Advances in Group Theory and Applications

13 (2022), pp. 61–70 ISSN: 2499-1287
DOI: 10.32037/agta-2022-005

The Multiple Conjugacy Search Problem in Virtually Nilpotent Polycyclic Groups *

C. Monetta — A. Tortora

(*Received June 20, 2021; Accepted July 29, 2021 — Communicated by M.R. Dixon*)

Abstract

We present an algorithm for solving the multiple conjugacy search problem in virtually nilpotent polycyclic groups.

Mathematics Subject Classification (2020): 20F16, 20F10, 94A60

Keywords: conjugacy problem; centralizer; polycyclic group

1 Introduction

In cryptography one of the first attempts to use non-abelian groups was the key exchange protocol of Anshel-Anshel-Goldfeld [1], AAG for short, which we briefly describe here.

Let G be a non-abelian group and let $a_1, \dots, a_k, b_1, \dots, b_m \in G$. The group G as well as the elements a_i, b_j are public. For arbitrary $x, g \in G$, we write x^g for $g^{-1}xg$.

(1) Alice chooses a private $x \in G$ as a word in $a_1, \dots, a_k$, and sends $b_1^x, \dots, b_m^x$ to Bob.

(2) Bob chooses a private $y \in G$ as a word in $b_1, \dots, b_m$, and sends $a_1^y, \dots, a_k^y$ to Alice.

* This research is supported by a grant of the University of Campania "Luigi Vanvitelli", in the framework of the project GoAL (V:ALERE 2019)

(3) Alice computes x^y and Bob computes y^x.

(4) Alice and Bob can use $[x,y] = x^{-1}y^{-1}xy$ as a shared key: in fact, $[x,y] = x^{-1}x^y = (y^x)^{-1}y$.

This protocol requires that the *Word Problem* can be solved efficiently in G: given a finitely presentation $\langle X \mid R \rangle$ for G and a word w in the elements of X, decide whether $w = 1$ in G. On the other hand, the security relies on the *Multiple Conjugacy Search Problem*: given $x_1, \ldots, x_n$, $y_1, \ldots, y_n \in G$ such that $x_i = y_i^g$ for any i in $\{1, \ldots, n\}$ and some $g \in G$, find such an element g.

As shown in [2], the multiple conjugacy search problem can be reduced to the (single) conjugacy search problem ($i = 1$), provided that the centralizer corresponding to conjugating element can be computed. Indeed, from $x_1 = y_1^{g_1}$ and $x_1 = y_1^{g_2}$, it follows that

$$x_1^{g_1^{-1}} = x_1^{g_2^{-1}},$$

that is, $g_1^{-1}g_2 = c \in C_G(x_1)$.

By induction, suppose that an element $g_j \in G$ is such that $x_k = y_k^{g_j}$ for any $k \in \{1, \ldots, j\}$, and let

$$G_j = C_{G_{j-1}}(x_j) = C_G(x_1, \ldots, x_j)$$

where $G_1 = C_G(x_1)$. Then an element $g_{j+1} \in G$ with $x_k = y_k^{g_{j+1}}$ for all k and $x_{j+1} = y_{j+1}^{g_{j+1}}$ is of the form $g_{j+1} = g_j c$ with $c \in G_j$. Hence one can determine $c \in G_j$ such that $x_{j+1} = (y_{j+1}^{g_j})^c$ and, simultaneously, $G_{j+1} = C_{G_j}(x_{j+1})$. This process yields an element $g = g_n$ eventually.

In [2], Eick and Kahrobaei proposed the use of polycyclic groups as a platform for the AAG protocol. Recall that a group G is said to be *polycyclic* if G admits a cyclic series, that is, a (finite) series of subgroups

$$G = G_1 \rhd G_2 \rhd \ldots \rhd G_{n+1} = 1$$

whose factors are cyclic; in particular the group G is called *supersoluble* if each G_i is a normal subgroup of G. The class of polycyclic groups is an important subclass of soluble groups. In fact, a group is polycyclic if and only if it is soluble and satisfies the maximal condition [9, 5.4.12]. Hence all subgroups of a polycyclic group are finitely generated [9, 3.1.6]. Conversely, finite soluble groups as well

as finitely generated nilpotent groups [9, 5.2.17] are polycyclic.

Unlike an arbitrary group, a polycyclic group allows effective computations within the group. For example, an induced presentation for any subgroup of a polycyclic group can be computed from a generating set. Thus, for polycyclic groups, the multiple conjugacy search problem reduces to n applications of the conjugacy search problem with the determination of the corresponding centralizers. For this purpose there exists an algorithm due to Eick and Ostheimer [4], which makes use of finite orbits and stabilizer computations. The situation is much easier for finitely generated nilpotent groups. In [10], for such groups, Sims proposed two efficient algorithms for solving the conjugacy search problem and computing centralizers. These algorithms depends on some techniques which are used to find the image, the kernel and the inverse images of elements of a homomorphism between polycyclic groups with a polycyclic presentation (see Section 2). An analysis of the computational complexity of the algorithms can be found in [8].

The aim of this note is to show that Sim's algorithms can be extended to polycyclic groups which are virtually nilpotent, namely with a nilpotent subgroup of finite index. This is in part motivated by the fact that supersoluble groups are virtually nilpotent [9, 5.2.17]. On the other hand, it is known that the growth rate of a polycyclic group is polynomial if and only if the group is virtually nilpotent [11]. Thus, in our context, the existence of practical algorithms for virtually nilpotent groups confirms that these groups are not good candidates for the AAG protocol using polycyclic groups. We refer to [5] for the definition of growth rate and an account on polycyclic group-based cryptography.

2 Computational properties of polycyclic groups

Let G be a polycyclic group with a cyclic series

$$G = G_1 \rhd G_2 \rhd \cdots \rhd G_{n+1} = 1$$

and a polycyclic (generating) sequence of elements $X = (x_1, \ldots, x_n)$, that is, such that

$$x_i \in G_i \quad \text{and} \quad \langle x_i G_{i+1} \rangle = G_i / G_{i+1}.$$

Denote by $I = I(x_1, \dots, x_n)$ the set of all $i \in \{1, \dots, n\}$ such that G_i/G_{i+1} is finite, and let $r_i = |G_i : G_{i+1}|$ for any $i \in I$. Then every $g \in G$ can be expressed uniquely as

$$x_1^{e_{j1}} \cdots x_n^{e_{jn}},$$

where e_{jk} is an integer and $0 \leqslant e_{jk} < r_k$ if $k \in I$ (see, for instance, [6, Lemma 8.3]). Moreover, the polycyclic sequence X induces a set of conjugate relations and powers relations, which give rise to the so-called *(standard) polycyclic presentation* for G relative to X (see [6, Lemma 8.6]). The Word Problem in a polycyclic presentation can be solved effectively using the *collection to the left* (see [10, p. 395]). This method consists in rewriting an element $g \in G$ moving left to the beginning of the word all occurrences of x_1; next, all occurrences of x_2 are moved left until they are adjacent to the $x_1's$, and so on.

Following [10], we say that the sequence $U = (g_1, \dots, g_m)$ of elements of G is in standard form if the m-by-n matrix $A = (e_{jk})$ satisfies the following properties:

- all rows of A are nonzero (i.e., no g_j is the identity);
- A is in row Hermite normal form;
- if e_{jk} is the leading entry in the j-th row and $k \in I$, then e_{jk} divides r_k.

Suppose that U is in standard form. An admissible sequence of exponents for U is a sequence $(a_1, \dots, a_m)$ of integers such that, if e_{jk} is the leading entry in the j-th row and $k \in I$, then $0 \leqslant a_j < r_k/e_{jk}$. Let $S(U)$ be the set of all products $g_1^{a_1} \cdots g_m^{a_m}$, where $(a_1, \dots, a_m)$ is an admissible sequence of exponents for U. According to Proposition 5.2 of [10], $S(U)$ is a subgroup of G if and only if U is full, namely the following conditions hold:

- for $1 \leqslant j_1 < j_2 \leqslant m$ the set $S(U)$ contains $g_{j_1}^{-1} g_{j_2} g_{j_1}$;
- if e_{jk} is a leading entry of A and $k \in I$, then $S(U)$ contains g_j^q, where $q = r_k/e_{kj}$.

In particular, if U is full then $(g_1, \dots, g_m)$ is a polycyclic generating sequence for $S(U)$. We also recall that for any subgroup H of G there is a unique full sequence U of elements of G such that $H = S(U)$, and

this sequence can be determined by the POLY_SUBGROUP algorithm (see [10, Section 9.5] for more details).

The following result is taken from [10, Section 9.6].

Proposition 2.1 *Let* G *and* H *be polycyclic groups, and let* $x_1, \ldots, x_n$ *and* $y_1, \ldots, y_{n'}$ *be polycyclic sequences for* G *and* H*, respectively. Suppose further to know the polycyclic presentations of* G *and* H*. If* f *is a homomorphism from* G *to* H *then:*

(i) *there are* $h_1, \ldots, h_l \in H$ *such that the sequence* $U = (h_1, \ldots, h_l)$ *is full and* $S(U)$ *coincides with the image of* f*;*

(ii) *there are* $g_{l+1}, \ldots, g_m \in G$ *such that the sequence* $V = (g_{l+1}, \ldots, g_m)$ *is full and* $S(V)$ *coincides with the kernel of* f*;*

(iii) *given* $h = h_1^{a_1} \cdots h_l^{a_l} \in S(U)$ *there exist* $g_1, \ldots, g_l \in G$ *such that* $f(g) = h$*, where* $g = g_1^{a_1} \cdots g_l^{a_l}$*.*

We mention that the sequences U and V of Proposition 2.1 can be obtained with one application of the POLY_SUBGROUP algorithm. More precisely, assuming that K is the subgroup of the cartesian product $H \times G$ generated by the elements $f(x_1)x_1, \ldots, f(x_n)x_n$, the algorithm provides the unique full sequence $W = (w_1, \ldots, w_m)$ of elements of $H \times G$ such that $K = S(W)$, where m depends only on n. Then each w_j can be written uniquely as $h_j g_j$, where $h_j \in H$ and $g_j \in G$, and one can take l to be the largest integer in $\{1, \ldots, m\}$ such that h_l is not trivial.

3 The algorithm

In what follows we assume that, for a homomorphism f from a polycyclic group to a finitely generated abelian group, one can appeal to Proposition 2.1 for determining the image and the kernel of f.

Proposition 3.1 *Let* G *be a polycyclic group which has a nontrivial normal subgroup* H *such that* H *is nilpotent and* G/H *is finite.*

(i) *Let* x *and* y *be elements of* G*. Then there is an algorithm which decides if* x *and* y *are conjugate in* G *and, if so, finds* $g \in G$ *such that* $x = y^g$*.*

(ii) *Given an element* x *of* G*, there is an algorithm which determines a full sequence* U *such that* $S(U) = C_G(x)$*.*

Proof — Let n be the nilpotency class of H and consider the mapping

$$f_1 : h\gamma_2(H) \in H/\gamma_2(H) \mapsto [x,h]\gamma_2(H) \in H/\gamma_2(H).$$

where $\gamma_i(H)$ denotes the ith term of the lower central series of H. As $H/\gamma_2(H)$ is abelian, arguing modulo $\gamma_2(H)$, for any $h,k \in H$, we have

$$\begin{aligned}[x,h^{-1}k] &= [x,k][x,h^{-1}]^k = [x,k][x,h]^{-hk} \\ &= [x,h]^{-hk}[x,k] = [x,h]^{-1}[x,k].\end{aligned}$$

Therefore, if $h\gamma_2(H) = k\gamma_2(H)$, from the previous equation it follows that f_1 is well-defined. Similarly, $[x,hk] = [x,h][x,k]$, and f_1 is a homomorphism. For any $i \in \{1,\dots,n\}$, we use induction to define the homomorphism

$$f_i : h\gamma_{i+1}(H) \in L_i/\gamma_{i+1}(H) \mapsto [x,h]\gamma_{i+1}(H) \in \gamma_i(H)/\gamma_{i+1}(H),$$

where $L_1 = H$ and, for $i \geqslant 2$, L_i is the inverse image in L_{i-1} of the kernel of f_{i-1}.

(i) Let $T = \{t_1,\dots,t_r\}$ be a left transversal to H in G. Then x and y are conjugate in G if and only if there exists $g \in G = \cup_{j=1}^{r} t_j H$ such that $x = y^g$. This is equivalent to say that x and y^{t_j} are conjugate in H for some $j \in \{1,\dots,r\}$ or, equivalently, that

$$x^{-1}y^{t_j} \in f_n(L_n).$$

Indeed, $x^h = y^{t_j}$ for some $h \in H$ if and only if $[x,h] = x^{-1}y^{t_j}$. Using the subgroup membership problem (see [10, Section 9.5]), we can decide if $x^{-1}y^{t_j} \in f_n(L_n)$. Furthermore, in the affirmative case, we can apply (iii) of Proposition 2.1 to find an element $z \in L_n$ such that $f_n(z) = x^{-1}y^{t_j}$. Then $[x,z] = x^{-1}y^{t_j}$ yields $x^z = y^{t_j}$, showing that $t_j z^{-1}$ conjugates y into x.

(ii) For any $i \in \{1,\dots,n\}$, consider L_i and f_i as above. Thus the kernel of f_n is L_{n+1}, which coincides with $C_H(x)$. Let $T = \{t_1,\dots,t_r\}$ be a left transversal to H in G. Using (i), we can determine the set

$$S = \{j \in \{1,\dots,r\} \mid x = x^{t_j h_j} \text{ for some } h_j \in H\}.$$

Notice that, if $x = x^{t_j h_j} = x^{t_j h}$ for some $h_j, h \in H$, then $t_j(h_j h^{-1})t_j^{-1}$ belongs to $C_H(x)$. For any $j \in S$, taking an element $h_j \in H$ such

that $x = x^{t_j h_j}$, it follows that

$$C_G(x) = \langle C_H(x), t_j h_j \mid j \in S \rangle.$$

Finally, using the POLY_SUBGROUP algorithm, the full sequence U such that $S(U) = C_G(x)$ is determined, and we are done. □

Let $G = T(4, \mathbb{Z})$ be the group of upper triangular 4×4 matrices over the set $\mathbb{Z}$ of integers, and let $H = U(4, \mathbb{Z})$ be the subgroup of G consisting of all upper unitriangular matrices. Then H is normal in G [9, p. 128] and we have $H = \langle a_1, a_2, a_3, a_4, a_5, a_6 \rangle$, where

$$a_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad a_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$a_4 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad a_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad a_6 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Also, $\gamma_2(H) = \langle a_4, a_5, a_6 \rangle$, $\gamma_3(H) = \langle a_6 \rangle$ and $\gamma_4(H) = 1$ [9, Exercise 5.1.13]. So H is nilpotent of class 3. A left transversal to H in G is given by $T = \langle t_1, t_2, t_3, t_4 \rangle$, where

$$t_1 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad t_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$t_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad t_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

Obviously, $t_i^2 = 1$ for $i = 1, 2, 3, 4$. Thus $T = \langle t_1 \rangle \times \langle t_2 \rangle \times \langle t_3 \rangle \times \langle t_4 \rangle$ is an elementary abelian 2-group of order 16, and $G = H \rtimes T$ is the semidirect product of H and T. In particular, G is supersoluble.

Example 3.2 Let G and H as above, $x = t_1 a_1 a_6^{-1}$ and $y = t_1 a_1^{-1}$. According to (i) of Proposition 3.1, x and y are conjugate if and only

if $x^{-1}y^{t_1} \in f_3(L_3)$. Notice that t_1 inverts a_1, so $x^{-1}y^{t_1} = a_6$. Now consider the map

$$f_1 : h\gamma_2(H) \in H/\gamma_2(H) \mapsto [x,h]\gamma_2(H) \in H/\gamma_2(H).$$

Since

$$[x,a_1] = a_1^2, \quad [x,a_2] = a_4, \quad [x,a_3] = 1, \quad [x,a_4] = 1, \quad [x,a_5] = a_6,$$

the kernel of f_1 is generated by $a_2\gamma_2(H)$ and $a_3\gamma_2(H)$. Then

$$L_2 = \langle a_2, a_3, a_4, a_5, a_6\rangle.$$

Similarly, the kernel of the map

$$f_2 : h\gamma_2(H) \in L_2/\gamma_3(H) \mapsto [x,h]\gamma_3(H) \in g_2(H)/\gamma_3(H)$$

is generated by $a_3\gamma_3(H)$, $a_4\gamma_3(H)$ and $a_5\gamma_3(H)$. Hence

$$L_3 = \langle a_3, a_4, a_5, a_6\rangle.$$

Next, for

$$f_3 : h \in L_3 \to [x,h] \in \gamma_3(H),$$

we have $f_3(a_5) = a_6$. Thus x and y are conjugate, and $t_1a_5^{-1}$ is the conjugating element.

Example 3.3 Let G and H as above, and let $x = a_1a_3^2$. In Example 7.1 of [10], Sims applied his algorithm to compute

$$C_H(x) = \langle a_1, a_3, a_4a_5^2, a_6\rangle.$$

In order to determine $C_G(x)$, by (ii) of Proposition 3.1 it is enough to find $t \in T$ such that $x^{-1}x^t \in f_3(L_3) \leqslant \gamma_3(H) = \langle a_6\rangle$. Since

$$x^{-1}x^t = \begin{cases} 1 & \text{if } t = 1, t_1t_2, t_3t_4, t_1t_2t_3t_4 \\ a_1^{-2} & \text{if } t = t_1, t_2, t_1t_3t_4, t_2t_3t_4 \\ a_3^{-4} & \text{if } t = t_3, t_4, t_1t_2t_3, t_1t_2t_4 \\ a_1^{-2}a_3^{-4} & \text{if } t = t_1t_3, t_1t_4, t_2t_3, t_2t_4 \end{cases}$$

we obtain that

$$C_G(x) = \langle a_1, a_3, a_4 a_5^2, a_6, t_1 t_2, t_3 t_4 \rangle.$$

4 Conclusion

The security of the key agreement AAG relies on the good choice of the platform group, in which the multiplication and comparison of elements must be easy but the computation of conjugating elements is required to be difficult. The algorithmic properties of polycyclic groups suggest to use such a group as a platform. However, although the word problem can be solved in an efficient way in these groups, the conjugacy search problem is not so hard in some cases. For instance, Sims [10] provided an algorithm to solve the conjugacy search problem for finitely generated nilpotent groups, whose computational complexity was analyzed in [8]. In Proposition 3.1 we extended this algorithm to virtually nilpotent polycyclic groups, showing that AAG is inviable for this class of polycyclic groups, too. Nevertheless, it remains unclear whether it runs in polynomial time.

REFERENCES

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

[2] B. Eick – D. Kahrobaei: "Polycyclic groups: a new platform for cryptology"; *arXiv*:math/0411077 (2004).

[3] B. Eick – W. Nickel: "Polycyclic — computing with polycyclic groups" (2000), a GAP 4 package.

[4] B. Eick – G. Ostheimer: "On the orbit-stabilizer problem for integral matrix actions of polycyclic groups", *Math. Comp.* 72 (2003), 1511–1529.

[5] J. Gryak – D. Kahrobaei: "The status of polycyclic group-based cryptography: a survey and open problems", *Groups Complex. Cryptol.* 8, no. 2 (2016), 171–186.

[6] D.F. Holt – B. Eick – E.A. O'Brien: "Handbook of computational group theory", *CRC Press* (2005).

[7] J.C. Lennox – D.J.S. Robinson: "The Theory of Infinite Soluble Groups", *Clarendon Press*, Oxford (2004).

[8] J. Macdonald – A. Myasnikov – A. Nikolaev – S. Vassileva: "Logspace and compressed-word computations in nilpotent groups"; *arXiv*:1503.03888v2 (2015).

[9] D.J.S. Robinson: "A Course in the Theory of Groups", 2nd Eds., *Springer*, New York (1996).

[10] C. Sims: "Computation with Finitely Presented Groups", *Cambridge University Press*, Cambridge (1994).

[11] J. Wolf: "Growth of finitely generated solvable groups and curvature of Riemannian manifolds", *J. Differential Geom.* 2 (1968), 421–446.

Carmine Monetta
Dipartimento di Matematica, Università di Salerno
via Giovanni Paolo II, 132 - 84084 - Fisciano, Salerno (Italy)
e-mail: cmonetta@unisa.it

Antonio Tortora
Dipartimento di Matematica e Fisica
Università della Campania "Luigi Vanvitelli"
viale Lincoln, 5 - 81100 - Caserta (Italy)
e-mail: antonio.tortora@unicampania.it