Pseudo-Free Families of Computational Universal Algebras

Mikhail Anokhin

Information Security Institute,
Lomonosov University, Moscow, Russia

anokhin@mccme.ru

December 2, 2018

Abstract

Let Ω be a finite set of operation symbols. We initiate the study of (weakly) pseudo-free families of computational Ω-algebras in arbitrary varieties of Ω-algebras. Most of our results concern (weak) pseudo-freeness in the variety $\mathfrak{O}$ of all Ω-algebras. A family $(H_d \mid d \in D)$ of computational Ω-algebras (where $D \subseteq \{0,1\}^*$) is called polynomially bounded (resp., having exponential size) if there exists a polynomial η such that for all $d \in D$, the length of any representation of every $h \in H_d$ is at most $\eta(|d|)$ (resp., $|H_d| \le 2^{\eta(|d|)}$). First, we prove the following trichotomy: (i) if Ω consists of nullary operation symbols only, then there exists a polynomially bounded pseudo-free family in $\mathfrak{O}$; (ii) if $\Omega = \Omega_0 \cup \{\omega\}$, where Ω_0 consists of nullary operation symbols and the arity of ω is 1, then there exist an exponential-size pseudo-free family and a polynomially bounded weakly pseudo-free family (both in $\mathfrak{O}$); (iii) in all other cases, the existence of polynomially bounded weakly pseudo-free families in $\mathfrak{O}$ implies the existence of collision-resistant families of hash functions. Second, assuming the existence of collision-resistant families of hash functions, we construct a polynomially bounded weakly pseudo-free family and an exponential-size pseudo-free family of computational m-ary groupoids (both in $\mathfrak{O}$), where $m \ge 1$. In particular, for arbitrary $m \ge 2$, polynomially bounded weakly pseudo-free families of computational m-ary groupoids in $\mathfrak{O}$ exist if and only if collision-resistant families of hash functions exist. Moreover, we present some simple constructions of cryptographic primitives from pseudo-free families satisfying certain additional conditions. These constructions demonstrate the potential of pseudo-free families.

Keywords: Universal algebra, family of computational universal algebras, pseudo-free family, weakly pseudo-free family, collision-resistant family of hash functions, n-ary groupoid.

Contents

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 generating random elements can be performed efficiently. Loosely speaking, a family of computational groups is called pseudo-free if, given a random member G of the family (for a given security parameter) and random elements $g_1, \dots, g_m \in G$, it is computationally hard to find a system of group equations

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

in the variables $x_1, \dots, x_n$ together with elements $h_1, \dots, h_n \in G$ such that (1) is unsatisfiable in the free group freely generated by $a_1, \dots, a_m$, but

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

in G for all $i \in \{1, \dots, s\}$. If a family of computational groups satisfies this definition with the additional requirement that $n = 0$ (i.e., that the equations in (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) 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].

Let Ω be a finite set of operation symbols and let $\mathfrak{V}$ be a variety of Ω-algebras. (See Subsection 2.2 for definitions.) Then the notions of pseudo-freeness and weakly pseudo-freeness can be naturally extended to families of computational Ω-algebras in the variety $\mathfrak{V}$. Informally, a family of computational Ω-algebras is a family of Ω-algebras whose elements are represented by bit strings in such a way that equality testing, the fundamental operations, and generating random elements can be performed efficiently. To define a (weakly) pseudo-free family of computational Ω-algebras in $\mathfrak{V}$, we require that all Ω-algebras in the family belong to $\mathfrak{V}$ and replace the free group by the $\mathfrak{V}$-free Ω-algebra in the above definition of a (weakly) pseudo-free family of groups. In this case, $v_i(a_1, \dots, a_m; x_1, \dots, x_n)$ and $w_i(a_1, \dots, a_m; x_1, \dots, x_n)$ in (1) are elements of the $\mathfrak{V}$-free Ω-algebra freely generated by $a_1, \dots, a_m, x_1, \dots, x_n$. Of course, pseudo-free families in different varieties are completely different objects.

Until now, researchers have considered pseudo-freeness (in various versions) only in the varieties of all groups [Hoh03, Riv04a, Riv04b, HT07, HIST09, Ano13], of all abelian groups [Hoh03, Riv04a, Riv04b, HT07, Mic10, JB09, CFW11, FHI$^+$13, FHIS14a, FHIS14b, Ano18], and of all elementary abelian p-groups, where p is a prime [Ano17]. A survey of some results concerning pseudo-freeness can be found in [Fuk14, Chapter 1].

In this paper, we initiate the study of (weakly) pseudo-free families of computational Ω-algebras in arbitrary varieties of Ω-algebras. We hope that the study of these families will open up new opportunities in mathematical cryptography.

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 (weakly) pseudo-free families of computational Ω-algebras and related notions. In particular, the results of Subsections 3.4–3.5 can be considered as tools for constructing (weakly) pseudo-free families of computational Ω-algebras. Furthermore, Subsection 3.6 contains simple constructions of the following cryptographic primitives from pseudo-free families satisfying certain additional conditions:

- one-way family of permutations (see Example 3.22);
- family of trapdoor permutations (see Example 3.23);
- claw-resistant family of pairs of permutations (see Example 3.24).

These constructions demonstrate the potential of pseudo-free families of computational Ω-algebras.

Let $\mathfrak{O}$ denote the variety of all Ω-algebras. In Section 4, we study the following question: When polynomially bounded (weakly) pseudo-free families in $\mathfrak{O}$ exist unconditionally? A family $\mathtt{H} = (H_d \mid d \in D)$ of computational Ω-algebras (where $D \subseteq \{0,1\}^*$) is called polynomially bounded if there exists a polynomial η such that the length of any representation of every $h \in H_d$ is at most $\eta(|d|)$ for all $d \in D$. (See also Definition 3.3.) Furthermore, the family $\mathtt{H}$ is said to have exponential size if there exists a polynomial η such that $|H_d| \leq 2^{\eta(|d|)}$ for all $d \in D$. (See Definition 3.2.) It should be noted that a (weakly) pseudo-free family can have applications in cryptography only if it is polynomially bounded or at least has exponential size. (Weakly) pseudo-free families that do not have exponential size *per se* are of little interest; they can be constructed unconditionally (see Subsection 3.4). Loosely speaking, the main results of Section 4 can be summarized as follows:

(i) If Ω consists of nullary operation symbols only, then there exists a polynomially bounded pseudo-free family in $\mathfrak{O}$.

(ii) Assume that $\Omega = \Omega_0 \cup \{\omega\}$, where Ω_0 consists of nullary operation symbols and the arity of ω is 1. Then there exist an exponential-size pseudo-free family and a polynomially bounded weakly pseudo-free family (both in $\mathfrak{O}$).

(iii) In all other cases, the existence of polynomially bounded weakly pseudo-free families in $\mathfrak{O}$ implies the existence of collision-resistant families of hash functions. Thus, in these cases, such weakly pseudo-free families cannot be constructed unconditionally.

Moreover, the (weakly) pseudo-free families unconditionally existing in Cases (i)–(ii) have unique representations, i.e., each element of any Ω-algebra in these families is represented by a single bit string. (See Definition 3.4.) This property seems to be useful in applications. For precise statements of these results, see Subsection 4.3 and references therein.

In Section 5, we consider the case where Ω consists of a single operation symbol of arity $m \geq 1$. In this case, Ω-algebras are called m-ary groupoids. Assuming the existence of collision-resistant families of hash functions, we construct a polynomially bounded weakly pseudo-free family and an exponential-size pseudo-free family of computational m-ary groupoids (both in $\mathfrak{O}$). Moreover, the first family has unique representations. Combining this with the results of Section 4, we obtain that for arbitrary $m \geq 2$, polynomially bounded weakly pseudo-free families of computational m-ary groupoids in $\mathfrak{O}$ exist if and only if collision-resistant families of hash functions exist. The same holds if the weakly pseudo-free family is additionally required to have unique representations. These results are stated loosely here; for precise statements, we refer the reader to Subsections 5.1–5.2.

Finally, Section 6 concludes and suggests some directions for future research.

2 Preliminaries

2.1 General Preliminaries

In this paper, $\mathbb{N}$ denotes the set of all nonnegative integers. Let $n \in \mathbb{N}$. For a set Y, we denote by Y^n the set of all (ordered) n-tuples of elements from Y. The operation of disjoint union is denoted by $\sqcup$. We consider elements of $\{0,1\}^n$ as bit strings of length n. Furthermore, let $\{0,1\}^{\leq n} = \bigsqcup_{i=0}^{n} \{0,1\}^i$ and $\{0,1\}^* = \bigsqcup_{i=0}^{\infty} \{0,1\}^i$. 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 $(q_i \,|\, i \in I)$ the family of all such objects and by $\{q_i \,|\, i \in I\}$ 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.

Suppose ϕ is a function. We denote by $\operatorname{dom}\phi$ the domain of ϕ. Also, we use the same notation for ϕ and for the function $(y_1, \dots, y_n) \mapsto (\phi(y_1), \dots, \phi(y_n))$, where $n \in \mathbb{N}$ and $(y_1, \dots, y_n) \in (\operatorname{dom}\phi)^n$.

Let ρ be a function from a subset of $\{0,1\}^*$ onto a set S and let $s \in S$. Then, unless otherwise specified, $[s]_\rho$ denotes an arbitrary preimage of s under ρ. A similar notation was used by Boneh and Lipton in [BL96] and by Hohenberger in [Hoh03]. In general, $[s]_\rho$ denotes many strings in $\{0,1\}^*$ unless ρ is one-to-one. We use any of these strings as a representation of s for computational purposes.

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 Algebraic Preliminaries

In this subsection, we recall the basic definitions and simple facts from universal algebra. For a detailed introduction to this subject, the reader is referred to standard books, e.g., [Coh81], [BS12], or [Wec92].

Throughout the paper, Ω denotes a set of operation symbols. Each $\omega \in \Omega$ is assigned a nonnegative integer called the *arity* of ω and denoted by $\operatorname{ar}\omega$. An Ω-*algebra* is a set H called the *carrier* (or the *underlying set*) together with a family $(\widehat{\omega}\colon H^{\operatorname{ar}\omega} \to H \,|\, \omega \in \Omega)$ of finitary operations on H called the *fundamental operations*. For simplicity of notation, the fundamental operation $\widehat{\omega}$ associated with a symbol $\omega \in \Omega$ will be denoted by ω. Furthermore, we denote an Ω-algebra and its carrier by the same symbol.

Let H be an Ω-algebra. A set $G \subseteq H$ is called a *subalgebra* of H if it is closed under the fundamental operations of H. If S is a system of elements of H, then we denote by $\langle S \rangle$ the subalgebra of H generated by S, i.e., the smallest subalgebra of H containing S.

An equivalence relation θ on H is said to be a *congruence* (on H) if

$$(h_1, h_1'), \dots, (h_{\operatorname{ar}\omega}, h_{\operatorname{ar}\omega}') \in \theta \implies (\omega(h_1, \dots, h_{\operatorname{ar}\omega}), \omega(h_1', \dots, h_{\operatorname{ar}\omega}')) \in \theta$$

for any $\omega \in \Omega$ and $h_1, h_1', \dots, h_{\operatorname{ar}\omega}, h_{\operatorname{ar}\omega}' \in H$. Suppose θ is a congruence on H. For arbitrary $h \in H$, we denote by h/θ the equivalence class of h under θ. Moreover, let $H/\theta = \{h/\theta \,|\, h \in H\}$. Then H/θ is an Ω-algebra whose fundamental operations are well defined as follows:

$$\omega(h_1/\theta, \dots, h_{\operatorname{ar}\omega}/\theta) = \omega(h_1, \dots, h_{\operatorname{ar}\omega})/\theta, \quad \omega \in \Omega,\ h_1, \dots, h_{\operatorname{ar}\omega} \in H.$$

This Ω-algebra is called the *quotient algebra* of H by θ. Also, let $\theta^{\neq} = \{(h, h') \in \theta \,|\, h \neq h'\}$. If $\rho\colon Y \to H$, then ρ/θ denotes the function $y \mapsto \rho(y)/\theta$, where $y \in Y$.

A *homomorphism* of H to an Ω-algebra L is a function $\phi\colon H \to L$ such that for every $\omega \in \Omega$ and $h_1, \dots, h_{\operatorname{ar}\omega} \in H$,

$$\phi(\omega(h_1, \dots, h_{\operatorname{ar}\omega})) = \omega(\phi(h_1), \dots, \phi(h_{\operatorname{ar}\omega})).$$

If a homomorphism of H onto L is one-to-one, then it is called an *isomorphism*. Let $\phi\colon H \to L$ be a homomorphism. Then its *kernel* is defined as $\{(h, h') \in H^2 \,|\, \phi(h) = \phi(h')\}$. It is evident that the kernel of ϕ is a congruence on H. For example, if θ is a congruence on H, then $h \mapsto h/\theta$ (where $h \in H$) is a homomorphism of H onto H/θ (called the natural homomorphism) with kernel θ.

An Ω-algebra with only one element is said to be *trivial*. It is obvious that all trivial Ω-algebras are isomorphic.

If $\Omega = \{\omega\}$, where $\operatorname{ar}\omega = m \geq 1$, then Ω-algebras are called m-*ary groupoids* (or m-*groupoids*). When $m = 2$, these Ω-algebras are called simply *groupoids*. Note that some authors consider m-ary groupoids only for $m \geq 2$.

Put $\Omega_0 = \{\omega \in \Omega \,|\, \operatorname{ar}\omega = 0\}$. We note that if $\Omega_0 = \emptyset$, then an Ω-algebra can be empty. Whenever $\omega \in \Omega_0$, it is common to write ω instead of $\omega()$.

Let Z be a set of objects called variables. We always assume that any variable is not in Ω. The set $\operatorname{Tm}(Z)$ of all Ω-*terms* (or simply *terms*) over Z is defined as the smallest set such that $\Omega_0 \sqcup Z \subseteq \operatorname{Tm}(Z)$ and if $\omega \in \Omega \setminus \Omega_0$ and $v_1, \dots, v_{\operatorname{ar}\omega} \in \operatorname{Tm}(Z)$, then the formal expression $\omega(v_1, \dots, v_{\operatorname{ar}\omega})$ is in $\operatorname{Tm}(Z)$. The

Ω-terms can be considered as strings over the alphabet consisting of all symbols from $\Omega \sqcup Z$, parentheses, and comma. Of course, $\mathrm{Tm}(Z)$ is an Ω-algebra under the natural fundamental operations. This Ω-algebra is called the *Ω-term algebra* over Z.

Suppose $v \in \mathrm{Tm}(Z)$. Then the set $\mathrm{subt}(v)$ of *subterms* of the term v is defined inductively as follows:

$$\mathrm{subt}(v) = \begin{cases} \{v\} & \text{if } v \in \Omega_0 \sqcup Z, \\ \{v\} \sqcup \bigcup_{i=1}^{\mathrm{ar}\,\omega} \mathrm{subt}(v_i) & \text{if } v = \omega(v_1, \dots, v_{\mathrm{ar}\,\omega}),\ \omega \in \Omega \setminus \Omega_0,\ v_1, \dots, v_{\mathrm{ar}\,\omega} \in \mathrm{Tm}(Z). \end{cases}$$

Let the string $P(v)$ over $\Omega \sqcup Z$ be obtained from the term v by removing all parentheses and commas. The string $P(v)$ is known as the term v written in *Polish notation.* It is well known that the function $v \mapsto P(v)$ $(v \in \mathrm{Tm}(Z))$ is one-to-one. Moreover, if the arities of operation symbols occurring in v are known, then v can be easily recovered from $P(v)$. See [Coh81, Chapter III, Section 2] for details, although in that book reverse Polish notation is used.

Consider the case where $Z = \{z_1, z_2, \dots\}$, where $z_1, z_2, \dots$ are distinct. Assume that $v \in \mathrm{Tm}(\{z_1, \dots, z_m\})$ for some $m \in \mathbb{N}$. Furthermore, let $h_1, \dots, h_m \in H$. Then the element $v(h_1, \dots, h_m) \in H$ is defined inductively in the natural way. It is easy to see that $\{v(h_1, \dots, h_m) \,|\, v \in \mathrm{Tm}(\{z_1, \dots, z_m\})\} = \langle h_1, \dots, h_m \rangle$.

An *identity* (or a *law*) over Ω is a closed first-order formula of the form $\forall\, z_1, \dots, z_m\ (v = w)$, where $v, w \in \mathrm{Tm}(\{z_1, \dots, z_m\})$ $(m \in \mathbb{N})$. A class $\mathfrak{V}$ of Ω-algebras is said to be a *variety* if it can be defined by a set Φ of identities. This means that for any Ω-algebra G, $G \in \mathfrak{V}$ if and only if G satisfies all identities in Φ. By the famous Birkhoff variety theorem (see, e.g., [Coh81, Chapter IV, Theorem 3.1], [BS12, Chapter II, Theorem 11.9], or [Wec92, Subsection 3.2.3, Theorem 21]), a class of Ω-algebras is a variety if and only if it is closed under taking subalgebras, homomorphic images, and direct products. Note that if a class of Ω-algebras is closed under taking direct products, then it contains a trivial Ω-algebra as the direct product of the empty family of Ω-algebras. Recall that if $(H_i \,|\, i \in I)$ is a family of Ω-algebras, then the fundamental operations of the *direct product* of this family are defined as follows:

$$\omega((h_{1,i} \,|\, i \in I), \dots, (h_{\mathrm{ar}\,\omega,i} \,|\, i \in I)) = (\omega(h_{1,i}, \dots, h_{\mathrm{ar}\,\omega,i}) \,|\, i \in I),$$

where $\omega \in \Omega$ and $h_{1,i}, \dots, h_{\mathrm{ar}\,\omega,i} \in H_i$ $(i \in I)$.

The variety consisting of all Ω-algebras with at most one element is said to be *trivial*; all other varieties of Ω-algebras are called *nontrivial.* The trivial variety is defined by the identity $\forall\, z_1, z_2\ (z_1 = z_2)$. When $\Omega_0 = \emptyset$, the trivial variety contains not only trivial Ω-algebras, but also the empty Ω-algebra. If $\mathfrak{C}$ is a class of Ω-algebras, then the *variety generated by* $\mathfrak{C}$ is the smallest variety of Ω-algebras containing $\mathfrak{C}$. This variety is defined by the set of all identities holding in all Ω-algebras in $\mathfrak{C}$.

Let $\mathfrak{V}$ be a variety of Ω-algebras. Then an Ω-algebra $F \in \mathfrak{V}$ is said to be *$\mathfrak{V}$-free* if it has a generating system $(f_i \,|\, i \in I)$ such that for every system of elements $(g_i \,|\, i \in I)$ of any Ω-algebra $G \in \mathfrak{V}$ there exists a homomorphism $\alpha\colon F \to G$ satisfying $\alpha(f_i) = g_i$ for all $i \in I$ (evidently, this homomorphism α is unique). Any generating system $(f_i \,|\, i \in I)$ with this property is called *free* and the Ω-algebra F is said to be *freely generated by* every such system. It is well known (see, e.g., [Coh81, Chapter IV, Corollary 3.3], [BS12, Chapter II, Definition 10.9 and Theorem 10.10], or [Wec92, Subsection 3.2.3, Theorem 16]) that for any set I there exists a unique $\mathfrak{V}$-free Ω-algebra (up to isomorphism) with a free generating system indexed by I. It is easy to see that if $\mathfrak{V}$ is nontrivial, then for each free generating system $(f_i \,|\, i \in I)$ of a $\mathfrak{V}$-free Ω-algebra, f_i are distinct. In this case, one can consider free generating systems as sets.

We denote by $F_{\infty,\infty}(\mathfrak{V})$ the $\mathfrak{V}$-free Ω-algebra freely generated by $a_1, a_2, \dots, x_1, x_2, \dots$. Of course, the elements of this free generating system are assumed to be distinct. Furthermore, suppose $m, n \in \mathbb{N}$ and let $F_\infty(\mathfrak{V}) = \langle a_1, a_2, \dots \rangle$, $F_{m,n}(\mathfrak{V}) = \langle a_1, \dots, a_m, x_1, \dots, x_n \rangle$, $F_m(\mathfrak{V}) = F_{m,0}(\mathfrak{V}) = \langle a_1, \dots, a_m \rangle$. For elements of $F_{m,n}(\mathfrak{V})$, we use the notation $v(a_1, \dots, a_m; x_1, \dots, x_n) = v(a; x)$, where v is an Ω-term. It is well known that a_i and x_j can be considered as variables taking values in arbitrary Ω-algebra $G \in \mathfrak{V}$. That is, for any $v(a; x) \in F_{m,n}(\mathfrak{V})$, $g_1, \dots, g_m \in G$, and $h_1, \dots, h_n \in G$ (separated from $g_1, \dots, g_m$), the element $v(g_1, \dots, g_m; h_1, \dots, h_n) \in G$ is well defined as $\alpha(v(a; x))$, where α is the unique homomorphism of $F_{m,n}(\mathfrak{V})$ to G such that $\alpha(a_i) = g_i$ and $\alpha(x_j) = h_j$ for each $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$. If $g = (g_1, \dots, g_m)$ and $h = (h_1, \dots, h_n)$, then we sometimes write $v(g; h)$ instead of $v(g_1, \dots, g_m; h_1, \dots, h_n)$. Whenever $n = 0$, we omit the semicolon in the above notation (e.g., $v(a) = v(a;\,)$ for any $v(a;\,) \in F_\infty(\mathfrak{V})$).

Denote by $\mathfrak{O}$ the variety of all Ω-algebras. We write $F_{\infty,\infty}$, F_∞, $F_{m,n}$, and F_m instead of $F_{\infty,\infty}(\mathfrak{O})$, $F_\infty(\mathfrak{O})$, $F_{m,n}(\mathfrak{O})$, and $F_m(\mathfrak{O})$, respectively. These Ω-algebras are the Ω-term algebras over the respective sets of variables.

2.3 Probabilistic Preliminaries

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

We use the notation $\mathbf{y}_1, \dots, \mathbf{y}_n \leftarrow \mathcal{Y}$ to indicate that $\mathbf{y}_1, \dots, \mathbf{y}_n$ (denoted by upright bold letters) are independent random variables distributed according to $\mathcal{Y}$. 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 $\mathcal{Y}$ in this notation can be random. For example, suppose $(\mathcal{Y}_i \mid i \in I)$ is a probability ensemble consisting of distributions on the set Y, where the set I is finite or countably infinite. Moreover, let $\mathcal{I}$ be a probability distribution on I. Then $\mathbf{i} \leftarrow \mathcal{I}$ and $\mathbf{y} \leftarrow \mathcal{Y}_{\mathbf{i}}$ mean that the joint distribution of the random variables $\mathbf{i}$ and $\mathbf{y}$ is given by $\Pr[\mathbf{i} = i, \mathbf{y} = y] = \Pr_{\mathcal{I}}\{i\} \Pr_{\mathcal{Y}_i}\{y\}$ for each $i \in I$ and $y \in Y$.

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

For any $n \in \mathbb{N}$, we denote by $\mathcal{Y}^n$ the distribution of $(\mathbf{y}_1, \dots, \mathbf{y}_n)$, where $\mathbf{y}_1, \dots, \mathbf{y}_n \leftarrow \mathcal{Y}$. Furthermore, if Z is a nonempty finite set, then $\mathcal{U}(Z)$ denotes the uniform probability distribution on Z.

The *collision probability* $\mathrm{CP}(\mathcal{Y})$ of the probability distribution $\mathcal{Y}$ is defined by

$$\mathrm{CP}(\mathcal{Y}) = \sum_{y \in Y} (\Pr_{\mathcal{Y}}\{y\})^2 = \Pr[\mathbf{y} = \mathbf{y}'],$$

where $\mathbf{y}, \mathbf{y}' \leftarrow \mathcal{Y}$. The next lemma is well known.

Lemma 2.1. *Let Z be a finite set and let $\mathcal{Z}$ be a probability distribution on Z. Then $\mathrm{CP}(\mathcal{Z}) \geq 1/|Z|$. Furthermore, $\mathrm{CP}(\mathcal{Z}) = 1/|Z|$ if and only if $\mathcal{Z} = \mathcal{U}(Z)$.*

Proof. It is easy to see that

$$\mathrm{CP}(\mathcal{Z}) - \frac{1}{|Z|} = \sum_{z \in Z} \left(\Pr_{\mathcal{Z}}\{z\} - \frac{1}{|Z|} \right)^2.$$

The lemma follows immediately from this. □

2.4 Cryptographic Preliminaries

Let $\mathcal{Y} = (\mathcal{Y}_i \mid i \in I)$ be a probability ensemble consisting of distributions on $\{0,1\}^*$, where $I \subseteq \{0,1\}^*$. Then $\mathcal{Y}$ 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 $A(i)$ coincides with $\mathcal{Y}_i$. It is easy to see that if $\mathcal{Y}$ is polynomial-time samplable, then there exists a polynomial π satisfying $\operatorname{supp}\mathcal{Y}_i \subseteq \{0,1\}^{\leq \pi(|i|)}$ for any $i \in I$. Furthermore, let $\mathcal{Z} = (\mathcal{Z}_j \mid j \in J)$ be a probability ensemble consisting of distributions on $\{0,1\}^*$, where $J \subseteq \mathbb{N}$. Unless otherwise specified, when we speak of polynomial-time samplability of $\mathcal{Z}$, we assume that the indices are represented in binary. If, however, these indices are represented in unary, then we specify this explicitly. Thus, the ensemble $\mathcal{Z}$ is called *polynomial-time samplable when the indices are represented in unary* if there exists a probabilistic polynomial-time algorithm B such that for every $j \in J$ the distribution of $B(1^j)$ coincides with $\mathcal{Z}_j$.

Suppose K is an infinite subset of $\mathbb{N}$, D is a subset of $\{0,1\}^*$, and $\mathcal{D} = (\mathcal{D}_k \mid k \in K)$ is a probability ensemble consisting of distributions on D. We always assume that $\mathcal{D}$ is polynomial-time samplable when the indices are represented in unary. Furthermore, put $1^K = \{1^k \mid k \in K\}$. This notation is used throughout the paper.

A function $\nu\colon K \to \mathbb{R}_+ = \{r \in \mathbb{R} \mid r \geq 0\}$ is called *negligible* if for every polynomial π there exists a nonnegative integer n such that $\nu(k) \leq 1/\pi(k)$ whenever $k \in K$ and $k \geq n$. Of course, if $\epsilon, \nu\colon K \to \mathbb{R}_+$, ν is negligible, and $\epsilon(k) \leq \nu(k)$ for all sufficiently large $k \in K$, then ϵ is also negligible. Moreover, it is easy to see that if $\nu, \nu'\colon K \to \mathbb{R}_+$ are negligible and η is a polynomial, then $\nu(k) + \nu'(k)$ and $\eta(k)\nu(k)$

are negligible as functions of $k \in K$. We denote by negl an unspecified negligible function on K. Any (in)equality containing $\mathrm{negl}(k)$ 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,

$$|\Pr[A(1^k, \mathbf{r}_k) = 1] - \Pr[A(1^k, \mathbf{s}_k) = 1]| = \mathrm{negl}(k).$$

In this case, we write $\mathbf{r}_k \approx \mathbf{s}_k$. It is easy to see that computational indistinguishability is an equivalence relation. Furthermore, if A is a probabilistic polynomial-time algorithm, then $\mathbf{r}_k \approx \mathbf{s}_k$ implies $A(1^k, \mathbf{r}_k) \approx A(1^k, \mathbf{s}_k)$.

Definition 2.2 (polynomial parameter; see also [Lub96, Preliminaries])**.** A function $\xi\colon D \to \mathbb{N}$ is called a *polynomial parameter* (on D) if the function $d \mapsto 1^{\xi(d)}$ ($d \in D$) is polynomial-time computable. It is easy to see that the function ξ is a polynomial parameter if and only if it is polynomial-time computable and there exists a polynomial π satisfying $\xi(d) \le \pi(|d|)$ 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, i.e., the function $1^i \mapsto 1^{\eta(i)}$ ($i \in I$) is polynomial-time computable.

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

Let $\Phi = (\phi_d\colon Y_d \to \{0,1\}^* \,|\, d \in D)$ be a family of functions, where $Y_d \subseteq \{0,1\}^*$ 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. Also, suppose $\mathcal{Y} = (\mathcal{Y}_d \,|\, d \in D)$ is a polynomial-time samplable probability ensemble, where $\mathcal{Y}_d$ is a probability distribution on Y_d for any $d \in D$.

Definition 2.3 (one-way family)**.** The family Φ is called *one-way* with respect to $\mathcal{D}$ and $\mathcal{Y}$ 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 $\mathbf{y} \leftarrow \mathcal{Y}_{\mathbf{d}}$.

Let $\mathcal{E} = (\mathcal{E}_k \,|\, k \in K)$, where $\mathcal{E}_k$ is a probability distribution on $D \times \{0,1\}^*$. Assume that $\mathcal{E}$ is polynomial-time samplable when the indices are represented in unary. If $(\mathbf{d}, \mathbf{t}) \leftarrow \mathcal{E}_k$, where $k \in K$, then we denote by $\mathcal{E}'_k$ the distribution of the random variable $\mathbf{d}$.

Definition 2.4 (family of trapdoor functions)**.** The family Φ is said to be a *family of trapdoor functions* with respect to $\mathcal{E}$ and $\mathcal{Y}$ if it is one-way with respect to $(\mathcal{E}'_k \,|\, k \in K)$ and $\mathcal{Y}$ and there exists a deterministic polynomial-time algorithm B such that $B(1^k, d, t, z) \in \phi_d^{-1}(z)$ for all $k \in K$, $(d, t) \in \operatorname{supp} \mathcal{E}_k$, and $z \in \phi_d(Y_d)$.

Of course, if ϕ_d is a permutation of Y_d for every $d \in D$, then we use the term "*family of trapdoor permutations*" instead of "family of trapdoor functions."

Let $\alpha\colon Y \to S$ and $\beta\colon Z \to S$. A *claw* for the pair (α, β) is a pair $(y, z) \in Y \times Z$ such that $\alpha(y) = \beta(z)$.

Definition 2.5 (claw-resistant family of pairs of permutations)**.** Suppose each $d \in D$ is assigned a pair (α_d, β_d) of permutations of the set Y_d. Then the family $((\alpha_d, \beta_d) \,|\, d \in D)$ is called *claw-resistant* (or *claw-free*) with respect to $\mathcal{D}$ if the families $(\alpha_d \,|\, d \in D)$ and $(\beta_d \,|\, d \in D)$ are polynomial-time computable and for any probabilistic polynomial-time algorithm A, $\Pr[A(1^k, \mathbf{d})$ is a claw for $(\alpha_{\mathbf{d}}, \beta_{\mathbf{d}})] = \mathrm{negl}(k)$, where $\mathbf{d} \leftarrow \mathcal{D}_k$.

Definition 2.6 (family of hash functions)**.** Assume that $D = \bigsqcup_{k \in K} D_k$. For each $d \in D$, define $\kappa(d)$ to be the unique $k \in K$ such that $d \in D_k$. Suppose the following two conditions hold:

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

Furthermore, let ξ and η be polynomial parameters on K. Then a family $(\psi_d\colon \{0,1\}^{\xi(\kappa(d))} \to \{0,1\}^{\eta(\kappa(d))} \,|\, d \in D)$ of functions is said to be a *family of hash functions* if this family is polynomial-time computable and $\xi(k) > \eta(k)$ for all $k \in K$.

In what follows, we use the assumptions and notation of Definition 2.6 when speaking of families of hash functions. In this case, we also assume that for every $k \in K$, $\mathcal{D}_k$ is a probability distribution on D_k.

Recall that a *collision* for a function ϕ is a pair $(y, z) \in (\operatorname{dom} \phi)^2$ such that $y \ne z$ and $\phi(y) = \phi(z)$.

Definition 2.7 (collision-resistant family of hash functions)**.** A family $(\psi_d\colon \{0,1\}^{\xi(\kappa(d))} \to \{0,1\}^{\eta(\kappa(d))} \,|\, d \in D)$ of hash functions is called *collision-resistant* (or *collision-intractable*) with respect to $\mathcal{D}$ if for any probabilistic polynomial-time algorithm A, $\Pr[A(\mathbf{d})$ is a collision for $\psi_{\mathbf{d}}] = \operatorname{negl}(k)$, where $\mathbf{d} \leftarrow \mathcal{D}_k$.

Note that the algorithm A in Definition 2.7 can compute 1^k as $1^{\kappa(\mathbf{d})}$.

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 "claw-resistant family of pairs of permutations" and "collision-resistant family of hash functions."

***Remark* 2.8.** Let $\Psi = (\psi_d\colon \{0,1\}^{\xi(\kappa(d))} \to \{0,1\}^{\eta(\kappa(d))} \,|\, d \in D)$ be a family of hash functions. Assume that the family Ψ is collision-resistant with respect to $\mathcal{D}$. Suppose A is a probabilistic polynomial-time algorithm that on input $d \in D$ chooses $\mathrm{y}, \mathrm{y}' \leftarrow \mathcal{U}(\{0,1\}^{\xi(\kappa(d))})$ and outputs $(\mathrm{y}, \mathrm{y}')$. Let $k \in K$, $\mathbf{d} \leftarrow \mathcal{D}_k$, and $\mathbf{y}, \mathbf{y}' \leftarrow \mathcal{U}(\{0,1\}^{\xi(k)})$. Then

$$\begin{aligned}\operatorname{negl}(k) &= \Pr[A(\mathbf{d}) \text{ is a collision for } \psi_{\mathbf{d}}] = \Pr[\psi_{\mathbf{d}}(\mathbf{y}) = \psi_{\mathbf{d}}(\mathbf{y}')] - \Pr[\mathbf{y} = \mathbf{y}'] \\ &\geq \frac{1}{2^{\eta(k)}} - \frac{1}{2^{\xi(k)}} \geq \frac{1}{2^{\eta(k)}} - \frac{1}{2^{\eta(k)+1}} = \frac{2^{-\eta(k)}}{2}\end{aligned}$$

(see Lemma 2.1) and hence $2^{-\eta(k)} = \operatorname{negl}(k)$.

The next lemma is well known and can be proved using a variant of the Merkle–Damgård construction (see, e.g., [Gol04, Subsubsection 6.2.3.2]). For completeness, we give a short proof of this lemma.

Lemma 2.9. *Let $(\psi_d\colon \{0,1\}^{\xi(\kappa(d))} \to \{0,1\}^{\eta(\kappa(d))} \,|\, d \in D)$ be a family of hash functions that is collision-resistant with respect to $\mathcal{D}$. Suppose ξ' is a polynomial parameter on K satisfying $\xi'(k) > \eta(k)$ for all $k \in K$. Then there exists a family $(\psi'_d\colon \{0,1\}^{\xi'(\kappa(d))} \to \{0,1\}^{\eta(\kappa(d))} \,|\, d \in D)$ of hash functions that is collision-resistant with respect to $\mathcal{D}$.*

Proof. For each $k \in K$, put $\beta(k) = \lceil \xi'(k)/(\xi(k) - \eta(k)) \rceil$ and $\delta(k) = \beta(k)(\xi(k) - \eta(k)) - \xi'(k)$. Then β and δ are polynomial parameters on K.

Let $d \in D$, $k = \kappa(d)$, and $y \in \{0,1\}^{\xi'(k)}$. Express $y0^{\delta(k)}$ as $y_1 \dots y_{\beta(k)}$, where $y_1, \dots, y_{\beta(k)} \in \{0,1\}^{\xi(k)-\eta(k)}$. Define $\gamma_i(y) \in \{0,1\}^{\eta(k)}$ inductively as follows:

$$\gamma_0(y) = 0^{\eta(k)}, \quad \gamma_i(y) = \psi_d(y_i\gamma_{i-1}(y)) \text{ for } i \in \{1, \dots, \beta(k)\}.$$

Then we put $\psi'_d(y) = \gamma_{\beta(k)}(y)$. It is evident that $(\psi'_d \,|\, d \in D)$ is a family of hash functions.

Suppose (y, z) is a collision for ψ'_d. Let $y0^{\delta(k)} = y_1 \dots y_{\beta(k)}$ and $z0^{\delta(k)} = z_1 \dots z_{\beta(k)}$, where $y_i, z_i \in \{0,1\}^{\xi(k)-\eta(k)}$ for all $i \in \{1, \dots, \beta(k)\}$. Since $y0^{\delta(k)} \neq z0^{\delta(k)}$, there exists an $i \in \{1, \dots, \beta(k)\}$ such that $y_i\gamma_{i-1}(y) \neq z_i\gamma_{i-1}(z)$. Choose the largest such i. Then it is easy to see that $(y_i\gamma_{i-1}(y), z_i\gamma_{i-1}(z))$ is a collision for ψ_d. This implies that the family $(\psi'_d \,|\, d \in D)$ is collision-resistant with respect to $\mathcal{D}$. □

3 (Weakly) Pseudo-Free Families of Computational Ω-Algebras: Definitions, Properties, and Applications

From now on, we assume that Ω is finite and that algorithms can work with its elements. Let $\mathtt{H} = ((H_d, \rho_d, \mathcal{R}_d) \,|\, d \in D)$ be a family of triples, where H_d is an Ω-algebra, ρ_d is a function from a subset of $\{0,1\}^*$ onto H_d, and $\mathcal{R}_d$ is a probability distribution on $\operatorname{dom} \rho_d$ for any $d \in D$. If $H_d \subseteq \{0,1\}^*$ and ρ_d is the identity function on H_d, then we denote this function simply by id because its domain is clear.

3.1 Families of Computational Ω-Algebras

Definition 3.1 (family of computational Ω-algebras)**.** The family $\mathtt{H}$ is called a *family of computational Ω-algebras* if the following conditions hold:

(i) There exists a deterministic polynomial-time algorithm that, given $d \in D$ and $[g]_{\rho_d}, [h]_{\rho_d}$ (for any $g, h \in H_d$), decides whether $g = h$.

(ii) For every $\omega \in \Omega$ there exists a deterministic polynomial-time algorithm that, given $d \in D$ and $[h_1]_{\rho_d}, \dots, [h_{\operatorname{ar}\omega}]_{\rho_d}$ (where $h_1, \dots, h_{\operatorname{ar}\omega} \in H_d$), computes $[\omega(h_1, \dots, h_{\operatorname{ar}\omega})]_{\rho_d}$.

(iii) The probability ensemble $(\mathcal{R}_d \mid d \in D)$ is polynomial-time samplable.

Definition 3.2 (family having exponential size)**.** The family $\mathtt{H}$ is said to have *exponential size* if there exists a polynomial η such that $|H_d| \le 2^{\eta(|d|)}$ for all $d \in D$.

Of course, exponential size is a property of the family $(H_d \mid d \in D)$, but it is convenient to define this property for families of the form $((H_d, \rho_d, \mathcal{R}_d) \mid d \in D)$.

Definition 3.3 (polynomially bounded family)**.** We say that the family $\mathtt{H}$ is *polynomially bounded* if there exists a polynomial η such that $\operatorname{dom}\rho_d \subseteq \{0,1\}^{\le \eta(|d|)}$ for all $d \in D$.

It is obvious that if $\mathtt{H}$ is polynomially bounded, then $\mathtt{H}$ has exponential size.

Definition 3.4 (family having unique representations)**.** The family $\mathtt{H}$ is said to have *unique representations* if the function ρ_d is one-to-one for each $d \in D$.

***Remark* 3.5.** Suppose $\mathtt{H}$ has unique representations. Then we can assume that for every $d \in D$, $H_d \subseteq \{0,1\}^*$ and the unique representation of each element $h \in H_d$ is h itself. In other words, we consider the family $((\operatorname{dom}\rho_d, \operatorname{id}, \mathcal{R}_d) \mid d \in D)$ instead of $\mathtt{H}$. Here $\operatorname{dom}\rho_d$ denotes the unique Ω-algebra such that ρ_d is an isomorphism of this Ω-algebra onto H_d $(d \in D)$.

3.2 (Weakly) Pseudo-Free Families of Computational Ω-Algebras

Throughout the paper, we denote by $\mathfrak{V}$ a variety of Ω-algebras and by σ a function from a subset of $\{0,1\}^*$ onto $F_{\infty,\infty}(\mathfrak{V})$. Also, suppose $s \in \mathbb{N} \setminus \{0\}$, $H \in \mathfrak{V}$, ρ is a function from a subset of $\{0,1\}^*$ onto H, and $g \in H^m$, where $m \in \mathbb{N} \setminus \{0\}$. Then we denote by $\Sigma_s(H, \mathfrak{V}, \sigma, \rho, g)$ the set of all tuples

$$(([v_1]_\sigma, [w_1]_\sigma), \dots, ([v_s]_\sigma, [w_s]_\sigma), ([h_1]_\rho, \dots, [h_n]_\rho))$$

such that the following conditions hold:

- $n \in \mathbb{N}$, $v_i, w_i \in F_{m,n}(\mathfrak{V})$ for all $i \in \{1, \dots, s\}$, and $h_j \in H$ for all $j \in \{1, \dots, n\}$;
- the system of equations
$$v_i(a; x) = w_i(a; x), \quad i \in \{1, \dots, s\},$$
in the variables $x_1, \dots, x_n$ is unsatisfiable in $F_m(\mathfrak{V})$ (or, equivalently, in $F_\infty(\mathfrak{V})$);
- $v_i(g; h) = w_i(g; h)$ in H for each $i \in \{1, \dots, s\}$, where $h = (h_1, \dots, h_n)$.

Furthermore, let $\Sigma'_s(H, \mathfrak{V}, \sigma, g)$ be the set of all tuples $(([v_1]_\sigma, [w_1]_\sigma), \dots, ([v_s]_\sigma, [w_s]_\sigma))$ such that

- $v_i, w_i \in F_m(\mathfrak{V})$ for all $i \in \{1, \dots, s\}$,
- $v_j \ne w_j$ for some $j \in \{1, \dots, s\}$, and
- $v_i(g) = w_i(g)$ in H for each $i \in \{1, \dots, s\}$.

In other words, $(p_1, \dots, p_s) \in \Sigma'_s(H, \mathfrak{V}, \sigma, g)$ if and only if $(p_1, \dots, p_s, ()) \in \Sigma_s(H, \mathfrak{V}, \sigma, \rho, g)$ (the last condition does not depend on ρ). Thus, $\Sigma'_s(H, \mathfrak{V}, \sigma, g)$ is obtained from $\Sigma_s(H, \mathfrak{V}, \sigma, \rho, g)$ by imposing the restriction $n = 0$ and removing the last element $()$ of the tuples. Elements of $\Sigma'_1(H, \mathfrak{V}, \sigma, g)$ will be written as $([v]_\sigma, [w]_\sigma)$ instead of $(([v]_\sigma, [w]_\sigma))$. Moreover, let

$$\Sigma(H, \mathfrak{V}, \sigma, \rho, g) = \bigsqcup_{s=1}^{\infty} \Sigma_s(H, \mathfrak{V}, \sigma, \rho, g) \quad \text{and} \quad \Sigma'(H, \mathfrak{V}, \sigma, g) = \bigsqcup_{s=1}^{\infty} \Sigma'_s(H, \mathfrak{V}, \sigma, g).$$

We say that the family $\mathtt{H} = ((H_d, \rho_d, \mathcal{R}_d) \mid d \in D)$ is in $\mathfrak{V}$ if $H_d \in \mathfrak{V}$ for all $d \in D$. In this subsection, we assume that $\mathtt{H}$ is a family of computational Ω-algebras in $\mathfrak{V}$.

Definition 3.6 (pseudo-free family)**.** The family $\mathtt{H}$ is called *pseudo-free* in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ if for any polynomial π and any probabilistic polynomial-time algorithm A,

$$\Pr[A(1^k, \mathbf{d}, \mathbf{r}) \in \Sigma(H_{\mathbf{d}}, \mathfrak{V}, \sigma, \rho_{\mathbf{d}}, \rho_{\mathbf{d}}(\mathbf{r}))] = \operatorname{negl}(k),$$

where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{r} \leftarrow \mathcal{R}_{\mathbf{d}}^{\pi(k)}$.

***Remark* 3.7.** If $\mathfrak{V}$ is trivial, then $\Sigma(H, \mathfrak{V}, \sigma, \rho, g) = \emptyset$ for any $H \in \mathfrak{V}$, any function ρ from a subset of $\{0,1\}^*$ onto H, and any $g \in H^m$, where $m \in \mathbb{N} \setminus \{0\}$. Therefore, in this case the considered family $\mathtt{H}$ of computational Ω-algebras is always pseudo-free in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ.

The condition of the next definition is obtained from the condition of Definition 3.6 by replacing $\Sigma(\dots)$ by $\Sigma'(\dots)$.

Definition 3.8 (weakly pseudo-free family)**.** The family $\mathtt{H}$ is called *weakly pseudo-free* in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ if for any polynomial π and any probabilistic polynomial-time algorithm A,

$$\Pr[A(1^k, \mathbf{d}, \mathbf{r}) \in \Sigma'(H_{\mathbf{d}}, \mathfrak{V}, \sigma, \rho_{\mathbf{d}}(\mathbf{r}))] = \operatorname{negl}(k),$$

where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{r} \leftarrow \mathcal{R}_{\mathbf{d}}^{\pi(k)}$.

It is evident that if $\mathtt{H}$ is pseudo-free in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ, then $\mathtt{H}$ is weakly pseudo-free in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ.

***Remark* 3.9.** Let $s \in \mathbb{N} \setminus \{0\}$. Define the notion of *$s$-pseudo-freeness* (resp., *weak s-pseudo-freeness*) in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ by replacing $\Sigma(\dots)$ by $\Sigma_s(\dots)$ in Definition 3.6 (resp., $\Sigma'(\dots)$ by $\Sigma'_s(\dots)$ in Definition 3.8). Note that in many works (see, e.g., [Hoh03, Riv04a, Riv04b, Mic10, JB09]), pseudo-freeness (resp., weak pseudo-freeness) is understood as 1-pseudo-freeness (resp., weak 1-pseudo-freeness). It is evident that any pseudo-free (resp., weakly pseudo-free) family of computational Ω-algebras in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ is also s-pseudo-free (resp., weakly s-pseudo-free) in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ. Rivest remarked that in the variety of all groups, 1-pseudo-freeness is equivalent to pseudo-freeness (see [Riv04a, Subsection 5.1]). Micciancio obtained the same result for the variety of all abelian groups (see [Mic10, Corollary 1]). Moreover, Anokhin proved that in the variety of all elementary abelian p-groups, where p is an arbitrary prime, any weakly 1-pseudo-free family of computational groups is pseudo-free (see [Ano17, Theorem 3.7]). Note that these results hold only under certain additional conditions.

Assume that there exists a deterministic polynomial-time algorithm that, given $[v]_\sigma$ and $[w]_\sigma$ (for any $v, w \in F_\infty(\mathfrak{V})$), decides whether $v = w$. (This is true in many important cases.) Then it is easy to see that weak 1-pseudo-freeness in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ is equivalent to weak pseudo-freeness in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ.

***Remark* 3.10** (see also [Ano13, Remark 3.6])**.** Assume that the family $\mathtt{H}$ is weakly 1-pseudo-free in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ. Let D' be a subset of D such that $\{H_d \,|\, d \in D'\}$ does not generate the variety $\mathfrak{V}$. Then there exist distinct elements $v, w \in F_m(\mathfrak{V})$ (for some $m \in \mathbb{N} \setminus \{0\}$) such that $v(g) = w(g)$ for all $d \in D'$ and $g \in H_d^m$. It is evident that $([v]_\sigma, [w]_\sigma) \in \Sigma'_1(H_d, \mathfrak{V}, \sigma, g)$ for every $d \in D'$ and $g \in H_d^m$. This implies that $\Pr_{\mathcal{D}_k} D' = \operatorname{negl}(k)$. Thus, we see that if D' is a subset of D such that $\Pr_{\mathcal{D}_k} D'$ is not negligible as a function of $k \in K$ (in particular, if $D' = D$), then $\{H_d \,|\, d \in D'\}$ generates the variety $\mathfrak{V}$. This shows that the family $\mathtt{H}$ can be weakly 1-pseudo-free (with respect to $\mathcal{D}$ and σ) only in the variety generated by $\{H_d \,|\, d \in D\}$.

***Remark* 3.11.** For each $d \in D$, let S_d be a subset of $\operatorname{dom} \rho_d$ such that $\rho_d(S_d) = H_d$ and $\operatorname{supp} \mathcal{R}_d \subseteq S_d$. Also, assume that for every $\omega \in \Omega$ there exists a deterministic polynomial-time algorithm that, given $d \in D$ and $[h_1]_{\rho_d}, \dots, [h_{\operatorname{ar}\omega}]_{\rho_d} \in S_d$ (where $h_1, \dots, h_{\operatorname{ar}\omega} \in H_d$), computes $[\omega(h_1, \dots, h_{\operatorname{ar}\omega})]_{\rho_d} \in S_d$. Then $\mathtt{H}' = ((H_d, \rho_d|_{S_d}, \mathcal{R}_d) \,|\, d \in D)$ is a family of computational Ω-algebras in $\mathfrak{V}$. Moreover, if $\mathtt{H}$ is pseudo-free (resp., weakly pseudo-free) in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ, then $\mathtt{H}'$ is also pseudo-free (resp., weakly pseudo-free) in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ. For weak pseudo-freeness, the converse also holds.

3.3 Two Examples of the Function σ

In this subsection, we introduce two functions nat and SLP. In what follows, we will often assume that $\sigma = \mathrm{nat}$ or $\sigma = \mathrm{SLP}$.

Example 3.12 (natural representation)**.** Denote by $T_{\infty,\infty}$ the Ω-term algebra over the set $\{\mathsf{a}_1, \mathsf{a}_2, \dots \mathsf{x}_1, \mathsf{x}_2, \dots\}$ of distinct variables. Let $v(a;x)$ be an arbitrary element of $F_{\infty,\infty}(\mathfrak{V})$, where $v \in T_{\infty,\infty}$. In general, unless $\mathfrak{V} = \mathfrak{O}$, the term v is not uniquely determined by $v(a;x)$. We represent $v(a;x)$ by the term v written in Polish notation. Moreover, we encode each variable b_i by $\overline{\mathsf{b}_i} = \mathsf{b}\operatorname{bin} i$, where $\mathsf{b} \in \{\mathsf{a}, \mathsf{x}\}$, $i \in \mathbb{N} \setminus \{0\}$, and $\operatorname{bin} i$ is the binary representation of i without leading zeros. More formally, consider the term v as a string over the alphabet consisting of all symbols from $\Omega \sqcup \{\mathsf{b}_i \mid \mathsf{b} \in \{\mathsf{a}, \mathsf{x}\}, i \in \mathbb{N} \setminus \{0\}\}$, parentheses, and comma. Let $\overline{v}$ be obtained from v by removing all parentheses and commas and replacing all occurrences of b_i by $\overline{\mathsf{b}_i}$ for every $\mathsf{b} \in \{\mathsf{a}, \mathsf{x}\}$ and $i \in \mathbb{N} \setminus \{0\}$, where $\overline{\mathsf{b}_i}$ is defined above. Then $v \mapsto \overline{v}$ is a one-to-one function from $T_{\infty,\infty}$ to the set of all strings over the finite alphabet $\Omega \sqcup \{\mathsf{a}, \mathsf{x}, 0, 1\}$. It is convenient to use $\overline{v}$ as a representation of $v(a;x)$ for computational purposes. We call this representation *natural* and denote the function $\overline{v} \mapsto v(a;x)$, where $v \in T_{\infty,\infty}$, by nat. Of course, the function nat is well defined. For each $m \in \mathbb{N}$, let nat_m be the restriction of nat to $\langle \mathsf{a}_1, \dots, \mathsf{a}_m \rangle$. Then nat and nat_m are functions onto $F_{\infty,\infty}(\mathfrak{V})$ and $F_m(\mathfrak{V})$, respectively.

Assume that $\mathfrak{V} = \mathfrak{O}$. In this case, the function nat is one-to-one. For every $i \in \mathbb{N} \setminus \{0\}$, we identify a_i with a_i and x_i with x_i. Then $\mathrm{nat}^{-1}(w) = \overline{w}$ for all $w \in F_{\infty,\infty}$. This allows us to simplify the notation.

Example 3.13 (representation by straight-line programs)**.** By a *straight-line program* over $F_{\infty,\infty}(\mathfrak{V})$ we mean a sequence $(u_1, \dots, u_n)$ of tuples such that $n \in \mathbb{N} \setminus \{0\}$ and for any $i \in \{1, \dots, n\}$, either $u_i = (b, m)$, where $b \in \{a, x\}$ and $m \in \mathbb{N} \setminus \{0\}$, or $u_i = (\omega, m_1, \dots, m_{\operatorname{ar}\omega})$, where $\omega \in \Omega$ and $m_1, \dots, m_{\operatorname{ar}\omega} \in \{1, \dots, i-1\}$. Here a and x are considered as symbols that are not in Ω. Any straight-line program $u = (u_1, \dots, u_n)$ over $F_{\infty,\infty}(\mathfrak{V})$ naturally defines the sequence $(v_1, \dots, v_n)$ of elements of $F_{\infty,\infty}(\mathfrak{V})$ by induction. Namely, for every $i \in \{1, \dots, n\}$, we put $v_i = b_m$ if $u_i = (b, m)$ and $v_i = \omega(v_{m_1}, \dots, v_{m_{\operatorname{ar}\omega}})$ if $u_i = (\omega, m_1, \dots, m_{\operatorname{ar}\omega})$, where b, m, ω, and $m_1, \dots, m_{\operatorname{ar}\omega}$ are as above. The straight-line program u is said to represent the element v_n. We denote by SLP the function $u \mapsto v_n$, where $u = (u_1, \dots, u_n)$ is a straight-line program over $F_{\infty,\infty}(\mathfrak{V})$ and v_n is defined above. It is evident that SLP is a function onto $F_{\infty,\infty}(\mathfrak{V})$. Note that this method of representation (for elements of the free group) was used in [Hoh03].

***Remark* 3.14.** Assume that $\mathfrak{V} = \mathfrak{O}$. Unlike nat, the function SLP is not one-to-one. However, there exists a deterministic polynomial-time algorithm that, given $[v]_{\mathrm{SLP}}$ and $[w]_{\mathrm{SLP}}$ (where $v, w \in F_{\infty,\infty}$), decides whether $v = w$. This algorithm can be easily constructed using the following observation: For any $b, c \in \{a, x\}$, $i, j \in \mathbb{N} \setminus \{0\}$, $\omega, \mu \in \Omega$, and $v_1, \dots, v_{\operatorname{ar}\omega}, w_1, \dots, w_{\operatorname{ar}\mu} \in F_{\infty,\infty}$, we have

- $b_i = c_j$ if and only if $b = c$ and $i = j$;
- $b_i \neq \omega(v_1, \dots, v_{\operatorname{ar}\omega})$;
- $\omega(v_1, \dots, v_{\operatorname{ar}\omega}) = \mu(w_1, \dots, w_{\operatorname{ar}\mu})$ if and only if $\omega = \mu$ and $v_i = w_i$ for all $i \in \{1, \dots, \operatorname{ar}\omega\}$.

***Remark* 3.15.** As in Remark 3.14, assume that $\mathfrak{V} = \mathfrak{O}$. Let $u = (u_1, \dots, u_n)$ be a straight-line program over $F_{\infty,\infty}$ and let $(v_1, \dots, v_n)$ be the sequence of elements of $F_{\infty,\infty}$ naturally defined by u as in Example 3.13, i.e., $v_i = \mathrm{SLP}(u_1, \dots, u_i)$ for all $i \in \{1, \dots, n\}$. Then an easy induction on n shows that $\operatorname{subt}(v_n) \subseteq \{v_1, \dots, v_n\}$. Moreover, there exists a deterministic polynomial-time algorithm that, given u, computes $(j_1, \dots, j_m)$ such that $1 \le j_1 < \dots < j_m \le n$ and $\operatorname{subt}(v_n) = \{v_{j_1}, \dots, v_{j_m}\}$.

***Remark* 3.16.** It is easy to see that, given $[w]_{\mathrm{nat}}$ for arbitrary $w \in F_{\infty,\infty}(\mathfrak{V})$, one can compute $[w]_{\mathrm{SLP}}$ in polynomial time. Therefore pseudo-freeness (resp., weak pseudo-freeness) in $\mathfrak{V}$ with respect to $\mathcal{D}$ and SLP implies pseudo-freeness (resp., weak pseudo-freeness) in $\mathfrak{V}$ with respect to $\mathcal{D}$ and nat. The same holds for (weak) s-pseudo-freeness for arbitrary $s \in \mathbb{N} \setminus \{0\}$. However, the inverse transformation $[w]_{\mathrm{SLP}} \mapsto [w]_{\mathrm{nat}}$, in general, cannot be performed in polynomial time. This is because the unique representation $[w]_{\mathrm{nat}}$ (when $\mathfrak{V} = \mathfrak{O}$) can have length exponential in the length of the binary representation of $[w]_{\mathrm{SLP}}$. For example, assume that $\mathfrak{V} = \mathfrak{O}$ and $\Omega \ni \zeta, \omega$, where $\operatorname{ar}\zeta = 0$ and $\operatorname{ar}\omega = 2$. For each $n \in \mathbb{N}$, let $w_n = \mathrm{SLP}((\zeta), (\omega, 1, 1), \dots, (\omega, n, n))$. This means that $w_0 = \zeta$ and $w_{n+1} = \omega(w_n, w_n)$. Then an induction on n shows that the length of $\overline{w_n} = \mathrm{nat}^{-1}(w_n)$ (as a string over Ω) is $2^{n+1} - 1$.

3.4 Certain Families of $\mathfrak{V}$-Free Ω-Algebras Are Pseudo-Free

The next lemma is similar to Lemma 3.8 in [Ano13].

Lemma 3.17. *For each $u \in 1^K$, suppose τ_u is a function from a subset of $\{0,1\}^*$ onto $F_{\gamma(u)}(\mathfrak{V})$ (where $\gamma\colon 1^K \to \mathbb{N} \setminus \{0\}$) and $\mathcal{F}_u$ is a probability distribution on $\operatorname{dom}\tau_u$. Assume that the following conditions hold:*

(i) $\mathtt{F} = ((F_{\gamma(u)}(\mathfrak{V}), \tau_u, \mathcal{F}_u) \mid u \in 1^K)$ *is a family of computational* Ω*-algebras;*

(ii) $\tau_u(\operatorname{supp} \mathcal{F}_u) \subseteq \{a_1, \dots, a_{\gamma(u)}\}$ *for all* $u \in 1^K$*;*

(iii) $\mathrm{CP}(\tau_{1^k}(\mathcal{F}_{1^k})) = \mathrm{negl}(k)$.

Then $\mathtt{F}$ *is pseudo-free in* $\mathfrak{V}$ *with respect to* $(\mathcal{U}(\{1^k\}) \mid k \in K)$ *and* σ*.*

Proof. Suppose π is a polynomial and A is a probabilistic polynomial-time algorithm. Let $k \in K$ and $f_1, \dots, f_{\pi(k)} \in \operatorname{supp} \mathcal{F}_{1^k}$. Assume that

$$A(1^k, 1^k, (f_1, \dots, f_{\pi(k)})) \in \Sigma(F_{\gamma(1^k)}(\mathfrak{V}), \mathfrak{V}, \sigma, \tau_{1^k}, (\tau_{1^k}(f_1), \dots, \tau_{1^k}(f_{\pi(k)}))).$$

Then, in particular, there exist $v_1, \dots, v_s, w_1, \dots, w_s \in F_{\pi(k),n}(\mathfrak{V})$ (for some $s \in \mathbb{N} \setminus \{0\}$ and $n \in \mathbb{N}$) such that the system of equations

$$v_i(a_1, \dots, a_{\pi(k)}; x_1, \dots, x_n) = w_i(a_1, \dots, a_{\pi(k)}; x_1, \dots, x_n), \quad i \in \{1, \dots, s\},$$

is unsatisfiable in $F_\infty(\mathfrak{V})$, but the system

$$v_i(\tau_{1^k}(f_1), \dots, \tau_{1^k}(f_{\pi(k)}); x_1, \dots, x_n) = w_i(\tau_{1^k}(f_1), \dots, \tau_{1^k}(f_{\pi(k)}); x_1, \dots, x_n), \quad i \in \{1, \dots, s\},$$

is satisfiable even in $F_{\gamma(1^k)}(\mathfrak{V})$. Here, of course, $x_1, \dots, x_n$ are considered as variables. Since $\{\tau_{1^k}(f_1), \dots, \tau_{1^k}(f_{\pi(k)})\} \subseteq \{a_1, \dots, a_{\gamma(1^k)}\}$ (see Condition (ii)), this implies that $\tau_{1^k}(f_1), \dots, \tau_{1^k}(f_{\pi(k)})$ are not distinct. Hence,

$$\begin{aligned}&\Pr[A(1^k, 1^k, (\mathbf{f}_1, \dots, \mathbf{f}_{\pi(k)})) \in \Sigma(F_{\gamma(1^k)}(\mathfrak{V}), \mathfrak{V}, \sigma, \tau_{1^k}, (\tau_{1^k}(\mathbf{f}_1), \dots, \tau_{1^k}(\mathbf{f}_{\pi(k)})))] \\ &\quad \le \Pr[\tau_{1^k}(\mathbf{f}_1), \dots, \tau_{1^k}(\mathbf{f}_{\pi(k)}) \text{ are not distinct}] \le \frac{\pi(k)(\pi(k)-1)}{2} \mathrm{CP}(\tau_{1^k}(\mathcal{F}_{1^k})) = \mathrm{negl}(k),\end{aligned}$$

where $\mathbf{f}_1, \dots, \mathbf{f}_{\pi(k)} \leftarrow \mathcal{F}_{1^k}$. (Here we use Condition (iii).) Thus, the family $\mathtt{F}$ is pseudo-free in $\mathfrak{V}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and σ. □

In the next corollary, $\overline{a_i} = \mathrm{nat}^{-1}(a_i)$ (see Example 3.12).

Corollary 3.18. *Let* η *be a polynomial parameter on* K *such that* $2^{-\eta(k)} = \mathrm{negl}(k)$*. Then*

$$((F_{2^{\eta(|u|)}}, \mathrm{nat}_{2^{\eta(|u|)}}, \mathcal{U}(\{\overline{a_1}, \dots, \overline{a_{2^{\eta(|u|)}}}\})) \mid u \in 1^K)$$

is a pseudo-free family of computational Ω*-algebras in* $\mathfrak{O}$ *with respect to* $(\mathcal{U}(\{1^k\}) \mid k \in K)$ *and* σ*.*

Corollary 3.18 follows from Lemma 3.17 because $\mathrm{CP}(\mathcal{U}(\{a_1, \dots, a_{2^{\eta(k)}}\})) = 2^{-\eta(k)} = \mathrm{negl}(k)$ (see Lemma 2.1).

3.5 (Weakly) Pseudo-Free Families of Quotient Algebras

In this subsection, as in Subsection 3.2, we assume that the family $\mathtt{H} = ((H_d, \rho_d, \mathcal{R}_d) \mid d \in D)$ is a family of computational Ω-algebras in $\mathfrak{V}$.

Definition 3.19 (σ-compatible family)**.** We call the family $\mathtt{H}$ *σ-compatible* if there exists a deterministic polynomial-time algorithm that, given

$$(d, [u]_\sigma, ([g_1]_{\rho_d}, \dots, [g_m]_{\rho_d}), ([h_1]_{\rho_d}, \dots, [h_n]_{\rho_d}))$$

for any $d \in D$, $u \in F_{m,n}(\mathfrak{V})$ $(m, n \in \mathbb{N})$, and $g_1, \dots, g_m, h_1, \dots, h_n \in H_d$, computes $[u(g_1, \dots, g_m; h_1, \dots, h_n)]_{\rho_d}$.

Note that if the family $\mathtt{H}$ is polynomially bounded, then it is SLP-compatible and hence nat-compatible (see Remark 3.16).

In Lemmas 3.20 and 3.21 below, let $(\mathcal{E}_d \mid d \in D)$ be a polynomial-time samplable probability ensemble such that for every $d \in D$, $\mathcal{E}_d$ is a probability distribution on a set $E_d \subseteq \{0,1\}^{\le \xi(|d|)}$, where ξ is a fixed polynomial. (We can let $E_d = \operatorname{supp} \mathcal{E}_d$ for all $d \in D$.) Furthermore, suppose each pair (d, e) with $d \in D$ and $e \in E_d$ is assigned a congruence $\theta_{d,e}$ on H_d. Finally, we denote by $\mathcal{D}'_k$ the distribution of the random variable $(\mathbf{d}, \mathbf{e})$, where $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{e} \leftarrow \mathcal{E}_{\mathbf{d}}$ $(k \in K)$.

The next lemma is similar to Theorem 3.7 in [Ano13].

Lemma 3.20. *Assume that the following conditions hold:*

(i) *There exists a deterministic polynomial-time algorithm that, given $d \in D$, $e \in E_d$, and $[g]_{\rho_d}, [h]_{\rho_d}$ (where $g, h \in H_d$), decides whether $(g, h) \in \theta_{d,e}$.*

(ii) *If $\mathbf{d} \leftarrow \mathcal{D}_k$ and $\mathbf{e} \leftarrow \mathcal{E}_{\mathbf{d}}$, then for any probabilistic polynomial-time algorithm A,*

$$\Pr[A(1^k, \mathbf{d}, \mathbf{e}) = ([y]_{\rho_\mathbf{d}}, [z]_{\rho_\mathbf{d}}) \text{ s.t. } (y, z) \in \theta^{\neq}_{\mathbf{d},\mathbf{e}}] = \operatorname{negl}(k).$$

Also, suppose the family `H` *is σ-compatible and pseudo-free (resp., weakly pseudo-free) in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ. Then* `H'` $= ((H_d/\theta_{d,e}, \rho_d/\theta_{d,e}, \mathcal{R}_d) \,|\, d \in D,\ e \in E_d)$ *is a pseudo-free (resp., weakly pseudo-free) family of computational Ω-algebras in $\mathfrak{V}$ with respect to $(\mathcal{D}'_k \,|\, k \in K)$ and σ.*

Proof. It is evident that for any $d \in D$, $e \in E_d$, and $h \in H_d$, the set $(\rho_d/\theta_{d,e})^{-1}(h/\theta_{d,e})$, where $h/\theta_{d,e}$ is considered as an element of $H_d/\theta_{d,e}$, coincides with the set $\rho_d^{-1}(h/\theta_{d,e})$, where $h/\theta_{d,e}$ is considered as a subset of H_d. This together with Condition (i) implies that `H'` is a family of computational Ω-algebras.

We consider only the case where `H` is pseudo-free. When `H` is weakly pseudo-free, the proof is the same, *mutatis mutandis*. Suppose π is a polynomial and A is a probabilistic polynomial-time algorithm. Let B be a probabilistic polynomial-time algorithm that on input $(1^k, d, r)$ for arbitrary $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, and $r \in (\operatorname{supp} \mathcal{R}_d)^{\pi(k)}$ chooses $\mathrm{e} \leftarrow \mathcal{E}_d$, runs A on input $(1^k, (d, \mathrm{e}), r)$, and returns the output of A (if it exists). Furthermore, suppose C is a probabilistic polynomial-time algorithm that on input $(1^k, d, e)$ for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, and $e \in \operatorname{supp} \mathcal{E}_d$ proceeds as follows:

1. Choose $\mathrm{r} \leftarrow \mathcal{R}_d^{\pi(k)}$.
2. Run A on input $(1^k, (d, e), \mathrm{r})$. Assume that the output is
$$(([v_1]_\sigma, [w_1]_\sigma), \ldots, ([v_s]_\sigma, [w_s]_\sigma), (q_1, \ldots, q_n)), \tag{2}$$
where $s \in \mathbb{N} \setminus \{0\}$, $n \in \mathbb{N}$, $v_i, w_i \in F_{\pi(k),n}(\mathfrak{V})$ for all $i \in \{1, \ldots, s\}$, and $q_j = [h_j]_{\rho_d} = [h_j/\theta_{d,e}]_{\rho_d/\theta_{d,e}}$ ($h_j \in H_d$) for all $j \in \{1, \ldots, n\}$. If this is not true, then C fails.
3. Compute $[v_i(\rho_d(\mathrm{r}); h)]_{\rho_d}$ and $[w_i(\rho_d(\mathrm{r}); h)]_{\rho_d}$ for all $i \in \{1, \ldots, s\}$, where $h = (h_1, \ldots, h_n)$. (This can be done in deterministic polynomial time because `H` is σ-compatible.)
4. If there exists an $i \in \{1, \ldots, s\}$ such that $v_i(\rho_d(\mathrm{r}); h) \neq w_i(\rho_d(\mathrm{r}); h)$, then output $([v_i(\rho_d(\mathrm{r}); h)]_{\rho_d}, [w_i(\rho_d(\mathrm{r}); h)]_{\rho_d})$ for some such i. Otherwise, the algorithm C fails.

Assume that the algorithm A is invoked by B or C on input $(1^k, (d, e), r)$ (where $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, $e \in \operatorname{supp} \mathcal{E}_d$, and $r \in (\operatorname{supp} \mathcal{R}_d)^{\pi(k)}$) and that the output of A (denoted by u) is in $\Sigma(H_d/\theta_{d,e}, \mathfrak{V}, \sigma, \rho_d/\theta_{d,e}, (\rho_d/\theta_{d,e})(r))$. In particular, this means that u has the form (2) and $(v_i(\rho_d(r); h), w_i(\rho_d(r); h)) \in \theta_{d,e}$ for all $i \in \{1, \ldots, s\}$. If $v_i(\rho_d(r); h) = w_i(\rho_d(r); h)$ for every $i \in \{1, \ldots, s\}$, then the algorithm B outputs $u \in \Sigma(H_d, \mathfrak{V}, \sigma, \rho_d, \rho_d(r))$. Otherwise, the algorithm C outputs a pair $([y]_{\rho_d}, [z]_{\rho_d})$ such that $(y, z) \in \theta^{\neq}_{d,e}$. Hence,

$$\begin{aligned}\Pr[A(1^k, (\mathbf{d}, \mathbf{e}), \mathbf{r}) \in \Sigma(H_\mathbf{d}/\theta_{\mathbf{d},\mathbf{e}}, \mathfrak{V}, \sigma, \rho_\mathbf{d}/\theta_{\mathbf{d},\mathbf{e}}, (\rho_\mathbf{d}/\theta_{\mathbf{d},\mathbf{e}})(\mathbf{r}))] &\le \Pr[B(1^k, \mathbf{d}, \mathbf{r}) \in \Sigma(H_\mathbf{d}, \mathfrak{V}, \sigma, \rho_\mathbf{d}, \rho_\mathbf{d}(\mathbf{r}))] \\ + \Pr[C(1^k, \mathbf{d}, \mathbf{e}) = ([y]_{\rho_\mathbf{d}}, [z]_{\rho_\mathbf{d}}) \text{ s.t. } (y, z) \in \theta^{\neq}_{\mathbf{d},\mathbf{e}}] &= \operatorname{negl}(k) + \operatorname{negl}(k) = \operatorname{negl}(k),\end{aligned}$$

where $k \in K$, $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{e} \leftarrow \mathcal{E}_\mathbf{d}$, and $\mathbf{r} \leftarrow \mathcal{R}_\mathbf{d}^{\pi(k)}$. Thus, `H'` is pseudo-free in $\mathfrak{V}$ with respect to $(\mathcal{D}'_k \,|\, k \in K)$ and σ. □

Lemma 3.21. *Assume that the following conditions hold:*

(i) *There exists a deterministic polynomial-time algorithm that, given $d \in D$, $e \in E_d$, $[g]_{\rho_d}$, and $[h]_{\rho_d}$ (where $g, h \in H_d$), decides whether $(g, h) \in \theta_{d,e}$ (as in Lemma 3.20).*

(ii) *For any polynomial π and any probabilistic polynomial-time algorithm A,*

$$\Pr[A(1^k, \mathbf{d}, \mathbf{e}, \mathbf{r}) = ([v]_\sigma, [w]_\sigma) \text{ s.t. } v, w \in F_{\pi(k)}(\mathfrak{V}) \text{ and } (v(\rho_\mathbf{d}(\mathbf{r})), w(\rho_\mathbf{d}(\mathbf{r}))) \in \theta^{\neq}_{\mathbf{d},\mathbf{e}}] = \operatorname{negl}(k),$$

where $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{e} \leftarrow \mathcal{E}_\mathbf{d}$, and $\mathbf{r} \leftarrow \mathcal{R}_\mathbf{d}^{\pi(k)}$.

Also, suppose the family $\mathtt{H}$ *is weakly pseudo-free in* $\mathfrak{V}$ *with respect to* $\mathcal{D}$ *and* σ*. Then* $\mathtt{H}' = ((H_d/\theta_{d,e}, \rho_d/\theta_{d,e}, \mathcal{R}_d) \mid d \in D,\ e \in E_d)$ *is a weakly pseudo-free family of computational* Ω*-algebras in* $\mathfrak{V}$ *with respect to* $(\mathcal{D}'_k \mid k \in K)$ *and* σ*.*

Proof. As in the proof of Lemma 3.20, we see that $\mathtt{H}'$ is a family of computational Ω-algebras.

Let π be a polynomial and let A be a probabilistic polynomial-time algorithm. Suppose B is a probabilistic polynomial-time algorithm that on input $(1^k, d, r)$ for arbitrary $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, and $r \in (\operatorname{supp} \mathcal{R}_d)^{\pi(k)}$ chooses $\mathrm{e} \leftarrow \mathcal{E}_d$, runs A on input $(1^k, (d, \mathrm{e}), r)$, and returns the output of A (if it exists). Furthermore, let C be a probabilistic polynomial-time algorithm that on input $(1^k, d, e, r)$ for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, $e \in \operatorname{supp} \mathcal{E}_d$, and $r \in (\operatorname{supp} \mathcal{R}_d)^{\pi(k)}$ proceeds as follows:

1. Run A on input $(1^k, (d, e), \mathrm{r})$. Assume that the output is $(p_1, \dots, p_s)$, where $s \in \mathbb{N} \setminus \{0\}$ and $p_i \in (\operatorname{dom} \sigma)^2$ for all $i \in \{1, \dots, s\}$. If this is not true, then C fails.

2. Choose $\mathrm{j} \leftarrow \mathcal{U}(\{1, \dots, 2^{\lceil \log_2 s \rceil}\})$.

3. If $\mathrm{j} \in \{1, \dots, s\}$, then output p_{j}. Otherwise, the algorithm C fails.

Assume that the algorithm A is invoked by B or C on input $(1^k, (d, e), r)$ (where $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, $e \in \operatorname{supp} \mathcal{E}_d$, and $r \in (\operatorname{supp} \mathcal{R}_d)^{\pi(k)}$) and that the output of A is

$$u = (([v_1]_\sigma, [w_1]_\sigma), \dots, ([v_s]_\sigma, [w_s]_\sigma)) \in \Sigma'(H_d/\theta_{d,e}, \mathfrak{V}, \sigma, (\rho_d/\theta_{d,e})(r)).$$

This means that $s \in \mathbb{N} \setminus \{0\}$, $v_i, w_i \in F_{\pi(k)}(\mathfrak{V})$ for all $i \in \{1, \dots, s\}$, $v_j \neq w_j$ for some $j \in \{1, \dots, s\}$, and $(v_i(\rho_d(r)), w_i(\rho_d(r))) \in \theta_{d,e}$ for each $i \in \{1, \dots, s\}$. For brevity, put

$$\Pi(k, d, e, r) = \{([v]_\sigma, [w]_\sigma) \mid v, w \in F_{\pi(k)}(\mathfrak{V}),\ (v(\rho_d(r)), w(\rho_d(r))) \in \theta^{\neq}_{d,e}\}.$$

Moreover, let $\mathsf{v}^u(g) = (v_1(g), \dots, v_s(g))$ and $\mathsf{w}^u(g) = (w_1(g), \dots, w_s(g))$ for arbitrary $g \in H_d^{\pi(k)}$. Choose a polynomial η satisfying $2^{\lceil \log_2 s \rceil} \leq \eta(k)$. If $\mathsf{v}^u(\rho_d(r)) = \mathsf{w}^u(\rho_d(r))$, then the algorithm B outputs $u \in \Sigma'(H_d, \mathfrak{V}, \sigma, \rho_d(r))$. Assume that $\mathsf{v}^u(\rho_d(r)) \neq \mathsf{w}^u(\rho_d(r))$. Then it is evident that the algorithm C outputs an element of $\Pi(k, d, e, r)$ if and only if $\mathrm{j} \in \{1, \dots, s\}$ and $v_{\mathrm{j}}(\rho_d(r)) \neq w_{\mathrm{j}}(\rho_d(r))$, where j is defined in Step 2 of C. This shows that

$$\begin{aligned} &\Pr[C(1^k, d, e, r) \in \Pi(k, d, e, r) \mid A(1^k, (d, e), r) = u] \\ &\quad = \Pr[\mathbf{j} \in \{1, \dots, s\},\ v_{\mathbf{j}}(\rho_d(r)) \neq w_{\mathbf{j}}(\rho_d(r))] \geq \frac{1}{2^{\lceil \log_2 s \rceil}} \geq \frac{1}{\eta(k)}, \end{aligned}$$

where $\mathbf{j} \leftarrow \mathcal{U}(\{1, \dots, 2^{\lceil \log_2 s \rceil}\})$. (The random bits of the algorithm A are considered as a part of the random bits of the algorithm C.) Hence,

$$\Pr[A(1^k, (d, e), r) = u] \leq \eta(k) \Pr[C(1^k, d, e, r) \in \Pi(k, d, e, r),\ A(1^k, (d, e), r) = u]$$

and

$$\begin{aligned} &\Pr[A(1^k, (d, e), r) = u' \in \Sigma'(H_d/\theta_{d,e}, \mathfrak{V}, \sigma, (\rho_d/\theta_{d,e})(r)) \text{ s.t. } \mathsf{v}^{u'}(\rho_d(r)) \neq \mathsf{w}^{u'}(\rho_d(r))] \\ &\quad \leq \Pr[C(1^k, d, e, r) \in \Pi(k, d, e, r)]. \end{aligned}$$

Therefore we have

$$\begin{aligned} &\Pr[A(1^k, (\mathbf{d}, \mathbf{e}), \mathbf{r}) \in \Sigma'(H_{\mathbf{d}}/\theta_{\mathbf{d},\mathbf{e}}, \mathfrak{V}, \sigma, (\rho_{\mathbf{d}}/\theta_{\mathbf{d},\mathbf{e}})(\mathbf{r}))] \\ &\quad = \Pr[A(1^k, (\mathbf{d}, \mathbf{e}), \mathbf{r}) = u' \in \Sigma'(H_{\mathbf{d}}/\theta_{\mathbf{d},\mathbf{e}}, \mathfrak{V}, \sigma, (\rho_{\mathbf{d}}/\theta_{\mathbf{d},\mathbf{e}})(\mathbf{r})) \text{ s.t. } \mathsf{v}^{u'}(\rho_{\mathbf{d}}(\mathbf{r})) = \mathsf{w}^{u'}(\rho_{\mathbf{d}}(\mathbf{r}))] \\ &\qquad + \Pr[A(1^k, (\mathbf{d}, \mathbf{e}), \mathbf{r}) = u' \in \Sigma'(H_{\mathbf{d}}/\theta_{\mathbf{d},\mathbf{e}}, \mathfrak{V}, \sigma, (\rho_{\mathbf{d}}/\theta_{\mathbf{d},\mathbf{e}})(\mathbf{r})) \text{ s.t. } \mathsf{v}^{u'}(\rho_{\mathbf{d}}(\mathbf{r})) \neq \mathsf{w}^{u'}(\rho_{\mathbf{d}}(\mathbf{r}))] \\ &\quad \leq \Pr[B(1^k, \mathbf{d}, \mathbf{r}) \in \Sigma'(H_{\mathbf{d}}, \mathfrak{V}, \sigma, \rho_{\mathbf{d}}(\mathbf{r}))] + \eta(k) \Pr[C(1^k, \mathbf{d}, \mathbf{e}, \mathbf{r}) \in \Pi(k, \mathbf{d}, \mathbf{e}, \mathbf{r})] \\ &\quad = \operatorname{negl}(k) + \eta(k) \operatorname{negl}(k) = \operatorname{negl}(k), \end{aligned}$$

where $k \in K$, $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{e} \leftarrow \mathcal{E}_{\mathbf{d}}$, and $\mathbf{r} \leftarrow \mathcal{R}_{\mathbf{d}}^{\pi(k)}$. Thus, $\mathtt{H}'$ is weakly pseudo-free in $\mathfrak{V}$ with respect to $(\mathcal{D}'_k \mid k \in K)$ and σ. □

3.6 Some Applications of Pseudo-Free Families

In this subsection, we construct some cryptographic primitives from pseudo-free (more specifically, 1-pseudo-free) families satisfying certain additional conditions. We do not discuss the existence of such families. The purpose of this subsection is to demonstrate the potential of pseudo-free families.

In Examples 3.22–3.24 below, let $\mathtt{H} = ((H_d, \mathrm{id}, \mathcal{H}_d) \,|\, d \in D)$, where $H_d \subseteq \{0,1\}^{\le \eta(|d|)}$ is an Ω-algebra in the specified variety $\mathfrak{V}$ and $\mathcal{H}_d$ is a probability distribution on H_d for every $d \in D$. We assume that $\mathtt{H}$ is a 1-pseudo-free family of computational Ω-algebras in $\mathfrak{V}$ with respect to $\mathcal{D}$ and nat. Recall that 1-pseudo-freeness is implied by pseudo-freeness (see Remark 3.9). We say that the family $\mathtt{H}$ has *pseudo-uniform distributions* with respect to $\mathcal{D}$ if $(\mathbf{d}, \mathbf{h}) \approx (\mathbf{d}, \mathbf{u})$, where $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{h} \leftarrow \mathcal{H}_{\mathbf{d}}$, and $\mathbf{u} \leftarrow \mathcal{U}(H_{\mathbf{d}})$ for all $k \in K$.

Example 3.22 (one-way family of permutations from $\mathtt{H}$)**.** Assume that the following conditions hold:

- $\Omega = \{\omega\}$, where $\operatorname{ar} \omega = 1$, and $\mathfrak{V} = \mathfrak{O}$;
- ω is a permutation of H_d for each $d \in D$;
- $\mathtt{H}$ has pseudo-uniform distributions with respect to $\mathcal{D}$;
- there exists a deterministic polynomial-time algorithm that, given $d \in D$ and $u \in \{0,1\}^{\le \eta(|d|)}$, decides whether $u \in H_d$.

Let $\Phi = (\omega\colon H_d \to H_d \,|\, d \in D)$. Suppose A is a probabilistic polynomial-time algorithm. For every $k \in K$, let $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{h} \leftarrow \mathcal{H}_{\mathbf{d}}$, and $\mathbf{u} \leftarrow \mathcal{U}(H_{\mathbf{d}})$. Then $(\mathbf{d}, \mathbf{h}) \approx (\mathbf{d}, \mathbf{u})$ and $(\mathbf{d}, \omega(\mathbf{h})) \approx (\mathbf{d}, \omega(\mathbf{u}))$, where $(\mathbf{d}, \mathbf{u})$ and $(\mathbf{d}, \omega(\mathbf{u}))$ are identically distributed. Hence, $(\mathbf{d}, \omega(\mathbf{h})) \approx (\mathbf{d}, \mathbf{h})$ and

$$\Pr[A(1^k, \mathbf{d}, \omega(\mathbf{h})) = \mathbf{h}] \le \Pr[A(1^k, \mathbf{d}, \mathbf{h}) = \omega^{-1}(\mathbf{h})] + \operatorname{negl}(k).$$

But $\Pr[A(1^k, \mathbf{d}, \mathbf{h}) = \omega^{-1}(\mathbf{h})] = \operatorname{negl}(k)$ because the equation $\omega(x_1) = a_1$ (in the variable x_1) is unsatisfiable in F_1 and $\omega^{-1}(\mathbf{h})$ is the (unique) solution to the equation $\omega(x_1) = \mathbf{h}$ in $H_{\mathbf{d}}$. This shows that the family Φ of permutations is one-way with respect to $\mathcal{D}$ and $(\mathcal{H}_d \,|\, d \in D)$.

Example 3.23 (family of trapdoor permutations from $\mathtt{H}$)**.** Assume that the following conditions hold:

- $\Omega = \{\omega, \epsilon, \delta\}$, where $\operatorname{ar} \omega = 1$, $\operatorname{ar} \epsilon = \operatorname{ar} \delta = 2$, and $\epsilon \ne \delta$;
- the variety $\mathfrak{V}$ is defined by the identity $\forall z_1, z_2\ (\delta(z_1, \epsilon(\omega(z_1), z_2)) = z_2)$;
- ω is a permutation of H_d for each $d \in D$;
- $\mathtt{H}$ has pseudo-uniform distributions with respect to $\mathcal{D}$;
- there exists a deterministic polynomial-time algorithm that, given $d \in D$ and $u \in \{0,1\}^{\le \eta(|d|)}$, decides whether $u \in H_d$.

Let $k \in K$, $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{h}, \mathbf{y} \leftarrow \mathcal{H}_{\mathbf{d}}$, and $\mathbf{u} \leftarrow \mathcal{U}(H_{\mathbf{d}})$. For every $d \in D$ and $h, y \in H_d$, we put $\psi_{d,h}(y) = \epsilon(h, y) \in H_d$. Since $H_d \in \mathfrak{V}$, $\psi_{d,h}$ is a permutation of H_d and $y \mapsto \delta(\omega^{-1}(h), y)$ $(y \in H_d)$ is its inverse.

Suppose A is a probabilistic polynomial-time algorithm. Since $(\mathbf{d}, \mathbf{h}) \approx (\mathbf{d}, \mathbf{u})$, we have $(\mathbf{d}, \mathbf{h}, \mathbf{y}) \approx (\mathbf{d}, \mathbf{u}, \mathbf{y})$. Hence, $(\mathbf{d}, \omega(\mathbf{h}), \mathbf{y}) \approx (\mathbf{d}, \omega(\mathbf{u}), \mathbf{y}) \approx (\mathbf{d}, \mathbf{h}, \mathbf{y})$ because $(\mathbf{d}, \mathbf{u}, \mathbf{y})$ and $(\mathbf{d}, \omega(\mathbf{u}), \mathbf{y})$ are identically distributed. Therefore, $(\mathbf{d}, \omega(\mathbf{h}), \psi_{\mathbf{d},\omega(\mathbf{h})}(\mathbf{y})) \approx (\mathbf{d}, \mathbf{h}, \psi_{\mathbf{d},\mathbf{h}}(\mathbf{y}))$. Furthermore, $(\mathbf{d}, \mathbf{h}, \mathbf{y}) \approx (\mathbf{d}, \mathbf{h}, \mathbf{u})$ because $(\mathbf{d}, \mathbf{h}) \approx (\mathbf{d}, \mathbf{u})$. This implies that $(\mathbf{d}, \mathbf{h}, \psi_{\mathbf{d},\mathbf{h}}(\mathbf{y})) \approx (\mathbf{d}, \mathbf{h}, \psi_{\mathbf{d},\mathbf{h}}(\mathbf{u}))$. But $(\mathbf{d}, \mathbf{h}, \mathbf{u})$ and $(\mathbf{d}, \mathbf{h}, \psi_{\mathbf{d},\mathbf{h}}(\mathbf{u}))$ are identically distributed. Hence we obtain that

$$(\mathbf{d}, \omega(\mathbf{h}), \psi_{\mathbf{d},\omega(\mathbf{h})}(\mathbf{y})) \approx (\mathbf{d}, \mathbf{h}, \psi_{\mathbf{d},\mathbf{h}}(\mathbf{y})) \approx (\mathbf{d}, \mathbf{h}, \mathbf{y})$$

and

$$\Pr[A(1^k, \mathbf{d}, \omega(\mathbf{h}), \psi_{\mathbf{d},\omega(\mathbf{h})}(\mathbf{y})) = \mathbf{y}] \le \Pr[A(1^k, \mathbf{d}, \mathbf{h}, \mathbf{y}) = \psi_{\mathbf{d},\mathbf{h}}^{-1}(\mathbf{y})] + \operatorname{negl}(k). \tag{3}$$

Let G be the Ω-algebra with carrier $\mathbb{N}$ and fundamental operations defined as follows:

$$\omega(m) = m, \quad \epsilon(m, n) = m + n, \quad \delta(m, n) = \max\{n - m, 0\}, \quad m, n \in \mathbb{N}.$$

Then it is evident that $G \in \mathfrak{V}$ and the equation $\epsilon(1, x_1) = 0$ is unsatisfiable in G. This implies that the equation $\epsilon(a_1, x_1) = a_2$ (in the variable x_1) is unsatisfiable in $F_2(\mathfrak{V})$. Also, $\psi^{-1}_{\mathbf{d},\mathbf{h}}(\mathbf{y})$ is the (unique) solution to the equation $\epsilon(\mathbf{h}, x_1) = \mathbf{y}$ in $H_{\mathbf{d}}$. Therefore, $\Pr[A(1^k, \mathbf{d}, \mathbf{h}, \mathbf{y}) = \psi^{-1}_{\mathbf{d},\mathbf{h}}(\mathbf{y})] = \mathrm{negl}(k)$ and hence $\Pr[A(1^k, \mathbf{d}, \omega(\mathbf{h}), \psi_{\mathbf{d},\omega(\mathbf{h})}(\mathbf{y})) = \mathbf{y}] = \mathrm{negl}(k)$ (see (3)). Moreover, $\psi^{-1}_{d,\omega(h)}(y) = \delta(h, y)$ for all $d \in D$ and $h, y \in H_d$. Thus, $(\psi_{d,h} \,|\, d \in D,\ h \in H_d)$ is a family of trapdoor permutations with respect to $(\mathcal{E}_k \,|\, k \in K)$ and $(\mathcal{H}_d \,|\, d \in D,\ h \in H_d)$, where $\mathcal{E}_k$ is the distribution of the random variable $((\mathbf{d}, \omega(\mathbf{h})), \mathbf{h})$.

Example 3.24 (claw-resistant family of pairs of permutations from $\mathtt{H}$)**.** Assume that the following conditions hold:

- $\Omega = \{\omega_1, \omega_2\}$, where $\operatorname{ar} \omega_1 = \operatorname{ar} \omega_2 = 1$, $\omega_1 \neq \omega_2$, and $\mathfrak{V} = \mathfrak{O}$;
- ω_1 and ω_2 are permutations of H_d for each $d \in D$.

Then the family $((\omega_1 \colon H_d \to H_d, \omega_2 \colon H_d \to H_d) \,|\, d \in D)$ of pairs of permutations is claw-resistant with respect to $\mathcal{D}$. This is because the equation $\omega_1(x_1) = \omega_2(x_2)$ is unsatisfiable in F_∞ and a pair $(h_1, h_2) \in H_d^2$ is a claw for (ω_1, ω_2) if and only if it is a solution to this equation.

4 When Polynomially Bounded (Weakly) Pseudo-Free Families in $\mathfrak{O}$ Exist Unconditionally?

In this section, we mostly consider the case where $\mathfrak{V} = \mathfrak{O}$. Recall that $\overline{w} = \mathrm{nat}^{-1}(w)$ for any $w \in F_{\infty,\infty}$ (see Example 3.12).

4.1 Unconditional Results

***Remark* 4.1.** Assume that Ω consists of nullary operation symbols only. By Corollary 3.18,

$$\mathtt{F} = ((F_{2^{|u|}}, \mathrm{nat}_{2^{|u|}}, \mathcal{U}(\{\overline{a_1}, \dots, \overline{a_{2^{|u|}}}\})) \,|\, u \in 1^K)$$

is a pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \,|\, k \in K)$ and σ. Also, $\mathtt{F}$ has unique representations. Furthermore, it is easy to see that $F_{2^k} = \Omega \sqcup \{a_1, \dots, a_{2^k}\}$ for all $k \in K$. Therefore each string (over the alphabet $\Omega \sqcup \{a, 0, 1\}$) in $\operatorname{dom} \mathrm{nat}_{2^k}$ has length at most $k + 2$. This shows that $\mathtt{F}$ is polynomially bounded.

***Remark* 4.2.** Assume that $\Omega = \Omega_0 \sqcup \{\omega\}$, where Ω_0 consists of nullary operation symbols and $\operatorname{ar} \omega = 1$. For arbitrary $n \in \mathbb{N}$, denote by ω^n the n-fold composition of ω with itself. It is easy to see that every element of F_∞ can be uniquely represented as $\omega^i(b)$, where $i \in \mathbb{N}$ and $b \in \Omega_0 \sqcup \{a_1, a_2, \dots\}$.

Let $k \in K$. Denote by θ_{1^k} the following binary relation on F_{2^k}:

$$\theta_{1^k} = \{(v, w) \in F_{2^k}^2 \,|\, v = w \text{ or } v = \omega^i(b),\ w = \omega^j(b),\ \text{where } i, j \geq 2^k,\ b \in \Omega_0 \sqcup \{a_1, \dots, a_{2^k}\}\}.$$

This relation is a congruence on F_{2^k}. The equivalence classes under θ_{1^k} are

$$\{\omega^0(b)\}, \dots, \{\omega^{2^k - 1}(b)\}, \{\omega^{2^k}(b), \omega^{2^k + 1}(b), \dots\},$$

where b ranges over $\Omega_0 \sqcup \{a_1, \dots, a_{2^k}\}$.

By Corollary 3.18,

$$\mathtt{F} = ((F_{2^{|u|}}, \mathrm{nat}_{2^{|u|}}, \mathcal{U}(\{\overline{a_1}, \dots, \overline{a_{2^{|u|}}}\})) \,|\, u \in 1^K)$$

is a pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \,|\, k \in K)$ and nat. We observe that, given $(1^k, \overline{v}, \overline{w})$ (where $v, w \in F_{2^k}$), one can decide whether $(v, w) \in \theta_{1^k}$ in deterministic polynomial time. Also, if $(v, w) \in \theta^{\neq}_{1^k}$, then both $\overline{v}$ and $\overline{w}$ have length at least $2^k + 1$ as strings over $\Omega \sqcup \{a, 0, 1\}$. This implies that for any probabilistic polynomial-time algorithm A, we have $\Pr[A(1^k, 1^k) = (\overline{v}, \overline{w}) \text{ s.t. } (v, w) \in \theta^{\neq}_{1^k}] = 0$ for all sufficiently large $k \in K$. Moreover, it is easy to see that the family $\mathtt{F}$ is nat-compatible. Thus, by Lemma 3.20,

$$\mathtt{F}' = ((F_{2^{|u|}}/\theta_u, \mathrm{nat}_{2^{|u|}}/\theta_u, \mathcal{U}(\{\overline{a_1}, \dots, \overline{a_{2^{|u|}}}\})) \,|\, u \in 1^K)$$

is a pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and nat. (We apply this lemma to $\mathtt{H} = \mathtt{F}$, $E_u = \{e\}$, where $e \in \{0,1\}^*$ is arbitrary, $\mathcal{E}_u = \mathcal{U}(E_u)$, and $\theta_{u,e} = \theta_u$ for all $u \in 1^K$. Since e is fixed, we omit it.) The family $\mathtt{F}'$ has exponential size because $|F_{2^k}/\theta_{1^k}| = (2^k+1)(|\Omega_0| + 2^k)$ for all $k \in K$. But this family is not polynomially bounded and does not have unique representations. The last disadvantage can be overcome by restricting the function $\mathrm{nat}_{2^{|u|}}$ to the set

$$S_u = \{\overline{\omega^i(b)} \mid i \in \{0,\dots,2^{|u|}\},\ b \in \Omega_0 \sqcup \{a_1,\dots,a_{2^{|u|}}\}\},$$

where $u \in 1^K$. Namely, let

$$\mathtt{F}'' = ((F_{2^{|u|}}/\theta_u, (\mathrm{nat}_{2^{|u|}}|_{S_u})/\theta_u, \mathcal{U}(\{\overline{a_1},\dots,\overline{a_{2^{|u|}}}\})) \mid u \in 1^K).$$

Then by Remark 3.11, $\mathtt{F}''$ is a pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and nat (note that $(\mathrm{nat}_{2^{|u|}}|_{S_u})/\theta_u = (\mathrm{nat}_{2^{|u|}}/\theta_u)|_{S_u}$ for all $u \in 1^K$). This family has exponential size and unique representations, but is not polynomially bounded.

***Remark* 4.3.** In this remark, as in Remark 4.2, we assume that $\Omega = \Omega_0 \sqcup \{\omega\}$, where Ω_0 consists of nullary operation symbols and $\operatorname{ar}\omega = 1$. Also, we use the notation of Remark 4.2.

Let $k \in K$. Define the function δ_{1^k} by $\delta_{1^k}(i, \overline{b}) = \omega^i(b)$ for each $i \in \mathbb{N}$ and $b \in \Omega_0 \sqcup \{a_1,\dots,a_{2^k}\}$. This function provides a more succinct representation of elements of F_{2^k} than nat_{2^k}. By Lemma 3.17,

$$\mathtt{F} = ((F_{2^{|u|}}, \delta_u, \mathcal{U}(\{(0,\overline{a_1}),\dots,(0,\overline{a_{2^{|u|}}})\})) \mid u \in 1^K)$$

is a pseudo-free (and hence weakly pseudo-free) family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and SLP. Of course, given 1^k and $[v]_{\delta_{1^k}}, [w]_{\delta_{1^k}}$ (where $v, w \in F_{2^k}$), one can decide whether $(v,w) \in \theta_{1^k}$ in deterministic polynomial time. Suppose $v, w \in F_m$ and $f \in \{a_1,\dots,a_{2^k}\}^m$ (where $m \in \mathbb{N}$) are such that $(v(f), w(f)) \in \theta^{\neq}_{1^k}$. Let $v = \omega^i(b)$ and $w = \omega^j(c)$, where $i, j \in \mathbb{N}$ and $b, c \in \Omega_0 \sqcup \{a_1,\dots,a_m\}$. Then $v(f) = \omega^i(b(f))$ and $w(f) = \omega^j(c(f))$, where $b(f), c(f) \in \Omega_0 \sqcup \{a_1,\dots,a_{2^k}\}$. Therefore we have $i, j \geq 2^k$. It is evident that $\mathrm{subt}(v) = \{\omega^l(b) \mid l \in \{0,\dots,i\}\}$ and $\mathrm{subt}(w) = \{\omega^l(c) \mid l \in \{0,\dots,j\}\}$. Hence it follows from Remark 3.15 that if $(u_1,\dots,u_n) \in \mathrm{SLP}^{-1}(v) \sqcup \mathrm{SLP}^{-1}(w)$, then $n \geq \min\{i,j\} + 1 \geq 2^k + 1$. This implies that for any polynomial π and any probabilistic polynomial-time algorithm A,

$$\Pr[A(1^k, 1^k, \mathbf{r}) = ([v]_{\mathrm{SLP}}, [w]_{\mathrm{SLP}}) \text{ s.t. } v, w \in F_{\pi(k)} \text{ and } (v(\delta_{1^k}(\mathbf{r})), w(\delta_{1^k}(\mathbf{r}))) \in \theta^{\neq}_{1^k}] = 0$$

for all sufficiently large $k \in K$, where $\mathbf{r} \leftarrow \mathcal{U}(\{(0,\overline{a_1}),\dots,(0,\overline{a_{2^k}})\})^{\pi(k)}$. Thus, by Lemma 3.21,

$$\mathtt{F}' = ((F_{2^{|u|}}/\theta_u, \delta_u/\theta_u, \mathcal{U}(\{(0,\overline{a_1}),\dots,(0,\overline{a_{2^{|u|}}})\})) \mid u \in 1^K)$$

is a weakly pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and SLP. (As in Remark 4.2, we apply this lemma to $\mathtt{H} = \mathtt{F}$, $E_u = \{e\}$, where $e \in \{0,1\}^*$ is arbitrary, $\mathcal{E}_u = \mathcal{U}(E_u)$, and $\theta_{u,e} = \theta_u$ for all $u \in 1^K$. Since e is fixed, we omit it.) The family $\mathtt{F}'$ has exponential size, but is not polynomially bounded and does not have unique representations. However, we can overcome both of these disadvantages by restricting the function δ_u to the set $S_u = \{(i, \overline{b}) \mid i \in \{0,\dots,2^{|u|}\},\ b \in \Omega_0 \sqcup \{a_1,\dots,a_{2^{|u|}}\}\}$, where $u \in 1^K$. Namely, let

$$\mathtt{F}'' = ((F_{2^{|u|}}/\theta_u, (\delta_u|_{S_u})/\theta_u, \mathcal{U}(\{(0,\overline{a_1}),\dots,(0,\overline{a_{2^{|u|}}})\})) \mid u \in 1^K).$$

Then by Remark 3.11, $\mathtt{F}''$ is a weakly pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and SLP (note that $(\delta_u|_{S_u})/\theta_u = (\delta_u/\theta_u)|_{S_u}$ for all $u \in 1^K$). It is easy to see that the family $\mathtt{F}''$ is polynomially bounded and has unique representations.

Note that neither $\mathtt{F}'$ nor $\mathtt{F}''$ is 1-pseudo-free in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and nat. This is because the equation $x_1 = \omega(x_1)$ is unsatisfiable in F_∞, but $\omega^{2^{|u|}}(a_1)/\theta_u = \delta_u(2^{|u|}, \overline{a_1})/\theta_u$ is a solution to this equation in $F_{2^{|u|}}/\theta_u$ $(u \in 1^K)$. In particular, neither $\mathtt{F}'$ nor $\mathtt{F}''$ is pseudo-free in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and SLP (see Remarks 3.16 and 3.9).

4.2 Some Cases Where the Existence of Weakly Pseudo-Free Families Implies the Existence of Collision-Resistant Families of Hash Functions

Construction 4.4. Suppose $\chi\colon \bigsqcup_{n \in N} \{0,1\}^n \to F_\infty(\mathfrak{V})$ is a function satisfying the following conditions:

(i) N is an infinite polynomial-time enumerable subset of $\mathbb{N}$. This means that the function $i \mapsto \min\{n \in N \mid n > i\}$ is a polynomial parameter on $\mathbb{N}$ (see [Gol01, Subsubsection 2.2.3.1]).

(ii) There exists a deterministic polynomial-time algorithm that, given $y \in \bigsqcup_{n \in N} \{0,1\}^n$, computes $[\chi(y)]_{\mathrm{nat}}$.

(iii) There exists a polynomial γ such that $\chi(\{0,1\}^n) \subseteq F_{\gamma(n)}(\mathfrak{V})$ for all $n \in N$.

(iv) For any $n \in N$, $\chi|_{\{0,1\}^n}$ is one-to-one.

Also, let $\mathtt{H} = ((H_d, \rho_d, \mathcal{R}_d) \mid d \in D)$ be a polynomially bounded family of computational Ω-algebras in $\mathfrak{V}$ (see Subsection 3.2). Choose a polynomial parameter η on K such that $\operatorname{dom} \rho_d \subseteq \{0,1\}^{\le \eta(k)}$ for each $k \in K$ and $d \in \operatorname{supp} \mathcal{D}_k$. Denote by ξ the polynomial parameter $k \mapsto \min\{n \in N \mid n > \eta(k) + 1\}$ on K (see Condition (i)). Then $\xi(k) \in N$ and $\xi(k) > \eta(k) + 1$ for all $k \in K$.

For any $n \in \mathbb{N}$, let α_n be the one-to-one function from $\{0,1\}^{\le n}$ onto $\{0,1\}^{n+1} \setminus \{0^{n+1}\}$ defined by $\alpha_n(y) = y10^{n-|y|}$ for all $y \in \{0,1\}^{\le n}$. Then the function $(1^n, y) \mapsto \alpha_n(y)$, where $n \in \mathbb{N}$ and $y \in \{0,1\}^{\le n}$, is polynomial-time computable.

Choose a polynomial π such that $\chi(\{0,1\}^{\xi(k)}) \subseteq F_{\pi(k)}(\mathfrak{V})$ for all $k \in K$. Condition (iii) implies that such a polynomial exists. Put

$$E_k = \{(1^k, d, r) \mid d \in \operatorname{supp} \mathcal{D}_k,\ r \in (\operatorname{dom} \rho_d)^{\pi(k)}\} \text{ (where } k \in K) \quad \text{and} \quad E = \bigsqcup_{k \in K} E_k.$$

For each $e \in E$, define $\kappa(e)$ to be the unique $k \in K$ such that $e \in E_k$. It is easy to see that $E_k \subseteq \{0,1\}^{\le \zeta(k)}$ for all $k \in K$, where ζ is a fixed polynomial, and κ is a polynomial parameter on E, as in Definition 2.6. Finally, let

$$\phi_{(1^k,d,r)}(y) = \alpha_{\eta(k)}([\chi(y)(\rho_d(r))]_{\rho_d}),$$

for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, $r \in (\operatorname{dom} \rho_d)^{\pi(k)}$, and $y \in \{0,1\}^{\xi(k)}$. Here $[\chi(y)(\rho_d(r))]_{\rho_d}$ denotes the preimage of $\chi(y)(\rho_d(r))$ under ρ_d computed by the following deterministic polynomial-time algorithm:

1. Given y, compute $[\chi(y)]_{\mathrm{nat}}$ (see Condition (ii)).
2. Given d, $[\chi(y)]_{\mathrm{nat}}$, and r, compute and output $[\chi(y)(\rho_d(r))]_{\rho_d}$. (This can be done in deterministic polynomial time because $\mathtt{H}$ is nat-compatible.)

Thus, $\Phi = (\phi_e \colon \{0,1\}^{\xi(\kappa(e))} \to \{0,1\}^{\eta(\kappa(e))+1} \mid e \in E)$ is a family of hash functions.

Theorem 4.5. *Let* $\mathtt{H}$, π, *and* Φ *be as in Construction 4.4. Assume that the family* $\mathtt{H}$ *is weakly* 1*-pseudo-free in* $\mathfrak{V}$ *with respect to* $\mathcal{D}$ *and* nat. *For each* $k \in K$, *denote by* $\mathcal{E}_k$ *the distribution of the random variable* $(1^k, \mathbf{d}, \mathbf{r})$, *where* $\mathbf{d} \leftarrow \mathcal{D}_k$ *and* $\mathbf{r} \leftarrow \mathcal{R}_{\mathbf{d}}^{\pi(k)}$. *Then the family* Φ *is collision-resistant with respect to* $\mathcal{E} = (\mathcal{E}_k \mid k \in K)$. *(It is evident that the probability ensemble* $\mathcal{E}$ *is polynomial-time samplable when the indices are represented in unary.)*

Proof. Let A be a probabilistic polynomial-time algorithm. Suppose B is a probabilistic polynomial-time algorithm that on input $e = (1^k, d, r)$ for every $k \in K$, $d \in \operatorname{supp} \mathcal{D}_k$, and $r \in (\operatorname{supp} \mathcal{R}_d)^{\pi(k)}$ proceeds as follows:

1. Run A on input e. Assume that the output is a collision (y, z) for the function ϕ_e. If this is not true, then B fails.
2. Compute and output $([\chi(y)]_{\mathrm{nat}}, [\chi(z)]_{\mathrm{nat}})$, where χ is related to π and Φ as in Construction 4.4. (It is easy to see that this pair is in $\Sigma'_1(H_d, \mathfrak{V}, \mathrm{nat}, \rho_d(r))$.)

Let $k \in K$, $\mathbf{d} \leftarrow \mathcal{D}_k$, and $\mathbf{r} \leftarrow \mathcal{R}_{\mathbf{d}}^{\pi(k)}$. Then the random variable $\mathbf{e} = (1^k, \mathbf{d}, \mathbf{r})$ is distributed according to $\mathcal{E}_k$. Furthermore, we have

$$\Pr[A(\mathbf{e}) \text{ is a collision for } \phi_{\mathbf{e}}] = \Pr[B(1^k, \mathbf{d}, \mathbf{r}) \in \Sigma'_1(H_{\mathbf{d}}, \mathfrak{V}, \mathrm{nat}, \rho_{\mathbf{d}}(\mathbf{r}))] = \operatorname{negl}(k)$$

because $\mathtt{H}$ is weakly 1-pseudo-free in $\mathfrak{V}$ with respect to $\mathcal{D}$ and nat. Thus, the family Φ is collision-resistant with respect to $\mathcal{E}$. □

Corollary 4.6. *Assume that there exists a function $\chi\colon \bigsqcup_{n\in N}\{0,1\}^n \to F_\infty(\mathfrak{V})$ satisfying Conditions (i)–(iv) of Construction 4.4. Then the existence of polynomially bounded weakly 1-pseudo-free families of computational Ω-algebras in $\mathfrak{V}$ with respect to $\mathcal{D}$ and* nat *implies the existence of collision-resistant families of hash functions (with respect to some probability ensemble that is indexed by K and is polynomial-time samplable when the indices are represented in unary).*

Corollary 4.6 follows immediately from Theorem 4.5.

***Remark* 4.7.** Here are some cases where a function $\chi\colon \bigsqcup_{n\in N}\{0,1\}^n \to F_\infty(\mathfrak{V})$ satisfying Conditions (i)–(iv) of Construction 4.4 exists:

(i) *$\Omega \ni \omega$, where $\operatorname{ar}\omega = 2$, and $\mathfrak{V}$ is a nontrivial variety of Ω-algebras such that any $H \in \mathfrak{V}$ is a groupoid with an identity element (denoted by 1_H) under ω.* (In particular, this holds if $\mathfrak{V}$ is a nontrivial variety of monoids, loops, groups, or rings.) In this case, the required function $\chi\colon \{0,1\}^* \to F_\infty(\mathfrak{V})$ can be defined as follows. For any $y \in \{0,1\}^*$, let $\{i_1,\dots,i_m\}$ (where $i_1 < \dots < i_m$) be the set of all $i \in \{1,\dots,|y|\}$ such that the ith bit of y is 1. Then

$$\chi(y) = \begin{cases} 1_{F_\infty(\mathfrak{V})} & \text{if } m = 0,\\ a_{i_1} & \text{if } m = 1,\\ \omega(\dots\omega(\omega(a_{i_1},a_{i_2}),a_{i_3}),\dots,a_{i_m}) & \text{if } m \geq 2.\end{cases}$$

Choose an Ω-algebra $H \in \mathfrak{V}$ with at least two elements. Furthermore, let $h \in H \setminus \{1_H\}$. Suppose y and z are distinct bit strings of the same length. We assume that the jth bits of y and z are 0 and 1, respectively. Let α be the homomorphism of $F_\infty(\mathfrak{V})$ to H such that $\alpha(a_j) = h$ and $\alpha(a_i) = 1_H$ for all $i \in \mathbb{N}\setminus\{j\}$. Then it is easy to see that $\alpha(\chi(y)) = 1_H \neq h = \alpha(\chi(z))$ and hence $\chi(y) \neq \chi(z)$. (We note that $\alpha(1_{F_\infty(\mathfrak{V})}) = \omega(\alpha(1_{F_\infty(\mathfrak{V})}),\alpha(a_{j+1})) = \alpha(a_{j+1}) = 1_H$.) Thus, $\chi|_{\{0,1\}^n}$ is one-to-one for every $n \in \mathbb{N}$.

(ii) *$\Omega \ni \omega_0, \omega_1$, where $\operatorname{ar}\omega_0 = \operatorname{ar}\omega_1 = 1$ and $\omega_0 \neq \omega_1$, and $\mathfrak{V} = \mathfrak{O}$.* In this case, the required function $\chi\colon \{0,1\}^* \to F_\infty$ can be defined by $\chi(y) = \omega_{y_n}(\dots\omega_{y_2}(\omega_{y_1}(a_1))\dots)$ for all $y = y_1\dots y_n \in \{0,1\}^*$, where $n \in \mathbb{N}$ and $y_1,\dots,y_n \in \{0,1\}$.

(iii) *$\Omega \ni \omega$, where $\operatorname{ar}\omega = m \geq 2$, and $\mathfrak{V} = \mathfrak{O}$.* In this case, the required function $\chi\colon \{0,1\}^* \to F_\infty$ can be defined inductively as follows:

$$\chi(\epsilon) = a_1, \quad \chi(y0) = \chi(y), \quad \chi(y1) = \omega(\underbrace{a_{|y|+1},\dots,a_{|y|+1}}_{m-1 \text{ times}},\chi(y)),$$

where ϵ is the empty string and $y \in \{0,1\}^*$. Using induction on $|z|$, it is easy to see that for any $z \in \{0,1\}^*$ and any $i \in \{1,\dots,|z|\}$, the ith bit of z is 1 if and only if $\chi(z)$ contains a subterm of the form $\omega(a_i,\dots,a_i,v)$, where $v \in F_\infty$. This implies that for each $n \in \mathbb{N}$, $\chi|_{\{0,1\}^n}$ is one-to-one.

By Corollary 4.6, in any of these cases, the existence of polynomially bounded weakly 1-pseudo-free families of computational Ω-algebras in $\mathfrak{V}$ with respect to $\mathcal{D}$ and nat implies the existence of collision-resistant families of hash functions (with respect to some probability ensemble that is indexed by K and is polynomial-time samplable when the indices are represented in unary).

4.3 Summary of Results

The main results of this section can be summarized as follows:

- Assume that Ω consists of nullary operation symbols only. Then there exists a polynomially bounded pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\})\,|\,k \in K)$ and σ. Moreover, this family has unique representations. See Remark 4.1.

- Assume that $\Omega = \Omega_0 \sqcup \{\omega\}$, where Ω_0 consists of nullary operation symbols and $\operatorname{ar}\omega = 1$. Then there exist
  - an exponential-size pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\})\,|\,k \in K)$ and nat and

  - a polynomially bounded weakly pseudo-free family of computational Ω-algebras in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and SLP.

  Moreover, both of these families have unique representations. See Remarks 4.2 and 4.3.

- In all other cases, the existence of polynomially bounded weakly pseudo-free families of computational Ω-algebras in $\mathfrak{O}$ with respect to $\mathcal{D}$ and nat implies the existence of collision-resistant families of hash functions (with respect to some probability ensemble that is indexed by K and is polynomial-time samplable when the indices are represented in unary). See Corollary 4.6 and Remark 4.7 (Cases (ii) and (iii)). Note that by Remark 3.9, weak pseudo-freeness in $\mathfrak{O}$ with respect to $\mathcal{D}$ and nat is equivalent to weak 1-pseudo-freeness in $\mathfrak{O}$ with respect to $\mathcal{D}$ and nat.

5 (Weakly) Pseudo-Free Families in the Variety of All m-Ary Groupoids

In this section, we assume that $\Omega = \{\omega\}$, where $\operatorname{ar}\omega = m \geq 1$. In other words, we consider m-ary groupoids.

Lemma 5.1. *Let G be an m-ary groupoid and let $g = (g_1, \dots, g_n) \in G^n$, where $n \in \mathbb{N}$. Assume that $g_1, \dots, g_n$ are distinct and that $g_i \notin \omega(G^m)$ for all $i \in \{1, \dots, n\}$. Also, suppose v and w are distinct elements of F_n such that $v(g) = w(g)$. Then there exist $v_1, \dots, v_m, w_1, \dots, w_m \in F_n$ such that the following two conditions hold:*

(i) *$\omega(v_1, \dots, v_m) \in \operatorname{subt}(v)$ and $\omega(w_1, \dots, w_m) \in \operatorname{subt}(w)$;*

(ii) *$(v_1(g), \dots, v_m(g)) \neq (w_1(g), \dots, w_m(g))$, but $\omega(v_1(g), \dots, v_m(g)) = \omega(w_1(g), \dots, w_m(g))$.*

Proof. Denote by V the set of all $v \in F_n$ satisfying the following condition:

$$\forall\, w \in F_n\ (v \neq w,\ v(g) = w(g) \implies \exists\, v_1, \dots, v_m, w_1, \dots, w_m \in F_n \text{ s.t. Conditions (i) and (ii) hold}).$$

To prove the lemma, it suffices to show that $\{a_1, \dots, a_n\} \subseteq V$ and that V is an m-ary subgroupoid (i.e., subalgebra) of F_n.

If $v \in \{a_1, \dots, a_n\}$, then the assumptions on g imply that for any $w \in F_n$, we have $v = w$ or $v(g) \neq w(g)$. This shows that $\{a_1, \dots, a_n\} \subseteq V$.

Let $v'_1, \dots, v'_m \in V$ and $v = \omega(v'_1, \dots, v'_m)$. Also, suppose w is an element of F_n such that $v \neq w$ and $v(g) = w(g)$. Then it follows from the assumptions on g that $w = \omega(w'_1, \dots, w'_m)$, where $w'_1, \dots, w'_m \in F_n$. If $(v'_1(g), \dots, v'_m(g)) \neq (w'_1(g), \dots, w'_m(g))$, then Conditions (i) and (ii) hold for $v_i = v'_i$ and $w_i = w'_i$ ($i \in \{1, \dots, m\}$). Otherwise, choose an index $j \in \{1, \dots, m\}$ satisfying $v'_j \neq w'_j$; such an index exists because $v \neq w$. In this case, Conditions (i) and (ii) hold for some $v_1, \dots, v_m, w_1, \dots, w_m \in F_n$ such that $\omega(v_1, \dots, v_m) \in \operatorname{subt}(v'_j)$ and $\omega(w_1, \dots, w_m) \in \operatorname{subt}(w'_j)$. This is because $v'_j \in V$, $v'_j \neq w'_j$, and $v'_j(g) = w'_j(g)$. Thus, we obtain that $v \in V$. This shows that V is an m-ary subgroupoid of F_n. □

In Subsections 5.1–5.2 below, we use the assumptions and notation of Definition 2.6. In these subsections, we also assume that $\operatorname{supp} \mathcal{D}_k \subseteq D_k$ for every $k \in K$.

5.1 Constructing a Polynomially Bounded Weakly Pseudo-Free Family from a Collision-Resistant Family of Hash Functions

Construction 5.2. Suppose $\Psi = (\psi_d \colon \{0,1\}^{m\xi(\kappa(d))} \to \{0,1\}^{\eta(\kappa(d))} \mid d \in D)$ is a family of hash functions, where ξ and η are polynomial parameters on K satisfying $\xi(k) > \eta(k)$ for all $k \in K$. Then for every $d \in D$, let G_d be the m-ary groupoid with carrier $\{0,1\}^{\xi(\kappa(d))}$ and fundamental operation defined by

$$\omega(g_1, \dots, g_m) = \psi_d(g_1 \dots g_m) 1^{\xi(\kappa(d)) - \eta(\kappa(d))}, \quad g_1, \dots, g_m \in \{0,1\}^{\xi(\kappa(d))}.$$

Finally, put $M_k = \{0,1\}^{\eta(k)} 0^{\xi(k) - \eta(k)}$ for each $k \in K$.

Theorem 5.3. *Let Ψ, G_d $(d \in D)$, and M_k $(k \in K)$ be as in Construction 5.2. Assume that the family Ψ is collision-resistant with respect to $\mathcal{D}$. Then $\mathtt{G} = ((G_d, \mathrm{id}, \mathcal{U}(M_{\kappa(d)})) \mid d \in D)$ is a polynomially bounded weakly pseudo-free family of computational m-ary groupoids in $\mathfrak{O}$ with respect to $\mathcal{D}$ and* SLP.

Proof. It is easy to see that $\mathtt{G}$ is a polynomially bounded family of computational m-ary groupoids. Let π be a polynomial and let A be a probabilistic polynomial-time algorithm. Suppose B is a probabilistic polynomial-time algorithm that on input $d \in D$ proceeds as follows:

1. Choose $\mathrm{g}_1, \dots, \mathrm{g}_{\pi(k)} \leftarrow \mathcal{U}(M_k)$, where $k = \kappa(d)$. If $\mathrm{g}_1, \dots, \mathrm{g}_{\pi(k)}$ are distinct, then put $\mathrm{g} = (\mathrm{g}_1, \dots, \mathrm{g}_{\pi(k)})$. Otherwise, the algorithm B fails.

2. Run A on input $(1^k, d, \mathrm{g})$. Assume that the output is $([v]_{\mathrm{SLP}}, [w]_{\mathrm{SLP}}) \in \Sigma_1'(G_d, \mathfrak{O}, \mathrm{SLP}, \mathrm{g})$. If this is not true, then B fails.

3. Find (by exhaustive search) a pair $(([v_1]_{\mathrm{SLP}}, \dots, [v_m]_{\mathrm{SLP}}), ([w_1]_{\mathrm{SLP}}, \dots, [w_m]_{\mathrm{SLP}}))$ of m-tuples such that the following conditions hold:

   - $\omega(v_1, \dots, v_m) \in \mathrm{subt}(v)$ and $\omega(w_1, \dots, w_m) \in \mathrm{subt}(w)$;
   - $(v_1(\mathrm{g}), \dots, v_m(\mathrm{g})) \neq (w_1(\mathrm{g}), \dots, w_m(\mathrm{g}))$, but $\omega(v_1(\mathrm{g}), \dots, v_m(\mathrm{g})) = \omega(w_1(\mathrm{g}), \dots, w_m(\mathrm{g}))$.

   By Lemma 5.1, such a pair exists. (We note that $\mathrm{g}_i \notin \omega(G_d^m)$ for all $i \in \{1, \dots, \pi(k)\}$. This is because the last bits of g_i and of any string in $\omega(G_d^m)$ are 0 and 1, respectively.) The exhaustive search can be performed in polynomial time by Remark 3.15.

4. Output $(v_1(\mathrm{g}) \dots v_m(\mathrm{g}), w_1(\mathrm{g}) \dots w_m(\mathrm{g}))$. (By the last condition of the previous step, together with the definition of ω on G_d, this is a collision for ψ_d.)

Let $k \in K$, $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{g}_1, \dots, \mathbf{g}_{\pi(k)} \leftarrow \mathcal{U}(M_k)$, and $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_{\pi(k)})$. Then

$$\begin{aligned}
&\Pr[A(1^k, \mathbf{d}, \mathbf{g}) \in \Sigma_1'(G_{\mathbf{d}}, \mathfrak{O}, \mathrm{SLP}, \mathbf{g})] \\
&\quad = \Pr[A(1^k, \mathbf{d}, \mathbf{g}) \in \Sigma_1'(G_{\mathbf{d}}, \mathfrak{O}, \mathrm{SLP}, \mathbf{g}),\ \mathbf{g}_1, \dots, \mathbf{g}_{\pi(k)} \text{ are distinct}] \\
&\qquad + \Pr[A(1^k, \mathbf{d}, \mathbf{g}) \in \Sigma_1'(G_{\mathbf{d}}, \mathfrak{O}, \mathrm{SLP}, \mathbf{g}),\ \mathbf{g}_1, \dots, \mathbf{g}_{\pi(k)} \text{ are not distinct}] \\
&\quad \le \Pr[B(\mathbf{d}) \text{ is a collision for } \psi_{\mathbf{d}}] + \frac{\pi(k)(\pi(k)-1)}{2^{\eta(k)+1}} = \mathrm{negl}(k) + \mathrm{negl}(k) = \mathrm{negl}(k)
\end{aligned}$$

because Ψ is collision-resistant with respect to $\mathcal{D}$ and $2^{-\eta(k)} = \mathrm{negl}(k)$ (see Remark 2.8). Thus, the family $\mathtt{G}$ is weakly pseudo-free in $\mathfrak{O}$ with respect to $\mathcal{D}$ and SLP (see Remarks 3.14 and 3.9). □

Corollary 5.4. *Assume that $m \ge 2$. Then the following conditions are equivalent:*

(i) *There exists a collision-resistant family of hash functions with respect to some probability ensemble that is indexed by K and is polynomial-time samplable when the indices are represented in unary.*

(ii) *There exists a polynomially bounded weakly pseudo-free family of computational m-ary groupoids in $\mathfrak{O}$ with respect to some probability ensemble (with the same properties as in Condition (i)) and* SLP.

(iii) *The same as Condition (ii), but with* nat *instead of* SLP.

Proof. The implication (i) $\Longrightarrow$ (ii) follows from Lemma 2.9 and Theorem 5.3. The implication (ii) $\Longrightarrow$ (iii) follows from Remark 3.16. Finally, the implication (iii) $\Longrightarrow$ (i) follows from Corollary 4.6 and Remark 4.7 (Case (iii)). □

***Remark* 5.5.** Note that the family $\mathtt{G}$ in Theorem 5.3 has unique representations. Therefore Corollary 5.4 remains valid if we require that the weakly pseudo-free families in Conditions (ii) and (iii) additionally have unique representations.

5.2 Constructing an Exponential-Size Pseudo-Free Family from a Collision-Resistant Family of Hash Functions

Construction 5.6. Suppose $\Psi = (\psi_d \colon \{0,1\}^{m\xi(\kappa(d))} \to \{0,1\}^{\eta(\kappa(d))} \,|\, d \in D)$ and G_d $(d \in D)$ are as in Construction 5.2. Let $d \in D$ and $k = \kappa(d)$. For each $n \in \{0, \dots, 2^{\eta(k)} - 1\}$, denote by $\beta_k(n) \in \{0,1\}^{\eta(k)}$ the binary representation of length $\eta(k)$ of n (with enough leading zeros to obtain $\eta(k)$ bits). Thus, β_k is a one-to-one function from $\{0, \dots, 2^{\eta(k)} - 1\}$ onto $\{0,1\}^{\eta(k)}$. Suppose λ_d is the homomorphism of $F_{2^{\eta(k)}}$ to G_d such that $\lambda_d(a_i) = \beta_k(i-1)0^{\xi(k)-\eta(k)}$ for all $i \in \{1, \dots, 2^{\eta(k)}\}$ and θ_d is the kernel of this homomorphism.

Recall that $\overline{w} = \mathrm{nat}^{-1}(w)$ for any $w \in F_{\infty,\infty}$ (see Example 3.12).

Theorem 5.7. *Let Ψ, η, and θ_d $(d \in D)$ be as in Construction 5.6. Assume that the family Ψ is collision-resistant with respect to $\mathcal{D}$. Then*

$$\mathtt{Q} = ((F_{2^{\eta(\kappa(d))}}/\theta_d, \mathrm{nat}_{2^{\eta(\kappa(d))}}/\theta_d, \mathcal{U}(\{\overline{a_1}, \dots, \overline{a_{2^{\eta(\kappa(d))}}}\})) \mid d \in D)$$

is a pseudo-free family of computational m-ary groupoids in $\mathfrak{O}$ with respect to $\mathcal{D}$ and nat*. Moreover, the family $\mathtt{Q}$ has exponential size.*

Proof. Remark 2.8 shows that $2^{-\eta(k)} = \mathrm{negl}(k)$. Therefore, by Corollary 3.18,

$$\mathtt{F} = ((F_{2^{\eta(|u|)}}, \mathrm{nat}_{2^{\eta(|u|)}}, \mathcal{U}(\{\overline{a_1}, \dots, \overline{a_{2^{\eta(|u|)}}}\})) \mid u \in 1^K)$$

is a pseudo-free family of computational m-ary groupoids in $\mathfrak{O}$ with respect to $(\mathcal{U}(\{1^k\}) \mid k \in K)$ and nat. Furthermore, it is easy to see that $\mathtt{F}$ is nat-compatible.

Suppose λ_d $(d \in D)$ is as in Construction 5.6. It is not hard to show that, given $(d, \overline{v})$ (where $d \in D$ and $v \in F_{2^{\eta(\kappa(d))}}$), one can compute $\lambda_d(v)$ in polynomial time. Hence there exists a deterministic polynomial-time algorithm that, given $(1^k, d, \overline{v}, \overline{w})$, where $k \in K$, $d \in D_k$, and $v, w \in F_{2^{\eta(k)}}$, decides whether $(v, w) \in \theta_d$.

Let A be a probabilistic polynomial-time algorithm. Suppose B is a probabilistic polynomial-time algorithm that on input $d \in D$ proceeds as follows:

1. Run A on input $(1^k, 1^k, d)$, where $k = \kappa(d)$. Let $g = \lambda_d(a_1, \dots, a_{2^{\eta(k)}})$. Assume that the output is $(\overline{v}, \overline{w})$ such that $(v, w) \in \theta_d^{\neq}$. (This means that $v, w \in F_{2^{\eta(k)}}$, $v \neq w$, and $v(g) = w(g)$.) If this is not true, then B fails.

2. Find (by exhaustive search) a pair $((\overline{v_1}, \dots, \overline{v_m}), (\overline{w_1}, \dots, \overline{w_m}))$ of m-tuples such that the following conditions hold:

   - $\omega(v_1, \dots, v_m) \in \mathrm{subt}(v)$ and $\omega(w_1, \dots, w_m) \in \mathrm{subt}(w)$;
   - $(v_1(g), \dots, v_m(g)) \neq (w_1(g), \dots, w_m(g))$, but $\omega(v_1(g), \dots, v_m(g)) = \omega(w_1(g), \dots, w_m(g))$. (Of course, $\mathrm{subt}(v) \cup \mathrm{subt}(w) \subseteq \langle a_{i_1}, \dots, a_{i_n} \rangle$, where $1 \le i_1 < \dots < i_n \le 2^{\eta(k)}$ and $n \le \pi(k)$ for some fixed polynomial π.)

   By Lemma 5.1, such a pair exists. (We note that the elements of the $2^{\eta(k)}$-tuple g are distinct. Moreover, these elements are not in $\omega(G_d^m)$ because the last bits of each such element and of any element in $\omega(G_d^m)$ are 0 and 1, respectively. See also Step 3 of the algorithm B in the proof of Theorem 5.3.)

3. Output $(v_1(g) \dots v_m(g), w_1(g) \dots w_m(g))$. (By the last condition of the previous step, together with the definition of ω on G_d, this is a collision for ψ_d. See also Step 4 of the algorithm B in the proof of Theorem 5.3.)

Let $k \in K$ and $\mathbf{d} \leftarrow \mathcal{D}_k$. Then

$$\Pr[A(1^k, 1^k, \mathbf{d}) = (\overline{v}, \overline{w}) \text{ s.t. } (v, w) \in \theta_{\mathbf{d}}^{\neq}] = \Pr[B(\mathbf{d}) \text{ is a collision for } \psi_{\mathbf{d}}] = \mathrm{negl}(k)$$

because Ψ is collision-resistant with respect to $\mathcal{D}$.

For every $k \in K$, denote by $\mathcal{D}'_k$ the distribution of the random variable $(1^k, \mathbf{d})$, where $\mathbf{d} \leftarrow \mathcal{D}_k$. It follows from the above and from Lemma 3.20 that

$$\mathtt{F}' = ((F_{2^{\eta(|u|)}}/\theta_d, \mathrm{nat}_{2^{\eta(|u|)}}/\theta_d, \mathcal{U}(\{\overline{a_1}, \dots, \overline{a_{2^{\eta(|u|)}}}\})) \mid u \in 1^K,\ d \in D_{|u|})$$

is a pseudo-free family of computational m-ary groupoids in $\mathfrak{O}$ with respect to $(\mathcal{D}'_k \mid k \in K)$ and nat.

For each $d \in D$, put $\alpha(d) = (1^{\kappa(d)}, d)$. Then α is a one-to-one function from D onto $\{(u, d) \mid u \in 1^K,\ d \in D_{|u|}\}$. Both α and α^{-1} are polynomial-time computable. Therefore the family $\mathtt{F}'$ can be indexed by D instead of $\{(u, d) \mid u \in 1^K,\ d \in D_{|u|}\}$. Furthermore, $\alpha^{-1}(\mathcal{D}'_k) = \mathcal{D}_k$ for all $k \in K$. Thus, we see that $\mathtt{Q}$ is a pseudo-free family of computational m-ary groupoids in $\mathfrak{O}$ with respect to $\mathcal{D}$ and nat. Moreover, the family $\mathtt{Q}$ has exponential size because $|F_{2^{\eta(\kappa(d))}}/\theta_d| \le |G_d| = 2^{\xi(\kappa(d))}$ for all $d \in D$, where κ and ξ are polynomial parameters on D and K, respectively. □

6 Conclusion

We have initiated the study of (weakly) pseudo-free families of computational Ω-algebras in arbitrary varieties of Ω-algebras. We hope that the assumption of the existence of polynomially bounded or exponential-size (weakly) pseudo-free families in an appropriate variety of Ω-algebras will be useful in mathematical cryptography. The results of the paper show that this assumption can be quite strong, but not unrealistic. Moreover, this assumption can hold in a post-quantum world (see Subsections 5.1–5.2).

Here are some suggestions for further research:

- Find applications of (weakly) pseudo-free families of computational Ω-algebras. For example, construct a cryptographic primitive or a secure cryptographic protocol from a polynomially bounded or exponential-size (weakly) pseudo-free family in a suitable variety of Ω-algebras. See Subsections 3.6 and 4.2 for results in this direction.
- Construct a polynomially bounded or exponential-size (weakly) pseudo-free family in some interesting variety of Ω-algebras under a standard cryptographic assumption. See Subsections 5.1–5.2 for results in this direction.
- Modify the definition of a (weakly) pseudo-free family of computational Ω-algebras to make this definition more useful.

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.

[Ano17] M. Anokhin. Pseudo-free families of finite computational elementary abelian p-groups. *Groups Complexity Cryptology*, 9(1):1–18, 2017. Preliminary version: Cryptology ePrint Archive (`http://eprint.iacr.org/`), Report 2015/1127.

[Ano18] M. Anokhin. A certain family of subgroups of $\mathbb{Z}_n^\star$ is weakly pseudo-free under the general integer factoring intractability assumption. *Groups Complexity Cryptology*, 10(2):99–110, 2018. Preliminary version: Cryptology ePrint Archive (`http://eprint.iacr.org/`), Report 2017/1131.

[BL96] D. Boneh and R. J. Lipton. Algorithms for black-box fields and their application to cryptography. In *Proceedings of CRYPTO 96*, volume 1109 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 1996.

[BS12] S. Burris and H. P. Sankappanavar. *A course in universal algebra*. Available at `http://www.math.uwaterloo.ca/~snburris/htdocs/ualg.html`, the Millennium edition, 2012.

[CFW11] D. Catalano, D. Fiore, and B. Warinschi. Adaptive pseudo-free groups and applications. In *Proceedings of EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 207–223. Springer, 2011. Full version: Cryptology ePrint Archive (`http://eprint.iacr.org/`), Report 2011/053.

[Coh81] P. M. Cohn. *Universal algebra*. D. Reidel Publishing Company, Dordrecht–Boston–London, 1981.

[FHI$^+$13] M. Fukumitsu, S. Hasegawa, S. Isobe, E. Koizumi, and H. Shizuya. Toward separating the strong adaptive pseudo-freeness from the strong RSA assumption. In *Proceedings of the 18th Australasian Conference on Information Security and Privacy (ACISP 2013)*, volume 7959 of *Lecture Notes in Computer Science*, pages 72–87. Springer, 2013.

[FHIS14a] M. Fukumitsu, S. Hasegawa, S. Isobe, and H. Shizuya. On the impossibility of proving security of strong-RSA signatures via the RSA assumption. In *Proceedings of the 19th Australasian Conference on Information Security and Privacy (ACISP 2014)*, volume 8544 of *Lecture Notes in Computer Science*, pages 290–305. Springer, 2014.

[FHIS14b] M. Fukumitsu, S. Hasegawa, S. Isobe, and H. Shizuya. The RSA group is adaptive pseudo-free under the RSA assumption. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Special Section on Cryptography and Information Security*, E97.A(1):200–214, 2014.

[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.

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

[Gol04] O. Goldreich. *Foundations of cryptography. Volume 2 (Basic applications).* Cambridge University Press, 2004.

[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.

[HT07] T. Hirano and K. Tanaka. Variations on pseudo-free groups. Research Reports on Mathematical and Computing Sciences, Series C: Computer Science, C-239, Tokyo Institute of Technology, Department of Mathematical and Computing Sciences, January 2007.

[JB09] M. P. Jhanwar and R. Barua. Sampling from signed quadratic residues: RSA group is pseudofree. In *Proceedings of INDOCRYPT 2009*, volume 5922 of *Lecture Notes in Computer Science*, pages 233–247. Springer, 2009.

[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.

[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].

[Wec92] W. Wechler. *Universal algebra for computer scientists.* Springer, Berlin et al., 1992.