Pseudo-Free Families of Finite Computational Elementary Abelian p-Groups

Mikhail Anokhin

Information Security Institute,
Lomonosov University, Moscow, Russia
anokhin@mccme.ru

Abstract

We initiate the study of (weakly) pseudo-free families of computational elementary abelian p-groups, where p is an arbitrary fixed prime. We restrict ourselves to families of computational elementary abelian p-groups G_d such that for every index d, each element of G_d is represented by a single bit string of length polynomial in the length of d.

First, we prove that pseudo-freeness and weak pseudo-freeness for families of computational elementary abelian p-groups are equivalent. Second, we give some necessary and sufficient conditions for a family of computational elementary abelian p-groups to be pseudo-free (provided that at least one of two additional conditions holds). These necessary and sufficient conditions are formulated in terms of collision-intractability or one-wayness of certain homomorphic families of knapsack functions. Third, we establish some necessary and sufficient conditions for the existence of pseudo-free families of computational elementary abelian p-groups. With one exception, these conditions are the existence of certain homomorphic collision-intractable families of p-ary hash functions or certain homomorphic one-way families of functions.

As an example, we construct a Diffie-Hellman-like key agreement protocol from an arbitrary family of computational elementary abelian p-groups. Unfortunately, we do not know whether this protocol is secure under reasonable assumptions.

Keywords: Family of computational groups, pseudo-free family of computational groups, weakly pseudo-free family of computational groups, elementary abelian p-group, homomorphic family of functions, collision-intractable family of functions, one-way family of functions, family of p-ary hash functions.

Contents

This is a preliminary version of the paper (with the same title) published in *Groups — Complexity — Cryptology*, 9(1):1–18, May 2017.

1 Introduction

Informally, a family of computational groups is a family of groups whose elements are represented by bit strings in such a way that equality testing, multiplication, inversion, computing the identity element, and sampling random elements can be performed efficiently. Loosely speaking, a family of computational groups is called pseudo-free if, given a random group G in the family (for an arbitrary value of the security parameter) and random elements $g_1, \ldots, g_m \in G$, it is computationally hard to find a system of group equations

$$v_i(a_1, \ldots, a_m; x_1, \ldots, x_n) = w_i(a_1, \ldots, a_m; x_1, \ldots, x_n), \quad i = 1, \ldots, s, \tag{1.1}$$

and elements $h_1, \ldots, h_n \in G$ such that (1.1) is unsatisfiable in the free group freely generated by $a_1, \ldots, a_m$ (over variables $x_1, \ldots, x_n$), but

$$v_i(g_1, \ldots, g_m; h_1, \ldots, h_n) = w_i(g_1, \ldots, g_m; h_1, \ldots, h_n)$$

in G for all $i \in \{1, \ldots, s\}$. If a family of computational groups satisfies this definition with the additional requirement that $n = 0$ (i.e., that the equations in (1.1) be variable-free), then this family is said to be weakly pseudo-free. Of course, (weak) pseudo-freeness depends heavily on the form in which system (1.1) is required to be found, i.e., on the representation of such systems.

The notion of pseudo-freeness (which is a variant of weak pseudo-freeness in the above sense) was introduced by Hohenberger in [Hoh03, Section 4.5] (for black-box groups). Rivest gave formal definitions of a pseudo-free family of computational groups (see [Riv04a, Definition 2], [Riv04b, Slide 17]) and a weakly pseudo-free one (see [Riv04b, Slide 11]). Note that the definitions of (weak) pseudo-freeness in those works are based on single group equations rather than systems of group equations. For motivation of the study of pseudo-freeness, we refer the reader to [Hoh03, Riv04a, Mic10]. Also, the above cited works contain definitions of (weak) pseudo-freeness in the variety $\mathfrak{A}$ of all abelian groups (using different terminology). (A *variety* of groups can be defined as a class of groups that is closed under taking subgroups, homomorphic images, and cartesian products. In particular, any variety of groups contains the trivial group because this group is the cartesian product of the empty family of groups.) Note that most works on pseudo-free families of computational groups deal with pseudo-freeness in $\mathfrak{A}$. To define a (weakly) pseudo-free family in $\mathfrak{A}$, it is natural to require that all groups in the family be abelian and to replace the free group by the free abelian group in the above definition of a (weakly) pseudo-free family. Similarly, we can define a (weakly) pseudo-free family in an arbitrary variety $\mathfrak{V}$ of groups. To do this, we require that all groups in the family belong to $\mathfrak{V}$ and replace the free group by the $\mathfrak{V}$-free group in the above definition of a (weakly) pseudo-free family. See [Ano13, Definition 3.3] for a formal definition of a pseudo-free family of computational groups in an arbitrary variety of groups. Of course, pseudo-free families of computational groups in different varieties are completely different objects. A survey of results concerning pseudo-freeness can be found in [Fuk14, Chapter 1].

In this paper, we study (weakly) pseudo-free families of computational groups in the variety of all elementary abelian p-groups, where p is an arbitrary fixed prime number. We call these families (weakly) pseudo-free families of computational elementary abelian p-groups. Note that we restrict ourselves to families $(G_d \,|\, d \in D)$ of computational elementary abelian p-groups (where $D \subseteq \{0,1\}^*$) such that for every $d \in D$, each element of G_d is represented by a single bit string of length polynomial in $|d|$. Hence we can assume that $G_d \subseteq \{0,1\}^{\leq \eta(|d|)}$ for some polynomial η and that the representation of each element $g \in G_d$ is g itself (see Definition 3.1).

Let $(H_i \,|\, i \in I)$ (where $I \subseteq \{0,1\}^*$) be a weakly pseudo-free family of finite computational groups in an arbitrary variety of infinite exponent (or, in another terminology, of exponent zero), e.g., in the

variety of all groups or all abelian groups. (The *exponent* of a variety $\mathfrak{V}$ of groups is equal to the order of a free generator of the $\mathfrak{V}$-free group.) Assume that, given a positive integer n, a representation of the variable-free equation $a_1^n = 1$ can be computed in polynomial time. Then it is easy to prove that the problem of finding $|H_i|$ for a given $i \in I$ is computationally hard (see [Riv04a, Subsection 4.1] or [Riv04b, Slide 12] for a guideline). It can be expected that this does not necessarily hold for (weakly) pseudo-free families of finite computational groups in varieties of finite exponent (provided that such families exist). In particular, this applies to (weakly) pseudo-free families of computational elementary abelian p-groups (see Corollary 4.8 and Remark 4.9). Note that the problem of extending the theory of pseudo-freeness to families of computational groups of easily computable order was posed by Rivest (see [Riv04a, Section 7], [Riv04b, Slide 22]).

The main contributions of this paper are as follows:

- The equivalence of pseudo-freeness and weak pseudo-freeness for families of computational elementary abelian p-groups (see Theorem 3.7). This enables us to use the definition of weak pseudo-freeness (which is more convenient for our purposes than the definition of pseudo-freeness) for proving results concerning pseudo-freeness. Bearing in mind this equivalence, we do not use the terms "weakly pseudo-free" and "weak pseudo-freeness" when speaking of families of computational elementary abelian p-groups after the proof of Theorem 3.7.

- Some necessary and sufficient conditions for a family Γ of computational elementary abelian p-groups to be pseudo-free, provided that at least one of two additional conditions holds (see Theorem 4.11). These necessary and sufficient conditions are formulated in terms of collision-intractability or one-wayness of certain homomorphic families $\mathrm{Kn}_\Gamma^\gamma$ of functions, where γ is a polynomial parameter. See Subsection 4.1 for the definition of these families of functions.

- Some necessary and sufficient conditions for the existence of pseudo-free families of computational elementary abelian p-groups (see Theorem 4.12). With one exception, these conditions are the existence of certain homomorphic collision-intractable families of p-ary hash functions or certain homomorphic one-way families of functions.

In Subsection 4.4, we construct a Diffie-Hellman-like key agreement protocol from an arbitrary family of computational elementary abelian p-groups. Also, the protocol uses a polynomial parameter ρ. Unfortunately, we do not know whether this protocol is secure (in some natural sense) under reasonable assumptions on the underlying family of computational elementary abelian p-groups and the polynomial parameter ρ. We leave this for further research (see Problem 5.3).

The rest of the paper is organized as follows. Section 2 contains notation, basic definitions, and general results used in the paper. In Section 3, we formally define and discuss families of computational elementary abelian p-groups (with the above restrictions), as well as pseudo-free and weakly pseudo-free ones. Also, Section 3 contains the proof of equivalence of pseudo-freeness and weak pseudo-freeness for families of computational elementary abelian p-groups. In Section 4, we give some necessary and sufficient conditions for pseudo-freeness and for the existence of pseudo-free families of computational elementary abelian p-groups. Finally, Section 5 contains some problems concerning families of computational elementary abelian p-groups. We suggest these problems for further research.

2 Preliminaries

2.1 General Preliminaries

In this paper, $\mathbb{N}$ denotes the set of all nonnegative integers. Let $m, n \in \mathbb{N}$. For a set X, we denote by X^n the set of all (ordered) n-tuples of elements from X and by $X^{m\times n}$ the set of all $m \times n$ matrices over X. When necessary, we consider tuples as matrices with one row. As usual, $(x_{i,j})$ denotes the $m \times n$ matrix (for some specified m and n) whose (i, j) entry is $x_{i,j}$ for all $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. The transpose of a matrix M is denoted by M^T.

We consider elements of $\{0,1\}^n$ as bit strings of length n. Furthermore, let $\{0,1\}^{\leq n} = \bigcup_{i=0}^n \{0,1\}^i$ and $\{0,1\}^* = \bigcup_{i=0}^\infty \{0,1\}^i$. If $u, v \in \{0,1\}^*$, then we denote by $|u|$ the length of u and by uv the concatenation of u and v. The unary representation of n, i.e., the string of n ones, is denoted by 1^n. Similarly, 0^n denotes the string of n zeros.

Let I be a set. Suppose each $i \in I$ is assigned an object q_i. Then we denote by $(q_i \mid i \in I)$ the family of all such objects and by $\{q_i \mid i \in I\}$ the set of all elements of this family.

When necessary, we assume that all "finite" objects (e.g., integers, tuples of integers, tuples of tuples of integers) are represented by bit strings in some natural way. Sometimes we identify such objects with their representations. Unless otherwise specified, integers are represented by their binary expansions.

Throughout the paper, p denotes an arbitrary fixed prime number. Also, we denote by $\mathbb{Z}_p$ the set $\{0, \dots, p-1\}$. If necessary, this set is considered as a field under addition and multiplication modulo p or as the additive group of this field. The intended meaning will be clear from the context.

In this paper, we deal with elementary abelian p-groups. Recall that a group G is called an *elementary abelian p-group* if G is abelian and $pg = 0$ for any $g \in G$. (We use additive notation for abelian groups.) In fact, elementary abelian p-groups are the same as vector spaces over the field $\mathbb{Z}_p$. Therefore a group is an elementary abelian p-group if and only if it is isomorphic to a direct power of the additive group of this field. For any $n \in \mathbb{N}$ and any group G, G^n denotes the nth direct power of G. If S is a system of elements of a group, then we denote by $\langle S \rangle$ the subgroup of this group generated by S.

For convenience, we say that a function $\pi \colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ is a *polynomial* if there exist $c \in \mathbb{N} \setminus \{0\}$ and $d \in \mathbb{N}$ such that $\pi(n) = cn^d$ for any $n \in \mathbb{N} \setminus \{0\}$ ($\pi(0)$ can be an arbitrary positive integer).

2.2 Probabilistic Preliminaries

Let $\mathcal{X}$ be a probability distribution on a finite or countably infinite sample space X. Then we denote by $\operatorname{supp} \mathcal{X}$ the *support* of $\mathcal{X}$, i.e., the set $\{x \in X \mid \Pr_{\mathcal{X}}\{x\} \neq 0\}$. In many cases, one can consider $\mathcal{X}$ as a distribution on $\operatorname{supp} \mathcal{X}$. Suppose α is a function from X to a finite or countably infinite set Y. Then α can be considered as a random variable. The distribution of this random variable is denoted by $\alpha(\mathcal{X})$. Recall that this distribution is defined by $\Pr_{\alpha(\mathcal{X})}\{y\} = \Pr_{\mathcal{X}} \alpha^{-1}(y)$ for each $y \in Y$.

We use the notation $\mathbf{x}_1, \dots, \mathbf{x}_n \leftarrow \mathcal{X}$ to indicate that $\mathbf{x}_1, \dots, \mathbf{x}_n$ (denoted by upright bold letters) are independent random variables distributed according to $\mathcal{X}$. We assume that these random variables are independent of all other random variables defined in such a way. Furthermore, all occurrences of an upright bold letter (possibly indexed or primed) in a probabilistic statement refer to the same (unique) random variable. Of course, all random variables in a probabilistic statement are assumed to be defined on the same sample space. Other specifics of random variables do not matter for us. Note that the probability distribution $\mathcal{X}$ in this notation can be random. For example, suppose $(\mathcal{X}_i \mid i \in I)$ is a probability ensemble consisting of distributions on the set X, where the set I is also finite or countably infinite. Moreover, let $\mathcal{I}$ be a probability distribution on I. Then $\mathbf{i} \leftarrow \mathcal{I}$ and $\mathbf{x} \leftarrow \mathcal{X}_{\mathbf{i}}$ mean that the joint distribution of the random variables $\mathbf{i}$ and $\mathbf{x}$ is given by $\Pr[\mathbf{i} = i, \mathbf{x} = x] = \Pr_{\mathcal{I}}\{i\} \Pr_{\mathcal{X}_i}\{x\}$ for each $i \in I$ and $x \in X$.

For any $n \in \mathbb{N}$, we denote by $\mathcal{X}^n$ the distribution of $(\mathbf{x}_1, \dots, \mathbf{x}_n)$, where $\mathbf{x}_1, \dots, \mathbf{x}_n \leftarrow \mathcal{X}$. Similarly, for arbitrary $m, n \in \mathbb{N}$, $\mathcal{X}^{m \times n}$ denotes the distribution of $(\mathbf{x}_{i,j})$, where $\mathbf{x}_{i,j} \leftarrow \mathcal{X}$ for all $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$.

The notation $\mathrm{x}_1, \dots, \mathrm{x}_n \leftarrow \mathcal{X}$ indicates that $\mathrm{x}_1, \dots, \mathrm{x}_n$ (denoted by upright medium-weight letters) are fixed elements of the set X chosen independently at random according to the distribution $\mathcal{X}$.

Let $\mathcal{R}$ and $\mathcal{S}$ be probability distributions on the set X. Then the *statistical distance* (also known as *variation distance*) between $\mathcal{R}$ and $\mathcal{S}$ is defined as

$$\Delta(\mathcal{R}, \mathcal{S}) = \frac{1}{2} \sum_{x \in X} |\Pr_{\mathcal{R}}\{x\} - \Pr_{\mathcal{S}}\{x\}|.$$

It is well known that $\Delta(\mathcal{R}, \mathcal{S}) = \max_{M \subseteq X} |\Pr_{\mathcal{R}} M - \Pr_{\mathcal{S}} M|$. See also, e.g., [Sho08, Section 8.8] or [Lub96, Lecture 7].

For a nonempty finite set Z, we denote by $\mathcal{U}(Z)$ the uniform probability distribution on Z.

2.3 Computational and Cryptographic Preliminaries

We need to generate random elements $\mathrm{y} \leftarrow \mathcal{U}(\mathbb{Z}_p)$. But if $p \neq 2$, then there is no probabilistic bounded-time algorithm (in the standard sense) that does this (see [Sho08, Exercise 9.4]). For this reason, we slightly modify the standard definition of a probabilistic algorithm. The only modification we make to this definition is allowing a probabilistic algorithm to use random elements $\mathrm{y} \leftarrow \mathcal{U}(\mathbb{Z}_p)$ instead of random bits $\mathrm{b} \leftarrow \mathcal{U}(\{0,1\})$. (Recall that p is fixed.) Unless otherwise specified, probabilistic algorithms considered

in this paper use random elements of $\mathbb{Z}_p$. Note that there exists a probabilistic polynomial-time algorithm A such that A uses random bits and for any $n \in \mathbb{N}$ the statistical distance between the distribution of $A(1^n)$ and $\mathcal{U}(\mathbb{Z}_p)$ is at most 2^{-n} (see Algorithm RN′ in [Sho08, Section 9.2]). Similarly, it is easy to see that there exists a probabilistic polynomial-time algorithm B such that B uses random elements of $\mathbb{Z}_p$ and for any $n \in \mathbb{N}$ the statistical distance between the distribution of $B(1^n)$ and $\mathcal{U}(\{0,1\})$ is at most p^{-n}. This shows that the computational power of probabilistic polynomial-time algorithms using random elements of $\mathbb{Z}_p$ is almost the same as that of such algorithms using random bits.

Let $\mathcal{X} = (\mathcal{X}_i \,|\, i \in I)$ be a probability ensemble consisting of distributions on $\{0,1\}^*$, where $I \subseteq \{0,1\}^*$ or $I \subseteq \mathbb{N}$. Then $\mathcal{X}$ is called *polynomial-time samplable* (or *polynomial-time constructible*) if there exists a probabilistic polynomial-time algorithm A such that for every $i \in I$ the distribution of $A(i)$ (if $I \subseteq \{0,1\}^*$) or $A(1^i)$ (if $I \subseteq \mathbb{N}$) coincides with $\mathcal{X}_i$. It is evident that if $\mathcal{X}$ is polynomial-time samplable, then there exists a polynomial π satisfying $\operatorname{supp} \mathcal{X}_i \subseteq \{0,1\}^{\leq \pi(|i|)}$ (if $I \subseteq \{0,1\}^*$) or $\operatorname{supp} \mathcal{X}_i \subseteq \{0,1\}^{\leq \pi(i)}$ (if $I \subseteq \mathbb{N}$) for any $i \in I$.

Suppose K is an infinite set of nonnegative integers, D is a subset of $\{0,1\}^*$, and $\mathcal{D} = (\mathcal{D}_k \,|\, k \in K)$ is a polynomial-time samplable probability ensemble consisting of distributions on D. Let $\mathcal{E}_k$ be the distribution of $(1^k, \mathbf{d})$, where $k \in K$ and $\mathbf{d} \leftarrow \mathcal{D}_k$, and let $\mathcal{E} = (\mathcal{E}_k \,|\, k \in K)$. Then $\mathcal{E}$ is a polynomial-time samplable probability ensemble. Also, denote $\bigcup_{k \in K} \operatorname{supp} \mathcal{E}_k$ by E. That is, $E = \{(1^k, d) \,|\, k \in K,\, d \in \operatorname{supp} \mathcal{D}_k\}$. This notation is used throughout the paper.

A function $\epsilon \colon K \to \{r \in \mathbb{R} \,|\, r \geq 0\}$ is called *negligible* if for every polynomial π there exists a nonnegative integer n such that $\epsilon(k) \leq 1/\pi(k)$ whenever $k \in K$ and $k \geq n$. We denote by negl an unspecified negligible function on K. Any (in)equality containing $\operatorname{negl}(k)$ is meant to hold for all $k \in K$.

Let $(\mathbf{r}_k \,|\, k \in K)$ and $(\mathbf{s}_k \,|\, k \in K)$ be probability ensembles consisting of random variables that take values in $\{0,1\}^*$. Then these ensembles are said to be *computationally indistinguishable* (or *indistinguishable in polynomial time*) if for any probabilistic polynomial-time algorithm A, $|\Pr[A(1^k, \mathbf{r}_k) = 1] - \Pr[A(1^k, \mathbf{s}_k) = 1]| = \operatorname{negl}(k)$. Furthermore, two probability ensembles $(\mathcal{R}_k \,|\, k \in K)$ and $(\mathcal{S}_k \,|\, k \in K)$ consisting of distributions on $\{0,1\}^*$ are said to be *computationally indistinguishable* (or *indistinguishable in polynomial time*) if $(\mathbf{r}_k \,|\, k \in K)$ and $(\mathbf{s}_k \,|\, k \in K)$ are computationally indistinguishable, where $\mathbf{r}_k \leftarrow \mathcal{R}_k$ and $\mathbf{s}_k \leftarrow \mathcal{S}_k$ for all $k \in K$.

Definition 2.1. Suppose $(H_d \,|\, d \in D)$ is a family of groups. We call a family $(\phi_d \colon H_d \to \{0,1\}^* \,|\, d \in D)$ of functions *homomorphic* if the following two conditions hold:

(i) For any $d \in D$, the operation $\circ_d$ in $\phi_d(H_d)$ given by $\phi_d(y) \circ_d \phi_d(z) = \phi_d(yz)$, where $y, z \in H_d$, is well defined. (Hence for every $d \in D$, $\phi_d(H_d)$ is a group under $\circ_d$ and ϕ_d is a homomorphism from H_d onto this group.)

(ii) The functions $(d, v, w) \mapsto v \circ_d w$, $(d, v) \mapsto v^{-1}$ (in the group $\phi_d(H_d)$), and $d \mapsto \phi_d(1)$, where $d \in D$ and $v, w \in \phi_d(H_d)$, are polynomial-time computable. (Here, of course, $\phi_d(1)$ is the identity element of the group $\phi_d(H_d)$.)

Let $d \in D$. It is evident that if ϕ_d is a homomorphism from H_d to a group $G_d \subseteq \{0,1\}^*$, then $\circ_d$ is well defined and coincides with the restriction of the multiplication in G_d to the subgroup $\phi_d(H_d)$. Also, it is easy to see that the operation $\circ_d$ is well defined if and only if $\phi_d^{-1}(\phi_d(1))$ is a normal subgroup of H_d and $\phi_d^{-1}(\phi_d(y)) = y\phi_d^{-1}(\phi_d(1))$ for all $y \in H_d$. In particular, this holds if ϕ_d is one-to-one.

We emphasize that a homomorphic family of functions does not just consist of group homomorphisms. It is also required that multiplication, inversion, and computing the identity element in the group $\phi_d(H_d)$ can be performed in polynomial time when d is given. Note that we use the term "homomorphic family of functions" by analogy with the term "homomorphic encryption."

We will use Definition 2.1 in the case when H_d is an elementary abelian p-group for each $d \in D$. In this case, of course, we will switch to additive notation. It is evident that if H_d is an elementary abelian p-group and $\circ_d$ is well defined, then $\phi_d(H_d)$ is an elementary abelian p-group under $\circ_d$ as a homomorphic image of H_d.

Definition 2.2 (see also [Lub96, Preliminaries])**.** A function $\rho \colon D \to \mathbb{N}$ is called a *polynomial parameter* (on D) if the function $d \mapsto 1^{\rho(d)}$ $(d \in D)$ is polynomial-time computable. It is easy to see that the function ρ is a polynomial parameter if and only if it is polynomial-time computable and there exists a polynomial π satisfying $\rho(d) \leq \pi(|d|)$ for all $d \in D$. A function $\eta \colon I \to \mathbb{N}$, where $I \subseteq \mathbb{N}$, is said to be a *polynomial parameter* (on I) if the function $1^i \mapsto \eta(i)$ $(i \in I)$ is a polynomial parameter on the set $\{1^i \,|\, i \in I\}$ in the above sense.

Note that the restriction of any polynomial to a set $I \subseteq \mathbb{N}$ is a polynomial parameter on I.

Example 2.3. We will use the following types of polynomial parameters on E:

- $(1^k, d) \mapsto \eta(k)$, where η is a polynomial parameter on K (in particular, a polynomial restricted to K).
- $(1^k, d) \mapsto \rho(d)$, where ρ is a polynomial parameter on D.

Remark 2.4. Suppose $(\mathcal{R}_d \,|\, d \in D)$ and $(\mathcal{S}_d \,|\, d \in D)$ are polynomial-time samplable probability ensembles consisting of distributions on $\{0,1\}^*$. Let σ be a polynomial parameter on E and let, for $k \in K$, $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{r}_0, \ldots, \mathbf{r}_{\sigma(1^k,\mathbf{d})} \leftarrow \mathcal{R}_\mathbf{d}$, and $\mathbf{s}_0, \ldots, \mathbf{s}_{\sigma(1^k,\mathbf{d})} \leftarrow \mathcal{S}_\mathbf{d}$. Assume that the probability ensembles $((\mathbf{d}, \mathbf{r}_0) \,|\, k \in K)$ and $((\mathbf{d}, \mathbf{s}_0) \,|\, k \in K)$ are computationally indistinguishable. A standard hybrid argument (see [Gol01, proof of Theorem 3.2.6]) shows that the probability ensembles $((\mathbf{d}, \mathbf{r}_1, \ldots, \mathbf{r}_{\sigma(1^k,\mathbf{d})}) \,|\, k \in K)$ and $((\mathbf{d}, \mathbf{s}_1, \ldots, \mathbf{s}_{\sigma(1^k,\mathbf{d})}) \,|\, k \in K)$ are computationally indistinguishable. (It suffices to prove this in the case when $\sigma(1^k, d) = \pi(k)$ for all $(1^k, d) \in E$, where $\pi\colon K \to \{p^l \,|\, l \in \mathbb{N}\}$ is a polynomial parameter.)

Let $\Phi = (\phi_d\colon Y_d \to \{0,1\}^* \,|\, d \in D)$ be a family of functions such that there exists a polynomial η satisfying $Y_d \subseteq \{0,1\}^{\leq \eta(|d|)}$ for all $d \in D$. Recall that the family Φ is called *polynomial-time computable* if the function $(d, y) \mapsto \phi_d(y)$ (where $d \in D$ and $y \in Y_d$) is polynomial-time computable. Moreover, recall that a *collision* for a function ϕ is a pair of distinct elements in its domain having the same image under ϕ.

Definition 2.5. The family Φ is called *collision-intractable* (or *collision-resistant*) *with respect to* $\mathcal{D}$ if for any probabilistic polynomial-time algorithm A, $\Pr[A(1^k, \mathbf{d}) \text{ is a collision for } \phi_\mathbf{d}] = \mathrm{negl}(k)$, where $\mathbf{d} \leftarrow \mathcal{D}_k$.

In particular, if ϕ_d is one-to-one for each $d \in D$, then Φ is collision-intractable with respect to $\mathcal{D}$.

Definition 2.6. Suppose $(\mathcal{Y}_d \,|\, d \in D)$ is a polynomial-time samplable probability ensemble, where $\mathcal{Y}_d$ is a probability distribution on Y_d for any $d \in D$. Then the family Φ is said to be *one-way with respect to* $\mathcal{D}$ *and* $(\mathcal{Y}_d \,|\, d \in D)$ if it is polynomial-time computable and for any probabilistic polynomial-time algorithm A, $\Pr[A(1^k, \mathbf{d}, \phi_\mathbf{d}(\mathbf{y})) \in \phi_\mathbf{d}^{-1}(\phi_\mathbf{d}(\mathbf{y}))] = \mathrm{negl}(k)$, where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{y} \leftarrow \mathcal{Y}_\mathbf{d}$.

We use the term "one-way family of functions" instead of the more common term "family of one-way functions" because one-wayness is a property of the whole family of functions rather than of its individual members. For the same reason, we use the terms "homomorphic family of functions" and "collision-intractable family of functions."

The next lemma is well known.

Lemma 2.7. *Suppose the family Φ is polynomial-time computable and collision-intractable with respect to $\mathcal{D}$. Also, assume that the following conditions hold:*

- *$Y_d \neq \emptyset$ for all $d \in D$.*
- *The probability ensemble $(\mathcal{U}(Y_d) \,|\, d \in D)$ is polynomial-time samplable.*
- *For $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{y} \leftarrow \mathcal{U}(Y_\mathbf{d})$, where $k \in K$, we have $\Pr[\phi_\mathbf{d}^{-1}(\phi_\mathbf{d}(\mathbf{y})) = \{\mathbf{y}\}] = \mathrm{negl}(k)$.*

Then the family Φ is one-way with respect to $\mathcal{D}$ and $(\mathcal{U}(Y_d) \,|\, d \in D)$.

Lemma 2.7 can be proved using an argument similar to that used in the proof of Proposition 8.4 in [GB08] (see also [GB08, Proposition 8.2]). We will apply Lemma 2.7 to families Φ consisting of group homomorphisms that are not one-to-one. It is evident that if ϕ is such a homomorphism defined on a group Y, then $\phi^{-1}(\phi(y)) \neq \{y\}$ for all $y \in Y$.

We need the following variant of the well-known Goldreich-Levin theorem for $\mathbb{Z}_p$.

Lemma 2.8 (follows from [DGK+10, Theorem 1])**.** *Suppose $(\psi_d\colon \mathbb{Z}_p^{\rho(d)} \to \{0,1\}^* \,|\, d \in D)$ is a one-way family of functions with respect to $\mathcal{D}$ and $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \,|\, d \in D)$, where ρ is a polynomial parameter on D. For every $k \in K$, let $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{y}, \mathbf{z} \leftarrow \mathcal{U}(\mathbb{Z}_p^{\rho(\mathbf{d})})$, and $\mathbf{t} \leftarrow \mathcal{U}(\mathbb{Z}_p)$. Then the probability ensembles $((\mathbf{d}, \psi_\mathbf{d}(\mathbf{y}), \mathbf{z}, \mathbf{y}\mathbf{z}^\mathsf{T}) \,|\, k \in K)$ and $((\mathbf{d}, \psi_\mathbf{d}(\mathbf{y}), \mathbf{z}, \mathbf{t}) \,|\, k \in K)$ are computationally indistinguishable.*

Note that for any $y = (y_1, \ldots, y_n) \in \mathbb{Z}_p^n$ and $z = (z_1, \ldots, z_n) \in \mathbb{Z}_p^n$ (where $n \in \mathbb{N}$), yz^{T} is the inner product of y and z over the field $\mathbb{Z}_p$, i.e., $y_1 z_1 + \cdots + y_n z_n$.

Definition 2.9. Let $(I_k \mid k \in K)$ be a pairwise disjoint family of nonempty subsets of $\{0,1\}^*$ and let $I = \bigcup_{k \in K} I_k$. For each $i \in I$, define $\kappa(i)$ as the unique $k \in K$ such that $i \in I_k$. Assume that the following two conditions hold:

- There exists a polynomial π such that $I_k \subseteq \{0,1\}^{\leq \pi(k)}$ for any $k \in K$.
- The function $\kappa\colon I \to K$ defined above is a polynomial parameter.

Moreover, suppose σ and τ are polynomial parameters on K. Then a family $(\chi_i\colon \mathbb{Z}_p^{\sigma(\kappa(i))} \to \mathbb{Z}_p^{\tau(\kappa(i))} \mid i \in I)$ of functions is called a *family of p-ary hash functions* if this family is polynomial-time computable and $\sigma(k) > \tau(k)$ for all $k \in K$.

3 (Weakly) Pseudo-Free Families of Computational Elementary Abelian p-Groups

3.1 Families of Computational Elementary Abelian p-Groups

Loosely speaking, a family of computational groups consists of groups G_d (where $d \in D$) whose elements are represented by bit strings in such a way that equality testing, multiplication, inversion, computing the identity element, and sampling random elements in G_d can be performed efficiently when d is given. See [Ano13, Definition 3.1] for a formal definition of a family of computational groups. In this paper, we consider only families $(G_d \mid d \in D)$ of computational elementary abelian p-groups such that the following additional conditions hold:

- For any $d \in D$, each element of G_d is represented by a single bit string. Hence we can assume that $G_d \subseteq \{0,1\}^*$ and that the representation of each element $g \in G_d$ is g itself.
- There exists a polynomial η such that $G_d \subseteq \{0,1\}^{\leq \eta(|d|)}$ for all $d \in D$. In this case, the family of computational groups has exponential size, i.e., there exists a polynomial η' such that $|G_d| \leq 2^{\eta'(|d|)}$ for all $d \in D$. See also [Ano13, Definition 3.2]. As noted in [Ano13], pseudo-free families that do not have exponential size *per se* are of little interest.

Now we give a formal definition of a family of computational elementary abelian p-groups (with the above restrictions).

Definition 3.1. Let $((G_d, \mathcal{G}_d) \mid d \in D)$ be a family of pairs, where $G_d \subseteq \{0,1\}^*$ is an elementary abelian p-group and $\mathcal{G}_d$ is a probability distribution on G_d for any $d \in D$. Then this family is said to be a *family of computational elementary abelian p-groups* if the following conditions hold:

- There exists a polynomial η such that $G_d \subseteq \{0,1\}^{\leq \eta(|d|)}$ for all $d \in D$.
- There exists a deterministic polynomial-time algorithm that, given $d \in D$ and $g, h \in G_d$, computes $g + h$ in G_d.
- The probability ensemble $(\mathcal{G}_d \mid d \in D)$ is polynomial-time samplable.

For example, if ρ is a polynomial parameter on D, then $((\mathbb{Z}_p^{\rho(d)}, \mathcal{U}(\mathbb{Z}_p^{\rho(d)})) \mid d \in D)$ is a family of computational elementary abelian p-groups. Note that before Definition 3.1 we do not specify the distributions on the sets of representations of group elements when speaking of families of computational groups. This is because these distributions do not matter for us there.

In the rest of the paper, $\Gamma = ((G_d, \mathcal{G}_d) \mid d \in D)$ denotes a family of computational elementary abelian p-groups.

Remark 3.2. It is evident that, given $d \in D$ and $g \in G_d$, $-g$ (in G_d) can be computed in polynomial time as $(p-1)g$. Moreover, the identity element of G_d can also be computed in polynomial time from $d \in D$ as pg for an arbitrary element $g \in G_d$ (which can be obtained by sampling from the distribution $\mathcal{G}_d$).

Remark 3.3. Suppose $((H_d, \mathcal{H}_d) \mid d \in D)$ is a family of computational elementary abelian p-groups and $\Phi = (\phi_d \colon H_d \to \{0,1\}^* \mid d \in D)$ is a polynomial-time computable family of functions. Assume that Φ satisfies Condition (i) of Definition 2.1 and that the function $(d, v, w) \mapsto v \circ_d w$ ($d \in D$, $v, w \in \phi_d(H_d)$) is polynomial-time computable, where $\circ_d$ is defined in this condition. Then it is easy to see that $((\phi_d(H_d), \phi_d(\mathcal{H}_d)) \mid d \in D)$ is a family of computational elementary abelian p-groups. Furthermore, Remark 3.2 shows that Φ also satisfies Condition (ii) of Definition 2.1. Thus, the family Φ is homomorphic.

3.2 Pseudo-Free Families of Computational Elementary Abelian p-Groups

Suppose $F_{\infty,\infty}$ is the elementary abelian p-group with basis $a_1, a_2, \dots, x_1, x_2, \dots$ (as a vector space over the field $\mathbb{Z}_p$). We consider $F_{\infty,\infty}$ as a free group in the variety of all elementary abelian p-groups. Furthermore, let $F_\infty = \langle a_1, a_2, \dots \rangle$, $F_{m,n} = \langle a_1, \dots, a_m, x_1, \dots, x_n \rangle$, and $F_m = F_{m,0} = \langle a_1, \dots, a_m \rangle$ for any $m, n \in \mathbb{N}$. It is well known that a_i and x_j (for all $i, j \in \mathbb{N} \setminus \{0\}$) can be considered as variables taking values in an arbitrary elementary abelian p-group G. Namely, suppose $w = \sum_{i=1}^{\infty} y_i a_i + \sum_{j=1}^{\infty} z_j x_j \in F_{\infty,\infty}$, where $y_i, z_j \in \mathbb{Z}_p$ for all $i, j \in \mathbb{N} \setminus \{0\}$. Here, of course, $y_1, y_2, \dots, z_1, z_2, \dots$ are uniquely determined by w and the sets $I_w = \{i \in \mathbb{N} \setminus \{0\} \mid y_i \neq 0\}$ and $J_w = \{j \in \mathbb{N} \setminus \{0\} \mid z_j \neq 0\}$ are finite. Assume that $w \in F_{m,n}$ for some $m, n \in \mathbb{N}$. (This means that $y_i = z_j = 0$ for all $i > m$ and $j > n$.) Let $g = (g_1, \dots, g_m, \dots)$ be an m'-tuple, where $m' \geq m$, or an infinite sequence of elements of G. Similarly, let $h = (h_1, \dots, h_n, \dots)$ be an n'-tuple, where $n' \geq n$, or an infinite sequence of elements of G. Then the element $w(g; h) \in G$ is defined as $y_1 g_1 + \dots + y_m g_m + z_1 h_1 + \dots + z_n h_n$. Whenever $n = 0$, we omit the semicolon in this notation, i.e., we write $w(g)$ instead of $w(g;)$. Note that $w = w(a; x)$, where, of course, $a = (a_1, a_2, \dots)$ and $x = (x_1, x_2, \dots)$.

In this paper, we use either of the two following representations of the element w for computational purposes:

- $(((i_1, y_{i_1}), \dots, (i_s, y_{i_s})), ((j_1, z_{j_1}), \dots, (j_t, z_{j_t})))$, where $\{i_1, \dots, i_s\} = I_w$, $i_1 < \dots < i_s$, $\{j_1, \dots, j_t\} = J_w$, and $j_1 < \dots < j_t$.
- $((y_1, \dots, y_m), (z_1, \dots, z_n))$, where $m = \max I_w$ and $n = \max J_w$. Here we put $\max \emptyset = 0$.

All our results depending on such a representation hold for both representations defined above. Note that every element of $F_{\infty,\infty}$ has a unique representation of each of the above forms.

Remark 3.4. By a *straight-line program over* $F_{\infty,\infty}$ we mean a sequence $(u_1, \dots, u_n)$ of tuples such that for any $l \in \{1, \dots, n\}$, either $u_l = (b, m)$, where $b \in \{a, x\}$ and $m \in \mathbb{N} \setminus \{0\}$, or $u_l = (i, j, +)$, where $i, j \in \{1, \dots, l-1\}$. Here a, x, and $+$ are considered as symbols. A straight-line program $(u_1, \dots, u_n)$ over $F_{\infty,\infty}$ naturally defines the sequence $(v_1, \dots, v_n)$ of elements of $F_{\infty,\infty}$ by induction. Namely, for every $l \in \{1, \dots, n\}$, we put $v_l = b_m$ if $u_l = (b, m)$ and $v_l = v_i + v_j$ if $u_l = (i, j, +)$, where b, m, i, and j are as above. Then the straight-line program $(u_1, \dots, u_n)$ *represents* the element v_n. Note that we do not need tuples u_l of the form $(i, -)$ defining $v_l = -v_i$ (where $i \in \{1, \dots, l-1\}$) because they can be replaced by sequences of at most $2\lfloor \log_2(p-1) \rfloor$ tuples of the form $(i, j, +)$. Also, 0 can be represented by a straight-line program over $F_{\infty,\infty}$ consisting of one tuple of the form (b, m) and at most $2\lfloor \log_2 p \rfloor$ tuples of the form $(i, j, +)$.

It is easy to see that, given a straight-line program over $F_{\infty,\infty}$ representing an element $w \in F_{\infty,\infty}$, the first of the above representations of w can be computed in polynomial time. Conversely, given the first of the above representations of an element $w \in F_{\infty,\infty}$, a straight-line program over $F_{\infty,\infty}$ representing w can also be computed in polynomial time. This is why we do not use the representation of elements of $F_{\infty,\infty}$ by straight-line programs over $F_{\infty,\infty}$ for computational purposes (unlike [Hoh03]).

Let G be an elementary abelian p-group and let $g = (g_1, \dots, g_m) \in G^m$, where $m \in \mathbb{N}$. Denote by $\Sigma(G, g)$ the set of all tuples $((v_1, w_1), \dots, (v_s, w_s), h)$ such that the following conditions hold:

- $s \in \mathbb{N} \setminus \{0\}$, $h \in G^n$ for some $n \in \mathbb{N}$, and $v_1, w_1, \dots, v_s, w_s \in F_{m,n}$.
- The system of equations
$$v_i(a; x) = w_i(a; x), \quad i = 1, \dots, s,$$
over variables $x_1, \dots, x_n$ is unsatisfiable in F_m (or, equivalently, in F_∞).
- $v_i(g; h) = w_i(g; h)$ in G for all $i \in \{1, \dots, s\}$.

Definition 3.5. The family Γ of computational elementary abelian p-groups is called *pseudo-free with respect to* $\mathcal{D}$ if for any polynomial π and any probabilistic polynomial-time algorithm A, $\Pr[A(1^k, \mathbf{d}, \mathbf{g}) \in \Sigma(G_{\mathbf{d}}, \mathbf{g})] = \mathrm{negl}(k)$, where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{g} \leftarrow \mathcal{G}_{\mathbf{d}}^{\pi(k)}$.

A more general definition of a pseudo-free family of computational groups (in an arbitrary variety $\mathfrak{V}$ of groups with respect to $\mathcal{D}$ and a representation for elements of the $\mathfrak{V}$-free group by bit strings) was given in [Ano13, Definition 3.3]. Our Definition 3.5 is a special case of that definition (in the variety of all elementary abelian p-groups and with respect to the above representations for elements of $F_{\infty,\infty}$).

3.3 Weak Pseudo-Freeness and Its Equivalence to Pseudo-Freeness for Families of Computational Elementary Abelian p-Groups

We define a weakly pseudo-free family of computational elementary abelian p-groups similarly to the definition of a weakly pseudo-free family of computational groups given by Rivest in [Riv04b, Slide 11]. For an elementary abelian p-group G and a tuple $g = (g_1, \ldots, g_m) \in G^m$, where $m \in \mathbb{N}$, let

$$\Sigma'(G, g) = \{v \in F_m \mid ((v, 0), ()) \in \Sigma(G, g)\} = \{v \in F_m \setminus \{0\} \mid v(g) = 0\}.$$

The condition of the next definition is obtained from the condition of Definition 3.5 by replacing $\Sigma(G_{\mathbf{d}}, \mathbf{g})$ by $\Sigma'(G_{\mathbf{d}}, \mathbf{g})$.

Definition 3.6. The family Γ of computational elementary abelian p-groups is said to be *weakly pseudo-free with respect to* $\mathcal{D}$ if for any polynomial π and any probabilistic polynomial-time algorithm A, $\Pr[A(1^k, \mathbf{d}, \mathbf{g}) \in \Sigma'(G_{\mathbf{d}}, \mathbf{g})] = \mathrm{negl}(k)$, where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{g} \leftarrow \mathcal{G}_{\mathbf{d}}^{\pi(k)}$.

Theorem 3.7. *The family Γ is pseudo-free with respect to $\mathcal{D}$ if and only if it is weakly pseudo-free with respect to $\mathcal{D}$.*

Proof. It is sufficient to construct deterministic polynomial-time algorithms B and C such that for every $d \in D$, $g \in G_d^m$ (where $m \in \mathbb{N}$), $u \in \Sigma(G_d, g)$, and $v \in \Sigma'(G_d, g)$, we have $B(u) \in \Sigma'(G_d, g)$ and $C(v) \in \Sigma(G_d, g)$.

Let $d \in D$ and $g \in G_d^m$, where $m \in \mathbb{N}$. Also, let $u = ((v_1, w_1), \ldots, (v_s, w_s), h) \in \Sigma(G_d, g)$, where $s \in \mathbb{N} \setminus \{0\}$, $h \in G_d^n$ for some $n \in \mathbb{N}$, and $v_i, w_i \in F_{m,n}$ for all $i \in \{1, \ldots, s\}$. Suppose B is a deterministic polynomial-time algorithm that proceeds on input u as follows:

1. By rearranging the scalar multiples of $a_1, \ldots, a_m, x_1, \ldots, x_n$ in $v_i(a; x)$ and $w_i(a; x)$, transform the system of equations
$$v_i(a; x) = w_i(a; x), \quad i = 1, \ldots, s, \tag{3.1}$$
into an equivalent system of the form
$$v_i'(x) = w_i'(a), \quad i = 1, \ldots, s, \tag{3.2}$$
where $v_i'(x) \in \langle x_1, \ldots, x_n \rangle$ and $w_i'(a) \in F_m$ for all $i \in \{1, \ldots, s\}$.

2. By using Gaussian elimination, transform system (3.2) into an equivalent system of the form
$$\begin{aligned} x_{n_j} + v_j''(x) &= w_j''(a), \quad j = 1, \ldots, t, \\ 0 &= w_l''(a), \quad l = t+1, \ldots, s, \end{aligned}$$
where $1 \le n_1 < \cdots < n_t \le n$, $0 \le t \le s$, $v_j''(x) \in \langle x_{n_j+1}, \ldots, x_n \rangle$ for all $j \in \{1, \ldots, t\}$, $w_i''(a) \in F_m$ for all $i \in \{1, \ldots, s\}$. (Since (3.1) is unsatisfiable in F_m, this system is also unsatisfiable in this group. This means that $w_l''(a) \neq 0$ for some $l \in \{t+1, \ldots, s\}$.)

3. Choose an index $l \in \{t+1, \ldots, s\}$ such that $w_l''(a) \neq 0$ (see the previous item) and return $w_l''(a)$. (It is easy to see that $w_l''(g) = 0$. Therefore, $B(u) \in \Sigma'(G_d, g)$.)

Let $v \in \Sigma'(G_d, g)$. Suppose C is a deterministic polynomial-time algorithm that returns $((v, 0), ())$ on input v. Then $C(v) \in \Sigma(G_d, g)$. □

In what follows, bearing in mind Theorem 3.7, we do not use the terms "weakly pseudo-free" and "weak pseudo-freeness" when speaking of families of computational elementary abelian p-groups.

Remark 3.8. For an elementary abelian p-group G and a tuple $g = (g_1, \dots, g_m) \in G^m$, where $m \in \mathbb{N}$, let

$$\Lambda(G, g) = \{(y_1, \dots, y_m) \in \mathbb{Z}_p^m \setminus \{0\} \,|\, y_1 g_1 + \dots + y_m g_m = 0\}.$$

It is easy to see that the condition of Definition 3.6 (and by Theorem 3.7, the condition of Definition 3.5 as well) holds if and only if for any polynomial π and any probabilistic polynomial-time algorithm A, $\Pr[A(1^k, \mathbf{d}, \mathbf{g}) \in \Lambda(G_{\mathbf{d}}, \mathbf{g})] = \mathrm{negl}(k)$, where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{g} \leftarrow \mathcal{G}_{\mathbf{d}}^{\pi(k)}$. In the sequel, we use only the last condition as a characterization of families of computational elementary abelian p-groups that are pseudo-free with respect to $\mathcal{D}$. Note that this condition does not depend on the representation of elements of $F_{\infty,\infty}$.

3.4 Some Remarks

Remark 3.9. Let Ξ be a set of polynomial parameters on E such that for any polynomial π there exists a polynomial parameter $\xi \in \Xi$ satisfying $\pi(k) \le \xi(1^k, d)$ for all $(1^k, d) \in E$. For example, we can take the set of all polynomial parameters on E as Ξ. Replace the polynomial π by $\xi \in \Xi$ and $\pi(k)$ by $\xi(1^k, \mathbf{d})$ in the condition defined in Remark 3.8. Then the modified version of this condition is equivalent to the original one. The same holds for the conditions of Definitions 3.5 and 3.6.

We prove that if the family Γ satisfies the original version of the condition defined in Remark 3.8, then it satisfies the modified version of this condition. The converse and the equivalence of the two versions for the conditions of Definitions 3.5 and 3.6 can be proved similarly. Let $\xi \in \Xi$ and let A be a probabilistic polynomial-time algorithm. Choose a polynomial π such that $\xi(1^k, d) \le \pi(k)$ for all $(1^k, d) \in E$. Suppose B is a probabilistic polynomial-time algorithm that proceeds on input $(1^k, d, g)$ for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, and $g = (g_1, \dots, g_{\pi(k)}) \in G_d^{\pi(k)}$ as follows:

1. Invoke A on input $(1^k, d, g')$, where $g' = (g_1, \dots, g_{\xi(1^k, d)})$.
2. If A returns a $\xi(1^k, d)$-tuple of elements of $\mathbb{Z}_p$, then return this tuple right-padded with $\pi(k) - \xi(1^k, d)$ zeros (to obtain a $\pi(k)$-tuple of elements of $\mathbb{Z}_p$). Otherwise, the algorithm B fails.

It is evident that $B(1^k, d, g) \in \Lambda(G_d, g)$ if and only if $A(1^k, d, g') \in \Lambda(G_d, g')$. Therefore,

$$\Pr[A(1^k, \mathbf{d}, \mathbf{g}') \in \Lambda(G_{\mathbf{d}}, \mathbf{g}')] = \Pr[B(1^k, \mathbf{d}, \mathbf{g}) \in \Lambda(G_{\mathbf{d}}, \mathbf{g})] = \mathrm{negl}(k),$$

where $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{g}_1, \dots, \mathbf{g}_{\pi(k)} \leftarrow \mathcal{G}_{\mathbf{d}}$, $\mathbf{g}' = (\mathbf{g}_1, \dots, \mathbf{g}_{\xi(1^k, \mathbf{d})})$, and $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_{\pi(k)})$.

Remark 3.10. Let $G_{1^k,d} = G_d$ and $\mathcal{G}_{1^k,d} = \mathcal{G}_d$ for each $(1^k, d) \in E$. Then $((G_e, \mathcal{G}_e) \,|\, e \in E)$ is a family of computational elementary abelian p-groups. Furthermore, this family is pseudo-free with respect to $\mathcal{E}$ if and only if the family Γ is pseudo-free with respect to $\mathcal{D}$.

By Remark 3.10, we can use both D and E as an index set for the family Γ when studying or using its pseudo-freeness. The advantage of using E is that it is the union of the pairwise disjoint family $(\operatorname{supp} \mathcal{E}_k \,|\, k \in K)$ satisfying the requirements of Definition 2.9. Therefore E is suitable for indexing families of p-ary hash functions. But we use D (except in the proof of Theorem 4.12) because we prefer to separate Γ from the probability ensemble $\mathcal{D}$.

Remark 3.11. Let $\rho\colon D \to \mathbb{N} \setminus \{0\}$ be a polynomial parameter. It is obvious that $\Gamma^\rho = ((G_d^{\rho(d)}, \mathcal{G}_d^{\rho(d)}) \,|\, d \in D)$ is a family of computational elementary abelian p-groups. Moreover, if the family Γ is pseudo-free with respect to $\mathcal{D}$, then the family Γ^ρ is also pseudo-free with respect to $\mathcal{D}$. Indeed, suppose π is a polynomial and A is a probabilistic polynomial-time algorithm. Let B be a probabilistic polynomial-time algorithm that proceeds on input $(1^k, d, g)$ for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, and $g = (g_1, \dots, g_{\pi(k)}) \in G_d^{\pi(k)}$ as follows:

1. Choose $\mathrm{g}_{i,j} \leftarrow \mathcal{G}_d$ for all $i \in \{1, \dots, \pi(k)\}$ and $j \in \{2, \dots, \rho(d)\}$.
2. Invoke A on input $(1^k, d, (\mathrm{v}_1, \dots, \mathrm{v}_{\pi(k)}))$, where $\mathrm{v}_i = (g_i, \mathrm{g}_{i,2}, \dots, \mathrm{g}_{i,\rho(d)})$ for any $i \in \{1, \dots, \pi(k)\}$.
3. Return the output of A (if it exists).

It is evident that $\Lambda(G_d^{\rho(d)}, (\mathrm{v}_1, \ldots, \mathrm{v}_{\pi(k)})) \subseteq \Lambda(G_d, g)$. Therefore,

$$\Pr[A(1^k, \mathbf{d}, \mathbf{v}) \in \Lambda(G_{\mathbf{d}}^{\rho(\mathbf{d})}, \mathbf{v})] \leq \Pr[B(1^k, \mathbf{d}, \mathbf{g}) \in \Lambda(G_{\mathbf{d}}, \mathbf{g})] = \mathrm{negl}(k),$$

where $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{v} \leftarrow (\mathcal{G}_{\mathbf{d}}^{\rho(\mathbf{d})})^{\pi(k)}$, and $\mathbf{g} \leftarrow \mathcal{G}_{\mathbf{d}}^{\pi(k)}$.

4 Necessary and Sufficient Conditions for Pseudo-Freeness and for the Existence of Pseudo-Free Families

4.1 The Functions $\mathrm{kn}_{G,g}$, the Families Kn_Γ^ρ, and the Probability Ensembles $\mathcal{I}_{\mathcal{D},\Gamma}^\rho$

Let G be an elementary abelian p-group and let $g = (g_1, \ldots, g_m) \in G^m$, where $m \in \mathbb{N}$. Then we define the function $\mathrm{kn}_{G,g} \colon \mathbb{Z}_p^m \to G$ by $\mathrm{kn}_{G,g}(y) = y_1 g_1 + \cdots + y_m g_m$ for all $y = (y_1, \ldots, y_m) \in \mathbb{Z}_p^m$. The function $\mathrm{kn}_{G,g}$ can be considered as a knapsack function (see [MM11]). But, unlike many other variants of knapsack functions and like discrete exponential functions, $\mathrm{kn}_{G,g}$ is a group homomorphism. Also, it is obvious that, given $d \in D$, $g \in G_d^m$, and $y \in \mathbb{Z}_p^m$, $\mathrm{kn}_{G_d,g}(y)$ can be computed in polynomial time.

Suppose ρ is a polynomial parameter on D. Then we denote by Kn_Γ^ρ the family $(\mathrm{kn}_{G_d,g} \,|\, d \in D,\ g \in G_d^{\rho(d)})$. Of course, Kn_Γ^ρ depends only on $(G_d \,|\, d \in D)$ and ρ. We use the notation with Γ and ρ because of its convenience. It is easy to see that Kn_Γ^ρ is homomorphic and polynomial-time computable. Moreover, for any $k \in K$, let $\mathcal{I}_{\mathcal{D},\Gamma,k}^\rho$ be the distribution of $(\mathbf{d}, \mathbf{g})$, where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{g} \leftarrow \mathcal{G}_{\mathbf{d}}^{\rho(\mathbf{d})}$. The probability ensemble $(\mathcal{I}_{\mathcal{D},\Gamma,k}^\rho \,|\, k \in K)$ is denoted by $\mathcal{I}_{\mathcal{D},\Gamma}^\rho$. For brevity, we use $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \,|\, d \in D)$ as a shorthand for $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \,|\, d \in D,\ g \in G_d^{\rho(d)})$ when speaking of the one-wayness of Kn_Γ^ρ with respect to $\mathcal{I}_{\mathcal{D},\Gamma}^\rho$ and $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \,|\, d \in D)$. This notation is used throughout the paper.

By the *problem of inverting* $\mathrm{kn}_{G_d,g}$ we mean the problem of finding an element in $\mathrm{kn}_{G_d,g}^{-1}(f)$ when given (d, g, f), where $d \in D$, $g \in G_d^m$, and $f \in \mathrm{kn}_{G_d,g}(\mathbb{Z}_p^m)$ $(m \in \mathbb{N})$. The next three remarks show that this problem has some nice properties.

Remark 4.1. Since $\mathrm{kn}_{G_d,g}$ is a group homomorphism, the problem of inverting this function is random self-reducible. Namely, there exists a probabilistic polynomial-time oracle algorithm A such that for any $d \in D$, $g \in G_d^m$, $f \in \mathrm{kn}_{G_d,g}(\mathbb{Z}_p^m)$ $(m \in \mathbb{N})$, and any probabilistic oracle O, we have

$$\Pr[A^O(d, g, f) \in \mathrm{kn}_{G_d,g}^{-1}(f)] = \Pr[O(\mathrm{kn}_{G_d,g}(\mathbf{y})) \in \mathrm{kn}_{G_d,g}^{-1}(\mathrm{kn}_{G_d,g}(\mathbf{y}))],$$

where $\mathbf{y} \leftarrow \mathcal{U}(\mathbb{Z}_p^m)$. This means that if O returns a preimage of $\mathrm{kn}_{G_d,g}(\mathbf{y})$ under $\mathrm{kn}_{G_d,g}$ with some probability $\delta(d, g)$, then A^O computes a preimage of any $f \in \mathrm{kn}_{G_d,g}(\mathbb{Z}_p^m)$ under $\mathrm{kn}_{G_d,g}$ with the same probability $\delta(d, g)$. A similar result for the discrete logarithm problem is well known.

The required algorithm A is similar to the algorithm in [Lub96, Lecture 4] for the discrete logarithm problem. The algorithm A proceeds on input (d, g, f), where d, g, and f are as above, as follows:

1. Choose $\mathrm{u} \leftarrow \mathcal{U}(\mathbb{Z}_p^m)$.
2. Query the oracle on $f + \mathrm{kn}_{G_d,g}(\mathrm{u})$. If the oracle returns a tuple $z \in \mathbb{Z}_p^m$, then return $z - \mathrm{u}$ (computed in $\mathbb{Z}_p^m$). Otherwise, the algorithm A fails.

The above result follows from the obvious fact that if $f = \mathrm{kn}_{G_d,g}(y)$ for some $y \in \mathbb{Z}_p^m$ and $\mathbf{u} \leftarrow \mathcal{U}(\mathbb{Z}_p^m)$, then $f + \mathrm{kn}_{G_d,g}(\mathbf{u}) = \mathrm{kn}_{G_d,g}(y + \mathbf{u})$, where $y + \mathbf{u}$ is distributed uniformly on $\mathbb{Z}_p^m$.

Remark 4.2. The problem of inverting $\mathrm{kn}_{G_d,g}$ is self-reducible in the following sense. For every $d \in D$, let O_d be an oracle that on input $(b, h) \in G_d^n \times G_d$ $(n \in \mathbb{N})$ returns 1 if $h \in \mathrm{kn}_{G_d,b}(\mathbb{Z}_p^n)$ and 0 otherwise. Then there exists a deterministic polynomial-time oracle algorithm A such that $A^{O_d}(d, g, f) \in \mathrm{kn}_{G_d,g}^{-1}(f)$ for all $d \in D$, $g \in G_d^m$, and $f \in \mathrm{kn}_{G_d,g}(\mathbb{Z}_p^m)$ $(m \in \mathbb{N})$. This fact seems to be well known (even for knapsack functions with polynomially bounded input coefficients; such knapsack functions are considered in [MM11]). But we provide a proof of it (for $\mathrm{kn}_{G_d,g}$) for completeness and for the convenience of the reader.

Let d, $g = (g_1, \ldots, g_m)$, and f be as above. The required algorithm A on input (d, g, f) successively finds (by exhaustive search and using the oracle O_d) some elements $y_1, \ldots, y_m \in \mathbb{Z}_p$ such that $f - y_1 g_1 - \cdots - y_i g_i \in \mathrm{kn}_{G_d,(g_{i+1},\ldots,g_m)}(\mathbb{Z}_p^{m-i})$ for all $i \in \{1, \ldots, m\}$. Then the algorithm A returns $y = (y_1, \ldots, y_m)$. By construction, we have $\mathrm{kn}_{G_d,g}(y) = f$. It is easy to see that such elements $y_1, \ldots, y_m$ exist.

Remark 4.3. There exists a deterministic polynomial-time oracle algorithm A such that for any $d \in D$, $g = (g_1, \ldots, g_m) \in G_d^m$, $f \in \mathrm{kn}_{G_d,g}(\mathbb{Z}_p^m)$ ($m \in \mathbb{N}$), and any basis b of G_d, we have $A^{\mathrm{kn}_{G_d,b}^{-1}}(d, g, f) \in \mathrm{kn}_{G_d,g}^{-1}(f)$. (It is evident that if $b = (b_1, \ldots, b_n) \in G_d^n$ is a basis of G_d, then $\mathrm{kn}_{G_d,b}$ is a group isomorphism from $\mathbb{Z}_p^n$ to G_d.) Namely, the algorithm A on input (d, g, f) (where d, g, and f are as above) returns a solution $(y_1, \ldots, y_m) \in \mathbb{Z}_p^m$ to the system of linear equations $y_1 \mathrm{kn}_{G_d,b}^{-1}(g_1) + \cdots + y_m \mathrm{kn}_{G_d,b}^{-1}(g_m) = \mathrm{kn}_{G_d,b}^{-1}(f)$. In particular, this implies the following fact: If β is a polynomial-time computable function on D such that $\beta(d)$ is a basis of G_d for all $d \in D$, then the problem of inverting $\mathrm{kn}_{G_d,g}$ is Cook-reducible to its special case when $g = \beta(d)$.

The problem of inverting $\mathrm{kn}_{G_d,g}$ might be of independent interest. One of the purposes of this paper is to draw attention to this problem. See also Problem 5.4 below.

4.2 Auxiliary Results

The proof of the next lemma is similar to that of Theorem 2.2 in [IN96].

Lemma 4.4. *Suppose ρ is a polynomial parameter on D such that the family Kn_Γ^ρ is one-way with respect to $\mathcal{I}_{\mathcal{D},\Gamma}^\rho$ and $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \,|\, d \in D)$. For every $k \in K$, let $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{g}_1, \ldots, \mathbf{g}_{\rho(\mathbf{d})}, \mathbf{h} \leftarrow \mathcal{G}_{\mathbf{d}}$, $\mathbf{u}_1, \ldots, \mathbf{u}_{\rho(\mathbf{d})}, \mathbf{v} \leftarrow \mathcal{U}(G_{\mathbf{d}})$, $\mathbf{y} \leftarrow \mathcal{U}(\mathbb{Z}_p^{\rho(\mathbf{d})})$, $\mathbf{g} = (\mathbf{g}_1, \ldots, \mathbf{g}_{\rho(\mathbf{d})})$, and $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_{\rho(\mathbf{d})})$. Assume that*

$$((\mathbf{d}, \mathbf{g}, \mathbf{h}) \,|\, k \in K) \text{ and } ((\mathbf{d}, \mathbf{u}, \mathbf{v}) \,|\, k \in K) \text{ are computationally indistinguishable.} \tag{4.1}$$

Then the probability ensembles $((\mathbf{d}, \mathbf{g}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{g}}(\mathbf{y})) \,|\, k \in K)$ and $((\mathbf{d}, \mathbf{g}, \mathbf{h}) \,|\, k \in K)$ are computationally indistinguishable.

Proof. Suppose $(\mathbf{e}_k \,|\, k \in K)$ and $(\mathbf{f}_k \,|\, k \in K)$ are probability ensembles consisting of random variables taking values in $\{0,1\}^*$. For brevity, we write $\mathbf{e}_k \approx \mathbf{f}_k$ if these probability ensembles are computationally indistinguishable.

Let $\mathbf{z} \leftarrow \mathcal{U}(\mathbb{Z}_p^{\rho(\mathbf{d})})$ and $\mathbf{t} \leftarrow \mathcal{U}(\mathbb{Z}_p)$. Then (4.1) and Lemma 2.8 imply that

$$\begin{aligned}(\mathbf{d}, \mathbf{u}, \mathbf{v}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y}), \mathbf{z}, \mathbf{y}\mathbf{z}^\mathsf{T}) &\approx (\mathbf{d}, \mathbf{g}, \mathbf{h}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{g}}(\mathbf{y}), \mathbf{z}, \mathbf{y}\mathbf{z}^\mathsf{T}) \\ &\approx (\mathbf{d}, \mathbf{g}, \mathbf{h}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{g}}(\mathbf{y}), \mathbf{z}, \mathbf{t}) \approx (\mathbf{d}, \mathbf{u}, \mathbf{v}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y}), \mathbf{z}, \mathbf{t}).\end{aligned} \tag{4.2}$$

Suppose A is a probabilistic polynomial-time algorithm. Let B be a probabilistic polynomial-time algorithm that proceeds on input $(1^k, d, u, v, f, z, t)$ for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, $u = (u_1, \ldots, u_{\rho(d)}) \in G_d^{\rho(d)}$, $v \in G_d$, $f \in \mathrm{kn}_{G_d,u}(\mathbb{Z}_p^{\rho(d)})$, $z = (z_1, \ldots, z_{\rho(d)}) \in \mathbb{Z}_p^{\rho(d)}$, and $t \in \mathbb{Z}_p$ as follows:

1. For all $i \in \{1, \ldots, \rho(d)\}$, compute $u_i' = u_i + z_i v$.
2. Invoke A on input $(1^k, d, u', f + tv)$, where $u' = (u_1', \ldots, u_{\rho(d)}')$.
3. Return the output of A (if it exists).

It is evident that if $f = \mathrm{kn}_{G_d,u}(y)$, where $y \in \mathbb{Z}_p^{\rho(d)}$, then $f + tv = \mathrm{kn}_{G_d,u'}(y) + (t - yz^\mathsf{T})v$.

Let $\mathbf{u}_i' = \mathbf{u}_i + \mathbf{z}_i \mathbf{v}$ for all $i \in \{1, \ldots, \rho(\mathbf{d})\}$ and $\mathbf{u}' = (\mathbf{u}_1', \ldots, \mathbf{u}_{\rho(\mathbf{d})}')$. It is easy to see that the random variable $(\mathbf{d}, \mathbf{u}', \mathbf{y})$ has the same distribution as $(\mathbf{d}, \mathbf{u}, \mathbf{y})$. Therefore,

$$\begin{aligned}\Pr[B(1^k, \mathbf{d}, \mathbf{u}, \mathbf{v}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y}), \mathbf{z}, \mathbf{y}\mathbf{z}^\mathsf{T}) = 1] &= \Pr[A(1^k, \mathbf{d}, \mathbf{u}', \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}'}(\mathbf{y})) = 1] \\ &= \Pr[A(1^k, \mathbf{d}, \mathbf{u}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y})) = 1].\end{aligned} \tag{4.3}$$

Furthermore, conditioned on $\mathbf{t} \neq \mathbf{y}\mathbf{z}^\mathsf{T}$, the random variables $(\mathbf{d}, \mathbf{u}', \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}'}(\mathbf{y}) + (\mathbf{t} - \mathbf{y}\mathbf{z}^\mathsf{T})\mathbf{v})$ and $(\mathbf{d}, \mathbf{u}, \mathbf{v})$ are identically distributed. Hence,

$$\begin{aligned}\Pr[B(1^k, \mathbf{d}, \mathbf{u}, \mathbf{v}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y}), \mathbf{z}, \mathbf{t}) = 1] &= \Pr[A(1^k, \mathbf{d}, \mathbf{u}', \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}'}(\mathbf{y}) + (\mathbf{t} - \mathbf{y}\mathbf{z}^\mathsf{T})\mathbf{v}) = 1] \\ &= \Pr[A(1^k, \mathbf{d}, \mathbf{u}', \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}'}(\mathbf{y}) + (\mathbf{t} - \mathbf{y}\mathbf{z}^\mathsf{T})\mathbf{v}) = 1 \,|\, \mathbf{t} = \mathbf{y}\mathbf{z}^\mathsf{T}] \Pr[\mathbf{t} = \mathbf{y}\mathbf{z}^\mathsf{T}] \\ &\quad + \Pr[A(1^k, \mathbf{d}, \mathbf{u}', \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}'}(\mathbf{y}) + (\mathbf{t} - \mathbf{y}\mathbf{z}^\mathsf{T})\mathbf{v}) = 1 \,|\, \mathbf{t} \neq \mathbf{y}\mathbf{z}^\mathsf{T}] \Pr[\mathbf{t} \neq \mathbf{y}\mathbf{z}^\mathsf{T}] \\ &= \frac{1}{p} \Pr[A(1^k, \mathbf{d}, \mathbf{u}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y})) = 1] + \frac{p-1}{p} \Pr[A(1^k, \mathbf{d}, \mathbf{u}, \mathbf{v}) = 1].\end{aligned} \tag{4.4}$$

It follows from (4.2)–(4.4) that

$$\begin{aligned}
&|\Pr[A(1^k, \mathbf{d}, \mathbf{u}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y})) = 1] - \Pr[A(1^k, \mathbf{d}, \mathbf{u}, \mathbf{v}) = 1]| \\
&\quad = \frac{p}{p-1}|\Pr[B(1^k, \mathbf{d}, \mathbf{u}, \mathbf{v}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y}), \mathbf{z}, \mathbf{y}\mathbf{z}^{\mathsf{T}}) = 1] - \Pr[B(1^k, \mathbf{d}, \mathbf{u}, \mathbf{v}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y}), \mathbf{z}, \mathbf{t}) = 1]| \\
&\quad = \mathrm{negl}(k).
\end{aligned}$$

Therefore, $(\mathbf{d}, \mathbf{u}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y})) \approx (\mathbf{d}, \mathbf{u}, \mathbf{v})$. On the other hand, (4.1) implies that $(\mathbf{d}, \mathbf{g}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{g}}(\mathbf{y})) \approx (\mathbf{d}, \mathbf{u}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{u}}(\mathbf{y}))$ and $(\mathbf{d}, \mathbf{u}, \mathbf{v}) \approx (\mathbf{d}, \mathbf{g}, \mathbf{h})$. Thus, $(\mathbf{d}, \mathbf{g}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{g}}(\mathbf{y})) \approx (\mathbf{d}, \mathbf{g}, \mathbf{h})$. □

Note that Lemma 4.4 is very close to a special case of Lemma 4.2 in [MM11] (or Corollary 1 in the preliminary version of that paper). We provide a proof of Lemma 4.4 for completeness and for the convenience of the reader.

Remark 4.5. For every $k \in K$, let $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{h} \leftarrow \mathcal{G}_{\mathbf{d}}$, and $\mathbf{v} \leftarrow \mathcal{U}(G_{\mathbf{d}})$, as in Lemma 4.4. By Remark 2.4 (with σ of the second type given in Example 2.3), if $(\mathcal{U}(G_d) \,|\, d \in D)$ is polynomial-time samplable and $((\mathbf{d}, \mathbf{h}) \,|\, k \in K)$ and $((\mathbf{d}, \mathbf{v}) \,|\, k \in K)$ are computationally indistinguishable, then Condition (4.1) holds for any polynomial parameter ρ on D. Moreover, if $\max_{d \in \operatorname{supp} \mathcal{D}_k} \Delta(\mathcal{G}_d, \mathcal{U}(G_d)) = \mathrm{negl}(k)$, then it is easy to see that the statistical distance between the distributions of $(\mathbf{d}, \mathbf{g}, \mathbf{h})$ and $(\mathbf{d}, \mathbf{u}, \mathbf{v})$ (in the notation of Lemma 4.4) is negligible as a function of $k \in K$. Therefore in this case Condition (4.1) also holds for any polynomial parameter ρ on D.

Lemma 4.6. *Assume that the family Γ is pseudo-free with respect to $\mathcal{D}$. Then for any polynomial parameter ρ on D, the family $\mathrm{Kn}^{\rho}_{\Gamma}$ is collision-intractable with respect to $\mathcal{I}^{\rho}_{\mathcal{D},\Gamma}$.*

Proof. Suppose ρ is a polynomial parameter on D and A is a probabilistic polynomial-time algorithm. Let B be a probabilistic polynomial-time algorithm that proceeds on input $(1^k, d, g)$ for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, and $g \in G_d^{\rho(d)}$ as follows:

1. Invoke A on input $(1^k, d, g)$.
2. If A returns a pair $(y, y') \in \mathbb{Z}_p^{\rho(d)} \times \mathbb{Z}_p^{\rho(d)}$, then return $y - y'$ (computed in $\mathbb{Z}_p^{\rho(d)}$). Otherwise, the algorithm B fails.

It is evident that $B(1^k, d, g) \in \Lambda(G_d, g)$ if and only if $A(1^k, d, g)$ is a collision for $\mathrm{kn}_{G_d,g}$. This implies that

$$\Pr[A(1^k, \mathbf{d}, \mathbf{g}) \text{ is a collision for } \mathrm{kn}_{G_{\mathbf{d}},\mathbf{g}}] = \Pr[B(1^k, \mathbf{d}, \mathbf{g}) \in \Lambda(G_{\mathbf{d}}, \mathbf{g})] = \mathrm{negl}(k),$$

where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{g} \leftarrow \mathcal{G}_{\mathbf{d}}^{\rho(\mathbf{d})}$. Here the second probability is negligible by Remark 3.9 with Ξ being the set of all polynomial parameters on E. We apply the modification (according to Remark 3.9) of the condition defined in Remark 3.8 to the polynomial parameter $(1^k, d) \mapsto \rho(d)$ on E (see the second type of polynomial parameters given in Example 2.3). □

Lemma 4.7. *Let $((H_d, \mathcal{H}_d) \,|\, d \in D)$ be a family of computational elementary abelian p-groups. Also, suppose $\Phi = (\phi_d \colon H_d \to G_d \,|\, d \in D)$ is a family of functions such that the following conditions hold:*

- *For any $d \in D$, ϕ_d is a homomorphism.*
- *The family Φ is one-way with respect to $\mathcal{D}$ and $(\mathcal{H}_d \,|\, d \in D)$.*
- *For $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{g} \leftarrow \mathcal{G}_{\mathbf{d}}$, and $\mathbf{h} \leftarrow \mathcal{H}_{\mathbf{d}}$, the probability ensembles $((\mathbf{d}, \mathbf{g}) \,|\, k \in K)$ and $((\mathbf{d}, \phi_{\mathbf{d}}(\mathbf{h})) \,|\, k \in K)$ are computationally indistinguishable.*

Then the family Γ is pseudo-free with respect to $\mathcal{D}$.

Proof. Suppose $\pi \colon K \to \{p^l \,|\, l \in \mathbb{N}\}$ is a polynomial parameter and A is a probabilistic polynomial-time algorithm. Let B be a probabilistic polynomial-time algorithm that proceeds on input $(1^k, d, f)$ for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, and $f \in \phi_d(H_d)$ as follows:

1. Choose $\mathrm{i} \leftarrow \mathcal{U}(\{1, \dots, \pi(k)\})$ and $\mathrm{r}_1, \dots, \mathrm{r}_{\mathrm{i}-1}, \mathrm{r}_{\mathrm{i}+1}, \dots, \mathrm{r}_{\pi(k)} \leftarrow \mathcal{H}_d$.
2. Invoke A on input $(1^k, d, w)$, where $w = (\phi_d(\mathrm{r}_1), \dots, \phi_d(\mathrm{r}_{\mathrm{i}-1}), f, \phi_d(\mathrm{r}_{\mathrm{i}+1}), \dots, \phi_d(\mathrm{r}_{\pi(k)}))$.

3. If A returns a tuple $(z_1, \dots, z_{\pi(k)}) \in \mathbb{Z}_p^{\pi(k)}$, where $z_{\mathrm{i}} \neq 0$, then return $-z_{\mathrm{i}}^{-1}(z_1 \mathrm{r}_1 + \cdots + z_{\mathrm{i}-1}\mathrm{r}_{\mathrm{i}-1} + z_{\mathrm{i}+1}\mathrm{r}_{\mathrm{i}+1} + \cdots + z_{\pi(k)}\mathrm{r}_{\pi(k)})$ (of course, z_{i}^{-1} is computed in the field $\mathbb{Z}_p$). Otherwise, the algorithm B fails.

Let $k \in K$, $\mathbf{i} \leftarrow \mathcal{U}(\{1, \dots, \pi(k)\})$, $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{r}_1, \dots, \mathbf{r}_{\pi(k)}, \mathbf{h} \leftarrow \mathcal{H}_{\mathbf{d}}$, $\mathbf{v} \leftarrow \mathcal{G}_{\mathbf{d}}^{\pi(k)}$, and

$$\mathbf{w} = (\phi_{\mathbf{d}}(\mathbf{r}_1), \dots, \phi_{\mathbf{d}}(\mathbf{r}_{\mathbf{i}-1}), \phi_{\mathbf{d}}(\mathbf{h}), \phi_{\mathbf{d}}(\mathbf{r}_{\mathbf{i}+1}), \dots, \phi_{\mathbf{d}}(\mathbf{r}_{\pi(k)})).$$

It is evident that $B(1^k, d, f) \in \phi_d^{-1}(f)$ if and only if $A(1^k, d, w) = (z_1, \dots, z_{\pi(k)}) \in \Lambda(G_d, w)$, where $z_{\mathrm{i}} \neq 0$. This implies that

$$\Pr[B(1^k, \mathbf{d}, \phi_{\mathbf{d}}(\mathbf{h})) \in \phi_{\mathbf{d}}^{-1}(\phi_{\mathbf{d}}(\mathbf{h}))] = \Pr[A(1^k, \mathbf{d}, \mathbf{w}) = (z_1, \dots, z_{\pi(k)}) \in \Lambda(G_{\mathbf{d}}, \mathbf{w}),\ z_{\mathbf{i}} \neq 0]. \tag{4.5}$$

Denote by $\nu(v)$ the number of random elements of $\mathbb{Z}_p$ used by the algorithm A on input v. Let $\mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_p^{\nu(1^k, \mathbf{d}, \mathbf{w})})$ represent the sequence of random elements of $\mathbb{Z}_p$ used by A on input $(1^k, \mathbf{d}, \mathbf{w})$. It is easy to see that the random variables $(\mathbf{d}, \mathbf{w}, \mathbf{s})$ and $\mathbf{i}$ are independent. Therefore,

$$\Pr[A(1^k, \mathbf{d}, \mathbf{w}) = (z_1, \dots, z_{\pi(k)}) \in \Lambda(G_{\mathbf{d}}, \mathbf{w}),\ z_{\mathbf{i}} \neq 0] \geq \frac{1}{\pi(k)} \Pr[A(1^k, \mathbf{d}, \mathbf{w}) \in \Lambda(G_{\mathbf{d}}, \mathbf{w})]. \tag{4.6}$$

By Remark 2.4 (with σ of the first type given in Example 2.3), the probability ensembles $((\mathbf{d}, \mathbf{v}) \,|\, k \in K)$ and $((\mathbf{d}, \mathbf{w}) \,|\, k \in K)$ are computationally indistinguishable. (It is obvious that $(\mathbf{d}, \mathbf{w})$ and $(\mathbf{d}, (\phi_{\mathbf{d}}(\mathbf{r}_1), \dots, \phi_{\mathbf{d}}(\mathbf{r}_{\pi(k)})))$ are identically distributed.) Hence,

$$\Pr[A(1^k, \mathbf{d}, \mathbf{v}) \in \Lambda(G_{\mathbf{d}}, \mathbf{v})] \leq \Pr[A(1^k, \mathbf{d}, \mathbf{w}) \in \Lambda(G_{\mathbf{d}}, \mathbf{w})] + \mathrm{negl}(k). \tag{4.7}$$

It follows from (4.5)–(4.7) that

$$\Pr[A(1^k, \mathbf{d}, \mathbf{v}) \in \Lambda(G_{\mathbf{d}}, \mathbf{v})] \leq \pi(k) \Pr[B(1^k, \mathbf{d}, \phi_{\mathbf{d}}(\mathbf{h})) \in \phi_{\mathbf{d}}^{-1}(\phi_{\mathbf{d}}(\mathbf{h}))] + \mathrm{negl}(k) = \mathrm{negl}(k).$$

By Remark 3.9, Γ is pseudo-free with respect to $\mathcal{D}$. Here we use this remark with Ξ being the set of all functions $\xi\colon E \to \mathbb{N}$ such that there exists a polynomial parameter $\xi'\colon K \to \{p^l \,|\, l \in \mathbb{N}\}$ satisfying $\xi(1^k, d) = \xi'(k)$ for all $(1^k, d) \in E$. □

The next corollary follows from Remark 3.3 and Lemma 4.7.

Corollary 4.8. *Let $((H_d, \mathcal{H}_d) \,|\, d \in D)$ be a family of computational elementary abelian p-groups. Also, suppose $(\phi_d\colon H_d \to \{0,1\}^* \,|\, d \in D)$ is a homomorphic family of functions that is one-way with respect to $\mathcal{D}$ and $(\mathcal{H}_d \,|\, d \in D)$. Then $((\phi_d(H_d), \phi_d(\mathcal{H}_d)) \,|\, d \in D)$ is a pseudo-free family of computational elementary abelian p-groups with respect to $\mathcal{D}$, where $\phi_d(H_d)$ is considered as an elementary abelian p-group under the operation $\circ_d$ defined in Condition (i) of Definition 2.1.*

Remark 4.9. Corollary 4.8 can be considered as a tool for constructing pseudo-free families of computational elementary abelian p-groups. For example, assume that there exist a family $((H_d, \mathcal{H}_d) \,|\, d \in D)$ of computational elementary abelian p-groups and a family $\Phi = (\phi_d\colon H_d \to \{0,1\}^* \,|\, d \in D)$ of functions such that the following conditions hold:

- Φ is one-way with respect to $\mathcal{D}$ and $(\mathcal{H}_d \,|\, d \in D)$.
- For any $d \in D$, ϕ_d is one-to-one. (Hence, Φ satisfies Condition (i) of Definition 2.1.)
- There exists a deterministic polynomial-time algorithm that, given $d \in D$ and $v, w \in \phi_d(H_d)$, computes $\phi_d(\phi_d^{-1}(v) + \phi_d^{-1}(w))$. (Hence by Remark 3.3, Φ satisfies Condition (ii) of Definition 2.1.)

Then Corollary 4.8 enables us to construct a pseudo-free family of computational elementary abelian p-groups with respect to $\mathcal{D}$. Moreover, we can conjecture that a family Φ satisfying the above conditions exists even in the case when $H_d = \mathbb{Z}_p^{\rho(d)}$ and $\mathcal{H}_d = \mathcal{U}(\mathbb{Z}_p^{\rho(d)})$ for all $d \in D$, where ρ is an appropriate polynomial parameter on D.

Lemma 4.10. *Suppose there exists a polynomial parameter ρ on D such that the following two conditions hold:*

- *The family* Kn_Γ^ρ *is one-way with respect to* $\mathcal{I}_{\mathcal{D},\Gamma}^\rho$ *and* $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \mid d \in D)$.
- *For* $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{g}' \leftarrow \mathcal{G}_{\mathbf{d}}^{\rho(\mathbf{d})+1}$, *and* $\mathbf{u}' \leftarrow \mathcal{U}(G_{\mathbf{d}})^{\rho(\mathbf{d})+1}$, *the probability ensembles* $((\mathbf{d}, \mathbf{g}') \mid k \in K)$ *and* $((\mathbf{d}, \mathbf{u}') \mid k \in K)$ *are computationally indistinguishable. (This condition is obviously equivalent to Condition (4.1) in Lemma 4.4.)*

Then the family Γ *is pseudo-free with respect to* $\mathcal{D}$.

Proof. Let ρ be a polynomial parameter on D such that the above two conditions hold. By Lemma 4.4, for $\mathbf{d} \leftarrow \mathcal{D}_k$ (where $k \in K$), $\mathbf{g} \leftarrow \mathcal{G}_{\mathbf{d}}^{\rho(\mathbf{d})}$, $\mathbf{h} \leftarrow \mathcal{G}_{\mathbf{d}}$, and $\mathbf{y} \leftarrow \mathcal{U}(\mathbb{Z}_p^{\rho(\mathbf{d})})$, the probability ensembles $((\mathbf{d}, \mathbf{g}, \mathbf{h}) \mid k \in K)$ and $((\mathbf{d}, \mathbf{g}, \mathrm{kn}_{G_{\mathbf{d}},\mathbf{g}}(\mathbf{y})) \mid k \in K)$ are computationally indistinguishable. Furthermore, Lemma 4.7 implies that the family $\Gamma' = ((G_d, \mathcal{G}_d) \mid d \in D,\ g \in G_d^{\rho(d)})$ of computational elementary abelian p-groups is pseudo-free with respect to $\mathcal{I}_{\mathcal{D},\Gamma}^\rho$. But it is easy to see that Γ' is pseudo-free with respect to $\mathcal{I}_{\mathcal{D},\Gamma}^\rho$ if and only if Γ is pseudo-free with respect to $\mathcal{D}$. □

4.3 Putting It All Together

Theorem 4.11. *Let* Θ *be the set of all polynomial parameters* $\theta\colon D \to \mathbb{N}$ *such that* $p^{\theta(d)} > |G_d|$ *for all sufficiently large* $k \in K$ *and all* $d \in \operatorname{supp} \mathcal{D}_k$. *Assume that at least one of the following two conditions (from Remark 4.5) holds:*

- $(\mathcal{U}(G_d) \mid d \in D)$ *is polynomial-time samplable and for* $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{h} \leftarrow \mathcal{G}_{\mathbf{d}}$, *and* $\mathbf{v} \leftarrow \mathcal{U}(G_{\mathbf{d}})$, *the probability ensembles* $((\mathbf{d}, \mathbf{h}) \mid k \in K)$ *and* $((\mathbf{d}, \mathbf{v}) \mid k \in K)$ *are computationally indistinguishable.*
- $\max_{d \in \operatorname{supp} \mathcal{D}_k} \Delta(\mathcal{G}_d, \mathcal{U}(G_d)) = \mathrm{negl}(k)$.

Then the following conditions are equivalent:

(i) *The family* Γ *is pseudo-free with respect to* $\mathcal{D}$.

(ii) *For any polynomial parameter* ρ *on* D, *the family* Kn_Γ^ρ *is collision-intractable with respect to* $\mathcal{I}_{\mathcal{D},\Gamma}^\rho$.

(iii) *For any polynomial parameter* $\theta \in \Theta$, *the family* $\mathrm{Kn}_\Gamma^\theta$ *is collision-intractable with respect to* $\mathcal{I}_{\mathcal{D},\Gamma}^\theta$.

(iv) *There exists a polynomial parameter* $\theta \in \Theta$ *such that the family* $\mathrm{Kn}_\Gamma^\theta$ *is collision-intractable with respect to* $\mathcal{I}_{\mathcal{D},\Gamma}^\theta$.

(v) *For any polynomial parameter* $\theta \in \Theta$, *the family* $\mathrm{Kn}_\Gamma^\theta$ *is one-way with respect to* $\mathcal{I}_{\mathcal{D},\Gamma}^\theta$ *and* $(\mathcal{U}(\mathbb{Z}_p^{\theta(d)}) \mid d \in D)$.

(vi) *There exists a polynomial parameter* $\theta \in \Theta$ *such that the family* $\mathrm{Kn}_\Gamma^\theta$ *is one-way with respect to* $\mathcal{I}_{\mathcal{D},\Gamma}^\theta$ *and* $(\mathcal{U}(\mathbb{Z}_p^{\theta(d)}) \mid d \in D)$.

Proof. (i) $\Longrightarrow$ (ii): Follows from Lemma 4.6.
(ii) $\Longrightarrow$ (iii): Trivial.
(iii) $\Longrightarrow$ (iv): Trivial (because $\Theta \neq \emptyset$).
(iii) $\Longrightarrow$ (v): Follows from Lemma 2.7.
(iv) $\Longrightarrow$ (vi): Follows from Lemma 2.7.
(v) $\Longrightarrow$ (vi): Trivial (because $\Theta \neq \emptyset$).
(vi) $\Longrightarrow$ (i): Follows from Remark 4.5 and Lemma 4.10. □

Theorem 4.12. *The following conditions are equivalent:*

(i) *There exists a pseudo-free family of computational elementary abelian p-groups (with respect to some probability ensemble of the required form).*

(ii) *For any polynomial parameter* $\eta\colon \mathbb{N} \to \mathbb{N}$ *such that* $\eta(n) > n$ *for all* $n \in \mathbb{N}$, *there exist a pairwise disjoint family* $(I_k \mid k \in K)$ *(consisting of nonempty subsets of* $\{0,1\}^*$*) satisfying the requirements of Definition 2.9 and a homomorphic collision-intractable (with respect to some polynomial-time samplable probability ensemble* $(\mathcal{I}_k \mid k \in K)$ *satisfying* $\operatorname{supp} \mathcal{I}_k \subseteq I_k$ *for all* $k \in K$*) family* $(\chi_i\colon \mathbb{Z}_p^{\eta(\tau(\kappa(i)))} \to \mathbb{Z}_p^{\tau(\kappa(i))} \mid i \in I)$ *of p-ary hash functions, where* $I = \bigcup_{k \in K} I_k$, $\kappa\colon I \to K$ *is from Definition 2.9, and* τ *is a polynomial parameter on* K.

(iii) *There exist a pairwise disjoint family* $(I_k \mid k \in K)$ *(consisting of nonempty subsets of* $\{0,1\}^*$*) satisfying the requirements of Definition 2.9 and a homomorphic collision-intractable (with respect to some polynomial-time samplable probability ensemble* $(\mathcal{I}_k \mid k \in K)$ *satisfying* $\operatorname{supp} \mathcal{I}_k \subseteq I_k$ *for all* $k \in K$*) family of* p*-ary hash functions indexed by* $\bigcup_{k\in K} I_k$.

(iv) *There exists a homomorphic family* $(\phi_u \colon \mathbb{Z}_p^{\rho(u)} \to \{0,1\}^* \mid u \in U)$ *of functions (where* U *is a subset of* $\{0,1\}^*$ *and* ρ *is a polynomial parameter on* U*) that is one-way with respect to some probability ensemble of the required form and the probability ensemble* $(\mathcal{U}(\mathbb{Z}_p^{\rho(u)}) \mid u \in U)$.

(v) *There exist a set* $V \subseteq \{0,1\}^*$*, a family* $((H_v, \mathcal{U}(H_v)) \mid v \in V)$ *of computational elementary abelian* p*-groups, and a homomorphic family* $(\psi_v \colon H_v \to \{0,1\}^* \mid v \in V)$ *of functions that is one-way with respect to some probability ensemble of the required form and the probability ensemble* $(\mathcal{U}(H_v) \mid v \in V)$.

(vi) *There exists a pseudo-free family* $((A_w, \mathcal{U}(A_w)) \mid w \in W)$ *of computational elementary abelian* p*-groups (with respect to some probability ensemble of the required form), where* W *is a subset of* $\{0,1\}^*$.

Proof. (i) $\Longrightarrow$ (ii): For any $n \in \mathbb{N}$, let α_n be the one-to-one function from $\{0,1\}^{\leq n}$ onto $\{0,1\}^{n+1} \setminus \{0^{n+1}\}$ defined by $\alpha_n(u) = u10^{n-|u|}$ for all $u \in \{0,1\}^{\leq n}$. Then the functions $(1^n, u) \mapsto \alpha_n(u)$ and $(1^n, v) \mapsto \alpha_n^{-1}(v)$, where $n \in \mathbb{N}$, $u \in \{0,1\}^{\leq n}$, and $v \in \{0,1\}^{n+1} \setminus \{0^{n+1}\}$, are polynomial-time computable.

Assume that Γ is pseudo-free with respect to $\mathcal{D}$. Choose a polynomial π such that $G_d \subseteq \{0,1\}^{\leq \pi(k)}$ for every $k \in K$ and $d \in \operatorname{supp} \mathcal{D}_k$. For each such k and d, let $\overline{G}_{1^k,d} = \alpha_{\pi(k)}(G_d)$ and $\overline{\mathcal{G}}_{1^k,d} = \alpha_{\pi(k)}(\mathcal{G}_d)$. Consider $\overline{G}_{1^k,d}$ as an elementary abelian p-group under the unique operation such that the restriction of $\alpha_{\pi(k)}$ to G_d is a group isomorphism from G_d to $\overline{G}_{1^k,d}$. Remark 3.10 implies that $\overline{\Gamma} = ((\overline{G}_e, \overline{\mathcal{G}}_e) \mid e \in E)$ is a pseudo-free family of computational elementary abelian p-groups with respect to $\mathcal{E}$. Suppose ρ is the polynomial parameter on E such that $\rho(1^k, d) = \eta(\pi(k) + 1)$ for any $(1^k, d) \in E$. Then $(\{(e, w) \mid e \in \operatorname{supp} \mathcal{E}_k,\ w \in \overline{G}_e^{\rho(e)}\} \mid k \in K)$ and $\operatorname{Kn}_{\overline{\Gamma}}^{\rho}$ satisfy the requirements of Condition (ii) ($\operatorname{Kn}_{\overline{\Gamma}}^{\rho}$ is collision-intractable with respect to $\mathcal{I}_{\mathcal{E},\overline{\Gamma}}^{\rho}$ by Lemma 4.6).

(ii) $\Longrightarrow$ (iii): Trivial.

(iii) $\Longrightarrow$ (iv): Follows from Lemma 2.7.

(iv) $\Longrightarrow$ (v): Trivial.

(v) $\Longrightarrow$ (vi): Follows from Corollary 4.8 and the well-known fact that if $\psi \colon H \to G$ is a group homomorphism, where H is finite, then $\psi(\mathcal{U}(H)) = \mathcal{U}(\psi(H))$.

(vi) $\Longrightarrow$ (i): Trivial. □

4.4 A Diffie-Hellman-Like Key Agreement Protocol

In this subsection, we construct a Diffie-Hellman-like key agreement protocol from the family Γ of computational elementary abelian p-groups. To describe this protocol, we need some notation. Let $Y = (y_{i,j}) \in \mathbb{Z}_p^{s\times m}$ and $Q = (q_{i,j}) \in G^{m\times n}$, where $s, m, n \in \mathbb{N}$ and G is an elementary abelian p-group. Then it is natural to define YQ as the $s \times n$ matrix over G whose (i,j) entry is $\sum_{l=1}^{m} y_{i,l} q_{l,j}$. We can consider G as a right vector space over the field $\mathbb{Z}_p$ such that $qz = zq$ for all $q \in G$ and $z \in \mathbb{Z}_p$. Hence for any $Z = (z_{i,j}) \in \mathbb{Z}_p^{n\times t}$ (where $t \in \mathbb{N}$), QZ is naturally defined as the $m \times t$ matrix over G whose (i,j) entry is $\sum_{l=1}^{n} q_{i,l} z_{l,j}$. It is easy to see that $(YQ)Z = Y(QZ)$. Note that in this notation, $\operatorname{kn}_{G,g}(y) = yg^{\mathsf{T}} = gy^{\mathsf{T}}$ for all $y \in \mathbb{Z}_p^m$ and $g \in G^m$.

The protocol uses a polynomial parameter $\rho \colon D \to \mathbb{N} \setminus \{0\}$. The public parameters of the protocol are $k \in K$ (the security parameter), $\mathrm{d} \leftarrow \mathcal{D}_k$, and $\mathrm{Q} \leftarrow \mathcal{G}_{\mathrm{d}}^{\rho(\mathrm{d})\times\rho(\mathrm{d})}$. We assume that the parties of the protocol (traditionally called Alice and Bob) communicate over a channel providing sender authenticity and message integrity. The protocol proceeds as follows:

1. Alice chooses $\mathrm{y} \leftarrow \mathcal{U}(\mathbb{Z}_p^{\rho(\mathrm{d})})$, computes yQ, and sends yQ to Bob.
2. Bob chooses $\mathrm{z} \leftarrow \mathcal{U}(\mathbb{Z}_p^{\rho(\mathrm{d})})$, computes Qz^{T}, and sends Qz^{T} to Alice.
3. Alice computes the common secret key $\mathrm{y}(\mathrm{Qz}^{\mathsf{T}})$.

4. Bob computes the common secret key $(\mathrm{y}\mathrm{Q})\mathrm{z}^{\mathsf{T}}$.

For any $k \in K$, let $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{Q} \leftarrow \mathcal{G}_{\mathbf{d}}^{\rho(\mathbf{d})\times\rho(\mathbf{d})}$, $\mathbf{y}, \mathbf{z} \leftarrow \mathcal{U}(\mathbb{Z}_p^{\rho(\mathbf{d})})$, and $\mathbf{u} \leftarrow \mathcal{U}(G_{\mathbf{d}})$. Standard security requirements for this protocol are as follows:

(i) For any probabilistic polynomial-time algorithm A, $\Pr[A(1^k, \mathbf{d}, \mathbf{Q}, \mathbf{yQ}, \mathbf{Qz}^{\mathsf{T}}) = \mathbf{yQz}^{\mathsf{T}}] = \mathrm{negl}(k)$. This condition is similar to the condition of computational hardness of the computational Diffie-Hellman problem.

(ii) The probability ensembles $((\mathbf{d}, \mathbf{Q}, \mathbf{yQ}, \mathbf{Qz}^{\mathsf{T}}, \mathbf{yQz}^{\mathsf{T}}) \,|\, k \in K)$ and $((\mathbf{d}, \mathbf{Q}, \mathbf{yQ}, \mathbf{Qz}^{\mathsf{T}}, \mathbf{u}) \,|\, k \in K)$ are computationally indistinguishable. This condition is similar to the condition of computational hardness of the decisional Diffie-Hellman problem.

It is easy to prove the following results:

- If the expectation of $1/|G_{\mathbf{d}}|$ is negligible as a function of $k \in K$, then (ii) implies (i).
- Let $\Gamma^\rho = ((G_d^{\rho(d)}, \mathcal{G}_d^{\rho(d)}) \,|\, d \in D)$, as in Remark 3.11. Then (i) implies that the family $\mathrm{Kn}_{\Gamma^\rho}^\rho$ is one-way with respect to $\mathcal{I}_{\mathcal{D},\Gamma^\rho}^\rho$ and $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \,|\, d \in D)$. (We identify a matrix with the tuple of its rows.)

Moreover, we have the following facts:

- Let $\mathbf{Q}' \leftarrow \mathcal{G}_{\mathbf{d}}^{(\rho(\mathbf{d})+1)\times\rho(\mathbf{d})}$ and $\mathbf{U}' \leftarrow \mathcal{U}(G_{\mathbf{d}})^{(\rho(\mathbf{d})+1)\times\rho(\mathbf{d})}$. Assume that the probability ensembles $((\mathbf{d}, \mathbf{Q}') \,|\, k \in K)$ and $((\mathbf{d}, \mathbf{U}') \,|\, k \in K)$ are computationally indistinguishable. Then, by Lemma 4.10, one-wayness of $\mathrm{Kn}_{\Gamma^\rho}^\rho$ with respect to $\mathcal{I}_{\mathcal{D},\Gamma^\rho}^\rho$ and $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \,|\, d \in D)$ implies pseudo-freeness of Γ^ρ with respect to $\mathcal{D}$.
- Recall that if Γ is pseudo-free with respect to $\mathcal{D}$, then Γ^ρ is also pseudo-free with respect to $\mathcal{D}$ (see Remark 3.11).

Unfortunately, we do not know whether (i) or (ii) holds under reasonable assumptions on Γ and ρ (e.g., under the one-wayness of $\mathrm{Kn}_{\Gamma^\rho}^\rho$ with respect to $\mathcal{I}_{\mathcal{D},\Gamma^\rho}^\rho$ and $(\mathcal{U}(\mathbb{Z}_p^{\rho(d)}) \,|\, d \in D)$ or the pseudo-freeness of Γ with respect to $\mathcal{D}$). We leave this as an interesting open question. See also Problem 5.3 below.

5 Problems for Further Research

In this section, we suggest some problems concerning families of computational elementary abelian p-groups for further research. Note that similar problems for some other objects were already posed. For example, see [Hoh03, Section 6.1, Problem 2], [Mic10, Section 5], [Riv04a, Section 7, Conjecture 2], and [Riv04b, Slide 22] for natural analogues of Problems 5.1–5.3 for pseudo-free families in the varieties of all groups and all abelian groups. Analogues of Problem 5.4 for numerous candidates for one-way families of functions are well known.

Problem 5.1. Construct a pseudo-free family of computational elementary abelian p-groups (with respect to a probability ensemble of the required form) under some standard cryptographic assumptions (e.g., under the general integer factoring intractability assumption).

Corollary 4.8 enables us to construct a pseudo-free family of computational elementary abelian p-groups under some nonstandard cryptographic assumption. See also Remark 4.9 and Theorem 4.12.

Problem 5.2. Find applications of pseudo-free families of computational elementary abelian p-groups. For example, construct some cryptographic primitives or secure cryptographic protocols from an arbitrary pseudo-free family of computational elementary abelian p-groups.

The proof of the implication (i) $\Longrightarrow$ (ii) of Theorem 4.12 shows how to construct a homomorphic collision-intractable family of p-ary hash functions (that is also one-way by Lemma 2.7) from a pseudo-free family of computational elementary abelian p-groups.

Problem 5.3. Explore the security of the Diffie-Hellman-like key agreement protocol presented in Subsection 4.4 (under reasonable assumptions on the family Γ and the polynomial parameter ρ).

Of course, Problem 5.3 is connected with Problem 5.2. Note that the results presented in Subsection 4.4 give only necessary conditions for the security of the protocol. Hasegawa et al. [HIST09, Theorem 6] proved that the computational Diffie-Hellman problem in a pseudo-free family (satisfying some additional condition) is in some sense computationally hard. Their proof is valid for pseudo-free families in any variety of infinite exponent, provided that, given an integer $n \geq 2$, a representation of the equation $x_1^n = a_1$ can be computed in polynomial time. (Here we use the notation of Section 1.) Informally speaking, the proof of this result in [HIST09] is based on a reduction from the proper power problem (also known as the strong RSA problem, see [Riv04a, Section 4]) to the computational Diffie-Hellman problem. Since the proper power problem is computationally hard in any pseudo-free family (see [Riv04a, Theorem 4] or [Riv04b, Slide 19]), the computational Diffie-Hellman problem in every such family is computationally hard, too. In fact, the reduction used in [HIST09] was proposed by Azimian in [Azi05] and goes back to the work of Shmuely [Shm85].

Problem 5.4. Explore the cryptographic properties of families of functions $\mathrm{kn}_{G_d,g}$ (where $d \in D$, $g \in G_d^m$, and $m \in \mathbb{N}$) for suitable families Γ of computational elementary abelian p-groups. The properties of the problem of inverting $\mathrm{kn}_{G_d,g}$ are particularly interesting.

Theorem 4.11 shows that Problem 5.4 is connected with Problem 5.1. See Subsection 4.1 for some remarks concerning the problem of inverting $\mathrm{kn}_{G_d,g}$.

Acknowledgement

This research was supported in part by the Russian Foundation for Basic Research (grant no. 13-01-00183).

References

[Ano13] M. Anokhin. Constructing a pseudo-free family of finite computational groups under the general integer factoring intractability assumption. *Groups — Complexity — Cryptology*, 5(1):53–74, 2013. Preliminary version: Electronic Colloquium on Computational Complexity (ECCC, https://eccc.weizmann.ac.il/), TR12-114, 2012.

[Azi05] K. Azimian. Breaking Diffie-Hellman is no easier than root finding. Electronic Colloquium on Computational Complexity (ECCC, https://eccc.weizmann.ac.il/), TR05-124, 2005.

[DGK+10] Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *Proceedings of the 7th Theory of Cryptography Conference (TCC 2010)*, volume 5978 of *Lecture Notes in Computer Science*, pages 361–381. Springer, 2010.

[Fuk14] M. Fukumitsu. *Pseudo-free groups and cryptographic assumptions.* PhD thesis, Department of Computer and Mathematical Sciences, Graduate School of Information Sciences, Tohoku University, January 2014.

[GB08] S. Goldwasser and M. Bellare. Lecture notes on cryptography. Available at http://cseweb.ucsd.edu/~mihir/papers/gb.html, July 2008.

[Gol01] O. Goldreich. *Foundations of cryptography. Volume 1 (Basic tools).* Cambridge University Press, 2001.

[HIST09] S. Hasegawa, S. Isobe, H. Shizuya, and K. Tashiro. On the pseudo-freeness and the CDH assumption. *International Journal of Information Security*, 8(5):347–355, 2009.

[Hoh03] S. R. Hohenberger. The cryptographic impact of groups with infeasible inversion. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 2003.

[IN96] R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of Cryptology*, 9(4):199–216, 1996. Preliminary version: Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS 1989), p. 236–241, 1989.

[Lub96] M. Luby. *Pseudorandomness and cryptographic applications*. Princeton University Press, 1996.

[Mic10] D. Micciancio. The RSA group is pseudo-free. *Journal of Cryptology*, 23(2):169–186, 2010. Preliminary version: Proceedings of EUROCRYPT 2005, v. 3494 of Lecture Notes in Computer Science, p. 387–403, Springer, 2005.

[MM11] D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2011/521, 2011. Preliminary version: Proceedings of CRYPTO 2011, v. 6841 of Lecture Notes in Computer Science, p. 465–484, Springer, 2011.

[Riv04a] R. L. Rivest. On the notion of pseudo-free groups. In *Proceedings of the 1st Theory of Cryptography Conference (TCC 2004)*, volume 2951 of *Lecture Notes in Computer Science*, pages 505–521. Springer, 2004.

[Riv04b] R. L. Rivest. On the notion of pseudo-free groups. Available at https://people.csail.mit.edu/rivest/pubs/Riv04e.slides.pdf, https://people.csail.mit.edu/rivest/pubs/Riv04e.slides.ppt, and http://people.csail.mit.edu/rivest/Rivest-TCC04-PseudoFreeGroups.ppt, February 2004. Presentation of [Riv04a].

[Shm85] Z. Shmuely. Composite Diffie-Hellman public-key generating systems are hard to break. Technical Report 356, Technion — Israel Institute of Technology, Department of Computer Science, Haifa, Israel, February 1985.

[Sho08] V. Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, 2nd edition, 2008.