Multilinear and Aggregate Pseudorandom Functions: New Constructions and Improved Security

Michel Abdalla, Fabrice Benhamouda, and Alain Passelègue

ENS, CNRS, INRIA, and PSL, Paris, France.
{michel.abdalla,fabrice.ben.hamouda,alain.passelegue}@ens.fr,
http://www.di.ens.fr/∼{mabdalla,fbenhamo,passeleg}

Abstract. Since its introduction, pseudorandom functions (PRFs) have become one of the main building blocks of cryptographic protocols. In this work, we revisit two recent extensions of standard PRFs, namely multilinear and aggregate PRFs, and provide several new results for these primitives. In the case of aggregate PRFs, one of our main results is a proof of security for the Naor-Reingold PRF with respect to read-once boolean aggregate queries under the standard Decision Diffie-Hellman problem, which was an open problem. In the case of multilinear PRFs, one of our main contributions is the construction of new multilinear PRFs achieving indistinguishability from random symmetric and skew-symmetric multilinear functions, which was also left as an open problem. In order to achieve these results, our main technical tool is a simple and natural generalization of the recent linear independent polynomial framework for PRFs proposed by Abdalla, Benhamouda, and Passelègue in Crypto 2015, that can handle larger classes of PRF constructions. In addition to simplifying and unifying proofs for multilinear and aggregate PRFs, our new framework also yields new constructions which are secure under weaker assumptions, such as the decisional k-linear assumption.

Keywords. Pseudorandom functions, multilinear PRFs, aggregate PRFs.

1 Introduction

Pseudorandom functions (PRFs) are one of the most fundamental primitives in cryptography. One of the features that makes PRFs so useful is the fact that they behave as truly random functions with respect to computationally bounded adversaries. Since being introduced by Goldreich, Goldwasser, and Micali [15], PRFs have been used in many cryptographic applications, varying from symmetric encryption and authentication schemes to key exchange. In particular, they are very useful for modeling the security of concrete block ciphers, such as AES [4].

Given the large applicability of pseudorandom functions, several extensions have been proposed in the literature over the years, with the goal of providing additional functionalities to these functions. One concrete example of such an extension are constrained PRFs [17,11,9], which provides the owner of the secret

key with the capability of delegating the computation of the pseudorandom function for different subsets of the input domain, without compromising the pseudorandomness property for the other points of the input domain. In this paper, we focus on two recent extensions of pseudorandom functions, namely multilinear PRFs [13], and aggregate PRFs [12], and solve several open problems related to the construction of these primitives.

Aggregate Pseudorandom Functions. Aggregate pseudorandom functions were introduced by Cohen, Goldwasser, and Vaikuntanathan in [12]. The main interest of an aggregate PRF is to provide the user with the possibility of aggregating the values of the function over *super-polynomially* many PRF values with only a *polynomial-time* computation, without enabling a polynomial-time adversary to distinguish the function from a truly random function. For instance, one such example of an aggregate query could be to compute the product of all the output values of the PRF corresponding to a given exponentially-sized interval of the input domain.

In addition to proposing the notion of aggregate PRFs, Cohen, Goldwasser, and Vaikuntanathan [12] also proposed new constructions for several different classes of aggregate queries, such as decision trees, hypercubes, and read-once boolean formulas, achieving different levels of expressiveness. Unfortunately, for most of the constructions proposed in [12], the proofs of security suffer from an exponential (in the input length) overhead in their running time and have to rely on the sub-exponential hardness of the Decisional Diffie-Hellman (DDH) problem.

Indeed, to prove the security of their constructions, the authors use a generic result which is simply saying the following: given an adversary $\mathscr{A}$ against the AGG-PRF security of a PRF F, one can build an adversary $\mathscr{B}$ against the standard PRF security of F. $\mathscr{B}$ simply queries all the values required to compute the aggregate values (or the PRF values), and computes the aggregate values itself before sending them to $\mathscr{A}$.

Clearly, this reduction proves that any secure PRF is actually also a secure aggregate PRF. However, this reduction is *not efficient*, since to answer to just one aggregate query, the adversary $\mathscr{B}$ may have to query an exponential number of values to its oracle. Hence, as soon as we can aggregate in one query a superpolynomial number of PRF values, this generic reduction does not run in polynomial time.

Multilinear Pseudorandom Functions. In order to overcome the shortcomings of the work of Cohen, Goldwasser, and Vaikuntanathan [12], Cohen and Holmgren introduced the concept of multilinear pseudorandom functions in [13]. Informally speaking, a multilinear pseudorandom function is a variant of the standard notion of pseudorandom functions, which works with vector spaces and which guarantees indistinguishability from random multilinear functions with the same domain and range. As shown in [13], multilinear pseudorandom functions can be used to prove the AGG-PRF security of the Naor-Reingold (NR) PRF [18] with a polynomial time reduction for the case of hypercubes and decision trees aggregations. Unfortunately, their technique does not extend to the more general

case of read-once formulas aggregation, which is the most expressive form of aggregation in [12].

Our Techniques. In this work, we provide an alternative way of overcoming the limitations of the work of Cohen, Goldwasser, and Vaikuntanathan [12], based on a natural extension of the recent algebraic framework for pseudorandom functions proposed by Abdalla, Benhamouda, and Passelègue in [1], known as the linear independent polynomial (LIP) framework.

In a nutshell, the LIP framework essentially says that for any linearly independent polynomials $P_1, \ldots, P_q \in \mathbb{Z}_p[T_1, \ldots, T_n]$, the group elements

$$[P_1(\vec{a}) \cdot b], \ldots, [P_q(\vec{a}) \cdot b],$$

with $\vec{a} \xleftarrow{\$} \mathbb{Z}_p^n$ and $b \xleftarrow{\$} \mathbb{Z}_p$, are computationally indistinguishable from independent random group elements in $\mathbb{G}$, under the DDH assumption (when polynomials are multilinear) or the d-DDHI assumption (where d is the maximum degree of $P_1, \ldots, P_q$ in any indeterminate T_i). As a toy example, the LIP framework directly proves the security of the NR PRF defined as:

$$\mathsf{NR}((b, \vec{a}), x) = \left[b \prod_{i=1}^{n} a_i^{x_i} \right],$$

where $(b, \vec{a} = (a_1, \ldots, a_n)) \in \mathcal{K} = \mathbb{Z}_p \times \mathbb{Z}_p^n$ and $x \in \mathcal{D} = \{0,1\}^n$. Indeed, all the polynomials $P_x = b \prod_{i=1}^{n} a_i^{x_i}$ are linearly independent.

Unfortunately, the LIP framework is not enough to prove the security of multilinear PRFs or aggregate PRFs, as the outputs of the function (and the corresponding polynomials) may not be independent. To overcome these limitations, we provide a natural extension of the LIP framework, which we call polynomial linear pseudorandomness security (PLP), that can handle such dependences. Despite being a simple extension, the new PLP framework yields significant improvements over previous works on multilinear and aggregate PRFs. In particular, the multilinear constructions in [13] can be seen as a special case of our new PLP framework.

Main Results. Using our new PLP framework for pseudorandom functions, we obtain the following results.

First, we prove the security of the aggregate PRF for read-once formulas proposed in [12], under the DDH assumption and with a polynomial-time reduction. This in turn implies the security of all the other aggregate PRFs in [12], as the latter are particular cases of the aggregate PRFs for read-once formulas. The proof is very simple and based on linear algebra. Up to now, the only known reduction incurred an exponential blow-up in the length n of the input.

Second, we show that our PLP framework enables to very easily prove the security of the multilinear pseudorandom function construction in [13]. More importantly, it enables us to directly show the security of the symmetric variant of this construction, under the d-DDHI assumption, which was left as an open problem in [13].

Third, we extend all the above constructions to weaker assumptions, as the k-Lin assumption, which can hold in symmetric k-linear groups, contrary to DDH or d-DDHI. Again, these extensions are straightforward to prove thanks to our PLP framework.

Additionally, we solve two other open problems respectively in [12, end of Section 1 and Section 2.2] and in [13]: We show that unless NP=BPP, there cannot exist aggregate PRFs for DNF formulas, although satisfiability of DNF formulas can be tested in polynomial time; and we propose the first skew-symmetric multilinear PRF.

Additional Contributions. As a side contribution, we prove the hardness of $\mathcal{E}_{k,d}$-MDDH (defined in [1] and recalled in Section 2) in the generic (symmetric) k-linear group model, which was left as an open problem in [1] for $k > 2$ and $d > 1$. This result directly implies that all the results stated in [1] under the $\mathcal{E}_{2,d}$-MDDH now holds also for $\mathcal{E}_{k,d}$-MDDH, for any $k \geq 2$, which is also an interesting side contribution. To prove this result, we essentially need to prove there are no non-trivial polynomial relations of degree k between the elements of the assumptions (these elements being themselves polynomials), as in [8,10,14]. The proof is by induction over k: for the base case $k = 1$, the proof is straightforward as all the elements we consider are linearly independent; for the inductive case $k = 2$, we basically set some indeterminates to some carefully chosen values (for the polynomials defining the elements we consider) to come down to previous cases.

Paper Organization. The rest of the paper is composed of the following sections. In Section 2 and the full version [3], we give necessary background and notations. We introduce our general PLP security notion and explain our main result, termed PLP theorem (Theorem 1), in Section 3. We then present our new constructions and improved security bounds for aggregate and multilinear pseudorandom functions in Section 4 as well as some side results. The proofs of these results are detailed in the full version [3]. Finally, in the full version [3], we prove the hardness of our main assumption (the $\mathcal{E}_{k,d}$-MDDH assumption) in the generic k-linear group model.

2 Definitions

Notations and Conventions. We denote by κ the security parameter. Let $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a function that takes a key $K \in \mathcal{K}$ and an input $x \in \mathcal{D}$ and returns an output $F(K, x) \in \mathcal{R}$. The set of all functions $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ is denoted by $\mathsf{Fun}(\mathcal{K}, \mathcal{D}, \mathcal{R})$. Likewise, $\mathsf{Fun}(\mathcal{D}, \mathcal{R})$ denotes the set of all functions mapping $\mathcal{D}$ to $\mathcal{R}$. Also, if $\mathcal{D}$ and $\mathcal{R}$ are vector spaces, we denote by $\mathsf{L}(\mathcal{D}, \mathcal{R})$ the vector space of linear functions from $\mathcal{D}$ to $\mathcal{R}$. In addition, if $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are n vector spaces, then $\mathsf{L}(\mathcal{D}_1 \otimes \cdots \otimes \mathcal{D}_n, \mathcal{R})$ is the vector space of n-linear functions from $\mathcal{D}_1 \times \cdots \times \mathcal{D}_n$ to $\mathcal{R}$.

If S is a set, then $|S|$ denotes its size. We denote by $s \xleftarrow{\$} S$ the operation of picking at random s in S. If $\vec{x}$ is a vector then we denote by $|\vec{x}|$ its length, so $\vec{x} = (x_1, \ldots, x_{|\vec{x}|})$. For a binary string x, we denote its length by $|x|$ so

$x \in \{0,1\}^{|x|}$, x_i its i-th bit, so $x = x_1 \| \ldots \| x_{|x|}$. For a matrix $\boldsymbol{A}$ of size $k \times m$, we denote by $a_{i,j}$ the coefficient of $\boldsymbol{A}$ in the i-th row and the j-th column. We denote by $\mathbb{Z}_p[T_1, \ldots, T_n]$ the subspace of multivariate polynomials in indeterminates $T_1, \ldots, T_n$, and by $\mathbb{Z}_p[T_1, \ldots, T_n]_{\leq d}$ the subring of polynomials of degree at most d in each indeterminate. For a polynomial $P \in \mathbb{Z}_p[T_1, \ldots, T_n]$, we denote by $P(\vec{T})$ the polynomial $P(T_1, \ldots, T_n)$ and by $P(\vec{a})$ its evaluation by setting $\vec{T}$ to $\vec{a}$, meaning that we set $T_1 = a_1, \ldots, T_n = a_n$.

We often implicitly consider a multiplicative group $\mathbb{G} = \langle g \rangle$ with public generator g of order p and we denote by $[a]$ the element g^a, for any $a \in \mathbb{Z}_p$. Similarly, if $\boldsymbol{A}$ is a matrix in $\mathbb{Z}_p^{k \times m}$, $[\boldsymbol{A}]$ is a matrix $\boldsymbol{U} \in \mathbb{G}^{k \times m}$, such that $u_{i,j} = [a_{i,j}]$ for $i = 1, \ldots, k$ and $j = 1, \ldots, m$. All vector spaces are implicitly supposed to be $\mathbb{Z}_p$-vector spaces.

We denote by **TestLin** a procedure which takes as inputs a list $\mathcal{L}$ of polynomials $(R_1, \ldots, R_L)$ (such that $R_1, \ldots, R_L$ are linearly independent as polynomials) and a polynomial R and which outputs:

$$\begin{cases} \perp & \text{if } R \text{ is linearly independent of the set } \{R_1, \ldots, R_L\} \\ \vec{\lambda} = (\lambda_1, \ldots, \lambda_L) & \text{otherwise, so that } R = \lambda_1 R_1 + \ldots + \lambda_L R_L \end{cases}$$

$\vec{\lambda}$ is uniquely defined since we assume that polynomials from the input list are linearly independent. No such procedure is known for multivariate polynomials, if we require the procedure to be deterministic and polynomial-time. However, it is easy to construct such a randomized procedure which is correct with overwhelming probability. Such a statistical procedure is sufficient for our purpose and was given in [2]. We recall this procedure in Fig. 1. This procedure is correct with probability at least $\frac{p-1}{p}$ as soon as $nd \leq \sqrt{p}$, where d is the maximum degree in one indeterminate and n is the number of indeterminates.

Games [5]. Most of our definitions and proofs use the code-based game-playing framework, in which a game has an **Initialize** procedure, procedures to respond to adversary oracle queries, and a **Finalize** procedure. In the case where the **Finalize** procedure is not explicitly defined, it is implicitly defined as the procedure that simply outputs its input. To execute a game G with an adversary $\mathscr{A}$, we proceed as follows. First, **Initialize** is executed and its outputs become the input of $\mathscr{A}$. When $\mathscr{A}$ executes, its oracle queries are answered by the corresponding procedures of G. When $\mathscr{A}$ terminates, its outputs become the input of **Finalize**. The output of the latter, denoted $\mathrm{G}^{\mathscr{A}}$ is called the output of the game, and we let "$\mathrm{G}^{\mathscr{A}} \Rightarrow 1$" denote the event that this game output takes the value 1. The running time of an adversary by convention is the worst case time for the execution of the adversary with any of the games defining its security, so that the time of the called game procedures is included.

Pseudorandom Functions. A PRF is an efficiently computable ensemble of functions $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$, implicitly indexed by the security parameter κ, such that, when $K \xleftarrow{\$} \mathcal{K}$, the function $x \in \mathcal{D} \mapsto F(K, x) \in \mathcal{R}$ is indistinguishable from a random function. Formally, we say that F is a pseudorandom function if the advantage of any adversary $\mathscr{A}$ in attacking the standard PRF security of F is

procTestLin$(\mathcal{L}, R)$
// $\mathcal{L}[\ell] = R_\ell$ for $\ell = 1, \ldots, L$ and $L = |\mathcal{L}|$
$R_{L+1} \leftarrow R$
$N \leftarrow 2L + 4$
For $k = 1, \ldots, N$
  $\overrightarrow{\gamma_k} \xleftarrow{\$} \mathbb{Z}_p^n$
$\boldsymbol{M}$ matrix over $\mathbb{Z}_p$ of $L + 1$ rows and N columns
For $\ell = 1, \ldots, L + 1$
  For $k = 1, \ldots, N$
    $m_{\ell,k} \leftarrow R_\ell(\overrightarrow{\gamma_k})$
Apply Gaussian elimination on $\boldsymbol{M}$
If $\boldsymbol{M}$ is full-rank then
  Return $\perp$
Else
  Let $\overrightarrow{\lambda'}$ be the row vector such that $\overrightarrow{\lambda'} \cdot \boldsymbol{M} = \overrightarrow{0}$
  $\overrightarrow{\lambda} \leftarrow (\lambda'_1/\lambda'_{L+1}, \ldots, \lambda'_L/\lambda'_{L+1})$
  Return $\overrightarrow{\lambda}$

Fig. 1. TestLin procedure

negligible, where this advantage is defined via

$$\mathbf{Adv}_F^{\mathsf{prf}}(\mathscr{A}) = \Pr\left[\mathrm{PRFReal}_F^{\mathscr{A}} \Rightarrow 1\right] - \Pr\left[\mathrm{PRFRand}_F^{\mathscr{A}} \Rightarrow 1\right],$$

where games $\mathrm{PRFReal}_F$ and $\mathrm{PRFRand}_F$ are depicted in Fig. 2.

Aggregation Function. Let $f\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a function. We define an aggregation function by describing two objects:

- a collection $\mathscr{S}$ of subsets S of the domain $\mathcal{D}$;
- an aggregation function $\Gamma\colon \mathcal{R}^* \to \mathcal{V}$ that takes as input a tuple of values from the range $\mathcal{R}$ of F and aggregates them to produce a value in an output set $\mathcal{V}$.

In addition, we require the set ensemble $\mathscr{S}$ to be *efficiently recognizable*, meaning that for any $S \in \mathscr{S}$, there exists a polynomial time procedure to check if $x \in S$, for any $x \in \mathcal{D}$. Also, we require the aggregation function Γ to be polynomial time and the output of the function not to depend on the order of the elements provided as inputs. Finally, we require all sets S to have a representation of size polynomial in the security parameter κ.

Given an aggregation function $(\mathscr{S}, \Gamma)$, we define the aggregate function $\mathrm{AGG} = \mathrm{AGG}_{f,\mathscr{S},\Gamma}$ as the function that takes as input a set $S \in \mathscr{S}$ and outputs the aggregation of all values $f(x)$ for all $x \in S$. That is, $\mathrm{AGG}(S)$ outputs $\Gamma(f(x_1), \ldots, f(x_{|S|}))$, where $S = \{x_1, \ldots, x_{|S|}\}$. We will require the computation of AGG to be polynomial time (even if the input set S is exponentially large) if the function f provided is the pseudorandom function $F(K, \cdot)$ we consider, where K is some key.

Aggregate Pseudorandom Functions. Let $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a pseudorandom function and let $(\mathscr{S}, \Gamma)$ be an associated aggregation function. We say that F is an $(\mathscr{S}, \Gamma)$-aggregate pseudorandom function ($(\mathscr{S}, \Gamma)$-AGG-PRF) if the advantage of any adversary in attacking the AGG-PRF security of F is negligible, where this advantage is defined via

$$\mathbf{Adv}^{\mathsf{agg\text{-}prf}}_{F,\mathscr{S},\Gamma}(\mathscr{A}) = \Pr\left[\,\mathrm{AGGPRFReal}^{\mathscr{A}}_{F} \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{AGGPRFRand}^{\mathscr{A}}_{F} \Rightarrow 1\,\right],$$

where games $\mathrm{AGGPRFReal}_F$ and $\mathrm{AGGPRFRand}_F$ are depicted in Fig. 2. Game $\mathrm{AGGPRFRand}_F$ may not be polynomial-time, as $\mathrm{AGG}_{f,\mathscr{S},\Gamma}$ may not require to compute an exponential number of values $f(x)$. However, for all the aggregate PRFs that we consider, this game is statistically indistinguishable from a polynomial-time game, using the **TestLin** procedure, similarly to what is done in our new PLP security notion (see Section 3 and Fig. 3).

Multilinear Pseudorandom Functions. Multilinear pseudorandom functions are a variant of the standard notion of pseudorandom functions, which works with vector spaces. More precisely, a multilinear pseudorandom function $F\colon \mathcal{K} \times \mathcal{D} \to \mathcal{R}$, is an efficiently computable function with key space $\mathcal{K}$, domain $\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_n$ (a cartesian product of n vector spaces $\mathcal{D}_1, \ldots, \mathcal{D}_n$, for some integer n), range $\mathcal{R}$ which is a vector space, and which is indistinguishable from a random n-linear function with same domain and range. We say that F is a multilinear pseudorandom function (MPRF) if the advantage of any adversary in attacking the MPRF security of F is negligible, where this advantage is defined via

$$\mathbf{Adv}^{\mathsf{mprf}}_{F}(\mathscr{A}) = \Pr\left[\,\mathrm{MPRFReal}^{\mathscr{A}}_{F} \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{MPRFRand}^{\mathscr{A}}_{F} \Rightarrow 1\,\right],$$

where games $\mathrm{MPRFReal}_F$ and $\mathrm{MPRFRand}_F$ are depicted in Fig. 2. As explained in [13], Game $\mathrm{MPRFRand}_F$ can be implemented in polynomial time using a deterministic algorithm checking linearity of simple tensors [6]. Also, similarly to Game $\mathrm{AGGPRFRand}_F$, it is also possible to implement a polynomial-time game that is statistically indistinguishable from $\mathrm{MPRFRand}_F$ using **TestLin**.

Assumptions. Our main theorem is proven under the same MDDH assumption [14] introduced in [1] and termed $\mathcal{E}_{k,d}$-MDDH assumption. This MDDH assumption is defined by the matrix distribution $\mathcal{E}_{k,d}$ which samples matrices $\boldsymbol{\Gamma}$ as follows

$$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{A}^0 \cdot \boldsymbol{B} \\ \boldsymbol{A}^1 \cdot \boldsymbol{B} \\ \vdots \\ \boldsymbol{A}^d \cdot \boldsymbol{B} \end{pmatrix} \in \mathbb{Z}_p^{k(d+1)\times k} \quad \text{with } \boldsymbol{A}, \boldsymbol{B} \xleftarrow{\$} \mathbb{Z}_p^{k\times k}. \tag{1}$$

The advantage of an adversary $\mathscr{D}$ against the $\mathcal{E}_{k,d}$-MDDH assumption is

$$\mathbf{Adv}^{\mathcal{E}_{k,d}\text{-}\mathsf{mddh}}_{\mathbb{G}}(\mathscr{D}) = \Pr\left[\,\mathscr{D}(g, [\boldsymbol{\Gamma}], [\boldsymbol{\Gamma}\cdot\boldsymbol{W}])\,\right] - \Pr\left[\,\mathscr{D}(g, [\boldsymbol{\Gamma}], [\boldsymbol{U}])\,\right],$$

where $\boldsymbol{\Gamma} \xleftarrow{\$} \mathcal{E}_{k,d}$, $\boldsymbol{W} \xleftarrow{\$} \mathbb{Z}_p^{k\times 1}$, $\boldsymbol{U} \xleftarrow{\$} \mathbb{Z}_p^{k(d+1)\times 1}$. This assumption is random self-reducible, as any other MDDH assumption (we will make use of this property in the proof of our main theorem, and recall this property in the full version [3]).

$\mathrm{PRFReal}_F$	$\mathrm{PRFRand}_F$
proc Initialize	**proc Initialize**
$K \xleftarrow{\$} \mathcal{K}$	$f \xleftarrow{\$} \mathsf{Fun}(\mathcal{D}, \mathcal{R})$
proc Fn(x)	**proc Fn**(x)
Return $F(K, x)$	Return $f(x)$

$\mathrm{AGGPRFReal}_F$	$\mathrm{AGGPRFRand}_F$
proc Initialize	**proc Initialize**
$K \xleftarrow{\$} \mathcal{K}$	$f \xleftarrow{\$} \mathsf{Fun}(\mathcal{D}, \mathcal{R})$
proc Fn(x)	**proc Fn**(x)
Return $F(K, x)$	Return $f(x)$
proc AGG(S)	**proc AGG**(S)
Return $\mathrm{AGG}_{F(K,\cdot),\mathscr{S},\Gamma}(S)$	Return $\mathrm{AGG}_{f,\mathscr{S},\Gamma}(S)$

$\mathrm{MPRFReal}_F$	$\mathrm{MPRFRand}_F$
proc Initialize	**proc Initialize**
$K \xleftarrow{\$} \mathcal{K}$	$f \xleftarrow{\$} \mathsf{L}(\mathcal{D}_1 \otimes \cdots \otimes \mathcal{D}_n, \mathcal{R})$
proc Fn$(\vec{x})$	**proc Fn**$(\vec{x})$
Return $F(K, \vec{x})$	Return $f(\vec{x})$

Fig. 2. Security games for (classical, aggregate, multilinear — from top to bottom) pseudorandom functions

Table 1. Security of $\mathcal{E}_{k,d}$-MDDH

	$k=1$	$k=2$	$k\geq 3$
$d=1$	$=\mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}$	$\lesssim 2\cdot\mathbf{Adv}^{\mathcal{U}_2\text{-mddh}}_{\mathbb{G}}$	$\lesssim k\cdot\mathbf{Adv}^{\mathcal{U}_k\text{-mddh}}_{\mathbb{G}}$
$d\geq 2$	$\lesssim d\cdot\mathbf{Adv}^{d\text{-ddhi}}_{\mathbb{G}}$ ¶	generic bilinear group†	generic k-linear group‡

$\mathbf{Adv}^{\mathsf{ddh}}_{\mathbb{G}}$, $\mathbf{Adv}^{d\text{-ddhi}}_{\mathbb{G}}$ and $\mathbf{Adv}^{\mathcal{U}_k\text{-mddh}}_{\mathbb{G}}$ are advantages for DDH, DDHI, and $\mathcal{U}_k$-MDDH. This later assumption is weaker than k-Lin;
¶ proven in [1];
† proven in the generic (symmetric) bilinear group model [7] in [1];
‡ proven in the generic (symmetric) k-linear group model [19,16] in the full version [3].

In Table 1, we summarize security results for $\mathcal{E}_{k,d}$-MDDH. For $k=1$ or $d=1$, the $\mathcal{E}_{k,d}$-MDDH assumption is implied by standard assumptions (DDH, DDHI, or k-Lin, as recalled in the full version [3]). $\mathcal{E}_{1,1}$-MDDH is actually exactly DDH. In [1], the question of the hardness of the $\mathcal{E}_{k,d}$-MDDH problem in the generic k-linear group model was left as an open problem when $d>1$ and $k>2$. One of our contributions is to give a proof of hardness of these assumptions, which is detailed in the full version [3].

3 Polynomial Linear Pseudorandomness Security

As we already mentioned in the introduction, while the LIP theorem from [1] is quite powerful to prove the security of numerous constructions of pseudorandom functions (and related-key secure pseudorandom functions), it falls short when we need to prove the security of multilinear pseudorandom functions or aggregate pseudorandom functions. Indeed, the LIP theorem requires that there is no linear dependence between the outputs of the function. Thus, for the latter primitives, it is clear that one cannot use the LIP theorem, since the main point of these primitives is precisely that outputs can be related.

In order to deal with these primitives, we introduce a new security notion, termed polynomial linear pseudorandomness security (PLP), which encompasses the LIP security notion, but allows to handle multilinear pseudorandom functions and aggregate pseudorandom functions.

3.1 Intuition

Intuitively, the polynomial linear pseudorandomness security notion says that for any polynomials $P_1,\ldots,P_q\in\mathbb{Z}_p[T_1,\ldots,T_n]$, the group elements

$$[P_1(\vec{a})\cdot b],\ldots,[P_q(\vec{a})\cdot b],$$

with $\vec{a}\xleftarrow{\$}\mathbb{Z}_p^n$ and $b\xleftarrow{\$}\mathbb{Z}_p$, are computationally indistinguishable from the group elements:

$$[U(P_1)],\ldots,[U(P_q)],$$

with $U \xleftarrow{\$} \mathsf{L}(\mathbb{Z}_p[T_1, \ldots, T_n]_{\leq d}, \mathbb{Z}_p)$ being a random linear function from the polynomial vector space $\mathbb{Z}_p[T_1, \ldots, T_n]_{\leq d}$ (with d the maximum degree of $P_1, \ldots, P_q$ in any indeterminate T_i) to the base field $\mathbb{Z}_p$. Our main theorem (Theorem 1) shows that this security notion holds under the $\mathcal{E}_{1,d}$-MDDH assumption (and thus also under DDH for $d = 1$ and d-DDHI for $d \geq 2$).

When $P_1, \ldots, P_q$ are linearly independent, $[U(P_1)], \ldots, [U(P_q)]$ are independent random group elements in $\mathbb{G}$. In that sense, the polynomial linear pseudorandomness security notion is a generalization of the LIP security notion.

We remark that, in the generic group model, the polynomial linear pseudorandomness security notion holds trivially, by definition. The difficulty of the work is to prove it under classical assumptions such as the $\mathcal{E}_{1,d}$-MDDH assumption.

Polynomial-Time Games. When we want to formally define the polynomial linear pseudorandomness security notion, we quickly face a problem: how to compute $[U(P_i)]$ for a random linear map $U \xleftarrow{\$} \mathsf{L}(\mathbb{Z}_p[T_1, \ldots, T_n]_{\leq d}, \mathbb{Z}_p)$? Such a map can be represented by a (random) vector with $(d+1)^n$ entries. But doing so would make the game in the security notion exponential time. The idea is to define or draw U lazily: each time we need to evaluate it on a polynomial P_i linearly independent of all the previous polynomials P_j (with $j < i$), we define $U(P_i) \xleftarrow{\$} \mathbb{Z}_p$; otherwise, we compute $U(P_i)$ as a linear combination of $U(P_j)$. More precisely, if $P_i = \sum_{j=1}^{i-1} \lambda_j \cdot P_j$, $U(P_i) = \sum_{j=1}^{i-1} \lambda_j \cdot U(P_j)$. As explained in Section 2, no deterministic polynomial-time algorithm for checking linear dependency between polynomials in $\mathbb{Z}_p[T_1, \ldots, T_n]$ is known. But we can use one which is correct which overwhelming probability. We recall that we denote by **TestLin** such an algorithm.

On the Representation of the Polynomials. A second challenge is to define how the polynomials are represented. We cannot say they have to be given in their expanded form, because it would restrict us to polynomials with a polynomial number of monomials and forbid polynomials such as $\prod_{i=1}^{n}(a_i + 1)$.

Instead, we only suppose that polynomials can be (partially) evaluated, in polynomial time (in n and d, the maximum degree in each indeterminate). This encompasses polynomials defined by an expression (with $+$ and $\cdot$ operations, indeterminates, and scalars) of polynomial size (in n and d). Details are given in the full version [3].

Extension to Weaker Assumptions. Before, showing the formal definition and theorem, let us show an extension of our polynomial linear pseudorandomness security notion to handle weaker assumptions, namely $\mathcal{E}_{k,d}$-MDDH, with $k \geq 2$. In that case, we need to evaluate polynomials on matrices: $[P_i(\boldsymbol{A}) \cdot \boldsymbol{B}]$, with $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_p^{k \times k}$ and $\boldsymbol{B} \xleftarrow{\$} \mathbb{Z}_p^{k \times m}$ (with $m \geq 1$ being a positive integer). As multiplication of matrices is not commutative, we need to be very careful. We therefore consider that T_n appears before T_{n-1} (in products), T_{n-1} before T_{n-2}, ... (or any other fixed ordering).

More formally, we suppose that polynomials are represented by an expression (similar to the case $k = 1$), such that in any subexpression $Q \cdot R$, if Q contains T_i

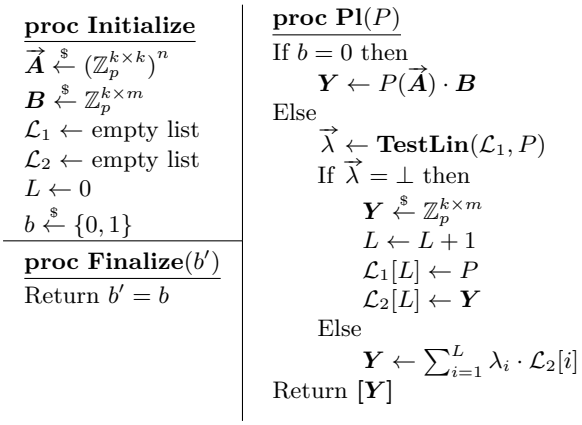

Fig. 3. Game defining the (n, d, k, m)-PLP security for a group $\mathbb{G}$

(formally as an expression and not just when the expression is expanded), then R contains no monomial T_j with $j > i$. Details are given in the full version [3].

3.2 Formal Security Notion and Theorem

Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p. We define the advantage of an adversary $\mathscr{A}$ against the (n, d, k, m)-PLP security of $\mathbb{G}$, denoted $\mathbf{Adv}_{\mathbb{G}}^{(n,d,k,m)\text{-plp}}(\mathscr{A})$ as the probability of success in the game defined in Fig. 3, with $\mathscr{A}$ being restricted to make queries $P \in \mathbb{Z}_p[T_1, \ldots, T_n]_{\leq d}$. When not specified, $m = 1$. When $k = m = 1$, we get exactly the intuitive security notion defined previously, as in that case $\vec{\boldsymbol{A}} = \vec{a} \in \mathbb{Z}_p^n$ and $\boldsymbol{B} = b \in \mathbb{Z}_p$.

Theorem 1 (PLP). *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p. Let $\mathscr{A}$ be an adversary against the (n, d, k, m)-PLP security of $\mathbb{G}$ that makes q oracle queries $P_1, \ldots, P_q$. Then we can design an adversary $\mathscr{B}$ against the $\mathcal{E}_{k,d}$-MDDH problem in $\mathbb{G}$, such that $\mathbf{Adv}_{\mathbb{G}}^{(n,d,k,m)\text{-plp}}(\mathscr{A}) \leq n \cdot d \cdot \mathbf{Adv}_{\mathbb{G}}^{\mathcal{E}_{k,d}\text{-mddh}}(\mathscr{B}) + O(ndqN/p)$, where N is an integer polynomial in the size of the representations of the polynomials and $N = 1$ when $k = 1$ (see the full version [3] for details). The running time of $\mathscr{B}$ is that of $\mathscr{A}$ plus the time to perform a polynomial number (in q, n, and d) of operations in $\mathbb{Z}_p$ and $\mathbb{G}$.*

The proof of Theorem 1 is detailed in the full version [3]. It is similar to the proof of the LIP theorem (in the matrix case) in [1]. More precisely, we show a series of indistinguishable games where the first game corresponds to the (n, d, k, m)-PLP security game when $b = 0$, and the last game corresponds to this security game when $b = 1$. Basically, all the games except for the last two games are the same as in the proof of the LIP theorem. The two last games differ, as follows: for the LIP theorem, all polynomials are supposed to be linearly independent, and so in the last two games, all the returned values are drawn

uniformly and independently, while for the PLP theorem, the returned values still have linear dependencies.

4 Applications

In this section, we describe how PLP theorem (Theorem 1) can be used to prove the security of aggregate pseudorandom functions as well as multilinear pseudorandom functions. In particular, we obtain polynomial-time reduction for all previous constructions of aggregate-pseudorandom, even for aggregate where only exponential-time reduction were known (read-once formulas). We also obtain a very simple proof of the multilinear pseudorandom function designed in [13]. Finally, we briefly explain how these results can be extended to build constructions based on weaker assumptions in an almost straightforward manner, by simply changing the key space. The proofs of security remain almost the same and consist in reducing the security to the adequate PLP security game.

4.1 Aggregate Pseudorandom Functions

In this subsection, we show that for all constructions proposed in [12], one can prove the AGG-PRF security with a polynomial time reduction, while proofs proposed in this seminal paper suffered from an exponential (in the input size) overhead in the running time of the reduction. Moreover, our reductions are almost straightforward via the PLP theorem.

A first attempt to solve the issue of the exponential time of the original reductions was done in [13]. By introducing multilinear pseudorandom functions and giving a particular instantiation, Cohen and Holmgren showed that one can prove the AGG-PRF security of NR with a polynomial time reduction for hypercubes and decision trees aggregation. However, their technique does not extend to the more general case of read-once formulas aggregation. Also, as we will show it the next subsection, their construction can be seen as a particular case of our main theorem, and then can be proven secure very easily using our result.

Here, we provide a polynomial time reduction for the general case of read-once formulas. This implies in particular the previous results on hypercubes and decision trees which are particular cases of read-once formulas.

Intuitively, if we consider the PLP security for $k = 1$ and aggregation with the Naor-Reingold PRF, our PLP theorem (Theorem 1) implicitly says that as long as the aggregate values can be computed as a group element whose discrete logarithm is the evaluation of a multivariate polynomial on the key, then, if the corresponding polynomials have a small representation, the PLP theorem guarantees the security (with a polynomial time reduction), even if the number of points aggregated is superpolynomial. Please notice that if these polynomials do not have any small representation (e.g. the smallest representation is exponential in the input size), then there is no point of considering such aggregation, since the whole

point of aggregate pseudorandom function lies in the possibility of aggregating superpolynomially many PRF values with a very efficient computation.

Read-Once Formulas. A read-once formula is a circuit on $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ composed of only AND, OR and NOT gates with fan-out 1, so that each input literal is fed into at most one gate and each gate output is fed into at most one other gate. We denote by ROF_n the family of all read-once boolean formulas over $x_1, \ldots, x_n$ variables. In order to ease the reading, we restrict these circuits to be in a standard form, so that they are composed of fan-in 2 and fan-out 1 AND and OR gates, and NOT gates occurring only at the inputs. This common restriction can be done without loss of generality. Hence, one can see such a circuit as a binary tree where each leaf is labeled by a variable x_i or its negation $\bar{x}_i$ and where each internal node has a label C and has two children with labels C_L and C_R and represents either an AND or an OR gate (with fan-in 2). We identify a formula (and the set it represents) with the label of its root C_ϕ.

Aggregation for Read-Once Formulas. We recall the definition of read-once formula aggregation used in [12]. For the sake of simplicity, we only consider the case of the Naor-Reingold PRF, defined as $\mathsf{NR}(\vec{a}, x) = [a_0 \prod_{i=1}^n a_i^{x_i}]$, where $a_0, \ldots, a_n \xleftarrow{\$} \mathbb{Z}_p$ and $x \in \{0,1\}^n$. We define the aggregation function for read-once formulas of length n as follows.

The collection $\mathscr{S}_{\mathsf{rof}} \subseteq \{0,1\}^n$ corresponds to all the subsets of $S \subseteq \{0,1\}^n$ such that there exists a read-once formula $C_\phi \in \mathrm{ROF}_n$ such that $S = \{x \in \{0,1\}^n \mid C_\phi(x) = 1\}$.

The aggregation function Γ_{rof} is defined as the product (assuming the group is a multiplicative group) of the values on such a subset. Hence, we have:

$$\begin{aligned}\mathrm{AGG}_{\mathsf{NR},\mathscr{S}_{\mathsf{rof}},\Gamma_{\mathsf{rof}}}(C_\phi) &= \prod_{x|C_\phi(x)=1} \left[a_0 \prod_{i=1}^n a_i^{x_i}\right] = \left[a_0 \sum_{x|C_\phi(x)=1} \prod_{i=1}^n a_i^{x_i}\right] \\ &= \left[a_0 \cdot A_{C_\phi,1}(\vec{a})\right] ,\end{aligned}$$

where $A_{C,b}$ is the polynomial $\sum_{x\in\{0,1\}^n|C(x)=b} \prod_{i=1}^n T_i^{x_i}$ for any $C \in \mathrm{ROF}_n$ and $b \in \{0,1\}$.

Efficient Evaluation of $A_{C,b}$. One can efficiently compute $A_{C,b}$ recursively as follows:

- If C is a literal for variable x_i, then $A_{C,1} = T_i$ and $A_{C,0} = 1$ if $C = x_i$; and $A_{C,1} = 1$ and $A_{C,0} = T_i$ if $C = \bar{x}_i$;
- If C is an AND gate with C_L and C_R its two children, then we have:
  $A_{C,1} = A_{C_L,1} \cdot A_{C_R,1}$
  $A_{C,0} = A_{C_L,0} \cdot A_{C_R,0} + A_{C_L,1} \cdot A_{C_R,0} + A_{C_L,0} \cdot A_{C_R,1}$;
- If C is an OR gate with C_L and C_R its two children, then we have:
  $A_{C,1} = A_{C_L,1} \cdot A_{C_R,1} + A_{C_L,1} \cdot A_{C_R,0} + A_{C_L,0} \cdot A_{C_R,1}$
  $A_{C,0} = A_{C_L,0} \cdot A_{C_R,0}$.

Now we have introduced everything, we can prove that NR (or more general constructions) is an $(\mathscr{S}_{\mathsf{rof}}, \Gamma_{\mathsf{rof}})$-AGG-PRF under the standard DDH assumption, as stated in the lemma below.

Lemma 2. *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and* NR *be the Naor-Reingold PRF defined as $\mathsf{NR}(\vec{a}, x) = [a_0 \prod_{i=1}^{n} a_i^{x_i}]$, where the key is $(a_0, \ldots, a_n) \xleftarrow{\$} \mathbb{Z}_p^{n+1}$ and the input is $x \in \{0,1\}^n$. Then one can reduce the $(\mathscr{S}_{\mathsf{rof}}, \Gamma_{\mathsf{rof}})$*-AGG-PRF *security of* NR *to the hardness of the* DDH *problem in $\mathbb{G}$, with a loss of a factor n. Moreover, the time overhead is* polynomial *in n and in the number of queries made by the adversary.*

The proof is straightforward using the PLP theorem: all queries in the security game for the aggregate PRF can be seen as a queries of the form $\mathbf{Pl}(P)$ for some polynomial P with a small representation: $\mathbf{Fn}(x)$ returns $\mathbf{Pl}(T_0 \prod_{i=1}^{n} T_i^{x_i})$ and $\mathrm{AGG}(C_\phi)$ returns $\mathbf{Pl}(T_0 \cdot A_{C_\phi,1}(\vec{T}))$. Details can be found in the full version [3].

Extensions. One can easily extend this result for k-Lin-based PRFs similar to NR using our main theorem. Also, one can easily use our PLP theorem (Theorem 1) to prove the security for any aggregate (for instance with NR) as soon as the aggregate values can be represented as group elements whose discrete logarithms are the evaluation of a (multivariate) polynomial on the key (and that this polynomial is efficiently computable).

Impossibility Result for CNF (Conjunctive Normal Form) and DNF (Disjunctive Normal Form) Formulas. In [12], the authors show that, unless NP=BPP, there does not exist an $(\mathscr{S}, \Gamma)$-aggregate pseudorandom function[1], with $\mathcal{D} = \{0,1\}^n$, $\mathscr{S}$ containing the following sets:

$$S_\phi = \{x \in \{0,1\}^n \mid \phi(x) = 1\}$$

with ϕ a CNF formula with n-bit input, and Γ a "reasonable" aggregate function, e.g., Γ_{rof} (assuming $\mathcal{R}$ is a cyclic group $\mathbb{G}$ of prime order p). The proof consists in showing that if such aggregate pseudorandom function exists, then we can solve SAT in polynomial time. More precisely, given a SAT instance, i.e., a CNF formula ϕ, we can compute $\mathrm{AGG}(\phi)$. If ϕ is not satisfiable, $\mathrm{AGG}(\phi) = 1 \in \mathbb{G}$, while otherwise $\mathrm{AGG}(\phi) = \prod_{x \in \{0,1\}^n,\, \phi(x)=1} F(K, x)$. This latter value is not 1 with high probability, otherwise we would get a non-uniform distinguisher against aggregate pseudorandomness.

The case of DNF formulas (or more generally of any class for which satisfiability is tractable) was left as an important open problem in [12]. Here, we show that unless NP=BPP, there also does not exist an $(\mathscr{S}, \Gamma)$-aggregate pseudorandom function as above, when $\mathscr{S}$ contains S_ϕ for any DNF (instead of CNF) formula ϕ with n-bit input. For that, we first remark that the formula $\top$, always true, is a DNF formula (it is the disjunction of all the possible literals), and that

[1] We suppose that the aggregate pseudorandomness security property holds non-uniformly. When $\mathscr{S}$ is expressive enough, we can also do the proof when this security property holds uniformly, see [12, Section 2.2] for details.

the negation $\bar{\phi}$ of a CNF formula ϕ is a DNF formula. Then, given a SAT instance, a CNF formula ϕ, we compute $\mathrm{AGG}(\bar{\phi})$ and $\mathrm{AGG}(\top)$. If ϕ is not satisfiable, $\bar{\phi}$ is always true and $\mathrm{AGG}(\bar{\phi}) = \mathrm{AGG}(\top)$, while otherwise, $\mathrm{AGG}(\bar{\phi}) = \mathrm{AGG}(\top)/\prod_{x\in\{0,1\}^n,\ \phi(x)=1} F(K,x)$. This latter value is not $\mathrm{AGG}(\top)$ with high probability, otherwise we would get a non-uniform distinguisher against aggregate pseudorandomness.

4.2 Multilinear Pseudorandom Functions

Here, we explain how our main theorem can be used to prove directly the security of the multilinear pseudorandom function built in [13]. We first recall their construction before explaining how to prove its security.

Cohen-Holmgren multilinear pseudorandom function (CH). Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p. The key space of the multilinear pseudorandom function is $\mathbb{Z}_p^{l_1} \times \cdots \times \mathbb{Z}_p^{l_n}$. The input space is the same as the key space. Given a key $(\overrightarrow{a_1}, \ldots, \overrightarrow{a_n})$ taken uniformly at random in the key space, the evaluation of the multilinear pseudorandom function on the input $(\overrightarrow{x_1}, \ldots, \overrightarrow{x_n})$) outputs:

$$\mathsf{CH}((\overrightarrow{a_1}, \ldots, \overrightarrow{a_n}), (\overrightarrow{x_1}, \ldots, \overrightarrow{x_n})) = \left[\prod_{i=1}^{n} \langle \overrightarrow{a_i}, \overrightarrow{x_i} \rangle\right]$$

where $\langle \overrightarrow{a}, \overrightarrow{x} \rangle$ denotes the canonical inner product $\langle \overrightarrow{a}, \overrightarrow{x} \rangle = \sum_{i=1}^{l} a_i \cdot x_i$, with l being the length of vectors $\overrightarrow{a}$ and $\overrightarrow{x}$.

In [13], Cohen and Holmgren prove that this construction is a secure multilinear pseudorandom function under the standard DDH assumption. One of their main contributions is to achieve a polynomial time reduction. Their technique can be seen as a special case of ours. In particular, using our main theorem, one can easily obtain the following lemma.

Lemma 3. *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and* CH*: $(\mathbb{Z}_p^{l_1} \times \cdots \times \mathbb{Z}_p^{l_n}) \times (\mathbb{Z}_p^{l_1} \times \cdots \times \mathbb{Z}_p^{l_n}) \to \mathbb{G}$ denote the above multilinear pseudorandom function. Then we can reduce the multilinear PRF security of* CH *to the hardness of the* DDH *problem in $\mathbb{G}$, with a loss of a factor $l = \sum_{i=1}^{n} l_i$. Moreover, the time overhead is* polynomial *in l and in the number of queries made by the adversary.*

A detailed proof can be found in the full version [3], but we give an intuition of the proof in what follows.

Proof. Let $\overrightarrow{T} = (T_{1,1}, \ldots, T_{1,l_1}, \ldots, T_{n,1}, \ldots, T_{n,l_n})$ be a vector of indeterminates, and let $\overrightarrow{T_i} = (T_{i,1}, \ldots, T_{i,l_i})$. The PLP theorem shows that $\mathsf{CH}(\overrightarrow{a_1}, \ldots, \overrightarrow{a_n}, \overrightarrow{x_1}, \ldots, \overrightarrow{x_n})$ (using a random key $\overrightarrow{a}$) is computationally indistinguishable from

$$\left[U\left(\prod_{i=1}^{n} \langle \overrightarrow{T_i}, \overrightarrow{x_i} \rangle\right)\right] = [f(\overrightarrow{x_1}, \ldots, \overrightarrow{x_n})]$$

with $U \xleftarrow{\$} \mathsf{L}(\mathbb{Z}_p[\overrightarrow{T}]_{\leq 1}, \mathbb{Z}_p)$ and

$$f : \begin{pmatrix} \mathbb{Z}_p^{l_1} \times \cdots \times \mathbb{Z}_p^{l_n} \to \mathbb{Z}_p \\ (\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}) \mapsto U(\prod_{i=1}^n \langle \overrightarrow{T_i}, \overrightarrow{x_i} \rangle) \end{pmatrix}.$$

To conclude, we just need to prove that f is a random n-linear function in $\mathsf{L}(\mathbb{Z}_p^{l_1} \otimes \cdots \otimes \mathbb{Z}_p^{l_n}, \mathbb{Z}_p)$.

For that purpose, let us introduce the following n-linear application:

$$\psi : \begin{pmatrix} \mathbb{Z}_p^{l_1} \times \cdots \times \mathbb{Z}_p^{l_n} \to \mathbb{Z}_p[\overrightarrow{T}]_{\leq 1} \\ (\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}) \mapsto \prod_{i=1}^n \langle \overrightarrow{T_i}, \overrightarrow{x_i} \rangle \end{pmatrix}.$$

We remark that f is the composition of U and ψ: $f = U \circ \psi$.

Furthermore, if we write $\overrightarrow{e_{i,l}} = (0, \ldots, 0, 1, 0, \ldots, 0)$ the i-th vector of the canonical base of $\mathbb{Z}_p^l$, then:

$$\psi(\overrightarrow{e_{i_1,l_1}}, \ldots, \overrightarrow{e_{i_n,l_n}}) = T_{1,i_1} \cdots T_{n,i_n};$$

and as the monomials $T_{1,i_1} \cdots T_{n,i_n}$ are linearly independent, ψ is injective. Since $f = U \circ \psi$ and $U \xleftarrow{\$} \mathsf{L}(\mathbb{Z}_p[\overrightarrow{T}]_{\leq 1}, \mathbb{Z}_p)$, the function f is a uniform random linear function from $\mathsf{L}(\mathbb{Z}_p^{l_1} \otimes \cdots \otimes \mathbb{Z}_p^{l_n}, \mathbb{Z}_p)$. This is exactly what we wanted to show. □

Symmetric Multilinear Pseudorandom Function. In [12], constructing symmetric multilinear pseudorandom functions was left as an open problem. The definition of this notion is the same as the notion of multilinear pseudorandom function, except that we only require the function to be indistinguishable from a random *symmetric* multilinear function. In that case, we suppose that $l_1 = \cdots = l_n = l$, i.e., all the vectors $\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}$ have the same size l. The authors wrote in [12] that the natural modification of the CH construction to obtain a symmetric construction consisting in setting $\overrightarrow{a_1} = \overrightarrow{a_2} = \cdots = \overrightarrow{a_n}$ (simply denoted $\overrightarrow{a}$ in what follows) leads to a symmetric multilinear pseudorandom function whose security is less clear, but claimed that it holds under the $\mathcal{E}_{1,n}$-MDDH assumption (which is exactly the n-Strong DDH assumption), when $l = |\overrightarrow{a}| = 2$. We show that this construction is actually secure under the same assumption for any $l = |\overrightarrow{a}| \geq 2$ as stated in the following lemma, whose proof is detailed in the full version [3] and is almost the same as the proof of Lemma 3.

Lemma 4. *Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p and $\mathsf{CH}_{\mathsf{sym}}$: $\mathbb{Z}_p^l \times (\mathbb{Z}_p^l)^n \to \mathbb{G}$ that takes as input a key $\overrightarrow{a} \in \mathbb{Z}_p^l$ and an input $\overrightarrow{x} = (\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}) \in (\mathbb{Z}_p^l)^n$ and outputs $[\prod_{i=1}^n \langle \overrightarrow{a}, \overrightarrow{x_i} \rangle]$. Then we can reduce the symmetric multilinear PRF security of $\mathsf{CH}_{\mathsf{sym}}$ to the hardness of the n-*DDHI *problem in $\mathbb{G}$, with a loss of a factor l. Moreover, the time overhead is* polynomial *in l and in the number of queries made by the adversary.*

Skew-Symmetric Multilinear Pseudorandom Function. In [12], the author left as an open problem the construction of a skew-symmetric multilinear

pseudorandom function. The definition of this notion is the same as the notion of multilinear pseudorandom function, except that we only require the function to be indistinguishable from a random *skew-symmetric* multilinear function. We assume that $l_1 = \cdots = l_n = l = n$, i.e., all the vectors $\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}$ have the same size $l = n$. We need $l = n$ because there is no skew-symmetric n-multilinear map from $\left(\mathbb{Z}_p^l\right)^n$ to $\mathbb{Z}_p$, when $l < n$.

We know that any skew-symmetric n-multilinear map f is of the form:

$$f(\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}) = c \cdot \det(\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}),$$

with c being a scalar in $\mathbb{Z}_p$ and det being the determinant function. Therefore, the function

$$F(a, (\overrightarrow{x_1}, \ldots, \overrightarrow{x_n})) = [a \cdot \det(\overrightarrow{x_1}, \ldots, \overrightarrow{x_n})]$$

is a skew-symmetric multilinear PRF with key $a \in \mathbb{Z}_p$. The proof is trivial since, $(\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}) \mapsto F(a, (\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}))$ is actually a random skew-symmetric n-multilinear map when a is a random scalar in $\mathbb{Z}_p$. No assumption is required. Our analysis shows that skew-symmetric multilinear PRFs are of limited interest, but our construction still solves an interesting open problem in [12].

Extensions. As for aggregate pseudorandom functions, it is very easy to build multilinear pseudorandom functions under k-Lin and to prove their security applying our PLP theorem (Theorem 1), for instance using the same construction but changing the key components from elements in $\mathbb{Z}_p$ to elements in $\mathbb{Z}_p^{k\times k}$ while keeping the same inputs space, and by defining $\langle \overrightarrow{\boldsymbol{A}}, \vec{x} \rangle = \sum_{i=1}^{l} x_i \cdot \boldsymbol{A_i}$, with $\overrightarrow{\boldsymbol{A}} = (\boldsymbol{A}_1, \ldots, \boldsymbol{A}_l) \in \left(\mathbb{Z}_p^{k\times k}\right)^l$ and $x = (x_1, \ldots, x_l) \in \mathbb{Z}_p^l$. This leads to the following construction:

$$F : \left(\begin{array}{l} \mathbb{Z}_p^{l_1} \times \cdots \times \mathbb{Z}_p^{l_n} \to \mathbb{G}^{k\times m} \\ \quad (\overrightarrow{x_1}, \ldots, \overrightarrow{x_n}) \mapsto \left[\left(\prod_{i=1}^{n} \langle \overrightarrow{\boldsymbol{A}_i}, \overrightarrow{x_i} \rangle \right) \cdot \boldsymbol{B} \right] \end{array} \right)$$

with $(\overrightarrow{\boldsymbol{A}_1}, \ldots, \overrightarrow{\boldsymbol{A}_n}) \in \left(\mathbb{Z}_p^{k\times k}\right)^{l_1} \times \cdots \times \left(\mathbb{Z}_p^{k\times k}\right)^{l_n}$ and $\boldsymbol{B} \in \mathbb{Z}_p^{k\times m}$.

References

1. Abdalla, M., Benhamouda, F., Passelègue, A.: An algebraic framework for pseudorandom functions and applications to related-key security. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 388–409. Springer, Heidelberg (Aug 2015)
2. Abdalla, M., Benhamouda, F., Passelègue, A., Paterson, K.G.: Related-key security for pseudorandom functions beyond the linear barrier. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 77–94. Springer, Heidelberg (Aug 2014)
3. Abdalla, M., Benhamouda, F., Passelègue, A.: Multilinear and aggregate pseudorandom functions: New constructions and improved security, full version of this paper available at Cryptology ePrint Archive, http://eprint.iacr.org

4. Advanced Encryption Standard (AES). National Institute of Standards and Technology (NIST), FIPS PUB 197, U.S. Department of Commerce (Nov 2001)
5. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (May / Jun 2006)
6. Bogdanov, A., Wee, H.: A stateful implementation of a random function supporting parity queries over hypercubes. In: Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D. (eds.) APPROX-RANDOM 2004. LNCS, vol. 3122, pp. 298–309. Springer, Heidelberg (Aug 2004)
7. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (May 2004)
8. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (May 2005)
9. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 280–300. Springer, Heidelberg (Dec 2013)
10. Boyen, X.: The uber-assumption family (invited talk). In: Galbraith, S.D., Paterson, K.G. (eds.) PAIRING 2008. LNCS, vol. 5209, pp. 39–56. Springer, Heidelberg (Sep 2008)
11. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 501–519. Springer, Heidelberg (Mar 2014)
12. Cohen, A., Goldwasser, S., Vaikuntanathan, V.: Aggregate pseudorandom functions and connections to learning. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 61–89. Springer, Heidelberg (Mar 2015)
13. Cohen, A., Holmgren, J.: Multilinear pseudorandom functions. In: Halldórsson, M.M., Iwama, K., Kobayashi, N., Speckmann, B. (eds.) ICALP 2015, Part I. LNCS, vol. 9134, pp. 331–342. Springer, Heidelberg (Jul 2015)
14. 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, Heidelberg (Aug 2013)
15. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM 33(4), 792–807 (Oct 1986)
16. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (Aug 2007)
17. Kiayias, A., Papadopoulos, S., Triandopoulos, N., Zacharias, T.: Delegatable pseudorandom functions and applications. In: Sadeghi, A.R., Gligor, V.D., Yung, M. (eds.) ACM CCS 13. pp. 669–684. ACM Press (Nov 2013)
18. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: 38th FOCS. pp. 458–467. IEEE Computer Society Press (Oct 1997)
19. 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/2007/074