bit hash functions for much less than 2^n work. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 474–490. Springer, Heidelberg (2005)
18. Lucks, S.: A failure-friendly design principle for hash functions. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 474–494. Springer, Heidelberg (2005)
19. Maurer, U.: Indistinguishability of random systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002)
20. Maurer, U.: Abstract models of computation in cryptography. In: Smart, N.P. (ed.) Cryptography and Coding. LNCS, vol. 3796, pp. 1–12. Springer, Heidelberg (2005)
21. Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 21–39. Springer, Heidelberg (2005)
22. Maurer, U., Sjödin, J.: Single-key AIL-MACs from any FIL-MAC. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 472–484. Springer, Heidelberg (2005)
23. Maurer, U., Tessaro, S.: Full version of this paper. Available at `http://eprint.iacr.org/`
24. Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1989)
25. Moran, T., Shaltiel, R., Ta-Shma, A.: Non-interactive timestamping in the bounded storage model. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 460–476. Springer, Heidelberg (2004)
26. Shoup, V.: On fast and provably secure message authentication based on universal hashing. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 313–328. Springer, Heidelberg (1996)
27. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
28. Ta-Shma, A., Umans, C., Zuckerman, D.: Lossless condensers, unbalanced expanders, and extractors. In: STOC '01: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, pp. 143–152. ACM Press, New York (2001)