Efficient Pseudorandom Functions via On-the-Fly Adaptation

Nico Döttling [⋆,⋆⋆1] and Dominique Schröder[⋆⋆⋆2]

[1] Dept. of Computer Science, Aarhus University
[2] Dept. of Computer Science, Saarland University, CISPA

Abstract. Pseudorandom functions (PRFs) are one of the most fundamental building blocks in cryptography with numerous applications such as message authentication codes and private key encryption. In this work, we propose a new framework to construct PRFs with the overall goal to build efficient PRFs from standard assumptions with an almost tight proof of security. The main idea of our framework is to start from a PRF for any small domain (i.e. poly-sized domain) and turn it into an ℓ-bounded pseudorandom function, i.e., into a PRF whose outputs are pseudorandom for the first ℓ distinct queries to F. In the second step, we apply a novel technique which we call *on-the-fly adaptation* that turns any bounded PRF into a fully-fledged (large domain) PRF. Both steps of our framework have a tight security reduction, meaning that any successful attacker can be turned into an efficient algorithm for the underlying hard computational problem without any significant increase in the running time or loss of success probability.

Instantiating our framework with specific number theoretic assumptions, we construct a PRF based on k-LIN (and thus DDH) that is faster than all known constructions, which reduces almost tightly to the underlying problem, and which has shorter keys. Instantiating our framework with general assumptions, we construct a PRF with very flat circuits whose security tightly reduces to the security of some small domain PRF.

Keywords: Pseudorandom Functions, Efficient Reductions, DDH, K-LIN, LWE

⋆ The authors acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part of this work was performed; and also from the CFEM research center (supported by the Danish Strategic Research Council) within which part of this work was performed.

⋆⋆ Supported by European Research Commission Starting Grant no. 279447.

⋆⋆⋆ Supported by the German Federal Ministry of Education and Research (BMBF) through funding for the Center for IT-Security, Privacy and Accountability (CISPA, `www.cispa-security.org`) and also by an Intel Early Career Faculty Honor Program Award.

1 Introduction

Goldreich, Goldwasser, and Micali (GGM) introduced pseudorandom functions (PRFs) in 1984 [13]. Roughly, a PRF is a keyed deterministic function, whose output is indistinguishable from a random function. PRFs are one of the most fundamental building blocks in cryptography, with numerous applications such as private-key encryption, message authentication codes, key derivation, and many more, e.g., [2,14,22,29,32]. In this work, we propose a novel framework to construct PRFs with the overall goal of constructing *efficient* PRFs based on *standard assumptions* with an almost *tight* proof of security. The basic idea of this framework is to transform a PRF for a small domain (i.e., poly-size) into a fully fledged domain that handles large input spaces. This transformation tightly reduces to the underlying small domain PRF. The main steps of our framework and the novel techniques are shown in Figure 1. We begin with a PRF that works

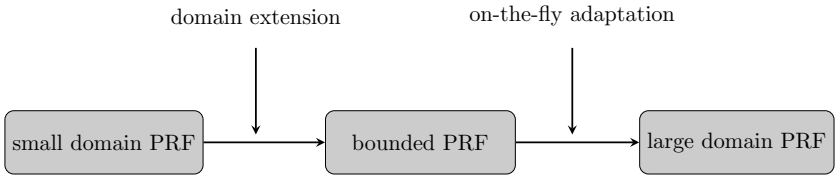

Fig. 1. Overview of the main steps and the techniques.

over a small domain, say $\{0,1\}^{\log \ell}$, and which can be evaluated very efficiently in time $\mathsf{poly}(\lambda, \log \ell)$, for some parameter ℓ.

The first step in our framework is to extend the domain of a small-domain PRF into a *bounded pseudorandom function* (bPRF). A function F is an ℓ-bounded pseudorandom function (for an $\ell \leq \mathsf{poly}(\lambda)$), if the outputs of F are pseudorandom for the first ℓ distinct queries to F and if F can be computed "super efficiently" (i.e., in time $\mathsf{poly}(\lambda, \log(\ell))$). In some sense, this primitive can be seen as the computational analogue to ℓ-wise independent functions.

The second step in our framework is a reduction technique we call *on-the-fly adaptation*. The goal of this technique is to construct a PRF F in which we can dynamically embed an ℓ-bounded PRF F_ℓ for every ℓ that grows at most polynomially. Now assume we have a PPT distinguisher $\mathcal{D}$ that distinguishes F from a truly random function. Since $\mathcal{D}$ is efficient, it sends at most $q = \mathsf{poly}(\lambda)$ queries to its oracle (for an a-priori unknown q). On-the-fly adaptation allows us to turn this distinguisher against F into a distinguisher $\mathcal{D}'$ against a bounded PRF F_q that has the same advantage.

We will demonstrate this idea with a simple on-the-fly adaptation technique that works for any bounded PRF. The basic idea of this technique is to compute F as a sum of functions F_ℓ, for an exponentially increasing ℓ. An important

point is that all F_ℓ have the *same domain*. The function F is computed by

$$F(K,x) = \bigoplus_{i=0}^{t} F_{2^i}(K_{2^i}, x),$$

where $K = (K_{2^i})_{i=1,\ldots,t}$. If we choose the parameter $t = \omega(\log(\lambda))$ slightly super-logarithmic, we will be able to embed any F_ℓ into F. Notice that F can be computed efficiently, as we required that bounded PRFs can be computed in time $\mathsf{poly}(\lambda, \log(\ell))$. To illustrate the main idea, assume that there exists a distinguisher $\mathcal{D}$ that makes at most $q = \mathsf{poly}(\lambda)$ queries distinguishes F from a truly random function. We will provide a reduction that turns this distinguisher into a distinguisher against the small domain PRF $F_{2^{\lceil \log q \rceil}}$. Observe that we can express F by

$$F(K,x) = \bigoplus_{i=0}^{\log q - 1} F_{2^i}(K_{2^i}, x) \oplus F_{2^{\lceil \log q \rceil}}(K_{2^{\lceil \log q \rceil}}, x) \oplus \bigoplus_{i=\log q + 1}^{t} F_{2^i}(K_{2^i}, x).$$

The reduction can now replace the middle term $F_{2^{\lceil \log q \rceil}}(K_{2^{\lceil \log q \rceil}}, x)$ by its own oracle and provide to $\mathcal{D}$ an oracle $\mathcal{O}'$ that computes the function

$$\mathcal{O}'(x) = \bigoplus_{i=0}^{\log q - 1} F_{2^i}(K_{2^i}, x) \oplus \mathcal{O}(x) \oplus \bigoplus_{i=\log q + 1}^{t} F_{2^i}(K_{2^i}, x).$$

Clearly, if $\mathcal{O}$ computes the function $F_{2^{\lceil \log q \rceil}}$, then $\mathcal{O}'$ computes the function F. On the other hand, if $\mathcal{O}$ is a random function, then $\mathcal{O}'$ also is a random function. This reduction is tight. Notice that it is crucial for this technique to work that all F_ℓ have the same input domain. Basically the domain extension step in our framework is geared towards *equalizing* the domains of small domain PRFs. This generic technique is similar to a transformation from non-adaptive to adaptive pseudorandom functions by Berman and Haitner [4]. The construction of [4] however yields no tight security proof (which was not the purpose of that work), as their construction does not start from bounded PRFs.

We will now discuss domain extension for arbitrary PRFs and provide a simple domain extension technique that uses only linear functions to pre- and post-process a small domain PRF. This, together with the generic on-the-fly adaptation technique described above yields a PRF construction from any small domain PRF. We will discuss an instantiation of this general construction based on LWE.

Domain Extension for Arbitrary PRFs The problem of domain extension for pseudorandom functions was first considered by Levin [20]. Levin showed that if the domain of a certain PRF is already sufficiently large, then it can be extended by using a universal hash function to hash larger inputs into the domain of this PRF. However, this technique is vulnerable to a "birthday attack", which means that after a certain number of queries there is a high probability of finding

a collision in the hash function. Levin's technique also fails for small domain PRFs, i.e., PRFs with domains of polynomial size. Jain, Pietrzak, and Tentes [19] provided a domain extension technique which also works for small domains, but has an unfavorable security loss in this case. Moreover, as mentioned by the authors, their technique does not seem to be directly applicable to efficient PRF such as the one's based on DDH [19]. The work of Jain et al. [19] was refined by Chandran and Garg [8]. Berman et al. [5] also showed how to bypass the birthday barrier via Cuckoo hashing.

We provide a simple general domain extension technique that preserves the parallel complexity of an underlying small domain PRF. This domain extension technique is inspired by the construction of universal hash functions by Ishai et al. [18] and can be seen as an amplified version of Levin's trick. For a small domain pseudorandom function $\mathsf{PRF}_\ell : \{0,1\}^{\log(2\lambda\ell)} \to \mathcal{Y}$, we construct a large domain bounded PRF $F_\ell : \mathcal{X} \to \mathcal{Y}$ by

$$F_\ell(K', x) = \bigoplus_{j=1}^{\lambda} \mathsf{PRF}_\ell(K, \mathsf{BIN}(j) \| H_j(x)),$$

where $K' = (K, H_1, \ldots, H_\lambda)$, $H_1, \ldots, H_\lambda \leftarrow_\$ \mathcal{H}$ are randomly chosen universal hash functions from a family $\mathcal{H}$ that maps $\mathcal{X}$ to $\{0,1\}^{\log(2\ell)}$ and $\mathsf{BIN}(j)$ is the $\log(\lambda)$ bit binary representation of an integer $j \in \{1, \ldots, \lambda\}$.

1.1 A General Transformation

Above we described our on-the-fly adaptation technique that works for any bounded PRF. Combining this technique with a general domain extension technique, we obtain large domain pseudorandom function with almost tight security (i.e., only a logarithmic loss) from any suitable small domain PRF. In a nutshell, a small domain PRF is suitable for this technique if its security loss only depends on the size of its input domain, but not (polynomially) on the number queries a distinguisher sends[3]. The computational problems from which PRFs with such a small security loss can be constructed usually have one feature in common: they support a statistical random self-reduction. Candidate PRFs with this property are PRFs based on the LWE [30,28] problem, such as the PRF of Banerjee, Peikert, and Rosen [1]. Using the BPR PRF as small domain PRF in our general construction, we obtain a large domain PRF which is secure under a weaker assumption, which has a tighter proof of security, and a shallower evaluation circuit than instantiating the BPR scheme with a large domain directly.

In the remaining part of this section, we discuss more efficient instantiations based on DDH and k-LIN. Here, we exploit specific number theoretic properties in order to improve the efficiency and security of the resulting PRF.

[3] The Naor Reingold PRF would be such a suitable PRF as its security reduction only loses a factor of n. However, as discussed above we provide a much more efficient direct construction based on the NR PRF.

1.2 Efficient PRFs based on DDH and k-LIN

One appealing property of our framework is that it yields several new constructions of PRFs based on weak standard assumptions, such as k-LIN (and thus DDH) with an almost tight proof of security. A tight reduction means that a successful attacker can be turned into an efficient algorithm for the hard computational problem without any significant increase in the running time or significant loss of success probability[4]. We will provide a specific on-the-fly adaptation technique that exploits algebraic properties of the underlying number theoretic assumptions. We can thus avoid the blow up of the general on-the-fly adaptation technique described in the last paragraph and obtain PRFs that improve upon known constructions in terms of efficiency, security, and key-size.

Instantiation based on DDH. In the following we discuss our construction based on the DDH assumption. Our underlying small domain PRF is the Naor-Reingold PRF based on DDH [26]. For an input domain $\{0,1\}^n$, the Naor Reingold PRF $\mathsf{NR} : \mathcal{K}_n \times \{0,1\}^n \to \mathbb{G}$ is defined by

$$\mathsf{NR}_n(K, x) = g^{a \prod_{j=0}^{n-1} s_j^{x_j}},$$

where $K = (a, s_0, \ldots, s_{n-1})$ and $a, s_0, \ldots, s_{n-1} \leftarrow_\$ \mathbb{Z}_p$. In the first step, we turn the small domain PRF $\mathsf{NR}_{\log(\ell)}$, which has a domain of size ℓ into an ℓ-bounded PRF that has input domain $\mathbb{Z}_p$, i.e., a large input domain. In contrast to our generic construction (that we will discuss later), we can exploit specific number theoretic properties in order to improve the efficiency, the tightness of the security reduction, and the key-size. The bounded PRF $F_\ell : \mathcal{K} \times \mathbb{Z}_p \to \mathbb{G}$ is defined as follows:

$$F_\ell(K, x) = g^{a \prod_{j=0}^{\log(\ell)-1} (s_j + x^{2^j})}. \tag{1}$$

We will briefly discuss why security of this PRF tightly reduces to the security of $\mathsf{NR}_{\log(\ell)}$. Expanding the exponent of $F_\ell(K, x)$ yields

$$a \prod_{j=0}^{\log(\ell)-1} (s_j + x^{2^j}) = \sum_{\mathbf{c} \in \{0,1\}^{\log(\ell)}} \underbrace{\left(a \prod_{j=0}^{\log(\ell)-1} s_j^{1-c_j} \right)}_{E(\mathbf{c})} x^{\sum_{j=1}^{t} c_j 2^j} \tag{2}$$

Now, observe that the term $E(\mathbf{c})$ on the right side is an exponent of the Naor Reingold PRF $\mathsf{NR}_{\log(\ell)}$. Specifically, it holds that $g^{E(\mathbf{c})} = \mathsf{NR}_{\log(\ell)}(\neg\mathbf{c})$ (where $\neg\mathbf{c}$ is the bitwise negation of $\mathbf{c}$). Changing the sum on the right side of (2) to run over all $j = 0, \ldots, 2^{\lceil \ell \rceil} - 1$ and setting $\mathbf{c} = \mathsf{BIN}(j)$ (where $\mathsf{BIN}(j)$ is the $\log(\ell)$-bit

[4] Usually even a polynomially-bounded increase/loss is considered as significant, if the polynomial may be large. An increase/loss by a small constant factor is not considered as significant.

binary representation of j), we get that F_ℓ can be equivalently computed as

$$F_\ell(K,x) = \prod_{j=0}^{2^{\lceil \log(\ell) \rceil}-1} \left(\mathsf{NR}_{\log(\ell)}(K, \neg \mathsf{BIN}(j))\right)^{x^j}. \tag{3}$$

Notice that this expression can still be efficiently computed as long as $\ell \leq \mathsf{poly}(\lambda)$. Now, observe that if we replace $\mathsf{NR}_{\log(\ell)}$ by a random function in this expression, then F_ℓ becomes an (information theoretic) ℓ-wise independent function. We can therefore use this alternative description of F_ℓ to show that it is an ℓ-bounded PRF.

The observation that functions of the form as F_ℓ in (3) can be computed in time $\log \ell$ via a closed form as (1) was previously made by Benadbbas, Gennaro, and Vahlis for the Naor-Reingold PRF [3] and Fiore and Gennaro for the Lewko-Waters PRF [12]. The fact that F_ℓ is a bounded PRF was independently observed by Hazay [15].

In the last step, we apply an "in-place" on-the-fly adaptation technique to this function. We will not use the generic technique described above, but one that exploits the specific algebraic properties of F_ℓ. We define the full fledged PRF F by

$$\mathsf{F}(K,x) = g^{a \prod_{j=0}^{t-1}(s_j + x^{2^j})},$$

where the parameter $t = \omega(\log(\lambda))$ is chosen slightly super-logarithmic. Now, notice that we can embed the bounded PRF F_ℓ (for any $\ell \leq \mathsf{poly}(\lambda)$) into F by

$$\mathsf{F}(K,x) = \left(g^{a \prod_{j=0}^{log(\ell)-1}(s_j + x^{2^j})}\right)^{\prod_{j=\log(\ell)}^{t-1}(s_j + x^{2^j})} = (F_\ell(K_\ell, x))^{\prod_{j=\log(\ell)}^{t-1}(s_j + x^{2^j})}.$$

In the security proof, we replace F_ℓ by a truly random function. The main part of the proof consists in showing that the exponent $\prod_{j=\log(\ell)}^{t-1}(s_j + x^{2^j})$ only accounts for a negligible error.

Comparison to Naor-Reingold [25]. Our full fledged PRF with input domain $\mathbb{Z}_p$ improves upon the Naor-Reingold PRF (NR-PRF) in terms of tightness of the security reduction and compactness. In contrast to the NR-PRF, the loss of our security reduction is only a factor of $\log(q)$ (where $q = \mathsf{poly}(\lambda)$ is the number of queries required by the distinguisher $\mathcal{D}$), compared to a factor n for the NR-PRF. Our PRF is very compact as it only requires $\omega(\log(\lambda))$ $\mathbb{Z}_p$ elements for its key, whereas the Naor-Reingold needs n $\mathbb{Z}_p$ elements. Since the exponentiation is the dominating factor in the computation of both PRFs, the costs to evaluate both functions is roughly the same.

Instantiation based on k-LIN. In the main body, we directly provide a PRF construction based on a family of weaker computational problems known as k-LIN [31,17]. The decisional k-linear assumption becomes (generically) weaker as the parameter k grows, where the instance $k = 1$ corresponds to DDH and

$k = 2$ to the linear assumption [6]. The main motivation for these assumptions is that groups are known where the DDH assumption is easy, but the computational Diffie Hellman problem is supposedly hard [16]. It is thus desirable to have constructions of cryptographic primitives based on the decisional k-linear assumption instead of DDH. Our generalized PRF is defined as follows: Let $k \geq 1$ be a positive integer, $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order p and $t = \omega(\log(\lambda))$. The function $F : \mathcal{K} \times \mathbb{Z}_p \to \mathbb{G}^k$ is defined by

$$F(K, x) = g^{\mathbf{a}^\top \cdot \prod_{j=0}^{t-1} (\mathbf{S}_j + x^{2^j} \cdot \mathbf{I})},$$

where $K = (\mathbf{a}, \mathbf{S}_0, \ldots, \mathbf{S}_{t-1})$ with $\mathbf{a} \leftarrow_\$ \mathbb{Z}_p^k$, $\mathbf{S}_0, \ldots, \mathbf{S}_{t-1} \leftarrow_\$ \mathbb{Z}_p^{k \times k}$ and $\mathbf{I}$ the identity matrix. Clearly, if we only need a single group element as output, we can truncate the exponent and perform only 1 exponentiation.

Comparison to Lewko-Waters [21]. Our PRF improves upon the Lewko-Waters PRF (LW-PRF) in terms of efficiency, tightness of the security reduction, and compactness. A single evaluation of the LW-PRF involves n matrix multiplications and a single exponentiation. In our case, the computation requires only $t = \omega(\log(\lambda))$ matrix multiplications and a single exponentiation. For larger k, the cost of the matrix multiplication dominates the cost of the exponentiation, so in this case our construction is more efficient. The security reduction of Lewko and Waters loses a factor of $k \cdot n$ while our reduction only losses a factor of $k \log q$. The keys of the LW-PRF consist of n $k \times k$ matrices over $\mathbb{Z}_p$, while ours consists merely of $t = \omega(\log \lambda)$ such matrices.

1.3 Other Related Work

Many number-theoretic PRF constructions follow the GGM paradigm [13], such as [25,21,1]. Naor and Reingold introduced pseudorandom synthesizer (PRS) that can be used to construct parallel computable pseudorandom function [24,1]. A PRF construction that is not based on either the GGM or synthesizers paradigm is the PRF of Dodis-Yampolskiy, which is in fact a direct construction, but whose security is closely related to its underlying bilinear q-type assumption [10]. Recently, Chase and Meiklejohn showed that this q-type assumption can be reduced to the subgroup hiding assumption in composite order groups [9]. The PRF of Naor, Reingold, and Rosen is a clever variant of the Naor-Reingold PRF that is secure under the factoring assumption [27]. The work of Boneh, Montgomery, and Raghunathan combines a generalization of the GGM tree with the Dodis-Yampolskiy PRF to get a large-domain (simulateable) verifiable random function [7].

2 Preliminaries

Throughout this paper, we will use λ to denote the security parameter. We will denote the concatenation of two bit strings x and y by $x \| y$. We will generally

assume that logarithms are rounded to the next biggest integer, i.e., when we write $\log(\ell)$ we actually mean $\lceil \log(\ell) \rceil$. To avoid confusion, we will sometimes still write $\lceil \log(\ell) \rceil$, e.g. when we write $2^{\lceil \log(\ell) \rceil}$ to indicate that this can be different from ℓ.

Definition 1 (Pseudorandom Functions). *Let $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ be two finite sets depending on λ. We say that an efficiently computable keyed function $\mathsf{PRF} : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ with key-space $\mathcal{K}_\lambda$ is a pseudorandom function (PRF), if it holds for every PPT oracle distinguisher $\mathcal{D}$ that*

$$|\Pr[\mathcal{D}^{\mathsf{PRF}(K,\cdot)}(1^\lambda) = 1] - Pr[\mathcal{D}^R(1^\lambda) = 1]| \leq \mathsf{negl}(\lambda),$$

where $K \leftarrow_\$ \mathcal{K}_\lambda$ and $R : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ is a randomly chosen function. If $|\mathcal{X}| \leq \mathsf{poly}(\lambda)$, then we say that PRF is a small-domain PRF, otherwise we call PRF a large-domain PRF.

We will usually omit the λ subscript in the definition of $\mathcal{K}$, $\mathcal{X}$ and $\mathcal{Y}$. Moreover, we will henceforth implicitly assume that distinguisher gets 1^λ as an additional input without explicitly stating this.

As mentioned in the outline, bounded pseudorandom functions can be seen as a computational analogue of limited-wise independent functions. Basically, the difference between true PRFs and bounded PRFs manifests itself in their security guarantee. While a distinguisher against a true PRF can query the PRF an a-priori unbounded number of times, a distinguisher against an ℓ-bounded PRF can query the PRF with at most ℓ distinct queries.

Definition 2 (Bounded Pseudorandom Functions). *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets (depending on λ). We say that a keyed function $F_\ell : \mathcal{K}_\ell \times \mathcal{X} \to \mathcal{Y}$ parametrized by a parameter ℓ is a bounded pseudorandom function (bPRF), if F_ℓ is computable in time $\mathsf{poly}(\lambda, \log(\ell))$ and if it holds for all efficiently computable $\ell^* = \ell(\lambda) \leq \mathsf{poly}(\lambda)$ and all ℓ^*-query distinguishers $\mathcal{D}$ (i.e. distinguishers that send at most ℓ^* distinct queries) that*

$$|\Pr[\mathcal{D}^{F_\ell(K,\cdot)} = 1] - Pr[\mathcal{D}^R = 1]| \leq \mathsf{negl}(\lambda),$$

where $K \leftarrow_\$ \mathcal{K}_\ell$ and $R : \mathcal{X} \to \mathcal{Y}$ is a randomly chosen function.

Notice that in the definition of bounded PRFs we allow the key-space to depend on ℓ, but $\mathcal{X}$ and $\mathcal{Y}$ are independent of ℓ. Moreover, as we require that F_ℓ is computable in time $\mathsf{poly}(\lambda, \log(\ell))$, we implicitly also require that $|\mathcal{K}_\ell| \leq \mathsf{poly}(\lambda, \log(\ell))$. Requiring that F_ℓ can be computed in time $\mathsf{poly}(\lambda, \log(\ell))$ allows us to evaluate F_ℓ for super-polynomial ℓ, while we only require security for ℓ^* which are at most polynomial.

The following lemma states that if a function F outputs uniformly random outputs under *benign* inputs, then the statistical distance from F to a uniformly random function F' can be bounded by the probability that a non-adaptive sequence of inputs is not benign. Intuitively, an adaptive distinguisher $\mathcal{D}$ learns nothing about the set of bad inputs unless it finds such an input by chance, as

otherwise the function F reveals no information about the set of bad inputs. This lemma is a simplified version of a more general statement due to Maurer [23].

Lemma 1. *Let $\mathcal{X}$ and $\mathcal{Z}$ be two finite sets. Let $F_{K,\mathsf{aux}} : \mathcal{X} \to \mathcal{Z}$ be a function that takes two additional parameters $K \in \mathcal{K}$ and $\mathsf{aux} \in \mathsf{AUX}$. Let $\mathsf{good}(\cdot,\cdot)$ be a predicate with the following property: If $\mathsf{good}(\{x_1,\ldots,x_i\},\mathsf{aux})$ holds, then $F_{K,\mathsf{aux}}(x_1),\ldots,F_{K,\mathsf{aux}}(x_i)$ are distributed uniformly at random over the choice of $K \leftarrow_\$ \mathcal{K}$. Let $\mathcal{D}$ be a (possibly unbounded) distinguisher that makes at most ℓ distinct queries, $K \leftarrow_\$ \mathcal{K}$, $\mathsf{aux} \leftarrow_\$ \mathsf{AUX}$ and let F' be a uniformly chosen function from $\mathcal{X}$ to $\mathcal{Z}$. Then it holds that*

$$|\Pr[\mathcal{D}^{F_{K,\mathsf{aux}}} = 1] - \Pr[\mathcal{D}^{F'} = 1]| \leq \max_S \Pr[\neg\mathsf{good}(S,\mathsf{aux})],$$

where S runs over all subsets of $\mathcal{X}$ of size at most ℓ.

A proof of Lemma 1 can be found in the full version.

3 A Generic Construction

In this section, we will first provide an efficient construction of ℓ-bounded pseudorandom function any small domain PRF with input space of (polynomial) size $n \cdot \ell$. Security of the ℓ-bounded PRF follows tightly from the underlying small domain PRF. Second, we will provide a general construction of a PRF from ℓ-bounded PRFs, where security also follows tightly.

3.1 Bounded PRFs via Domain Extension of Small Domain PRFs

We will need universal hash functions for our domain extension technique.

Definition 3 (Universal Hash Functions). *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. We say that a family $\mathcal{H}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$ is a family of universal hash functions, if it holds for all $x \neq x' \in \mathcal{X}$ that $\Pr[H(x) = H(x')] \leq 1/|\mathcal{Y}|$, where the probability is taken over the random choice of $H \leftarrow_\$ \mathcal{H}$.*

Universal hash functions can be constructed very efficiently, see e.g.,[18].

Construction 1 *Let $\mathsf{PRF}_\ell : \mathcal{K}_\ell \times \{0,1\}^{\log(2\lambda\ell)} \to \{0,1\}^m$ be a keyed function with key space $\mathcal{K}_\ell$. Let $\mathcal{H}_\ell$ be a family of universal hash functions that map $\mathcal{X}$ to $\{0,1\}^{\log(2\ell)}$. Let $\mathsf{BIN}(j)$ denote the $\log(\lambda)$ bit binary representation of a number $j \in \{1,\ldots,\lambda\}$. We define the keyed function $F_\ell : \mathcal{K}' \times \mathcal{X} \to \{0,1\}^m$ with key space $\mathcal{K}'_\ell = \mathcal{H}^\lambda \times \mathcal{K}_\ell$ by*

$$F_\ell(K',x) = \bigoplus_{j=1}^{\lambda} \mathsf{PRF}_\ell(K, \mathsf{BIN}(j) \| H_j(x)),$$

where $H_j \leftarrow_\$ \mathcal{H}_\ell$ for $j = 1,\ldots,\lambda$, $K \leftarrow_\$ \mathcal{K}_\ell$ and $K' = (H_1,\ldots,H_\lambda,K)$.

The following theorem states that F_ℓ is an ℓ-bounded pseudorandom function if PRF_ℓ is a pseudorandom function.

Theorem 1. *Let PRF_ℓ and F_ℓ be as in Construction 1. If PRF_ℓ is a pseudorandom function, then F_ℓ is an ℓ-bounded pseudorandom function. More specifically, assume there exists an $\ell^* \leq \mathsf{poly}(\lambda)$ and an ℓ^*-query PPT distinguisher $\mathcal{D}$ that distinguishes F_{ℓ^*} from a truly random function with advantage ϵ, then there exists a PPT distinguisher $\mathcal{D}'$ with essentially the same runtime as $\mathcal{D}$ that distinguishes PRF_{ℓ^*} from a truly random function with advantage at least $\epsilon - \ell^* \cdot 2^{-\lambda}$.*

The proof of Theorem 1 will be given in the full version.

3.2 PRFs via On-the-Fly Adaptation of bounded PRFs

In this section we provide a generic on-the-fly adaptation technique which converts a bounded PRF into a standard PRF.

Construction 2 *Let $t = \omega(\log(\lambda))$ be slightly super-logarithmic. For a given parameter ℓ, let $F_\ell : \mathcal{K}_\ell \times \mathcal{X} \to \{0,1\}^m$ be a keyed function with corresponding key space $\mathcal{K}_\ell$. Define the function $F : \mathcal{K} \times \mathcal{X} \to \{0,1\}^m$ with key-space $\mathcal{K} = \prod_{i=0}^{t} \mathcal{K}_{2^i}$ by*

$$F(K, x) = \bigoplus_{i=0}^{t} F_{2^i}(K_{2^i}, x),$$

where $K_{2^i} \leftarrow_\$ \mathcal{K}_{2^i}$ for $i = 1, \ldots, t$ and $K = (K_{2^i})_{i=1,\ldots,t}$.

We will now show that F is in fact a pseudorandom function.

Theorem 2. *Let F_ℓ and F be as in Construction 2. Assume that F_ℓ is an ℓ-bounded PRF for every efficiently computable $\ell = \ell(\lambda)$. Then F is a pseudorandom function. Specifically, if $\mathcal{D}$ is a PPT distinguisher against F with advantage ϵ that makes at most $q = \mathsf{poly}(\lambda)$ distinct queries, then there exists a PPT distinguisher $\mathcal{D}'$ (with essentially the same runtime as $\mathcal{D}$) with advantage ϵ against F_{ℓ^*}, where $\ell^* = 2^{\lceil \log(q) \rceil} \leq 2q = \mathsf{poly}(\lambda)$.*

Proof. Let $\mathcal{D}$ be a PPT distinguisher against F with advantage ϵ that makes at most q distinct queries. Note that since $q = \mathsf{poly}(\lambda)$ and $t = \omega(\log(\lambda))$, it holds $\log(q) \leq t$ (for a sufficiently large λ). We will now construct an ℓ^*-query distinguisher $\mathcal{D}'$ against F_{ℓ^*}, which is given in Figure 2.

Notice first that $\mathcal{D}'$ sends at most $q \leq 2^{\lceil \log(q) \rceil} = \ell^*$ queries to its oracle, as $\mathcal{D}$ sends at most q oracle queries. We will now analyze the distinguishing advantage of $\mathcal{D}'$. First, assume that $\mathcal{D}'$'s oracle $\mathcal{O}$ implements the function $F_{\ell^*}(K, \cdot)$ for a randomly chosen $K \leftarrow_\$ \mathcal{K}_{\ell^*}$. Then, the oracle $\mathcal{O}$ provided by $\mathcal{D}'$ to $\mathcal{D}$ implements exactly the function $F(K, \cdot)$ for a randomly chosen $K \leftarrow_\$ \mathcal{K}$. On the other hand, if $\mathcal{O}$ behaves like a uniformly random function R', then the oracle $\mathcal{O}'$ also

Distinguisher $\mathcal{D}'$
Has access to oracle $\mathcal{O}(\cdot)$
Set $i^* \leftarrow \lceil \log(q) \rceil$
For $i \in \{1, \ldots, t\} \backslash \{i^*\}$
$K_{2^i} \leftarrow_\$ \mathcal{K}_{2^i}$
$b' \leftarrow \mathcal{D}^{\mathcal{O}'(\cdot)}(1^\lambda)$
return b'

$\mathcal{O}'(x)$:
$y \leftarrow \bigoplus_{i=0}^{i^*-1} F_{2^i}(K_{2^i}, x) \oplus \mathcal{O}(x) \oplus \bigoplus_{i=i^*+1}^{t} F_{2^i}(K_{2^i}, x)$
return y

Fig. 2. The distinguisher $\mathcal{D}'$

implements a uniformly random function R, as R' is independent of the K_{2^i}. Consequently, it holds that

$$\begin{aligned}\mathsf{Adv}(\mathcal{D}') &= |\Pr[{\mathcal{D}'}^{F_{\ell^*}(K_{\ell^*}, \cdot)} = 1] - \Pr[{\mathcal{D}'}^{R'} = 1]| \\ &= |\Pr[\mathcal{D}^F = 1] - \Pr[\mathcal{D}^R = 1]| = \epsilon,\end{aligned}$$

i.e. $\mathcal{D}'$ distinguishes F_ℓ from a uniformly random function R' with advantage ϵ. This concludes the proof.

3.3 Instantiations

Combining Theorem 1 and Theorem 2 yields the following

Theorem 3. *Let $t = \omega(\log(\lambda))$. Let $\mathsf{PRF}_\ell : \{0,1\}^{\log(2\lambda\ell)} \to \{0,1\}^m$ be a small domain PRF, let $\mathcal{H}_\ell : \mathcal{X} \to \{0,1\}^{\log(2\ell)}$ be a family of universal hash functions. Define the keyed function $F : \mathcal{K} \times \mathcal{X} \to \{0,1\}^m$ by*

$$F(K, x) = \bigoplus_{i=1}^{t} \bigoplus_{j=1}^{\lambda} \mathsf{PRF}_{2^i}(K_{2^i}, \mathsf{BIN}(j) \| H_{2^i,j}(x)),$$

where $K_{2^i} \leftarrow_\$ \mathcal{K}_{2^i}$ for $i = 1, \ldots, t$ and $H_{2^i,j} \leftarrow_\$ \mathcal{H}_{2^i}$ for $i = 1, \ldots, t$ and $j = 1, \ldots, \lambda$.

If PRF_ℓ is a PRF for every $\ell = \mathsf{poly}(\lambda)$, then F is a PRF. More specifically, if there exists an distinguisher $\mathcal{D}$ that makes at most $q = \mathsf{poly}(\lambda)$ queries and distinguishes F with advantage ϵ, then there exists a distinguisher $\mathcal{D}'$ with essentially the same runtime as $\mathcal{D}$ that distinguishes $\mathsf{PRF}_{2^{\lceil q \rceil}}$ with advantage $\epsilon - q \cdot 2^{-\lambda}$.

We will briefly discuss efficiency aspects of the construction provided in Theorem 3. First of all notice that the transformation preserves the parallel complexity of the underlying small domain PRF. Moreover, the pre- and post-processing steps are entirely linear, i.e. the computation of universal hash functions and XOR-ing the results.

We will now discuss an instantiation of this PRF using a small domain PRF based on lattice problems. As already mentioned in the introduction, the main purpose of our constructions is obtaining PRFs from standard assumptions that are as tight as possible. Since the construction in the last section allows reducing the security of the constructed large domain PRF to the security of an *adversary specific* small domain PRF, we need a family of small domain PRFs with security as tight as possible. The Naor-Reingold PRF with domain $\{0,1\}^n$ allows for a security loss of a factor of n, while the security loss of a comparable GGM PRF is $q \cdot n$. This holds because the DDH problem possesses a *statistical random self-reduction* which allows to compute an arbitrary number of DDH samples from a given sample. The learning with errors (LWE) problem enjoys a similar property, which is stated explicitly in the assumption.

Definition 4 (Decisional LWE [30,28]). *Let $p = p(\lambda)$ be a modulus, $k = k(\lambda) = \mathsf{poly}(\lambda)$ be a positive integer and $\chi_r = D_{\mathbf{Z},r}$ be a gaussian distribution with noise parameter r. Let $\mathbf{s} \leftarrow_\$ \mathbb{Z}_p^k$ be chosen uniformly at random. The goal of the $\mathsf{LWE}(p, n, \chi_r)$ problem is to distinguish an arbitrary number of samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ where $\mathbf{a} \leftarrow_\$ \mathbb{Z}_p^k$ and $e \leftarrow_\$ \chi_\alpha$ from samples $(\mathbf{a}, u)$ where $u \leftarrow_\$ \mathbb{Z}_p$ is chosen uniformly at random.*

Banerjee, Peikert and Rosen [1] constructed a PRF based on the LWE problem. The PRF has a structure which is similar to the Lewko-Waters PRF but uses a rounding operation instead of exponentiation. Let $p_1 \gg p_2$. For an $x \in \mathbb{Z}_{p_1}$ define $\lfloor x \rceil_{p_2} = \lceil (p_2/p_1) \cdot x \rfloor \mod p_2$. For vectors $\mathbf{x} \in \mathbb{Z}_{p_1}^k$ define $\lfloor \cdot \rceil_{p_2}$ componentwise. We can now state the BPR PRF.

Theorem 4. *Let $n = n(\lambda)$ be a positive integer, $r = r(n)$ be a noise parameter, $k = k(\lambda) = \mathsf{poly}(\lambda)$ be a positive integer and let p_1, p_2 be moduli such that $p_1 \geq p_2 \cdot n \cdot (Cr\sqrt{k})^n \cdot k^{\omega(1)}$, where C is a universal constant. The keyed function $\mathsf{BPR}_n : \mathcal{K}_n \times \{0,1\}^n \to \mathbb{Z}_{p_2}^k$ with key space $\mathcal{K}_n = \mathbb{Z}_{p_2}^k \times \left(\mathbb{Z}_{p_2}^{k \times k}\right)^n$ is defined by*

$$\mathsf{BPR}_n(K, x) = \left\lfloor \mathbf{a}^\top \prod_{j=1}^{n} \mathbf{S}_j^{x_j} \right\rceil_{p_2},$$

where $\mathbf{a} \leftarrow_\$ \mathbb{Z}_{p_1}^k$ and $\mathbf{S}_1, \ldots, \mathbf{S}_n \leftarrow_\$ \chi_\alpha^{k \times k}$ and $K = (\mathbf{a}, \mathbf{S}_1, \ldots, \mathbf{S}_n)$.

Assume that $\mathsf{LWE}(p_1, k, \chi_r)$ is hard. Then BPR_n is a pseudorandom function. Specifically, if there exists a distinguisher $\mathcal{D}$ that distinguishes BPR_n with advantage ϵ from a random function, then there exists a distinguisher $\mathcal{D}'$ with essentially the same runtime as $\mathcal{D}$ that distinguishes $\mathsf{LWE}(p_1, k, \chi_r)$ with advantage $\epsilon/(k \cdot n)$.

Observe that in Theorem 4 the underlying hardness assumption changes when we increase the input length n. More specifically, the smaller the term p_1/r is, the harder the underlying LWE problem $\mathsf{LWE}(p_1, k, \chi_r)$ becomes. The term p_1/r is dominated by $(Cr\sqrt{k})^n$, thus we aim towards minimizing n. Observe that we can fix a modulus p_2 for the whole family BPR_n, therefore all functions in this

family have the same output domain. Plugging the BPR_n as small domain PRF in the construction of Theorem 3 yields that n never becomes larger than $\log(q)$ for some $q = \mathsf{poly}(n)$. Thus we can base the security of the PRF in Theorem 3 on $\mathsf{LWE}(p_1, k, \chi_r)$ with $p_1 = p_2 \cdot n \cdot (Cr\sqrt{k})^{\log(2\lambda q)} \cdot k^{\omega(1)}$, which is slightly super-polynomial (instead of sub-exponential). Moreover, since the BPR_n loses only a factor $k \cdot n$ in its security reduction to LWE, the resulting PRF from Theorem 3 loses only a factor of $k \cdot \log(2\lambda q)$. We remark that using the more efficient and tighter Ring-LWE based PRF of [1], the security reduction to Ring-LWE loses only a factor of $\log(2\lambda q)$.

While the construction from Theorem 3 preserves the parallel complexity of the small domain PRF, the overall complexity of evaluating the PRF may actually increase. We consider it an interesting problem to find a PRF construction which enjoys similar properties as the k-LIN based construction in Section 4, i.e. one improves the underlying small domain PRF in all aspects, in particular key size and evaluation complexity.

4 A Direct Construction from the k-LIN Problem

In this section, we will provide our efficient constructions of number-theoretic PRFs. As discussed above, we will first develop a specialized domain extension technique and then construct a large domain PRF using a tailor-made on-the-fly adaptation strategy.

4.1 Preliminaries

In this section, we will generally index vectors of length n with indices $0, \ldots, n-1$. We will denote the identity matrix in $\mathbb{Z}_p^{k \times k}$ by $\mathbf{I}$. For vectors $\mathbf{a} \in \mathbb{Z}_p^k$ we define exponentiation component-wise, i.e. $g^{\mathbf{a}} = (g^{a_0}, \ldots, g^{a_{k-1}})$. The decisional k-linear assumption (k-LIN) [31,17] generalizes the decisional DDH problem. The decisional k-Linear assumption becomes (generically) weaker when the parameter k grows, where the instance $k = 1$ corresponds to DDH and $k = 2$ to the linear assumption [6]. The main motivation for these assumptions is that groups are known, where the DDH assumption is easy, but the computational Diffie Hellman problem is supposedly hard [16].

Definition 5 (Decisional k-LIN Problem). *Let $\mathbb{G}$ be a cyclic group of prime order p. Let $g_0, g_1, \ldots, g_k \leftarrow_\$ \mathbb{G}$ and $s_1, \ldots, s_k, r \leftarrow_\$ \mathbb{Z}_p$ be chosen uniformly at random. The goal of the k-LIN problem in $\mathbb{G}$ is to distinguish the distributions*

$$(g_0, \ldots, g_k, g_1^{s_1}, \ldots, g_k^{s_k}, g_0^{\sum_{i=1}^{k} s_i}) \text{ and } (g_0, \ldots, g_k, g_1^{s_1}, \ldots, g_k^{s_k}, g_0^{r}).$$

We will use the PRF construction of Lewko and Waters [21] as underlying small domain PRF in our construction.

Theorem 5. *Let $k \geq 1$ be a positive integer, $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order p and $n = n(\lambda)$ be a positive integer. Define the keyed function $\mathsf{LW}_n : \mathcal{K}_n \times \{0,1\}^n \to \mathbb{G}$ with key space $\mathcal{K}_n = \mathbb{Z}_p^k \times \left(\mathbb{Z}_p^{k \times k}\right)^n$ by*

$$\mathsf{LW}_n(K, x) = g^{\mathbf{a}^\top \cdot \prod_{j=0}^{n-1} \mathbf{S}_j^{x_j}},$$

where $\mathbf{a} \leftarrow_\$ \mathbb{Z}_p^k$, $\mathbf{S}_0, \ldots, \mathbf{S}_{n-1} \leftarrow_\$ \mathbb{Z}_p^{k \times k}$ and $K = (\mathbf{a}, \mathbf{S}_0, \ldots, \mathbf{S}_{n-1})$. If the k-LIN problem is hard in $\mathbb{G}$, then LW_n is a pseudorandom function. More specifically, assume that there exists a PPT distinguisher $\mathcal{D}$ that distinguishes LW_n with advantage ϵ from a random function. Then there exists a PPT distinguisher $\mathcal{D}'$ that distinguishes the k-LIN problem with advantage $\epsilon/(k \cdot n)$.

The Lewko-Waters PRF LW as described in the construction in Theorem 5 outputs k group elements and therefore requires k exponentiations. We can truncate the output of the LW PRF to a single group element, thereby only requiring a single exponentiation.

4.2 A bounded PRF from k-LIN

We will now provide an efficient construction of a bounded PRF from k-LIN. The security of this bounded PRF tightly reduces to the security of a small domain LW PRF and therefore to k-LIN with only a logarithmic loss.

Construction 3 *Let $k \geq 1$ be a positive integer, $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order p. The keyed function $F_\ell : \mathcal{K}_\ell \times \mathbb{Z}_p \to \mathbb{G}$ with key space $\mathcal{K}_\ell = \mathbb{Z}_p^k \times \left(\mathbb{Z}_p^{k \times k}\right)^{\log(\ell)}$ is defined by*

$$F_\ell(K_\ell, x) = g^{\mathbf{a}^\top \cdot \prod_{j=0}^{\log(\ell)-1} (\mathbf{S}_j + x^{2^j} \cdot \mathbf{I})},$$

where $\mathbf{a} \leftarrow_\$ \mathbb{Z}_p^k$, $\mathbf{S}_0, \ldots, \mathbf{S}_{\log(\ell)-1} \leftarrow_\$ \mathbb{Z}_p^{k \times k}$ and $K_\ell = (\mathbf{a}, \mathbf{S}_0, \ldots, \mathbf{S}_{\log(\ell)-1})$

For a bit $b \in \{0,1\}$ let $\neg b = 1 - b$ denote the negation of b. For a bit-vector $\mathbf{c} \in \{0,1\}^m$ let $\neg \mathbf{c}$ denote the bit-wise negation of $\mathbf{c}$. We will need the following technical lemma.

Lemma 2. *Let p be a prime integer. It holds for all $r \in \mathbb{N}_{>0}$, all matrices $\mathbf{S}_0, \ldots, \mathbf{S}_{r-1} \in \mathbb{Z}_p^{k \times k}$ and all $x \in \mathbb{Z}_p$ that*

$$\prod_{j=0}^{r-1} (\mathbf{S}_j + x^{2^j} \mathbf{I}) = \sum_{\mathbf{c} \in \{0,1\}^r} \left(\prod_{j=0}^{r-1} \mathbf{S}_j^{\neg c_j} \right) x^{\sum_{j=0}^{r-1} c_j 2^j}.$$

The proof of Lemma 2 works by inductively expanding the left side of the equation and can be found in the full version of this paper. We will now show that the function F_ℓ given in Construction 3 is a bounded PRF.

Theorem 6. *Assume that the k-LIN problem is hard in $\mathbb{G}$. Then the function F_ℓ defined in Construction 3 is a bounded PRF. More specifically let $\ell^* \leq \mathsf{poly}(\lambda)$ and assume that $\mathcal{D}$ is an ℓ^*-query PPT distinguisher with advantage ϵ against the pseudorandomness of F_{ℓ^*}. Then there exists a distinguisher $\mathcal{D}'$ (with essentially the same runtime as $\mathcal{D}$) with advantage $\frac{\epsilon}{k \cdot \log(\ell^*)}$ against k-LIN.*

Proof. First observe that F_ℓ can be computed in time $\mathsf{poly}(\lambda, \log(\ell))$. Notice that $\mathsf{LW}_{\log(\ell)}$ and F_ℓ have identical key-spaces. Let $K = (\mathbf{a}, \mathbf{S}_0, \ldots, \mathbf{S}_{\log(\ell)-1})$ be a key for F_ℓ. It follows immediately by Lemma 2 that we can compute F_ℓ by

$$\begin{aligned}
F_\ell(K, x) &= g^{\mathbf{a}^\top \cdot \prod_{j=0}^{\log(\ell)-1} (\mathbf{S}_j + x^{2^j} \cdot \mathbf{I})} \\
&= g^{\mathbf{a}^\top \cdot \sum_{\mathbf{c} \in \{0,1\}^{\log(\ell)}} \left(\prod_{i=0}^{\log(\ell)-1} \mathbf{S}_i^{\neg c_i} \right) x^{\sum_{i=0}^{\log(\ell)-1} c_i 2^i}} \\
&= \prod_{\mathbf{c} \in \{0,1\}^{\log(\ell)}} g^{\mathbf{a}^\top \cdot \left(\prod_{i=0}^{\log(\ell)-1} \mathbf{S}_i^{\neg c_i} \right) x^{\sum_{i=0}^{\log(\ell)-1} c_i 2^i}} \\
&= \prod_{\mathbf{c} \in \{0,1\}^{\log(\ell)}} \left(\mathsf{LW}_{\log(\ell)}(K, \neg \mathbf{c}) \right)^{x^{\sum_{i=0}^{\log(\ell)-1} c_i 2^i}}
\end{aligned}$$

For an integer $j \in \{0, \ldots, 2^{\lceil \log(\ell) \rceil} - 1\}$ let $\mathsf{BIN}(j)$ denote the $\log(\ell)$ bit binary representation of j, i.e. it holds that $j = \sum_{i=0}^{\log(\ell)-1} \mathsf{BIN}(j)_i 2^i$. Thus, it holds that

$$F_\ell(K, x) = \prod_{j=0}^{2^{\lceil \log(\ell) \rceil} - 1} \left(\mathsf{LW}_{\log(\ell)}(K, \neg \mathsf{BIN}(j)) \right)^{x^j}. \tag{4}$$

Now, let $\ell^* \leq \mathsf{poly}(\lambda)$ and assume that $\mathcal{D}$ is a ℓ^*-query PPT distinguisher that distinguishes F_{ℓ^*} with advantage ϵ from a random function. We will construct a PPT distinguisher $\mathcal{D}'$ that distinguishes $\mathsf{LW}_{\log(\ell^*)}$ from a random function with advantage ϵ. Since the function table of a function $\{0,1\}^{\log(\ell^*)} \to \mathbb{G}^k$ has size $2^{\lceil \log(\ell^*) \rceil} \cdot k \log(|\mathbb{G}|) \leq 2\ell^* k \log(|\mathbb{G}|) = \mathsf{poly}(\lambda)$, we can assume that $\mathcal{D}'$'s input is an explicit function table.

Distinguisher $\mathcal{D}'$
Input: Function $T : \{0,1\}^{\log(\ell^*)} \to \mathbb{G}^k$, encoded as a function table
$b' \leftarrow \mathcal{D}^{\mathcal{O}(\cdot)}(1^\lambda)$
return b'

$\mathcal{O}(x)$:
$y \leftarrow \prod_{j=0}^{2^{\lceil \log(\ell^*) \rceil} - 1} (T(\neg \mathsf{BIN}(j)))^{x^j}$
return y

First observe that $\mathcal{D}'$ is efficient as $\mathcal{D}$ is efficient and the oracle $\mathcal{O}$ can be implemented efficiently (as $2^{\lceil \log(\ell^*) \rceil} \leq 2\ell^*$). We will now analyze the advantage of $\mathcal{D}'$. If $\mathcal{D}'$'s input T is a function $\mathsf{LW}_{\log(\ell^*)}(K, \cdot)$ for a randomly chosen $K \leftarrow_\$ \mathcal{K}_{\log(\ell^*)}$,

then clearly by (4) it holds that the oracle $\mathcal{O}$ implements exactly $F_{\ell^*}(K,\cdot)$. On the other hand, if T implements a random function $R' : \{0,1\}^{\log(\ell^*)} \to \mathbb{G}^k$, then we can express R' by $R'(\neg\mathsf{BIN}(j)) = g^{\mathbf{a}_j^\top}$ for all $j = 0, \ldots, 2^{\lceil \log(\ell^*)\rceil - 1}$, where the $\mathbf{a}_0, \ldots, \mathbf{a}_{2^{\lceil \log(\ell^*)\rceil - 1}} \leftarrow_\$ \mathbb{G}^k$ are chosen uniformly at random. Thus, in this case the function computed by $\mathcal{O}$ is

$$\begin{aligned}\mathcal{O}(x) &= \prod_{j=0}^{2^{\lceil \log(\ell^*)\rceil - 1}} g^{\mathbf{a}_j^\top x^j} \\ &= g^{\sum_{j=0}^{2^{\lceil \log(\ell^*)\rceil - 1}} \mathbf{a}_j^\top x^j},\end{aligned}$$

which is an ℓ^*-wise independent function. To see this, note that g-exponentiation is an isomorphism and the function in the exponent $\sum_{j=0}^{2^{\lceil \log(\ell^*)\rceil - 1}} \mathbf{a}_j^\top x^j$ is a random polynomial of degree $2^{\lceil \log(\ell^*)\rceil} - 1 \geq \ell^* - 1$, which is an ℓ^*-wise independent function. Thus, from the view of $\mathcal{D}$ the oracle $\mathcal{O}$ implements a random function R, as $\mathcal{D}$ sends at most ℓ^* distinct queries. We conclude

$$\begin{aligned}\mathsf{Adv}(\mathcal{D}') &= |\Pr[\mathcal{D}'^{\mathsf{LW}_{\log(\ell^*)}(K,\cdot)} = 1] - \Pr[\mathcal{D}'^{R'} = 1]| \\ &= |\Pr[\mathcal{D}^{F_{\ell^*}(K,\cdot)} = 1] - \Pr[\mathcal{D}^{R} = 1]| = \epsilon.\end{aligned}$$

By Theorem 5, the distinguisher $\mathcal{D}'$ yields a distinguisher $\mathcal{D}''$ with advantage $\frac{\epsilon}{k\log(\ell^*)}$ against k-LIN.

4.3 In-Place On-the-Fly Adaptation

While the general on-the-fly adaptation strategy we will provide in Section 3.2 needs to replicate the the underlying bounded PRF t times, we will now provide a specific on-the-fly adaptation technique for the bounded PRF F_ℓ provided in the last paragraph that involves no expansion whatsoever. Due to the special algebraic structure of F_ℓ, this on-the-fly adaptation can be done in-place. To obtain an unbounded PRF from the bounded PRF of Construction 3, we will set the upper limit of the product in the exponent from $\log(\ell)$ to some $t = \omega(\log(\lambda))$. We thereby ensure that t is large enough that we can embed F_{ℓ^*} in this PRF for any $\ell^* \leq \mathsf{poly}(\lambda)$.

Construction 4 *Let $k \geq 1$ be a positive integer and $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order p. Let $t = \omega(\log(\lambda))$. The keyed function $F : \mathcal{K} \times \mathbb{Z}_p \to \mathbb{G}$ with key space $\mathcal{K} = \mathbb{Z}_p^k \times \left(\mathbb{Z}_p^{k\times k}\right)^t$ is defined by*

$$F(K,x) = g^{\mathbf{a}^\top \cdot \prod_{j=0}^{t-1} (\mathbf{S}_j + x^{2^j}\cdot \mathbf{I})},$$

where $\mathbf{a} \leftarrow_\$ \mathbb{Z}_p^k$, $\mathbf{S}_0, \ldots, \mathbf{S}_{t-1} \leftarrow_\$ \mathbb{Z}_p^{k\times k}$ and $K = (\mathbf{a}, \mathbf{S}_0, \ldots, \mathbf{S}_{t-1})$.

We still need the following auxiliary lemma which states that a randomly chosen matrix from $\mathbb{Z}_p^{k\times k}$ has full rank, except with small probability.

Lemma 3. *Let p be a prime and $\mathbf{S} \leftarrow_\$ \mathbb{Z}_p^{k\times k}$ be chosen uniformly at random. Then it holds that*

$$\Pr[\mathsf{rank}(\mathbf{S}) < k] \leq \frac{1}{p-1}.$$

The proof of Lemma 3 is standard.

Theorem 7. *Assume that the k-LIN problem is hard in $\mathbb{G}$. Then the function F defined in Construction 4 is a PRF. More specifically assume that $\mathcal{D}$ is PPT distinguisher that makes at most $q = \mathsf{poly}(\lambda)$ queries and distinguishes F with advantage ϵ from a uniformly random function. Then there exists a PPT distinguisher $\mathcal{D}^*$ (with essentially the same runtime as $\mathcal{D}$) with advantage $\frac{1}{k\cdot\log(q)} \cdot \left(\epsilon - \frac{qt}{(p-1)}\right)$ against k-LIN in $\mathbb{G}$.*

Proof. Let $\mathcal{D}$ be a distinguisher with advantage ϵ against the pseudorandomness of F which makes at most $q = \mathsf{poly}(n)$ queries. Note that since $q = \mathsf{poly}(\lambda)$ and $t = \omega(\log(\lambda))$, it holds $\log(q) \leq t-1$ (for a sufficiently large λ). We will define 3 hybrid experiments. In hybrid i $\mathcal{D}$ is given access to a function $F^{(i)} : \mathbb{Z}_p \to \mathbb{G}^k$.

- Hybrid $\mathfrak{H}_1$: In this experiment $\mathcal{D}$ is given oracle access to the function $F^{(1)}$ given by $F^{(1)}(x) = F(K, x)$ for a randomly chosen $K \leftarrow_\$ \mathcal{K}$.
- Hybrid $\mathfrak{H}_2$: In this experiment $\mathcal{D}$ is given oracle access to the function $F^{(2)}$ defined by

  $$F^{(2)}(x) = g^{\mathbf{r}(x)^\top \cdot \prod_{j=\log(q)}^{t} (\mathbf{S}_j + x^{2^j}\mathbf{I})},$$

  where $\mathbf{r} : \mathbb{Z}_p \to \mathbb{Z}_p^k$ is a uniformly random function and $\mathbf{S}_{\log(q)}, \ldots, \mathbf{S}_{t-1} \leftarrow_\$ \mathbb{Z}_p^{k\times k}$.
- Hybrid $\mathfrak{H}_3$: In this experiment $\mathcal{D}$ is given oracle access to a uniformly random function $F^{(3)}$.

Clearly, it holds that

$$|\Pr[\mathcal{D}^{F^{(1)}} = 1] - \Pr[\mathcal{D}^{F^{(3)}} = 1]| \geq \epsilon.$$

Define

$$\epsilon_1 = |\Pr[\mathcal{D}^{F^{(1)}} = 1] - \Pr[\mathcal{D}^{F^{(2)}} = 1]|$$
$$\epsilon_2 = |\Pr[\mathcal{D}^{F^{(2)}} = 1] - \Pr[\mathcal{D}^{F^{(3)}} = 1]|.$$

By the triangle inequality it holds that

$$\epsilon \leq \epsilon_1 + \epsilon_2.$$

We will first show that $\epsilon_2 \leq qt/(p-1)$. Define

$$\mathbf{M}(x) = \prod_{j=\log(q)}^{t-1} (\mathbf{S}_j + x^{2^j}\mathbf{I}),$$

and observe that $F^{(2)}(x) = g^{\mathbf{r}^\top(x)\cdot\mathbf{M}(x)}$. Now, if it holds for distinct $x_1, \ldots, x_q \in \mathbb{Z}_p$ that $\mathsf{rank}(\mathbf{M}(x_i)) = k$ for $i = 1, \ldots, q$, then $\mathbf{r}^\top(x_1)\cdot\mathbf{M}(x_1), \ldots, \mathbf{r}^\top(x_q)\cdot\mathbf{M}(x_q)$ are distributed independently and uniformly at random. Thus it also holds that $F^{(2)}(x_1), \ldots, F^{(2)}(x_q)$ are distributed independently and uniformly at random. We can define the predicate $\mathsf{good}(\{x_1, \ldots, x_q\}, \mathbf{M})$ to be true if and only if it holds $\mathsf{rank}(\mathbf{M}(x_i)) = k$ for $i = 1, \ldots, q$. Applying Lemma 1 yields

$$\begin{aligned}\epsilon_2 &= |\Pr[\mathcal{D}^{F^{(2)}} = 1] - \Pr[\mathcal{D}^{F^{(3)}} = 1]| \\ &\leq \max_{x_1,\ldots,x_\ell} \Pr[\neg\mathsf{good}(\{x_1, \ldots, x_q\}, \mathbf{M})] \\ &= \max_{x_1,\ldots,x_\ell} \Pr[\exists i : \mathsf{rank}(\mathbf{M}(x_i)) < k].\end{aligned}$$

For a fixed x it holds that $\mathsf{rank}(\mathbf{M}(x)) < k$ if there exists a $j \in \{\log q, \ldots, t-1\}$ with $\mathsf{rank}(\mathbf{S}_j + x^{2^j}\mathbf{I}) < k$. Since $\mathbf{S}_j$ is chosen uniformly at random it holds by Lemma 3 that

$$\Pr[\mathsf{rank}(\mathbf{S}_j + x^{2^j}\mathbf{I}) < k] = \Pr[\mathsf{rank}(\mathbf{S}_j) < k] \leq \frac{1}{p-1}.$$

By a union bound over the j it holds that $\Pr[\mathsf{rank}(\mathbf{M}(x)) < k] \leq \frac{t}{p-1}$. By another union bound over $i = 1, \ldots, q$ it holds that

$$\Pr[\exists i : \mathsf{rank}(\mathbf{M}(x_i)) < k] \leq \frac{qt}{p-1}$$

We conclude $\epsilon_2 \leq qt/(p-1)$ and therefore $\epsilon_1 \geq \epsilon - qt/(p-1)$.

Now let $\ell^* = 2^{\lceil \log(q) \rceil}$. We will now construct a PPT distinguisher $\mathcal{D}'$ that distinguishes the bounded PRF F_{ℓ^*} with advantage ϵ_2. The distinguisher $\mathcal{D}'$ is given in Figure 3.

Distinguisher $\mathcal{D}'$	$\mathcal{O}'(x)$:
Has access to oracle $\mathcal{O}(\cdot)$	$y_0 \leftarrow \mathcal{O}(x)$
$\mathbf{S}_{\log(q)}, \ldots, \mathbf{S}_t \leftarrow_\$ \mathbb{Z}_p^{k\times k}$	$y \leftarrow y_0^{\prod_{j=\log(q)}^{t-1}(\mathbf{S}_j + x^{2^j}\mathbf{I})}$
$b' \leftarrow \mathcal{D}^{\mathcal{O}'(\cdot)}(1^\lambda)$	return y
return b'	

Fig. 3. The distinguisher $\mathcal{D}'$

First assume that $\mathcal{D}'$'s oracle $\mathcal{O}$ implements the function $F_{\ell^*}(K_{\ell^*}, x) = g^{\mathbf{a}^\top \prod_{j=0}^{\log(q)-1}(\mathbf{S}_j + x^{2^j}\mathbf{I})}$ where $K_\ell = (\mathbf{a}, \mathbf{S}_0, \ldots, \mathbf{S}_{\log(q)-1})$ is a uniformly chosen key

for F_{ℓ^*}. Then the oracle $\mathcal{O}'$ implements the function

$$\begin{aligned}\mathcal{O}'(x) &= \left(g^{\mathbf{a}^\top \prod_{j=0}^{\log(q)-1}(\mathbf{S}_j+x^{2^j}\mathbf{I})}\right)^{\prod_{j=\log(q)}^{t-1}(\mathbf{S}_j+x^{2^j}\mathbf{I})} \\ &= g^{\left(\mathbf{a}^\top \prod_{j=0}^{\log(q)-1}(\mathbf{S}_j+x^{2^j}\mathbf{I})\right)\cdot\prod_{j=\log(q)}^{t-1}(\mathbf{S}_j+x^{2^j}\mathbf{I})} \\ &= g^{\mathbf{a}^\top \prod_{j=0}^{t-1}(\mathbf{S}_j+x^{2^j}\mathbf{I})}.\end{aligned}$$

Thus $\mathcal{O}'$ implements exactly $F^{(1)}$. On the other hand, if $\mathcal{D}'$'s oracle $\mathcal{O}$ implements a random function R with $R(x) = g^{\mathbf{r}(x)^\top}$, where $\mathbf{r} : \mathbb{Z}_p \to \mathbb{Z}_p^k$ is a random function, then the oracle $\mathcal{O}'$ implements the function

$$\begin{aligned}\mathcal{O}'(x) &= \left(g^{\mathbf{r}^\top(x)}\right)^{\prod_{j=\log(q)}^{t-1}(\mathbf{S}_j+x^{2^j}\mathbf{I})} \\ &= g^{\mathbf{r}^\top(x)\cdot\prod_{j=\log(q)}^{t-1}(\mathbf{S}_j+x^{2^j}\mathbf{I})}.\end{aligned}$$

Thus $\mathcal{O}'(x)$ implements exactly $F^{(2)}$. We conclude that

$$\begin{aligned}\mathsf{Adv}(\mathcal{D}') &= |\Pr[{\mathcal{D}'}^{F_{\ell^*}(K_{\ell^*},\cdot)} = 1] - \Pr[{\mathcal{D}'}^{R} = 1]| \\ &= |\Pr[\mathcal{D}^{F^{(1)}} = 1] - \Pr[\mathcal{D}^{F^{(2)}} = 1]| = \epsilon_1 \geq \epsilon - \frac{qt}{p-1}.\end{aligned}$$

By Theorem 6 this yields a distinguisher $\mathcal{D}^*$ with advantage $\frac{1}{k\cdot\log(q)}\cdot\left(\epsilon - \frac{qt}{(p-1)}\right)$ against k-LIN in $\mathbb{G}$. This concludes the proof.

PRF with Shorter Keys. Escala et al. [11] suggested a framework that generalizes Diffie-Hellman like decisional assumptions and proposed a variant of the Lewko-Waters PRF with short keys based on the so-called Matrix-DDH (MDDH) assumption. The proof of Theorem 6 immediately generalizes to this setting. Theorem 7 also holds in this setting, given that the distribution of *aggregated transformation matrices* $\mathbf{T}$ corresponding to the matrix distribution $\mathcal{D}_{\ell,k}$ (c.f. [11], Section 5.3) used in the MDDH problem satisfies $\Pr[\mathsf{rank}(\mathbf{T} + x\cdot\mathbf{I}) < k] \leq \mathsf{negl}$ for all $x \in \mathbb{Z}_p$.

Acknowledgements

We thank Max Rabkin and the reviewers of CRYPTO 2015 for their helpful comments and feedback.

References

1. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Berlin, Germany, Cambridge, UK (Apr 15–19, 2012)

2. Bellare, M., Goldwasser, S.: New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 194–211. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 20–24, 1990)
3. Benabbas, S., Gennaro, R., Vahlis, Y.: Verifiable delegation of computation over large datasets. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 111–131. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 14–18, 2011)
4. Berman, I., Haitner, I.: From non-adaptive to adaptive pseudorandom functions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 357–368. Springer, Berlin, Germany, Taormina, Sicily, Italy (Mar 19–21, 2012)
5. Berman, I., Haitner, I., Komargodski, I., Naor, M.: Hardness preserving reductions via Cuckoo hashing. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 40–59. Springer, Berlin, Germany, Tokyo, Japan (Mar 3–6, 2013)
6. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Berlin, Germany, Interlaken, Switzerland (May 2–6, 2004)
7. Boneh, D., Montgomery, H.W., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 10. pp. 131–140. ACM Press, Chicago, Illinois, USA (Oct 4–8, 2010)
8. Chandran, N., Garg, S.: Balancing output length and query bound in hardness preserving constructions of pseudorandom functions. In: Meier, W., Mukhopadhyay, D. (eds.) INDOCRYPT 2014. LNCS, vol. 8885, pp. 89–103. Springer, Berlin, Germany, New Delhi, India (Dec 14–17, 2014)
9. Chase, M., Meiklejohn, S.: Déjà Q: Using dual systems to revisit q-type assumptions. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 622–639. Springer, Berlin, Germany, Copenhagen, Denmark (May 11–15, 2014)
10. Dodis, Y., Yampolskiy, A.: A verifiable random function with short proofs and keys. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 416–431. Springer, Berlin, Germany, Les Diablerets, Switzerland (Jan 23–26, 2005)
11. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 18–22, 2013)
12. Fiore, D., Gennaro, R.: Publicly verifiable delegation of large polynomials and matrix computations, with applications. In: Yu, T., Danezis, G., Gligor, V.D. (eds.) ACM CCS 12. pp. 501–512. ACM Press, Raleigh, NC, USA (Oct 16–18, 2012)
13. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions (extended abstract). In: 25th FOCS. pp. 464–479. IEEE Computer Society Press, Singer Island, Florida (Oct 24–26, 1984)
14. Goldreich, O., Goldwasser, S., Micali, S.: On the cryptographic applications of random functions. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 276–288. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 19–23, 1984)
15. Hazay, C.: Oblivious polynomial evaluation and secure set-intersection from algebraic PRFs. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 90–120. Springer, Berlin, Germany, Warsaw, Poland (Mar 23–25, 2015)
16. Herranz, J., Hofheinz, D., Kiltz, E.: The kurosawa-desmedt key encapsulation is not chosen-ciphertext secure. Cryptology ePrint Archive, Report 2006/207 (2006), http://eprint.iacr.org/

17. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 19–23, 2007)
18. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Cryptography with constant computational overhead. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 433–442. ACM Press, Victoria, British Columbia, Canada (May 17–20, 2008)
19. Jain, A., Pietrzak, K., Tentes, A.: Hardness preserving constructions of pseudo-random functions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 369–382. Springer, Berlin, Germany, Taormina, Sicily, Italy (Mar 19–21, 2012)
20. Levin, L.: One way functions and pseudorandom generators. Combinatorica 7(4), 357–363 (1987)
21. Lewko, A.B., Waters, B.: Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In: Al-Shaer, E., Jha, S., Keromytis, A.D. (eds.) ACM CCS 09. pp. 112–120. ACM Press, Chicago, Illinois, USA (Nov 9–13, 2009)
22. Luby, M.: Pseudorandomness and Cryptographic Applications. Princeton University Press, Princeton, NJ, USA (1994)
23. Maurer, U.M.: Indistinguishability of random systems. In: Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings. pp. 110–132 (2002)
24. Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. In: 36th FOCS. pp. 170–181. IEEE Computer Society Press, Milwaukee, Wisconsin (Oct 23–25, 1995)
25. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: 38th FOCS. pp. 458–467. IEEE Computer Society Press, Miami Beach, Florida (Oct 19–22, 1997)
26. Naor, M., Reingold, O.: On the construction of pseudo-random permutations: Luby-Rackoff revisited (extended abstract). In: 29th ACM STOC. pp. 189–199. ACM Press, El Paso, Texas, USA (May 4–6, 1997)
27. Naor, M., Reingold, O., Rosen, A.: Pseudo-random functions and factoring (extended abstract). In: 32nd ACM STOC. pp. 11–20. ACM Press, Portland, Oregon, USA (May 21–23, 2000)
28. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009. pp. 333–342 (2009)
29. Razborov, A.A., Rudich, S.: Natural proofs. In: 26th ACM STOC. pp. 204–213. ACM Press, Montréal, Québec, Canada (May 23–25, 1994)
30. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005. pp. 84–93 (2005)
31. Shacham, H.: 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/
32. Valiant, L.G.: A theory of the learnable. Commun. ACM 27(11), 1134–1142 (Nov 1984), http://doi.acm.org/10.1145/1968.1972