Cryptanalysis of protocols using (Simultaneous) Conjugacy Search Problem in certain Metabelian Platform Groups

Delaram Kahrobaei, Carmine Monetta, Ludovic Perret, Maria Tota, Martina Vigorito

Cryptanalysis of protocols using (Simultaneous) Conjugacy Search Problem in certain Metabelian Platform Groups

Delaram Kahrobaei[1,2,3,5], Carmine Monetta[4], Ludovic Perret[6], Maria Tota[4], Martina Vigorito[4]

September 26, 2023

[1] Department of Computer Science, University of York, UK
[2] Departments of Computer Science and Mathematics, Queens College, City University of New York, USA
[3] Department of Computer Science and Engineering, Tandon School of Engineering, New York University, USA
[4] Department of Mathematics, University of Salerno, IT
[5] Initiative for the Theoretical Sciences, Graduate Center, City University of New York, USA
[6] Sorbonne University, CNRS, LIP6, PolSys, Paris, France

Abstract

There are many group-based cryptosystems in which the security relies on the difficulty of solving Conjugacy Search Problem (CSP) and Simultaneous Conjugacy Search Problem (SCSP) in their underlying platform groups. In this paper we give a cryptanalysis of these systems which use certain semidirect product of abelian groups.

1 Introduction

The field of group-based cryptography began with the seminal work of Anshel, Anshel and Goldfeld in 1999 when they proposed a commutator key-exchange protocol based on the difficulty of simultaneus conjugacy search problem in certain classes of groups, namely braid groups [1]. The search for the platform group for this protocol has been an active area including several cryptanalysis. For a survey on group-based cryptography in the quantum era see [13] and book [14]. Polycyclic group-based cryptography was introduced by Eick and Kahrobaei in [5]. More precisely, they proposed such groups as platform for the Commutator Key-Exchange Protocol, also known as Anshel-Anshel-Goldfeld (a.k.a. AAG) [1], as well as for the non-commutative Diffie-Hellman Key-Exchange Protocol (a.k.a. Ko-Lee) [18]. The security of these protocols relies on

the difficulty to solve the Simultaneous Conjugacy Search Problem (SCSP) and the Conjgacy Search Problem (CSP) in some classes of groups. Their argument is based on experimental results for the CSP for certain metabelian polycyclic groups arising from field extensions. These groups are not virtually nilpotent, hence the CSP cannot be solved using the analysis provided in [20]. Nevertheless, some of these groups can be avoided as platform since, in [19], Kotov and Ushakov did a cryptanalysis for some groups of this type. A connected work is due to Gryak, Kahrobaei, and Martinez Perez who investigated another class of metabelian groups. Indeed, in [10] they obtain a complexity result concerning the CSP which is proved to be at most exponential for the analyzed class of groups.

The methods used to test conjugacy decision problem are different and include experiments conducted with machine learning algorithms, as done by Gryak, Kahrobaei and Haralick, in [8], but also Length-based attack. Garber, Kahrobaei, and Lam, in [7], showed that the Length-based attack is inefficient for certain classes of metabelian polycyclic groups.

There are other proposed cryptosystems based on the difficulty of CSP in certain classes of groups, (see the survey by Gryak and Kahrobaei [9]), for example Kahrobaei-Koupparis Digital Signature Scheme [16], and Khan-Kahrobaei Non-commutative El Gamal Key-exchange [15].

In this paper we go further to the results Field-Based-Attack (FBA) in [19] and show how to cryptanalyze the CSP and SCSP for some other classes of metabelian groups.

The authors in [19] investigated security properties of the Commutator Key-Exchange Protocol used with certain polycyclic groups. They showed that despite low success of the length based attack the protocol can be broken by a deterministic polynomial-time algorithm. They call this approach FBA and they implemented it in GAP to compare LBA and FBA.

In this paper we show that FBA could be generalized for protocols based on the difficulty of CSP and SCSP in certain classes of metabelian groups. In particular we prove the followings theorems:
Theorem 4.1: Let $G = M \ltimes N$, where $M \cong \mathbb{Z}^n$ and $N = \mathbb{Z}[m_1^{\pm 1}, \ldots, m_n^{\pm 1}]$ (as additive groups), with $m_1, \ldots, m_n$ positive integers, then there exists a polynomial-time algorithm to break Commutator Key-Exchange protocol for such a group G.
Theorem 4.2: Let $G = M \ltimes N$, where $M \cong \mathbb{Z}^n$ and $N = \mathbb{Z}[m_1^{\pm 1}, \ldots, m_n^{\pm 1}]$ (as additive groups), with $m_1, \ldots, m_n$ positive integers, then there exists a polynomial-time algorithm to break Diffie-Hellmann Key-Exchange protocol for such a group G.

This paper is structured as follow: in Section 2, we recall the definitions

of Conjugacy Search Problem and Simultaneous Conjugacy Search Problem and we describe some Key-Exchange Protocols such as Non-commutative Diffie-Hellman and nshel-Anshel-Goldfeld Commutator. Section 3 presents the family of metabelian groups we are interested in with some examples. In Section 4 we prove the main result i.e. how to cryptanalyze the CSP and SCSP in such platform groups and we provide the proofs of Theorem 4.1 and Theorem 4.2. The conclusions of our work are in Section 5.

2 Background

2.1 (Simultaneous) Conjugacy Search Problem

We start out by giving a brief description of two group-theoretic algorithmic problems on which the security of a number of protocols is based. Here and in the following, if x and g are group-elements, the conjugate of g by x, which is denoted by g^x, is the element $x^{-1}gx$.

The Conjugacy Search Problem (CSP): Let G be a finitely presented group such that the conjugacy decision problem is solvable. Given $g \in G$ and $h = g^x$ for some $x \in G$, the *Conjugacy Search Problem* asks to search such an element $x \in G$.

The Simultaneous Conjugacy Search Problem (SCSP): Given a finitely presented group G and $g_1, \ldots, g_n$, $h_1, \ldots, h_n$ elements of G such that $h_i = g_i^x$, for all $i \in \{1, \ldots, n\}$ and some $x \in G$, the *Simultaneous Conjugacy Search Problem* asks to recover such an element $x \in G$.

Please note that CSP and SCSP are always solvable since we assume that the decision conjugacy problem is solvable in the definitions of these problems. Also, a solution of $g^x = h$ is not unique. In fact, given a solution x, the set of solutions is $\{ax \ : \ a \in C_G(g)\}$.
Examples of well known protocols whose security is based on the difficulty of solving the CSP or the SCSP are the non-commutative Diffie-Hellman (a.k.a Ko-Lee) Key-Exchange Protocol and the Anshel-Anshel-Goldfeld Commutator Key-Exchange Protocol. We recall these protocols below.

2.2 Non-commutative Diffie-Hellman (a.k.a. Ko-Lee) Key Exchange Protocol

Originally proposed by Ko, Lee, et al. [18] 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 such a protocol is based on the difficulty to get a and b, which are private, from public information g, g^a and g^b. That is to solve the conjugacy equations

$$g^x = h \qquad \text{and} \qquad g^y = h'$$

where $h = g^a$ and $h' = g^b$. In other words, the security of Ko-Lee rests upon solving the conjugacy search problem within the subgroups A, B.

2.3 Anshel-Anshel-Goldfeld Commutator (a.k.a. AAG) Key-Exchange Protocol

The Anshel-Anshel-Goldfeld Commutator Key-Exchange Protocol [1] is a two-party protocol performed as follows:

- Fix a finitely presented group G, called the platform group, a set of generators $g_1, \ldots, g_k$ for G and some positive integers n_1, n_2, l, m. All this information are made public.
- Alice prepares a tuple of elements $\bar{a} = (a_1, \ldots, a_{n_1})$ called Alice's public tuple. Each a_i is generated randomly as a product of g_i's and their inverses.
- Bob prepares a tuple of elements $\bar{b} = (b_1, \ldots, b_{n_2})$ called Bob's public tuple. Each b_i is generated randomly as a product of g_i's and their inverses.
- Alice generates a random element A as a product $a_{s_1}^{\epsilon_1} \cdots a_{s_l}^{\epsilon_l}$ of a_i's and their inverses. The element A (or more precisely its factorization) is called the Alice's private element.
- Bob generates a random element B as a product $b_{t_1}^{\delta_1} \cdots b_{t_m}^{\delta_m}$ of b_i's and their inverses. The element B (or more precisely its factorization) is called the Bob's private element.
- Alice publishes the tuple of conjugates $\bar{b}^A = (A^{-1} b_1 A, \ldots, A^{-1} b_{n_2} A)$.
- Bob publishes the tuple of conjugates $\bar{a}^B = (B^{-1} a_1 B, \ldots, B^{-1} a_{n_1} B)$.
- Finally, Alice computes the element K_A as a product:

  $$A^{-1}(B^{-1} a_{s_1}^{\epsilon_1} B \cdots B^{-1} a_{s_l}^{\epsilon_l} B) = A^{-1} B^{-1} A B = [A, B]$$

  using the elements of Bob's conjugate tuple $\bar{a}^B$.

- Similarly, Bob computes the element K_B as a product:

$$(A^{-1}b_{t_1}^{\delta_1}A\cdots A^{-1}b_{t_m}^{\delta_m}A)^{-1}B = A^{-1}B^{-1}AB = [A,B]$$

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

- The shared key is then $K = K_A = K_B = [A,B]$.

The security of such a protocol is based on the fact that it is difficult to recover A and B from $\bar{a}, \bar{b}, \bar{b}^A$ and $\bar{a}^B$, which are public. In practice, if $\bar{b}^A = (b'_1, \ , b'_{n_2})$ and $\bar{a}^B = (a'_1, \ldots, a'_{n_1})$, it is achieved by solving a system of conjugacy equations for A and B:

$$\begin{cases} X^{-1}b_1X = b'_1 \\ \cdots \\ X^{-1}b_{n_2}X = b'_{n_2} \end{cases} \tag{1}$$

$$\begin{cases} Y^{-1}a_1Y = a'_1 \\ \cdots \\ Y^{-1}a_{n_1}Y = a'_{n_1} \end{cases} \tag{2}$$

This means that the security of AAG rests upon solving the simultaneous conjugacy search problem in G.

3 Examples of Metabelian Groups

Here we describe some families of metabelian groups whose CSP and SCSP will be discussed in the next section. To be more precise, we are interested in groups G of the form $G = M \ltimes N$, with both groups M and N abelian. We use multiplicative notation for the whole group G but additive notation for N. So if $s \in M$ and $c \in N$, the action of the element s maps c to

$$c \cdot s \text{ with additive notation or,}$$

$$c^s = s^{-1}cs \text{ with multiplicative notation.}$$

This kind of groups are metabelian and arise quite naturally in linear algebra and ring theory, as we will show in more details in the following examples.

Example 3.1. In [19], Kotov and Ushakov studied the security of AAG protocol for some polycyclic platform groups. More precisely they considered the group M as the multiplicative group of a specific field F and the group N as the additive group of the same field F; hence $G = F^* \ltimes F$. To construct F they considered an irreducible monic polynomial $f(x) \in \mathbb{Z}[x]$ and put:

$$F = \mathbb{Q}[x]/(f). \tag{3}$$

If $a \in F^*$ and $b \in F$, the action of a maps b to $b \cdot a$. They showed that in such a group it is possible to reduce the systems (1) and (2) to two systems of linear equations over the field F. Then there exist conditions under which each system has a unique solution.

Example 3.2. Let $V(+,\cdot)$ be a vector space over a field F. Take the group M as the multiplicative group F^* of F and the group N as the additive group of V. If $\lambda \in F^*$ and $v \in V$, the action of λ maps v to $v \cdot \lambda$. Hence $G = F^* \ltimes V$ has the same structure of the general group we considered before. Notice that, for $V = F$, if F is of the form described in (3) we obtain the same example we found in [19]. Similarly we could start with a module over a commutative unitary ring.

Such examples are interesting from a mathematical point of view but more practical examples, as they have been described in [10], follow.

Example 3.3. Split metabelian groups of finite Prüfer rank. We will focus in the case when the group G is given by a presentation of the form

$$G = \langle q_1, \ldots, q_n, b_1, \ldots, b_s \mid [q_i, q_j] = 1, [b_l, b_t] = 1, b_i^{q_l} = q_l b_i q_l^{-1} = b_1^{m_{1i}} b_2^{m_{2i}} \ldots b_s^{m_{si}} \rangle.$$

Observe that $q_1, \ldots, q_n$ generate a free abelian group which we denote by M and $b_1, \ldots, b_s$ generate the abelian group N as normal subgroup of G. Then $G = M \ltimes N$. Under these conditions one can show that there is an embedding $N \to \mathbb{Q}^s$ mapping the family $b_1, \ldots, b_s$ to a free basis of $\mathbb{Q}^s$. This means that our group is torsion free metabelian of finite Prüfer rank (meaning that the number of generators needed to generate any finitely generated subgroup is bounded). Observe that the action of M on N can be described using integer matrices: the action of q_l is encoded by the $(s \times s)$-matrix M_l with columns $m_{1i}, \ldots, m_{s_i}$. Moreover G is polycyclic if and only if the matrices M_i can be taken to be integer matrices with integral inverses [3].

One of the main advantages of these groups is that they admit the following fairly simple set of normal forms:

$$q_1^{\alpha_1} \ldots q_n^{\alpha_n} b_1^{\beta_1} \ldots b_s^{\beta_s} q_1^{\gamma_1} \ldots q_n^{\gamma_n}.$$

with $\gamma_1, \ldots, \gamma_n > 0$. Moreover there is an efficient algorithm (collection) to transform any word in the generators to the corresponding normal form: given an arbitrary word in the generating system, move all of the instances of q_i with negative exponent to the left and all the instances of q_i with positive exponents to the right.

Example 3.4. Generalized metabelian Baumslag-Solitar groups. Let $m_1, \ldots, m_n$ be positive integers. We call the group given by the following presentation a *generalized metabelian Baumslag-Solitar group*

$$G = \langle q_1, \ldots, q_n, b \mid [q_i, q_j] = 1, b^{q_i} = b^{m_i}, i, j = 1, \ldots, n \rangle.$$

It is a constructible metabelian group of finite Prüfer rank and $G \cong M \ltimes N$ with $M = \langle q_1, \ldots, q_n \rangle \cong \mathbb{Z}^n$ and $N = \mathbb{Z}[m_1^{\pm 1}, \ldots, m_n^{\pm 1}]$ (as additive groups). In [10], the authors showed the CSP in such groups reduce to the Discrete Logarithm Problem.

Example 3.5. Let $L : \mathbb{Q}$ be a Galois extension of degree n and fix an integer basis $\{u_1, \ldots, u_k\}$ of L over $\mathbb{Q}$. Then $\{u_1, \ldots, u_k\}$ freely generates the maximal order $\mathcal{O}_L$ as a $\mathbb{Z}$-module.

Now, we choose integer elements, $q_1, \ldots, q_n$, generating a free abelian multiplicative subgroup of $L - \{0\}$. Each q_i acts on L by left multiplication and using the basis $\{u_1, \ldots, u_k\}$, we may represent this action by means of an integer matrix M_i. Let N be the smallest sub $\mathbb{Z}$-module of L closed under multiplication with the elements q_i and q_i^{-1} and such that $\mathcal{O}_L \subseteq N$, i.e.,

$$N = \mathcal{O}_L[q_1^{\pm 1}, \ldots, q_n^{\pm 1}].$$

We then may define $G = M \ltimes N$ where the action of M on N is given by multiplication by the elements q_i. The generalized Baumslag-Solitar groups of the previous example are a particular case of this situation for $L = \mathbb{Q}$. If the elements q_i lie in $\mathcal{U}_L$ which is the group of units of $\mathcal{O}_L$, then the group G is polycyclic.

4 Cryptanalysis of the Commutator and the Non-Commutative Diffie-Hellman key exchange Protocols

In this section, we show that the AAG and the Ko-Lee Key Exchange Protocols are not suitable in the case of the generalised metabelian Baumslag-Solitar groups (Example 3.4). Similar arguments can be used with minor modifications for the other examples in Section 3.

We begin studying the CSP and SCSP in a metabelian group of the form $G = M \ltimes N$, as described in Section 3. Assume that we have conjugated elements $g, h \in G$ and we want to solve the CSP for g, h, i.e., we want to find $x \in G$ such that

$$g^x = h.$$

We put $g = sc$, $h = s'c'$ and $x = td$, where $s, s', t \in M$ and $c, c', d \in N$. Then

$$g^x = x^{-1}gx = d^{-1}t^{-1}sctd = d^{-1}st^{-1}ctd = s(d^{-1})^s c^t d.$$

Now $g^x = h$ implies $s' = s$ and $c' = (d^{-1})^s c^t d$. Since the element $(d^{-1})^s c^t d$ belongs to N we can write it additively as

$$-d \cdot s + c \cdot t + d = d \cdot (1 - s) + c \cdot t.$$

This means that the CSP above is equivalent to the problem of finding $t \in M$ and $d \in N$ such that

$$d \cdot (1 - s) + c \cdot t = c', \tag{4}$$

where $s \in M$ and $c, c' \in N$ are given.

In particular, if we need to face the SCSP, which means to solve system (1), we can apply the reduction process described above. Then, if we put $b_i = s_i c_i$, $b'_i = s'_i c'_i$ with $s_i, s'_i \in M$ and $c_i, c'_i \in N$, for all $i \in \{1, \dots, n_2\}$, and $X = td$ with $t \in M$, $d \in N$ we will get the following system of equations

$$\begin{cases} d \cdot (1 - s_1) + c_1 \cdot t = c'_1 \\ \dots \\ d \cdot (1 - s_{n_2}) + c_{n_2} \cdot t = c'_{n_2} \end{cases} \tag{5}$$

where $s_i \in M$ and $c_i, c'_i \in N$ are given and we need to find $t \in M$ and $d \in N$.

Then the next results follow.
We start analyzing the cryptanalysis of AAG protocol in a generalized metabelian Baumslag-Solitar groups, as described in Example 3.4.

Theorem 4.1. *Let $G = M \ltimes N$, where $M \cong \mathbb{Z}^n$ and $N = \mathbb{Z}[m_1^{\pm 1}, \dots, m_n^{\pm 1}]$ (as additive groups), with $m_1, \dots, m_n$ positive integers, then there exists a polynomial-time algorithm to break Commutator Key-Exchange protocol for such a group G.*

Proof. In AAG protocol the attacker knows $b_1^X, b_2^X, \dots, b_{n_2}^X$ for some $b_1, \dots, b_{n_2}$ (which are public) and $n_2 > 1$. To find $X = td$, with $t \in M$ and $d \in N$, the attacker has to solve several equations as (5). Let us consider two of them

$$d \cdot (1 - s) + c \cdot t = c'$$
$$d \cdot (1 - \tilde{s}) + \tilde{c} \cdot t = \tilde{c}'.$$

Here $s, \tilde{s}, c, \tilde{c}, c', \tilde{c}'$ are known and the attacker has to find t and d. Recall that $c', \tilde{c}', c, \tilde{c}, d$ lie in N which is a subring of $\mathbb{Q}$. If we identify s and t with the integer they act by, then they also lie in N. So the above can be seen as a system of two equations in N, moreover we know a priori that the system has a solution. This means that unless the second equation is a multiple of the first one, this solution is unique and the standard procedure to solve the system yields then the suitable value of t and d in polynomial time (see [10, Section 2]). □

The argument in the previous proof applies also when G is as described in Example 3.2, choosing $V = F^n$ with $n \in \mathbb{N}$.

Next, let us move to the non-commutative Diffie-Hellmann key exchange protocol (Section 2.2).

Theorem 4.2. *Let $G = M \ltimes N$, where $M \cong \mathbb{Z}^n$ and $N = \mathbb{Z}[m_1^{\pm 1}, \dots, m_n^{\pm 1}]$ (as additive groups), with $m_1, \dots, m_n$ positive integers, then there exists a polynomial-time algorithm to break Diffie-Hellmann Key-Exchange protocol for such a group G.*

Proof. In Ko-Lee protocol the main problem is that Alice and Bob must agree on a set Ω of pairwise commuting elements and then choose their conjugators a and b from that set. Recall that we are denoting $G = M \ltimes N$. As M is abelian a possible choice would be $\Omega = M$, and if a lies in M then the attacker can find a from g^a in polynomial time. Another possibility would be to choose $a, b \in N$. But then $a = d$ and equation (4) for g^a is

$$d \cdot (1 - s) + c = c',$$

and the only unknown is d which can be found easily in polynomial time (see [10, Section 3]).

In the case when a is an arbitrary element not in M or N, Ω must be a subset of the centralizer $C_G(a)$ of a in G.

Things are particularly easy in the case when the element a belongs to M^r for some $r \in N$, which happens if and only if

$$a = td = t_1^r = r^{-1} t_1 r = t_1 t_1^{-1} r^{-1} t_1 r = t_1 r^{-t_1} r,$$

for some $t, t_1 \in M$ and $d \in N$. Additively this is equivalent to

$$d = r - r \cdot t = r \cdot (1 - t).$$

It is a standard fact that $M^r = \{x\delta(x) \mid x \in M\}$ where δ is the inner derivation given by $\delta(x) = r \cdot (1 - x)$. In this case it is easy to check that

$$\Omega \subseteq C_G(a) = M^r.$$

If the attacker has the extra information that a belongs to M^r for some r, then the equation that he has to solve is

$$r \cdot (1 - t)(1 - s) + c \cdot t = c'$$

equivalently

$$(c - r + r \cdot s) \cdot t = c' - r + s \cdot r.$$

This can be seen as an equation in $\mathbb{Q}$ and only requires to perform the quotient of $c' - r + r \cdot s$ by $c - r + r \cdot s$ thus can be solved in polynomial time (see [10, Section 3]).

Moreover, we are going to see now that by embedding our group G in a bigger group we may always assume that a lies in some conjugated subgroup of M. Let $\tilde{G} = M \ltimes \tilde{N}$ where $\tilde{N} = N \otimes_{\mathbb{Z}} \mathbb{Q} = \mathbb{Q}$. Then $a = td$ lies in M^r for some $r \in \tilde{N}$ if and only if $d = r \cdot (1 - t)$. This can always be solved in $\mathbb{Q}$, in other words, we can always find a suitable $r \in \mathbb{Q}$. Then one proceeds as we did before with this r. The fact that r might not belong to N does not create any troubles: recall that we are dealing not with the conjugacy problem but with the conjugacy *search* problem, meaning that we know a priory that our equations have a solution so the procedure above yields the right values of t, d even if r does not belong to N.

Observe that behind what we said above is the fact that for the group $\tilde{G}$, the first cohomology group $\mathrm{H}^1(\mathrm{M}, \tilde{\mathrm{N}})$ is zero, thus all the complements of $\tilde{N}$ in $\tilde{G}$ are conjugated. □

Notice that exactly the same argument in the previous proof happens for any group $G = M \ltimes N$ with $N \subseteq \mathbb{Q}^n$ for some n, so can be extended to the more general version of our groups (Example 3.3).

5 Conclusion

In this paper we do cryptanalysis for the CSP and SCSP in certain metabelian groups. In particular we show the following.

1. The generalized metabelian Baumslag-Solitar groups can not be used as platform groups in commutator key-exchange protocol.

2. The generalized metabelian Baumslag-Solitar groups can not be used as platform groups in non-commutative Diffie-Hellman protocol.

Finally we want to point out that this cryptanalysis could be extended to the other examples in Section 3 and to all cryptosystems based on the difficulty of CSP and SCSP.

Acknowledgement

DK thanks the University of Salerno (Italy), where most of this paper was discussed and written. We thank Professor Conchita Martinez-Perez for fruitful discussions. MV thanks Initiative for the Theoretical Sciences at CUNY GC which hosted her Fall 2022.

References

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

[2] I. Anshel, D. Atkins, D. Goldfeld, and P. E. Gunnells, *WalnutDSA: a group theoretic digital signature algorithm*, International Journal of Computer Mathematics: Computer Systems Theory 6, 260–284, 2021.

[3] L. Auslander, *On a problem of Philip Hall*, Annals of Mathematics 86(1), 112–116, 1967.

[4] K. Boudgoust, *Polycyclic groups and applications in cryptography*, Master's Thesis, Karlsruhe Institute of Technology, https://katinkabou.github.io/Documents/Masterarbeit.pdf, 1–73, 2018.

[5] B. Eick and D. Kahrobaei, *Polycyclic groups: A new platform for cryptology?*, 2004.

[6] V. P. Elizarov, *Necessary conditions for solvability of a system of linear equations over a ring*, Discrete Math. Appl., Vol. 14, No. 2, 153–162, 2004.

[7] D. Garber, D. Kahrobaei, and H. T. Lam, *Length Based attack for Polycyclic Groups*, Journal of Mathematical Cryptology, De Gruyter, 33–44, 2015.

[8] J. Gryak, R. Haralick, and D. Kahrobaei, *Solving the Conjugacy Decision Problem via Machine Learning*, Experimental Mathematics, Taylor & Francis **29**, 66–78, 2020.

[9] J. Gryak and D. Kahrobaei, *The Status of the Polycyclic Group-Based Cryptography: A Survey and Open Problems*, Groups Complexity Cryptology, De Gruyter, Volume **8**, Issue 2, 171–186, 2016.

[10] J. Gryak, D. Kahrobaei, and C. Martinez-Perez, *On the conjugacy problem in certain metabelian groups*, Glasgow Mathematical Journal, Cambridge University Press **61**, Issue 2, 251–269, 2019.

[11] J. A. Hermida and T. Sanchez-Giralda, *Linear equations over commutative rings and determinantal ideals*, Journal Of Algebra 99, 72–79, 1986.

[12] D. Hofheinz and R. Steinwandt, *A practical attack on some braid group based cryptographic primitives*, Springer, In Advances in Cryptology, volume 2567 of Lecture Notes Comp. Sc., 187–198, 2003.

[13] D. Kahrobaei, R. Flores and M. Noce, *Group-based Cryptography in the Quantum Era*, Notices of the American Mathematical Society, 752–763, 2023.

[14] D. Kahrobaei, R. Flores, M. Noce, M. Habeeb, and C. Battarbee, *Applications of Group Theory in Cryptography*, American Mathematical Society, The Mathematical Surveys and Monographs series of the American Mathematical Society, 2023, to appear.

[15] D. Kahrobaei and B. Khan, *Nis05-6: A non-commutative generalization of ElGamal key exchange using polycyclic groups*, IEEE Global Telecommunications Conference (GLOBECOM '06), IEEE Press, Piscataway, 1–5, 2006.

[16] D. Kahrobaei and C. Koupparis, *Non-commutative digital signatures using non-commutative groups*, Groups, Complexity, Cryptology **4**, 377–384, 2012.

[17] R. Kannan and A. Bachem, *Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix*, SIAM J. Compt. 8, 8(4), 499–507, 1979.

[18] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang, and C. Park, *New public-key cryptosystem using braid groups*, Advances in cryptology, CRYPTO 2000, vol. 1880, 166–183, 2000.

[19] M. Kotov and A. Ushakov, *Analysis of a certain polycyclic-group-based cryptosystem*, Journal of Mathematical Cryptology, vol. 9, 161–167, 2015.

[20] C. Monetta and A. Tortora, *The multiple conjugacy search problem in virtually nilpotent polycyclic groups*, Advances in Group Theory and Applications, vol. 13, 61-70, 2022.

[21] V. Shpilrain and A. Ushakov, *The conjugacy search problem in public key cryptography: unnecessary and insufficient*, Appal. Algebra Eng. Comm. Comput., 285–289, 2006.