CRYPTANALYSIS VIA ALGEBRAIC SPANS

ADI BEN-ZVI, ARKADIUS KALKA, AND BOAZ TSABAN

Abstract. We describe a new method for obtaining polynomial time solutions of problems in noncommutative algebraic cryptography. This method is easier to apply and more efficient than the linear centralizer method. We use it to provide a cryptanalysis of the Triple Decompostion key exchange protocol, the only classic group theory based key exchange protocol that was not cryptanalyzed thus far.

1. Introduction

Since Diffie and Hellman's 1976 key exchange protocol, few alternative proposals for key exchange protocols withstood cryptanalysis, all based on commutative algebraic structures. In 1999, Anshel, Anshel, and Goldfeld [1] introduced the *Commutator* key exchange protocol, a general method for constructing key exchange protocols based on *noncommutative* algebraic structures. Concurrently, Ko, Lee, Cheon, Han, Kang and Park [8] introduced the *Braid Diffie–Hellman* key exchange protocol, another general method achieving the same goal. The security of these protocols is based on variations of the conjugacy problem in groups. Both papers [1, 8] proposed to use Artin's braid group $\mathbf{B}_N$, a finitely presented, infinite noncommutative group parameterized by a natural number N, as the platform group.

The introduction of the Commutator key exchange protocol and the Braid Diffie–Hellman key exchange protocol was followed by a number of heuristic attacks (see references in [23]). These attacks were foiled by changing the distributions on the group [4, 22]. In the breakthrough papers [3, 23], polynomial time algorithms were found for the precise computational problems on which these, and a number of related key exchange protocols, are based. These algorithms constitute cryptanalyses of these key exchange protocols that do not depend on the distributions used to generate their keys. In a series of works ([14, 15, 16, 17], see also [13] and references therein), Roman'kov developed a provable polynomial time cryptanalysis method that applies to key exchange protocols with certain commuting substructures. He applied this method successfuly to a large number of group-theory based key exchange protocols.

The *Triple Decomposition* key exchange protocol was introduced by Kurt in 2006 [9, 10]. Its security is based on a problem very different from those of the above-mentioned key exchange protocols. The Triple Decomposition key exchange protocol is well known and appears in the first textbooks in the field [11, 12]. It is mentioned in [23] as a distinguished key exchange protocol that remains challenging.

We present a general approach for provable polynomial time solutions of computational problems in groups with efficient, faithful representation as matrix groups. This approach improves upon those of [3, 23], in simplicity of application and efficiency. This approach

Key words and phrases. noncommutative algebraic cryptography, group theory-based cryptography, braid-based cryptography, Triple Decomposition key exchange, Commutator key exchange, Centralizer key exchange, Braid Diffie–Hellman key exchange, linear cryptanalysis, algebraic span cryptanalysis.

covers all problems that were solved by earlier provable polynomial time methods. Moreover, with a novel view at the public information provided by the Triple Decomposition key exchange protocol, it provides the first cryptanalysis of this key exchange protocol.

History. The preliminary note [24] is incorporated into the present paper (Sections 1–3). Section 4 is joint work of all three authors. Comprehensive accounts of the field of group theory based cryptography are provided in the textbooks [12, 6].

Acknowledgements. We thank Avraham (Rami) Eizenbud and Craig Gentry for intriguing discussions. A part of this work was carried out while the third named author was in a Sabbatical leave at the Weizmann Institute of Science. This author thanks his hosts for their kind hospitality.

2. Algebraic span cryptanalysis in a nutshell

Let $\mathbb{F}$ be a finite field. For a set $S \subseteq \mathrm{M}_n(\mathbb{F})$, let $\mathrm{Alg}(S)$ be the algebra generated by S, that is, the smallest Algebra $A \subseteq \mathrm{M}_n(\mathbb{F})$ that contains S as a subset. Every subalgebra of $\mathrm{M}_n(\mathbb{F})$ is also a vector space over $\mathbb{F}$. For a group $G \leq \mathrm{GL}_n(\mathbb{F})$, we have that $\mathrm{Alg}(G) = \mathrm{span}(G)$, the vector space spanned by G. For simplicity we assume, throughout, that the dimension of the vector space $\mathrm{Alg}(G)$ is at least a positive constant times n. Notice that even for cyclic groups G, this is typically the case.

Throughout, let ω be the linear algebra constant, the minimal real number such that the complexity of $n \times n$ matrix multiplication is $O(n^\omega)$ field operations.

Proposition 1. *Let $G = \langle g_1, \ldots, g_k \rangle \leq \mathrm{GL}_n(\mathbb{F})$ be a group. A basis for the vector space $\mathrm{Alg}(G)$ can be computed in time $O(kd^2n^2)$, measured by number of field operations, where $d \leq n^2$ is the dimension of this vector space.*

Proof. Initialize a sequence $s = (I)$, the identity matrix, and $i := 1$. Repeat the following as long as there is an element in position i of the sequence s:

(1) For $j = 1, \ldots, k$, if $s_i g_j \notin \operatorname{span} S$, append $s_i g_j$ at the end of s.
(2) $i := i + 1$.

The resulting sequence S is a basis for $\operatorname{span} G$. Let d be the dimension of $\mathrm{Alg}(G)$. For each i and each j, the complexity of computing the products $s_i g_j$ is n^ω field operations. Assume that the matrices are stored in S in a vector form, and the matrix S is kept in Echelon normal form throughout the process. Since there are at most d vectors in S, each of length n^2, the complexity of checking whether a vector is in $\operatorname{span} S$ is at most $O(dn^2)$. Thus, the overall complexity is $O(kd(n^\omega + dn^2))$ field operations. Since we assume that d is at least a constant multiple of n, the second term dominates the first one. □

Proposition 1 holds, more generally, for semigroups of matrices; but this will not be used here. There are advanced methods, via representation theory, to slightly reduce the complexity of this computation [7].

Algebraic span cryptanalysis is applied as follows. Let $G_1, \ldots, G_k \leq \mathrm{GL}_n(\mathbb{F})$ be given, and $g_1 \in G_1, \ldots, g_k \in G_k$ be unknown elements. Assume that we have a system of linear equations (or constraints) on the entries of these unknown matrices, and we wish to find $f(g_1, \ldots, g_k)$ for some prescribed function f. Instead of solving the given linear equations subject to the restrictions $g_1 \in G_1, \ldots, g_k \in G_k$ (which may be computationally hard), we solve the linear equations subject to the *linear* constraints $g_1 \in \mathrm{Alg}(G_1), \ldots, g_k \in \mathrm{Alg}(G_k)$.

We then try to prove (or at least verify by experiments) that, for each solution $\tilde{g}_1, \ldots, \tilde{g}_k$, we have that $f(\tilde{g}_1, \ldots, \tilde{g}_k) = f(g_1, \ldots, g_k)$.

This method applies in all cases of noncommutative algebraic cryptography where polynomial-time algorithms are known [8, 23, 14, 15, 16, 17, 13], and in a case that was not cryptanalyzed thus far. We provide some details later.

The equations do not have to be given as linear. For example, an equation $g_1 a g_2 = b$ with a and b known can be transformed to the equation $a g_2 = g_1^{-1} b$, which is linear in the entries of g_1^{-1} and g_2. Also, if (as in the latter example) some elements in our solution have to be invertible, we may pick random solutions until they are. Since there *is* an invertible solution, namely, $(g_1, \ldots, g_k)$, we have by the Invertibility Lemma [23, Lemma 9] that random solutions will be invertible with probability bounded away from zero, provided that the field is not too small.

The next section provides concrete applications of this approach to several problems in the field of noncommutative algebraic cryptography. Enough examples are provided so that the reader can apply this method to additional problems in the field, including essentially all known key exchange protocols based on groups with efficient representations as matrix groups. The final section addresses a hitherto unsolved problem. In these examples, the proposed platform group is the braid group $\mathbf{B}_N$. However, the problems can be transformed into a matrix group $G \leq \mathrm{GL}_n(\mathbb{F})$, where $\mathbb{F}$ is a finite field of cardinality that is much to our choice [8, 23]. The reduction uses the the Lawrence–Krammer representation, and thus the matrices are of rank $n = \binom{N}{2}$. In this reduction, the cardinality of $\mathbb{F}$ is, roughly, $2^{M^3 N^2}$, for some length parameter M. We may assume that $M \approx N$. Then the cost of field multiplication is about N^5, ignoring a logarithmic factor. Tighter scrutiny of this reduction is likely to lead to substantially smaller field sizes; the extra factor of N^5 should not be considered definite.

3. Sample applications

3.1. **The Commutator key exchange protocol.** For a noncommutative group G and group elements $g, x \in G$, we use the notation $g^x = x^{-1} g x$. Useful identities involving this notation include $g^{xy} = (g^x)^y$, and $g^c = g$ for every element $c \in G$ that commutes with g, such that $cg = gc$.

The *Commutator key exchange protocol* [1] is described succinctly in Figure 1.[1] In some detail:

(1) A noncommutative group G and elements $a_1, \ldots, a_k, b_1, \ldots, b_k \in G$ are publicly given.[2]
(2) Alice and Bob choose free-group words in the variables $x_1, \ldots, x_k$, $v(x_1, \ldots, x_k)$ and $w(x_1, \ldots, x_k)$, respectively.[3]
(3) Alice substitutes $a_1, \ldots, a_k$ for $x_1, \ldots, x_k$, to obtain a secret element $a = v(a_1, \ldots, a_k) \in G$. Similarly, Bob computes $b = w(b_1, \ldots, b_k) \in G$.

[1] In our diagrams, green letters indicate publicly known elements, and red ones indicate secret elements, known only to their holders. Results of computations involving elements of both colors may be either publicly known, or secret, depending on the context.

[2] By adding elements, if needed, we assume that the number of elements a_i is equal to the number of elements b_i.

[3] A free group word in the variables $x_1, \ldots, x_k$ is a product of the form $x_{i_1}^{\epsilon_1} x_{i_2}^{\epsilon_2} \cdots x_{i_m}^{\epsilon_m}$, with $i_1, \ldots, i_m \in \{1, \ldots, k\}$ and $\epsilon_1, \ldots, \epsilon_m \in \{1, -1\}$, and with no subproduct of the form $x_i x_i^{-1}$ or $x_i^{-1} x_i$.

(4) Alice sends the conjugated elements ${b_1}^a, \ldots, {b_k}^a$ to Bob, and Bob sends ${a_1}^b, \ldots, {a_k}^b$ to Alice.
(5) The shared key is the *commutator* $a^{-1}b^{-1}ab$.

As conjugation is a group isomorphism, we have that

$$v({a_1}^b, \ldots, {a_k}^b) = v(a_1, \ldots, a_k)^b = a^b = b^{-1}ab.$$

Thus, Alice can compute the shared key $a^{-1}b^{-1}ab$ as $a^{-1}v({a_1}^b, \ldots, {a_k}^b)$, using her secret $a, v(x_1, \ldots, x_k)$ and the public elements ${a_1}^b, \ldots, {a_k}^b$. Similarly, Bob computes $a^{-1}b^{-1}ab$ as $w({b_1}^a, \ldots, {b_k}^a)^{-1}b$.

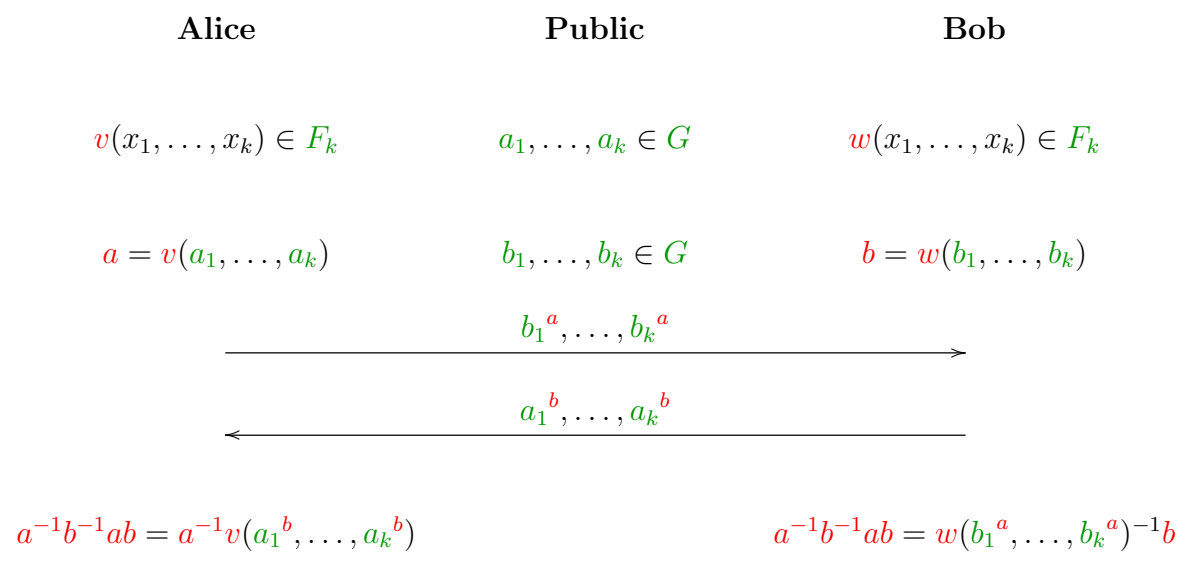

FIGURE 1. The Commutator key exchange protocol

The security of the Commutator key exchange protocol is determined by the difficulty of the following problem. As usual, for a group G and elements $g_1, \ldots, g_k \in G$, $\langle g_1, \ldots, g_k \rangle$ denotes the subgroup of G generated by $g_1, \ldots, g_k$.

Problem 2. *Let G be a group. Let $a_1, \ldots, a_k, b_1, \ldots, b_k \in G$. Let $a \in \langle a_1, \ldots, a_k \rangle, b \in \langle b_1, \ldots, b_k \rangle$.*
Given $a_1, \ldots, a_k, b_1, \ldots, b_k, a_1^b, \ldots, a_k^b, b_1^a, \ldots, b_k^a$, compute $a^{-1}b^{-1}ab$.

We solve the Problem 2 in matrix groups.

Lemma 3. *Let $x, \tilde{x} \in \mathrm{GL}_n(\mathbb{F})$ and $G = \langle g_1, \ldots, g_k \rangle \leq \mathrm{GL}_n(\mathbb{F})$. If ${g_i}^x = {g_i}^{\tilde{x}}$ for all $i = 1, \ldots, k$, then $g^x = g^{\tilde{x}}$ for all $g \in \mathrm{Alg}(G)$.*

Proof. Conjugation is an automorphism of the matrix algebra. □

We apply the algebraic span method to the Commutator key exchange protocol problem:

(1) Compute bases for the vector spaces $\mathrm{Alg}(A)$ and $\mathrm{Alg}(B)$. Let d be the maximum of the sizes of these bases.

(2) Solve the following homogeneous system of linear equations in the unknown matrix $x \in \mathrm{Alg}(A)$:

$$\begin{aligned} b_1 \cdot x &= x \cdot {b_1}^a \\ &\vdots \\ b_k \cdot x &= x \cdot {b_k}^a, \end{aligned}$$

a system of linear equations on the d coefficients determining x.
(3) Fix a basis for the solution space, and pick random solutions $\tilde{a}$ until $\tilde{a}$ is invertible.
(4) Solve the following homogeneous system of linear equations in the unknown matrix $y \in \mathrm{Alg}(B)$:

$$\begin{aligned} a_1 \cdot y &= y \cdot {a_1}^b \\ &\vdots \\ a_k \cdot y &= y \cdot {a_k}^b, \end{aligned}$$

a system of linear equations on the d coefficients determining y.
(5) Fix a basis for the solution space, and pick random solutions $\tilde{b}$ until $\tilde{b}$ is invertible.
(6) *Output:* $\tilde{a}^{-1}\tilde{b}^{-1}\tilde{a}\tilde{b}$.

That Step (3) terminates quickly follows from the Invertibility Lemma [23]. We show that the output is correct. As $\tilde{b} \in \mathrm{Alg}(B)$, we have by Lemma 3 that $\tilde{b}^{\tilde{a}} = \tilde{b}^a$, and therefore

$$(\tilde{b}^{-1})^{\tilde{a}} = (\tilde{b}^{\tilde{a}})^{-1} = (\tilde{b}^a)^{-1} = (\tilde{b}^{-1})^a.$$

It follows that

$$\tilde{a}^{-1}\tilde{b}^{-1}\tilde{a}\tilde{b} = (\tilde{b}^{-1})^{\tilde{a}}\tilde{b} = (\tilde{b}^{-1})^a\tilde{b} = a^{-1}\tilde{b}^{-1}a\tilde{b} = a^{-1}a^{\tilde{b}}.$$

As $a \in \mathrm{Alg}(A)$, we have by Lemma 3 that $a^{\tilde{b}} = a^b$, and thus

$$\tilde{a}^{-1}\tilde{b}^{-1}\tilde{a}\tilde{b} = a^{-1}a^b = a^{-1}b^{-1}ab.$$

The step with linear equations computes the nullspace of a $kn^2 \times d$ matrix. Thus, its complexity is $O(\frac{kn^2}{d}d^\omega) = O(kn^2d^{\omega-1})$, which is dominated by the complexity $O(kd^2n^2)$ of computing the algebraic spans.

3.2. **The Centralizer key exchange protocol.** For a group G and an element $g \in G$, the *centralizer of g in G* is the set

$$C_G(g) := \{h \in G : gh = hg\}.$$

The *Centralizer key exchange protocol*, introduced by Shpilrain and Ushakov in 2006 [20], is described in Figure 2. In this protocol, a_1 commutes with b_1 and a_2 commutes with b_2. Consequently, the keys computed by Alice and Bob are identical, and equal to $a_1b_1ga_2b_2$.

The security of the Centralizer key exchange protocol is determined by the difficulty of the following problem.

Problem 4. *Let $G \leq \mathrm{GL}_n(\mathbb{F})$. Assume that $g, a_1, b_2 \in G$, $g_1, \ldots, g_k \in C_G(a_1)$, $h_1, \ldots, h_k \in C_G(b_2)$, $a_2 \in \langle h_1, \ldots, h_k\rangle$, and $b_1 \in \langle g_1, \ldots, g_k\rangle$.*
Given $g, g_1, \ldots, g_k, h_1, \ldots, h_k, a_1ga_2, b_1gb_2$, compute $a_1b_1ga_2b_2$.

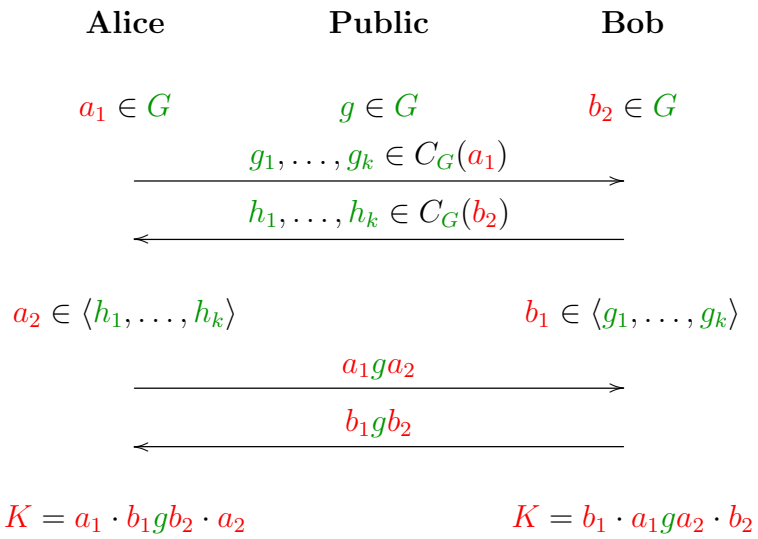

FIGURE 2. The Centralizer key exchange protocol

The algebraic span method applies to this problem: We note that $a_1^{-1}(a_1 g a_2) = g a_2$. Find a solution to the system

$$\begin{aligned} x(a_1 g a_2) &= gy \\ x g_1 &= g_1 x \\ &\vdots \\ x g_k &= g_k x \end{aligned}$$

with x invertible and $y \in \mathrm{Alg}(\{h_1, \ldots, h_k\})$. In practice, we may start with y which has d variables, and this determines x and then we solve for x.

Let $(\tilde{a}_1, \tilde{a}_2) = (x^{-1}, y)$. Then $\tilde{a}_1 g \tilde{a}_2 = x^{-1} g y = a_1 g a_2$. As $\tilde{a}_1 = x^{-1}$ commutes with $g_1, \ldots, g_k$, it commutes with b_1. As b_2 commutes with $h_1, \ldots, h_k$ and $\tilde{a}_2 \in \mathrm{Alg}(\{h_1, \ldots, h_k\})$, we have that $b_2 \tilde{a}_2 = \tilde{a}_2 b_2$. Thus,

$$\tilde{a}_1 b_1 g b_2 \tilde{a}_2 = b_1 \tilde{a}_1 g \tilde{a}_2 b_2 = b_1 a_1 g a_2 b_2.$$

Here, too, the complexity is $O(kd^2n^2)$.

3.3. **The Braid Diffie–Hellman key exchange protocol and the Double Coset key exchange protocol.** The *Braid Diffie–Hellman key exchange protocol*, introduced by Ko, Lee, Cheon, Han, Kang and Park [8], is illustrated in Figure 3. For subsets A, B of a group G, $[A, B] = 1$ means that a and b commute, that is, $ab = ba$ for all $a \in A$ and $b \in B$. Since, in the Braid Diffie–Hellman key exchange protocol, the subgroups A and B of G commute element-wise, the keys computed by Alice and Bob are identical.

The security of the Braid Diffie–Hellman key exchange protocol for a platform group G (Figure 3) is captured by the following problem.

Problem 5. *Let A and B be subgroups of $\mathrm{GL}_n(\mathbb{F})$ with $[A, B] = 1$ and $g \in \mathrm{GL}_n(\mathbb{F})$ be given. Given a pair (g^a, g^b) where $a \in A$ and $b \in B$, find g^{ab}.*

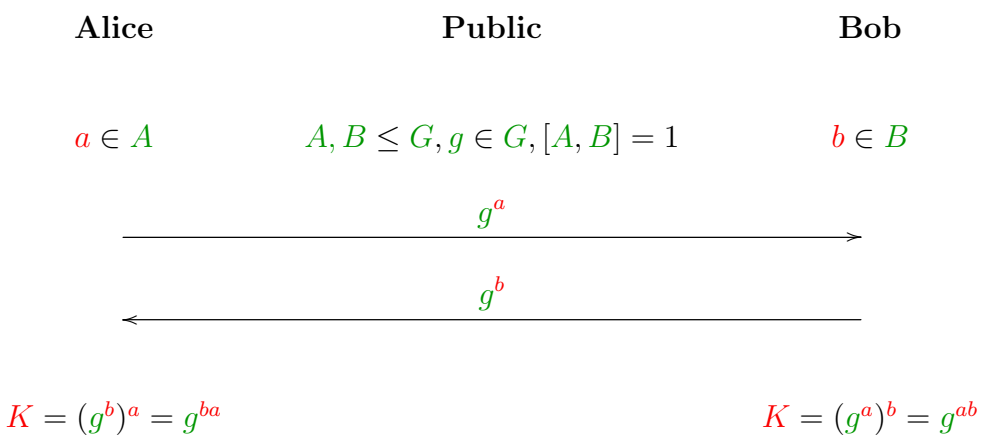

FIGURE 3. The Braid Diffie–Hellman key exchange protocol

To apply the algebraic span method to this problem, solve the equation $g^x = g^a$ subject to $x \in \mathrm{Alg}(A)$, finding an invertible solution $\tilde{a}$. Then

$$(g^b)^{\tilde{a}} = g^{b\tilde{a}} = g^{\tilde{a}b} = (g^{\tilde{a}})^b = (g^a)^b = g^{ab}.$$

Again, the complexity of the solution is dominated by the computation of $\mathrm{Alg}(A)$.

A generalization of the Braid Diffie–Hellman key exchange protocol was proposed by Cha, Ko, Lee, Han and Cheon [2]. A variation of this protocol was proposed in 2005, by Shpilrain and Ushakov [19]. These protocols are both special cases of the *Double Coset key exchange protocol*, illustrated in Figure 4.

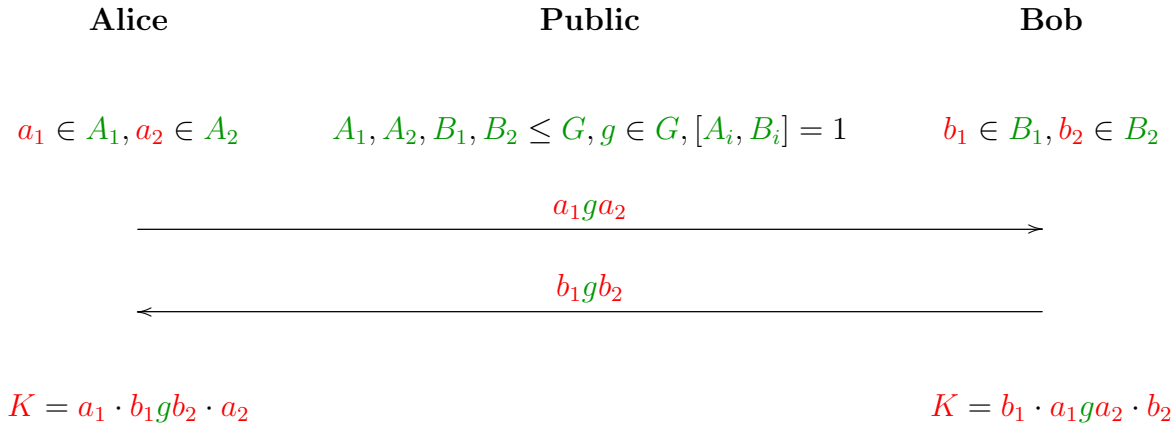

FIGURE 4. The Double Coset key exchange protocol

One may state the underlying problem as before. Here is how to solve it: Solve the equation $x_1(a_1 g a_2) = g x_2$ subject to $x_1 \in \mathrm{Alg}(A_1)$ and $x_2 \in \mathrm{Alg}(A_2)$, with x_1 invertible. Let $(\tilde{a}_1, a_2) = (x_1^{-1}, x_2)$. Then

$$\tilde{a}_1(b_1 g b_2)\tilde{a}_2 = b_1\tilde{a}_1 g \tilde{a}_2 b_2 = b_1 a_1 g a_2 b_2.$$

3.4. **Stickel's key exchange protocol.** We conclude with an example where the complexity of the cryptanalysis is surprisingly small. The key exchange protocol described in Figure 5 was introuduced by Stickel in 2005 [21].

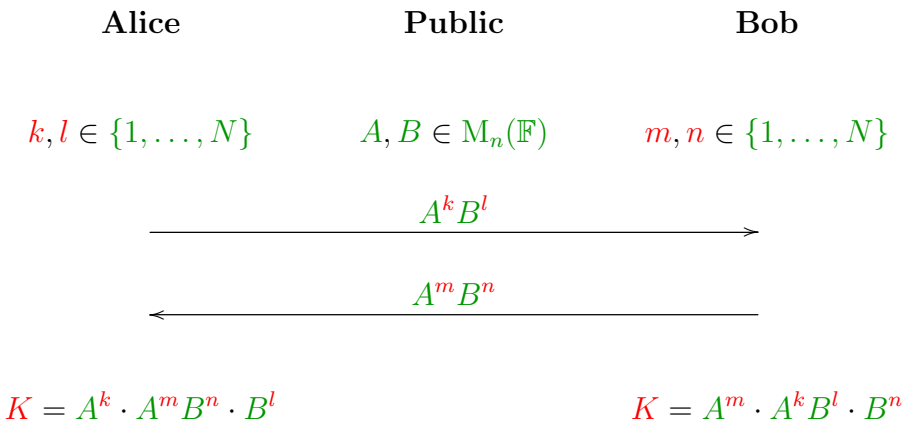

FIGURE 5. Stickel's key exchange protocol

A successful heuristic cryptanalysis of complexity roughly $n^{2\omega}$ was presented by Shpilrain [18]. Shpilrain's cryptanalysis turned out provable [23]. The algebraic span method provides a simple alternative, of smaller complexity.

The dimension of the *algberas* spanned by the matrices A and B is, by the Cayley–Hamilton Theorem, at most n. Find a matrix $\tilde{A} \in \mathrm{Alg}(\{A\})$ and an invertible matrix $D \in \mathrm{Alg}(\{B\})$ such by solving the linear equation $\tilde{A} = A^k B^l D$. Since the dimension is $O(n)$, the complexity is $O(n^4)$. Let $\tilde{B} = D^{-1}$. A cyclic algebra is commutative. Moreover, the matrix $\tilde{B}$ is a finite power of D, and is thus in $\mathrm{Alg}(\{B\})$. Thus,

$$\tilde{A} \cdot A^m B^n \cdot \tilde{B} = A^m \tilde{A} \tilde{B} B^n = A^m A^k B^l B^n = K.$$

4. Cryptanalysis of the Triple Decomposition key exchange protocol

Kurt's *Triple Decomposition* key exchange protocol ([10], [12, 4.2.5]) is described in Figure 6. In this figure, uppercase letters denote subgroups. An edge between two subgroups means that these subgroups commute elementwise. This ensures that the keys computed by Alice and Bob are both equal to $ab_1a_1b_2a_2b$.

Let $c := x_1^{-1} a_1 x_2$. By moving the matrix x_1 or x_2 to the other side of the equation, the public information $x_1^{-1} a_1 x_2$ provides a quadratic equation, and similarly for the public information $y_1^{-1} b_2 y_2$. Solving quadratic equations may be very difficult. This prevented the application of earlier methods to this key exchange protocol. The natural approach would be to ignore this part of the pubic information, and solve the linear equations provided by the other public items. This works for generic matrix groups, but fails, according to our experiments, for the actual groups proposed in [10]. We provide here a way that takes the triple products into account, in a linear way, which still provably obtains the correct key. In the framework of algebraic spans, this solution is natural.

The following sets can be computed from the public information:

$$\begin{aligned}
\mathrm{Alg}(B_1) y_1 &= \mathrm{Alg}(B_1) \cdot b_1 y_1 \\
\mathrm{Alg}(B_2 \cup Y_2) y_1 &= \mathrm{Alg}(B_2 \cup Y_2) \cdot y_2^{-1} b_2^{-1} y_1 = \mathrm{Alg}(B_2 \cup Y_2) \cdot (y_1^{-1} b_2 y_2)^{-1} \\
\mathrm{Alg}(A_2) x_2 &= \mathrm{Alg}(A_2) \cdot a_2^{-1} x_2 \\
\mathrm{Alg}(A_1 \cup X_1) x_2 &= \mathrm{Alg}(A_1 \cup X_1) \cdot x_1^{-1} a_1 x_2
\end{aligned}$$

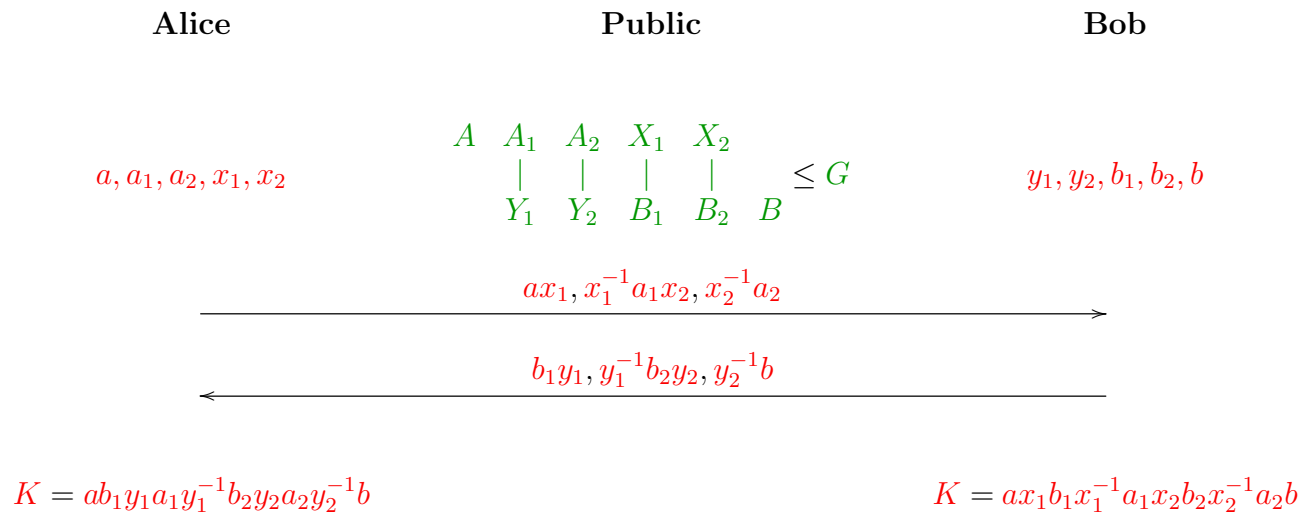

FIGURE 6. The Triple Decomposition key exchange protocol

The invertible matrices y_1 and x_2 are, respectively, in the following intersections of subspaces of $\mathrm{M}_n(\mathbb{F})$:

$$\mathrm{span}(Y_1) \cap \mathrm{Alg}(B_1)y_1 \cap \mathrm{Alg}(B_2 \cup Y_2)y_1;$$
$$\mathrm{span}(X_2) \cap \mathrm{Alg}(A_2)x_2 \cap \mathrm{Alg}(A_1 \cup X_1)x_2.$$

By the Invertibility Lemma [23, Lemma 9], we can pick invertible elements $\tilde{y}_1$ and $\tilde{x}_2$ in these intersections, respectively. Then:

(1) Since the elements y_1 and $\tilde{y}_1$ are in $\mathrm{Alg}(Y_1)$, they commute with the elements of A_1.
(2) Since $\tilde{y}_1 \in \mathrm{Alg}(B_1)y_1$, we have that $\tilde{y}_1y_1^{-1} \in \mathrm{Alg}(B_1)$, and thus the element $\tilde{y}_1y_1^{-1}$ commutes with the elements of X_1. By (1), it also commutes with the elements of A_1.
(3) Since $\tilde{y}_1 \in \mathrm{Alg}(B_2 \cup Y_2)y_1$, we have that $\tilde{y}_1y_1^{-1} \in \mathrm{Alg}(B_2 \cup Y_2)$.

Similarly, we have that:

(1) The elements x_2 and $\tilde{x}_2$ commute with the elements of B_2.
(2) The element $\tilde{x}_2x_2^{-1}$ commutes with the elements of $Y_2 \cup B_2$.
(3) $\tilde{x}_2x_2^{-1} \in \mathrm{Alg}(A_1 \cup X_1)$.

It suffices to use one of the items numbered (3). We will use here the former.

Using the public information, compute

$$\tilde{K} := ax_1 \cdot b_1y_1 \cdot \tilde{y}_1^{-1} \cdot x_1^{-1}a_1x_2 \cdot \tilde{x}_2^{-1} \cdot \tilde{y}_1 \cdot y_1^{-1}b_2y_2 \cdot \tilde{x}_2 \cdot x_2^{-1}a_2 \cdot y_2^{-1}b.$$

We claim that $\tilde{K} = K = ab_1a_1b_2a_2b$, the key that Alice and Bob established.

Since X_1 commutes with B_1 elementwise and $\tilde{y}_1y_1^{-1} \in \mathrm{Alg}(B_1)$, we have that

$$x_1 \cdot b_1 \cdot y_1\tilde{y}_1^{-1} \cdot x_1^{-1} = b_1y_1\tilde{y}_1^{-1}.$$

Since $\tilde{x}_2x_2^{-1}$ commutes with the elements of $Y_2 \cup B_2$ and $\tilde{y}_1y_1^{-1} \in \mathrm{Alg}(B_2 \cup Y_2)$, we have that

$$x_2\tilde{x}_2^{-1} \cdot \tilde{y}_1y_1^{-1} \cdot b_2 \cdot y_2 \cdot \tilde{x}_2x_2^{-1} = \tilde{y}_1y_1^{-1}b_2y_2.$$

Thus,

$$\tilde{K} = ab_1y_1\tilde{y}_1^{-1}a_1\tilde{y}_1y_1^{-1}b_2y_2a_2y_2^{-1}b.$$

Since Y_2 commutes with A_2 elementwise, we have that

$$y_2 a_2 y_2^{-1} = a_2.$$

Since $\tilde{y}_1 y_1^{-1}$ commutes with the elements of A_1, we have that

$$y_1 \tilde{y}_1^{-1} \cdot a_1 \cdot \tilde{y}_1 y_1^{-1} = a_1.$$

It follows that

$$\tilde{K} = a b_1 a_1 b_2 a_2 b,$$

as required.

The complexity of this cryptanalysis is dominated by the calculation of the algebraic spans, which is $O(kd^2n^2)$, where k the maximum number of generators of the given subgroups, and d is the maximum dimension of the Algebra generated by them. In particular, it is not greater than $O(kn^6)$.

References

[1] I. Anshel, M. Anshel, D. Goldfeld, *An algebraic method for public-key cryptography*, Mathematical Research Letters **6** (1999), 287–291.

[2] J. Cha, K. Ko, S. Lee, J. Han, J. Cheon, *An efficient implementation of braid groups*, ASIACRYPT 2001, LNCS **2248** (2001), 144–156.

[3] J. Cheon, B. Jun, *A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem*, CRYPTO 2003, Lecture Notes in Computer Science **2729** (2003), 212–224.

[4] R. Gilman, A. Myasnikov, A. Myasnikov, A. Ushakov, *New developments in Commutator Key Exchange*, Proceedings of the First International Conference on Symbolic Computation and Cryptography, Beijing, 2008, 146–150.
`http://www-calfor.lip6.fr/~jcf/Papers/scc08.pdf`

[5] D. Giry, *BlueKrypt: Cryptographic Key Length Recommendation*, `http://www.keylength.com`

[6] M. González–Vasco, R. Steinwandt, **Group Theoretic Cryptography**, Cryptography and Network Security Series, Chapman and Hall/CRC Press, 2015.

[7] D. Holt, answer to MathOverflow question `http://mathoverflow.net/questions/154761`

[8] K. Ko, S. Lee, J. Cheon, J. Han, J. Kang, C. Park, *New public-key cryptosystem using braid groups*, CRYPTO 2000, Lecture Notes in Computer Science **1880** (2000), 166–183.

[9] Y. Kurt, *A new key exchange primitive based on the triple decomposition problem*, IACR eprint 2006/378.

[10] Y. Kurt Peker, *A new key agreement scheme based on the triple decomposition problem*, International Journal of Network Security **16** (2014), 340–350.

[11] A. Myasnikov, V. Shpilrain, A. Ushakov, **Group-based cryptography**, Birkhäuser, 2008.

[12] A. Myasnikov, V. Shpilrain, A. Ushakov, **Non-commutative Cryptography and Complexity of Group-theoretic Problems**, American Mathematical Society Surveys and Monographs **177**, 2011.

[13] *A linear decomposition attack*, Groups Complexity Cryptology **7** (2015), 81–94.

[14] V. Roman'kov, **Algebraic cryptography**, Omsk State Dostoevsky University, 2013. (In Russian)

[15] V. Roman'kov, *Cryptanalysis of some schemes applying automorphisms*, Prikladnaya Discretnaya Matematika **3** (2013), 35–51. (In Russian)

[16] V. Roman'kov, *A polynomial time algorithm for the braid double shielded public key cryptosystems*, arXiv eprint 1412.5277, 2014.

[17] V. Roman'kov, *Linear decomposition attack on public key exchange protocols using semidirect products of (semi)group*, arXiv eprint 1501.01152, 2015.

[18] V. Shpilrain, *Cryptanalysis of Stickel's key exchange scheme*, in: **Computer Science in Russia**, Lecture Notes in Computer Science 5010 (2008), 283–288.

[19] V. Shpilrain, A. Ushakov, *Thompson's group and public key cryptography*, ACNS 2005, Lecture Notes in Computer Science **3531** (2005), 151–164.

[20] V. Shpilrain, A. Ushakov, *A new key exchange protocol besed on the decomposition problem*, in: L. Gerritzen, D. Goldfeld, M. Kreuzer, G. Rosenberger and V. Shpilrain, eds., **Algebraic Methods in Cryptography**, Contemporary Mathematics **418** (2006), 161–167.

[21] E. Stickel, *A new method for exchanging secret keys*, Proceedings of the Third International Conference on Information Technology and Applications (ICITA05), 2005, 426–430.

[22] B. Tsaban, *The Conjugacy Problem: cryptoanalytic approaches to a problem of Dehn*, minicourse, Düsseldorf University, Germany, July–August 2012.
http://reh.math.uni-duesseldorf.de/~gcgta/slides/Tsaban_minicourses.pdf

[23] B. Tsaban, *Polynomial time solutions of computational problems in noncommutative-algebraic cryptography*, Journal of Cryptology, to appear. DOI: 10.1007/s00145-013-9170-9

[24] B. Tsaban, *Practical polynomial time solutions of several major problems in noncommutative-algebraic cryptography (preliminary announcement)*, IACR eprint 2014/041. Version 20140115:201530, Januray 2014.

Department of Mathematics, Bar-Ilan University, Ramat Gan 5290002, Israel
E-mail address, Ben-Zvi: adi2lugassy@gmail.com
E-mail address, Kalka: tschussle@gmail.com
E-mail address, Tsaban: tsaban@math.biu.ac.il
URL, Kalka: http://homepage.ruhr-uni-bochum.de/arkadius.kalka
URL, Tsaban: http://www.cs.biu.ac.il/~tsaban