Université Bordeaux 1

Master 1 Cryptologie et Sécurité Informatique

Internship Report

Pseudo-Free Groups

Author:
Firas Kraïem

Supervisor:
Hiroki Shizuya
Department of Computer and
Mathematical Sciences
Graduate School of Information Sciences
Tohoku University
Japan

September 8, 2013

Acknowledgements

I thank Professor Hiroki Shizuya for agreeing to host me in his laboratory for this internship, and everyone in said laboratory for making my stay an enjoyable one. In particular, I thank Masayuki Fukumitsu for sparking my interest in pseudo-free groups. I also thank my teachers at Bordeaux and at UC Irvine for all the knowledge acquired so far, and Professor Gilles Zémor also for his helpful advice. Finally, I thank the French Ministry of Higher Education and Research and the Japan Student Services Organization for their very generous financial support during this internship.

Introduction

The main subject of our study here is *pseudo-free groups*, or, to be more precise, *pseudo-free families of computational groups*. The concept of pseudo-free groups was first introduced by Hohenberger [3], and subsequently formalised by Rivest [6], as a way to unify many common cryptographic assumptions under a common framework.

Most public-key cryptosystems today are based on the perceived intractability of certain mathematical problems. Many of those problems involve computations in a group and have the common property that they can become completely unsolvable if the corresponding group is replaced with a free group. For example, the discrete logarithm problem (given $g, h \in G$, find x such that $h = g^x$) has no solution if G is a free group of rank 2 with generating set $\{g, h\}$. Thus, a solution to the above discrete logarithm problem in G provides a proof that G is *not* a free group of rank 2 with generating set $\{g, h\}$. Another example is the RSA problem: given $e \in \mathbf{N}^*$ (the encryption exponent) and $c \in G$ (the ciphertext), find $m \in G$ (the plaintext) such that $m^e = c$. This problem has no solution if G is the free group generated by c, so a solution to it also provides a proof that G is *not* the free group generated by c.

So, instead of making the assumption that such-and-such problem is difficult to solve in a group, we say once and for all that the given group is difficult to distinguish from a free group, in other words, that it is *pseudo-free*. This assumption, which implies many other common cryptographic assumption, is thus stronger than all of them, and is dubbed by Rivest [6] the "super-strong RSA assumption".

Having formulated the pseudo-freeness assumption, a central problem is then to either prove that a known family of groups is pseudo-free, or to propose new ones. Rivest [6] only conjectured that the family of RSA groups ($\mathbf{Z}_{pq}^*$, with p, q two safe primes) is pseudo-free; this was proved under the strong RSA assumption, first by Micciancio [5] and then by Jhanwar and Barua [4]. We will instead study a result of Anokhin [1], which proves that a certain family of groups of matrices is pseudo-free under a much more common cryptographic assumption: the difficulty of factoring integers.

The rest of this report is organised as follows: Chapter 1 will present some background material, definitions and notation which will be used in subsequent chapters. Chapter 2 will define computational groups and pseudo-free families thereof, following the definitions in the article of Anokhin [1], which differ somewhat from those used in earlier works. Chapter 3 will give general theorems which may be used to prove that a given family of computational groups is pseudo-free, and lemmas which will be used in Chapter 4 to prove that the proposed family satisfies the conditions of the theorems, and is therefore pseudo-free.

Contents

Chapter 1

Background, Definitions, and Notation

Binary Strings In all of the following, $\mathbf{N} = \{0, 1, 2, \dots\}$ and $\mathbf{N}^* = \mathbf{N} \setminus \{0\}$. For an integer $n \in \mathbf{N}$, $\{0,1\}^n$ denotes the set of all binary strings of length n. We note $\{0,1\}^*$ for $\bigcup_{i=0}^{\infty}\{0,1\}^i$ (the set of all finite binary strings), and $\{0,1\}^{\leq n}$ for $\bigcup_{i=0}^{n}\{0,1\}^i$ (the set of all binary strings of length at most n).

We denote by 1^k the k-bit string with all bits 1, and by $|s|$ the length of a string s. Also, for $n \in \mathbf{N}^*$, $\operatorname{bin} n$ denotes the binary string corresponding to the usual binary representation of n (with the most significant bit first). It is known that $|\operatorname{bin} n| = \lfloor \log_2 n \rfloor + 1$.

The Ring $\mathbf{Z}_n$ In the following, $\mathbf{Z}_n$ is the quotient ring $\mathbf{Z}/n\mathbf{Z}$. In particular, we distinguish between an element $k \in \mathbf{Z}$ and the element $k+n\mathbf{Z} \in \mathbf{Z}_n$. The canonical projection homomorphism from $\mathbf{Z}$ to $\mathbf{Z}_n$ will be noted ν_n. $\mathbf{Z}_n^*$ is the group of units of $\mathbf{Z}_n$.

Polynomials Here, for simplicity, a *polynomial* means a function $\pi : \mathbf{N} \to \mathbf{N}^*$ such that $\pi(n) = cn^d$, for some $c \in \mathbf{N}^*$ and $d \in \mathbf{N}$, if $n \neq 0$ ($\pi(0)$ can be any arbitrary positive integer).

Negligible Functions Let K be an infinite set of non-negative integers. A function $\epsilon : K \to \mathbf{R}^+$ is *negligible* if for every polynomial π there exists a non-negative integer n such that $\epsilon(k) \leq 1/\pi(k)$ for all $k \geq n$.

Probability Distributions and Random Variables Given a probability distribution $\mathcal{Y}$ with sample space Y, we use the notation $\mathbf{y}_1, \dots, \mathbf{y}_n \leftarrow \mathcal{Y}$ to denote the fact that $\mathbf{y}_1, \dots, \mathbf{y}_n$ (denoted by Roman bold letters) are n independent random variables distributed acording to $\mathcal{Y}$. (All random variables defined in such a way are assumed to be independent.) We also write $\mathrm{y}_1, \dots, \mathrm{y}_n \leftarrow \mathcal{Y}$ (Roman medium-weight letters) to indicate that $\mathrm{y}_1, \dots, \mathrm{y}_n$ are elements of Y chosen independently according to $\mathcal{Y}$. We also denote by $\mathcal{U}(S)$ the uniform probability distribution on a finite or countably infinite sample space S.

Given a probability distribution $\mathcal{Y}$ on a sample space Y, an element $y \in Y$, and a subset $M \subseteq Y$, we note $P_{\mathcal{Y}}(y)$ and $P_{\mathcal{Y}}(M)$ respectively the probabilities that a random variable with probability distribution $\mathcal{Y}$ takes the value y or some value $x \in M$. The *support* of $\mathcal{Y}$ (*i.e.*, the set $\{y \in Y, P_{\mathcal{Y}}(y) \neq 0\}$) is noted $\operatorname{supp} \mathcal{Y}$.

Statistical Distance Given two probability distributions $\mathcal{P}$ and $\mathcal{Q}$ on the same finite or countably infinite sample space S, we define their *statistical distance* (or *variation distance*), noted $\Delta(\mathcal{P}, \mathcal{Q})$, by

$$\Delta(\mathcal{P}, \mathcal{Q}) = \frac{1}{2} \sum_{s \in S} |P_{\mathcal{P}}(s) - P_{\mathcal{Q}}(s)|.$$

It is known (see [8, Section 8.8]) that we have also

$$\Delta(\mathcal{P}, \mathcal{Q}) = \max_{M \subseteq S} |P_{\mathcal{P}}(M) - P_{\mathcal{Q}}(M)|.$$

Polynomial-Time Samplable Probability Ensembles Let $I \subseteq \{0,1\}^*$ or $I \subseteq \mathbf{N}$, $D \subseteq \{0,1\}^*$, and $\mathcal{E} = (\mathcal{E}_i)_{i \in I}$ be a family or probability distributions on D. $\mathcal{E}$ is said to be *polynomial-time samplable* if there exists a probabilistic polynomial-time algorithm A such that for all $i \in I$, the probability distribution of the random variable $A(i)$ or $A(1^i)$ coincides with $\mathcal{E}_i$.

Groups For a (finite) group G, we denote by $|G|$ the order of G. For a subset S of G, $\langle S \rangle$ is the subgroup of G generated by the elements of S (*i.e.*, the smallest subgroup of G which contains S). For an element $g \in G$, we denote by $\operatorname{ord} g$ the order of g (in G).

Representations of the Elements of a Group In the following, a map ρ from a subset of $\{0,1\}^*$ onto a group G must be understood as a representation of elements of G as binary strings. For an element g of G, we note $[g]_\rho$ an arbitrary string $r \in \{0,1\}^*$ such that $\rho(r) = g$. In other words, a *representation* of g as a binary string. Note that we do not require that ρ be injective, so an element of G may have more than one representation.

Free Groups and Group Varieties We recall briefly the basic properties of free groups and group varieties (for a thorough treatment, see for example [7, Chapter 12]). Given a non-empty set X, we say that a group F is a *free group on* X if there exists a map $\kappa : X \to F$ such that if G is any group and f is any map from X to G, there is exactly one group homomorphism $\tau_f : F \to G$ such that $f = \tau_f \circ \kappa$.

Let $X^{-1} = \{x^{-1}, x \in X\}$ and $X' = X \cup X^{-1}$. Then we can define the set X'^* of words over X^{-1}, for example if $X = \{a, b\}$, X'^* contain words such as $a^{-1}bab^{-1}a^{-1}$. We define an operation called *reduction* where we just suppress all pairs of symbols of the form xx^{-1} or $x^{-1}x$, so for example a reduction of $abb^{-1}ba$ is $abba$. We say that a word is *reduced* if no reduction can be applied to it. Then the set of reduced words of X'^* with the operation of concatenation followed by reduction is a free group on X, which in the following we call just "the free group on X". Thus, when we talk of the free group freely generated by some set of generators, it must be understood as the group of reduced words on X as defined above, where X is the set of given generators.

We recall also that a *group variety* is a class of groups which is closed under subgroups, quotients, and direct products. Given a group variety $\mathfrak{V}$, we can define free $\mathfrak{V}$-groups in an analogous manner as free groups. In our construction $\mathfrak{V}$ is the variety of all groups, so free $\mathfrak{V}$-groups are just free groups as defined above.

For convenience, we use several notations for free groups (on a group variety $\mathfrak{V}$):

- $F_{\infty,\infty}(\mathfrak{V})$ is the group freely generated by $a_1, a_2, \ldots, x_1, x_2, \ldots$;
- $F_\infty(\mathfrak{V})$ is the group freely generated by $a_1, a_2, \ldots$;
- if $m, n \in \mathbf{N}$, then $F_{m,n}(\mathfrak{V})$ is the group freely generated by $a_1, \ldots, a_m, x_1, \ldots, x_n$; and

- if $m \in \mathbf{N}$, then $F_m(\mathfrak{V})$ is the group freely generated by $a_1, \ldots, a_m$.

Also, for brevity, we will denote the group word $v(a_1, \ldots, a_m, x_1, \ldots, x_n) \in F_{m,n}(\mathfrak{V})$ by $v(a; x)$. If $g_1, \ldots, g_m, h_1, \ldots, h_n$ are elements of a group G, we will also analogously denote the element $v(g_1, \ldots, g_m, h_1, \ldots, h_n)$ of G obtained from $v(a; x)$ in the obvious manner by $v(g; h)$.

Chapter 2

Pseudo-Free Groups

2.1 Computational Groups

Individual Computational Groups Let

- G be a group;
- ρ be a map from a subset of $\{0,1\}^*$ onto G; and
- $\mathcal{R}$ be a probability distribution on $\operatorname{dom}\rho$.

Then the triple $(G, \rho, \mathcal{R})$ is a *computational group* if the following hold:

- The following computations can be done in deterministic polynomial time for any $f, g \in G$:
  - Given $[f]_\rho, [g]_\rho$, decide whether $f = g$
  - Given $[f]_\rho, [g]_\rho$, compute $[fg]_\rho$
  - Given $[g]_\rho$, compute $[g^{-1}]_\rho$
- There exists a probabilistic constant-time algorithm which takes no input and outputs a string in $\operatorname{dom}\rho$ (*i.e.*, a representation of an element of G) distributed according to $\mathcal{R}$.

Uniform Families of Computational Groups Let D be a subset of $\{0,1\}^*$ and $((G_d, \rho_d, \mathcal{R}_d))_{d\in D}$ be a family of triples where G_d is a group, ρ_d is a map from a subset of $\{0,1\}^*$ onto G_d, and $\mathcal{R}_d$ is a probability distribution on $\operatorname{dom}\rho_d$. $((G_d, \rho_d, \mathcal{R}_d))_{d\in D}$ is a *uniform family of computational groups* if the following hold:

- The following operations can be performed in deterministic polynomial time for any $d \in D$ and $f, g \in G_d$:
  - Given $d \in D$ and $[f]_{\rho_d}, [g]_{\rho_d}$, decide whether $f = g$
  - Given $d \in D$ and $[f]_{\rho_d}, [g]_{\rho_d}$, compute $[fg]_{\rho_d}$
  - Given $d \in D$ and $[g]_{\rho_d}$, compute $[g^{-1}]_{\rho_d}$
  - Given $d \in D$, compute $[1]_{\rho_d}$
- The probability ensemble $(\mathcal{R}_d)_{d\in D}$ is polynomial-time samplable. That is, there exists a probabilistic polynomial-time algorithm A which, on input d, outputs an element of $\operatorname{dom}\rho_d$ (*i.e.*, a representation of an element of G_d) chosen according to the distribution $\mathcal{R}_d$.

In the following, all families of computational groups are assumed to be uniform.

Exponential Size A family $((G_d, \rho_d, \mathcal{R}_d))_{d \in D}$ of computational groups is said to have *exponential size* if there exists a polynomial π such that $|G_d| \leq 2^{\pi(|d|)}$ for all $d \in D$.

2.2 Pseudo-Free Families of Computational Groups

Informally, we are interested in (families of) computational groups which, to an attacker, are indistinguishable, in a sense we will make precise in this section, from free groups. In other words, such that it is computationally infeasible for an attacker to prove that the groups are not free. Of course, since our groups are most often finite, it is obvious that they are not free since free groups are infinite. Thus, we ask for a different kind of proof: we ask that the attacker provide what we call a *non-freeness witness* for the group, in an analogy with the compositeness witnesses appearing in compositeness tests such as the Miller-Rabin test (this is not standard terminology).

Non-Freeness Witnesses Recall that a computational group consists of a group G, a representation ρ of the elements of G as binary strings, and a probability distribution $\mathcal{R}_d$ on the representations of the elements of G. We define non-freeness witnesses with respect to a particular group variety $\mathfrak{V}$, a particular representation σ of the elements of $F_{\infty,\infty}(\mathfrak{V})$ as binary strings, and a particular tuple $(f_1, \ldots, f_m)$ of elements of G. A non-freeness witness for $(G, \mathfrak{V}, \sigma, \rho, (f_1, \ldots, f_m))$ consists of a tuple $(g_1, \ldots, g_\ell)$ of (representations of) elements of G and a system of equations, given as (representations of) words in the $\mathfrak{V}$-free group freely generated by $a_1, \ldots, a_m, x_1, \ldots, x_\ell$, such that the given equations are unsatisfiable, over variables $x_1, \ldots, x_\ell$, in the free $\mathfrak{V}$-group freely generated by $a_1, \ldots, a_m$, but are satisfiable in G with solutions $g_1, \ldots, g_\ell$ if we replace $a_1, \ldots, a_m$ by $f_1, \ldots, f_m$ respectively. The formal definition of non-freeness witnesses is as follows, and we then give two examples:

Definition 1. Let $\mathfrak{V}$ be a variety of groups, and σ be a map from a subset of $\{0,1\}^*$ to $F_{\infty,\infty}(\mathfrak{V})$. For a group $G \in \mathfrak{V}$, a map ρ from a subset of $\{0,1\}^*$ onto G, an integer $m \geq 0$, and $f_1, \ldots, f_m \in G$, a tuple

$$(([v_1(a;x)]_\sigma, [w_1(a;x)]_\sigma), \ldots, ([v_s(a;x)]_\sigma, [w_s(a;x)]_\sigma), ([g_1]_\rho, \ldots, [g_\ell]_\rho))$$

is a *non-freeness witness* for $(G, \mathfrak{V}, \sigma, \rho, (f_1, \ldots, f_m))$ if the following hold:

- $s \geq 1$, $\ell \geq 0$
- $v_i(a;x), w_i(a;x) \in F_{m,\ell}(\mathfrak{V})$ for all $i \in \{1, \ldots, s\}$
- $g_i \in G$ for all $i \in \{1, \ldots, \ell\}$
- The system of equations

  $$v_i(a;x) = w_i(a;x), \qquad i = 1, \ldots, s$$

  over variables $x_1, \ldots, x_\ell$ is unsatisfiable in the free group $F_m(\mathfrak{V})$
- $v_i(f_1, \ldots, f_m, g_1, \ldots, g_\ell) = w_i(f_1, \ldots, f_m, g_1, \ldots, g_\ell)$ in G for all $i \in \{1, \ldots, s\}$

Example (RSA) Let $p = 59$, $q = 83$, $n = pq = 4897$, we have $\varphi(n) = 4756$. Let the public encryption exponent be $e = 3$, and the private decryption exponent is then $d = 3171$. Encryption of the message $m = 1234$ yields $c = 1234^3 = 4064$, so an attacker knows $e = 3$ and $c = 4064$, and wants to find m such that $m^e = c$.

Let $G = \mathbf{Z}_n^*$, $\mathfrak{V}$ be the variety of abelian groups (or the variety of all groups), σ, ρ be suitable representations of the elements of $F_{\infty,\infty}(\mathfrak{V})$ and G respectively, and $f_1 = c = 4064$. Then, breaking the instance of RSA above amounts to finding the non-freeness witness $(([x_1^3]_\sigma, [a_1]_\sigma), ([1234]_\rho))$ for $(G, \mathfrak{V}, \sigma, \rho, (4064))$. Indeed, the equation $x_1^3 = a_1$ is unsatisfiable in the free (abelian) group generated by a_1, but we do have $1234^3 = 4064$ in G.

Example (Discrete Logarithm) Let $p = 1019$ and $g = 3$. We have $3^{321} = 373 \pmod{1019}$. Let $G = \mathbf{Z}_p^*$, $\mathfrak{V}$ be the variety of all groups (or the variety of abelian groups), σ, ρ be suitable representations, and $(f_1, f_2) = (3, 373)$. Computing the discrete logarithm $\log_3 373 = 321$ then amounts to finding the non-freeness witness $(([a_1^{321}]_\sigma, [a_2]_\sigma), ())$ for $(G, \mathfrak{V}, \sigma, \rho, (f_1, f_2))$. Indeed, the equation $a_1^{321} = a_2$ is unsatisfiable in the free (or free abelian) group generated by (a_1, a_2), but we do have $3^{321} = 373$ in G.

Pseudo-Free Families of Computational Groups We now define pseudo-free families of computational groups, starting as above with an informal definition. We first need an infinite set K of non-negative integers, with values of K being parameters for pseudo-freeness. We then let $\mathcal{D} = (\mathcal{D}_k)_{k \in K}$ be a polynomial time samplable probability ensemble consisting of probability distributions on D. Recall that D is the indexing set for our family of computational groups, so $\mathcal{D}$ indicates how to choose a group of the family for a given value of the parameter k. For example, let $((G_d, \rho_d, \mathcal{R}_d))_{d \in D}$ be the family of all RSA groups (with suitable representations ρ_d and probability distributions $\mathcal{R}_d$). Then K can be the set of all non-negative integers above some minimum value (say, 1024), and the support of $\mathcal{D}_k$ can be the set of all d such that if we let $G_d = \mathbf{Z}_n^*$, n is k-bit long.

The family $((G_d, \rho_d, \mathcal{R}_d))_{d \in D}$ is then said to be pseudo-free if, for all polynomials π and probabilistic polynomial-time algorithm A, the probability that A, on input $k \in K$, proceeds as follows:

1. Choose a value $d \in D$ according to the distribution $\mathcal{D}_k$
2. Choose elements $r_1, \ldots, r_{\pi(k)}$ of G_d independently according to the distribution $\mathcal{R}_d$

and then outputs a non-freeness witness for $(G_d, \mathfrak{V}, \sigma, \rho_d, (r_1, \ldots, r_{\pi(k)}))$ is negligible as a function of k. Note that we do not allow an adversary to choose arbitrary values for $r_1, \ldots, r_{\pi(k)}$, because that would allow him to choose a value which makes finding a non-freeness witness trivial (for example, in a RSA group, a value of 4 makes it trivial to find a non-freeness witness using the equation $2^2 = 4$). Thus we require that the values $r_1, \ldots, r_{\pi(k)}$ be chosen at random according to a specified probability distribution. The formal definition of pseudo-free families of computational groups is as follows.

Definition 2. Let K be a set of non-negative integers, $\mathcal{D} = (\mathcal{D}_k)_{k \in K}$ be a polynomial-time samplable probability ensemble consisting of distributions on D. The family of computational groups $((G_d, \rho_d, \mathcal{R}_d))_{d \in D}$ is called *pseudo-free in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ* if for any polynomial π and probabilistic polynomial-time algorithm A, the following holds.

For every $k \in K$ let $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{r}_1, \ldots, \mathbf{r}_{\pi(k)} \leftarrow \mathcal{R}_\mathbf{d}$, and $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_{\pi(k)})$. Then the probability that A, on input $(1^k, \mathbf{d}, \mathbf{r})$, outputs a non-freeness witness for $(G_\mathbf{d}, \mathfrak{V}, \sigma, \rho_\mathbf{d}, \rho_\mathbf{d}(\mathbf{r}))$ is negligible as a function of k.

Chapter 3

Main Results

3.1 Theorems

The two following theorems give general methods to construct families of computational groups. In the following section, we give several lemmas which will be used in the last section to show that the proposed family of computational groups satisfies the conditions of the theorems, and is therefore pseudo-free. We remark that the theorems of this section are very general, and thus could be used to construct other pseudo-free families of computational groups.

The idea of the first theorem is that $F_{2^k}(\mathfrak{V})$ is pseudo-free if, with the notations of the previous sections, the support of each probability distribution $\mathcal{R}_d$ is the set of (representations of) generators of $F_{2^k}(\mathfrak{V})$. In other words, if all our chosen elements $\mathbf{r}_i$ are generators of $F_{2^k}(\mathfrak{V})$. Indeed, when we pick $\pi(k)$ generators at random, a non-freeness witness with that set of generators exists if and only if we pick the same generator at least twice. This happens with negligle probability as a function of k, since we pick a polynomial number of generators but there is exponentially many of them.

Theorem 1. *Let $D = \{1^k, k \in K\}$, and M be a set of integers such that $1 \in M$ and $-m \in M$ whenever $m \in M$. For all $k \in K$, let $\mathcal{D}_k$ be the probability distribution which takes the value 1^k with probability 1. Also, for each $1^k \in D$, let ρ_{1^k} be the mapping of*

$$\{((i_1, m_1), \ldots, (i_n, m_n)), i_j \in \{1, \ldots, 2^k\}, m_j \in M\}$$

onto $F_{2^k}(\mathfrak{V})$ defined by

$$\rho_{1^k}((i_1, m_1), \ldots, (i_n, m_n)) = a_{i_1}^{m_1} \ldots a_{i_n}^{m_n}.$$

Finally, let $\mathcal{R}_{1^k}$ be the distribution of the random variable $((\mathbf{i}, 1))$ where $\mathbf{i} \leftarrow \mathcal{U}(\{1, \ldots, 2^k\})$. We note that $((i, 1))$ is a representation of a_i for all $i \in \{1, \ldots, 2^k\}$.

Assume that there exists a deterministic polynomial-time algorithm which, given $1^k \in D$ and $[f]_{\rho_{1^k}}, [g]_{\rho_{1^k}}$ for any $f, g \in F_{2^k}(\mathfrak{V})$, decides whether $f = g$. Then $\Gamma = ((F_{2^k}(\mathfrak{V}), \rho_{1^k}, \mathcal{R}_{1^k}))_{1^k \in D}$ is a pseudo-free family of computational groups in $\mathfrak{V}$ with respect to $\mathcal{D} = (\mathcal{D}_k)_{k \in K}$ and σ.

Proof. It is easy to see that Γ is a family of computational groups. Let π be a polynomial and A be a probabilistic polynomial-time algorithm which, for all $k \in K$, on input $(1^k, 1^k, (((i_1, 1)), \ldots, ((i_{\pi(k)}, 1))))$, outputs a non-freeness witness for $(F_{2^k}(\mathfrak{V}), \mathfrak{V}, \sigma, \rho_{1^k}, (a_{i_1}, \ldots, a_{i_{\pi(k)}}))$. Then, for all $t = 1, \ldots, s$, there exist $v_t(a; x), w_t(a; x) \in F_{\pi(k),\ell}(\mathfrak{V})$ such that the system of equations

$$v_t(a_1, \ldots, a_{\pi(k)}; x_1, \ldots, x_\ell) = w_t(a_1, \ldots, a_{\pi(k)}; x_1, \ldots, x_\ell), \qquad t = 1, \ldots, s$$

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

$$v_t(a_{i_1},\ldots,a_{i_{\pi(k)}};x_1,\ldots,x_\ell) = w_t(a_{i_1},\ldots,a_{i_{\pi(k)}};x_1,\ldots,x_\ell), \qquad t=1,\ldots,s$$

is satisfiable in $F_{2^k}(\mathfrak{V})$, over variables $x_1,\ldots,x_\ell$. This implies that there must exist j,j' such that $j \neq j'$ and $i_j = i_{j'}$. Since $i_1,\ldots,i_{\pi(k)}$ are chosen independently and uniformly at random from $\{1,\ldots,2^k\}$, this happens with probability $\frac{\pi(k)(\pi(k)-1)}{2^{k+1}}$, which is clearly negligible as a function of k. □

The second theorem shows how to construct a new pseudo-free family of computational groups from a known one, by working in quotient groups of the groups of the family. Namely, for each group G_d of a pseudo-free family of computational groups, we define a family of normal subgoups $H_{d,e}$ such that we can decide in polynomial time whether an element of G_d is in $H_{d,e}$, but not generate elements of $H_{d,e} \setminus \{1\}$ in polynomial time with non-negligible probabilty. Then the family of groups $G_d/H_{d,e}$ is pseudo free, because finding a non-freeness witness for that family implies either finding a non-freeness witness for the family of groups G_d or finding a non-identity element of $H_{d,e}$, both of which cannot be done in probabilistic polynomial time with non-negligible probability.

Theorem 2. *Let $((G_d,\rho_d,\mathcal{R}_d))_{d\in D}$ be a pseudo-free family of computational groups in $\mathfrak{V}$ with respect to $\mathcal{D}$ and σ. Let η be a polynomial, and $(\mathcal{E}_d)_{d\in D}$ be a polynomial-time samplable probability ensemble consisting of distributions on a subset E_d of $\{0,1\}^{\leq\eta(|d|)}$ for all d. For all pairs $(d,e), d\in D, e\in E_d$, let $H_{d,e}$ be a normal subgroup of G_d. Assume further that the following computations can be performed in deterministic polynomial time:*

(i) Given $d\in D$, $[u(a;x)]_\sigma$ for some $u(a;x)\in F_{m,\ell}(\mathfrak{V})$, and $([f_1]_{\rho_d},\ldots,[f_m]_{\rho_d},[g_1]_{\rho_d},\ldots,[g_\ell]_{\rho_d})$ for some $f_1,\ldots,f_m,g_1,\ldots,g_\ell\in G_d$, compute $[u(f_1,\ldots,f_m;g_1,\ldots,g_\ell)]_{\rho_d}$

(ii) Given $d\in D$, $e\in E_d$, and $[g]_{\rho_d}$ for some $g\in G_d$, decide whether $g\in H_{d,e}$

and that

(iii) For all $k\in K$, $\mathbf{d}\leftarrow\mathcal{D}_k$, $\mathbf{e}\leftarrow\mathcal{E}_d$, any probabilistic polynomial-time algorithm, running on input $(1^k,\mathbf{d},\mathbf{e})$, outputs $[h]_{\rho_d}$ for a non-identity element h of $H_{d,e}$ with negligible probability (as a function of k).

For any $k\in K$, let $\mathcal{D}'_k$ be the distribution of the random variable $(\mathbf{d},\mathbf{e})$, where $\mathbf{d}\leftarrow\mathcal{D}_k$ and $\mathbf{e}\leftarrow\mathcal{E}_d$. Further, for every $d\in D$ and $e\in E_d$, define the map $\rho'_{d,e}:\operatorname{dom}\rho_d\to G_d/H_{d,e}$ by $\rho'_{d,e}(r)=\rho_d(r)H_{d,e}$. Then the family of computational groups $\Gamma=((G_d/H_{d,e},\rho'_{d,e},\mathcal{R}_d))_{d\in D,e\in E_d}$ is pseudo-free with respect to $(\mathcal{D}'_k)_{k\in K}$ and σ.

Proof. First we show that Γ is a family of computational groups. Since a representation of an element $gH_{d,e}$ of $G_d/H_{d,e}$ is a representation of some element of the coset $gH_{d,e}$ in G, and since we can by hypothesis multiply representations of elements of G in deterministic polynomial time, it follows that we can also multiply representations of elements of $G_d/H_{d,e}$ in deterministic polynomial time. A similar argument shows that we can compute representations of inverses in $G_d/H_{d,e}$ in deterministic polynomial time by computing the inverse (in G) of the given representative. For equality testing, representations of two elements $gH_{d,e}$ and $hH_{d,e}$ are representations of g and h respectively, and it is known that $gH_{d,e}=hH_{d,e}$ if and only if $g^{-1}h\in H_{d,e}$. Since we can by hypothesis decide in deterministic polynomial time whether an element of G_d is in $H_{d,e}$, it follows that we can decide in deterministic polynomial time whether $gH_{d,e}=hH_{d,e}$. Finally, the probability ensemble $(\mathcal{R}_d)_{d\in D}$ is polynomial-time samplable by hypothesis.

Now suppose Γ is not pseudo-free. That is, there exist a polynomial π and a probabilistic polynomial-time algorithm A which, for all $k \in K$, $\mathbf{d} \leftarrow \mathcal{D}_k$, $\mathbf{e} \leftarrow \mathcal{E}_d$, $\mathbf{r}_1, \ldots, \mathbf{r}_{\pi(k)} \leftarrow \mathcal{R}_d$, and $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_{\pi(k)})$. on input $(1^k, (\mathbf{d}, \mathbf{e}), \mathbf{r})$, outputs a non-freeness witness

$$(([v_1(a;x)]_\sigma, [w_1(a;x)]_\sigma), \ldots, ([v_s(a;x)]_\sigma, [w_s(a;x)]_\sigma), (t_1, \ldots, t_\ell))$$

for $(G_{\mathbf{d}}/H_{\mathbf{d},\mathbf{e}}, \mathfrak{V}, \sigma, \rho'_{\mathbf{d},\mathbf{e}}, \rho'_{\mathbf{d},\mathbf{e}}(\mathbf{r}))$ with non-negligible probability. We will show that this contradicts one of our hypotheses.

Running A on input $(1^k, (\mathbf{d}, \mathbf{e}), \mathbf{r})$, we obtain a non-freeness witness for some group $G_d/H_{d,e}$. That is, we have $v_i(\rho_{d,e}(r); g) = w_i(\rho_{d,e}(r); g)$ in $G_d/H_{d,e}$ for all $i = 1, \ldots, s$, and so $v_i(\rho_d(r); g) \equiv w_i(\rho_d(r); g) \pmod{H_{d,e}}$ in G_d. The probability that $v_i(\rho_d(r); g) = w_i(\rho_d(r); g)$ in G_d for all i must be negligible, since otherwise A would output a non-freeness witness for G_d with non-negligible probability, which is impossible in probabilistic polynomial time by assumption. This means that, with non-negligible probability, there is some i such that $v_i(\rho_d(r); g) \neq w_i(\rho_d(r); g)$ in G_d.

By assumption, we can then compute $[v_i(\rho_d(r); g)]_{\rho_d}$ and $[w_i(\rho_d(r); g)]_{\rho_d}$ in deterministic polynomial time, and so we can compute $[v_i(\rho_d(r); g)^{-1} w_i(\rho_d(r); g)]_{\rho_d}$. But $v_i(\rho_d(r); g)^{-1} w_i(\rho_d(r); g) \in H_{d,e} \setminus \{1\}$, contradicting our assumption that we cannot compute elements of $H_{d,e} \setminus \{1\}$ in probabilistic polynomial time with non-negligible probability. □

3.2 Lemmas

The following lemma says that, given an integer $n \in \mathbf{N}$, we can generate, in probabilistic polynomial time, elements of $\mathbf{Z}_n^*$ according to a distribution which is arbitrarily close (in terms of statistical distance) to a uniform distribution.

Lemma 1. *Let π be a polynomial. Then there exists a probability ensemble $(\mathcal{Z}_n)_{n \in \mathbf{N}^*}$ such that the following hold:*

(i) For any $n \in \mathbf{N}^*$, the support of $\mathcal{Z}_n$ is the set of integers in $\{0, \ldots, n-1\}$ which are coprime with n.

(ii) For all $n \in \mathbf{N}^*$, $\Delta(\nu_n(\mathcal{Z}_n), \mathcal{U}(\mathbf{Z}_n^*)) \leq 2^{-\pi(|\operatorname{bin} n|)}$.

(iii) The probability ensemble $(\mathcal{Z}_{\operatorname{bin}^{-1} s})_{s \in \operatorname{bin}(\mathbf{N}^*)}$, where $\operatorname{bin}^{-1} s$ denotes the integer whose binary representation is s, is polynomial-time samplable.

Proof. Let η be a polynomial such that $|\mathbf{Z}_n^*|/n \geq 1/\eta(|\operatorname{bin} n|)$ for all $n \in \mathbf{N}^*$ (this is possible because $|\mathbf{Z}_n^*|/n = \Omega(1/\log\log n)$). Let $n \in \mathbf{N}^*$ and $\ell = |\operatorname{bin} n|$. For brevity, let I_n be the set of all integers in $\{0, \ldots, n-1\}$ which are coprime to n (note that the canonical projection homomorphism ν_n then defines a bijection between I_n and $\mathbf{Z}_n^*$). Let A be a probabilistic polynomial time algorithms which, on input n, performs the following iteration at most $2\eta(n)\pi(n)$ times:

1. Choose $\mathrm{m} \leftarrow \mathcal{U}(\{0, \ldots, 2^{\lceil \log_2 n \rceil} - 1\}$
2. If $m \in I_n$, output m and terminate

If the algorithm does not terminate after $2\eta(n)\pi(n)$ steps, output 1 and terminate (note that this can happen only if $n \geq 2$). Note also that if $n \geq 2$, then the algorithm always outputs an element of I_n. Since it is clear that the algorithm runs in polynomial time, if we let $\mathcal{Z}_n$ be the distribution of the random variable $A(n)$, conditions (i) and (iii) are evident.

For condition (ii), let S_n be the event that A on input n terminates at some iteration, and S'_n be its complementary event. Then we have

$$\begin{aligned}
\Delta(\nu_n(\mathcal{Z}_n), \mathcal{U}(\mathbf{Z}_n^*)) &= \Delta(\mathcal{Z}_n, \mathcal{U}(I_n)) \\
&= \max_{M \subseteq I_n} |P(A_n \in M) - P_{\mathcal{U}(I_n)}(M)| \\
&= \max_{M \subseteq I_n} |P(A_n \in M | \mathsf{S}_n) P(\mathsf{S}_n) + P(A \in M, \mathsf{S}'_n) - P_{\mathcal{U}(I_n)}(M)| \\
&= \max_{M \subseteq I_n} |P_{\mathcal{U}(I_n)}(M) P(\mathsf{S}_n) + P(A \in M, \mathsf{S}'_n) - P_{\mathcal{U}(I_n)}(M)| \\
&= \max_{M \subseteq I_n} |P(A \in M, \mathsf{S}'_n) + (P(\mathsf{S}_n) - 1) P_{\mathcal{U}(I_n)}(M)| \\
&= \max_{M \subseteq I_n} |P(A \in M, \mathsf{S}'_n) - P(\mathsf{S}'_n) P_{\mathcal{U}(I_n)}(M)|
\end{aligned}$$

and since, for all M, $0 \leq P(A \in M, \mathsf{S}'_n) \leq P(\mathsf{S}'_n)$ and $0 \leq P(\mathsf{S}'_n) P_{\mathcal{U}(I_n)}(M) \leq P(\mathsf{S}'_n)$, we obtain

$$0 \leq P(A \in M, \mathsf{S}'_n) - P(\mathsf{S}'_n) P_{\mathcal{U}(I_n)}(M) \leq P(\mathsf{S}'_n),$$

and so

$$|P(A \in M, \mathsf{S}'_n) - P(\mathsf{S}'_n) P_{\mathcal{U}(I_n)}(M)| \leq P(\mathsf{S}'_n),$$

and finally

$$\max_{M \subseteq I_n} |P(A \in M, \mathsf{S}'_n) - P(\mathsf{S}'_n) P_{\mathcal{U}(I_n)}(M)| \leq P(\mathsf{S}'_n).$$

Now,

$$\begin{aligned}
P(\mathsf{S}'_n) &= \left(1 - \frac{|I_n|}{2^{\lceil \log_2 n \rceil}}\right)^{2\eta(\ell)\pi(\ell)} \\
&= \left(1 - \frac{n}{2^{\lceil \log_2 n \rceil}} \cdot \frac{|\mathbf{Z}_n^*|}{n}\right)^{2\eta(\ell)\pi(\ell)} \\
&\leq \left(1 - \frac{n}{2^{\lceil \log_2 n \rceil}} \cdot \frac{1}{\eta(\ell)}\right)^{2\eta(\ell)\pi(\ell)} \\
&\leq \left(1 - \frac{n}{2^{\lceil \log_2 n \rceil - 1}} \cdot \frac{1}{2\eta(\ell)}\right)^{2\eta(\ell)\pi(\ell)}.
\end{aligned}$$

But it is known that $\ell = \lfloor \log_2 n \rfloor + 1$, and so $\lceil \log_2 n \rceil = \ell$ if $n \neq 2^k$, and $\ell - 1$ otherwise. Thus $\lceil \log_2 n \rceil - 1 < \ell$ for all n. This means that $2^{\lceil \log_2 n \rceil - 1} \leq n$, and so $\frac{n}{2^{\lceil \log_2 n \rceil - 1}} \geq 1$. Now

$$\begin{aligned}
P(\mathsf{S}'_n) &\leq \left(1 - \frac{1}{2\eta(\ell)}\right)^{2\eta(\ell)\pi(\ell)} \\
&\leq \left(\left(1 - \frac{1}{2\eta\ell}\right)^{2\eta(\ell)}\right)^{\pi(\ell)},
\end{aligned}$$

and since the function $x \mapsto \left(1 - \frac{1}{x}\right)^x$ is bounded above (over positive real numbers) by $1/e$, we obtain finally

$$\begin{aligned}
P(\mathsf{S}'_n) &\leq \left(\frac{1}{e}\right)^{\pi(\ell)} \\
&\leq e^{-\pi(\ell)} \\
&\leq 2^{-\pi(\ell)},
\end{aligned}$$

which completes the proof. □

For the proof of the following lemma, see [2, Algorithm 2.3.5 and Lemma 2.3.6].

Lemma 2. *There exists a deterministic polynomial-time algorithm which, given an integer $n \geq 2$, decides whether there exist integers $a, b \geq 2$ such that $n = a^b$ (*i.e.*, whether n is a perfect power), and if so, outputs such a pair (a, b).*

The following is well-known:

Lemma 3. *Let $n \in \mathbf{N}^*$ and $y \in \mathbf{N}$. If $y \not\equiv \pm 1 \pmod{n}$ and $y^2 \equiv 1 \pmod{n}$, then $\gcd(y \pm 1, n)$ are non-trivial divisors of n.*

Proof. Since $y - 1 \not\equiv 0 \pmod{n}$, n is not a divisor of $y - 1$, and so $\gcd(y - 1, n) \neq n$. Likewise, $\gcd(y + 1, n) \neq n$. On the other hand, $y^2 - 1 = (y + 1)(y - 1) \equiv 0 \pmod{n}$ means that n divides $(y + 1)(y - 1)$. Assuming for contradiction that $\gcd(y - 1, n) = 1$, it follows that n divides $y + 1$, which is false. So $\gcd(y - 1, n) \neq 1$ is a non-trivial divisor of n, and likewise for $\gcd(y + 1, n)$. □

Finally:

Lemma 4. *Let n be an odd positive integer and $\tau(n)$ be the number of prime divisors of n. Also, let $\mathbf{u} \leftarrow \mathcal{U}(\mathbf{Z}_n^*)$. Then*

$$P(\operatorname{ord} \mathbf{u} \text{ is odd or } -1 + n\mathbf{Z} \in \langle \mathbf{u} \rangle) \leq \frac{1}{2^{\tau(n)-1}}.$$

Chapter 4

Construction

In this section, $\mathfrak{V}$ is the variety of all groups, and σ is the map from

$$\{((b_1, i_1, m_1), \ldots, (b_n, i_n, m_n)) | n \in \mathbf{N}, b_j \in \{a, x\}, i_j \in \mathbf{N}^*, m_j = \pm 1\}$$

onto $F_{\infty,\infty}$ defined by

$$\sigma((b_1, i_1, m_1), \ldots, (b_n, i_n, m_n)) = (b_1)_{i_1}^{m_1} \ldots (b_n)_{i_n}^{m_n},$$

where $(b)_i$ denotes a_i if $b = a$ and x_i if $b = x$.

We formulate our assumption of the general intractability of the integer factoring problem in the language of the previous sections as follows:

General Integer Factoring Intractability Assumption There exist an infinite set K of non-negative integers and a polynomial-time samplable probability ensemble $(\mathcal{N}_k)_{k \in K}$ such that the following hold:

- For all $k \in K$, the support of $\mathcal{N}_k$ is a set of composite integers.
- If $\mathbf{n} \leftarrow \mathcal{N}_k$ and A is a probabilistic polynomial-time algorithm, the probability that A, on input $(1^k, \mathbf{n})$, outputs a non-trivial divisor of $\mathbf{n}$ is negligible as a function of k.

Let $(\mathcal{N}_k)_{k \in K}$ be such a probability ensemble. For brevity, let $N = \bigcup_{k \in K} \operatorname{supp} \mathcal{N}_k$. Since all the elements of N are composite, we have $n \geq 4$ for all $n \in N$. Let $(\mathcal{Z}_n)_{n \in N}$ be a probability ensembe such that

- For any $n \in N$, $\operatorname{supp} \mathcal{Z}_n$ is a set of integers which are coprime to n
- $\sup_{n \in N} \Delta(\nu_n(\mathcal{Z}_n), \mathcal{U}(\mathbf{Z}_n^*)) \leq 1/2$
- The probability ensemble $(\mathcal{Z}_{\operatorname{bin}^{-1} u})_{u \in \operatorname{bin} N}$ is polynomial-time samplable

(such a probability ensemble exists by Lemma 1).

We start with the pseudo-free family of computational groups defined in Theorem 2. Namely, let $D = \{1^k, k \in K\}$ and M be a set of integers such that $1 \in M$ and $-m \in M$ whenever $m \in M$. For all $k \in K$, let $\mathcal{D}_k$ be the probability distribution of a random variable which takes the value 1^k with probability 1. Also, for each $1^k \in D$, let $G_{1^k} = F_{2^k}$, and ρ_{1^k} be the map from

$$\{((i_1, m_1), \ldots, (i_n, m_n)), i_j \in \{1, \ldots, 2^k\}, m_j \in M\}$$

onto F_{2^k} defined by $\rho_{1^k}((i_1, m_1), \ldots, (i_n, m_n)) = a_{i_1}^{m_1} \ldots a_{i_n}^{m_n}$. Finally, let $\mathcal{R}_{1^k}$ be the distribution of the random variable $((\mathbf{i}, 1))$, where $\mathbf{i} \leftarrow \mathcal{U}(\{1, \ldots, 2^k\})$. We recall that for all $i \in \{1, \ldots, 2^k\}$, $((i, 1))$ is a representation of the generator a_i of F_{2^k}. By Theorem 2, we obtain a pseudo-free family of computational groups.

For all $k \in K$, we denote by $\mathcal{E}_k$ the probability distribution of the random variable $(\mathbf{n}, (\mathbf{z}_1, \ldots, \mathbf{z}_k))$, where $\mathbf{n} \leftarrow \mathcal{N}_k$ and $\mathbf{z}_1, \ldots, \mathbf{z}_k \leftarrow \mathcal{Z}_n$. We also denote the support of $\mathcal{E}_{1^k}$ by E_{1^k}. Thus, the elements of E_{1^k} are tuples of the form $(n, (z_1, \ldots, z_k))$, where $z_1, \ldots, z_k$ are relatively prime with n.

For an integer $c \geq 2$, define the following matrices of $\mathrm{SL}_2(\mathbf{Z})$:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & c \\ 0 & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 \\ c & 1 \end{pmatrix}, \quad P_i = Q^{-i} P Q^i \text{ for all } i \in \mathbf{Z}.$$

Then P and Q freely generate a free subgroup of $\mathrm{SL}_2(\mathbf{Z})$, and so the set $\{P_i, i \in \mathbf{Z}\}$ also freely generates a free subgroup of $\mathrm{SL}_2(\mathbf{Z})$. Thus we can identify the free group F_{2^k} with the subgroup of $\mathrm{SL}_2(\mathbf{Z})$ freely generated by $\{P_i, i \in \{1, \ldots, 2^k\}\}$ by extending the assignment $a_i \mapsto P^i$ to an isomorphism. We denote this isomorphism by γ_{1^k}, and it is easy to see that we have

$$\gamma_{1^k}(\rho_{1^k}((i_1, m_1), \ldots, (i_n, m_n))) = Q^{-i_1} P^{m_1} Q^{i_1 - i_2} P^{m_2} \ldots Q^{i_{n-1} - i_n} P^{m_n} Q^{i_n}$$

where

$$P^m = \begin{pmatrix} 1 & cm \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad Q^j = \begin{pmatrix} 1 & 0 \\ cj & 1 \end{pmatrix}$$

for all integers m and j.

For a tuple $e = (n, (z_1, \ldots, z_k)) \in E_{1^k}$, we denote by $\mu(e)$ the least common multiple of the orders of $z_1 + n\mathbf{Z}, \ldots, z_k + n\mathbf{Z}$ in $\mathbf{Z}_n^*$. Finally, we let $H_{1^k,e}$ be the kernel of the homomorphism of F_{2^k} (viewed as a subset of $\mathrm{SL}_2(\mathbf{Z})$) into $\mathrm{SL}_2(\mathbf{Z}_{\mu(e)})$ induced by the projection homomorphism $\nu_{\mu(e)}$. Then $H_{1^k,e}$ is the set of all $g \in F_{2^k}$ such that all entries of the matrix $\gamma_{1^k}(g) - I$ are multiples of $\mu(e)$. Denote by $\mathcal{D}'_k$ the probability distribution of the random variable $(1^k, \mathbf{e})$ where $\mathbf{e} \leftarrow \mathcal{E}_{1^k}$, and for all $e \in E_{1^k}$, define the mapping $\rho'_{1^k,e} : \operatorname{dom} \rho_{1^k} \to F_{2^k}/H_{1^k,e}$ by $\rho'_{1^k,e}(r) = \rho_{1^k}(r) H_{1^k,e}$. We claim that the objects defined in this section satisfy the conditions of Theorem 2, which implies:

Theorem 3. *With the above notations, the family $((F_{2^k}/H_{1^k,e}, \rho'_{1^k,e}, \mathcal{R}_{1^k}))_{1^k \in D, e \in E_{1^k}}$ is a pseudo-free family of computational groups in the variety of all groups with respect to $(\mathcal{D}'_k)_{k \in K}$ and σ.*

Proof. $((F_{2^k}, \rho_{1^k}, \mathcal{R}_{1^k}))_{1^k \in D}$ is a pseudo-free family of computational groups by Theorem 1, so we only need to check that the conditions of Theorem 2 hold.

It is easy to see that given (a representation of) a word $w(a; x) \in F_{m,\ell}$ and (representations of) elements $f_1, \ldots, f_m, g_1, \ldots, g_\ell \in F_{2^k}$, according to the maps σ and ρ_{1^k} defined earlier, we can compute a representation of the element $w(f_1, \ldots, f_m; g_1, \ldots, g_\ell) \in F_{2^k}$ in deterministic polynomial time, so condition (i) holds.

For condition (ii), let $k \in K$, $e = (n, (z_1, \ldots, z_k)) \in E_{1^k}$, and $g \in F_{2^k}$. Recall that $H_{1^k,e}$ is the kernel of the homomorphism of F_{2^k} (viewed as a subset of $\mathrm{SL}_2(\mathbf{Z})$ by means of the isomorphism γ_{1^k}) into $\mathrm{SL}_2(\mathbf{Z}_{\mu(e)})$ induced by the projection homomorphism ν_e. It is then easy to see that $g \in H_{1^k,e}$ if and only if, letting $\gamma_{1^k}(g) = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, we have

$$\begin{pmatrix} (z_i + n\mathbf{Z})^a & (z_i + n\mathbf{Z})^b \\ (z_i + z\mathbf{Z})^c & (z_i + n\mathbf{Z})^d \end{pmatrix} = \begin{pmatrix} z_i + n\mathbf{Z} & 1 \\ 1 & z_i + n\mathbf{Z} \end{pmatrix}$$

for all $i = 1, \ldots, k$. Since this can be checked in deterministic polynomial time, condition (ii) holds.

For condition (iii), let A be a polynomial-time algorithm, which we will run on input $(1^k, 1^k, \mathbf{e})$. We wish to show that it outputs a non-identity element of $H_{1^k,\mathbf{e}}$ with negligible probability. We will use A to construct a probabilistic polynomial-time algorithm which, on input $(1^k, \mathbf{n})$, outputs a non-trivial divisor of $\mathbf{n}$. By assumption, this happens with negligible probability. Let then B be an algorithm which proceeds on input $(1^k, n)$, for all $k \in K$ and $n \in \operatorname{supp}\mathcal{N}_k$, as follows:

1. If n is even, output 2 and stop.
2. If n is a perfect power, compute integers $a, b \geq 2$ such that $n = a^b$, output a and stop. This can be done in deterministic polynomial time by Lemma 2.
3. Choose $z_1, \ldots, z_k \leftarrow \mathcal{Z}_n$, and let $e = (n, (z_1, \ldots, z_k))$.
4. Run A on input $(1^k, 1^k, e)$, and assume that A outputs a non-identity element of $H_{1^k,e}$. If it does not, output 1 and stop.
5. Let h be the output of A. Choose a non-zero entry s of $\gamma_{1^k}(h) - I$ (this is possible since $h \neq 1$ and γ_{1^k} is an isomorphism, so $h \neq 1$ implies $\gamma_{1^k}(h) \neq I$). Note also that the order of $z_i + n\mathbf{Z}$ in $\mathbf{Z}_n^*$ divides s, for all $i \in \{1, \ldots, k\}$. Let $s = 2^t s'$, with s' odd.
6. For every $i \in \{1, \ldots, k\}$ and $j \in \{0, \ldots, t\}$, compute an element $y_{i,j} \in (z_i + n\mathbf{Z})^{2^j s'}$. If there exist $i \in \{1, \ldots, k\}$ and $j \in \{0, \ldots, t-1\}$ such that $y_{i,j} \not\equiv \pm 1 \pmod{n}$ and $y_{i,j+1} \equiv 1 \pmod{n}$, output $\gcd(y_{i,j} - 1, n)$ and stop. Otherwise, output 1 and stop.

For brevity, let S be the set of all odd integers $n \geq 3$ which are not perfect powers. Also, for any $n \in \mathbf{N}^*$, let T_n be the set of all $u \in \mathbf{Z}_n^*$ such that u has even order and $-1 + n\mathbf{Z} \notin \langle u \rangle$ (note that this is the negative of the statement in Lemma 4). We make the following claim:

Claim 1. *We run the algorithm B on input $(1^k, n)$, for any $k \in K$ and $n \in \operatorname{supp}\mathcal{N}_k$. If the following three conditions hold, then B outputs a non-trivial divisor of n.*

- *$n \in S$ (*i.e.*, the algorithm does not terminate in steps 1-2).*
- *The algorithm does not terminate in step 4.*
- *There exists $i \in \{1, \ldots, k\}$ such that $z_i + n\mathbf{Z} \in T_n$.*

Proof of Claim 1. Let $i = \{1, \ldots, k\}$ be such that $z_i + n\mathbf{Z} \in T_n$. Since the order of $z_i + n\mathbf{Z}$ is even and since s' is odd, we have $y_{i,0} + n\mathbf{Z} = (z_i + n\mathbf{Z})^{s'} \neq 1 + n\mathbf{Z}$. Also, since the order of $z_i + n\mathbf{Z}$ divides s, we have $(y_{i,0} + n\mathbf{Z})^{2^t} = (z_i + n\mathbf{Z})^s = 1 + n\mathbf{Z}$. This means that there exists a unique $j \in \{0, \ldots, t-1\}$ such that $y_{i,j} \equiv y_{i,0}^{2^j} \not\equiv 1 \pmod{n}$ but $y_{i,j+1} \equiv y_{i,0}^{2^{j+1}} \equiv 1 \pmod{n}$. Moreover, since $z_i + n\mathbf{Z} \in T_n$ and since $y_{i,j} + n\mathbf{Z} = (z_i + n\mathbf{Z})^{2^j s'} \in \langle z_i + n\mathbf{Z} \rangle$, it is clear that $y_{i,j} + n\mathbf{Z} \not\equiv -1 \pmod{n}$. Thus the condition of step 6 of the algorithm holds, and the claim follows by Lemma 3. □

Let $\mathbf{e} = (\mathbf{n}, (\mathbf{z}_1, \ldots, \mathbf{z}_k))$, where $\mathbf{n} \leftarrow \mathcal{N}_k$ and $\mathbf{z}_1, \ldots, \mathbf{z}_k \leftarrow \mathcal{Z}_n$. We denote by A_k the event that $A(1^k, 1^k, \mathbf{e}) = [h]_{\rho^{1^k}}$ for some $h \in H_{1^k,e} \setminus \{1\}$. In other words, that $A(1^k, 1^k, \mathbf{e})$ outputs a representation of a non-identity element of $H_{1^k,e}$. It is then clear that

$$P(\mathbf{n} \notin S, \mathsf{A}_k) \leq P(\mathbf{n} \notin S) \leq P(B(1^k, \mathbf{n}) \text{ is a non-trivial divisor of } \mathbf{n}),$$

and so $P(\mathbf{n} \notin S, \mathsf{A}_k)$ is negligible.

Further, if we denote by B_k the event that there exists some $i \in \{1, \ldots, k\}$ such that $\mathbf{z}_i + \mathbf{n}\mathbf{Z} \in T_{\mathbf{n}}$, the above claim implies

$$P(\mathbf{n} \in S, \mathsf{A}_k, \mathsf{B}_k) \leq P(B(1^k, \mathbf{n}) \text{ is a non-trivial divisor of } \mathbf{n}),$$

and so this probability is also negligible.

Finally, suppose $n \in S \cap N$, and let $\tau(n)$ be the number of prime divisors of n. Since n is composite and not a perfect power, we have $\tau(n) \geq 2$. Let $\mathbf{u} \leftarrow \mathcal{U}(\mathbf{Z}_n^*)$, Lemma 4 asserts that $P(\mathbf{u} \notin T_n) \leq 1/2^{\tau(n)-1}$. Let now $\mathbf{g} \leftarrow \mathcal{Z}_n$ and $q = \sup_{\ell \in N} \Delta(\nu_\ell(\mathcal{Z}_\ell), \mathcal{U}(\mathbf{Z}_\ell^*))$. Then

$$P(\mathbf{g} + n\mathbf{Z} \notin T_n) \leq P(\mathbf{u} \notin T_n) + \Delta(\nu_n(\mathcal{Z}_n), \mathcal{U}(\mathbf{Z}_n^*)) \leq \frac{1}{2} + q.$$

Now, if we let B'_k be the complementary event of B_k, namely, that for all $i \in \{1, \ldots, k\}$ we have $\mathbf{z}_i + n\mathbf{Z} \notin T_n$, we obtain

$$P(\mathbf{n} \in S, \mathsf{A}_k, \mathsf{B}'_k) \leq P(\mathbf{n} \in S, \mathsf{B}'_k) \leq P(\mathbf{n} \in S) \left(\frac{1}{2} + q\right)^k.$$

Finally,

$$P(\mathsf{A}_k) = P(\mathbf{n} \notin S, \mathsf{A}_k) + P(\mathbf{n} \in S, \mathsf{A}_k, \mathsf{B}_k) + P(\mathbf{n} \in S, \mathsf{A}_k, \mathsf{B}'_k),$$

and since we have seen that the three probabilities on the right-hand side are negligible, this completes the proof that $P(\mathsf{A}_k)$ is negligible. □

Bibliography

[1] M. Anokhin, *Constructing a Pseudo-Free Family of Computational Groups under the General Integer Factoring Intractability Assumption. Electronic Colloquium on Computational Complexity*, 2012.

[2] M. Dietzfelbinger, *Primality Testing in Polynomial Time. Lecture Notes in Computer Science*, vol. 3000, Springer, 2004.

[3] S. Hohenberger, *The cryptographic impact of groups with infeasible inversion.* Master's thesis, EECS Dept., MIT, 2003.

[4] M. P. Jhanwar and R. Barua, *Sampling From Signed Quadratic Residues: RSA Group is Pseudofree.* Proceedings of INDOCRYPT 2009, *Lecture Notes in Computer Science*, vol. 5922, pp. 233-247, Springer, 2009.

[5] D. Micciancio, *The RSA Group is Pseudo-Free. Journal of Cryptology*, vol. 23, pp. 169-186, Springer, 2010.

[6] R. Rivest, *On the Notion of Pseudo-Free Groups.* Proceedings of TCC 2004, *Lecture Notes in Computer Science*, vol. 2951, pp. 505-521, Springer, 2004.

[7] S. Roman, *Fundamentals of Group Theory.* Birkhäuser, 2012.

[8] V. Shoup, *A Computational Introduction to Number Theory and Algebra*, second edition. Cambridge University Press, 2009. Also available online at `http://www.shoup.net/ntb/`.