Research Article

Jonathan Gryak and Delaram Kahrobaei*

The status of polycyclic group-based cryptography: A survey and open problems

DOI: 10.1515/gcc-2016-0013
Received June 22, 2016

Abstract: Polycyclic groups are natural generalizations of cyclic groups but with more complicated algorithmic properties. They are finitely presented and the word, conjugacy, and isomorphism decision problems are all solvable in these groups. Moreover, the non-virtually nilpotent ones exhibit an exponential growth rate. These properties make them suitable for use in group-based cryptography, which was proposed in 2004 by Eick and Kahrobaei [10]. Since then, many cryptosystems have been created that employ polycyclic groups. These include key exchanges such as non-commutative ElGamal, authentication schemes based on the twisted conjugacy problem, and secret sharing via the word problem. In response, heuristic and deterministic methods of cryptanalysis have been developed, including the length-based and linear decomposition attacks. Despite these efforts, there are classes of infinite polycyclic groups that remain suitable for cryptography. The analysis of algorithms for search and decision problems in polycyclic groups has also been developed. In addition to results for the aforementioned problems we present those concerning polycyclic representations, group morphisms, and orbit decidability. Though much progress has been made, many algorithmic and complexity problems remain unsolved; we conclude with a number of them. Of particular interest is to show that cryptosystems using infinite polycyclic groups are resistant to cryptanalysis on a quantum computer.

Keywords: Polycyclic groups, cryptography, complexity

MSC 2010: 94A60, 20F10

1 Introduction

In cryptography, many of the most common key exchange protocols, including RSA and Diffie–Hellman, rely upon hardness assumptions related to integer factorization and discrete logarithms for their security. While there are no known efficient algorithms for performing the above operations on conventional computers, Peter Shor devised a quantum algorithm [39] that solves both of these problems in polynomial time. This has motivated the search for alternative methods for constructing cryptosystems. One such methodology is non-commutative cryptography, which unlike the aforementioned conventional systems does not operate over the integers. Instead, non-commutative cryptographic systems are built upon groups and other algebraic structures whose underlying operations are non-commutative.

Jonathan Gryak: CUNY Graduate Center, PhD Program in Computer Science, City University of New York, USA, e-mail: jgryak@gradcenter.cuny.edu
***Corresponding author: Delaram Kahrobaei:** CUNY Graduate Center, PhD Program in Computer Science and NYCCT, Mathematics Department, City University of New York, USA, e-mail: dkahrobaei@gc.cuny.edu

In 1999, Anshel, Anshel and Goldfeld [1] and Ko, Lee, Cheon, Han, Kang and Park [25] introduced key exchange protocols whose security is based in part on the conjugacy search problem: for a group G, given that $u, v \in G$ are conjugate, find an x in G such that $u^x = v$. Though braid groups were the suggested platform for both protocols, other classes of groups can be employed. In general, groups suitable for use in non-commutative cryptography must be well known and possess the following properties: a solvable word problem, a computationally difficult group-theoretic problem, a "fast" word growth rate, and the namesake non-commutativity [33].

In 2004, Eick and Kahrobaei [10] investigated the algorithmic properties of polycyclic groups. In particular, they explored how the time complexity of the word and conjugacy problems varied with respect to a group's Hirsch length. Their experiments showed that the while the time complexity of the conjugacy problem grew exponentially with increased Hirsch length, the word problem remained efficiently solvable. These results suggested the suitability of polycyclic groups for use in cryptography, and stimulated research into cryptosystems based on these groups and their underlying algorithmic problems.

In this paper, we survey the development of group-based cryptography over polycyclic and metabelian groups. In Section 2 we discuss the algorithmic properties of polycyclic groups. Polycyclic groups and their intrinsic presentations are defined, as well as several other representations. A number of group-theoretic decision problems are introduced, including the word, conjugacy, and isomorphism decision problems. Note that in every polycyclic group, the three aforementioned problems are solvable. Moreover, the word problem can be solved efficiently in most cases by using a collection algorithm.

In Section 3 we describe a number of cryptosystems that have been built around these groups. These include additional key exchanges along with schemes for secret sharing, authentication, and digital signatures. This variety of cryptosystems evinces the flexibility and utility of polycyclic groups in non-commutative cryptography.

As new cryptosystems are created, so too are their dual in the form of cryptanalyses and attacks. In Section 4 we discuss the length-based attack, a heuristic technique that was the first to break the AAG protocol over braid groups. Other attacks exploit the linear representation that all polycyclic groups admit. Some, such as the field-based attack, are specific to a subclass of polycyclic groups. A more general approach is the linear decomposition attack, but its feasibility is dependent upon the size of a group's representation.

We conclude the paper with the current status of polycyclic groups cryptography. We also include a list of open problems, which we hope will guide researchers who wish to work in this exciting field.

2 Algorithmic problems in polycyclic groups

The nature of polycyclic groups enables them to be represented in several ways. These approaches give rise to complementary algorithms for solving search and decisions problems, with varying degrees of computational complexity. Due to this flexibility, we begin our study of the algorithmic problems in polycyclic groups by examining these representations.

2.1 Representations of polycyclic groups

2.1.1 Polycyclic sequences and Hirsch length

A group G is said to be *polycyclic* if it has a subnormal series

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

such that the quotient groups G_i/G_{i+1} are cyclic. This series is called a *polycyclic series*. The *Hirsch length* of a polycyclic group G is the number of infinite groups in its polycyclic series. Though a polycyclic group can have more than one polycyclic series, as a consequence of the Schreier Refinement Theorem, its Hirsch length is independent of the choice of series.

2.1.2 Polycyclic presentations

Every polycyclic group can be described by a polycyclic presentation:

$$\langle g_1, \dots, g_n \mid g_j^{g_i} = u_{ij} \text{ for } 1 \le i < j \le n,\ g_j^{g_i^{-1}} = v_{ij} \text{ for } 1 \le i < j \le n,\ g_i^{r_i} = w_{ii} \text{ for } i \in I \rangle,$$

where u_{ij}, v_{ij}, w_{ii} are words in the generators $g_{i+1}, \dots, g_n$ and I is the set of indices $i \in \{1, \dots, n\}$ such that $r_i = [G_i : G_{i+1}]$ is finite. This special type of finite presentation reveals the polycyclic structure of the underlying group, see [20, Chapter 10] for details. Unlike general finite presentations, a polycyclic presentation enables the word problem to be solved using an algorithm called *collection*. The collection algorithm is generally effective in practical applications, but its precise computational complexity remains unknown. For finite groups, collection from the left was shown to be polynomial by Leedham-Green and Soicher [27]. For infinite groups, the complexity of the collection algorithm and a modified version were analyzed by Gebhardt [16]. The resultant worst-case bound is in terms of the absolute values of all exponents occurring during the collection process, rather than the exponents of the input word. Thus a global complexity analysis of the collection algorithm remains elusive.

2.1.3 Polycyclic presentations with a Malcev basis

It has been shown by Assmann and Linton [2] that the efficacy of the collection algorithm can be improved significantly by exploiting the Malcev structure of the underlying group. This approach determines a large nilpotent normal subgroup of the given group and then exploits the Malcev correspondence for the normal subgroup. There is no known complexity analysis for this methodology.

2.1.4 Polycyclic presentations with multiplication polynomials

Du Sautoy [8] proved that every polycyclic group has a normal subgroup of finite index such that multiplication in this subgroup can be achieved by evaluating certain multiplication polynomials. This extends the well-known result by Hall [19] for torsion-free nilpotent polycyclic groups. If such multiplication polynomials are available the performance of collection in the considered group improves significantly. Additionally, it provides a basis for the complexity analysis of multiplication in polycyclic groups; it must be noted however that the index of the normal subgroup can be arbitrarily large.

2.1.5 Matrix groups

It is well known that every polycyclic group can be embedded into $\mathrm{GL}(n, \mathbb{Z})$ for some $n \in \mathbb{N}$. For groups that are additionally torsion-free and nilpotent, a matrix representation can be computed. The algorithm of Lo and Ostheimer [28] can be applied to a polycyclic presentation, while for multiplication polynomials the technique by Nickel [35] can be utilized. Multiplication of group elements in their matrix form is polynomial in the dimension n of the representation.

2.2 Growth rate

Let G be a finitely generated group. The growth rate of a group is specified by its *growth function* $\gamma : \mathbb{N} \to \mathbb{R}$ defined as $\gamma(n) = \#\{w \in G : l(w) \le n\}$, where $l(w)$ is the length of w as a word in the generators of G. As words are used as keys in group-based cryptography, there is a natural relationship between the growth rate of a group and the *key space*, the set of all possible keys. A fast growth rate engenders a large key space, making an exhaustive search of this space intractable.

A large class of polycyclic groups are known to have an exponential growth rate (namely those which are not virtually nilpotent, see Wolf [46] and Milnor [30]). Consequently, these polycyclic groups are potentially good candidates for use as platform groups.

2.3 Decision problems

In 1911, Max Dehn introduced [7] three decision problems on finitely presented groups – the word problem, the conjugacy problem, and the isomorphism problem. Below, let G be a finitely presented group:

Word Decision Problem. For any $g \in G$, determine if $g = 1_G$, the identity element of G.

Single Conjugacy Decision Problem. Determine for any $u, v \in G$ if u is conjugate to v (denoted $u \sim v$).

Isomorphism Decision Problem. Given groups G and G' with respective finite presentations $\langle X \mid R \rangle$ and $\langle X' \mid R' \rangle$, determine if G is isomorphic to G'.

For polycyclic groups all three of the above problems are decidable. The conjugacy decision problem for polycyclic groups is decidable by the results of Remeslennikov [36] and Formanek [13]. That the word problem is decidable can be observed from its formulation as a special case of the conjugacy decision problem (where $g = u, v = 1_G$), or by observing that every word has a unique normal form induced by a polycyclic presentation. The isomorphism decision problem for polycyclic groups is solvable by a result of Segal [38].

An additional decision problem called the *subgroup membership decision problem* (alternatively the generalized word decision problem) asks for any $g \in G$ and subgroup $H \leq G$, determine if $g \in H$. Malcev in [29] showed that this problem is indeed solvable for polycyclic groups.

2.4 The conjugacy search problem and its variations

Once the solvability of a group-theoretic decision problem is affirmed, the subsequent task is to produce elements (or morphisms, etc.) that are solutions to particular instances of it. The seminal protocols of noncommutative cryptography, Ko-Lee and AAG, are based in part on the conjugacy search problem (CSP). Their example spurred the development of many other protocols whose security is based on some variant of the CSP. In this section we explore these variations and the methods designed to solve them.

2.4.1 Conjugacy search problem

Let G be a group and $a_1, \ldots, a_n, b_1, \ldots, b_n$ elements of it, with $a_i \sim b_i$. The problem of finding a $c \in G$ such that for all i, $a_i^c = b_i$ is called the *(single) conjugacy search problem* for $i = 1$ and the *multiple conjugacy search problem* for $1 < i \leq n$. In polycyclic groups, the multiple conjugacy search problem for n elements reduces to n independent solutions of single conjugacy search [10]. We will therefore speak only of the conjugacy search problem without signifying arity. For any finitely presented group (polycyclic groups included) the conjugacy search problem can be solved exhaustively by recursively enumerating the conjugates of the element in question [40]. There are other approaches to solving the conjugacy search problem, many of which can solve it efficiently. However, the applicability of these methods and their relative efficiency is contingent upon addition restrictions on the group's properties, as well as the manner is which the polycyclic group is specified.

2.4.2 CSP using polycyclic presentations

For infinite polycyclic groups the algorithm proposed by Eick and Ostheimer [11] is applicable. This algorithm uses a variety of ideas: it exploits finite orbit and stabilizer computations, calculations in number fields,

and linear methods for polycyclic groups. The algorithm has been implemented and seems to be efficient for groups of small Hirsch length. An analysis of the algorithm's complexity is hindered by there being no bound on the length of the finite orbits that may occur in the computation.

The restriction of the applicability of the above algorithm to groups of small Hirsch length is supported by the experimental evidence provided by Eick and Kahrobaei in [10]. They compared the performance of the Eick–Ostheimer algorithm for the CSP against the collection algorithm for polycyclic groups of the form $G = \mathcal{O}_K \rtimes \mathcal{U}_K$, where $\mathcal{O}_K$ and $\mathcal{U}_K$ are respectively the maximal order and group of units of an algebraic number field K. In the table below, the column $H(G)$ is the Hirsch length of the group G, with the collection and conjugation entries representing the average running time over 100 trials using random words (respectively, random conjugate pairs) from G.

$H(G)$	Collection	Conjugation
2	0.00 sec	9.96 sec
6	0.01 sec	10.16 sec
14	0.05 sec	>100 hr

These results suggest that while collection remains efficient as the Hirsch length increases, the Eick–Ostheimer algorithm becomes impractical. Presently there are no known algorithms for infinite polycyclic groups of high Hirsch length. Such groups remain suitable for use as platform groups.

2.4.3 CSP using multiplication polynomials

Suppose that G instead is given by a polycyclic presentation with multiplication polynomials. Let $g_1, \ldots, g_k$ be the polycyclic generating set of the presentation and consider a generic element $g = g_1^{x_1} \cdots g_k^{x_k}$ of G. Note that g is a solution to the multiple conjugacy search problem if and only if

$$a_i g = g b_i \quad \text{for } 1 \le i \le k.$$

If $a_i = g_1^{l_{i1}} \cdots g_k^{l_{ik}}$ and $b_i = g_1^{m_{i1}} \cdots g_k^{m_{ik}}$, with $f_1, \ldots, f_k$ denoting the multiplication polynomials for G, then $a_i g = g b_i$ if and only if

$$f_j(l_i, x) = f_j(m_i, x) \quad \text{for } 1 \le i, j \le k.$$

If $f_1, \ldots, f_k$ are given as explicit polynomials over an extension field of $\mathbb{Q}$ and l_i, m_i are integer vectors, then the CSP is equivalent to determining an integer solution for a set of k polynomials in k indeterminates. Thus the CSP can also be considered from the perspective of algebraic geometry.

2.4.4 Power conjugacy search problem

The key exchange presented in Section 3.3.2 makes use of the *power conjugacy search problem*, where if it is known for some $a, b \in G$ and $n \in \mathbb{N}$ that $a^n = b^g$ for some $g \in G$, the task is to find one such n and g. Note that for $n = 1$ this reduces to the standard CSP, whereas if $g = 1_G$ this reduces to the *power search problem*.

Just as the conjugacy search problem is solvable by enumeration, so is the power conjugacy search variant, but no efficient algorithm is known.

2.4.5 Twisted conjugacy search problem

Twisted conjugacy arises in Nielsen theory, where the number of twisted conjugacy classes is related to the number of fixed points of a mapping. The *twisted conjugacy search problems* is to find, given a group G and

an endomorphism ϕ, an element $a \in G$ such that

$$t = a^{-1} w \phi(a),$$

provided that at least one such a exists.

The standard CSP can be seen as a special case of the twisted version where $\phi(x) = x$, the identity automorphism. The protocol by Shpilrain and Ushakov in Section 3.6 uses the double twisted conjugacy variant, in which the above definitions are modified to include an additional endomorphism α and the task is then to find an element $a \in G$ such that $t = \alpha(a^{-1}) w \phi(a)$.

The twisted conjugacy decision problem was proven to be decidable by Roman'kov [37]. Both the single and doubly twisted conjugacy search problems are solvable by the same method of enumeration as in the case of the standard conjugacy search problem. However, no efficient algorithm is known.

2.5 Properties of automorphism groups

The automorphism group $\mathrm{Aut}(G)$ and its subgroups have been studied extensively for polycyclic groups G. Like polycyclic groups themselves, $\mathrm{Aut}(G)$ is finitely presented [3], and the outer automorphism group $\mathrm{Out}(G)$ is isomorphic to a linear group [45].

A decision problem related to $\mathrm{Aut}(G)$ is the *orbit decision problem*. Given elements $g, h \in G$ and a subset $A \subseteq \mathrm{Aut}(G)$, determine if there exists $\alpha \in A$ such that $g = \alpha(h)$. Note that if $A = \mathrm{Inn}(G)$, this problem reduces to the standard conjugacy decision problem. When G is polycyclic, all cyclic subgroups $A \leq \mathrm{Aut}(G)$ are orbit decidable [5].

For groups G in the larger class of polycyclic-by-finite (or virtually polycyclic) groups, the conjugacy decision problem is decidable in $\mathrm{Aut}(G)$ (see [38]). Additionally, $\mathrm{Aut}(G)$ is either virtually polycyclic or it contains a non-abelian free subgroup [9].

2.6 Quantum algorithms

As mentioned in the introduction, the development of non-commutative cryptography was spurred by the publication of Shor's algorithm. The algorithm enables a sufficiently sized quantum computer to perform integer factorization and compute discrete logs in polynomial time, as opposed to in exponential time on a conventional computer.

From a group-theoretic perspective, Shor's algorithm can be seen as solving the *hidden subgroup problem* in finite cyclic groups. A subgroup $H \leq G$ is considered *hidden* by a function f from G to a set X if it is constant over all cosets of H. A 2003 paper by [4] by Batty, Rees, Braunstein and Duncan explores this and other applications of quantum algorithms to group theory, including an algorithm by Watrous that determines the order of a finite solvable group. Bonanome showed [6] that a modified version of Grover's algorithm can solve the automorphism and conjugacy decision problems in finite groups, as well as determine fixed points. The algorithm by Ivanyos, Sanselme and Santha [22] solves the hidden subgroup problem for finite nilpotent groups of class 2. There are also partial results to solving the power conjugacy problem [12].

Despite these developments in the use quantum algorithms for finite groups, there are no known quantum algorithms that are applicable to infinite groups.

3 Cryptosystems

For the systems described below, the chosen platform group G should be suitable for cryptography as delineated in the introduction. Let G be finitely presented and non-abelian. Group operations (products, inverses) and solving the word problem must be efficient. Additional criteria for each protocol or scheme are stated in

their respective descriptions. Note that the precise definitions of each algorithmic search or decision problem can be found in Section 2.

3.1 The Anshel–Anshel–Goldfeld key-exchange protocol

In their 1999 paper [1], Anshel, Anshel and Goldfeld introduced the *commutator key exchange protocol*, which is also referred to as AAG key exchange or Arithmetica. The group-based version of the key exchange described below is in the style of [31]. Prior to the key exchange, the protocol parameters $N_1, N_2, L_1, L_2, L \in \mathbb{N}$ with $1 \leq L_1 \leq L_2$ are chosen and made public:

(1) Alice chooses a set $\bar{A} = \{a_1, \ldots, a_{N_1}\}$, with Bob choosing $\bar{B} = \{b_1, \ldots, b_{N_2}\}$, where $a_i, b_j \in G$ are words of length in $[L_1, L_2]$. Note that $\bar{A}$ and $\bar{B}$ both generate subgroups of G. These sets are then exchanged publicly with each other.

(2) Alice constructs her private key as $A = a_{s_1}^{\varepsilon_1} \ldots a_{s_L}^{\varepsilon_L}$, with $a_{s_k} \in \bar{A}$ and $\varepsilon_k \in \{-1, 1\}$. Similarly, Bob computes as his private key $B = b_{t_1}^{\delta_1} \ldots b_{t_L}^{\delta_L}$, with $b_{t_k} \in \bar{B}$ and $\delta_k \in \{-1, 1\}$.

(3) Alice then computes $b_j' = A^{-1} b_j A$ for $1 \leq j \leq N_2$ and sends this collection to Bob, while Bob computes and sends Alice $a_i' = B^{-1} a_i B$ for $1 \leq i \leq N_1$.

(4) Alice and Bob can now compute a shared key $\kappa = A^{-1}B^{-1}AB$, which is the *commutator* of A and B, denoted $[A, B]$. Alice computes (using only the a_i' which correspond to some s_i of her private key):

$$\begin{aligned}
\kappa_A &= A^{-1} {a'}_{s_1}^{\varepsilon_1} \cdots {a'}_{s_L}^{\varepsilon_L} \\
&= A^{-1} B^{-1} a_{s_1}^{\varepsilon_1} B \cdots B^{-1} a_{s_L}^{\varepsilon_L} B \\
&= A^{-1} B^{-1} a_{s_1}^{\varepsilon_1} (BB^{-1}) a_{s_2}^{\varepsilon_2} B \cdots B^{-1} a_{s_{L-1}}^{\varepsilon_{L-1}} (BB^{-1}) a_{s_L}^{\varepsilon_L} B \\
&= A^{-1} B^{-1} a_{s_1}^{\varepsilon_1} a_{s_2}^{\varepsilon_2} \cdots a_{s_{L-1}}^{\varepsilon_{L-1}} a_{s_L}^{\varepsilon_L} B \\
&= A^{-1} B^{-1} AB.
\end{aligned}$$

Analogously, Bob computes $\kappa_B = B^{-1}A^{-1}BA$. The shared secret is then

$$\kappa = \kappa_A = \kappa_B^{-1}.$$

As noted in [41], the security of AAG is based on both the simultaneous conjugacy search problem and the subgroup membership search problem.

3.2 Ko–Lee key exchange protocol

Originally specified by Ko, Lee, Cheon, Han, Kang and Park [25] using braid groups, their non-commutative analogue of Diffie–Hellman key exchange can be generalized to work over other platform groups. Let G be a finitely presented group, with $A, B \leq G$ such that all elements of A and B commute.

An element $g \in G$ is chosen, and g, G, A, B are made public. A shared secret can then be constructed as follows:

- Alice chooses a random element $a \in A$ and sends g^a to Bob.
- Bob chooses a random element $b \in B$ and sends g^b to Alice.
- The shared key is then g^{ab}, as Alice computes $(g^b)^a$, which is equal to Bob's computation of $(g^a)^b$ as a and b commute.

The security of Ko–Lee rests upon solving the conjugacy search problem within the subgroups A and B.

3.3 Non-commutative ElGamal key-exchange

In the 2006 paper by Kahrobaei and Khan [23], the authors proposed two adaptations of the ElGamal asymmetric key encryption algorithm for use in non-commutative groups. Let S, T be finitely generated subgroups such that all elements of S and T commute. In any exchange, the triple $\langle G, S, T \rangle$ is made public.

3.3.1 Non-commutative key exchange using conjugacy search

- Bob chooses $s \in S$ as his private key, a random element $b \in G$, and publishes as his public key the tuple $\langle b, c\rangle$, with $c = b^s$.
- To create a shared secret $x \in G$, Alice chooses x and a $t \in T$. Using Bob's public key, she publishes $\langle h, E\rangle$, with $h = b^t$ and $E = x^{c^t}$.
- To recover x, Bob first computes h^s, which, as elements of S and T commute, yields

$$h^s = (b^t)^s = (b^s)^t = c^t.$$

  Bob can then calculate $x = E^{(c^t)^{-1}}$.
  The security of this scheme relies upon the conjugacy search problem in G.

3.3.2 Non-commutative key exchange using power conjugacy search

By imposing the additional requirement that the conjugacy search problem is efficiently solvable in G, we can now describe a variation of the previous protocol:

- Bob chooses $s \in S$ and $n \in \mathbb{Z}$ as his private key, as well as a random element $b \in G$, and publishes as his public key $\langle v, w\rangle$, with $v = g^n$ and $w = g^{-1}sg$. Note that $w^n = (s^{-1}gs)^n = s^{-1}g^n s = s^{-1}vs$.
- Alice chooses a shared secret $x \in G$, along with $m \in \mathbb{Z}$ and $t \in T$, and publishes $\langle h, E\rangle$, with $h = t^{-1}w^m t$ and $E = x^{-1}t^{-1}v^m tx$.
- To recover x, Bob first computes $E' = sh^n s^{-1} = st^{-1}sg^{mn}st$, which, as elements of S and T commute, yields

$$E' = t^{-1}v^m t.$$

  Knowing that $E = x^{-1}E'x$, Bob can then solve the conjugacy search problem to obtain the shared secret x.
  The security of this scheme rests upon the power conjugacy search problem in G.

3.4 Non-commutative digital signature

The following digital signature scheme was proposed in a paper by Kahrobaei and Koupparis [24]. The platform group G must be infinite. The scheme uses two functions: $f : G \to \{0, 1\}^*$, which encodes elements of the group as binary strings; and $H : \{0, 1\}^* \to G$, a collision-resistant hash function. Using these functions (which are made public along with G), a message can be signed and verified as follows:

- *Key Generation:* The signer first chooses an element $g \in G$, whose centralizer, the set of elements that commute with g, contains 1_G and powers of g exclusively. The private key consists of $s \in G$ and $n \in \mathbb{N}$, where n is chosen to be highly composite. The public key $x = g^{ns}$ is then published.
- *Signing Algorithm:* To sign a message m, the signer chooses a random element $t \in G$ and a random factorization $n_i n_j$ of n, and computes the following (with $\|$ denoting concatenation):

$$\begin{aligned} y &= g^{n_i t}, \\ h &= H(m \,\|\, f(y)), \\ \alpha &= t^{-1}shy. \end{aligned}$$

  The signature $\sigma = \langle y, \alpha, n_j\rangle$ and the message m are then send to the message recipient.
- *Verification:* To verify, the recipient computes $h' = H(m \,\|\, f(y))$, and accepts the message as authentic if and only if the following equality holds:

$$y^{n_j \alpha} = x^{h'y}.$$

The security of the signature scheme is based on the collision resistance of the hash function, the conjugacy search problem in G, and the Diffie–Hellman assumption. Moreover, Alice must maintain a public list of previously used factors of n, and regenerate s and n after a few uses.

3.5 A key exchange using the subgroup membership search problem

In [43], Shpilrain and Zapata proposed a public key exchange protocol over relatively free groups. Given a free group G_n of rank n and $R \trianglelefteq G_n$, the quotient group $\mathcal{G}_n = G_n/R$ is *relatively free* if for any endomorphism ψ of G_n, $\psi(R) \leq R$.

The protocol utilizes two types of automorphisms:

- Let $\{x_1, \dots, x_n\}$ be the generators of $\mathcal{G}_n$. The *Neilsen automorphisms* are defined as

$$\alpha_j(x_i) = \begin{cases} x_i^{-1}, & i = j, \\ x_i, & i \neq j, \end{cases}$$

and

$$\beta_{jk}(x_i) = \begin{cases} x_i x_j, & i = k, \\ x_i, & i \neq k. \end{cases}$$

- For relatively free groups like $\mathcal{G}_n$, the Nielsen automorphisms form a subgroup of $\mathrm{Aut}(\mathcal{G}_n)$ under composition. Elements in this subgroup are called *tame* automorphisms. In constructing a private key, the protocol uses both tame and non-tame automorphisms.

In the key exchange below, let $\mathcal{F}_n$ and $\mathcal{F}_{n+m}$ denote the relatively free groups of rank n and $n+m$, with respective generating sets $\{x_1, \dots, x_n\}$ and $\{x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}\}$. Moreover, let $\mathcal{F}_j^i = \prod_i \mathcal{F}_j$ denote the direct product of i instances of the relatively free group of rank j. Finally, let $z(x_1, \dots, x_{n+m})$ denote a word z written in the alphabet $\{x_1, \dots, x_{n+m}\}$. The exchange then proceeds as follows:

(1) Alice chooses an automorphism $\phi \in \mathrm{Aut}(\mathcal{F}_{n+m})$, where $\phi = \tau_1 \circ \cdots \circ \tau_k$, a composition of Nielsen automorphisms and non-tame automorphisms which are readily invertible. Alice uses $\phi^{-1} = \tau_k^{-1} \circ \cdots \circ \tau_1^{-1}$ as her private key. For each generator x_i of $\mathcal{F}_{n+m}$, Alice computes the word $\phi(x_i) = y_i(x_1, \dots, x_{n+m})$. She then computes $\hat{y}_i$, which is the restriction of each y_i to a word in the generators of $\mathcal{F}_n$. The tuple $\langle \hat{y}_1, \dots, \hat{y}_{n+m} \rangle$ is then published as the public key.

(2) Bob chooses a word w in the subgroup S of $\mathcal{F}_{n+m}^{n+m}$ consisting of words of the form

$$v = (v_1(x_1, \dots, x_n), \dots, v_n(x_1, \dots, x_n), 1, \dots, 1).$$

Thus $S \cong \mathcal{F}_n^n$, and $w = (w_1(x_1, \dots, x_n), \dots, w_n(x_1, \dots, x_n))$. Using the components of the public key, Bob encrypts w by replacing each instance of x_i in $\hat{y}_j$ by w_i. The encrypted tuple

$$\hat{\phi}(w) = \langle \hat{y}_1(w_1, \dots, w_n), \dots, \hat{y}_n(w_1, \dots, w_n) \rangle$$

is then sent to Alice.

(3) Alice applies ϕ^{-1} (restricted to $\mathcal{F}_n^n$) component-wise to $\hat{\phi}(w)$ to recover w', a unique normal form of w. This w' is the shared key.

The security of the protocol is two-fold. Decrypting a particular message $\hat{\phi}(w)$ is equivalent to solving the subgroup membership search problem in the subgroup generated by the public key. To recover the private key, an attacker must recover the automorphism ϕ and its inverse from the public image of the generators $\hat{y}_i$, restricted to the subgroup $\mathcal{F}_n$. Shpilrain and Zapata claim there is no known method of accomplishing this outside of an exhaustive search of $\mathrm{Aut}(\mathcal{F}_{n+m})$.

The authors suggest free metabelian groups of rank r (with $r = 10$, $n = 8$, $m = 2$) as platform groups for their protocol. Aside from meeting the standard criteria for platform groups, these groups have the requisite supply of non-tame automorphisms and the subgroup membership search problem is known to be super-polynomial in these groups.

3.6 An authentication scheme based on the twisted conjugacy problem

In [42], Shpilrain and Ushakov introduced a non-commutative authentication scheme based on the Fiat–Shamir scheme. The platform group G can in fact be a semigroup, provided that an antihomomorphism

$* : G \to G$, i.e., $(ab)^* = b^*a^*$, exists. The endomorphism group of G should also be sufficiently large to preclude an exhaustive search. In the simulation of the protocol below, Alice is authenticating herself to Bob:

(1) Alice chooses $s \in G$ as her private key. She then chooses $w, t \in G$ and endomorphisms ϕ, ψ such that $t = \psi(s^*)w\phi(s)$. The public key $\langle \phi, \psi, w, t \rangle$ is then published.
(2) The commitment/verification exchange proceeds as follows:
   (a) Alice chooses an $r \in G$ and computes the *commitment* $u = \psi(r^*)t\phi(r)$, sending it to Bob.
   (b) Bob chooses a random bit c and sends it to Alice.
   (c) Alice replies with $v = r$ if $c = 0$, and $v = sr$ otherwise.
   (d) Bob *verifies* the commitment u by computing u', and accepts if $u = u'$:
      - If $c = 0$, Bob computes $u' = \psi(v^*)t\phi(v) = \psi(r^*)t\phi(r)$.
      - If $c = 1$, Bob computes $u' = \psi(v^*)t\phi(v)$, where

$$u' = \psi((sr)^*)t\phi(sr) = \psi(r^*)\psi(s^*)w\phi(s)\phi(r) = \psi(r^*)t\phi(r).$$

Note that the commitment/verification steps must be performed k times to yield a probability of successful forgery less than $\frac{1}{2^k}$. The security of the scheme is based on the apparent hardness of the double twisted conjugacy search problem.

3.7 Authentication schemes based on semigroup actions

Drawing inspiration from the zero-knowledge proof by Feige, Fiat, and Shamir; Grigoriev and Shpilrain [17] introduced two generic protocol schema based upon (semi)group actions and provided several concrete examples.

3.7.1 An authentication scheme based on the endomorphism problem

One such instance of their second protocol is based upon the endomorphism problem. While this scheme can be used with a semigroup or some other algebraic structure, the structure S must meet several criteria:

- An algorithm exists to determine if the function over S is an endomorphism. If S is specified by a presentation, this criterion is satisfied by S having an efficiently solvable word problem.
- An algorithm exists to determine if the function over S is an automorphism of S.
- The endomorphism search problem in S should be demonstrably NP-hard.

As before, in the protocol exchange below Alice is authenticating herself to Bob:

(1) Alice chooses an endomorphism $\phi : S \to S$ as her private key. Alice then chooses elements $s, t \in S$ such that $t = \phi(s)$. The public key $\langle S, s, t \rangle$ is then published.
(2) The commitment/verification exchange proceeds as follows:
   (a) Alice chooses an automorphism ψ and computes the *commitment* $u = \psi(t)$, sending it to Bob.
   (b) Bob chooses a random bit c and sends it to Alice.
   (c) Alice replies with $v = \psi(t)$ if $c = 0$, and $v = \psi \circ \phi$ otherwise.
   (d) Bob *verifies* the commitment u by computing u':
      - If $c = 0$, Bob computes $u' = \psi(t)$ and accepts if $u = u'$ and ψ is an automorphism.
      - If $c = 1$, Bob computes $u' = (\psi \circ \phi)(s)$ and accepts if $u = u'$ and $\psi \circ \phi$ is an endomorphism.

3.7.2 An authentication scheme based on the group isomorphism problem

The following is a new instance of the first protocol, which requires a class of finitely presented groups $\mathcal{C}$ with the following algorithmic properties:

- The class $\mathcal{C}$ must have an efficiently solvable isomorphism decision problem.
- The isomorphism search problem in $\mathcal{C}$ should be demonstrably NP-hard.

The protocol exchange is as follows:

(1) Alice chooses two isomorphic groups G_1 and G_2 from $\mathcal{C}$. Alice then chooses an isomorphism $\alpha : G_1 \to G_2$ as her private key, and publishes $\langle G_1, G_2 \rangle$.
(2) The commitment/verification exchange proceeds as follows:
   (a) Alice chooses a group $G \in \mathcal{C}$ and an isomorphism $\beta : G \to G_1$, sending the *commitment* G to Bob.
   (b) Bob chooses a random bit c and sends it to Alice.
   (c) Alice replies with $\gamma = \alpha$ if $c = 0$, and $\gamma = \alpha \circ \beta$ otherwise.
   (d) Bob *verifies* the commitment G by computing $G' = \gamma G$:
      - If $c = 0$, Bob accepts if $G' \cong G_1$.
      - If $c = 1$, Bob accepts if $G' \cong G_2$.

For both of the above authentication schemes, the commitment/verification steps must be performed multiple times to yield a low probability of successful forgery.

3.8 Secret sharing schemes based on the word problem

Habeeb, Kahrobaei and Shpilrain [18] proposed two secret sharing schemes for groups whose presentations satisfy small cancellation conditions. In a (t, n) scheme, the *threshold* t is the number of participants that are required to recover the shared secret (created and disseminated by the "dealer"), with n the total number of participants. In both schemes, the dealer wishes to share a k-bit integer x that will be represented as a column vector $C \in \mathbb{B}^k$. Prior to initiating the secret sharing, the dealer chooses groups G_j given by the presentations $\langle X \mid R_j \rangle$, where X is a common generating set and R_j a unique set of relators for each participant P_j. The generating set X is then made public. Note that both schemes require secure communication channels between both the dealer and participants and between the participants themselves. These secure channels can be achieved using any preferred public key exchange protocol.

3.8.1 An (n, n)-threshold scheme

In this scheme, all n participants are required to reproduce the secret x:

(1) The dealer sends each participant P_j their unique relator set R_j.
(2) The dealer decomposes C into n vectors $C_j \in \mathbb{B}^k$ such that $C = \sum_j C_j$.
(3) Each entry c_{kj} of C_j is then encoded as a word $w_{kj} \in G_j$ such that $w_{kj} \equiv 1_{G_j}$ if $c_{kj} = 1$ and $w_{kj} \not\equiv 1_{G_j}$ otherwise. The words w_{kj} are then sent to P_J using an open channel.
(4) For each w_{kj}, participant P_j solves the word problem in G_j and reconstructs C_j.
(5) The participants can then recover C by summing over all vectors C_j. Note that a secure sum protocol can be employed so that the vectors C_j need not be divulged to the other participants.

3.8.2 A (t, n)-threshold scheme

In this scheme, t participants are required to reproduce the secret x. As in Shamir's secret sharing, x must be an element in $\mathbb{Z}_p$ with p prime, and a polynomial f of degree $t - 1$ must be chosen by the dealer such that $f(0) = x$. The dealer must also choose k-bit integers $y_j \equiv f(j) \pmod{p}$.

(1) The dealer sends each participant P_j their unique relator set R_j.
(2) Each y_j has its bits b_{kj} encoded as words $w_{kj} \in G_j$ as in the previous scheme.
(3) For each w_{kj}, participant P_j solves the word problem in G_j, yielding y_j.
(4) The participants can then perform polynomial interpolation using the integers y_j to recover f. The shared secret x is then revealed by evaluating $f(0)$. If $t \geq 3$, Lagrange interpolation can be employed so that the vectors B_j need not be divulged to the other participants.

The security of these schemes is contingent upon the relators R_J being kept secret.

4 Cryptanalysis and attacks

In this section we present a number of attacks against group-based cryptosystems, with an emphasis on those that are applicable to polycyclic groups.

4.1 Length-based attack

The length-based attack (LBA) is an incomplete, local search that attempts to solve the conjugacy search problem (or its generalized version) by using the length of a word as a heuristic. It was first introduced by Hughes and Tannenbaum [21] as a means to attack the AAG key exchange protocol over braid groups. In [15], Garber, Kaplan, Teicher, Tsaban and Vishne explored the use of length functions based on the Garside normal form of braid group elements. They demonstrated experimentally that the length-based attack in this context could break the AAG protocol, albeit inefficiently.

As the length-based attack is an iterative improvement search, it is susceptible to failing at peaks and plateaux in the search space. In [31], Myasnikov and Ushakov identified when these peaks occur and were able to make successive refinements to the algorithm to yield a high success rate.

More recently, the authors of [14] analyzed the LBA against AAG over polycyclic groups. They found that the success rate of the LBA decreased as the Hirsch length of the platform group increased. Their version of the LBA, essentially a local beam search, is presented in Algorithm 1. Note that the a_i, b', $\bar{b}'$, and N_1 are from the AAG protocol exchange in Section 3.1, while $\bar{c}'$ is a candidate conjugator set. The length of a conjugator set $\bar{c}' = (c_1, \ldots, c_j)$ is defined as $\sum_j |c_j|$.

Algorithm 1. LBA with Memory 2.

Initialize $S = \{(|\overline{b'}|, \overline{b'}, 1_G)\}$.
while not time out **do**
  for $(|\overline{c}|, \overline{c}, x) \in S$ **do**
    Remove $(|\overline{c}|, \overline{c}, x)$ from S
    Compute $\overline{c}^{a_i^{\varepsilon}}$ for all $i \in \{1 \ldots N_1\}$ and $\varepsilon = \pm 1$
    if $\overline{c}^{a_i^{\varepsilon}} = \overline{b}$ **then** output inverse of $a_i^{\varepsilon} x$ and stop
    Save $(|\overline{c}^{a_i^{\varepsilon}}|, \overline{c}^{a_i^{\varepsilon}}, a_i^{\varepsilon} x)$ in S'
  end for
  After all conjugation attempts, sort S' by the first element of every tuple
  Copy the smallest M elements into S and delete the rest of S'
end while
Otherwise, output FAIL

4.2 Linear decomposition attack

In [32], Miasnikov and Roman'kov introduced the linear decomposition attack. The attack is a general framework for the cryptanalysis of a number of group-theoretic analogues of Diffie–Hellman key exchange. For a protocol to be susceptible to the attack its platform groups must admit a linear representation. Moreover, the algorithmic security assumption of the protocol must be equivalent to commutative linear transformations. Note that the AAG protocol is not susceptible to this attack.

Given the linear structure V and subsets $W \leq V$ and $U \leq \mathrm{End}(V)$, the attack first computes a basis for the span of all vectors of the form w^u, with $w \in W$ and $u \in \langle U \rangle$. This can be done in polynomial time with respect to the dimension of V and the sizes of W and U. This calculation can be performed offline if the platform group for a particular protocol is fixed. The public group elements transmitted during the key exchange can then

be decomposed using this basis to reconstruct the shared secret without discovering the private information of each party, negating the need for an attacker to solve the underlying security problem.

The attack requires the platform group to be specified by either its linear representation V (as a vector space or an associative algebra) or by a presentation coupled with a faithful embedding into $\mathrm{GL}(V)$. Moreover, the linear space into which the group is embedded must be of sufficiently small dimension to make the attack tractable. While the dimension of the smallest linear embeddings of finite groups and some classes of infinite groups such as torsion-free nilpotent and polycyclic-by-finite are known, the authors concede that no such bounds are known for other linear groups, including general polycyclic groups and metabelian groups.

4.3 Field-based attack

Kotov and Ushakov [26] investigated the security of the AAG key-exchange protocol used with certain polycyclic groups of the form $G_F = \mathcal{O}_F \rtimes U_F$, where $\mathcal{O}_F$ is the maximal order and U_F is the unit group generated by an irreducible polynomial in the algebraic number field F. In the semidirect product, U_F acts on $\mathcal{O}_F$ by right multiplication. These groups were the original polycyclic platform groups suggested by Eick and Kahrobaei in [10]. In [14], Garber, Kahrobaei and Lam showed that such groups were resistant to the length-based attack, with the attack's success decreasing as the Hirsch length of the group G_F increased.

Contrary to these results, the field-based attack devised by the authors is able to recover the shared key regardless of the group's Hirsch length. Using a deterministic, polynomial time algorithm, the key is recovered by solving a linear system of conjugacy equations over the field F. If the group G_F is specified as a semidirect product and F is given in matrix form, the attack can be directly applied. However, if G_F is given by a polycyclic presentation, the authors construct a linear representation from the presentation prior to recovering the shared key.

While the field-based attack is successful in these particular groups, the authors concede that their attack does not preclude other polycyclic groups from consideration for the AAG protocol. We claim that there are other classes of polycyclic groups that are resistant to such an attack. Such platform groups would be specified by their polycyclic presentations and have matrix representations that are not readily computable.

4.4 Quotient attack

In attempting to recover the shared secret from the public information of the AAG protocol, the length-based attack (LBA) operates as if the platform group G is a free group. The success of the LBA on non-free groups motivated Miasnikov and Ushakov in [34] to investigate the asymptotic properties of the given platform groups. Ultimately they determined that the LBA is successful for groups in which a random choice of elements is very likely to generate a free subgroup of G.

These investigations led to a new form of attack for the AAG key exchange protocol and others that use some variation of the membership or conjugacy search problems. Dubbed the quotient attack, the algorithms solve the search problems in a quotient group G/N. If G/N possesses the exponentially-generic free basis property, the solution in the quotient will yield one in the original group. The time complexity of the attack is contingent upon the particular class of platform groups. For pure braid groups PB_n the authors prove that the complexity is $O(n^2)$.

As polycyclic groups do not possess the free basis property nor any free subgroups, this attack is not applicable.

4.5 Linear centralizer attack

Tsaban [44] devised the linear centralizer attack against AAG over the original braid group platform. The attack exploits a faithful linear representation of a braid group $\mathcal{B}_n$. Using this representation, the algorithm

computes a basis for the double centralizer of the public subsets of the AAG protocol (which are contained in their respective double centralizers). This process produces one half of the shared key, after which random elements are tested to find an inverse that yields the other half. The algorithm runs in expected polynomial time with respect to n, but is impractical for even modest values of n.

The applicability of the linear centralizer attack to other platform groups is limited to those whose faithful representations are known and whose linear representations are sufficiently small. As mentioned previously with respect to the linear decomposition attack, these aspects of polycyclic groups are currently unknown.

5 Conclusion

In this paper we have presented a survey of over ten years of research related to polycyclic group-based cryptography. We began with a study of the algorithmic properties of polycyclic groups. Polycyclic groups admit a number of representations, including polycyclic presentations, multiplication polynomials, and as matrices. In addition to the decidability of the classic decision problems of word, conjugacy, and isomorphism, the twisted conjugacy and orbit problem are also decidable. Moreover, the conjugacy decision problem for the automorphism group $\mathrm{Aut}(G)$ of a polycyclic group G is decidable.

We have seen that there are a variety of key exchanges, digital signature systems, and secret sharing schemes for which a polycyclic group is an appropriate choice of platform group. Moreover, many of these schemes use computational problems in polycyclic groups other than the conjugacy search problem. The security of the key exchange by Shpilrain–Zapata (Section 3.5) is based on the subgroup membership search problem, while the authentication schemes of Shpilrain–Ushakov (Section 3.6) and Grigoriev–Shpilrain (Section 3.7) are based on the doubly twisted conjugacy search problem and the endomorphism problem, respectively.

The cryptanalysis of polycyclic and other group-based cryptosystems has focused almost exclusively on the AAG key exchange protocol. The length-based attack (Section 4.1) was found to be effective for braid groups and other groups that possess the free basis property, which polycyclic groups do not. A special class of polycyclic groups formed from semidirect products over algebraic number fields was found to be susceptible to the field-based attack of Section 4.3. If a polycyclic groups is given by a matrix representation of tractable dimension, the linear decomposition attack of Section 4.2 can be used for cryptanalysis. However, there is no known general method for faithfully embedding a polycyclic group into the general linear group, and no known bound on the dimensionality of such an embedding. Polycyclic groups that are specified by a polycyclic presentation and that are non-virtually nilpotent remain a promising class of platform groups for group-based cryptography.

While there has been considerable research activity concerning polycyclic groups and their attendant cryptosystems over the last decade, many computational complexity and algorithmic questions remain unanswered. We have collected several of these outstanding problems below, with the hope of stimulating interest in their solutions:

(1) What is the complexity of the isomorphism search problem in polycyclic groups?
(2) What is the complexity of the twisted search conjugacy problem in polycyclic groups?
(3) What is the complexity of the power conjugacy problem in polycyclic groups?
(4) What is the complexity of the geodesic length problem in polycyclic groups?
(5) What is the complexity of the n-root problem in polycyclic groups?
(6) What is the complexity of finding matrix representation of polycyclic groups?
(7) What is the complexity of the conjugacy problem in the automorphism of polycyclic groups?
(8) What is the complexity of the search endomorphism (automorphism) problem in polycyclic groups?
(9) What is the complexity of the homomorphism problem in polycyclic groups?
(10) Are polycyclic group-based cryptosystems resistant to quantum algorithms?
(11) What is the complexity of the subgroup membership search problem in polycyclic groups?

Acknowledgment: We would like to thank Bettina Eick for her contributions regarding polycyclic groups and their algorithmic properties.

Funding: Delaram Kahrobaei is partially supported by a PSC-CUNY grant from the CUNY Research Foundation, the City Tech Foundation, and ONR (Office of Naval Research) grant N00014-15-1-2164. Delaram Kahrobaei has also partially supported by an NSF travel grant CCF-1564968 to IHP in Paris.

References

[1] I. Anshel, M. Anshel and D. Goldfeld, An algebraic method for public-key cryptography, *Math. Res. Lett.* **6** (1999), 287–291.
[2] B. Assmann and S. Linton, Using the Mal'cev correspondence for collection in polycyclic groups, *J. Algebra* **316** (2007), no. 2, 828–848.
[3] L. Auslander, The automorphism group of a polycyclic group, *Ann. of Math. (2)* **89** (1969), 314–322.
[4] M. Batty, S. Rees, S. Braunstein and A. Duncan, Quantum algorithms in group theory, in: *Computational and Experimental Group Theory* (Baltimore 2003), Contemp. Math. 349, American Mathematical Society, Providence (2004), 1–62.
[5] O. Bogopolski, A. Martino and E. Ventura, Orbit decidability and the conjugacy problem for some extensions of groups, *Trans. Amer. Math. Soc.* **362** (2010), no. 4, 2003–2036.
[6] M. Bonanome, *Quantum algorithms in combinatorial group theory*, Ph.D. thesis, City University of New York, 2007.
[7] M. Dehn, Über unendliche diskontinuierliche Gruppen, *Math. Ann.* **71** (1911), no. 1, 116–144.
[8] M. du Sautoy, Polycyclic groups, analytic groups and algebraic groups, *Proc. Lond. Math. Soc. (3)* **85** (2002), no. 1, 62–92.
[9] B. Eick, When is the automorphism group of a virtually polycyclic group virtually polycyclic?, *Glasg. Math. J.* **45** (2003), no. 3, 527–533.
[10] B. Eick and D. Kahrobaei, Polycyclic groups: A new platform for cryptography, preprint (2004), http://arxiv.org/abs/math/0411077.
[11] B. Eick and G. Ostheimer, On the orbit-stabilizer problem for integral matrix actions of polycyclic groups, *Math. Comp.* **72** (2003), no. 243, 1511–1529.
[12] A. Fesenko, Vulnerability of cryptographic primitives based on the power conjugacy search problem in quantum computing, *Cybernet. Systems Anal.* **50** (2014), no. 5, 815–816.
[13] E. Formanek, Conjugate separability in polycyclic groups, *J. Algebra* **42** (1976), no. 1, 1–10.
[14] D. Garber, D. Kahrobaei and H. T. Lam, Length-based attack for polycyclic groups, *J. Math. Cryptol.* **9** (2015), 33–44.
[15] D. Garber, S. Kaplan, M. Teicher, B. Tsaban and U. Vishne, Length-based conjugacy search in the braid group, in: *Algebraic Methods in Cryptography* (Bochum/Mainz 2005), Contemp. Math. 418, American Mathematical Society, Providence (2006), 75–87.
[16] V. Gebhardt, Efficient collection in infinite polycyclic groups, *J. Symbolic Comput.* **34** (2002), no. 3, 213–228.
[17] D. Grigoriev and V. Shpilrain, Zero-knowledge authentication schemes from actions on graphs, groups, or rings, *Ann. Pure Appl. Logic* **162** (2010), 194–200.
[18] M. Habeeb, D. Kahrobaei and V. Shpilrain, A secret sharing scheme based on group presentations and the word problem, in: *Computational and Combinatorial Group Theory and Cryptography* (Las Vegas/Ithaca 2011), Contemp. Math. 582, American Mathematical Society, Providence (2012), 143–150.
[19] P. Hall, *The Edmonton Notes on Nilpotent Groups*, Queen Mary College Math. Notes, Queen Mary College, London, 1969.
[20] D. F. Holt, B. Eick and E. A. O'Brien, *Handbook of Computational Group Theory*, Discrete Math. Appl. (Boca Raton), Chapman & Hall/CRC, Boca Raton, 2005.
[21] J. Hughes and A. Tannenbaum, Length-based attacks for certain group based encryption rewriting systems, preprint (2003), https://arxiv.org/abs/cs/0306032.
[22] G. Ivanyos, L. Sanselme and M. Santha, An efficient quantum algorithm for the hidden subgroup problem in nil-2 groups, in: *LATIN 2008 – Theoretical Informatics* (Buzios 2008), Lecture Notes in Comput. Sci. 4957, Springer, Berlin (2008), 759–771.
[23] D. Kahrobaei and B. Khan, Nis05-6: A non-commutative generalization of ElGamal key exchange using polycyclic groups, in: *IEEE Global Telecommunications Conference* (GLOBECOM '06), IEEE Press, Piscataway (2006), 1–5.
[24] D. Kahrobaei and C. Koupparis, Non-commutative digital signatures using non-commutative groups, *Groups Complex. Cryptol.* **4** (2012), 377–384.
[25] 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.
[26] M. Kotov and A. Ushakov, Analysis of a certain polycyclic-group-based cryptosystem, *J. Math. Cryptol.* **9** (2015), no. 3, 161–167.

[27] C. R. Leedham-Green and L. H. Soicher, Collection from the left and other strategies, *J. Symbolic Comput.* **9** (1990), no. 5–6, 665–675.
[28] E. Lo and G. Ostheimer, A practical algorithm for finding matrix representations for polycyclic groups, *J. Symbolic Comput.* **28** (1999), no. 3, 339–360.
[29] A. Mal'cev, On homomorphisms onto finite groups, *Trans. Amer. Math. Soc.* **119** (1983), 67–79.
[30] J. Milnor, Growth of finitely generated solvable groups, *J. Differential Geom.* **2** (1968), no. 4, 447–449.
[31] A. D. Myasnikov and A. Ushakov, Length-based attack and braid groups: Cryptanalysis of Anshel–Anshel–Goldfeld key-exchange protocol, in: *Public Key Cryptography – PKC 2007* (Beijing 2007), Lecture Notes in Comput. Sci. 4450, Springer, Berlin (2007), 76–88.
[32] A. G. Myasnikov and V. Roman'kov, A linear decomposition attack, *Groups Complex. Cryptol.* **7** (2015), no. 1, 81–94.
[33] A. G. Myasnikov, V. Shpilrain, A. Ushakov and N. Mosina, *Non-Commutative Cryptography and Complexity of Group-Theoretic Problems*, Math. Surveys Monogr. 177, American Mathematical Society, Providence, 2011.
[34] A. G. Myasnikov and A. Ushakov, Random subgroups and analysis of the length-based and quotient attacks, *J. Math. Cryptol.* **2** (2008), no. 1, 29–61.
[35] W. Nickel, Matrix representations for torsion-free nilpotent groups by Deep Thought, *J. Algebra* **300** (2006), no. 1, 376–383.
[36] V. Remeslennikov, Conjugacy in polycyclic groups, *Algebra Logic* **8** (1969), no. 6, 404–411.
[37] V. Roman'kov, The twisted conjugacy problem for endomorphisms of polycyclic groups, *J. Group Theory* **13** (2010), no. 3, 355–364.
[38] D. Segal, Decidable properties of polycyclic groups, *Proc. Lond. Math. Soc. (3)* **61** (1990), no. 3, 61–497.
[39] P. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in: *35th Annual Symposium on Foundations of Computer Science*, IEEE Press, Piscataway (1994), 124–134.
[40] V. Shpilrain, Search and witness problems in group theory, *Groups Complex. Cryptol.* **2** (2010), no. 2, 231–246.
[41] V. Shpilrain and A. Ushakov, The conjugacy search problem in public key cryptography: Unnecessary and insufficient, *Appl. Algebra Engrg. Comm. Comput.* **17** (2006), no. 3–4, 285–289.
[42] V. Shpilrain and A. Ushakov, An authentication scheme based on the twisted conjugacy problem, in: *Applied Cryptography and Network Security*, Lecture Notes in Comput. Sci. 5037, Springer, Berlin (2008), 366–372.
[43] V. Shpilrain and G. Zapata, Using the subgroup membership search problem in public key cryptography, in: *Algebraic Methods in Cryptography* (Bochum/Mainz 2005), Contemp. Math. 418, American Mathematical Society, Providence (2006), 169–178.
[44] B. Tsaban, Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography, *J. Cryptology* **28** (2015), 601–622.
[45] B. Wehrfritz, Two remarks on polycyclic groups, *Bull. Lond. Math. Soc.* **26** (1994), no. 6, 543–548.
[46] J. Wolf, Growth of finitely generated solvable groups and curvature of Riemannian manifolds, *J. Differential Geom.* **2** (1968), 421–446.