On the Notion of Pseudo-Free Groups

Ronald L. Rivest

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
rivest@mit.edu

Abstract. We explore the notion of a *pseudo-free group*, first introduced by Hohenberger [Hoh03], and provide an alternative stronger definition. We show that if $\mathbf{Z}_n^*$ is a pseudo-free abelian group (as we conjecture), then $\mathbf{Z}_n^*$ also satisfies the Strong RSA Assumption [FO97,CS00,BP97]. Being a "pseudo-free abelian group" may be the strongest natural cryptographic assumption one can make about a group such as $\mathbf{Z}_n^*$. More generally, we show that a pseudo-free group satisfies several standard cryptographic assumptions, such as the difficulty of computing discrete logarithms.

1 Introduction

Cryptographic schemes often work with finite groups in such a way that the security of the scheme depends upon an explicit complexity-theoretic assumption about computational problems in that group.

For example, the RSA public-key cryptosystem [RSA78] works with the multiplicative group $\mathbf{Z}_n^*$, where n is the product of two large primes. The security of RSA encryption depends upon the "RSA Assumption."

> *RSA Assumption*: It is computationally infeasible for a probabilistic polynomial-time adversary, given an integer n that is the product of two sufficiently large randomly chosen primes, an integer $e > 1$ that is relatively prime to $\phi(n)$, and an element a chosen randomly from $\mathbf{Z}_n^*$, to compute the $x \in \mathbf{Z}_n^*$ such that
>
> $$x^e = a \pmod{n}$$
>
> with non-negligible probability.[1]

[0] © IACR, Proceedings TCC 2004

[1] A function $f(k)$ is considered to be a negligible function of k if for all constants $c > 0$ and all sufficiently large k we have that $|f(k)| < 1/k^c$. In the RSA Assumption, the phrase "non-negligible probability" is interpreted to mean a non-negligible function of $\log(n)$.

Similarly, the Cramer-Shoup cryptosystem and signature scheme [CS98,CS99] depend upon the "Strong RSA Assumption," [FO97,BP97]. which allows the adversary himself to choose an exponent $e > 1$.

> *Strong RSA Assumption*: It is infeasible for a probabilistic polynomial-time adversary, given an integer n that is the product of two sufficiently large randomly chosen primes, and an element a chosen randomly from $\mathbf{Z}_n^*$, to compute an $x \in \mathbf{Z}_n^*$ and an integer $e > 1$ such that
>
> $$x^e = a \pmod{n}$$
>
> with non-negligible probability.

Assuming that $\mathbf{Z}_n^*$ is *pseudo-free* takes this progression one step further: the adversary may choose whatever equation he wishes and try to solve it, as long as the equation is "nontrivial"—unsatisfiable in the free group, with appropriate care for some details. The pseudo-free assumption is that the adversary will succeed with at most negligible probability. The assumption of pseudo-freeness may be made for any arbitrary finite group, such as an elliptic curve group or even a nonabelian group. We might call the assumption that $\mathbf{Z}_n^*$ is pseudo-free the *Super-Strong RSA Assumption.*

We explore the assumption that a group is *pseudo-free* or, more specifically, *pseudo-free abelian*, and show how it implies some of these other standard assumptions. Assuming that a finite group is pseudo-free thus appears to be quite a strong assumption.

Why formulate and study such a strong assumption? Doesn't this go against the traditional style of making only the minimal complexity-theoretic assumptions necessary for a cryptographic scheme or protocol? Perhaps, but we provide the following motivation and justifications.

- It seems quite plausible that $\mathbf{Z}_n^*$ (for n the product of two sufficiently large randomly chosen primes) is in fact pseudo-free.
- Making stronger assumptions may make proofs easier (this is especially useful for pedagogic purposes).
- It may turn out that the pseudo-freeness is not a "stronger" assumption after all—it may be implied by simpler assumptions, perhaps more standard ones.
- Reasoning in a free group can be quite simple and intuitive, so assuming pseudo-freeness allows one to capture "natural" security proofs in a plausible framework. (This was Hohenberger's [Hoh03] motivation.)

Section 2 provides some mathematical background, and then Section 3 develops the definition of a pseudo-free group. Section 4 studies some of the implications of assuming that a group is pseudo-free. Section 5 considers some variations and generalizations of the basic definition, and then Section 6 discusses further issues related to the notion of a pseudo-free group. Finally, Section 7 provides some conclusions and lists some open problems.

2 Mathematical Background

2.1 Mathematical Groups

We first restate the definition of a mathematical group.

Definition 1. *A* group $G = (S, \circ)$ *consists of a set* S *of elements, and a binary operator* $\circ$ *defined on* S*, such that:*

Closure: *For all elements* $x, y \in S$*, we have* $x \circ y \in S$.
Identity: *There is an element* $1 \in S$ *such that for all elements* $x \in S$*,* $x \circ 1 = 1 \circ x = x$.
Associativity: *For all elements* $x, y, z \in S$*,* $x \circ (y \circ z) = (x \circ y) \circ z$.
Inverses: *For every element* $x \in S$*, there is an element* $y \in S$ *such that* $x \circ y = y \circ x = 1$.

We use multiplicative notation: ab means $a \circ b$. The inverse of x is denoted x^{-1}. We let G also denote the set S. A group G is finite iff $|S|$ is finite. A group G is *abelian* if $\circ$ is commutative: for all $x, y \in G$, $xy = yx$. We use the usual exponent notation: a^e is the word $aaa \ldots a$ of length e, and a^{-e} is the corresponding inverse word $a^{-1}a^{-1} \ldots a^{-1}$ of length e.

2.2 Computational Groups

A mathematical group G has some representation $[G]$ when used in cryptography. We call such a representation $[G]$ a *computational group* implementing an underlying *mathematical group*. Many computational groups may implement the same mathematical group.

In a computational group $[G]$, each element $x \in G$ has one or more representations as a finite-length bit string $[x]$. We often omit brackets, understanding that each element has such representation(s). When G is finite, it is convenient to assume that there is a common bit-length N such that any representation of any element of G requires exactly N bits.

A computational group provides efficient (polynomial-time) algorithms for all of the following operations: [2]

Composition: Given (representations of) group elements x and y, compute (a representation of) $x \circ y$.

Identity: Compute (a representation of) the identity element 1.

Inverses: Given (a representation of) an element x, compute (a representation of) x^{-1}.

Equality Testing: Given (representations of) any two elements $x, y \in G$, determine if $x = y$.

Sampling: (Only if G is finite.) Return (a representation of) an element chosen uniformly at random from G, or in a manner that is indistinguishable from uniformly at random to a probabilistic polynomial-time (PPT) adversary. We denote such a procedure as $x \in_R G$.

As a running example: given n, the product of two large primes, anyone, including an adversary, can efficiently do all the group operations in $\mathbf{Z}_n^*$, using the usual representation of elements as residues modulo n.

2.3 Black Box Groups

The parties in a cryptographic protocol may access the group in a *black-box* manner, a notion introduced by Babai and Szemerédi [BS84] (see also Babai [Bab97], and see Boneh and Lipton [BL96] for extension of the black-box notion to fields).

Under the black-box assumption, each element of the computational group is a bit string of some common length N, and "black-box" subroutines are available for the group operations. [3]

The black-box assumption is that group operations may only be performed using the supplied implementations. Furthermore, the representation of group elements is "opaque": operations on them other than through the black-box routines are forbidden. [4]

---

[2] Hohenberger [Hoh03] studies a variant where inversion is not efficiently computable, at least by the adversary, and relates such groups to transitive signatures schemes.

[3] For Babai [Bab97], these operations include all but sampling, as he studies the implementation of the sampling procedure itself.

[4] In some applications side information such as the size or structure of the underlying group, such as the fact that the group is cyclic, is known, even though the group's representation is otherwise "black-box;" we don't consider such side information here.

It is natural to ask if there are black-box algorithms for various group-theoretic problems. The black-box assumption is reasonable for algorithm design; it amounts to a convention or a self-imposed restriction on what operations may be performed. To find an efficient algorithm under the black-box assumption is then a satisfying result; no unusual "tricks" are required.

For example, Tonelli and Shanks [BS96, Section 7.1] [Coh93, Section 1.5.1] give a probabilistic black-box algorithm for computing square roots in $\mathbf{Z}_p^*$; it finds the black-box representation $[x]$ of a value x satisfying

$$x^2 = a \pmod{p}$$

given the black-box representation $[a]$ of a (assumed to be a quadratic residue), and also given the prime p. Other algorithms for this problem, such as Cipolla's [BS96, Section 7.2], violate the black-box assumption for $\mathbf{Z}_p^*$ by utilizing both field operations available in $\mathbf{F}_p$.

If no efficient black-box algorithm can be found for a problem, then the black-box assumption may be too restrictive. For example, Shoup [Sho97] proves lower bounds for discrete logarithms and other problems in the black-box group model.

However, we are studying here not algorithmic efficiency, but cryptographic security. A typical adversary may willfully violate any black-box assumption: he may examine the bits of any representation or examine the code implementing any group operation.

Consider our running example: $\mathbf{Z}_n^*$. Here an adversary is given n, and code for composition (i.e., for multiplication modulo n). Nothing prevents him from examining this code or the bit-level representations of elements, or from using methods such as "index-calculus methods" [SS98] not allowed under a black-box assumption.

Therefore, we do not make black-box assumptions. [5] We assume that an adversary may use any available information and may use methods that depend upon representation or implementation details. The adversary has "non-black-box" access to the group implementation. Whether a group is pseudo-free may then depend on the details of its representation as a computational group; one should properly speak of whether a computational group is pseudo-free or not. In any case, for our purposes it will be relevant that an equation is satisifiable in a mathematical group if and only if it is satisfiable in any computational group representing it.

[5] One could easily develop a theory of black-box pseudo-free groups.

2.4 Free Groups

Free groups are infinite groups derivable from a given set of generators that have no non-trivial relationships.

Free groups are defined formally as follows. (See also Gutiérrez [Gut00], for example.) Let $A = \{a_1, a_2, \ldots, a_k\}$ be a nonempty set of distinct symbols, which are the *generators* of a free group. For each such symbol a_i, let a_i^{-1} be a new symbol representing the inverse of a_i. Let A^{-1} denote the set $\{a_i^{-1} \mid a_i \in A\}$, and let $A^{\pm 1}$ denote $A \cup A^{-1}$; $A^{\pm 1}$ is the *set of symbols* for the free group with set A of generators.

We let $F(A)$ denote the free group defined by the set A of generators. We may equivalently write $F(a_1, a_2, \ldots, a_k)$ when $A = \{a_1, a_2, \ldots, a_k\}$. Elements of this free group may be represented as words (sequences of symbols of this free group). As an example, the word

$$a_1 a_2^{-1} a_2 a_1^{-1} a_3^{-1} a_2$$

represents an element of $F(a_1, a_2, a_3, a_4)$.

A word may be simplified, or reduced, by repeatedly eliminating any two adjacent inverse symbols; the resulting word is equivalent to the original. Thus, the word in the above example is equivalent to $a_3^{-1} a_2$. A word that can not be reduced further is *reduced* or *in canonical form*.

The elements of a free group are thus words in canonical form. One could alternatively define the elements to be equivalence classes of words.

The operation $\circ$ for a free group is concatentation followed by simplification. For example, $a_1 a_2 \circ a_2^{-1} a_3 = a_1 a_2 a_2^{-1} a_3 = a_1 a_3$.

The identity for a free group is the empty word ϵ. Two words represent the same element of a free group if their reduced forms are the same. The inverse of a word is just the reverse of the word, with each symbol replaced by its inverse. The operator $\circ$ is closed and associative—for a proof see, for example, Lyndon and Schupp [LS77, Chapter I].

A free group on at least one generator is an infinite group, since there are an infinite number of distinct words in canonical form (e.g. $\{a^k\}$).

Since a free group is infinite, it is not possible to even approximately implement uniform sampling. However, it is easy to construct a computational group that implements a free group on a countable set of generators except for the uniform sampling requirement.

We note that if $A \subseteq B$, then $F(A)$ is a subgroup of $F(B)$.

2.5 Free Abelian Groups

A free abelian group $FA(a_1, a_2, \ldots, a_k)$ is defined similarly to ordinary free groups, except that the group is abelian. Thus, for any pair of symbols

a and b, we may replace the sequence ab by the sequence ba and preserve equivalence.

Commutativity enables one to define the canonical form for a word in $FA(a_1, a_2, ..., a_l)$ to be a word of the form:

$$a_1^{e_1} a_2^{e_2} \dots a_l^{e_l}$$

for some integers $e_1, e_2, \dots, e_l$. It is well known that $FA(a_1, a_2, \dots, a_l)$ is isomorphic to the l-fold direct sum $\mathbf{Z} \oplus \mathbf{Z} \oplus \cdots \oplus \mathbf{Z}$. We could represent an element $a_1^{e_1} a_2^{e_2} \dots a_k^{e_l}$ of $FA(a_1, a_2, \dots, a_l)$ by the vector $(e_1, e_2, \dots, e_l)$, and implement $\circ$ with vector addition.

3 Pseudo-Free Groups

A cryptographic scheme may utilize a particular mathematical group G; all parties have access to a computational group $[G]$ representing G.

Intuitively, a group is *pseudo-free* if it is indistinguishable from a free group. A free group has no surprising or anomalous identities; the only truths are those implied by the axioms of group theory.

Thus, informally, we say that a finite group G is pseudo-free if a probabilistic polynomial-time adversary can not efficiently produce an equation E and a solution to E in G where E has no solution in the "corresponding free group." Of course, we need to define what we mean by "corresponding free group."

Assuming that a finite group such as $\mathbf{Z}_n^*$ is pseudo-free is thus a complexity-theoretic assumption, similar to but stronger than the RSA Assumption or the Strong RSA Assumption.

This assumption turns out to be very strong, as it implies several standard cryptographic assumptions (at least for $G = \mathbf{Z}_n^*$). Nonetheless, it seems a plausible assumption in some cases, and it may be useful for new applications. In any case, we find its formulation and elaboration interesting.

For example, in a free group (abelian or not), there is no solution to

$$x^2 = a \tag{1}$$

where x is a variable ranging over group elements, and a is a generator of the free group, since for any value of x the reduced form of x^2 has even length. However, the corresponding equation in $\mathbf{Z}_n^*$,

$$x^2 = a \pmod{n} , \tag{2}$$

has a solution if a is a square in $\mathbf{Z}_n^*$. A solution to such a corresponding equation "proves" that $\mathbf{Z}_n^*$ is different than the corresponding free group.

The adversary may not claim that G is distinguishable from a free group merely because G is obviously finite, for example, because the elements of G all have N-bit representations. We insist on a different kind of proof: the adversary must provide a solution to an equation in G whose "corresponding equation" in a free group has no solution.

3.1 Equations in Free Groups

Let H denote a free group, such as $F(a_1, a_2, \ldots, a_l)$ or $FA(a_1, a_2, \ldots, a_l)$.

Let $x_1, x_2, \ldots, x_m$ denote variables that may take values in H.

An equation in H takes the form

$$w_1 = w_2$$

where w_1 and w_2 are words formed from the symbols of H and from the variables x_1, x_2, $\ldots$, x_m. One can always put such equations in a "canonical form" of the form $w = 1$ for some word w.

As an example, in $F(a_1, a_2)$ the equation

$$a_1 x_1 = x_2 a_2^{-1} ,$$

has many solutions (x_1, x_2), such as (a_2^{-1}, a_1) or $(1, a_1 a_2)$.

Equations that have solutions in the free group are called *satisfiable*, others are called *unsatisfiable*.

Our definition of a pseudo-free group depends on being able to distinguish effectively between satisfiable and unsatisfiable equations in a free group.

Can one decide whether a given equation is satisfiable or not? Fortunately, one can. In 1982 Makanin [Mak82] showed that it is decidable whether or not an equation in the free group is satisfiable. More recently Gutiérrez [Gut00] has shown that this problem is decidable in PSPACE. For our use, these results are quite sufficient; the decision procedure need not be in polynomial-time.

When the free group is the abelian group $FA(a_1, a_2, \ldots, a_l)$ it is easy to determine whether a given equation is satisfiable: the equation can always be rewritten in the form:

$$x_1^{d_1} x_2^{d_2} \cdots x_m^{d_m} = a_1^{e_1} a_2^{e_2} \cdots a_l^{e_l}$$

for integers $d_1, d_2, \ldots, d_m, e_1, e_2, \ldots, e_l$. Such an equation is satisfiable iff for all i, $1 \leq i \leq l$, we have

$$\gcd(d_1, d_2, \ldots, d_m) \mid e_i \ . \tag{3}$$

One can prove that this statement holds for $l = 1$ and that such solutions can be combined for larger l.

An equation that is satisfiable in $F(A)$ is also satisfiable in $FA(A)$ (but not necessarily conversely). This is useful since it provides an easy way to prove that an equation is *unsatisfiable* in a free group: merely prove that it is unsatisfiable in the corresponding free abelian group.

3.2 The correspondence

Given an equation that is unsatisfiable in a free group $F(A)$, what counts as a "corresponding equation" in a given group G?

We have to be a little careful, since there are trivial cases to avoid. For example, the previously mentioned quadratic equation:

$$x^2 = a \ ,$$

which is unsatisfiable in $F(a)$, may have "trivial" solutions in $\mathbf{Z}_n^*$, depending on how the element in $\mathbf{Z}_n^*$ corresponding to the generator a of the free group is selected. For example, if the adversary is allowed to specify that $a = 4$, then there is clearly the trivial solution $x = 2$.

We resolve this issue (following Hohenberger's thesis [Hoh03]) by requiring that when making the correspondence between interpreting the equation in the free group and interpreting it in G, *each of the generators a_i must correspond to an independently generated random element of G.*

The adversary thus has no control over the choice of elements in G that are to correspond to the generators in the free group.

Thus, for example, the adversary must take the square root of a randomly chosen element $a \in \mathbf{Z}_n^*$ in order to demonstrate an acceptable solution to the above equation, when G is the group $\mathbf{Z}_n^*$.

This requirement that generators in the free group correspond to randomly chosen elements of G fits naturally with common cryptographic usage where, for example, one party publishes randomly-chosen elements g and h such that finding the discrete logarithm of h base g is assumed to be hard. For the adversary, the randomly chosen elements g and h are the "generators" of the group he must attack.

Informally, an adversary succeeds in distinguishing G from a free group if he can produce:

- An equation E that is unsatisfiable in the free group, where this equation has variables $x_1, x_2, \ldots, x_m$ and generators $a_1, a_2, \ldots, a_l$.
- A sequence $\alpha_1, \ldots, \alpha_l$ of values produced as random samples from the group G, to use as values for the generators $a_1, a_2, \ldots, a_l$. (If the inverse symbols a_i^{-1} are used, then they are to be replaced by the inverses of the randomly chosen values.)
- Values for the variables $x_1, x_2, \ldots, x_m$ that satisfy the equation produced in G.

This definition allows the adversary to choose the equation himself, as long as the equation is unsatisfiable in the free group. This generalizes the situation for the Strong RSA assumption, where the adversary may choose the exponent e.

For efficiency in describing his equation, the adversary may use "exponential expressions," such as $a((ax)^{531}x^{17})$, (see [Gut00, Section 2.2.1]), or even the mathematically equivalent but potentially more compact notation of algebraic straight-line programs, as proposed in Hohenberger [Hoh03].

The adversary need not produce a proof that the equation is unsatisfiable in a free group, since this can be verified directly using Makanin's or Gutiérrez's algorithm. (One could alternatively require the adversary to produce an equation whose unsatisfiability can be verified in polynomial time, or to produce a polynomial-size proof of unsatisfiability; we do not study such a restriction here, since the impact of assuming pseudo-freeness is to support the infeasibility for an adversary to solve the equation, not to support using the equation in a protocol.)

We make our definition more precise as follows.

Definition 2. *A family $\mathcal{G} = \{G_k : k \geq 0\}$ of finite computational groups is pseudo-free if:*

- *All operations in G_k can be performed in time polynomial in k.*
- *For every probabilistic polynomial-time adversary $\mathcal{A}$, for every polynomial $p(\cdot)$, if $\alpha_1, \alpha_2, \ldots, \alpha_{p(k)}$ are elements chosen uniformly and independently at random from G_k, then the probability*

$$Pr[\mathcal{A}(G_k, \alpha_1, \alpha_2, \ldots, \alpha_{p(k)}) = (E, \beta_1, \beta_2, \ldots, \beta_m)]$$

where $\mathcal{A}$ is given access to the routines implementing the group G_k as well as the elements $\alpha_1, \alpha_2, \ldots, \alpha_{p(k)}$, and where

$$E = E(x_1, x_2, \ldots, x_m; a_1, a_2, \ldots, a_{p(k)})$$

is an equation over the free group $F(a_1, a_2, \ldots, a_{p(k)})$ with variables $x_1, x_2, \ldots, x_m$ such that E is unsatisfiable in $F(a_1, a_2, \ldots, a_{p(k)})$

but $E(\beta_1, \beta_2, \ldots, \beta_m; \alpha_1, \alpha_2, \ldots, \alpha_{p(k)})$ *is true in* G_k*, is a negligible function of* k*.*

This definition refers to a family of computational groups, but one may apply it to a family of mathematical groups with the understanding that the groups are implemented in some standard way as computational groups. One may also wish to specify whether the adversary has black-box access or non-black-box access to the group.

If the groups G_k are abelian, then we may also say that $\mathcal{G}$ is *pseudo-free abelian*, although we prefer just saying that $\mathcal{G}$ is pseudo-free when, as in the case $\mathcal{G} = \{\mathbf{Z}_n^*\}$, the groups are obviously abelian.

4 Pseudo-freeness implies many other cryptographic assumptions

If G is pseudo-free, then several standard complexity-theoretic assumptions follow. We look at the six fundamental problems studied by Lipschutz and Miller [LI71], and then examine other standard cryptographic assumptions, such as Diffie-Hellman.

Lipschutz and Miller [LI71] consider six fundamental problems: the *order problem* [solving $a^e = 1$ for e], the *power problem* (aka the discrete logarithm problem) [solving $a^e = b$ for e], the *root problem* (aka the RSA problem) [solving $x^e = a$ for x], the *proper power problem* (aka the strong RSA problem) [solving $x^e = a$ for x and $e > 1$], the *generalized power problem* [solving $a^e = b^f$ for nonzero e, f], and the *intersection problem for cyclic subgroups* [solving $a^e = b^f \neq 1$ for e, f]. They show these problems are independent: for each pair of problems there is a group such that one problem is solvable (i.e. satisfiability of the relevant equation is decidable) while the other problem is unsolvable. These problems, while studied with respect to their decidability, are familiar ones for the cryptographer; we explore their satisfiability in the free group, and consequent implications for pseudo-free groups.

4.1 Order problem

The *order problem* in G is the following: given an element $a \in G$, to determine a positive integer e (if any exist) such that

$$a^e = 1 \ . \tag{4}$$

The least positive such value e is the *order* of the element a in the group G. In a free group all elements except the identity have infinite order, implying the following theorem.

Theorem 1. *In a pseudo-free group G, it is infeasible for an adversary to determine the order of a randomly chosen element a.*

4.2 Discrete logarithm problem

The *discrete logarithm problem* in G is: given elements a and b from G, to determine an integer e (if any exist) such that

$$a^e = b \; ; \tag{5}$$

the value e is a "discrete logarithm" of b, to the base a, in the group G.

This problem is often assumed to be hard, for specific groups G; in their classic paper [DH76b], for example, Diffie and Hellman assumed that this problem was hard in Z_p^* for large primes p. (See also [DH76a] for a slightly earlier usage.)

In $F(a, b)$ and $FA(a, b)$ equation (5) never holds, for any value of e. Since a and b are distinct generators, the two sides of the equation are variable-free constant expressions that can not be equal.

Theorem 2. *In a pseudo-free group, the discrete logarithm problem is infeasible for an adversary to solve, for randomly chosen values a and b.*

4.3 RSA assumption

In the free group $F(a)$ or $FA(a)$ the equation

$$x^e = a \tag{6}$$

has no solution, for any fixed value of $e > 1$. (It has no solution in $FA(a)$, by our previous discussion of the condition of equation (3).)

Theorem 3. *In a pseudo-free group, the RSA assumption holds.*

4.4 Strong RSA Assumption

The Strong RSA Assumption, defined earlier, was introduced by Barić and Pfitzmann [BP97] and also by Fijisaki and Okamoto [FO97].

The ability of an adversary to himself choose an exponent $e > 1$ does not affect the satisfiability of equation (6) in a free group.

Theorem 4. *In a pseudo-free group, the Strong RSA Assumption holds.*

Similar equations, such as

$$x^e = a^f \ ,$$

where the adversary is given a and must find x, e, and f such that $e > 1$ and $\gcd(e, f) = 1$, are also infeasible for the adversary to solve in pseudo-free groups; indeed this problem equivalent to solving the strong RSA problem since $\tilde{x}^e = a$ where $\tilde{x} = x^{f'} a^{e'}$ and $ee' + ff' = 1$ (see [CS99, Lemma 1]).

4.5 Generalized Power Problem

The generalized power problem is: given group elements a and b, to find nonzero integers e, f satisfying

$$a^e = b^f \ . \tag{7}$$

Theorem 5. *In a pseudo-free group, it is infeasible for an adversary to solve the generalized power problem.*

4.6 Intersection Problem for Cyclic Subgroups

The intersection problem for cyclic subgroups is: given group elements a and b. to find integers e, f such that

$$a^e = b^f \neq 1 \ . \tag{8}$$

Theorem 6. *In a pseudo-free group, it is infeasible for an adversary to solve the intersection problem for cyclic subgroups.*

4.7 Diffie-Hellman assumption

Interestingly, the (computational) Diffie-Hellman problem seems not to fit within our formalism. It is a very interesting open problem whether the Diffie-Hellman assumption is implied by pseudo-freeness.

The Computational Diffie-Hellman problem (CDH) is the following: given a value g, and two values

$$a = g^e \tag{9}$$
$$b = g^f \ , \tag{10}$$

for large randomly chosen integers e and f, to compute

$$x = g^{ef} \ . \tag{11}$$

The CDH assumption is that an adversary will have a negligible chance of computing x, given a and b. The natural way of trying to show that the CDH assumption is implied by pseudo-freeness is via equations (9)–(11), where e and f are integer-valued variables, and x is a group element variable (see section 5). However, this argument fails because an adversary who violates CDH to compute x need not be able to find e and f (this is DLP). There doesn't seem to be any equation in variable x alone (i.e., without e, f) available to verify that an adversary has correctly computed x. In other words, the decisional Diffie-Hellman problem doesn't seem to be solvable by verifying an appropriate set of equations involving the single unknown x.

5 Generalizations

In this section we discuss some variations and generalizations on the basic notion of pseudo-freeness.

5.1 Multiple equations

Mal'cev [Mal60] (see also [KM, Lemma 3 and Corollaries 2–3]) shows that for any finite set of equations in the free group, one can construct a single equation having exactly the same set of solutions. Thus, we may consider sets of simultaneous equations as equivalent to a single equation. The method is based on showing that the two equations $x = 1$, $y = 1$ are equivalent to the single equation $x^2ax^2a^{-1} = (ybyb^{-1})^2$.

For abelian groups, it is easy to determine if a set of equations is satisfiable; one may apply standard techniques for solving a set of simultaneous equations over the integers (see Artin [Art91, Section 12.4], for example).

These results allow us to permit the adversary to produce a set of simultaneous equations rather than just a single equation, without loss of generality.

5.2 Adversary must prove that equation is unsatisfiable in the free group

One could require that the adversary provide a polynomial-time checkable proof that the equation he produces is indeed unsatisfiable in the corresponding free group. However, this restriction seems somewhat pointless, since the reason for assuming pseudo-freeness anyway is to conclude that finding an equation together with its solution should be infeasible.

5.3 Generation of α's

Instead of providing random α's to the adversary directly, one could allow the adversary to produce them himself, as long as they are guaranteed to be "random" in some way.

For example, the adversary might be allowed to use a hash function with range G to derive the relevant α. If the hash function is pseudorandom, or can be modeled as a random oracle [BR93], then its output could be considered as an acceptable α for purposes.

Similarly, if the output of h is an integer, then we may be able to accept $g^{h(x)}$ as an acceptable element α from G for our purposes. The essential criterion for sampling is that the adversary should have no control over the element chosen, and it should be reasonable to model the element chosen as being independently chosen (approximately) uniformly at random from G.

The values α supplied might also be constrained to ensure that a solution in G exists; we don't pursue this variant further here.

5.4 Generalized exponential expressions

In the most general form of exponential expressions, the exponents may themselves be integer-valued variables. Consider for example, the equation $(ax)^e b = x^f$ in $F(a, b)$ where x is a variable ranging over group elements and e, f are integer-valued variables. This equation is satisifiable, for example, with $x = b$, $e = 0$, $f = 1$. It is an open problem how to decide if such equations, containing both element-valued variables and integer-valued exponent variables, are satisfiable—see Problem 3 in Section 7.

We may nonetheless allow an adversary to use these general exponential expressions, with variable exponents, because it is still possible to verify that the adversary has "done the impossible." The adversary produces an equation E with variable exponents, and also a solution that satisfies E. If E is unsatisfiable, then so is the equation E' obtained by substituting into E the exponent values supplied in the adversary's solution. One can then verify that E' is unsatisfiable using Makanin's algorithm.

Hohenberger uses straight-line programs in her definition of "equation" or "identity", a natural further generalization of the exponential expressions, which could also be allowed here.

5.5 Adaptive attacks and side information

It may be possible generalize the definition of pseudo-freeness here to handle adaptive attacks and other forms of "side information." How might the definition of pseudo-freeness change if side information, such as the order of the group, is known? Is there a reasonable way to do this? Similarly, how can the notion of pseudo-freeness be adapted to handle adaptive attacks, where the adversary may obtain a solution to an equation before having to provide a different solution (perhaps with new generators)?

6 Discussion

We compare our definition of a pseudo-free group with that given in Hohenberger's thesis. Her work is motivated by transitive signature schemes, and does introduce the critical correspondence between elements drawn from G at random and generators in the free group.

However, Hohenberger doesn't use variables, which are necessary for setting up equations and showing how pseudo-freeness implies other cryptographic assumptions, and she doesn't address the decidability of determining which equations are satisfiable in a free group. Also, her definition requires that an adversary have only "black-box" access to G.

7 Conclusions and Open Problems

We have taken the definition of pseudo-free group introduced by Hohenberger [Hoh03], strengthened it, and shown how it implies a number of other well-known cryptographic assumptions. While stronger than many previous cryptographic number-theoretic assumptions, pseudo-freeness seems fairly natural, worthy of study in its own right, and quite plausible for commonly used groups.

The study of pseudo-freeness yields some intriguing open problems and conjectures. We begin with our main conjecture.

Conjecture 1 (Super-Strong RSA Assumption). $\mathbf{Z}_n^*$ is pseudo-free.

The next open problem is to relate the Diffie-Hellman assumption to pseudo-freeness.

Conjecture 2 (Diffie-Hellman holds for Pseudo-free groups). In a pseudo-free group, both the computational and decisional Diffie-Hellman assumptions hold.

The following interesting problem, discussed briefly earlier, also appears to be open.

Conjecture 3. It is decidable whether a given equation (or set of equations) with constants is satisfiable over a free group, when the equation is written in exponential notation and may have integer-valued variables in the exponents.

Here is a (satisfiable) example of such an equation: $a((ab)^e y)^f b = x^2$ where x and y are variables (over the group), a and b are constants (group elements), and e and f are integer-valued variables. Some partial results are known [Lyn60,LI71,CE84]; the introduction to [CE84] gives a brief survey. This problem may also be open over semigroups.

Another open research direction is to explore ways of showing that a group G is not a free group, other than by demonstrating the solution to an equation that has no solution in a free group. For example, some statement of the elementary theory of free groups may be (say) false, but provably true in G. Kharlampovich and Myasnikov [KM98] have shown that the elementary theory of a free group is decidable, even if constants are allowed, a much more general result than determining whether a given equation is satisfiable in the free group.

The theory of pseudo-free groups might also be expanded to handle cases such as Z_p^*; this group is typically not pseudo-free, since the size of the group is presumably known in a typical implementation.

Finally, we note that we have only scratched the surface of the study of adaptive attacks against cryptographic schemes defined on pseudo-free groups; much work remains to be done here.

Acknowledgments

I'd like to thank Susan Hohenberger, Albert Meyer, Olga Kharlampovich, and an anonymous referee for helpful guidance and advice.

References

[Art91] Michael Artin. *Algebra*. Prentice Hall, 1991.

[Bab97] L. Babai. Randomization in group algorithms: conceptual questions. In L. Finkelstein and W. M. Kantor, editors, *Groups and Computation II. Proc. 1995 DIMACS Workshop*, volume 28 of *DIMACS Ser. in Discr. Math. and Theor. Comp. Sci.*, pages 1–16. AMS, 1997.

[BL96] Dan Boneh and Richard J. Lipton. Algorithms for black-box fields and their application to cryptography. In Neal Koblitz, editor, *Advances in Cryptology—CRYPTO '96*, pages 283–297. Springer-Verlag, 1996. Lecture Notes in Computer Science No. 1109.

[BP97] Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Proc. EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer-Verlag, 1997.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, 1993. ACM.

[BS84] L. Babai and E. Szemerédi. On the complexity of matrix group problems I. In *Proc. 25th IEEE FOCS*, pages 229–240, 1984.

[BS96] Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory; Volume I: Efficient Algorithms*. The MIT Press, 1996.

[CE84] Leo P. Comerford, Jr. and Charles C. Edmunds. Quadratic parametric equations over free groups. In K. I. Appel, J. G. Ratcliffe, and P. E. Schupp, editors, *Contributions to Group Theory*, volume 33 of *Contemporary Mathematics*, pages 159–196. AMS, 1984.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1993.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Proceedings Crypto '98*, pages 13–25. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.

[CS99] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. In *Proceedings 6th ACM Conference on Computer and Communications Security*, pages 46–52. ACM, Nov 1999.

[CS00] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, 2000.

[DH76a] W. Diffie and M. E. Hellman. Multiuser cryptographic techniques. In *Proc. AFIPS 1976 National Computer Conference*, pages 109–112, Montvale, N.J., 1976. AFIPS.

[DH76b] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, November 1976.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Jr. Burton S. Kaliski, editor, *Proc. CRYPTO '97*, volume 1294 of *LNCS*, pages 16–30. Springer-Verlag, 1997.

[Gut00] Claudio Gutiérrez. Satisfiability of equations in free groups is in PSPACE. In *Proc. 32nd ACM STOC*, pages 21–27. ACM Press, 2000.

[HMR03] Susan Hohenberger, David Molnar, and Ronald L. Rivest. Special signatures need special algebra, May 2003. Submitted.

[Hoh03] Susan Hohenberger. The cryptographic impact of groups with infeasible inversion. Master's thesis, EECS Dept., MIT, June 2003.

[KM] Olga Kharlampovich and Alexei Myasnikov. Implicit function theorem over free groups. Available at `www.math.mcgill.ca/olga/publications.html`.

[KM98] Olga Kharlampovich and Alexei Myasnikov. Tarski's problem about the elementary theory of free groups has a positive solution. *Electronic Research Announcements of the American Mathematical Society*, 4:101–108, December 14, 1998. S 1079-6762(98)00047-X.

[LI71] Seymour Lipschutz and Charles F. Miller III. Groups with certain solvable and unsolvable decision problems. *Communications on Pure and Applied Mathematics*, XXIV:7–15, 1971.

[LS77] Roger C. Lyndon and Paul E. Schupp. *Combinatorial Group Theory.* Springer, 1977.

[Lyn60] R. C. Lyndon. Equations in free groups. *Trans. Amer. Math. Soc.*, 96:445–457, 1960.

[Mak82] G. S. Makanin. Equations in a free group. *Izvestiya NA SSSR*, 46:1199–1273, 1982. English translation in Math USSR Izvestiya, 21 (1983), 483–546.

[Mal60] A. I. Mal'cev. On some correspondence between rings and groups. *Math. Sbornik*, 50:257–266, 1960.

[MW99] Ueli Maurer and Stefan Wolf. The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms. *SIAM Journal on Computing*, 28(5):1689–1721, 1999.

[MW00] Ueli Maurer and Stefan Wolf. The Diffie-Hellman protocol. *Designs, Codes, and Cryptography*, 19:147–171, 2000.

[Raz84] A. A. Razborov. On systems of equations in free groups. *Izvestiya AN SSSR*, 48:779–832 (In Russian), 1984. English translation in Math. USSR Izvestiya 25,1 (1985) 115–162.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Proc. Eurocrypt '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer-Verlag, May 1997.

[SS98] Joseph H. Silverman and Joe Suzuki. Elliptic curve discrete logarithms and the index calculus. In *Proc. Asiacrypt '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 110–125. Springer-Verlag, 1998.