Cryptanalysis of a homomorphic public-key cryptosystem over a finite group

Su-Jeong Choi
Simon R. Blackburn
and
Peter R. Wild
Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX
United Kingdom

September 13, 2006

Abstract

The paper cryptanalyses a public-key cryptosystem recently proposed by Grigoriev and Ponomarenko, which encrypts an element from a fixed finite group defined in terms of generators and relations to produce a ciphertext from $\mathrm{SL}(2, \mathbb{Z})$. The paper presents a heuristic method for recovering the secret key from the public key, and so this cryptosystem should not be used in practice.

Key words: Cryptanalysis, homomorphic cryptosystem, combinatorial group theory.
AMS Mathematics Subject Classification: 94A60

1 Introduction

Dima Grigoriev and Ilia Ponomarenko [2] have recently proposed a public key cryptosystem which takes an arbitrary finite group, given in terms of generators and relations, as its plaintext space and encrypts to a ciphertext space which is a subset of $\mathrm{SL}(2, \mathbb{Z})$. The aim of this paper is to show that this cryptosystem is insecure when used in practice. The material in this paper is extracted from the PhD thesis of one of the authors [1].

The structure of the rest of this paper is as follows. Section 2 recaps and recasts some of the material on free groups used by Grigoriev and Pomomarenko [2], and establishes the notation used in the rest of the paper. Section 3 describes the cryptosystem itself. This description differs somewhat from that of Grigoriev and Ponomarenko, and is designed to clarify our cryptanalysis in Section 4 as much as possible. Section 5 contains a brief conclusion.

2 Free groups

This section (and the remainder of the paper) will use the standard terminology of combinatorial group theory; see Lyndon and Schupp [3], for example, for an introduction to the area.

Let $F_{\{a,b\}}$ be the free group with free generating set $\{a, b\}$. We represent the elements of $F_{\{a,b\}}$ (and the elements of any subgroup of $F_{\{a,b\}}$) as reduced words in $\{a, b\}^{\pm 1}$.

For $i \in \mathbb{Z}$, define $c_i = a^{-i}ba^i$. Define $\mathcal{C} = \{c_i : i \in \mathbb{Z}\}$, and let L be the subgroup generated by $\mathcal{C}$, so $L = \langle \{c_i \mid i \in \mathbb{Z}\} \rangle$. In fact, L is a free group of countable (infinite) rank with $\mathcal{C}$ as a free generating set, by [3, Page 7, Proposition 2.5]. There is a simple and efficient algorithm to test whether an element $g \in F_{\{a,b\}}$ lies in L, and if so to produce the representation of g as a reduced word in $\mathcal{C}^{\pm 1}$ (which we call the *$\mathcal{C}$-representation* of g). The algorithm may be stated as follows.

Algorithm (The $\mathcal{C}$-representation algorithm)

Input: A reduced word $g = a^{\alpha_0} b^{\beta_1} a^{\alpha_1} \cdots b^{\beta_u} a^{\alpha_u}$.

If $g = 1$, output 'g lies in L, and has $\mathcal{C}$-representation the empty word', and stop.

If $\sum_{i=0}^{u} \alpha_i \neq 0$, output '$g$ does not lie in L' and stop.

Define integers $\sigma_1, \sigma_2, \ldots, \sigma_u$ by $\sigma_i = -\sum_{j=0}^{i-1} \alpha_j$.

Output 'g lies in L, and has $\mathcal{C}$-representation $c_{\sigma_1}^{\beta_1} \cdots c_{\sigma_u}^{\beta_u}$.'

We remark that the condition that g is given in reduced form implies that $\alpha_j \neq 0$ when $1 \leq j \leq u-1$ and that $\beta_i \neq 0$ for $1 \leq i \leq u$.

The algorithm is not difficult to justify and we omit the details. Grigoriev and Ponomarenko give a similar algorithm [2, Lemma 2.3] in their paper, but for a certain finitely generated subgroup of L.

Let $S \subseteq \mathbb{Z}$. Define $\mathcal{C}_S = \{c_s : s \in S\}$, and let L_S be the subgroup of $F_{\{a,b\}}$ generated by $\mathcal{C}_S$. It is clear that the $\mathcal{C}$-representation algorithm above may be used to determine whether $g \in F_{\{a,b\}}$ lies in L_S, and to compute a representation of g as a word in $\mathcal{C}_S^{\pm 1}$ (the *$\mathcal{C}_S$-representation* of g) if this is the case. To see this, first use the $\mathcal{C}$-representation algorithm to compute a representation of g as a reduced word in $\mathcal{C}^{\pm 1}$. If $g \notin L$, or if $g \in L$ but the word in $\mathcal{C}^{\pm 1}$ involves generators c_i where $i \notin S$, then $g \notin L_S$. Otherwise $g \in L_S$ and we have computed the $\mathcal{C}_S$-representation of g.

We close this section by defining a little more notation that we need to describe the cryptosystem.

Let n be an integer, and suppose that $n \geq 2$. Define integer matrices A_n and B_n by

$$A_n = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix} \text{ and } B_n = \begin{pmatrix} 1 & 0 \\ n & 1 \end{pmatrix}.$$

Let $\Gamma_n = \langle A_n, B_n \rangle$. Then Γ_n is a free subgroup of $\mathrm{SL}(2, \mathbb{Z})$, freely generated by $\{A_n, B_n\}$ by [3, Page 168]. Write $\phi_n : F_{\{a,b\}} \to \Gamma_n$ for the isomorphism such that $\phi_n(a) = A_n$ and $\phi_n(b) = B_n$. It is clearly easy to compute $\phi_n(x)$ for an element $x \in F_{\{a,b\}}$ when x and n are given. In fact, though this is a little less obvious, given n and a matrix X in the image of ϕ_n it is not difficult to compute $\phi_n^{-1}(X)$. See [2, Lemma 2.4] for details.

3 The cryptosystem

This section describes the public key cryptosystem due to Grigoriev and Ponomarenko [2]. We use the notation defined in Section 2 extensively. Our notation differs slightly from that used by Grigoriev and Ponomarenko, to facilitate the description of our cryptanalyis in Section 4 below.

Let H be a fixed finite group, given in terms of generators and relations. So $H = \langle \mathcal{X} \mid \mathcal{R} \rangle$ where $\mathcal{X} = \{x_1, x_2, \dots, x_t\}$ is a finite set of generators, and $\mathcal{R}$ is a set of relations. We assume that $t \geq 2$. Write $F_{\mathcal{X}}$ for the free group freely generated by the set $\mathcal{X}$, so $\mathcal{R} \subseteq F_{\mathcal{X}}$. Define N to be the normal closure of $\mathcal{R}$ in $F_{\mathcal{X}}$. Then $H = F_{\mathcal{X}}/N$. We represent elements of H by reduced words in $F_{\mathcal{X}}$, so a given element of H will have many representations. The group H will form the plaintext space of the cryptosystem; both H and the representation of its elements will remain fixed throughout.

The secret and public keys of the cryptosystem are generated as follows. Choose distinct integers $s_1, s_2, \dots, s_t \in \mathbb{Z}$. Let $S = \{s_1, s_2, \dots, s_t\}$, and let $\mathbf{s} = (s_1, s_2, \dots, s_t) \in \mathbb{Z}^t$. Let $\psi_{\mathbf{s}} : F_{\mathcal{X}} \to L_S$ be the unique homomorphism such that $\psi_{\mathbf{s}}(x_i) = c_{s_i}$ for $i \in \{1, 2, \dots, t\}$. (Thus we have identified $F_{\mathcal{X}}$ with the subgroup L_S of $F_{\{a,b\}}$.) Note that it is easy to compute $\psi_{\mathbf{s}}$ when $\mathbf{s}$ is known; the same is true for $\psi_{\mathbf{s}}^{-1}$, since it is easy to compute the $\mathcal{C}_S$-representation of an element in L_S. Choose $r_1, r_2, \dots, r_t \in N$. Choose $n \in \mathbb{Z}$ such that $n \geq 2$. For $i \in \{1, 2, \dots, t\}$, define a matrix $W_i \in \mathrm{SL}(2, \mathbb{Z})$ by $W_i = \phi_n \circ \psi_{\mathbf{s}}(x_i r_i)$. The public key of the cipher is $\mathcal{X}$, $\mathcal{R}$ together with the matrices $W_1, W_2, \dots, W_t$. The private key is $\mathcal{X}$, $\mathcal{R}$, n and $\mathbf{s}$.

We remark that Grigoriev and Ponomarenko [2] define the private key as a different set of parameters. Define matrices $X_1, X_2, \dots, X_t$ by $X_i = A_n^{-s_i} B_n A_n^{s_i}$. Grigoriev and Ponomarenko replace $\mathbf{s}$ in the private key by the sequence of matrices $X_1, X_2, \dots, X_t$. Note that the two forms of the private key are equivalent. To see this, firstly note that the matrices X_i can obviously be calculated knowing $\mathbf{s}$ and n. But knowing n and the matrices X_i allows us to compute $\mathbf{s}$ easily, since $a^{-s_i} b a^{s_i} = \phi_n^{-1}(X_i)$, and ϕ_n can be efficiently inverted [2, Lemma 2.4] when n is known.

To encrypt a plaintext $h \in H$, a user represents h in the form $h = xN$, where

$$x = x_{d_1}^{\delta_1} x_{d_2}^{\delta_2} \cdots x_{d_u}^{\delta_u}$$

for some $d_i \in \{1, 2, \dots, t\}$ and some $\delta_i \in \{1, -1\}$. The user then chooses an element $r \in N$; we write

$$r = x_{e_1}^{\epsilon_1} x_{e_2}^{\epsilon_2} \cdots x_{e_v}^{\epsilon_v}$$

where $e_i \in \{1, 2, \dots, t\}$ and $\epsilon_i \in \{1, -1\}$. The ciphertext $M \in \mathrm{SL}(2, \mathbb{Z})$ is defined by

$$M = W_{e_1}^{\epsilon_1} W_{e_2}^{\epsilon_2} \cdots W_{e_v}^{\epsilon_v} W_{d_1}^{\delta_1} W_{d_2}^{\delta_2} \cdots W_{d_u}^{\delta_u}.$$

We give the following method to decrypt (which differs from that of Grigoriev and Ponomarenko [2], who base their decryption procedure on a normal form algorithm tailored to the group of matrices generated by $X_1, X_2, \ldots, X_t$). The holder of the private key computes $y = \psi_{\mathbf{s}}^{-1} \circ \phi_n^{-1}(M)$. This is possible since ϕ_n can be efficiently inverted as n is known, and $\psi_{\mathbf{s}}$ can be efficiently inverted since $\mathbf{s}$ is known. We claim that yN is a representative of h, and so the plaintext is yN. To see this, note that

$$\begin{aligned} M &\in \phi_n \circ \psi_{\mathbf{s}}(x_{e_1}^{\epsilon_1} x_{e_2}^{\epsilon_2} \cdots x_{e_v}^{\epsilon_v} x_{d_1}^{\delta_1} x_{d_2}^{\delta_2} \cdots x_{d_u}^{\delta_u} N) \quad (\text{since } W_i \in \phi_n \circ \psi_{\mathbf{s}}(x_i N)) \\ &= \phi_n \circ \psi_{\mathbf{s}}(x_{d_1}^{\delta_1} x_{d_2}^{\delta_2} \cdots x_{d_u}^{\delta_u} N) \quad (\text{since } r \in N) \\ &= \phi_n \circ \psi_{\mathbf{s}}(xN), \end{aligned}$$

and so $y \in xN$ and our claim follows.

We end this section with some comments on the complexity of the algorithm.

At several points during key generation and encryption elements are chosen from an infinite set; it is important to specify how this is done and Grigoriev and Ponomarenko suggest the following. Let k be a parameter that we use to measure complexity. The integers n and s_i are chosen to be random $O(k)$-bit integers. The random elements $r_i \in N$ (chosen during key generation) and $r \in N$ (chosen during encryption) are words of length $O(k)$ in the set $\mathcal{R}$ of relations in the presentation of H. (So these elements are not arbitrary elements of N, but rather are contained in the semigroup generated by $\mathcal{R}$.) When choices are made in this way, all the computations above described as efficient are polynomial in k. Note that the presentation of H does not vary as k varies; in particular t is a constant.

Note that the representative x of the plaintext $h \in H$ will in general be different to the representative y of h which is returned by the decryption process. If data is to be transmitted efficiently using the Grigoriev–Ponomarenko cryptosystem (or any cryptosystem with similar properties) then the problem of recognising that x and y represent the same piece of data will have to be confronted. One solution would be to insist that H has an efficiently soluble word problem.

4 A cryptanalysis

This section shows how the private key of the cryptosystem can be recovered from the public key. Parts of our argument are heuristic, backed by experimental evidence. In describing the cryptosystem above, we phrased the encryption process as the composition of two maps $\psi_{\mathbf{s}}$ and ϕ_n which depend on independent parts of the secret key. Of course, a user would not have access to the secret key, and so would not encrypt by computing images under these two maps; rather, this description of the cryptosystem is designed to suggest a method of cryptanalysis which proceeds in two phases.

In the first phase of our cryptanalysis, we claim that we can recover n efficiently from the public key. Let K be the kernel of the obvious homomorphism from $\mathrm{SL}(2,\mathbb{Z})$ to $\mathrm{SL}(2,\mathbb{Z}_n)$. Since $A_n, B_n \in K$, and all matrices in the cryptosystem lie in the group Γ_n generated by A_n and B_n, we find that $W_i \in K$ for all $i \in \{1, 2, \ldots, t\}$. Now,

$$K = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} : a - 1 \equiv b \equiv c \equiv d - 1 \equiv 0 \bmod n \right\}.$$

Writing $W_i = \left(\begin{smallmatrix} w_{i11} & w_{i12} \\ w_{i21} & w_{i22} \end{smallmatrix} \right)$ we find that n divides n', where

$$n' = \gcd\left(\bigcup_{i=1}^{t} \{w_{i11} - 1, w_{i12}, w_{i21}, w_{i22} - 1\} \right).$$

Since the integers w_{ijk} form part of the public key, we can compute n' efficiently. We would expect that the bitlength of the integers w_{ijk} will grow faster than the bitlength of n as the complexity parameter k increases, as each matrix W_i is the image of a product in $F_{\mathcal{X}}$ involving $O(k)$ relations. Thus we expect that the probability that $n = n'$ will remain large as $k \to \infty$ unless the presentation of H has a very unusual form. In our experiments, we always found that n'/n was very small; in 1000 trials, with k ranging up to 140, the largest value of n'/n we found was 28. Thus in all cases, it is reasonable to assume that n can be guessed, since n is of the form $n = n'/\ell$, where ℓ is small, with very high probability.

The second phase of our cryptanalysis aims to recover $\mathbf{s}$, the remaining part of the private key. We assume that n is known. Note that since the presentation of H does not vary as the complexity parameter k varies, it is sufficient to recover the set S: once S is known, there are at most $t!$ possibilities for $\mathbf{s}$ where t is a constant.

For $i \in \{1, 2, \ldots, t\}$, define $w_i \in F_{\{a,b\}}$ by $w_i = \psi_{\mathbf{s}}(x_i r_i)$. Since we are assuming that n is known, and since $w_i = \phi_n^{-1}(W_i)$, we are able to compute $w_1, w_2, \ldots, w_t$ efficiently. Note also that $w_i \in L_S$, since $w_i \in \mathrm{im}\psi_{\mathbf{s}}$. Using the

$\mathcal{C}$-representation algorithm from Section 2, we may write each w_i as a product of elements c_s. Since we know $w_i \in L_S$, we find that whenever c_s appears in the $\mathcal{C}$-representation of w_i we have that $s \in S$. For $i \in \{1, 2, \ldots, t\}$, define S_i to be the set of integers s such that c_s appears in the $\mathcal{C}$-representation of w_i. Define $S' = \cup_{i=1}^{t} S_i$. We know that $S' \subseteq S$. We will now argue that $S' = S$ with probability tending to 1 as $k \to \infty$, provided the following conjecture is true.

Conjecture 1 *Let $F_{\mathcal{X}}$ be a free group on the set $\{x_1, x_2, \ldots, x_t\}$. Let $i \in \{1, 2, \ldots, t\}$ be fixed. Let $z_1, z_2, \ldots, z_\ell \in F_{\mathcal{X}}$ be reduced words, at least one of which involves x_i. Let z be the reduced word formed by taking a product of k of the words z_j, chosen uniformly and independently at random. Then z involves x_i with probability tending to 1 as $k \to \infty$.*

To see that the conjecture implies that $S' = S$ with probability tending to 1, we argue as follows. Let $i \in \{1, 2, \ldots, t\}$ be fixed. We claim that $s_i \in S$ is contained in S' with probability tending to 1. Since H is a finite group, at least one of the relations in $\mathcal{R}$ involves x_i, and so the conjecture implies that the relations $r_1, r_2, \ldots, r_t$ involve x_i with probability tending to 1. Let $j \in \{1, 2, \ldots, t\} \setminus \{i\}$. Then $x_j r_j$ involves x_i with probability tending to 1, as $x_j r_j$ involves x_i if and only if r_j involves x_i. But the definition of $\psi_{\mathbf{s}}$ now implies that the $\mathcal{C}$-representation of w_j involves c_{s_i} with probability tending to 1; thus $s_i \in S_j \subseteq S'$ with probability tending to 1, which establishes our claim. Thus $S' = S$ with probability tending to 1, provided the conjecture above is true.

There is strong evidence for the truth of the conjecture. The conjecture seems to be true in all the examples we have examined by computer. Moreover, we can see that the conjecture is true in the 'generic' case as follows. Define the x_i-weight $\mathrm{wt}_{x_i}(z)$ of a word z to be the sum of the powers of x_i occurring in the word. So, for example,

$$\mathrm{wt}_{x_1}(x_1^2 x_2 x_4 x_1^{-1}) = 2 + (-1) = 1.$$

We claim that the conjecture is true in the case when at least one of the words z_i has non-zero x_i-weight. To see this, note that the map $z \mapsto \mathrm{wt}_{x_i}(z)$ is a homomorphism from $F_{\{a,b\}}$ to the additive group of $\mathbb{Z}$. Thus the x_i-weight of the word z in the conjecture is the result of k steps of a random walk on $\mathbb{Z}$, with step sizes $\mathrm{wt}_{x_i}(z_1), \mathrm{wt}_{x_i}(z_2), \ldots, \mathrm{wt}_{x_i}(z_\ell)$ (each taken with equal probability). It is not difficult to show that $\mathrm{wt}_{x_i}(z) = 0$ with probability tending to 0 as $k \to \infty$. (One way of proving this is to consider the integers modulo q, where q is a large prime, and use the Perron–Frobenius theorem to show that the kth state in an induced random walk on $\mathbb{Z}_q$ converges to the uniform distribution as $k \to \infty$.) Thus $\mathrm{wt}_{x_i}(z) \neq 0$ with probability tending to 1; but when $\mathrm{wt}_{x_i}(z) \neq 0$ then clearly z involves x_i. So the conjecture is true in this case. (It is possible that this argument could be generalised to prove the conjecture in full, using some sort of collection process.)

Finally, we note that if none of the generators x_i in the presentation of H are redundant, then $S' = S$ with probability 1. For suppose that $s_i \notin S'$. We will prove that x_i is a redundant generator. Our assumption implies in particular that $s_i \notin S_i$. Thus c_{s_i} does not occur in the $\mathcal{C}$-representation of w_i; the definitions of $\psi_{\mathbf{s}}$ and w_i then imply that $x_i r_i = z$, where $z = \psi_{\mathbf{s}}^{-1}(w_i)$ is a reduced word not involving x_i. Since $r_i \in N$, we find that $x_i N = zN$, and so $x_i N$ is in the subgroup generated by $x_1N, x_2N, \ldots, x_{i-1}N, x_{i+1}N, \ldots, x_tN$. Thus x_i is a redundant generator, as required.

5 Conclusion

We have presented a heuristic argument, backed up by theoretical and experimental evidence, which shows that the cryptosystem considered in this paper is insecure: the private key can be derived from the public key. It might be possible to salvage the security of the cryptosystem if the presentation of the group H is chosen very carefully. It would be interesting if specific presentations for groups H which were secure for practical parameter sizes could be found; however, it is far from obvious how to do this.

References

[1] Su-Jeong Choi, *Cryptanalysis of a homomorphic public-key cryptosystem*, University of London PhD thesis, 2006.

[2] Dima Grigoriev and Ilia Ponomarenko, ‘Homomorphic public-key cryptosystems over groups and rings’, in (Jan Krajíček, editor) *Complexity of computations and proofs, Quaderni di Matematica, Vol 13*, Dept. of Mathematics, Seconda Università di Napoli, Caserta, 2004, 305–325.

[3] Roger C. Lyndon and Paul E. Schupp, *Combinatorial group theory*, Springer-Verlag, Berlin–New York, 1977.