USING THE SUBGROUP MEMBERSHIP SEARCH PROBLEM IN PUBLIC KEY CRYPTOGRAPHY

VLADIMIR SHPILRAIN AND GABRIEL ZAPATA

ABSTRACT. There are several public key protocols around that use the computational hardness of either the *conjugacy search problem* or the *word (search) problem* for nonabelian groups. In this paper, we describe a cryptosystem whose security is based on the computational hardness of the *subgroup membership (search) problem*: given a group G, a subgroup H generated by $h_1, \dots, h_k$, and an element $h \in H$, find an expression of h in terms of $h_1, \dots, h_k$.

It is also interesting to note that groups which we suggest to use as the platform, *free metabelian groups*, are *infinitely presented*, in contrast with groups typically used in public key cryptography. Nevertheless, these group have efficiently (and, in fact, very easily) solvable word problem.

1. Introduction

In search for a more efficient and/or secure alternative to established cryptographic protocols (such as RSA), several authors have come up with public key establishment protocols as well as with complete public key cryptosystems based on allegedly hard *search problems* from combinatorial (semi)group theory, including the conjugacy search problem [1, 16], the homomorphism search problem [12], [13], [25], the decomposition search problem [6, 16, 23].

In this paper, we describe a cryptosystem whose security is based on the computational hardness of the *subgroup membership (search) problem*:

Given a group G, a subgroup H generated by $h_1, \dots, h_k$, and an element $h \in H$, find an expression of h in terms of $h_1, \dots, h_k$.

It should be mentioned that, in fact, the Anshel-Anshel-Goldfeld protocol which may seem to rely on the computational hardness of the conjugacy search problem alone, actually relies (perhaps somewhat implicitly) on the hardness of the subgroup membership (search) problem as well, as explained in [23]. We also note that there is some similarity, at a philosophical level, between our cryptosystem and homomorphic public-key cryptosystems of [12] and [13]. However, there are also essential differences, as explained in our Section 2.

First we outline the ideas behind our cryptosystem. These ideas can be traced back to [20], where a similar approach was used in commutative situation, but the resulting cryptosystem was not accepted by the cryptographic community as secure [11]. We believe that employing a nonabelian group instead of a polynomial algebra as the

Research of the first author was partially supported by the NSF grant DMS-0405105.

platform does make a difference in both the efficiency and security components. We note that various nonabelian groups have been used as platforms in the above mentioned protocols; in particular, braid groups [1], [16], Grigorchuk groups [10], Thompson's group [23], polycyclic groups [8]. In this paper, we use groups of a different nature.

To explain our approach, we need some more background. Let F_n denote the free group of rank n. It is well known that any map on the generators of F_n into F_n extends to an endomorphism of F_n. It is also well known that free groups are not the only groups with this property.

Definition 1. Let F be a free group and let R be a normal subgroup of F. The factor group F/R is called *relatively free* if R is fully invariant, i.e., if $\alpha(R) \leq R$ for any endomorphism α of F. If $x_1, \dots, x_n$ are free generators of F, then $x_1R, \dots, x_nR$ are called relatively free generators of F/R.

We are going to denote relatively free generators of F/R simply by $x_1, \dots, x_n$ when there is no ambiguity. Let $\mathcal{F}_n$ denote a relatively free group of rank n, i.e., $\mathcal{F}_n = F_n/R$ for some fully invariant R. Then any map on its generators into $\mathcal{F}_n$ can be extended to an endomorphism of $\mathcal{F}_n$. Because of this property, any relatively free group $\mathcal{F}_n$ can be a candidate for the platform of our cryptosystem.

We also need to recall one basic fact about automorphisms of F_n. Let $X = \{x_1, \dots, x_n\}$ be a set of generators of F_n and consider the maps $\alpha_j, \beta_{jk} : X \to F_n$ given by

$$\alpha_j : x_i \longmapsto \begin{cases} x_i^{-1} & \text{if } i = j \\ x_i & \text{if } i \neq j \end{cases} \quad \text{and} \quad \beta_{jk} : x_i \longmapsto \begin{cases} x_i x_j & \text{if } i = k \\ x_i & \text{if } i \neq k\,, \end{cases}$$

where $1 \leq i, j, k \leq n$. Then all the α_j's and β_{jk}'s define automorphisms of F_n, which are called *Nielsen automorphisms*, and generate the whole automorphism group of F_n.

Nielsen automorphisms can be defined for any relatively free group $\mathcal{F}_n$ the same way. They generate a subgroup of the group $Aut(\mathcal{F}_n)$ of all automorphisms of $\mathcal{F}_n$; this subgroup is called the group of *tame* automorphisms. In some cases, it is equal to the whole $Aut(\mathcal{F}_n)$ (see e.g. [2]); in other cases, it is a proper subgroup of $Aut(\mathcal{F}_n)$ (see e.g. [5, 22]). For our purposes, it is important that groups of the form $F_n/[R, R]$ (where R is a normal subgroup of F_n, and $[R, R]$ its commutator subgroup) tend to have many non-tame automorphisms by a result of [22]; we reproduce it in our Section 3.

To conclude the introduction, we introduce some more notation. Let $\langle x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}\,;\, R\rangle$ be a presentation of a relatively free group $\mathcal{F}_{n+m}$. Let $\varphi \in Aut\,\mathcal{F}_{n+m}$ be an automorphism such that

$$\varphi : x_i \longmapsto y_i\,, \quad \text{where} \quad y_i = y_i(x_1, \dots, x_{n+m})\,.$$

This φ acts on $\mathcal{F}_{n+m}^{n+m}$, the direct product of $n+m$ copies of $\mathcal{F}_{n+m}$, in the natural way:

$$\varphi : (x_1, \dots, x_{n+m}) \longmapsto (y_1, \dots, y_{n+m})\,,$$

and, more generally,

$$\varphi : (u_1, \dots, u_{n+m}) \longmapsto (y_1(u_1, \dots, u_{n+m}), \dots, y_{n+m}(u_1, \dots, u_{n+m}))$$

for any $u_i = u_i(x_1, \dots, x_{n+m}) \in \mathcal{F}_{n+m}$.

Then, let $\hat{\varphi}$ be the restriction of φ to the subgroup of $\mathcal{F}_{n+m}^{n+m}$ that consists of all elements of the form $(u_1(x_1,\dots,x_n),\dots,u_n(x_1,\dots,x_n),1,\dots,1)$ (this subgroup is isomorphic to $\mathcal{F}_n^n$), so that

$$\hat{\varphi} : (u_1,\dots,u_n,1,\dots,1) \longmapsto (\hat{y_1}(u_1,\dots,u_n),\dots,\hat{y}_{n+m}(u_1,\dots,u_n)),$$

where $\hat{y_i}(x_1,\dots,x_n) = y_i(x_1,\dots,x_n,1,\dots,1)$.

We note that $\hat{\varphi}$ is a one-to-one map because the restriction of a one-to-one map to a subgroup is itself one-to-one. This property will be important to us in Section 2. At the same time, there is no visible way of recovering φ from $\hat{\varphi}$.

2. The protocol

Let $\mathcal{F}_{n+m}$ be a relatively free group. We discuss a specific choice of the platform group in the following Section 3; here we present our public key encryption/decryption protocol without specifying the platform group.

Key Generation: (1) Alice privately chooses an automorphism $\varphi \in Aut\,\mathcal{F}_{n+m}$ as a random product of Nielsen automorphisms and some easily invertible automorphisms of the type given in the theorem in the Introduction: $\varphi = \tau_1 \cdots \tau_k$ and computes the inverse $\varphi^{-1} = \tau_k^{-1} \cdots \tau_1^{-1}$. This φ^{-1} is Alice's private decryption key.

(2) Let $y_i = y_i(x_1,\dots,x_{n+m}) = \varphi(x_i)$. Alice publishes $\hat{y_i}(x_1,\dots,x_n) = y_i(x_1,\dots,x_n,1,\dots,1)$, $i = 1,...,n+m$. This collection of $\hat{y_i}$ is the public encryption key.

Encryption and Decryption: The information considered for encryption (i.e., Bob's private message) is an element w of the subgroup of $\mathcal{F}_{n+m}^{n+m}$ that consists of all elements of the form $(u_1(x_1,\dots,x_n),\dots,u_n(x_1,\dots,x_n),1,\dots,1)$. Thus,

$$w = (w_1(x_1,\dots,x_n),\dots,w_n(x_1,\dots,x_n))$$

(one can suppress the 1's in the end).

(1) Encryption is defined by

$$w \mapsto \hat{\varphi}(w) = (\hat{y_1}(w_1,\dots,w_n),\dots,\hat{y}_{n+m}(w_1,\dots,w_n)).$$

Bob then transmits $\hat{\varphi}(w)$ as a tuple of words in the alphabet X.

(2) Decryption is defined by $\hat{\varphi}(w) \mapsto [\varphi^{-1}(\hat{\varphi}(w))] \mid_{x_{n+1}=\dots=x_{n+m}=1} = w$. This w (more precisely, a *normal form* of w, see the end of Section 3) is Alice's and Bob's common secret key.

We note that the encryption of w here is *not* a homomorphic image (or a preimage) of w, in contrast to homomorphic public-key cryptosystems of [12] and [13].

The adversary can recover the plaintext w if he finds an expression for each $x_1,\dots,x_n$ in terms of $\hat{y_1},\dots,\hat{y}_{n+m}$, i.e., if he solves the membership search problem for the subgroup generated by $\hat{y_1},\dots,\hat{y}_{n+m}$ (this subgroup is actually $\mathcal{F}_n$). Then from

$$x_i = u_i(\hat{y_1}(x_1,\dots,x_n),...,\hat{y_n}(x_1,\dots,x_n))$$

he can get

$$w_i = u_i(\hat{y_1}(w_1, \dots, w_n), ..., \hat{y_n}(w_1, \dots, w_n)).$$

If however the adversary wants to completely break the cryptosystem, i.e., to get the private decryption key, then he has to find the automorphism φ (and its inverse) based on the restriction $\hat{\varphi}$. This problem looks unapproachable to us by any deterministic method other than trying out products of "elementary" automorphisms of $\mathcal{F}_{n+m}$ until the one with the right restriction is found. This method is computationally infeasible for large n, m.

We discuss specific values of the parameters of our cryptosystem in Section 5. Here we give a "toy example" just to illustrate how our protocol works. This example is not platform-specific because we only use Nielsen automorphisms as building blocks here.

Example 1. *Let $n = 2, m = 1$, and let β_{jk} be Nielsen automorphisms of $\mathcal{F}_3$ defined in the Introduction. Let $\varphi = \beta_{23}^2\beta_{12}\beta_{31}\beta_{32}$. Then $\varphi : x_1 \to x_1x_3x_2^2,\ x_2 \to x_2x_1x_3x_2^2,\ x_3 \to x_3x_2^2$. Thus, $y_1 = x_1x_3x_2^2,\ y_2 = x_2x_1x_3x_2^2,\ y_3 = x_3x_2^2$, and therefore $\hat{y_1} = x_1x_2^2,\ \hat{y_2} = x_2x_1x_2^2,\ \hat{y_3} = x_2^2$.*

Thus, Bob's private message $(u_1(x_1, x_2), u_2(x_1, x_2))$ is encrypted as $(u_1u_2^2,\ u_2u_1u_2^2,\ u_2^2) = (v_1, v_2, v_3)$.

In this simple example the adversary can easily obtain x_1, x_2 from the public $\hat{y_i}$ as follows: $x_1 = \hat{y_1}\hat{y_3}^{-1},\ x_2 = \hat{y_2}\hat{y_1}^{-1}$. Therefore, he can recover $u_1 = v_1v_3^{-1},\ u_2 = v_2v_1^{-1}$.

As a comment to this example, we note that in a free group, the subgroup membership search problem can be efficiently solved by Nielsen's method (see e.g. [17]). However, adding automorphisms $\alpha_{u,j}$ described in the Theorem in the next Section 3 makes a difference because it makes working in a free group pointless. We illustrate this by Example 2 in the next section.

3. Free metabelian groups

In this section, we suggest specific relatively free groups, called *free metabelian groups*, which can be used as platforms for the cryptosystem described in the previous section. Some terminology has to be introduced first.

A group G is called *abelian* (or commutative) if $[a, b] = 1$ for any $a, b \in G$, where $[a, b]$ is the notation for $a^{-1}b^{-1}ab$. This can be generalized in different ways. A group G is called *metabelian* if $[[x, y], [z, t]] = 1$ for any $x, y, z, t \in G$. A group G is called *nilpotent of class* $c \geq 1$ if $[y_1, y_2, \dots, y_{c+1}] = 1$ for any $y_1, y_2, \dots, y_{c+1} \in G$, where $[y_1, y_2, y_3] = [[y_1, y_2], y_3]$, etc.

The commutator subgroup of G is the group $G' = [G, G]$ generated by all commutators, i.e., by expressions of the form $[u, v] = u^{-1}v^{-1}uv$, where $u, v \in G$. Furthermore, we can define, by induction, the kth term of the *lower central series* of G: $\gamma_1(G) = G, \gamma_2(G) = [G, G],\ \gamma_k(G) = [\gamma_{k-1}G, G]$. Note that one has $\alpha([u, v]) = [\alpha(u), \alpha(v)]$ for any endomorphism α of G. Therefore, $\gamma_k(G)$ is a fully invariant subgroup of G for any $k \geq 1$, and so is $G'' = [G', G']$.

Definition 2. Let F_n be the free group of rank n. The relatively free group F_n/F_n'' is called the *free metabelian group* of rank n, which we denote by M_n.

Our choice of free metabelian groups for the platform is motivated by the following facts:

(1) The word problem in the group M_n is solvable in time $O(m^2n)$ with respect to the length m of a given word. Moreover, every element of M_n has a unique associated "normal form", which makes it possible to use our protocol for an efficient encryption/decryption of actual messages without prior common key establishment.

(2) M_n has exponential growth which provides for an exponential (with respect to the key size) key space.

(3) Subgroup membership (search) problem in M_n has no known polynomial-time solution.

Here (2) is a well-established fact [19], so we leave it without further comments. As for (3), there are two different solutions of the membership (decision) problem in M_n known by now [7, 21], but none of them is "even close" to yielding a polynomial-time algorithm for solving the membership search problem. We give more details in Section 5; here we concentrate on the word problem in M_n.

It is also interesting to note that free metabelian groups M_n are infinitely presented if $n \geq 2$ (see e.g. [3]), i.e., they cannot be defined by finitely many relations, in contrast with groups typically used in public key cryptography. Nevertheless, as we shall see in the next section, these groups have efficiently (and, in fact, very easily) solvable word problem.

Now we reproduce a result from [22] which provides us with "non-standard" automorphisms of free metabelian groups.

Theorem. [22] Let $\mathcal{F}_n = F_n/[R, R]$, and let $u \in R$. If $R \leq \gamma_3(F_n)$ and $n \geq 2$, then the following automorphism $\alpha_{u,j}$ of $\mathcal{F}_n$ is not tame: $\alpha_{u,j} : x_j \to x_j[x_j, u, x_j],\ x_i \to x_i,\ i \neq j$.

Note that $\alpha_{u,j}^{-1} : x_j \to x_j[x_j, u^{-1}, x_j],\ x_i \to x_i,\ i \neq j$. Also note that this theorem cannot be applied to free metabelian groups $M_n = F_n/[F_n', F_n']$ because $F_n' \not\leq \gamma_3(F_n)$. In fact, it is known that free metabelian groups M_n have only tame automorphisms if $n \neq 3$ [2]. However, automorphisms $\alpha_{u,j}$ described in the Theorem above, although tame in the case of a free metabelian group, are tame in a very nontrivial ("nonmonotonic") way, i.e., to factor them into a product of Nielsen automorphisms one has to first increase the length of $\alpha_{u,j}(x_i)$ before it could be decreased. This is supposed to foil "length based" attacks on φ (see Section 5 for more details).

To conclude this section, we modify Example 1 from the previous section by composing the automorphism φ from that example with the map $\alpha = \alpha_{[x_1,x_2],1}$ which is an automorphism of a free metabelian group, but not of a free group.

Example 2. *Let $n = 2, m = 1$, and let β_{jk} be Nielsen automorphisms of $\mathcal{F}_3$ defined in the Introduction. Let $\varphi = \beta_{23}^2\beta_{12}\beta_{31}\beta_{32}$. Then $\varphi : x_1 \to x_1x_3x_2^2,\ x_2 \to x_2x_1x_3x_2^2,\ x_3 \to x_3x_2^2$. Now compose φ with the map $\alpha : x_1 \to x_1[x_1, [x_1, x_2], x_1],\ x_i \to x_i,\ i \neq 1$. The resulting automorphism takes x_1 to $y_1 = x_1[x_1, [x_1, x_2], x_1]x_3x_2^2$, x_2 to $y_2 =$*

$x_2x_1[x_1,[x_1,x_2],x_1]x_3x_2^2$, and x_3 to $y_3 = x_3x_2^2$, and therefore $\hat{y_1} = x_1[x_1,[x_1,x_2],x_1]x_2^2$, $\hat{y_2} = x_2x_1[x_1,x_2],x_1]x_2^2$, $\hat{y_3} = x_2^2$.

Now in the free group F_2, the elements x_1 and x_2 no longer belong to the subgroup generated by $\hat{y_1}, \hat{y_2}, \hat{y_3}$. At the same time, since α is an automorphism of the free metabelian group F_3/F_3'', x_1 and x_2, considered as elements of F_3/F_3'', do belong to the subgroup of F_3/F_3'' generated by $\hat{y_1}, \hat{y_2}, \hat{y_3}$. The adversary will therefore have to solve the subgroup membership search problem in a free metabelian group, which is much more difficult than to do that in a free group.

4. Normal forms in free metabelian groups

In this section, we describe two different normal forms for elements of a free metabelian group M_n. The first one is good for transmissions, because it is easily convertible back to a word representing a transmitted element. However, this normal form is not unique if $n > 2$, so it cannot be used as a shared secret by Alice and Bob. For the latter purpose, the other normal form (a 2×2 matrix) can be used.

Let $u \in M_n$. By u_{ab} we denote the abelianization of u, i.e., the image of u under the natural epimorphism $\alpha : M_n \to M_n/[M_n, M_n]$. Note that we can identify $M_n/[M_n, M_n]$ with $F_n/[F_n, F_n]$. Technically, u_{ab} is an element of a factor group of F_n, but we also use the same notation u_{ab} for any word in the generators x_i (i.e., an element of the ambient free group F_n) representing u_{ab} when there is no ambiguity.

For $u, v \in M_n$, by u^v we denote the expression $v^{-1}uv$; we also say that v acts on u by conjugation. If $u \in [M_n, M_n]$, then this action can be extended to the group ring $\mathbb{Z}(M_n/[M_n, M_n])$ which we are going to denote by $\mathbb{Z}A_n$, to simplify the notation. (Here $A_n = M_n/[M_n, M_n]$ is the free abelian group of rank n.) Let $W \in \mathbb{Z}A_n$ be expressed in the form $W = \sum a_i v_i$, where $a_i \in \mathbb{Z}$, $v_i \in A_n$. Then by u^W we denote the product $\prod (u^{a_i})^{v_i}$. This product is well-defined since any two elements of $[M_n, M_n]$ commute in M_n.

Now let $u \in M_n$. Then u can be written in the following normal form:

$$u = u_{ab} \cdot \prod_{i<j} [x_i, x_j]^{W_{ij}} \tag{1}$$

where $W_{ij} \in \mathbb{Z}A_n$. To get to this form, one can use a "collecting process" based on the following identities (recall that $[x, y] = x^{-1}y^{-1}xy$):

$$[y, x] = [x, y]^{-1}$$

$$xy = yx[x, y]$$

$$xy^{-1} = y^{-1}[y, x]^{y^{-1}x^{-1}}x$$

$$x^{-1}y = y[y, x]^{y^{-1}x^{-1}}x^{-1}$$

$$[x,y]z = z[x,y]^z.$$

The collecting process itself is simple:

(1) Using the above identities, go left to right along the word u collecting all "non-commutator" occurrences of x_1 on the left (that means, do not worry about occurrences of the form $[x_1, x_j]$ or $[x_j, x_1]$ created in the process). Repeat this with x_2, x_3, etc. In the end of this process, u will be written in the form $u = u_{ab} \cdot c$, where $c \in [M_n, M_n]$ is a product of expressions of the form $[x_i, x_j]^g$, $g \in M_n$.

(2) Since any two elements of $[M_n, M_n]$ commute in M_n, one can now easily regroup the expressions $[x_i, x_j]^g$ so that u takes the form $u = u_{ab} \cdot \prod_{i<j} [x_i, x_j]^{W_{ij}}$, where $W_{ij} \in \mathbb{Z}A_n$.

This process apparently takes quadratic time with respect to the length of u.

To convert the normal form (1) to a word is trivial because (1) is, in fact, already a word. The only problem with (1) is that it is not unique if $n > 2$, so it cannot be used as a shared secret by Alice and Bob. For the latter purpose, we are now going to introduce another normal form which is unique, efficiently computable (in quadratic time with respect to the length of u), but not so easily convertible back to a word.

We have to first introduce *Fox derivatives*, which are noncommutative analogs of usual Leibniz derivatives.

Definition 3. Let $\mathbb{Z}F$ be the group ring of a free group F generated by $x_1, x_2, \ldots$. A *Fox derivation* with respect to x_i is a map $\partial_{x_i} : \mathbb{Z}F \to \mathbb{Z}F$ such that $\partial_{x_i}(x_j) = \delta_{ij}$ and $\partial_{x_i}(vw) = \partial_{x_i}(v) + v \cdot \partial_{x_i}(w)$ for any $v, w \in F$. *This map can be extended to the whole* $\mathbb{Z}F$ *by linearity.*

Example 3. Let $g \in F$ and let 1 be the identity of F. Since $\partial(1) = \partial(1) + \partial(1)$, it follows that $\partial(1) = 0$. Therefore $\partial(gg^{-1}) = \partial(g) + g\partial(g^{-1}) = 0$, which implies $\partial(g^{-1}) = -g^{-1}\partial(g)$.

Example 4. Let x and y be generators of the free group $F(x, y)$. Then

$$\begin{aligned} \partial_x([x,y]) = \partial_x(x^{-1}y^{-1}xy) &= \partial_x(x^{-1}) + x^{-1}\partial_x(y^{-1}) + x^{-1}y^{-1}\partial_x(x) + x^{-1}y^{-1}x\partial_x(y) \\ &= -x^{-1} + x^{-1}y^{-1} = x^{-1}y^{-1}(1-y)\,. \end{aligned}$$

$$\begin{aligned} \partial_y([x,y]) &= \partial_y(x^{-1}) + x^{-1}\partial_y(y^{-1}) + x^{-1}y^{-1}\partial_y(x) + x^{-1}y^{-1}x\partial_y(y) \\ &= -x^{-1}y^{-1} + x^{-1}y^{-1}x = -x^{-1}y^{-1}(1-x)\,. \end{aligned}$$

Let F_{ab} denote the abelianization of a free group F, i.e., the factor group F/F'. Let $\alpha : F \to F_{ab}$ be the natural epimorphism; it can be extended to the map $\alpha : \mathbb{Z}F \to \mathbb{Z}F_{ab}$ by linearity. A proof of the following proposition can be found in [14].

Proposition 1. Let $w \in F_n$. Then $w \in F_n''$ if and only if $\alpha(\partial_{x_i}(w)) = 0$ for each generator x_i of F_n.

This proposition yields a simple algorithm for solving the word problem in a free metabelian group M_n: given $w \in M_n$ as a word in relatively free generators x_i, one considers w an element of the free group F_n with the same set of free generators, computes $\partial_{x_i}(w)$ for each x_i, and checks whether or not all of them abelianize to 0. The latter is straightforward since the word problem in the free abelian group F_{ab} is easily solvable.

This algorithm is not only simple but efficient, too:

Proposition 2. The algorithm for solving the word problem in M_n based on Proposition 1 has at most quadratic time complexity with respect to the length of the input word.

Proof. Let $w \in F_n$ and let $|w| = m$ denote the usual lexicographic length of the word w. The computation of $\partial_{x_i}(w)$, for any generator x_i, produces at most m summands in the free group ring $\mathbb{Z}F_n$. Thus, the computation of ∂_x has at most linear time complexity with respect to m. Then, deciding whether or not the abelianization of $\partial_{x_i}(w)$ is 0 amounts to collecting summands of the form $c \cdot h_i,\ c \in \mathbb{Z}, h_i \in F_n$, such that all h_i have the same abelianization. This is achieved by rewriting every h_i in the form $x_1^{a_1}x_2^{a_2} \cdot ... \cdot x_n^{a_n} \cdot u_i$, where $u_i \in F_n'$. Since any h_i has length $\leq m$ and the number of h_i is at most m, this part of the algorithm takes time $O(m^2)$, which completes the proof. □

Finally, we describe the normal form of $u \in M_n$ based on Fox derivatives.

For an element $u \in M_n$ of a free metabelian group, its normal form is a 2×2 matrix with the following entries:

(1) The entry in the lower left corner is 0
(2) The entry in the lower right corner is 1
(3) The entry in the upper left corner is the abelianization of u, so it is an element of the free abelian group $M_n/[M_n, M_n]$
(4) The entry in the upper right corner is the most essential one. It is a vector of n abelianized partial Fox derivatives of the word u.

We note that the free abelian group $M_n/[M_n, M_n]$ acts on vectors of abelianized Fox derivatives by (componentwise) multiplication. This makes the set of normal forms a group under multiplication. Furthermore, the representation of elements of M_n by their normal forms is faithful, i.e., we actually have an embedding of M_n into a group of matrices; this is called *Magnus embedding* (see e.g. [14]).

5. Parameters and cryptanalysis

There are two visible ways to attack our cryptosystem: (1) trying to find the automorphism φ (and its inverse) based on the restriction $\hat{\varphi}$ and (2) trying to solve the subgroup membership (search) problem in the platform group. As we have already pointed out in Section 2, the first way looks intractable to us, so we focus on the subgroup membership problem here, but first we make a relevant remark about the word problem.

In the previous section, we obtained the solvability of the word problem in free metabelian groups M_n by using Fox derivations. Here we note that the solvability of the word problem in M_n can also be obtained by a much less efficient but nevertheless interesting method, using the fact that M_n are *residually finite groups* [15].

Definition 4. A group G is said to be *residually finite* if for every $g \in G - \{1\}$ there exists a finite group H_g and a homomorphism $\varphi : G \to H_g$ such that $\varphi(g) \neq 1$.

If a group $G = F/R$ is residually finite, then it has the word problem solvable as follows. Given $g \in G$ as a word in the generators of F, two algorithms run in parallel. One of them goes over all finite homomorphic images of G (those are recursively enumerable) and checks whether $\varphi(g) \neq 1$ in any of them (note that the word problem is solvable in any finite group). Thus, if $g \neq 1$ in G, this algorithm will eventually detect that. The other algorithm just recursively enumerates all elements of R and compares them to g one at a time. Thus, if $g = 1$ in G, this algorithm will eventually detect that.

This way of solving the word problem is "cute", but it is obviously useless in real life. We have mentioned it here only because there is a similar (equally useless) way of solving the membership problem in a special class of groups.

Definition 5. A group G is said to be *locally extended residually finite* (LERF) if for every subgroup $K \in G$ and $g \in G - K$, there exists a finite group H and a homomorphism $\varphi : G \to H$ such that $\varphi(g) \in \varphi(G) - \varphi(K)$.

Sometimes, LERF groups are also called *subgroup separable*.

Coulbois [7] showed that free metabelian groups are LERF. If a group G is LERF, it follows that G has solvable subgroup membership (decision) problem by way of an algorithm similar to what was described above. However, as well as the algorithm for solving the word problem based on residual finiteness of a given group, this algorithm is anything but practical.

An alternative algorithm for solving the subgroup membership (both decision and search) problem in M_n was offered in [21]. This algorithm is somewhat more "down-to-earth", but it is still far from being practical because it involves, among other things, computing a Gröbner basis of an ideal of the ring of Laurent polynomials in n variables.

In the absence of feasible deterministic attacks, one might try heuristic ("length based") attacks in the spirit of [9]. We point out however that groups that were typically used as proving ground for "length based" attacks are very different in nature from free metabelian groups, and the length function for free metabelian groups is, informally speaking, much less predictable than it is, say, for braid groups, which probably means that "length based" attacks are going to be less successful.

We are now going to discuss specific parameters of our cryptosystem. First, we suggest to use a free metabelian group as the platform even though there is an approach to solving the subgroup membership problem in such a group mentioned above. We believe that this approach has only a theoretical advantage over the exhaustive search, and therefore it is not a threat to security of the cryptosystem.

In the interests of efficiency, we suggest to use a free metabelian group of a fairly small rank as the platform. The rank that we suggest is $r = n + m = 10$, where $n = 8,\ m = 2$.

Then, an important parameter is the number of "elementary" automorphisms in the factorization $\varphi = \tau_1 \cdots \tau_k$ of the key automorphism φ. One has to be careful here because, say, the automorphism group of a free group has exponential growth with respect to the (finite) generating set that consists of all Nielsen automorphisms. This implies, in particular, that the length of $\varphi(x_i)$ grows exponentially with k. However, in the free group of rank $r = 10$ the base of this exponent is so small (the higher the rank the smaller the base) that, according to our experiments, for a random product of 30 Nielsen automorphisms the average length of $\varphi(x_i)$ is under 30, which is quite reasonable.

Thus, we suggest $k = 30$, of which 90% should be random Nielsen automorphisms and 10% automorphisms of the type described in the Theorem in Section 3, i.e., $\alpha_{u,j} : x_j \to x_j[x_j, u, x_j],\ x_i \to x_i,\ i \neq j$, for random elements $u \in [F_r, F_r]$. That means, when building the automorphism φ as a product of "elementary" automorphisms one factor at a time, Alice selects, at each step, a Nielsen automorphism with probability 90% and an automorphism of type $\alpha_{u,j}$ with probability 10%.

This provides for sufficiently large key space because there are approximately 100 Nielsen automorphisms in the free (or relatively free) group of rank 10, so products of 30 Nielsen automorphisms already give us approximately $100^{30} = 10^{60}$ choices for φ. On top of that, there are automorphisms $\alpha_{u,j}$ with arbitrary $u \in [F_r, F_r]$. The length of u has to be bounded by, say, 10, to keep the complexity of φ within reasonable limits. This leaves us with the pool of approximately 10^7 elements to choose u from.

With these parameters, the average total length of $\varphi(x_i),\ i = 1, ..., 10$, is under 1000, according to our experiments.

Acknowledgement. We are grateful to Dennis Hofheinz for helpful comments and suggestions.

References

[1] I. Anshel, M. Anshel, D. Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Lett. **6** (1999), 287–291.

[2] S. Bachmuth, H. Y. Mochizuki, *$Aut(F) \to Aut(F/F'')$ is surjective for free group F of rank ≥ 4*, Trans. Amer. Math. Soc. **292** (1985), 81–101.

[3] G. Baumslag, R. Strebel, M. W. Thomson, *On the multiplicator of $F/\gamma_c R$*, J. Pure Appl. Algebra **16** (1980), 121–132.

[4] J.-C. Birget, S. Magliveras, M. Sramka, *On public-key cryptosystems based on combinatorial group theory*, preprint.

[5] R. M. Bryant, C. K. Gupta, F. Levin, and H. Y. Mochizuki, *Non-tame automorphisms of free nilpotent groups*, Comm. Algebra, **18** (1990), 3619–3631.

[6] J. C. Cha, K. H. Ko, S. J. Lee, J. W. Han, J. H. Cheon, *An Efficient Implementation of Braid Groups*, ASIACRYPT 2001, Lecture Notes in Comput. Sci. **2248** (2001), 144–156.

[7] T. Coulbois, *Propriétés de Ribes-Zaleskii, topologie profinie, produit libre et généralisations*, Ph. D. Thesis, Université de Paris VII, 2000.

[8] B. Eick, D. Kahrobaei, *Polycyclic groups: A new platform for cryptology?*, preprint. http://arxiv.org/abs/math.GR/0411077

[9] D. Garber, S. Kaplan, M. Teicher, B. Tsaban, U. Vishne, *Probabilistic solutions of equations in the braid group*, preprint.
http://arxiv.org/abs/math.GR/0404076
[10] M. Garzon, Y. Zalcstein, *The complexity of Grigorchuk groups with application to cryptography*, Theoret. Comput. Sci. **88** (1991), 83–98.
[11] L. Goubin, N. T. Courtois, *Cryptanalysis of the TTM cryptosystem*, Asiacrypt 2000. Lecture Notes in Comput. Sci. **1976**, pp. 44–57, Springer, 2000.
[12] D. Grigoriev, I. Ponomarenko, *On non-abelian homomorphic public-key cryptosystems*, preprint.
http://arxiv.org/abs/cs.CR/0207079
[13] D. Grigoriev, I. Ponomarenko, *Homomorphic public-key cryptosystems over groups and rings*, preprint.
http://arxiv.org/abs/cs.CR/0309010
[14] N. Gupta, *Free Group Rings*, Contemp. Math. **66**. Amer. Math. Soc., 1987.
[15] P. Hall, *On the finiteness of certain soluble groups*, Proc. London Math. Soc. **9** (1959), 565–622.
[16] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang, C. Park, *New public-key cryptosystem using braid groups*, Advances in cryptology—CRYPTO 2000 (Santa Barbara, CA), 166–183, Lecture Notes in Comput. Sci. **1880**, Springer, Berlin, 2000.
[17] R. C. Lyndon and P. E. Schupp, *Combinatorial Group Theory*, Ergebnisse der Mathematik, band 89, Springer 1977. Reprinted in the Springer Classics in Mathematics series, 2000.
[18] M. R. Magyarik, N. R. Wagner, *A Public Key Cryptosystem Based on the Word Problem*. CRYPTO 1984, 19–36, Lecture Notes in Comput. Sci. **196**, Springer, Berlin, 1985.
[19] J. Milnor, *Growth of finitely generated solvable groups*, J. Differential Geometry **2** (1968), 447–449.
[20] T. Moh, *A public key system with signature and master key functions*, Comm. Algebra **27** (1999), 2207–2222.
[21] N. S. Romanovskii, *The occurrence problem for extensions of abelian by nilpotent groups*, Sib. Math. J. **21** (1980), 170–174.
[22] V. Shpilrain, *Automorphisms of F/R' groups*, Internat. J. Algebra and Comput. **1** (1991), 177–184.
[23] V. Shpilrain and A. Ushakov, *Thompson's group and public key cryptography*, Lecture Notes Comp. Sc. **3531** (2005), 151–164.
[24] V. Shpilrain and A. Ushakov, *The conjugacy search problem in public key cryptography: unnecessary and insufficient*, Applicable Algebra in Engineering, Communication and Computing, to appear.
http://eprint.iacr.org/2004/321/
[25] V. Shpilrain and G. Zapata, *Combinatorial group theory and public key cryptography*, Applicable Algebra in Engineering, Communication and Computing, to appear.
http://eprint.iacr.org/2004/242/

Department of Mathematics, The City College of New York, New York, NY 10031
e-mail address: shpil@groups.sci.ccny.cuny.edu
http://www.sci.ccny.cuny.edu/˜shpil/

Department of Mathematics, CUNY Graduate Center, New York, NY 10016
e-mail address: nyzapata@verizon.net