Research Article

Matvei Kotov and Alexander Ushakov

Analysis of a certain polycyclic-group-based cryptosystem

Abstract: We investigate security properties of the Anshel–Anshel–Goldfeld commutator key-establishment protocol [1] used with certain polycyclic groups described by Eick and Kahrobaei [3]. We show that despite low success of the length based attack shown by Garber, Kahrobaei and Lam [5] the protocol can be broken by a deterministic polynomial-time algorithm.

Keywords: Cryptography, commutator-key establishment, conjugacy problem, polycyclic groups, metabelian groups

MSC 2010: 94A60, 68W30

Matvei Kotov, Alexander Ushakov: Department of Mathematics, Stevens Institute of Technology, Hoboken, NJ 07030, USA, e-mail: mkotov@stevens.edu, aushakov@stevens.edu

Communicated by: Spyros Magliveras

1 Introduction

In this paper we analyze the *commutator key-establishment* protocol [1] used with certain polycyclic groups described in [3]. The commutator key-establishment (CKE) protocol is a two-party protocol performed as follows.

- Fix a group G (called *the platform group*) and a set of generators $g_1, \dots, g_k$ for G. All this information is made public.
- Alice prepares a tuple of elements $\overline{a} = (a_1, \dots, a_{N_1})$ called *Alice's public tuple*. Each a_i is generated randomly as a product of g_i's and their inverses.
- Bob prepares a tuple of elements $\overline{b} = (b_1, \dots, b_{N_2})$ called *Bob's public tuple*. Each b_i is generated randomly as a product of g_i's and their inverses.
- Alice generates a random element A as a product $a_{s_1}^{\varepsilon_1} \dots a_{s_L}^{\varepsilon_L}$ of a_i's and their inverses. The element A (or more precisely its factorization) is called *Alice's private element*.
- Bob generates a random element B as a product $b_{t_1}^{\delta_1} \dots b_{t_L}^{\delta_L}$ of b_i's and their inverses, called *Bob's private element*.
- Alice publishes the tuple of conjugates $\overline{b}^A = (A^{-1} b_1 A, \dots, A^{-1} b_{N_2} A)$.
- Bob publishes the tuple of conjugates $\overline{a}^B = (B^{-1} a_1 B, \dots, B^{-1} a_{N_1} B)$.
- Finally, Alice computes the element K_A as a product:

$$A^{-1} \cdot (B^{-1} a_{s_1}^{\varepsilon_1} B \dots B^{-1} a_{s_L}^{\varepsilon_L} B)$$

  using the elements of Bob's conjugate tuple $\overline{a}^B$.
- Bob computes the key K_B as a product:

$$(A^{-1} b_{t_1}^{\delta_1} A \dots A^{-1} b_{t_L}^{\delta_L} A)^{-1} \cdot B$$

  using the elements of Alice's conjugate tuple $\overline{b}^A$.

It is easy to check that $K_A = K_B = A^{-1} B^{-1} A B$ in G. The obtained commutator is the *shared key*.

Security of the commutator key establishment protocol is based on computational hardness of computing the commutator $[A, B]$ based on the intercepted public information – the tuples $\overline{a}, \overline{b}$ and their conjugates $\overline{a}^B, \overline{b}^A$. In practice it is often achieved by solving systems of conjugacy equations for A and B, i.e., finding

$X = A'$ and $Y = B'$ satisfying:

$$\begin{cases} X^{-1}b_1X = b'_1, \\ \quad\vdots \\ X^{-1}b_{N_1}X = b'_{N_1}, \end{cases} \quad \text{and} \quad \begin{cases} Y^{-1}a_1Y = a'_1, \\ \quad\vdots \\ Y^{-1}a_{N_2}Y = a'_{N_1}, \end{cases}$$

and computing $K' = [A', B']$. In general it can happen that $K' \neq K$ as explained in [10], but as practice shows very often $K = K'$ (for instance, as in [6]).

A big advantage of the commutator key-establishment protocol over other group-based protocols is that it can be used with any group G satisfying certain computational properties. Originally, the group of braids B_n was suggested to be used as a platform group, but after a series of attacks it became clear that B_n can not provide good security. But the search for a good group is still very active and in [3] a certain class of polycyclic groups was proposed to be used with CKE. In this paper we show that this class can not provide good security. For more on group-based cryptography see [9].

Outline. In Section 2 we define the class of groups under investigation and discuss two different ways to represent the elements. In Sections 3 and 4 we describe the attacks on different group presentations.

2 The platform group

Consider an irreducible monic polynomial $f(x) \in \mathbb{Z}[x]$ and define a field

$$F = \mathbb{Q}[x]/(f).$$

The *ring of integers* of F is defined as

$$\mathcal{O}_F = \{a \in F \mid a \text{ is a zero of a monic polynomial } g(x) \in \mathbb{Z}[x]\}$$

and its *group of units* as

$$U_F = \{a \mid a^{-1} \in \mathcal{O}_F\}.$$

A semidirect product $U_F \ltimes \mathcal{O}_F$ of U_F and $\mathcal{O}_F$ is defined as a Cartesian product $U_F \times \mathcal{O}_F$ equipped with the following binary operation:

$$(\alpha, a) \cdot (\beta, b) = (\alpha\beta, a\beta + b). \tag{1}$$

The constructed group G_F is the platform group in [3]. It is easy to see that G_F is polycyclic and metabelian and there are several different ways to represent G_F.

(a) One can work with G_F as it is defined above, i.e., as a semidirect product, in which case its elements are represented as pairs and multiplication (1) is used.
(b) One can construct a polycyclic presentation for G_F and work with its elements as with words over the generating set.

Unfortunately, neither [3] nor [5] give any detail on how to treat G_F. Since computational properties of the same group can vary depending on a way we represent its elements, in the next sections we discuss both presentations of G_F.

2.1 G_F as a set of pairs of matrices

There are different ways to represent the elements of F. For instance, elements in F can be represented as polynomials over $\mathbb{Q}$ of degree up to $n-1$ with addition and multiplication performed modulo the original polynomial f. Also one can represent elements in F by matrices as described below. Recall that the *companion*

matrix for a monic polynomial $f = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0$ is a matrix of the form

$$M = \begin{bmatrix} 0 & 0 & \dots & 0 & -c_0 \\ 1 & 0 & \dots & 0 & -c_1 \\ 0 & 1 & \dots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -c_{n-1} \end{bmatrix}.$$

Denote by E the identity matrix. The characteristic and minimal polynomial of M is f and the set of matrices

$$F = \{a_0 E + a_1 M + a_2 M^2 + \cdots + a_{n-1} M^{n-1} \mid a_0, \dots, a_{n-1} \in \mathbb{Q}\} \tag{2}$$

equipped with the usual matrix addition and multiplication is a field. The correspondence between two presentations is obvious:

$$a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \longleftrightarrow a_0 + a_1 M + \cdots + a_{n-1} M^{n-1}.$$

Choosing a particular presentation, we do not change computational properties of F. Here we choose the matrix presentation for F.

Let $O_1, \dots, O_n$ be a basis of the ring of integers $\mathcal{O}_F$, where each O_i is a matrix. Hence

$$\mathcal{O}_F = \{a_1 O_1 + a_2 O_2 + \cdots + a_n O_n \mid a_1, \dots, a_n \in \mathbb{Z}\}.$$

Let $\{U_1, \dots, U_m\}$ be a generating set for the group U_F, where every U_i is a matrix. Hence

$$U_F = \{U_1^{a_1} \cdot U_2^{a_2} \cdot \ldots \cdot U_m^{a_m} \mid a_1, \dots, a_m \in \mathbb{Z}\}.$$

By the Dirichlet theorem [7, Chapter 8], we have $U_F \cong \mathbb{Z}_k \times \mathbb{Z}^{m-1}$, where $m = s + t - 1$, s is the number of real field monomorphisms $F \to \mathbb{R}$, and $2t$ is the number of complex field monomorphisms $F \to \mathbb{C}$. Without loss of generality it can be assumed that $U_1^k = E$.

Now naturally the group $G_F = U_F \ltimes \mathcal{O}_F$ is a set of pairs of matrices:

$$G = \{(C, S) \mid C \in U_F, S \in \mathcal{O}_F\},$$

equipped with multiplication given by

$$(C, S) \cdot (D, T) = (CD, SD + T).$$

It is easy to check that the inverse in $U_F \ltimes \mathcal{O}_F$ can be computed as

$$(C, S)^{-1} = (C^{-1}, -SC^{-1}),$$

which gives the following expression for the conjugate of (B, T) by (C, S):

$$(D, T)^{(C,S)} = (C, S)^{-1}(D, T)(C, S) = (D, S(E - D) + TC). \tag{3}$$

2.2 G_F given by polycyclic presentation

Recall that a group G is called polycyclic if there exists a subnormal series of G:

$$G = G_0 \rhd G_2 \rhd G_3 \rhd \cdots \rhd G_n = \{1\},$$

with cyclic factors G_{i-1}/G_i. Denote $[G_{i-1} : G_i]$ by r_i and put $I = \{i \mid r_i < \infty\}$. Relative to the series above one can find a generating set $g_1, \dots, g_n$ for G satisfying $\langle G_i, g_i \rangle = G_{i-1}$. Every element $g \in G$ can be uniquely expressed

as a product $g = g_1^{e_1} \dots g_n^{e_n}$, where $e_i \in \mathbb{Z}$, $i = 1, \dots, n$, and $0 \le e_i < r_i$ if $i \in I$. The polycyclic group G has a finite presentation of the form

$$G = \langle g_1, \dots, g_n \mid g_j^{g_i} = w_{ij},\ g_j^{g_i^{-1}} = v_{ij} \text{ for } 1 \le i < j \le n,\ g_k^{r_k} = u_k \text{ for } k \in I \rangle,$$

where w_{ij}, v_{ij}, and u_i are words in $g_{i+1}, \dots, g_n$. This presentation is called a *polycyclic presentation*. For more details see [7, Chapter 8].

It is straightforward to find a polycyclic presentation for the group $G_F = U_F \ltimes \mathcal{O}_F$. It has generators $g_1, \dots, g_m, g_{m+1}, \dots, g_{m+n}$, where $g_1, \dots, g_m$ correspond to the pairs $(U_1, O), \dots, (U_m, O) \in U_F \ltimes \mathcal{O}_F$ (O is the zero matrix), and $g_{m+1}, \dots, g_{m+n}$ correspond to the pairs $(E, O_1), \dots, (E, O_n) \in U_F \ltimes \mathcal{O}_F$. The set of relations for G is formed as follows.

- $g_{m+j}^{g_i} = g_{m+1}^{a_{ij1}} \dots g_{m+n}^{a_{ijn}}$, $i = 1, \dots, m$, $j = 1, \dots, n$, and $a_{ij1}, \dots, a_{ijn}$ are the coefficients in the expression $O_j U_i = a_{ij1} O_1 + \dots + a_{ijn} O_n$,
- $g_{m+j}^{g_i^{-1}} = g_{m+1}^{b_{ij1}} \dots g_{m+n}^{b_{ijn}}$, $i = 1, \dots, m$, $j = 1, \dots, n$, and $a_{ij1}, \dots, b_{ijn}$ are the coefficients in the expression $O_j U_i^{-1} = b_{ij1} O_1 + \dots + b_{ijn} O_n$,
- $g_1^k = e$,
- $[g_i, g_j] = e$, $1 \le i < j \le m$,
- $[g_i, g_j] = e$, $m + 1 \le i < j \le m + n$.

3 Attack on semidirect product

In this section we assume that the group G_F is given as a semidirect product and the field F is described using matrices as in (2). The general idea behind the attack is to extend the group G_F and work in $G_F^* = F^* \ltimes F$. The group G_F^* is, in general, not finitely generated and hence is not polycyclic. Nevertheless, the elements of G^* can be effectively represented by pairs of matrices as described in Section 2.1.

Consider a system of conjugacy equations related to Alice's private key:

$$\begin{cases} X^{-1} b_1 X = b_1', \\ \vdots \\ X^{-1} b_{N_2} X = b_{N_2}', \end{cases} \tag{4}$$

with unknown $X \in U_F \ltimes \mathcal{O}_F$. We treat the system as a system over $F^* \ltimes F$, and hence

$$X = (C, S),\ b_i = (B_i, T_i),\ b_i' = (B_i', T_i') \text{ in } F^* \ltimes F.$$

Using (3), we get the following system of N_2 linear equations over the field F with two unknowns C and S:

$$\begin{cases} S(E - B_1) + T_1 C = T_1', \\ \vdots \\ S(E - B_{N_2}) + T_{N_2} C = T_{N_2}'. \end{cases}$$

It has a unique solution when the coefficient matrix of the system has rank 2 over the field F, in which case the obtained solution A' is the same as the original private key of Alice. We call the described approach "field based attack" or simply *FBA*.

The described attack was implemented in GAP [4]. Its implementation can be found in [8]. Table 1 compares success rate and time efficiency of our attack and the attack in [5]. Our tests were run on an Intel Core i5 1.80GHz computer with 4GB of RAM, Ubuntu 12.04, GAP 4.7.

The first four columns of Table 1 are taken from [5]. For our tests we used the same parameter values: $N_1 = N_2 = 20$, and the same number of tests: 100.

Polynomial	$h(G)$	LBA w/ dynamic set, $L = 5$		FBA, $L = 5$		FBA, $L = 100$	
		Time	Success rate	Time	Success rate	Time	Success rate
$x^2 - x - 1$	3	0.20 h	100%	2.4 s	100%	2.8 s	100%
$x^5 - x^3 - 1$	7	76.87 h	35%	3.4 s	100%	5.3 s	100%
$x^7 - x^3 - 1$	10	94.43 h	8%	5.2 s	100%	9.7 s	100%
$x^9 - 7x^3 - 1$	14	95.18 h	5%	23.1 s	100%	57.7 s	100%
$x^{11} - x^3 - 1$	16	95.05 h	5%	15.3 s	100%	29.5 s	100%
$x^{15} - x - 2$	22	—	—	694.8 s	100%	607.4 s	100%
$x^{20} - x - 1$	30	—	—	208.5 s	100%	192.8 s	100%

Table 1. LBA and FBA efficacy comparison.

4 Attack on polycyclic presentation

In this section we assume that G_F is given by a polycyclic presentation described in Section 2.2. First we show that the group G_F can be presented as a semidirect product of an abelian matrix group and $\mathbb{Z}^n$. Then we present the attack on the obtained presentation.

4.1 Deduced semidirect product for G_F

Given a polycyclic presentation for G_F constructed in Section 2.2 it is straightforward to find the numbers m and n. For the relations

$$g_{m+j}^{g_i} = g_{m+1}^{a_{ij1}} \cdots g_{m+n}^{a_{ijn}}$$

we can define matrices $C_1, \dots, C_m$:

$$C_i = (a_{ijk})_{j=1,\dots,n}^{k=1,\dots,n}.$$

Next we form a semidirect product G of $\langle C_1, \dots, C_m \rangle$ and $\mathbb{Z}^n$ which is a set of pairs:

$$\{(C, \bar{s}) \mid C \in \langle C_1, \dots, C_m \rangle, \bar{s} \in \mathbb{Z}^n\}$$

equipped with the multiplication given by

$$(C, \bar{s}) \cdot (D, \bar{t}) = (CD, \bar{s}D + \bar{t}).$$

Let $\{\bar{e}_1, \dots, \bar{e}_n\}$ be the standard basis for $\mathbb{Z}^n$. It is easy to check that the map

$$\tau \colon \{g_1, \dots, g_{m+n}\} \to G, \quad \tau(g_i) = \begin{cases} (C_i, \bar{0}) & \text{if } i \le m, \\ (E, \bar{e}_j) & \text{if } i = m + j,\ 1 \le j \le n \end{cases}$$

defines an isomorphism between G_F and the constructed group. Furthermore, given an element $g = g_1^{e_1} \dots g_n^{e_n}$ it requires polynomial time to find its τ-image.

We also claim that given a pair $(C, \bar{v})$ it requires polynomial time to find a word g such that $\tau(g) = (C, \bar{v})$. To convert $(C, \bar{v})$ into a word in the generators $g_1, \dots, g_{m+n}$ one can express $(C, \bar{v})$ as a product:

$$(C, \bar{v}) = (C_1, \bar{0})^{a_1} \dots (C_m, \bar{0})^{a_m} (E, \bar{e}_1)^{a_{m+1}} \dots (E, \bar{e}_n)^{a_{m+n}},$$

for some $a_1, \dots, a_{n+m} \in \mathbb{Z}$, in which case $g = g_1^{a_1} \dots g_n^{a_n} g_{m+1}^{a_{m+1}} \dots g_{m+n}^{a_{m+n}}$. Clearly $(C, \bar{v}) = (C, \bar{0})(E, \bar{v})$. Therefore we have to solve two tasks. First, we need to find $a_1, \dots, a_m$ such that $C = C_1^{a_1} \dots C_m^{a_m}$ which can be done in polynomial time [2]. Second, we need to find $a_{m+1}, \dots, a_{m+n}$ such that $\bar{v} = a_{m+1}\bar{e}_1 + \dots + a_{m+n}\bar{e}_n$ which is obvious.

It follows from the discussion above that computational problems for G_F given by polycyclic presentation and by the deduced semidirect product are polynomial-time equivalent. Another important property of the computed presentation is that the ring

$$K = \mathbb{Q}[C_1, \dots, C_m]$$

Polynomial	$h(G)$	FBA2, $L = 5$		FBA2, $L = 100$	
		Time	Success rate	Time	Success rate
$x^2 - x - 1$	3	4.3 s	100%	3.9 s	100%
$x^5 - x^3 - 1$	7	4.9 s	100%	6.8 s	100%
$x^7 - x^3 - 1$	10	8.1 s	100%	10.1 s	100%
$x^9 - 7x^3 - 1$	14	34.0 s	100%	47.7 s	100%
$x^{11} - x^3 - 1$	16	20.9 s	100%	26.4 s	100%
$x^{15} - x - 2$	22	528.2 s	100%	761.3 s	100%
$x^{20} - x - 1$	30	164.6 s	100%	208.2 s	100%

Table 2. FBA2 efficacy.

generated by matrices $C_1, \dots, C_m$ is actually a field isomorphic to a subfield of F (because C_i's define the same action as U_i's, but in a basis $O_1, \dots, O_n$).

4.2 The attack

In the deduced presentation of G_F the system of conjugacy equations (4) is equivalent to the following system of equations with unknown $C \in K^*$ and $\overline{v} \in \mathbb{Z}^n$:

$$\begin{cases} \overline{v}(E - B_1) + \overline{t}_1 C = \overline{t}'_1, \\ \qquad\qquad \vdots \\ \overline{v}(E - B_{N_2}) + \overline{t}_{N_2} C = \overline{t}'_{N_2}, \end{cases} \tag{5}$$

where $(C, \overline{v})$ represents X, $(B, \overline{t}_i)$ represents b_i, $(B, \overline{t}'_i)$ represents b'_i for $i = 1, \dots, N_2$.

To solve system (5) we compute a basis $H_1, \dots, H_l$ of the field K as a vector space over $\mathbb{Q}$. Hence

$$C = c_1 H_1 + \cdots + c_l H_l$$

for some $c_1, \dots, c_n \in \mathbb{Q}$, and (5) can be rewritten as

$$\begin{cases} \overline{v}(E - B_1) + c_1 \overline{t}_1 H_1 + \cdots + c_l \overline{t}_1 H_l = \overline{t}'_1, \\ \qquad\qquad\qquad\qquad\qquad \vdots \\ \overline{v}(E - B_{N_2}) + c_1 \overline{t}_{N_2} H_1 + \cdots + c_l \overline{t}_{N_2} H_l = \overline{t}'_{N_2}, \end{cases}$$

which is a system of linear equations over field $\mathbb{Q}$ with unknown $\overline{v} = (v_1, \dots, v_n)$ and $c_1, \dots, c_l \in \mathbb{Q}$. The solution of this system provides us with the key A'.

We call this procedure *FBA2*. The attack was also implemented in GAP and tested on the same machine. Table 2 contains the results of our tests.

5 Conclusion

Our arguments show the following.

- The groups of the form $U_F \leftthreetimes \mathcal{O}_F$ can not be used as platform groups in the commutator key-establishment protocol.
- It is difficult to devise a successful length-based attack, and a low success rate does not mean much in terms of security.

Finally, we want to point out that our attack does not eliminate all polycyclic groups from consideration.

Funding: The second author has been partially supported by NSA Mathematical Sciences Program grant number H98230-14-1-0128.

References

[1] I. Anshel, M. Anshel and D. Goldfeld, An algebraic method for public-key cryptography, *Math. Res. Lett.* **6** (1999), no. 3–4, 287–291.
[2] L. Babai, R. Beals, J. Cai, G. Ivanyos and E. Luks, Multiplicative equations over commuting matrices, in: *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia (1996), 498–507.
[3] B. Eick and D. Kahrobaei, Polycyclic groups: A new platform for cryptology?, preprint (2004), http://arxiv.org/abs/math.GR/0411077.
[4] The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.7.7, (2015), www.gap-system.org.
[5] D. Garber, D. Kahrobaei and H. T. Lam, Length-based attacks in polycyclic groups, *J. Math. Crypt.* **9** (2015), 33–43.
[6] D. Hofheinz and R. Steinwandt, A practical attack on some braid group based cryptographic primitives, in: *Advances in Cryptology* (PKC 2003), Lecture Notes in Comput. Sci. 2567, Springer, Berlin (2003), 187–198.
[7] D. Holt, B. Eick and E. O'Brien, *Handbook of Computational Group Theory*, Chapman & Hall/CRC Press, Boca Raton, 2005.
[8] M. Kotov and A. Ushakov, Implementation of FBA, https://github.com/mkotov/polycyclic.
[9] A. G. Miasnikov, V. Shpilrain and A. Ushakov, *Non-Commutative Cryptography and Complexity of Group-Theoretic Problems*, Math. Surveys Monogr., American Mathematical Society, Providence, 2011.
[10] V. Shpilrain and A. Ushakov, The conjugacy search problem in public key cryptography: Unnecessary and insufficient, *Appl. Algebra Engrg. Comm. Comput.* **17** (2006), 285–289.

Received March 9, 2015; revised August 28, 2015; accepted September 7, 2015.