Research Article

Vitaliĭ Roman'kov*

A nonlinear decomposition attack

DOI: 10.1515/gcc-2016-0017
Received July 7, 2016

Abstract: This paper introduces a new type of attack, termed a nonlinear decomposition attack, against two known group-based key agreement protocols, namely, protocol based on extensions of (semi)groups by endomorphisms introduced by Kahrobaei, Shpilrain et al., and the noncommutative Diffie–Hellman protocol introduced by Ko, Lee et al. This attack works efficiently in the case when finitely generated nilpotent (more generally, polycyclic) groups are used as platforms. This attack is based on a deterministic algorithm that finds the secret shared key from the public data in both the protocols under consideration. Furthermore, we show that in this case one can break the schemes without solving the algorithmic problems on which the assumptions are based. The efficacy of the attack depends on the platform group, so it requires a more thorough analysis in each particular case.

Keywords: Group-based cryptography, key agreement protocol, (non)linear decomposition attack, polycyclic group, Diffie–Hellman protocol

MSC 2010: 20F10, 94A60

1 Introduction

In this paper we discuss a new attack, termed a *nonlinear decomposition* attack, on two known group-based cryptographic key agreement protocols. The original version of the first of the considered protocols, known as *protocol based on extensions of (semi)groups by endomorphisms*, was given in [13]. The second of the considered protocols, the so-called *noncommutative Diffie–Hellman protocol*, was introduced in [23].

The nonlinear decomposition attack can be efficiently applied to a key agreement protocol using as a platform group G for which the membership search problem is efficiently solvable, and one can efficiently construct finite generating sets for some subgroups associated with the protocol.

Recall that the *membership search problem* is as follows: given a group G and a subgroup H generated by $h_1, \dots, h_r$, and an element $g \in H$, find a group word $u(x_1, \dots, x_r)$ such that $g = u(h_1, \dots, h_r)$.

In particular, the proposed attack works in the case when a finitely generated nilpotent (more generally, polycyclic) group is used as platform for the protocol based on extensions of (semi)groups by endomorphisms, or for the noncommutative Diffie–Hellman protocol.

Note that in [22] polycyclic groups have been proposed as platforms for the protocol based on extensions of (semi)groups by endomorphisms. Several authors (see [5, 8, 10]) propose polycyclic groups as good platforms for different cryptographic protocols.

Each of the considered protocols produces a secret shared key. We show that one can solve the membership search problem for the subgroup of the platform group associated with the particular protocol to recover the shared key.

***Corresponding author: Vitaliĭ Roman'kov:** Omsk State University n.a. F. M. Dostoevskii, Omsk, Russia, e-mail: romankov48@mail.ru

The importance of different algorithmic and corresponding search problems, in particular the membership (search) problem, and algebraic properties of groups associated with the public data was emphasized in the monographs [31, 32] and in many papers. See, for instance, [24, 28, 29, 33, 41]. In many cases these problems can be formulated as questions about solvability of equations in groups (see, for instance, the survey [34]).

To put a mathematical base for the nonlinear decomposition attack against protocol based on the semidirect product as in [22], we show that for any endomorphism ϕ of a nilpotent (or, more generally, polycyclic) group and for any element g of the platform group G, the subgroup H generated by all $\phi^k(g)$ is generated by an efficiently computable finite set S of generators of the form $\phi^i(g)$. Using this observation, we show that to obtain a shared secret key in a protocol based on the semidirect product as in [22] one does not have to solve a "discrete-log-type" problem, but instead it is sufficient to solve the subgroup membership search problem with respect to the generating set S of the subgroup H. Therefore, if the set S is not too large and the subgroup membership search problem in G is efficiently solvable, then the shared secret key in the protocol in [22] can be recovered by an adversary efficiently.

A basic mathematical result for the attack against the noncommutative Diffie–Hellman protocol is as follows. Let g be an element of a polycyclic group G. Let A be a finitely generated subgroup of G. We show that there is an algorithm that finds a finite generating set T of the subgroup g^A generated by all elements of the form $g^a = aga^{-1}$, where $a \in A$. Then we show that we can efficiently find the shared secret key by solving the subgroup membership problem with respect to the generating set T of the subgroup g^A. Again, if the set T is not too large and the subgroup membership search problem in G is efficiently solvable, then the shared secret key in the noncommutative Diffie–Hellman protocol can be recovered by an adversary efficiently.

The original version of the protocol based on extensions of (semi)groups by endomorphisms was introduced in [13] and [20]. In the paper [20], the authors have employed matrices over group rings of a (small) symmetric group as platforms for this protocol. A linear decomposition attack was introduced by the author in [36]. Some applications of the introduced attack were given to show vulnerability of the series of protocols using automorphisms proposed by Mikhalev et al., Mahalanobis, Rososhek, and some other authors. The linear decomposition attack was developed in [35] and [30]. The protocol based on extensions of (semi)groups by endomorphisms was attacked in [30, 35, 38] by the linear decomposition attack. This was possible because the dimension of a linear representation of the platform (semi)group happens to be small enough in this case for the linear decomposition attack to be computationally feasible.

The basic scheme of the protocol based on extensions of (semi)groups by endomorphisms was improved in [22], where the different platform, the extension of a finite p-group of a special type by an endomorphism, was proposed. The basic subgroup of the platform group in this protocol is a free nilpotent p-group, for a sufficiently large prime p. A formal definition of this group will be given in Section 2. This is a finite group of exponent p^n for some fixed natural number n. Janusz [16] showed that a faithful representation of a finite p-group with at least one element of order p^n, as a group of matrices over a finite field of characteristic p, is of dimension at least $1 + p^{n-1}$. It can be too large to launch the linear decomposition attack provided p itself is large enough. The authors of [22] believed that this new offer of another platform should make the protocol invulnerable to the linear decomposition attacks of [30, 35, 38].

In this paper we show that the protocol using extensions of finite nilpotent groups by endomorphisms as platform groups can be broken by a deterministic algorithm, that we term a *nonlinear decomposition* attack. This attack works when the word and membership search problems can be solved by an efficient procedure in the platform group. In [22], elements of the platform group are written in the standard normal form. Then there is a deterministic algorithm to solve the word and membership search problems. Hence the nonlinear decomposition attack works in the case under consideration.

In [23], the original version of the noncommutative Diffie–Hellman protocol used the Artin braid groups B_n (n is a positive integer) as platform. Recall that, in [1], Anshel, Anshel and Goldfeld created a key exchange protocol (AAG protocol) that relied on solving the conjugacy search problem and proposed braid groups as the platform group. Other cryptographic protocols using the conjugacy search problem include [17–19].

Recall that the *conjugacy search problem* is as follows: given a group G and two conjugate elements g and f in G, find an element $x \in G$ such that $f = xgx^{-1}$.

The introduction of the AAG and noncommutative Diffie–Hellman protocols was followed by a stream of heuristic attacks. Hughes and Tannenbaum [15] emphasized the importance of choosing the correct length function originated the length-based attack on the AAG protocol. Later Garber, Kaplan, Teicher, Tsaban and Vishne [11, 12] gave several realizations of this approach. The authors of [33] gave an experimental evidence that the length-based attack can be modified to break the AAG protocol with a high rate of success. There are many other attacks to the AAG and noncommutative Diffie–Hellman protocols (see references in [10]).

In [6], Cheon and Jun proposed the first polynomial time probabilistic algorithm for attack to the noncommutative Diffie–Hellman protocol using the braid groups as platform. They used the efficient, faithful Lawrence–Krammer representation of the braid groups B_n by matrices. Also there is an efficient procedure that computes the preimages of matrices in B_n. Tsaban [43, 44] and Ben-Zvi, Kalka and Tsaban [3] presented a general approach for provable polynomial time solutions of computational problems in groups with efficient, faithful representation as matrix groups. Recall that the linear decomposition attack introduced and developed in [30, 35, 38], works when similar assumption is true. This attack in contrast with other approaches is absolutely deterministic. In view of these attacks a natural idea arises to use platforms that do not admit efficient, faithful linear representation, or the dimensions of such a representation is too large. It turned out that finitely generated nilpotent groups and, more generally, polycyclic groups have come to the fore in the past years as good candidates to use as platform groups.

In [8], Eick and Kahrobaei proposed polycyclic groups as secure platform for the AAG protocol. Later Garber, Kahrobaei and Lam [10] experimentally showed that polycyclic groups were resistant to many of the heuristic attacks that are strong against braid groups. In [5], Cavallo and Kahrobaei constructed polycyclic groups whose conjugacy search problem is at least as hard as the subset sum search which is well known to be NP-complete. Note that algebraic computations can be performed quickly when group elements are represented in the standard form as exponent vectors. It follows that there is a polynomial time algorithm that solves either conjugacy problem in these groups would imply P = NP.

In this paper we show that the noncommutative Diffie–Hellman protocol using polycyclic group as platform can be broken by a deterministic algorithm, the nonlinear decomposition attack, because the word and membership search problems can be solved by a deterministic algorithm in any polycyclic group. We suppose that elements of the platform group are written as exponent vectors corresponding to a chosen polycyclic base. See details in Section 3.

Further in the paper we denote by $\mathbb{N}$ the set of all natural numbers. Let G be a group. For two elements g and f of G we write $g^f = fgf^{-1}$ and $[g, f] = gfg^{-1}f^{-1}$ denoting conjugate and commutator, respectively. Then, inductively, let $[g_1, g_2, \dots, g_{k+1}] = [[g_1, g_2, \dots, g_k], g_{k+1}]$. By G' we denote the derived subgroup, generated by all commutators, and by $\gamma_k G$ ($k \in \mathbb{N}$) we denote the (normal) subgroup of G generated (as a subgroup) by all elements of the form $[g_1, g_2, \dots, g_k]$. In particular, $\gamma_2 G = G'$ This subgroup $\gamma_k G$ is the k-th member of the low central series of G. Also, G^k means the k-th power of G, i.e., the subgroup generated by all elements of the form g^k ($g \in G$). For $\phi \in \mathrm{End}(G)$ the expression $\phi(g)$ means the ϕ-image of $g \in G$.

1.1 Motivation

The purpose of this paper is two-fold. Firstly, we describe a new general attack on two known cryptographic schemes. The attack is based on algorithms that solve the membership search problem in the platform group. Secondly, we show that the security assumptions of the schemes under consideration do not hold in the case of a group in which the word and membership search problems are efficiently solvable. Indeed, we show how one can find the private keys without solving the underlying algorithmic problems.

In [20] (see also [13]), the key exchange scheme based on extension of a (semi)group by endomorphisms was introduced. Possible platforms that would make this scheme secure were discussed. As a good platform, the extension of a noncommutative semigroup of matrices by a conjugating automorphism has been proposed. It turned out this choice gives a vulnerable protocol to a linear decomposition attack offered in the further coming papers [30, 38] and the monograph [35]. A composition of a conjugating automorphism with a field automorphism proposed in [21] is also vulnerable to the linear decomposition attack (see [7]). This

general attack can be efficiently applied to many schemes and protocols that use as platforms parts of linear spaces (see [30, 35–37, 39]).

On the other hand, the (semi)group might be linear, but the dimensions of its linear representations could be so large that the dimension attacks become inefficient. The authors of [22] therefore offered another platform group that they believe should make the protocol to the linear decomposition attack. We describe their offer. Let F_r be the free group, freely generated by $x_1, \ldots, x_r$, and let $G = G(p, n) = F_r/F_r^{p^n}\gamma_{c+1}F_r$ be the free nilpotent group of rank r and exponent p^n (p prime, $n \in \mathbb{N}$). This group being a finitely generated nilpotent p-group, is finite. In [22], the group G was suggested as a platform for the proposed key exchange scheme. It was noted that, as any finite group, this group is linear, but Janusz [16] showed that a faithful representation of a finite p-group with at least one element of order p^n, as a group of matrices over a finite field of characteristic p is of dimension at least $1 + p^{n-1}$. It can be too large to launch the linear decomposition attack provided p itself is large enough. At the same time to keep computation in G efficient, the nilpotency class of the group has to be fairly small. It was noted in [22] that "for efficiency reasons, it seems better to keep c and r fairly small". In particular, it was suggested $c = 2$ or 3.

We are to show that under these new assumptions there exists another way to get the exchanged key without computing the private parameters (keys) of the protocol. We term this way a *nonlinear decomposition* attack. This attack works when the membership search problem is efficiently solvable for the platform group G. Also we suppose that every subgroup H of G is finitely generated, and the minimal number of generators of H is fairly small. We will show in our Section 2 that the proposed groups of the form $G(p, n)$ satisfy these properties. Hence, the nonlinear decomposition attack can be efficiently applied for the proposed protocol.

In [23], Ko, Lee, Cheon, Han, Kang and Park introduced the noncommutative Diffie–Hellman protocol, based on the conjugacy search problem for the platform group. They proposed the braid groups B_n ($n \in \mathbb{N}$) as platform for this protocol. It turned out that in the case when the platform is a part of a linear space of finite dimension, this protocol is vulnerable. It was shown firstly in [6], then in [3, 43, 44]. An efficient deterministic polynomial time algorithm computing the secret shared key of the noncommutative Diffie–Hellman protocol with the platform group embedded to a linear space was presented and developed in [35, 38] and [30]. As every group B_n admits efficient, faithful representation by matrices over a field and there is an efficient procedure to compute the preimages of matrices in B_n the proposed by Ko et al. version of the noncommutative Diffie–Hellman protocol is vulnerable.

Other versions of this protocol using different platform groups were proposed in many papers. See [5, 8, 10] and references in these papers. Polycyclic groups (in particular finitely generated nilpotent groups), finite or infinite, were proposed as a main candidate to be a platform group for the noncommutative Diffie–Hellman protocol. The polycyclic groups has good algorithmic theory (see [2]). Elements of polycyclic groups can be written in the normal form corresponding to polycyclic presentation. All group operations and the word problem in a consistent polycyclic presentation can be solved efficiently. We refer to [25, 42] for an analysis of strategies in polycyclic groups. In practice, such strategies and algorithms are known as an efficient method to solve the word problem in consistent polycyclic presentations. The method is implemented in GAP [45] and MAGMA [4] and it has proved to be practical for finite and infinite polycyclic groups.

In [2] it has been proven that there exists an algorithm to solve the conjugacy search problem in a polycyclic group. Another algorithm has been described in [9]. Unfortunately, it turned out that in a large polycyclic group G with a complex structure the solution of the conjugacy search problem is practically impossible. Indeed, the described in [9] algorithm uses induction down along a normal series with elementary or free abelian factors of G. This algorithm uses an orbit-stabilizer algorithm for finite groups, and methods from representation theory and algebraic number theory, such as the computation of submodule series and of unit groups, respectively. The complexity of the algorithm is very high. Thus, the polycyclic groups can be considered as good platform groups for protocols based on the conjugacy search problem.

On the other hand, the membership search problem can be solved efficiently in every polycyclic group given by the polycyclic presentation. See details below. Thus, the nonlinear decomposition attack can be applied to the discussed protocols using polycyclic groups as the platform groups.

1.2 Algorithms for polycyclic groups

A basic theory of polycyclic groups is presented in [40]. Note that it is well known (see for instance [14]) that every finitely generated nilpotent group, in particular every finite p-group, is polycyclic. A group G is said to be *polycyclic* if it has a subnormal series with cyclic quotients. Namely, G has subnormal subgroups $G_0 = G, G_1, \dots, G_q = \{1\}$ such that

$$\{1\} = G_q \lhd G_{q-1} \lhd \cdots \lhd G_0 = G \tag{1.1}$$

and the quotient G_i/G_{i+1} is a cyclic group. A series of the form (1.1) is called *polycyclic of length* q. Let $g_i \in G_i$ $(i = 0, \dots, q-1)$ be a preimage of any generating element $\bar{g}_i$ of the quotient G_i/G_{i+1} $(i = 0, \dots, q-1)$. Then every element $g \in G$ can be written in the form

$$g = \prod_{i=0}^{q-1} g_i^{c_i}, \quad c_i \in \mathbb{Z}. \tag{1.2}$$

Moreover, the presentation (1.2) is unique under the restriction $0 \le c_i \le e_i - 1$ for every i such that the order $e_i = |\bar{g}_i|$ of $\bar{g}_i$ is finite in G_i/G_{i+1}. Clearly, G is generated by the elements $g_0, \dots, g_{q-1}$. We call the set $\{g_0, \dots, g_{q-1}\}$ the *polycyclic base* of G and (1.2) the *normal form* of g.

In general (see [26]), we say that G has a *polycyclic presentation* in generators $g_0, \dots, g_{q-1}$ if the defining relations in this presentation are of the following types:

- $g_j^{g_i} = u_{ij}$ and $g_j^{g_i^{-1}} = v_{ij}$, where $0 \le i < j \le q-1$ and $u_{ij}, v_{ij} \in \mathrm{gp}(g_{i+1}, \dots, g_{q-1})$,
- $g_i^{e_i} = w_i$, where $0 < e_i < \infty$ for $i = 0, \dots, q-1$ and $w_i \in \mathrm{gp}(g_{i+1}, \dots, q_{q-1})$.

If we put $G_j = \mathrm{gp}(g_j, \dots, g_{q-1})$, then G has the polycyclic series (1.1) with polycyclic base $\{g_0, \dots, g_{q-1}\}$. A group G has a polycyclic presentation if and only if it is polycyclic (see [26, Statement 9.2.7]).

Every polycyclic group admits a more sophisticated presentation called *consistent polycyclic*. Let G be a polycyclic group with the polycyclic presentation specified above. By omitting the generators $g_0, \dots, g_{i-1}$ and all relations involving them, we obtain a subpresentation presenting a group L_i in generators $g_i, \dots, g_{q-1}$. There is a natural surjective homomorphism $L_i \to G_i$, where $G_i = \mathrm{gp}(g_i, \dots, g_{q-1})$, and also a natural homomorphism $L_i \to L_{i-1}$. It is shown in [26, Statement 9.2.8] that the following assertions are equivalent:

$$\begin{gathered} \text{the maps } L_i \to L_{i-1} \text{ are injective,} \\ \text{the maps } L_i \to G_i \text{ are isomorphisms,} \\ L_i \simeq G_i \text{ for each } i. \end{gathered} \tag{1.3}$$

A polycyclic presentation satisfying the equivalent conditions (1.3) is called *consistent*. It is shown in [26, Statement 9.2.7] that a consistent polycyclic presentation can be constructed for every polycyclic group. Hence every polycyclic group has a consistent polycyclic presentation. The polycyclic presentations are used as a basis for computations with polycyclic groups. We refer to [26] and [42] for background and a more detailed introduction to polycyclic presentations.

Let G be a polycyclic group given by a polycyclic presentation. Then there is an algorithm that computes the unique normal form for an arbitrary word in the generators. This algorithm uses the so-called collection process and is efficient, see [42]. It follows that the word problem in a polycyclic presentation can be solved efficiently. The basic idea of the collection process is that it applies iteratively the power and conjugate relations of the given presentation to subwords of a given word and thus it modifies the given word. The nature of the relations asserts that an iteration of this process will eventually produce a word in the normal form. The efficiency of the collection algorithm depends critically on the sequence of chosen subwords which are processed. There are various strategies which have been investigated for this purpose. We refer to [25] for an analysis of strategies in finite polycyclic groups. The resulting complexities of the methods depend on the growth of the exponents of generators occurring in intermediate stages of the algorithm. This growth can be bounded above if the considered group is finite. In infinite polycyclic groups there is the potential risk of an integer explosion inherent in the collection algorithm. In practise, collection is known as an efficient method to find normal forms of elements in polycyclic presentations. The method is implemented in GAP [45] and MAGMA [4] and it has proved to be practical for finite and infinite polycyclic groups.

Suppose that G admits a polycyclic series of length q. Every subgroup H of G admits a polycyclic series

$$\{1\} = H_r \lhd H_{r-1} \lhd \cdots \lhd H_0 = H \tag{1.4}$$

of length $r \leq q$. One can get this series as intersection of H with the members of the polycyclic series of G and deleting members corresponding to trivial quotients. Here $H_i = H \cap G_{f_i}$ for some sequence $f_0 < f_1 < \cdots < f_r$. It follows that H admits $r \leq q$ generating elements $h_1, \ldots, h_r$ such that the image $\bar{h}_i$ of each $h_i \in H_i$ in $G_{f_i}/G_{f_{i+1}}$ is equal to $\bar{g}_{f_i}^{m_i}$, $m_i \in \mathbb{N}$. The set $\{h_1, \ldots, h_r\}$ is a polycyclic base of H corresponding to the polycyclic base $\{g_0, \ldots, g_{q-1}\}$ of G.

Polycyclic groups are a natural class with good algorithmic theory. Indeed, the word problem, the membership problem and the conjugacy problem all have positive solutions for this class (see, e.g., [2, 26, 42]). At the same time new cryptosystems are proposed on polycyclic groups (see, e.g., [5, 8, 18, 27]). These cryptosystems are based on the fact that some algorithmic problems can be solved efficiently in polycyclic groups, while the known solutions to other problems are far less efficient. For instance, in [8], a new cryptosystem is based on the fact that the word problem can be solved efficiently in polycyclic groups, while the conjugacy problem has no known efficient solutions.

Efficacy of the membership search algorithm for polycyclic group. Let G be a polycyclic group with polycyclic base $\{g_0, \ldots, g_{q-1}\}$ corresponding to the polycyclic series (1.1). The membership search problem for a subgroup H can be efficiently solved as follows. Let $\{h_1, \ldots, h_r\}$ be a polycyclic base of H corresponding to the polycyclic base $\{g_0, \ldots, g_{q-1}\}$ of G. Let h be an element of H. Consider the image $\bar{h}$ of h in the cyclic quotient $G_0/G_1 = \mathrm{gp}(\bar{g}_0)$. Obviously, the quotient G_0/G_1 can be efficiently constructed. Then $h \in G_1$ or there are element h_0 in the polycyclic base of H and exponent s_1 such that $h' = h_0^{s_1} h \in G_1$. This exponent s_1 can be obtain efficiently by standard procedure for cyclic groups. We write h' in the normal form corresponding to the polycyclic base $\{g_1, \ldots, g_{q-1}\}$ and continue similar computations with respect to polycyclic group G_1, that has a polycyclic series of smaller length, and its subgroup $H \cap G_1$. As a final result we obtain the normal form of h, and so solve the membership search problem.

2 Key exchange protocol using semidirect product

Kahrobaei, Koupparis and Shpilrain [20] (see also [13] and [22]) proposed a group-based key exchanged scheme using semidirect product of (semi)groups. It works as follows.

Let G be a (semi)group. An element $g \in G$ is chosen and made public as well as an arbitrary automorphism $\phi \in \mathrm{Aut}(G)$ (or an arbitrary endomorphism $\phi \in \mathrm{End}(G)$). Let $H = G \rtimes \mathrm{gp}(\phi)$ be the semidirect product of G by $\mathrm{gp}(\phi)$.

Algorithm. Suppose that two correspondents, Alice and Bob, want to share a key. Both Alice and Bob are going to work with elements of H of the form (ϕ^l, g), where $g \in G$, $l \in \mathbb{N}$. Note that two elements of this form are multiplied as follows: $(\phi^l, g) \cdot (\phi^t, h) = (\phi^{l+t}, \phi^t(g) \cdot h)$. They act as follows.

(1) Alice picks a private number $m \in \mathbb{N}$. Then she computes

$$(\phi, g)^m = (\phi^m, \phi^{m-1}(g) \cdot \cdots \cdot \phi^2(g) \cdot \phi(g) \cdot g)$$

and sends only the second component of this pair to Bob. Thus, Alice sends to Bob only the element

$$g_m = \phi^{m-1}(g) \cdot \cdots \cdot \phi^2(g) \cdot \phi(g) \cdot g$$

of the (semi)group G.

(2) Bob picks a private $n \in \mathbb{N}$. Then he computes

$$(\phi, g)^n = (\phi^n, \phi^{n-1}(g) \cdot \cdots \cdot \phi^2(g) \cdot \phi(g) \cdot g)$$

and sends only the second component of this pair to Alice. Thus, Bob sends to Alice only the element

$$g_n = \phi^{n-1}(g) \cdot \cdots \cdot \phi^2(g) \cdot \phi(g) \cdot g$$

of the (semi)group G.

(3) Alice computes

$$(x, g_n) \cdot (\phi^m, g_m) = (x \cdot \phi^m, \phi^m(g_n) \cdot g_m).$$

Her key is now $K_A = \phi^m(g_n) \cdot g_m$. Note that she does not actually "compute" $x \cdot \phi^m$ because she does not know the automorphism $x = \phi^n$; recall that it was not transmitted to her. But she does not need it to compute K_A.

(4) Bob computes

$$(y, g_m) \cdot (\phi^n, g_n) = (y \cdot \phi^n, \phi^n(g_m) \cdot g_n).$$

His key is now $K_B = \phi^n(g_m) \cdot g_n$. Again, Bob does not actually "compute" $y \cdot \phi^n$ because he does not know the automorphism $y = \phi^m$.

(5) Since

$$(\phi^n, g_n) \cdot (\phi^m, g_m) = (\phi^m, g_m) \cdot (\phi^n, g_n) = (\phi, g)^{m+n} = (\phi^{m+n}, g_{m+n}),$$

Alice and Bob have $K_A = K_B = K$, the shared secret key.

Cryptanalysis. We introduce a general attack termed *nonlinear decomposition* attack, which can be applied to the just described protocol to find the exchanged key without computing of security parameters m and n. We suppose that G is group, and the word and membership search problems are efficiently solvable for G.

We are to describe the mathematical idea behind the this version of nonlinear decomposition attack.

Lemma 2.1 (Finding a generating set). *Let G be a group, $g \in G$ and $\phi \in \mathrm{End}(G)$. Let $H = \mathrm{gp}(g_k : k \in \mathbb{N})$, where $g_k = \phi^{k-1}(g) \cdots \phi^2(g) \cdot \phi(g) \cdot g$. Suppose that H is finitely generated and the membership problem is solvable for G. Then there is an algorithm that finds a finite generating set of the subgroup H in the form $\{g_1, \dots, g_t\}$ (or equivalently, in the form $\{g, \phi(g), \dots, \phi^{t-1}(g)\}$).*

Proof. For $k \in \mathbb{N}$, denote $H_k = \mathrm{gp}(g_1, \dots, g_k) = \mathrm{gp}(g, \phi(g), \dots, \phi^{k-1}(g))$. By our assumption, one can efficiently find the minimal t such that $\phi^t(g) \in H_t$. Suppose $H_1 < H_2 < \cdots < H_t$ and $\phi^t(g) \in H_t$, and thus $H_t = H_{t+1}$. Then there is a group word $u(z_1, \dots, z_t)$ such that $\phi^t(g) = u(g, \phi(g), \dots, \phi^{t-1}(g))$. It follows that

$$\begin{aligned}\phi^{t+1}(g) &= u(\phi(g), \dots, \phi^{t-1}(g), \phi^t(g)) \\ &= u(\phi(g), \dots, \phi^{t-1}(g), u(g, \phi(g), \dots, \phi^{t-1}(g))) \\ &= u_1(g, \phi(g), \dots, \phi^{t-1}(g))\end{aligned}$$

for some other group word $u_1(z_1, \dots, z_t)$. Hence $\phi^{t+1}(g) \in H_t$. Continuing this process, we prove that $\phi^{t+l}(g) \in H_t$ for every $l \in \mathbb{N}$, and thus $H_{t+l} = H_t$. Hence $H = H_t$. □

Remark 2.2. Let G be finitely generated nilpotent or, more generally, polycyclic group. Suppose that G admits a polycyclic series of length q. Then every subgroup H of G admits a polycyclic series of length $r \le q$. Thus every subgroup H of G can be generated by $r \le q$ elements. The groups suggested as platforms in [22] are of the order p^n, so they have polycyclic series of length n with cyclic factors of order p. Then every subgroup H is generated by $r \le n$ elements.

Suppose that all assumptions of Lemma 2.1 are satisfied, and $H = \mathrm{gp}(g_1, \dots, g_t)$. Then one can find a presentation of Alice's element g_m in the form

$$g_m = \prod_{j=1}^{k} g_{i_j}^{\epsilon_j}, \quad i_j \in \{1, \dots, t\},\ \epsilon_j = \pm 1,\ j = 1, \dots, k.$$

As $\phi^{i_j}(g_n) \cdot g_{i_j} = \phi^n(g_{i_j}) \cdot g_n$, we have

$$\begin{aligned}K = g_{m+n} = \phi^n(g_m) \cdot g_n = \phi^n\left(\prod_{j=1}^{k} g_{i_j}^{\epsilon_j}\right) \cdot g_n = \prod_{j=1}^{k} \phi^n(g_{i_j})^{\epsilon_j} \cdot g_n \\ = \prod_{j=1}^{k} (\phi^n(g_{i_j}) \cdot g_n \cdot g_n^{-1})^{\epsilon_j} \cdot g_n = \prod_{j=1}^{k} (\phi^{i_j}(g_n) \cdot g_{i_j} \cdot g_n^{-1})^{\epsilon_j} \cdot g_n.\end{aligned}$$

All elements g_n, g_{i_j}, ϵ_j and ϕ^{i_j} for $i_j \in \{1, \dots, t\}$ in (1.2) are known. Hence, K is determined. Note that we did not compute m, n, ϕ^m or ϕ^n. In other words, we did not solve the algorithmic problems on which the assumptions of the considered scheme are based.

Efficacy of the nonlinear decomposition attack to the key exchange protocol using semidirect product. Let F_r be the free group, freely generated by $x_1, \dots, x_r$, and let $G = F_r/F_r^{p^n}\gamma_{c+1}F_r$ be the free nilpotent group of rank r and exponent p^n (p prime). This group being a finitely generated nilpotent p-group, is finite. In [22], the group G was suggested as a platform for the proposed key exchange scheme. It was noted in [22] that "for efficiency reasons, it seems better to keep c and r fairly small". In particular, it was suggested $c = 2$ or 3. It was noted in [22] that the parameters p and r have to be chosen carefully to provide for both security and efficiency. It was also shown that there is a normal form of the elements of G, and all operations used in the scheme are efficient and can be done in polynomial time. See [22] for details.

On the other hand, to provide an efficient nonlinear decomposition attack one needs in efficient solution of the membership search problem for G. Let H be a subgroup of G generated by elements $h_1, \dots, h_k$. Let $\alpha_i : G \to G/\gamma_i G$ ($i = 1, \dots, c$) be the standard homomorphism. For any element $h \in H$, we find a group word $u_1(z_1, \dots, z_k)$ such that $\alpha_2(h) = u_1(\alpha_2(h_1), \dots, \alpha_2(h_k))$, i.e., we solve the membership search problem for an elementary abelian group $G/\gamma_2 G$ of the exponent p^n. In other words, we solve the set of linear equations over the modular ring $\mathbb{Z}_{p^n}$. It can be done by a deterministic polynomial time algorithm. Then we set $h^{(1)} = u_1(h_1, \dots, h_k)^{-1}h$. Now we reduce our membership search problem for the element h with respect to subgroup H to similar problem for the element $h^{(1)}$ with respect to subgroup

$$H^{(1)} = H \cap \gamma_2 G = \mathrm{gp}(h_1^{(1)}, \dots, h_{k_1}^{(1)}),$$

where $h_1^{(1)}, \dots, h_{k_1}^{(1)}$ is a generating set of $H^{(1)}$. Then we find a group word $u_2(z_1, \dots, z_{k_1})$ such that

$$\alpha_3(h^{(1)}) = u_2(\alpha_3(h_1^{(1)}), \dots, \alpha_3(h_{k_1}^{(1)})),$$

i.e., we solve the membership search problem for a finite abelian group $G'/\gamma_3 G$. Then we set

$$h^{(2)} = u_2(h_1^{(1)}, \dots, h_{k_1}^{(1)})^{-1}h^{(1)},$$

and get $h^{(2)} \in H^{(2)}$, where $H^{(2)} = H \cap \gamma_3 G$. Iterating this process yields a presentation of h as a group word in $h_1, \dots, h_t$ eventually.

3 Key exchange protocol using conjugation

The noncommutative Diffie–Hellman key exchange protocol was introduced in [23].

Algorithm. Let G be a noncommutative group with solvable word problem. Let $g \in G$ and let A and B be two subgroups of G such that $[a, b] = 1$ for every pair of elements $a \in A$ and $b \in B$. We suppose that A is generated by elements $a_1, \dots, a_k$ and B is generated by elements $b_1, \dots, b_l$. The group G, its elements $a_1, \dots, a_k$ and $b_1, \dots, b_l$ are public information. Suppose that two correspondents, Alice and Bob, want to share a secret key. They act as follows.

(1) Alice picks a private element $a \in A$. Then she computes g^a and sends this element to Bob.
(2) Bob picks a private element $b \in B$. Then he computes g^b and sends this element to Alice.
(3) Alice computes $(g^b)^a = g^{ab}$. Her key is now $K_A = g^{ab}$.
(4) Bob computes $(g^a)^b = g^{ba}$. His key is now $K_B = g^{ba}$.
(5) Since $ab = ba$, Alice and Bob have $K_A = K_B = K = g^{ab}$, the shared secret key.

Cryptanalysis. We apply the nonlinear decomposition attack to find the shared key K without computing of security parameters a and b.

We are to describe the mathematical idea behind the nonlinear decomposition attacks in this particular case.

Lemma 3.1 (Finding a generating set). *Let G be a group and $g \in G$. Suppose that the word and membership search problems are solvable for G. Also suppose that every subgroup is finitely generated in G. Then there is an algorithm that finds a finite generating set of the subgroup g^A generated by all elements of the form g^a, where $a \in A$.*

Proof. Let $L_0 = \{g\}$, and let M_i be the set of all elements g^a, where a is written as a group word in elements $a_1, \dots, a_k$ of length i ($i \in \mathbb{N}$). We set some order for every M_i. Now $M_1 = b_1, \dots, b_{2k}$ is an ordered set. Adding to L_0 one by one elements b_j and checking every time whether or not element b_j lies in $\mathrm{gp}(g, b_1, \dots, b_{j-1})$, one can efficiently construct a subset $L_1 = \{g, b_{j_1}, \dots, b_{j_{t_1}} : j_1 < \cdots < j_{t_1}\}$ of the set $L_0 \cup M_1$ which extends the set L_0 such that for every l one has $b_{j_{l+1}} \notin \mathrm{gp}(g, b_{j_1}, \dots, b_{j_l})$.

Notice that $g^A = \mathrm{gp}(L_1, \bigcup_{i=2}^{\infty} M_i)$. It follows that if $L_0 = L_1$, then $g^A = \mathrm{gp}(L_0)$. An argument is similar to the argument used in the proof of Lemma 2.1. If $L_0 \neq L_1$, then we continue the procedure for L_1 and find a subset L_2 of $L_1 \cup M_2$ extending L_1 such that every its element does not lie in the subgroup generated by all previous elements. Then $g^A = \mathrm{gp}(L_2, \bigcup_{i=3}^{\infty} M_i)$. Keep going one constructs a sequence of strictly increasing subsets $L_0 < L_1 < \cdots < L_t$ of G. Since every subgroup of G is finitely generated, the sequence stabilizes for some t, i.e., $L_t = L_{t+1}$. In this case L_t is a generating set of g^A and its elements are in the required form. □

By Remark 2.2, we can bound the number of generating elements of g^A in Lemma 3.1 in the case G is a finite polycyclic group.

Suppose that the assumptions of Lemma 3.1 are satisfied. The we can efficiently obtain a subset $\{c_1, \dots, c_s\}$ of A such that $g^A = \mathrm{gp}(g^{c_1}, \dots, g^{c_s})$. Also suppose that the word and membership search problems are efficiently solvable for G. Then we can find a presentation of the Alice's element g^a in the form

$$g^a = u(g^{c_1}, \dots, g^{c_s}), \quad \text{where } u \text{ is a group word.}$$

Then, since $bc_i = c_i b$ for $i = 1, \dots, s$, one has

$$u((g^b)^{c_1}, \dots, (g^b)^{c_s}) = u(g^{c_1}, \dots, g^{c_s})^b = g^{ab} = K.$$

All elements g^b and $c_1, \dots, c_s$ in (1.4) are known. Hence, K is determined. Note that we did not compute a or b. In other words, we did not solve the algorithmic problem on which the assumptions of the considered scheme are based.

Efficacy of the nonlinear decomposition attack to the noncommutative Diffie–Hellman protocol. Suppose that a finite polycyclic group G is chosen as a platform for the noncommutative Diffie–Hellman protocol. Let G has a polycyclic base $g_0, \dots, g_{q-1}$ corresponding to the subnormal series (1.1) with cyclic factors of orders $e_0, \dots, e_{q-1}$, respectively. Then $|G| = \prod_{i=0}^{q-1} e_i$. Every element $g \in G$ can be uniquely written in the normal form (1.2), where $0 \leq c_i \leq e_i - 1$ for $i = 0, \dots, q-1$. This form can be efficiently constructed for any given word g in generating elements of G.

Every subgroup H of G has the corresponding polycyclic base $h_1, \dots, h_r$. We explained in Section 1 that under these assumptions the membership search problem can be efficiently solved for G with respect to H. Hence, the nonlinear decomposition attack to the noncommutative Diffie–Hellman protocol in the considering case can be done efficiently.

Funding: This research was supported by Russian Science Foundation (project 16-11-10002).

References

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

[2] G. Baumslag, F. B. Cannonito, D. J. S. Robinson and D. Segal, The algorithmic theory of polycyclic-by-finite groups, *J. Algebra* **142** (1991), 118–149.

[3] A. Ben-Zvi, A. Kalka and B. Tsaban, Cryptanalysis via algebraic spans, preprint (2014), https://eprint.iacr.org/2014/041.

[4] W. Bosma, J. Cannon and C. Playoust, The MAGMA algebra system I: The user language, *J. Symbolic Comput.* **24** (1997), 235–265.
[5] B. Cavallo and D. Kahrobaei, A family of polycyclic groups over which the conjugacy problem is NP-complete, preprint (2014), https://arxiv.org/abs/1403.4153v2.
[6] J. Cheon and B. Jun, A polynomial time algorithm for the braid Diffie–Hellman conjugacy problem, in: *Advances in Cryptology* (CRYPTO 2003), Lecture Notes in Comput. Sci. 2729, Springer, Berlin (2003), 212–225.
[7] J. Ding, A. D. Miasnikov and A. Ushakov, A linear attack on a key exchange protocol using extensions of matrix semigroups, preprint (2015), https://eprint.iacr.org/2015/018.
[8] B. Eick and D. Kahrobaei, Polycyclic groups: A new platform for cryptology?, preprint (2004), https://arxiv.org/abs/math/0411077.
[9] B. Eick and G. Ostheimer, On the orbit-stabilizer problem for integral matrix actions of polycyclic groups, *Math. Comp.* **72** (2003), 1511–1529.
[10] D. Garber, D. Kahrobaei and H. T. Lam, Analysing the length-based attack on polycyclic groups, preprint (2013), https://arxiv.org/abs/1305.0548v1.
[11] D. Garber, S. Kaplan, M. Teicher, B. Tsaban and U. Vishne, Probabilistic solutions of equations in the braid group, *Adv. Appl. Math.* **35** (2005), 323–334.
[12] D. Garber, S. Kaplan, M. Teicher, B. Tsaban and U. Vishne, Length-based conjugacy search in the braid group, in: *Algebraic Methods in Cryptography* (Mainz/Bochum 2005), Contemp. Math. 418, American Mathematical Society, Providence (2006), 75–87.
[13] M. Habeeb, D. Kahrobaei, C. Koupparis and V. Shpilrain, Public key exchange using semidirect product of (semi)groups, preprint (2013), https://arxiv.org/abs/1304.6572v1.
[14] P. Hall, *Edmonton Notes on Nilpotent Groups*, Queen Mary College Math. Notes, Queen Mary College, London, 1969.
[15] J. Hughes and A. Tannenbaum, Length-based attacks for certain group-based encryption rewriting systems, Workshop SECI02 Securite de la Communication sur Internet (2002).
[16] G. J. Janusz, Faithful representations of p-groups at characteristic p, *J. Algebra* **15** (1970), 335–351.
[17] D. Kahrobaei and M. Anshel, Decision and search in non-abelian Cramer–Shoup public key cryptosystem, *Groups Complex. Cryptol.* **1** (2009), 217–225.
[18] D. Kahrobaei and B. Khan, A non-commutative generalization of ElGamal key exchange using polycyclic groups, in: *Global Telecommunications Conference 2006* (GLOBECOM '06), IEEE Press, Piscataway (2006), DOI 10.1109/GLOCOM.2006.290.
[19] D. Kahrobaei and C. Koupparis, Non-commutative digital structures using non-commutative groups, *Groups Complex. Cryptol.* **4** (2012), 377–384.
[20] D. Kahrobaei, C. Koupparis and V. Shpilrain, Key exchange using semidirect product of (semi)groups, in: *Applied Cryptography and Network Security* (ACNS 2013), Lecture Notes in Comput. Sci. 7954, Springer, Berlin (2013), 475–486.
[21] D. Kahrobaei, H. Lam and V. Shpilrain, Public key exchange using extensions by endomorphisms and matrices over a Galois field, preprint, http://www.sci.ccny.cuny.edu/~shpil/res.html.
[22] D. Kahrobaei and V. Shpilrain, Using semidirect product of (semi)groups in public key cryptography, preprint (2016), https://arxiv.org/abs/1604.05542v1.
[23] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang and C. Park, New public-key cryptosystem using braid groups, in: *Advances in Cryptology* (CRYPTO 2000), Lecture Notes in Comput. Sci. 1880, Springer, Berlin (2000), 166–183.
[24] M. Kreuzer, A. D. Myasnikov and A. Ushakov, A linear algebra attack to group-ring-based key exchange protocols, in: *Applied Cryptography and Network Security* (ACNS 2014), Lecture Notes in Comput. Sci. 8479, Springer, Berlin (2014), 37–43; https://eprint.iacr.org/2015/018.pdf.
[25] C. Leedham-Green and L. Soicher, Symbolic collection from the left and other strategies, *J. Symbolic Comput.* **9** (1990), 665–675.
[26] J. C. Lennox and D. J. S. Robinson, *The Theory of Infinite Soluble Groups*, Oxford Mathematical Monographs, Clarendon Press, Oxford, 2004.
[27] A. Mahalanobis, The Diffie–Hellman key exchange protocol and non-abelian nilpotent groups, *Israel J. Math.* **165** (2008), 161–87.
[28] A. G. Miasnikov, V. Shpilrain and A. Ushakov, Random subgroups of braid groups: An approach to cryptanalysis of a braid group based cryptographic protocol, in: *Public Key Cryptography* (PKC 2006), Lecture Notes in Comput. Sci. 3958, Springer, Berlin (2006), 302–314.
[29] A. G. Miasnikov and A. Ushakov, Random subgroups and analysis of the length-based and quotient attacks, *J. Math. Cryptol.* **2** (2008), 29–61.
[30] A. G. Myasnikov and V. A. Roman'kov, A linear decomposition attack, *Groups Complex. Cryptol.* **7** (2015), 81–94; see also https://arxiv.org/abs/1412.6401v1.
[31] A. Myasnikov, V. Shpilrain and A. Ushakov, *Group-Based Cryptography*, Adv. Courses Math. CRM Barcelona, Birkhäuser, Basel, 2008.
[32] A. Myasnikov, V. Shpilrain and A. Ushakov, *Non-Commutative Cryptography and Complexity of Group-theoretic Problems. With appendix by Natalia Mosina*, Math. Surveys Monogr. 177, American Mathematical Society, Providence, 2011.

[33] A. D. Myasnikov and A. Ushakov, Length based attack and braid groups: Cryptanalysis of Anshel–Anshel–Godlfeld key exchange protocol, in: *Public Key Cryptography* (PKC 2007), Lecture Notes in Comput. Sci. 4450, Springer, Berlin (2007), 76–88.
[34] V. A. Roman'kov, Equations over groups, *Groups Complex. Cryptol.* **4** (2012), no. 2, 191–239.
[35] V. A. Roman'kov, *Algebraic Cryptography* (in Russian), Omsk, Omsk State University, 2013.
[36] V. A. Roman'kov, Cryptanalysis of some schemes applying automorphisms (in Russian), *Appl. Discrete Math.* **3** (2013), 35–51.
[37] V. A. Roman'kov, A polynomial time algorithm for the braid double shielded public key cryptosystems, preprint (2014), https://arxiv.org/abs/1412.5277v1.
[38] V. A. Roman'kov, Linear decomposition attack on public key exchange protocols using semidirect products of (semi)groups, preprint (2015), https://arxiv.org/abs/1501.01152v1.
[39] V. A. Roman'kov and A. Menshov, Cryptanalysis of Andrecut's public key cryptosystem, preprint (2015), https://arxiv.org/abs/1507.01496v1.
[40] D. Segal, *Polycyclic Groups*, Cambridge Tracts in Math. 82, Cambridge University Press, Cambridge, 2005.
[41] V. Shpilrain and G. Zapata, Using the subgroup membership search problem in public key cryptography, in: *Algebraic Methods in Cryptography* (Mainz/Bochum 2005), Contemp. Math. 418, American Mathematical Society, Providence (2006), 169–181.
[42] C. C. Sims, *Computation with Finitely Presented Groups*, Encyclopedia Math. Appl. 48, Cambridge University Press, Cambridge, 1994.
[43] B. Tsaban, Practical polynomial time solutions of several major problems in noncommutative-algebraic cryptography (preliminary announcement), preprint (2014), https://eprint.iacr.org/2014/041/20140115:201530.
[44] B. Tsaban, Polynomial time solutions of computational problems in noncommutative algebraic cryptography, *J. Cryptology* **28** (2015), no. 2, 601–622.
[45] The GAP group, GAP – Groups, Algorithms and Programming, 2000.