Tightly-Secure Pseudorandom Functions via Work Factor Partitioning

Tibor Jager*

Horst Görtz Institute for IT Security, Ruhr-University Bochum
tibor.jager@rub.de

Abstract. We introduce a new technique for tight security proofs called *work factor partitioning*. Using this technique in a modified version of the framework of Döttling and Schröder (CRYPTO 2015), we obtain the first generic construction of *tightly-secure* pseudorandom functions (PRFs) from PRFs with small domain.

By instantiating the small-domain PRFs with the Naor-Reingold function (FOCS 1997) or its generalization by Lewko and Waters (ACM CCS 2009), this yields the first fully-secure PRFs whose black-box security proof loses a factor of only $O(\log^2 \lambda)$, where λ is the security parameter.

Interestingly, our variant of the Naor-Reingold construction can be seen as a standard Naor-Reingold PRF (whose security proof has a loss of $\Theta(\lambda)$), where a special encoding is applied to the input before it is processed. The tightness gain comes almost for free: the encoding is very efficiently computable and increases the length of the input only by a constant factor smaller than 2.

1 Introduction

Pseudorandom functions. A pseudorandom function (PRF) is a function $F : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{G}$ with the following security property. For random $k \xleftarrow{\$} \mathcal{K}$, the function $F(k, \cdot)$ is computationally indistinguishable from a random function $R(\cdot)$, given oracle access to either $F(k, \cdot)$ or $R(\cdot)$. PRFs are a foundational cryptographic primitive with countless applications. They can be used to obtain simple and efficient constructions of message authentication codes, symmetric encryption, and key derivation algorithms, and form useful building blocks for many other primitives, like digital signatures. See [19,4,6,20,26], for example. While PRFs can be constructed generically from one-way functions (via pseudorandom generators) [19], this generic construction is rather inefficient and not "tight". Therefore we seek to construct efficient PRFs from as-weak-as-possible assumptions and with tight security proof.

Tight security. In a cryptographic security proof, we often consider an adversary $\mathcal{A}$ against a primitive like a PRF, and describe a reduction $\mathcal{B}$ that runs $\mathcal{A}$ as a subroutine to break some computational problem which is assumed to be hard. Let $(t_{\mathcal{A}}, \epsilon_{\mathcal{A}})$ and

* Supported by DFG grant JA 2445/1-1.

$(t_{\mathcal{B}}, \epsilon_{\mathcal{B}})$ denote the running time and success probability of $\mathcal{A}$ and $\mathcal{B}$, respectively. Then we say that adversary $\mathcal{B}$ *loses* a factor ℓ, if

$$\frac{t_{\mathcal{B}}}{\epsilon_{\mathcal{B}}} \geq \ell \cdot \frac{t_{\mathcal{A}}}{\epsilon_{\mathcal{A}}}$$

A reduction is considered "efficient", if ℓ is bounded by a polynomial in the security parameter. We say that a reduction is "tight", if ℓ is small. Our goal is to construct reductions $\mathcal{B}$ such that ℓ is as small as possible. Ideally we would like to have $\ell = O(1)$ constant, but there are many examples of cryptographic constructions and primitives where this is impossible to achieve, see [15,25,22,28,3] for instance.

The search for tight reductions is motivated by the theoretical search for provably-secure cryptosystems whose security guarantees are independent of adversarial behavior and the practical necessity of concrete security bounds for the theoretically-sound selection of cryptographic parameters, such as key lengths.

Black-box vs. non-black-box reductions. One can consider two different types of reductions:

1. *Non-black-box reductions*, which may use some *a priori* information about the adversary.
   For example, in the PRF security experiment a reduction may have to prepare for a certain number of oracle queries by the adversary. If this number is not known, then the reduction may prepare for too many or too few queries. This would either increase the running time or decrease the success probability of the reduction, at the cost of tightness.
2. *Black-box reductions*, which do not require any *a priori* information about a given adversary.

There are settings where it may be reasonable to assume that a reduction has access to certain *a priori* information about the adversary. For example, an adversary $\mathcal{A}$ may be given as a circuit with a certain number of "oracle query gates". A non-black-box reduction may count the number of these gates to estimate the number of oracle queries. However, we caution that this only yields an *upper bound* on the number of queries made by $\mathcal{A}$. For example, circuit $\mathcal{A}$ may contain a very large number of "oracle query gates", but with most of these gates lying on an execution path which is rarely or never visited. Here a reduction which counts the "oracle query gates" would prepare for too many oracle queries, an thus lose tightness.

Furthermore, a reduction is usually not able to perform any sophisticated analysis of the given adversary without losing tightness. For example, a reduction may run the adversary to determine the (expected) number of oracle queries, but this would increase the running time of the reduction.

In the context of tight reductions we should therefore consider the black-box setting, where no *a priori* information about the adversary is assumed. There are tightly-secure constructions of many primitives like public key encryption [5,21,18], digital signatures [8,31,21,1,9], key exchange [2], or (hierarchical) identity-based encryption [14,10]. All of these works consider a black-box setting without *a priori* information about the adversary.

PRFs from small-domain PRFs. A very elegant approach for the construction of PRFs was presented in [16]. Let us sketch their approach. Döttling and Schröder [16] start from *small-domain* PRFs, where the input space is polynomially bounded. Such PRFs can be constructed very efficiently, from weak complexity assumptions, and with tight security. The authors then show that small-domain PRFs can be turned *generically* into full-fledged PRFs with exponential-size domain (e.g., $\{0,1\}^\lambda$, where λ is the security parameter), via an intermediate tool called *bounded PRFs*.

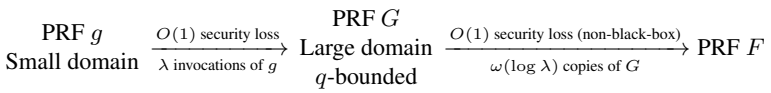

Fig. 1. High-level perspective on the approach of [16].

The framework of [16] performs two steps (cf. Figure 1). Starting with a small-domain PRF g, it first applies a generic *domain extension* technique to g. It is shown that a clever λ-wise execution of g yields an "intermediate" PRF G with large (exponential-size) domain. However, G is only secure against adversaries that issue at most q oracle queries, where q may be small. In a second step it is then shown how a full-fledged PRF F can generically be constructed from $\omega(\log \lambda)$ copies of G via a technique called *on-the-fly adaptation*.

The reduction in the first step is very tight, it loses only a negligibly small additive term. However, the reduction in the second step is only tight in a non-black-box setting, where the number of queries q of the PRF-adversary is known in advance. (One can turn this into a reduction black-box setting by simply guessing q, but this incurs a polynomial loss which makes the reduction non-tight).

Thus, so far there exists no generic construction of *tightly-secure* pseudorandom functions from small-domain PRFs, unless one considers non-black-box reductions.

Contributions. In this work we introduce a new technique for tight security proofs called *work factor partitioning*. Using this technique, we develop a variant of the approach of Döttling and Schröder [16] with the following properties:

- We obtain the first generic construction of full PRFs from bounded-domain PRFs, which is *tightly secure* (up to a small factor $O(\log \lambda)$) in a black-box setting which requires no *a priori* information about the adversary.
- Our generic construction is more efficient, since our construction of a bounded PRF G from a small-domain PRF g requires only a *single* invocation of g (in contrast to λ invocations in [16]). Our analysis of this step much simpler, too.
- In a non-black-box setting, where both the running time and the success probability of the adversary are known, we achieve a constant security loss of $O(1)$ (note that this is a stronger requirement than in [16], which only requires to know the number of oracle queries).

This gives rise to new variants of the Naor-Reingold PRF [29] and its generalization by Lewko and Waters [27]. Most interestingly, we consider a variant of the Naor-Reingold

construction, where a special encoding function E is applied to the input before it is processed by the PRF. That is, we consider a construction F' defined as

$$F'(K, x) := F(k, E(x))$$

where $K = (k, E)$, F is the Naor-Reingold PRF, and E is our special encoding function. We show that F' is *tightly* secure under the 1-Linear assumption (also known as the Decisional Diffie-Hellman assumption), with a security loss of only $O(\log^2 \lambda)$ in the black-box setting (and $O(\log \lambda)$ in a non-black-box setting). In comparison, the classical security proof of [29] loses a factor $\Theta(\lambda)$. The Naor-Reingold variant of [16] loses $O(\log \lambda)$, but only in a non-black-box setting. The tightness gain comes almost for free: the encoding scheme is very efficiently computable and increases the input length only by a factor of 2. Since the secret keys of the Naor-Reingold PRF grows linearly with the size of the input, we thus have secret keys that are larger by a factor of two, but receive much tighter security proofs in exchange.

Technical approach: work factor partitioning. Like Döttling and Schröder [16], we proceed in two steps (cf. Figure 2). The major difference of our approach is that we construct an intermediate tool with similar, but different security properties.

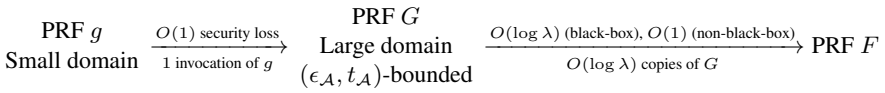

Fig. 2. High-level perspective on our approach.

Recall that Bellare and Ristenpart [7] have defined the *work factor* of an adversary $\mathcal{A}$ as $t_{\mathcal{A}}/\epsilon_{\mathcal{A}}$, where $t_{\mathcal{A}}$ is the running time and $\epsilon_{\mathcal{A}}$ is the success probability of $\mathcal{A}$. We start from a small-domain PRF g, and use g construct a large domain PRF G with security against adversaries with *bounded work factor*. (For simplicity we deviate from the standard definition of [7], and define the work factor as $q^2/\epsilon_{\mathcal{A}}$ in this work.) The construction is tightly-secure, it loses only a constant factor of 2.

In the second step, we then give a generic construction of a standard PRF F, which is composed of bounded-secure PRFs $G_1, \ldots, G_{\log \lambda}$. Here we require that G_j is secure against adversaries $\mathcal{B}_j$ with work factor $2q^2/\epsilon_{\mathcal{B}} \in [2^{2^{j-1}}, 2^{2^j}]$. Intuitively, our goal will be to turn a given adversary $\mathcal{A}$ on F into an adversary $\mathcal{B}_j$ on G_j, such that j is *as small as possible*. (This will enable us to instantiate G_j with tight security from an as weak as possible assumption).

To this end, we use a technique that we call *work factor partitioning*. We construct a family of reductions $\mathfrak{B} = (\mathcal{B}_1, \ldots, \mathcal{B}_{\log \lambda})$, where $\mathcal{B}_j$ is a *tight* reduction that uses an adversary $\mathcal{A}$ against F to break the bounded security of G_j. Given a PRF-adversary $\mathcal{A}$, we pick a reduction $\mathcal{B}_j \xleftarrow{\$} \mathfrak{B}$ at random, hoping that we pick j that satisfies $2^{2^{j-1}} < q^2/\epsilon_{\mathcal{A}} \leq 2^{2^j}$. Since $|\mathfrak{B}| = \log \lambda$ is very small, we pick the "right" reduction (i.e.,

the reduction $\mathcal{B}_j$ that allows us to break G_j with j as small as possible) correctly with probability $1/\log\lambda$. This yields a fully black-box reduction to the security of G_j with j as small as possible, with security loss of only $O(\log\lambda)$.

Considering alternatively a non-black-box setting, where the number of queries q and the success probability of a given adversary are explicitly known, we can compute j from $\epsilon_{\mathcal{A}}$ and q directly and thus do not have to guess. In this case we get a reduction with only constant security loss.

Efficiency. Both our constructions of G from g and of F from G are more efficient than the corresponding constructions of [16]. We obtain a tightly-secure bounded PRF G, which evaluates the underlying small-domain PRF g only *once* (where [16] needed λ evaluations of g). Also the construction of F from G is more efficient, but only very slightly. In our case $t = O(\log\lambda)$ functions $G_1, \ldots, G_t$ are sufficient, while [16] required $t = \omega(\log\lambda)$ to be slightly super-logarithmic.

2 Preliminaries

Let $\lambda \in \mathbb{N}$ denote a security parameter. All our results are in the asymptotic setting, that is, we view all expressions involving λ as functions in λ. This includes the running time $t_{\mathcal{A}} = t_{\mathcal{A}}(\lambda)$ and success probability $\epsilon_{\mathcal{A}} = \epsilon_{\mathcal{A}}(\lambda)$ of adversaries, even though we omit λ in this case to simplify our notation. Similarly, all algorithms implicitly receive the security parameter 1^λ as their first input.

Notation. If A is a finite set, then we write $a \xleftarrow{\$} A$ to denote the action of sampling a uniformly random element a from A. If A is a probabilistic algorithm, then $a \xleftarrow{\$} A(x)$ denotes the action of running $A(x)$ on input x with uniform coins and output a.

Let $\mathbb{G}$ be an algebraic group with generator g. Following the "implicit notation" of [17], we write $[a]$ shorthand for g^a whenever the reference to some given generator g is clear. This notation generalizes to vectors, where we write $[\mathbf{a}]$ shorthand for $g^{\mathbf{a}} = (g^{a_1}, \ldots, g^{a_k})^\top$ with $\mathbf{a} = (a_1, \ldots, a_k)^\top \in \mathbb{Z}_q^k$, and to matrices in the analogous way.

Pseudorandom Functions. Let $\mathcal{K}, \mathcal{D}$ be sets such that there is an efficient algorithm that samples uniformly random elements $k \xleftarrow{\$} \mathcal{K}$. Let $F : \mathcal{K} \times \mathcal{D} \to \mathcal{G}$ be an efficiently computable function. For an adversary $\mathcal{A}$ and $i \in \{0, 1\}$ define the following security experiment $\mathsf{Exp}^{\mathsf{prf},i}_{\mathcal{A},F}(\lambda)$.

1. The experiment generates a random key $k \xleftarrow{\$} \mathcal{K}$.
2. The experiment provides adversary $\mathcal{A}^{\mathcal{O}}(1^\lambda)$ with an oracle $\mathcal{O}$ which takes as input $x \in \mathcal{D}$ and responds as follows.
   - If $i = 0$, then $\mathcal{O}$ computes $y = F(k, x)$ and responds with y.
   - If $i = 1$, then $\mathcal{O}$ returns $S(x)$, where $S : \mathcal{D} \to \mathcal{G}$ is a random function.
3. When the adversary terminates and outputs a bit b, then the experiment outputs b.

Definition 1. *We say that adversary $\mathcal{A}$ $(\epsilon_{\mathcal{A}}, t_{\mathcal{A}}, q)$-breaks the pseudorandomness of F, if $\mathcal{A}$ runs in time $t_{\mathcal{A}}$, issues q queries in the PRF security experiment, and*

$$\left|\Pr\left[\mathsf{Exp}^{\mathsf{prf},0}_{\mathcal{A},F}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{prf},1}_{\mathcal{A},F}(\lambda) = 1\right]\right| \geq \epsilon_{\mathcal{A}}$$

Universal hash functions.

Definition 2 ([13]). *A family $\mathcal{H}$ of hash functions mapping finite set $\mathcal{D}$ to finite set $\{0,1\}^v$ is* universal*, if for all $x, x' \in \mathcal{D}$ with $x \neq x'$ holds that*

$$\Pr_{H \xleftarrow{\$} \mathcal{H}} [H(x) = H(x')] \leq 2^{-v}$$

Universal hash functions can be constructed very efficiently and without additional complexity assumptions, see e.g. [13,24].

The Decisional k-Linear assumption.

Definition 3. *Let $\mathbb{G}$ be a finite group of order q with generator g. Let*

$$[\mathbf{t}] := [(h_1, \ldots, h_k, h_1 \cdot s_1, \ldots, h_k \cdot s_k)]$$

for $h_1, \ldots, h_k, s_1, \ldots, s_k \xleftarrow{\$} \mathbb{Z}_q$, and let $r \xleftarrow{\$} \mathbb{Z}_q$. We say that adversary $\mathcal{A}$ ($\epsilon_{\mathcal{A}}$,$t_{\mathcal{A}}$)-breaks the k-Linear assumption in $\mathbb{G}$, if it runs in time $t_{\mathcal{A}}$ and

$$\epsilon_{\mathcal{A}} \leq \left| \Pr \left[\mathcal{A}([1], [\mathbf{t}], \left[\sum_{i=1}^{k} s_i \right]) = 1 \right] - \Pr \left[\mathcal{A}([1], [\mathbf{t}], [r]) = 1 \right] \right|$$

The Decisional k-Linear assumption [11] is a generalization of the Decisional Diffie-Hellman (DDH) assumption, which gets weaker with increasing k [23,32]. The DDH assumption corresponds to the case $k = 1$. The k-Linear assumption with $k \geq 2$ can, for instance, be used in groups with symmetric bilinear pairing, where the DDH assumption does not hold.

3 Generic Constructions

In this section we describe a new generic approach to construct pseudorandom functions from pseudorandom functions with small domain. Similar to the approach of [16], we first construct pseudorandom functions with bounded security as an "intermediate" primitive. The main difference is that we consider a different form of bounded security. Essentially, [16] considered a setting where the number of oracle queries in the PRF security experiment is bounded. Instead, we consider a setting where the *work-factor* of a given adversary is bounded. This allows us to obtain a more efficient construction

Then we describe a generic construction of standard (large-domain) pseudorandom functions from PRFs that are bounded-secure in our sense. The most important feature of this construction is that we obtain *tight* security with a black-box reduction, even though the underlying bounded-secure PRF depends on the work-factor of a given adversary. In contrast, [16] did not consider black-box reductions here, as they assume that the number of oracle queries of the adversary is known *a priori*.

3.1 Bounded-secure PRFs from small-domain PRFs

In this section we describe a generic construction of pseudorandom functions with security against adversaries with bounded work factor from small-domain PRFs.

Construction. Let $\beta > 0$ and let $\mathcal{H}$ be a family of universal hash functions with domain $\mathcal{D}$ and range $\{0,1\}^\beta$. Let $g : \mathcal{K} \times \{0,1\}^\beta \to \{0,1\}^w$ be a function with key space $\mathcal{K}$ and (small) domain $\{0,1\}^\beta$. Define function G with key space $\mathcal{K} \times \mathcal{H}$ and (large) domain $\mathcal{D}$ as

$$G(K, x) = g(k, H(x)) \tag{1}$$

where $K = (k, H)$ for $k \in \mathcal{K}$ and $H \in \mathcal{H}$.

Theorem 1. *From each adversary $\mathcal{A}$ that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of construction G with $\log(2q^2/\epsilon_\mathcal{A}) \leq \beta$ we can construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ as a black-box to $(t_\mathcal{B}, \epsilon_\mathcal{B})$-break the security of g with*

$$t_\mathcal{B} \approx t_\mathcal{A} \qquad \textit{and} \qquad \epsilon_\mathcal{B} \geq \epsilon_\mathcal{A}/2$$

PROOF. Consider the following sequence of games, where $\mathcal{O}_i$ denotes the function provided by the PRF security experiment to $\mathcal{A}$ in Game i. Note that the sequence of games is identical to the sequence of games in the proof of [16, Theorem 1], but their analysis is different.

Game 0. This game is identical to the "real" security experiment $\mathsf{Exp}^{\mathsf{prf},0}_{\mathcal{A},G}(\lambda)$ with function G from Equation (1) and adversary $\mathcal{A}$. Thus, we have

$$\mathcal{O}_0(x) = G(k, x) = g(k, H(x))$$

Game 1. This game is identical to Game 0, except that we replace function $g(k, \cdot)$ with a random function. Thus, the experiment in this game provides $\mathcal{A}$ with

$$\mathcal{O}_1(x) = R(H(x))$$

where R is a random function.

Game 2. This game is identical to Game 1, except that we now replace G_1 with a random function. Thus, the experiment in this game provides $\mathcal{A}$ with

$$\mathcal{O}_2(x) = S(x)$$

where S is random. Therefore Game 2 is identical to the "random" security experiment $\mathsf{Exp}^{\mathsf{prf},1}_{\mathcal{A},G}(\lambda)$.

Analysis. Let X_i denote the event that $\mathcal{A}$ outputs 1 in Game i. By definition of $\epsilon_\mathcal{A}$ and the triangle inequality we have

$$\begin{aligned}
\epsilon_\mathcal{A} &\leq \left| \Pr\left[\mathsf{Exp}^{\mathsf{prf},0}_{\mathcal{A},G}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{prf},1}_{\mathcal{A},G}(\lambda) = 1\right] \right| \\
&= |\Pr[X_0] - \Pr[X_2]| \\
&= |\Pr[X_0] - \Pr[X_1]| + |\Pr[X_1] - \Pr[X_2]|
\end{aligned} \tag{2}$$

We first show that

$$|\Pr[X_1] - \Pr[X_2]| < \frac{\epsilon_{\mathcal{A}}}{2} \tag{3}$$

To see this, consider an execution of a $\mathcal{A}$ in Game 1 with $\mathcal{O}_1(x) = R(H(x))$. When $\mathcal{A}$ queries $\mathcal{O}_1(x_1)$, then it receives back $y_1 = R(H(x_1))$, which is uniformly random and independent of H, because r is a random function. In particular, y_1 contains no information about H. Next, $\mathcal{A}$ may query $y_2 = \mathcal{O}_1(x_2)$. The experiment will evaluate R on a different position than in the first query, unless $H(x_2) = H(x_1)$. Due to the universality of H and the fact that y_1 is independent of H, this happens with probability at most $1/2^\beta$. Therefore $\mathcal{A}$ will receive another uniformly random value y_2, which is independent of H, except with probability at most $1/2^\beta$. Continuing this argument inductively over all q queries of $\mathcal{A}$, we see that on its i-th query $\mathcal{A}$ will receive a random response which is independent of H, except with probability $(i-1)/2^\beta$, provided that all previous responses were independent of H.

A union bound now yields that therefore Game 1 and Game 2 are indistinguishable, except with probability

$$|X_1 - X_2| \leq \sum_{i=2}^{q} \frac{i-1}{2^\beta} < \frac{q^2}{2^\beta} = \frac{q^2\epsilon_{\mathcal{A}}}{2q^2} = \frac{\epsilon_{\mathcal{A}}}{2}$$

Note that we use here that $\beta \geq \log(2q^2/\epsilon_{\mathcal{A}})$.

It remains to construct adversary $\mathcal{B}$ against g. $\mathcal{B}$ plays the $\mathsf{Exp}^{\mathsf{prf},i}_{\mathcal{B},g}(\lambda)$ security experiment with oracle $\mathcal{O}_g$, and runs $\mathcal{A}$ as a subroutine by simulating the $\mathsf{Exp}^{\mathsf{prf},i}_{\mathcal{A},G}(\lambda)$ experiment with oracle $\mathcal{O}_G$. $\mathcal{B}$ proceeds exactly like Game 0, except that it uses its oracle to implement function g. That is, at the beginning of the experiment $\mathcal{B}$ samples a random hash function $H \xleftarrow{\$} \mathcal{H}$. When adversary $\mathcal{A}$ outputs x to $\mathcal{O}_G$, then $\mathcal{B}$ computes $H(x)$, queries $y := \mathcal{O}_g(H(x))$, and returns y to $\mathcal{A}$. When $\mathcal{A}$ terminates, then $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs and terminates.

Note that if $\mathcal{O}_g$ implements g, then the view of $\mathcal{A}$ is identical to its view in Game 0, while if if $\mathcal{O}_g$ implements a random function, then $\mathcal{A}$'s view is identical to Game 1. Thus, $\mathcal{B}$ has advantage at least $\epsilon_{\mathcal{B}} \geq |\Pr[X_0] - \Pr[X_1]|$. By plugging (3) into (2) we obtain

$$\epsilon_{\mathcal{B}} \geq |\Pr[X_0] - \Pr[X_1]| \geq \epsilon_{\mathcal{A}} - \frac{\epsilon_{\mathcal{A}}}{2} = \frac{\epsilon_{\mathcal{A}}}{2}$$

□

Comparison to the bounded-secure PRFs of Döttling and Schröder. We remark that the construction in (1) is similar to the construction of bounded-secure PRFs from [16] in that the construction of G first applies a universal hash function H to its input, and then evaluates a small-domain PRF on the result. The main difference is that the construction in [16] is

$$G'(K, x) = \bigoplus_{i=1}^{\lambda} g(k, H_\ell(x))$$

which requires λ evaluations of g and λ independent hash functions $H_1, \ldots, H_\lambda$, while our construction from Equation (1) requires only a single evaluation of g and one hash function H.

From a high level perspective the proof of Theorem 1 uses the same sequence of games as the proof of [16, Theorem 1], but their analysis is different. In particular, it is simpler and more direct, as we do not need the concept of cover-free vectors used in [16].

3.2 Unbounded Generic Construction from Bounded-secure PRFs

In this section we give a generic, tightly-secure construction of standard (large-domain) pseudorandom functions with provable security against arbitrary polynomial-time adversaries. The security analysis is based on work-factor partitioning.

Construction. For $\ell \in \{1, \ldots, \log \lambda\}$ let $G_{2^\ell} : \mathcal{K}_\ell \times \mathcal{D} \to \{0,1\}^w$ be functions with key space $\mathcal{K}_\ell$ and domain $\mathcal{D}$. Define function F_λ with key space $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_{\log \lambda}$, domain $\mathcal{D}$, and range $\{0,1\}^w$ as

$$F_\lambda(K, x) = \bigoplus_{\ell=1}^{\log \lambda} G_{2^\ell}(k_\ell, x) \tag{4}$$

where $k_\ell \xleftarrow{\$} \mathcal{K}_\ell$ for $\ell \in \{1, \ldots, \log \lambda\}$ and $\oplus$ is the bitwise exclusive-or operation.

Security analysis. Before we state the formal security results for the above construction, we prove the following lemma, which contains the core argument of the security proof.

Lemma 1. *One can efficiently construct a family of adversaries $\mathfrak{B} = \{B_1, \ldots, B_{\log \lambda}\}$ such that the following holds: Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of F_λ and let $j \in \{1, \ldots, \log \lambda\}$ such that*

$$2^{j-1} < \log(2q^2/\epsilon_\mathcal{A}) \leq 2^j \tag{5}$$

Then $\mathcal{B}_j \in \mathfrak{B}$ $(t_\mathcal{B}, \epsilon_\mathcal{B}, q)$-breaks the of G_{2^j} with

$$t_\mathcal{B} \approx t_\mathcal{A} \qquad \textit{and} \qquad \epsilon_\mathcal{B} \geq \epsilon_\mathcal{A}$$

Remark 1. Note that the lemma states essentially that $\mathcal{B}_j$ is able to break the security of G_{2^j} with work factor

$$\frac{2q^2}{\epsilon_\mathcal{B}} \leq \frac{2q^2}{\epsilon_\mathcal{A}}$$

Thus, for F_λ to be secure it suffices to instantiate it with pseudorandom functions $G_{2^1}, \ldots, G_{2^{\log \lambda}}$, such that G_{2^j} is secure against adversaries with work factor $2q^2/\epsilon_\mathcal{A}$. Intuitively, the smaller the work factor of $\mathcal{A}$, the smaller is the security required from the underlying PRF. For each work factor below 2^λ (which holds for all adversaries with polynomially-bounded $t_\mathcal{A} = t_\mathcal{A}(\lambda)$ and $\epsilon_\mathcal{A} = \epsilon_\mathcal{A}(\lambda)$), construction F_λ contains a function G_{2^j} which is "just secure enough" for $\mathcal{A}$. Note also that we have $2^j \in O(\log \lambda)$.

Moreover, the family of reductions $\mathfrak{B}$ has only size $\log \lambda$. This will be useful for tight black-box reductions, where q and $\epsilon_\mathcal{A}$ may be unknown and hard to determine for a given adversary $\mathcal{A}$.

PROOF. Consider an adversary $\mathcal{B}_j$ in the PRF security experiment with function G_{2^j} and oracle $\mathcal{O}_G$. $\mathcal{B}_j$ runs $\mathcal{A}$ as a subroutine, by simulating the PRF security experiment with function F_λ for $\mathcal{A}$.

Initialization. $\mathcal{B}_j$ samples a key $k_\ell \xleftarrow{\$} \mathcal{K}_\ell$ for each $\ell \in \{1, \ldots, \log \lambda\} \setminus \{j\}$.

Handling of oracle queries. Whenever $\mathcal{A}$ outputs $x_i \in \mathcal{D}$, $\mathcal{B}_j$ queries its oracle to obtain $y_i := \mathcal{O}_G(x_i)$, computes

$$z_i := y_i \oplus \bigoplus_{\ell=1, \ell \neq j}^{\log \lambda} G_{2^\ell}(k_\ell, x)$$

and returns z_i to $\mathcal{A}$.

Finalization. Finally, when $\mathcal{A}$ terminates, then $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs, and terminates.

Analysis of $\mathcal{B}_j$. Note that the running time of $\mathcal{B}_j$ is essentially identical to the running time of $\mathcal{A}$, thus we have $t_\mathcal{B} \approx t_\mathcal{A}$. If $\mathcal{O}_G(x) = G_{2^j}(k_j, x)$, then it holds that $z_i = \bigoplus_{\ell=1}^{\log \lambda} G_{2^\ell}(k_\ell, x)$ for all $i \in \{1, \ldots, q\}$. Thus, the view of $\mathcal{A}$ is identical to the "real" PRF-security experiment $\mathsf{Exp}^{\mathsf{prf},0}_{\mathcal{A},F_\lambda}$ with F_λ. If $\mathcal{O}_G(x)$ implements a random function, then z_i is uniformly random for all $i \in \{1, \ldots, q\}$. Thus, in this case the view of $\mathcal{A}$ is identical to the "random" PRF-security experiment $\mathsf{Exp}^{\mathsf{prf},1}_{\mathcal{A},F_\lambda}$. This yields

$$\epsilon_\mathcal{B} \geq \epsilon_\mathcal{A}$$

□

Now we can state the formal security claims for construction F_λ.

Theorem 2. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of F_λ, such that $t_\mathcal{A}$ and $1/\epsilon_\mathcal{A}$ are bounded by a polynomial in λ. Then we can construct an adversary $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B}, q)$-breaks G_β with*

$$t_\mathcal{B} \approx t_\mathcal{A} \qquad \textit{and} \qquad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}}{\log \lambda} \qquad \textit{and} \qquad \beta \in O(\log \lambda)$$

PROOF. We construct the family of adversaries $\mathfrak{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_{\log \lambda}\}$ from Lemma 1, pick $j \xleftarrow{\$} \{1, \ldots, \log \lambda\}$ at random, and run algorithm $\mathcal{B}_j$. Since $t_\mathcal{A}$ and $1/\epsilon_\mathcal{A}$ are bounded by a polynomial in λ, also $2q^2/\epsilon_\mathcal{A}$ is polynomially bounded, such that we have $2q^2/\epsilon_\mathcal{A} < 2^\lambda$. With probability $1/\log \lambda$ we choose j that satisfies (5), and in this case we have $\beta = 2^j \in O(\log \lambda)$, so the claim follows from Lemma 1. □

Note that the above bounds are nearly tight, up to a small loss of only $\log \lambda$, which stems from that fact that we may not be able to efficiently determine q and $\epsilon_\mathcal{A}$ for a given adversary $\mathcal{A}$.

Assuming that we are able to efficiently determine q and $\epsilon_{\mathcal{A}}$ (in a non-black-box setting where e.g. the code of $\mathcal{A}$ provides us with this information) we can get a fully-tight reduction with only constant security loss.

Theorem 3. *Let $\mathcal{A}$ be an adversary that $(t_{\mathcal{A}}, \epsilon_{\mathcal{A}}, q)$-breaks the security of F_λ, such that q and $1/\epsilon_{\mathcal{A}}$ are bounded by a polynomial in λ. Provided that q and $\epsilon_{\mathcal{A}}$ are explicitly known, we can construct an adversary $\mathcal{B}$ that $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}}, q)$-breaks G_β with*

$$t_{\mathcal{B}} \approx t_{\mathcal{A}} \qquad \textit{and} \qquad \epsilon_{\mathcal{B}} \geq \epsilon_{\mathcal{A}} \qquad \textit{and} \qquad \beta \in O(\log \lambda)$$

The proof of Theorem 3 is identical to the proof of Theorem 2, except that we do not have to guess j because $t_{\mathcal{A}}$ and $\epsilon_{\mathcal{A}}$ are explicitly known.

Remark 2. We remark the the construction in (4) and its security analysis generalize from PRFs with range $\{0,1\}^w$ to any set $\mathcal{G}$, if the exclusive-or operation $\oplus$ over $\{0,1\}^w$ is replaced with an arbitrary operation $\otimes$ over $\mathcal{G}$ that satisfies the following two properties:

1. For uniformly distributed $y \xleftarrow{\$} \mathcal{G}$, the distributions $(x, x \otimes y)$ and (x, y) are identical for all $x \in \mathcal{G}$.
2. Operation $\otimes$ need not be efficiently computable, but we require that there exists an efficient algorithm that, given function value $G_{2^j}(k_j, x)$, input value x, and keys k_ℓ for all $\ell \in \{1, \ldots, \log \lambda\} \setminus \{j\}$, computes

$$F_\lambda(K, x) = \bigotimes_{\ell=1}^{\log \lambda} G_{2^\ell}(k_\ell, x)$$

This will be useful for the analysis of PRFs with range $\mathcal{G} = \mathbb{G}$, where $\mathbb{G}$ is an algebraic group.

Comparison to the generic PRF construction of Döttling and Schröder. Most importantly, the security analysis of the corresponding construction in [16] required that the number of queries q of $\mathcal{A}$ in the PRF experiment is known to achieve a reduction with constant security loss $O(1)$. Without this *a priori* information, the reduction in [16] has to guess q, which leads to a security loss of $O(p(\lambda))$ for some polynomial p. In contrast, we give a reduction which requires no information about $\mathcal{A}$, but loses a factor of $O(\log \lambda)$ in the black-box setting. In non-black-box setting, where both q and $\epsilon_{\mathcal{A}}$ are known (which is a stronger requirement than in [16]), we also achieve a constant security loss.

The generic construction in [16] needed $\omega(\log \lambda)$ copies of the underlying PRF. The above construction is slightly more efficient, as it needs only $O(\log \lambda)$ copies. But this efficiency gain is very small and may not be significant in practice, so we view this rather as a theoretical improvement.

3.3 Direct construction of standard PRFs from small-domain PRFs

For $\ell \in \{1, \ldots, \log \lambda\}$ let $g_{2^\ell} : \mathcal{K}_\ell \times \{0,1\}^{2^\ell} \to \{0,1\}^w$ be a function with key space $\mathcal{K}_\ell$, and let $\mathcal{H}_\ell$ be a family of universal hash functions with domain $\mathcal{D}$ and range

$\{0,1\}^{2^\ell}$. Define function F with key space $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_{\log \lambda}$, domain $\mathcal{D}$, and range $\{0,1\}^w$ as

$$F(K, x) = \bigoplus_{\ell=1}^{\log \lambda} g_{2^\ell}(k'_\ell, H(x)) \tag{6}$$

where $K = (k_1, \ldots, k_{\log \lambda})$ with $k_\ell = (k'_\ell, H_\ell) \in \mathcal{K}_\ell \times \mathcal{H}_\ell$ for $\ell \in \{1, \ldots, \log \lambda\}$.

Security analysis. By combining Theorem 1 with Theorem 2 we obtain the following statement.

Theorem 4. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of F, such that $t_\mathcal{A}$ and $1/\epsilon_\mathcal{A}$ are bounded by a polynomial in λ. Then we can construct an adversary $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B}, q)$-breaks g_{2^ℓ} with*

$$t_\mathcal{B} \approx t_\mathcal{A} \qquad \textit{and} \qquad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}}{2 \log \lambda} \qquad \textit{and} \qquad 2^\ell \in O(\log \lambda)$$

Note that the above bounds are nearly tight, up to a small loss of only $O(\log \lambda)$.

Again, additionally assuming that we are able to efficiently determine $t_\mathcal{A}$ and $\epsilon_\mathcal{A}$ for a given adversary $\mathcal{A}$, we can get a fully-tight reduction with only constant security loss. By combining Theorem 1 with Theorem 3 we obtain the following.

Theorem 5. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of F, such that q and $1/\epsilon_\mathcal{A}$ are bounded by a polynomial in λ. Provided that q and $\epsilon_\mathcal{A}$ are explicitly known, we can construct an adversary $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B}, q)$-breaks g_{2^ℓ} with*

$$t_\mathcal{B} \approx t_\mathcal{A} \qquad \textit{and} \qquad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}}{2} \qquad \textit{and} \qquad 2^\ell \in O(\log \lambda)$$

4 Tightly-Secure Number-Theoretic PRFs

In this section we show how the generic techniques described in Section 3 can be used to easily obtain a tight security proof for simple variants of the pseudorandom functions of Naor and Reingold [29] and Lewko and Waters [27].

Tighly-secure Naor-Reingold PRFs via Input Encoding. We first consider a "Naor-Reingold PRF with encoded input", which is identical to the standard Naor-Reingold function, except that a special encoding function E is applied to the input before it is processed by the PRF. When instantiated in a group of order q, we will be able to use the commutativity of multiplication in $\mathbb{Z}_q$ to explain this variant of the Naor-Reingold function as a particular instantiation of our construction, which in turn allows us to give a tight security proof under the Decisional 1-Linear assumption (i.e., Decisional Diffie-Hellmann).

Tighly-secure PRFs from the Decisional k-Linear asusmption. This approach seems not to generalize in a straightforward way to the Lewko-Waters PRF [27], essentially because here a product of matrices is computed "in the exponent", which is not commutative. However, via the generic approach from Section 3 we obtain a different construction of tightly-secure pseudorandom functions from the Decisional k-Linear assumption for any $k \geq 1$ with tight security reduction in the black-box setting.

4.1 Tightly-secure Naor-Reingold PRFs from Input Encoding

The Naor-Reingold PRF. Let $\mathbb{G}$ be an abelian group of prime order q, let $\mathbb{G}^* := \mathbb{G}\backslash\{1\}$, and let $g \xleftarrow{\$} \mathbb{G}^*$ be random. Note that g is a generator of $\mathbb{G}$. Let $\beta \in \mathbb{N}$. The Naor-Reingold pseudorandom function [29], instantiated with domain $\{0,1\}^\beta$, is defined as

$$F^\beta_{\mathsf{NR}} : (\mathbb{Z}_q^*)^\beta \times \{0,1\}^\beta \to \mathbb{G}^* \qquad F^\beta_{\mathsf{NR}}(k,x) := \left[\prod_{i=1}^{\beta} a_i^{x_i}\right] \tag{7}$$

where $k = (a_1, \ldots, a_\beta) \in (\mathbb{Z}_q^*)^\beta$ and $x = (x_1, \ldots, x_\beta) \in \{0,1\}^\beta$. The following theorem is from [29].

Theorem 6 ([29]). *From each adversary $\mathcal{A}$ that $(\epsilon_{\mathcal{A}}, t_{\mathcal{A}}, q)$-breaks the security of F^β_{NR} with input space $\{0,1\}^\beta$, we can construct an adversary $\mathcal{B}$ that $(\epsilon_{\mathcal{B}}, t_{\mathcal{B}})$-breaks the 1-Linear assumption (aka. Decisional Diffie-Hellman) in $\mathbb{G}$ with*

$$t_{\mathcal{B}} \approx t_{\mathcal{A}} \qquad \textit{and} \qquad \epsilon_{\mathcal{B}} \geq \frac{\epsilon_{\mathcal{A}}}{\beta}$$

Note that the security loss is linear in the size of the input space. In particular, if $\beta \in O(\log \lambda)$, then the security loss is only logarithmic in λ.

Encoding families. For each $\ell \in \{1, \ldots, \log \lambda\}$ let $\mathcal{H}_\ell$ be a family of universal hash functions with domain $\mathcal{D}$ and range $\{0,1\}^{2^\ell}$. Define

$$\mathcal{E} := \mathcal{H}_1 \times \cdots \times \mathcal{H}_{\log \lambda}$$

where each $E = (H_1, \ldots, H_{\log \lambda}) \in \mathcal{E}$ defines the function

$$E(x) := H_1(x) || \cdots || H_{\log \lambda}(x) \tag{8}$$

Note that

$$|E(x)| = \sum_{\ell=1}^{\log \lambda} |H_i(x)| = \sum_{i=1}^{\log \lambda} 2^\ell = 2^{\log \lambda + 1} - 2 = 2(\lambda - 1)$$

and thus E is a function $E : \mathcal{D} \to \{0,1\}^{2(\lambda-1)}$.

Naor-Reingold with input encoding. Let $\mathcal{E}$ be the above family of efficiently computable functions E with domain $\mathcal{D}$ and range $\{0,1\}^{2(\lambda-1)}$. Let $F_{\mathsf{NR}}^{2(\lambda-1)}$ be the Naor-Reingold PRF from (7), instantiated with $\beta = 2(\lambda-1)$. The Naor-Reingold pseudo-random function with $\mathcal{E}$-encoded input is the function defined as

$$F_{\mathsf{NR}}^{\mathcal{E}} : \left((\mathbb{Z}_q^*)^{2(\lambda-1)} \times \mathcal{E}\right) \times \mathcal{D} \to \mathbb{G}^* \qquad F_{\mathsf{NR}}^{\mathcal{E}}(K,x) := F_{\mathsf{NR}}^{2(\lambda-1)}(k, E(x))$$

with $K = (k, E) \in (\mathbb{Z}_q^*)^{\beta} \times \mathcal{E}$.

Security analysis. Again we can use work factor partitioning to analyze the security of this construction. The main idea behind the security proof is that the encoding $\mathcal{E}$ described above allows us to view $F_{\mathsf{NR}}^{\mathcal{E}}$ as a composition of $\ell = O(\log \lambda)$ copies of the standard Naor-Reingold function F_{NR}. This allows us to apply Theorems 4 and 5 to reduce breaking $F_{\mathsf{NR}}^{\mathcal{E}}$ to breaking function F_{NR}^{β} with small domain $\{0,1\}^{\beta}$, where $\beta \in O(\log \lambda)$. Again, the reductions are tight (with a security loss of $O(\log \lambda)$ in the black-box setting and $O(1)$ in the non-black-box setting).

We begin with the following lemma, which again captures the core argument of the proof.

Lemma 2. *One can efficiently construct a family of adversaries $\mathfrak{B} = \{B_1, \ldots, B_{\log \lambda}\}$ such that the following holds: Let $\mathcal{A}$ be an adversary that $(t_{\mathcal{A}}, \epsilon_{\mathcal{A}}, q)$-breaks the security of $F_{\mathsf{NR}}^{\mathcal{E}}$ and let $j \in \{1, \ldots, \log \lambda\}$ such that*

$$2^{j-1} < \log(2q^2/\epsilon_{\mathcal{A}}) \leq 2^j \tag{9}$$

Then $\mathcal{B}_j \in \mathfrak{B}$ $(t_{\mathcal{B}}, \epsilon_{\mathcal{B}}, q)$-breaks the security of the Naor-Reingold PRF $F_{\mathsf{NR}}^{2^j}$ with domain $\{0,1\}^{2^j}$ with

$$t_{\mathcal{B}} \approx t_{\mathcal{A}} \qquad \textit{and} \qquad \epsilon_{\mathcal{B}} \geq \epsilon_{\mathcal{A}}$$

Note that we have $2^j \in O(\log \lambda)$, thus the lemma claims that we can break the security of a Naor-Reingold PRF with very small input space.

PROOF. Instead of giving a direct security proof of $F_{\mathsf{NR}}^{\mathcal{E}}$ under the 1-Linear assumption, we show that $F_{\mathsf{NR}}^{\mathcal{E}}$ is actually a specific instantiation of the generic construction from Section 3.3, such that Lemma 2 follows from Lemma 1.

To this end, let us introduce some notation. For a given Naor-Reingold secret key $(a_1, \ldots, a_{2(\lambda-1)}) \in (\mathbb{Z}_q^*)^{2(\lambda-1)}$, let us define

$$\begin{aligned} k_1 &:= (a_1, a_2) \\ k_2 &:= (a_3, \ldots, a_6) \\ k_3 &:= (a_7, \ldots, a_{14}) \\ &\vdots \\ k_{\log \lambda} &:= (a_{\lambda-1}, \ldots, a_{2(\lambda-1)}) \end{aligned}$$

Using this notation and the definition of $E(x) := H_1(x)||\cdots||H_{\log\lambda}(x)$ from (8), we can write $F^{\mathcal{E}}_{\mathsf{NR}}$ equivalently as a composition of $\log\lambda$ copies of the Naor-Reingold PRFs F_{NR}:

$$F^{\mathcal{E}}_{\mathsf{NR}}(K,x) = \bigotimes_{\ell=1}^{\log\lambda} F^{2^\ell}_{\mathsf{NR}}(k_\ell, H_\ell(x)) \tag{10}$$

with operation $\otimes : \mathbb{G}^* \times \mathbb{G}^* \to \mathbb{G}^*$ defined with respect to the given generator g of $\mathbb{G}$ as $[x] \otimes [y] = [xy]$ for all $x, y \in \mathbb{Z}_p^*$.

This perspective on the Naor-Reingold function with encoded inputs allows us to view $F^{\mathcal{E}}_{\mathsf{NR}}$ as a specific instantiation of the generic construction considered in Lemma 1, applied to $\log\lambda$ copies of the "small-domain" functions $F^{2^\ell}_{\mathsf{NR}}$. Thus, the family of adversaries $\mathfrak{B}$ proceeds exactly like the family of adversaries from the proof of Lemma 1, using the following two claims (cf. Remark 2).

Lemma 3. *For uniformly distributed $[y] \xleftarrow{\$} \mathbb{G}^*$, the distributions $([x], [x] \otimes [y])$ and $([x], [y])$ are identical for all $[x] \in \mathbb{G}^*$.*

This follows directly from the fact that $(\mathbb{G}^*, \otimes)$ is isomorphic to $(\mathbb{Z}_q^*, \cdot)$.

Lemma 4. *Even though $\otimes$ is not necessarily efficiently computable, there exists an efficient algorithm that, given function value $F_{\mathsf{NR}}(k_j, H_j(x))$, input value x, and keys k_ℓ for all $\ell \in \{1, \ldots, \log\lambda\} \setminus \{j\}$, computes*

$$F^{\mathcal{E}}_{\mathsf{NR}}(K,x) = \bigotimes_{\ell=1}^{\log\lambda} F_{\mathsf{NR}}(k_\ell, H_\ell(x))$$

Its running time is dominated by a single exponentiation in $\mathbb{G}$.

The running time of each $\mathcal{B}_j \in \mathfrak{B}$ is essentially equal to the running time of $\mathcal{A}$, and by Lemma 1 we have $\epsilon_{\mathcal{B}} \geq \epsilon_{\mathcal{A}}$. □

Proof of Lemma 4. For a given Naor-Reingold key $(a_1, \ldots, a_{2(\lambda-1)}) \in (\mathbb{Z}_q^*)^{2(\lambda-1)}$, let us define $k_\ell \in (\mathbb{Z}_q^*)^{2^\ell}$ and $b_{i,j} \in (\mathbb{Z}_q^*)$ as

$$\begin{aligned} k_1 &:= (b_{1,1}, b_{1,2}) := (a_1, a_2) \\ k_2 &:= (b_{2,1}, b_{1,4}) := (a_3, \ldots, a_6) \\ k_3 &:= (b_{3,1}, b_{3,8}) := (a_7, \ldots, a_{14}) \\ &\vdots \end{aligned}$$

Then we can write $\bigotimes_{\ell=1}^{\log\lambda} F^{2^\ell}_{\mathsf{NR}}(k_\ell, H_\ell(x))$ equivalently as

$$\begin{aligned} \bigotimes_{\ell=1}^{\log\lambda} F^{2^\ell}_{\mathsf{NR}}(k_\ell, H_\ell(x)) &= \left[\prod_{\ell=1}^{\log\lambda}\prod_{i=1}^{2^j} b_{\ell,i}^{H_\ell(x)_i}\right] = \left[\prod_{i=1}^{2^j} b_{j,i}^{H_j(x)_i}\right]^{\prod_{\ell\neq j}\prod_{i=1}^{2^j} b_{\ell,i}^{H_\ell(x)_i}} \\ &= (F_{\mathsf{NR}}(k_j, H_j(x)))^{\prod_{\ell\neq j}\prod_{i=1}^{2^j} b_{\ell,i}^{H_\ell(x)_i}} \end{aligned} \tag{11}$$

where $H_\ell(x)_i$ denotes the i-th bit of $H(x)$. Clearly we can compute the last term (11) efficiently, given $(F_{\mathsf{NR}}(k_j, H_j(x)), x, (k_\ell)_{\ell \in \{1,\ldots,\log\lambda\}\setminus\{j\}})$. This yields Lemma 4.

Security of the Naor-Reingold PRF with input encoding By combining Lemma 2 with Theorem 6 we now obtain that $F^{\mathcal{E}}_{\mathsf{NR}}$ is tightly-secure under the 1-Linear assumption with security loss $O(\log^2 \lambda)$ in the black-box setting and $O(\log \lambda)$ in the non-black-box setting.

Theorem 7. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of $F^{\mathcal{E}}_{\mathsf{NR}}$ defined over group $\mathbb{G}$, such that $t_\mathcal{A}$ and $1/\epsilon_\mathcal{A}$ are bounded by a polynomial in λ. Then we can construct an adversary $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B})$-breaks the 1-Linear assumption in $\mathbb{G}$ with*

$$t_\mathcal{B} \approx t_\mathcal{A} \qquad \textit{and} \qquad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}}{O(\log^2 \lambda)}$$

PROOF. We construct the family of adversaries $\mathfrak{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_{\log\lambda}\}$ from Lemma 2, pick $j \xleftarrow{\$} \{1, \ldots, \log\lambda\}$ at random, and run algorithm $\mathcal{B}_j$. With probability $1/\log\lambda$ we choose j that satisfies (9). Thus, by Lemma 2 we obtain an algorithm $\mathcal{B}'$ that $(\epsilon'_\mathcal{B}, t'_\mathcal{B})$-breaks the security of the Naor-Reingold PRF with domain $2^j \in O(\log\lambda)$ with $t'_\mathcal{B} \approx t_\mathcal{A}$ and $\epsilon'_\mathcal{B} \geq \epsilon_\mathcal{A}/2$.

By Theorem 6 we furthermore know that from the above adversary $\mathcal{B}'$ we can construct an adversary $\mathcal{B}$ that $(\epsilon_\mathcal{B}, t_\mathcal{B})$-breaks the 1-Linear assumption in $\mathbb{G}$ with $t_\mathcal{B} \approx t'_\mathcal{B} \approx t_\mathcal{A}$ and

$$\epsilon_\mathcal{B} \geq \frac{\epsilon'_\mathcal{B}}{O(\log\lambda)} \geq \frac{\epsilon_\mathcal{A}}{O(\log^2\lambda)}$$

so the claim follows. □

Similarly, we obtain the following theorem for non-black-box reductions.

Theorem 8. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of $F^{\mathcal{E}}_{\mathsf{NR}}$ defined over group $\mathbb{G}$, such that $t_\mathcal{A}$ and $1/\epsilon_\mathcal{A}$ are bounded by a polynomial in λ. Provided that q and $\epsilon_\mathcal{A}$ are explicitly known, we can construct an adversary $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B})$-breaks 1-Linear assumption in $\mathbb{G}$ with*

$$t_\mathcal{B} \approx t_\mathcal{A} \qquad \textit{and} \qquad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}}{O(\log\lambda)}$$

The proof of Theorem 8 is identical to Theorem 7, except that again we can use the non-black-box setting to avoid guessing the bounds on work-factor of the adversary determined by j.

4.2 Tightly-secure PRFs from the k-Linear assumption

The Lewko-Waters PRF. Again let $\mathbb{G}$ be an abelian group of prime order q, let $\mathbb{G}^* := \mathbb{G}\setminus\{1\}$, and let $g \xleftarrow{\$} \mathbb{G}^*$ be random. Note that g is a generator of $\mathbb{G}$. Let $\beta \in \mathbb{N}$ and define

$\mathcal{K}^\beta := ((\mathbb{Z}_q^*)^{k\times k})^\beta$. The Lewko-Waters pseudorandom function [27], instantiated with domain $\{0,1\}^\beta$, is defined as

$$F_{\mathsf{LW}}^\beta : \mathcal{K}^\beta \times \{0,1\}^\beta \to (\mathbb{G}^k)^* \qquad F_{\mathsf{LW}}^\beta(k,x) := \pi_1\left(\left[\mathbf{a}^\top \prod_{i=1}^{\beta} \mathbf{A}_i^{x_i}\right]\right) \tag{12}$$

where $k = (\mathbf{a}, \mathbf{A}_1, \ldots, \mathbf{A}_\beta) \in \mathbb{Z}_q^k \times (\mathbb{Z}_q^{k\times k})\beta$, $x = (x_1, \ldots, x_\beta) \in \{0,1\}^\beta$, and $\pi_1 : \mathbb{G}^k \to \mathbb{G}$ is the projection that, on input a vector $\mathbf{h} = (h_1, \ldots, h_k)^\top \in \mathbb{G}^k$, outputs h_1. The following theorem is from [27].

Theorem 9 ([27]). *From each adversary $\mathcal{A}$ that $(\epsilon_\mathcal{A}, t_\mathcal{A}, q)$-breaks the security of F_{LW}^β with input space $\{0,1\}^\beta$, we can construct an adversary $\mathcal{B}$ that $(\epsilon_\mathcal{B}, t_\mathcal{B})$-breaks the k-Linear assumption in $\mathbb{G}$ with*

$$t_\mathcal{B} \approx t_\mathcal{A} \qquad \textit{and} \qquad \epsilon_\mathcal{B} \geq \frac{\epsilon_\mathcal{A}}{k\beta}$$

Note that k is a constant, so that the security loss is linear in the size of the input space. In particular, if $\beta \in O(\log \lambda)$, then the security loss is only logarithmic in λ.

Tightly-secure PRFs from the Lewko-Waters PRF. For $\ell \in \{1, \ldots, \log \lambda\}$ let $\mathcal{H}_\ell$ be a family of universal hash functions from some domain $\mathcal{D}$ to $\{0,1\}^{2^\ell}$. Define function F as

$$F(K,x) := \prod_{i=1}^{\ell} F_{\mathsf{LW}}^{2^\ell}(k_\ell, H_\ell(x))$$

where $k_\ell \in \mathbb{Z}_q^k \times (\mathbb{Z}_q^{k\times k})^\ell$ is a secret key for $F_{\mathsf{LW}}^{2^\ell}$, and $K := (k_1, H_1, \ldots, k_{\log\lambda}, H_{\log\lambda})$.

Security analysis. Note that F is identical to the generic construction from Section 3.3, where (6) is instantiated with $g_{2^\ell} := F_{\mathsf{LW}}^{2^\ell}$ and the $\oplus$-operation is replaced with multiplication in $\mathbb{G}$. Trivially, the properties from Remark 2 hold for this construction. Thus, by combining Theorem 9 with Theorems 4 and 5, we obtain the following results.

Theorem 10. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of F, such that $t_\mathcal{A}$ and $1/\epsilon_\mathcal{A}$ are bounded by a polynomial in λ. Then we can construct an adversary $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B})$-breaks the Decisional k-Linear assumption with $t_\mathcal{B} \approx t_\mathcal{A}$ and $\epsilon_\mathcal{B} \geq \epsilon_\mathcal{A}/O(\log^2 \lambda)$.*

Theorem 11. *Let $\mathcal{A}$ be an adversary that $(t_\mathcal{A}, \epsilon_\mathcal{A}, q)$-breaks the security of F, such that q and $1/\epsilon_\mathcal{A}$ are bounded by a polynomial in λ. Provided that q and $\epsilon_\mathcal{A}$ are explicitly known, we can construct an adversary $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B})$-breaks the Decisional k-Linear assumption with $t_\mathcal{B} \approx t_\mathcal{A}$ and $\epsilon_\mathcal{B} \geq \epsilon_\mathcal{A}/O(\log \lambda)$.*

References

1. Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In Pointcheval and Johansson [30], pages 572–590.

2. Christoph Bader, Dennis Hofheinz, Tibor Jager, Eike Kiltz, and Yong Li. Tightly-secure authenticated key exchange. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 629–658, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.
3. Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the impossibility of tight cryptographic reductions. Cryptology ePrint Archive, Report 2015/374, 2015. `http://eprint.iacr.org/`.
4. Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 602–619, Santa Barbara, CA, USA, August 20–24, 2006. Springer, Heidelberg, Germany.
5. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274, Bruges, Belgium, May 14–18, 2000. Springer, Heidelberg, Germany.
6. Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 194–211, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany.
7. Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424, Cologne, Germany, April 26–30, 2009. Springer, Heidelberg, Germany.
8. Daniel J. Bernstein. Proving tight security for Rabin-Williams signatures. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 70–87, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany.
9. Olivier Blazy, Saqib A. Kakvi, Eike Kiltz, and Jiaxin Pan. Tightly-secure signatures from chameleon hash functions. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 256–279, Gaithersburg, MD, USA, March 30 – April 1, 2015. Springer, Heidelberg, Germany.
10. Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (Hierarchical) identity-based encryption from affine message authentication. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.
11. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany.
12. Ran Canetti and Juan A. Garay, editors. *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.
13. Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.
14. Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. In Canetti and Garay [12], pages 435–460.
15. Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 272–287, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Heidelberg, Germany.
16. Nico Döttling and Dominique Schröder. Efficient pseudorandom functions via on-the-fly adaptation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 329–350, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.
17. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Canetti and Garay [12], pages 129–147.

18. Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly secure cca-secure encryption without pairings. Cryptology ePrint Archive, Report 2016/094, 2016. http://eprint.iacr.org/.
19. Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.
20. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. On the cryptographic applications of random functions. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 276–288, Santa Barbara, CA, USA, August 19–23, 1984. Springer, Heidelberg, Germany.
21. Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.
22. Dennis Hofheinz, Tibor Jager, and Edward Knapp. Waters signatures with optimal security reduction. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 66–83, Darmstadt, Germany, May 21–23, 2012. Springer, Heidelberg, Germany.
23. Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Heidelberg, Germany.
24. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 433–442, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press.
25. Saqib A. Kakvi and Eike Kiltz. Optimal security proofs for full domain hash, revisited. In Pointcheval and Johansson [30], pages 537–553.
26. Hugo Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 631–648, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Heidelberg, Germany.
27. Allison B. Lewko and Brent Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM CCS 09*, pages 112–120, Chicago, Illinois, USA, November 9–13, 2009. ACM Press.
28. Allison B. Lewko and Brent Waters. Why proving HIBE systems secure is difficult. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 58–76, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.
29. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.
30. David Pointcheval and Thomas Johansson, editors. *EUROCRYPT 2012*, volume 7237 of *LNCS*, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.
31. Sven Schäge. Tight proofs for signature schemes without random oracles. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 189–206, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany.
32. Hovav Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. http://eprint.iacr.org/.