A Polynomial Time Algorithm for the Braid Diffie-Hellman Conjugacy Problem

Jung Hee Cheon[1] and Byungheup Jun[2]

[1] School of Mathematical Sciences, Seoul National University, Republic of Korea
jhcheon@math.snu.ac.kr
[2] Korea Institute for Advanced Study, Republic of Korea
bhjun@kias.re.kr

Abstract. We propose the first polynomial time algorithm for the braid Diffie-Hellman conjugacy problem (DHCP) on which the braid key exchange scheme and the braid encryption scheme are based [9]. We show the proposed method solves the DHCP for the image of braids under the Lawrence-Krammer representation and the solutions play the equivalent role of the original key for the DHCP of braids. Given a braid index n and a canonical length ℓ, the complexity is about $O(n^{14.4}\ell^{3.2})$ or $O(n^{4\tau+2\epsilon}\ell^{2\epsilon})$ bit operations for $\tau = \log_2 7 \approx 2.8$ and $\epsilon > \log_2 3 \approx 1.57$.

Keywords: Braid group, Non-abelian group, Conjugacy Problem

1 Introduction

In 2000, a key agreement and an encryption scheme based on braid groups were proposed by Ko *et. al* [9]. The schemes are analogous to the Diffie-Hellman key agreement scheme and the ElGamal encryption scheme on abelian groups. Their basic mathematical problem is the *Conjugacy Problem* (CP) on braids: For a braid group B_n, we are asked to find a braid a from $u, b \in B_n$ satisfying $b = aua^{-1} \in B_n$. The security is based on the *Diffie-Hellman Conjugacy Problem* (DHCP) to find $baua^{-1}b^{-1} \in B_n$ for given $u, aua^{-1}, bub^{-1} \in B_n$ for a and b in two commuting subgroups of B_n respectively. There are only brute-force attack and super-summit set attack as the analysis. Both yields a complexity of exponential time [9, 4]. Recently, several heuristic algorithms were proposed using Burau representation. Though they may be implemented in quite efficient way, they do not solve the whole problem (their methods do not work for some parameters), so no theoretical bounds have been written yet [6, 14].

One may approach the CP using a representation in another group whose structure we know better. As mathematicians have developed linear algebra for more than hundred years, linear algebraic groups are possible candidates. There are two candidates as linear representations of braid groups: Burau and Lawrence-Krammer representations. Burau representation was used in *loc. cit.* to make a quite reasonable records. Unfortunately, it is known to be unfaithful, they cannot bound the complexity of the scheme as we expected.

Lawrence-Krammer representation is now chosen to analyze the PKC. It has been proved faithful for arbitrary index of Braids, several times in independent ways by several authors. In general it increases the rank of the representations, so it is complicated to describe. Nevertheless, it is known, but not written clearly, one can easily recover the original braid from its matrix of the representation [13]. Under this assumption, we describe an algorithm to solve the CP.

1. Find the images of u and $v = aua^{-1}$ in $GL_{n(n-1)/2}(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$ via the Lawrence-Krammer representation $\mathcal{K} : B_n \to GL_{n(n-1)/2}(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$.
2. Solve the CP for $\mathcal{K}(u)$ and $\mathcal{K}(v) = \mathcal{K}(a)\mathcal{K}(u)\mathcal{K}(a)^{-1}$ in $GL_{n(n-1)/2}(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$.
3. Recover the braid a in B_n from the matrix obtained above.

The above algorithm contains a couple of difficulties. Firstly, direct applications of Gaussian elimination should deal with coefficients as large as 2^{2^n}. Secondly, a solution of the CP in the matrix group might not be in the image of the representation. It is not easy to choose a matrix in the solution space which lies inside the image of the representation.

To avoid these difficulties, we take the DHCP into our consideration, instead of the CP. The algorithm is modified, roughly as follows:

1. Assume $a \in LB_n$, $b \in RB_n$, and $u \in B_n$ where LB_n and RB_n are two commuting subgroups of B_n.
2. Find the images of u, $v = aua^{-1}$, and $w = bub^{-1}$ in $GL_{n(n-1)/2}(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$ via the Lawrence-Krammer representation $\mathcal{K} : B_n \to GL_{n(n-1)/2}(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$.
3. By estimating the entries of $\mathcal{K}(awa^{-1})$, take a prime p and irreducible polynomials $f(t)$ over $\mathbb{Z}/p$ and $g(q)$ over $\mathbb{Z}[t]/(p, f(t))$ satisfying

$$\mathcal{K}(awa^{-1}) = t^{-d}N^{-1}\{t^d N\mathcal{K}(awa^{-1}) \mod (p, f(t), g(q))\}$$

   for some positive integer d and N.
4. Solve the simultaneous equations $\mathcal{K}(v)A = A\mathcal{K}(u)$ and $\mathcal{K}(\sigma_i)A = A\mathcal{K}(\sigma_i)$ with $n/2 < i \leq n$ over a residue class field $k = \mathbb{Z}[t, q]/(p, f(t), g(q))$, where σ_i with $n/2 < i \leq n$ generates RB_n.
5. This solution may not be $\mathcal{K}(a)$, but it plays an equivalent role of the key for the DHCP in braid groups. That is, any solution A of the above system of equations satisfies $A\mathcal{K}(w)A^{-1} = \mathcal{K}(b)\mathcal{K}(v)\mathcal{K}(b)^{-1} = \mathcal{K}(awa^{-1})$ since $\mathcal{K}(b)A = A\mathcal{K}(b)$ for $b \in RB_n$. The inverse of A can be computed in a similar way to the above method.
6. Recover the braid awa^{-1} in B_n by inverting the representation.

To reduce the complexity of this algorithm, we use $1/2$ instead of q (it is also faithful), reduce the bound of Krammer matrices, and remove several trivial variables and equations in the simultaneous equations. When ℓ is the Charney length of a, b, and u in B_n, the complexity of this algorithm analyzed in this article reaches about $2^{-3}n^{4\tau+2\epsilon}\ell^{2\epsilon}$ where $\tau = \log_2 7 \approx 2.8$ and $\epsilon > \log_2 3 \approx 1.57$. This is not a feasible complexity for the parameters recommended in [9, 4]. For example, for $n = 90$ and $\ell = 12$ as in [4] it is about 2^{103} bit operations. But even for $n = 10^5$ and $\ell = 10^4$, the complexity is just 2^{278}. Hence the braid encryption scheme can not be used in the future in this style. We would suggest that the protocol should be revised to use the full difficulty of CP to overcome the attack. In the near future, there may be modifications of this kind of attacks, since the chosen bounds of coefficients of the Krammer matrices are rather rough whereas an image of an Artin generator is almost sparse matrix with small coefficients. We also remark that the proposed algorithm does not give an answer to the CP. Thus the CP is still hard and unsolved.

The rest of the paper is composed as follows: In Section 2, we briefly review braid groups and braid cryptography. In Section 3, we introduce the Lawrence-Krammer representation and develop its properties. Also inverting algorithm is given in more concrete way with the complexity. In Section 4, we introduce an equivalent key which plays an equivalent role as the original braid and analyze the cryptosystem using this. Section 5 gives the conclusion of this paper.

2 An Overview of Braid Group Cryptography

2.1 Braid Groups

A *braid* is obtained by laying down a number of parallel strands and intertwining them so that they run in the same direction. The number of strands is called the *braid index*. The set B_n of isotopy classes of braids of index n is naturally equipped with a group structure, called the *n-braid group*, where the product of two braids x and y is nothing more than laying down the two braids in a row and then matching the end of x to the beginning of y.

Any braid can be decomposed as a product of simple braids. One type of simple braids is the *Artin generator* σ_i that have a single crossing between i-th and $(i+1)$-th strand. B_n is presented with the Artin generators $\sigma_1, \ldots, \sigma_{n-1}$ and relations $\sigma_i\sigma_j = \sigma_j\sigma_i$ for $|i-j| > 1$ and $\sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j$ for $|i-j| = 1$. When a braid a is expressed as a product of Artin generators, the minimum number of terms in the product is called the *w*ord length of a.

We have still other presentations. Let S_n be the symmetric group of an n-element set $I_n = \{1, 2, \ldots, n\}$. Let $\mathrm{Ref} = \{(i,j) | 1 \le i < j \le n\}$ be the set of reflections (that interchange two elements and fix the other elements of I_n) in S_n and S the subset $\{(i, i+1) | 1 \le i < n\}$ of Ref. We define $\ell(x)$ the *length of a permutation* x in S_n as

$$\ell(x) := \min\{k | x = x_1 \cdots x_k \text{ for } x_i \in S\}.$$

B_n admits another presentation with generators $\{rx | x \in S_n\}$ and relations $r(xy) = (rx)(ry)$ if $\ell(xy) = \ell(x) + \ell(y)$. In this presentation, the longest permutation w_0 with $w_0(i) = n+1-i$ yields a braid Δ, which is called the *fundamental braid* or the *half-twist* depending on authors. Let B_n^+ denote the submonoid of B_n generated by S_n. A braid in B_n^+ is said to be positive. A braid x is written uniquely, $x = \Delta^k x'$ where x' is in $B_n^+ - \Delta B_n^+$. This is called the normal form of x.

There is a partial order on B_n^+: $x \le y \Leftrightarrow y \in xB_n^+$. The ordering is inherited to S_n (We identify a permutation σ with the corresponding braid $r\sigma$ in B_n^+.). We denote rS_n by Ω for simplicity reason. For a braid $x \in B_n^+$, the greatest element of the set $\{y \in \Omega | y \le x\}$ is called the *left most factor* of x and denoted by $\mathrm{LF}(x)$. A sequence of braids $(x_1, \ldots, x_k)$ in $\Omega - \{1\}$ is called the *greedy form* of x if $x_1 \cdots x_k = x$, $\mathrm{LF}(x_i x_{i+1}) = x_i$ for all i. The above k in the greedy form is called the *Charney length* of x. This length function is easily extended to general braids using Thurston normal form, but we don't need it so general for our purpose and we will omit the general definition.

2.2 Braid Cryptography

Let G be a non-abelian group and $u, a, b, c \in G$. In order to perform the Diffie-Hellman key agreement on G we need to choose a, b in G satisfying $ab = ba$ in the

DHCP. Hence we introduce two commuting subgroups $G_1, G_2 \subset G$ satisfying $ab = ba$ for any $a \in G_1$ and $b \in G_2$. More precisely, the problems the braid cryptography are based on are as follows:

- Input: A non-abelian group G, two commuting subgroups $G_1, G_2 \subset G$
- Conjugacy Problem (CP): Given (u, aua^{-1}) with $u, a \in G$, compute a. (Note that if we denote aua^{-1} by u^a, it looks like the DLP.)
- Diffie-Hellman Conjugacy Problem (DHCP): Given (u, aua^{-1}, bub^{-1}) with $u \in G$, $a \in G_1$ and $b \in G_2$, compute $baua^{-1}b^{-1}$.
- Decisional Diffie-Hellman Conjugacy Problem (DDHCP): Given $(u, aua^{-1}, bub^{-1}, cuc^{-1})$ with $u, c \in G$, $a \in G_1$ and $b \in G_2$, decide whether $abub^{-1}a^{-1} = cuc^{-1}$.

In braids, we can easily take two commuting subgroups G_1 and G_2 of B_n (For simplicity, we only consider a braid group with an even braid index. But it is easy to extend this to an odd braid index.). For example, $G_1 = LB_n$ (resp. $G_2 = RB_n$) is the subgroup of B_n consisting of braids made by braiding left $n/2$ strands(resp. right $n/2$ strands) among n strands. Thus LB_n is generated by $\sigma_1, \ldots, \sigma_{n/2-1}$ and RB_n is generated by $\sigma_{n/2+1}, \ldots, \sigma_{n-1}$. Then we have the commutative property that for any $a \in G_1$ and $b \in G_2$, $ab = ba$.

[Key agreement] This is the braid group version of the Diffie-Hellman key agreement.

1. **Initial setup:** (a) Choose system parameters n and ℓ from positive integers.
   (b) Select a sufficiently complicated positive braid $u \in B_n$ with ℓ canonical factors.
2. **Key agreement:** Perform the following steps each time a shared key is required.
   (a) A chooses a random secret positive braid $a \in LB_n$ with ℓ canonical factors and sends $v_1 = aua^{-1}$ to B.
   (b) B chooses a random secret braid $b \in RB_n$ with ℓ canonical factors and sends $v_2 = bub^{-1}$ to A.
   (c) A receives v_2 and computes the shared key $K = av_2a^{-1}$.
   (d) B receives v_1 and computes the shared key $K = bv_1b^{-1}$.

Since $a \in LB_n$ and $b \in RB_n$, $ab = ba$. It follows

$$av_2a^{-1} = a(bub^{-1})a^{-1} = b(aua^{-1})b^{-1} = bv_1b^{-1}.$$

Thus Alice and Bob obtain the same braid.

[Public-key cryptosystem] Let $H\colon B_n \to \{0,1\}^k$ be a cryptographically secure hash function from the braid group to the message space.

1. **Initial setup:** (a) Choose system parameters n and ℓ from positive integers.
   (b) Select a sufficiently complicated positive braid $u \in B_n$ with ℓ canonical factors.

2. **Key generation:**
   (a) Choose a sufficiently complicated positive braid $u \in B_n$ with ℓ canonical factors.
   (b) Choose a positive braid $a \in LB_n$ with ℓ canonical factors.
   (c) Public key is (u, v), where $v = aua^{-1}$; Private key is a.
3. **Encryption:** Given a message $m \in \{0,1\}^k$ and the public key (u, v),
   (a) Choose a positive braid $b \in RB_n$ with ℓ canonical factors.
   (b) Ciphertext is (c, d), where $c = bub^{-1}$ and $d = H(bvb^{-1}) \oplus m$.
4. **Decryption:** Given a ciphertext (c, d) and private key a, compute $m = H(aca^{-1}) \oplus d$.

Since a and b commute, $aca^{-1} = abub^{-1}a^{-1} = baua^{-1}b^{-1} = bvb^{-1}$. So $H(aca^{-1}) \oplus d = H(bvb^{-1}) \oplus H(bvb^{-1}) \oplus m = m$ and the decryption recovers the original braid m.

We may take a non-positive braid for a system braid or secret braids. But since the problem in that case is reduced to the positive braid cases, positive braids are enough for the random braids in this cryptosystem.

3 The Lawrence-Krammer Representation

3.1 Definitions and Properties

Most definitions and facts in this section are taken from two papers [10] [11] of Krammer. Let us recall the Lawrence-Krammer representation of braid groups. This is a representation of B_n in $GL_m(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}]) = Aut(V_0)$, where $m = n(n-1)/2$ and V_0 is the free module of rank m over $\mathbb{Z}[t^{\pm 1}, q^{\pm 1}]$. We shall denote the representation by $\mathcal{K}$. With respect to $\{x_{ij}\}_{1 \le i < j \le n}$ the free basis of V_0 the image of each Artin generator under $\mathcal{K}$ is written as

$$\mathcal{K}(\sigma_k)(x_{ij}) = \begin{cases} tq^2 x_{k,k+1}, & i = k, \quad j = k+1; \\ (1-q)x_{i,k} + qx_{i,k+1}, & j = k, \quad i < k; \\ x_{ik} + tq^{k-i+1}(q-1)x_{k,k+1}, & j = k+1, \quad i < k; \\ tq(q-1)x_{k,k+1} + qx_{k+1,j}, & i = k, \quad k+1 < j; \\ x_{kj} + (1-q)x_{k+1,j}, & i = k+1, \quad k+1 < j; \\ x_{ij}, & i < j < k \quad \text{or} \quad k+1 < i < j; \\ x_{ij} + tq^{k-i}(q-1)^2 x_{k,k+1}, & i < k < k+1 < j. \end{cases} \tag{1}$$

The matrix $\mathcal{K}(\sigma_k)$ with respect to the basis x_{ij} will be called by the Krammer matrix of a braid σ_k.

To estimate the complexity of the algorithm proposed here, we need to estimate bounds for the entries of a Krammer matrix.

Two useful results in [11] follow below:

Fact 1 *[11] $\mathcal{K}(\Delta)(x_{n+1-j,n+1-i}) = tq^{i+j-1}x_{ij}$ for $1 \le i < j \le n$.*

Fact 2 *[11] Let $x \in B_n$. Consider the Laurent series of $\mathcal{K}(x)$ with respect to t,*

$$\mathcal{K}(x) = \sum_{i=k}^{\ell} A_i(q)t^i, \quad A_i \in M_m(\mathbb{Z}[q^{\pm 1}]), \quad A_k \neq 0, A_\ell \neq 0. \tag{2}$$

Then $\ell_\Omega(x) = \max(\ell - k, -k, \ell)$.

If we consider a different generator $Q = \{s(i,j)|$ the permutation braid of the reflection $(i,j) \in S_n\}$, we can define another length function ℓ_Q with respect to Q. This length is the canonical length in the band generator presentation. Remark that $\ell_Q(x)$ is bounded by $(n-1)$-times of the canonical length in the Artin presentation, because a band generator is written with upto $(n-1)$ Artin generators.

Define the anti-automorphism of B_n, written $x \mapsto \bar{x}$, by giving $[ij] \mapsto [n+1-i, n+1-j]$. This preserves B_n^+ as well as the canonical length. Then the dual representation is defined as $\mathcal{K}^* : B_n \to GL(V_0)$ by $\mathcal{K}^*(x) = \mathcal{K}(\bar{x})^T$, where T denotes the transpose. Consider another basis $\{v_{ij} | 1 \leq i < j \leq n\}$ of V_0. It is related to $\{x_{ij}\}$ by

$$v_{ij} = x_{ij} + (1-q) \sum_{i<k<j} x_{kj}, \quad x_{ij} = v_{ij} + (q-1) \sum_{i<k<j} q^{k-1-i} v_{kj}. \tag{3}$$

Fact 3 *[10] Let $x \in B_n$. Consider the Laurent series of $\mathcal{K}^*(x)$ with respect to q,*

$$\mathcal{K}^*(x) = \sum_{i=k}^{\ell} A_i(t)q^i, \quad A_i \in M_m(\mathbb{Z}[t^{\pm 1}]), \quad A_k \neq 0, A_\ell \neq 0. \tag{4}$$

Then $\ell_Q(x) = \frac{1}{2}\max(\ell - k, -k, \ell)$.

From the above three facts, we get the following theorem.

Theorem 1. *Let x be a braid with the canonical form $\Delta^k x_1 x_2 \cdots x_\ell$ where x_i is a permutation braid which is not the fundamental braid. Let δ be the minimal number of Artin generators in x. Then we have the following bounds for the coefficients of $\mathcal{K}(x)$:*

(a) The degree in t is bounded below by k and above by $k + \ell$.
(b) The degree in q is bounded below by $2(n-1)\min(0,k) + (n-2)$ and above by $2(n-1)\max(k+\ell, k) + (n-2)$.
(c) The coefficients of each entry inside the Krammer matrix are bounded by 2^δ when we consider the entries as a polynomial in t, q, and $1-q$.

Proof. (a) It is clear from Fact 2.
(b) Since $\bar{x}$ has the same canonical length with x and the band canonical length is bounded by $(n-1)$ times the Artin canonical length, the degree in q is bounded below by $2(n-1)\min(0,k)$ and above by $2(n-1)\max(k+\ell,k)$ in the $\{v_{ij}\}$ basis. While taking the basis change from $\{v_{ij}\}$ to $\{x_{ij}\}$, we have at most $(n-2)$ increase in the degree of q. Hence we get (b).

(c) If we consider entries of a Krammer matrix as a polynomial in t, q, and $1-q$, every entry of any Artin generator is a monomial with coefficients in $\{0, \pm 1\}$. For any Artin generator σ, each column of $\mathcal{K}(\sigma)$ has at most two nonzero terms (See the equation (1)). Hence a multiplication by a Krammer matrix of an Artin generator results in the increase of the coefficients by at most 2 times for each entries. Note that it happens when the same monomial occurs twice at an entry in the result matrix. Hence the coefficients of entries is bounded by 2^{δ} for the number of Artin generator in x.

For any positive integer n, the Krammer representation is faithful even if q is a real number with $0 < q < 1$ [11]. Also the inverting algorithm does not change even if q is replaced by a real number with $0 < q < 1$. From now on, we will consider the modified Krammer representation $\mathcal{K}'(x) = \mathcal{K}(x)_{q=1/2}$. In that case, q is equal to $1-q$.

Corollary 1. *Let x be a braid with the canonical form $\Delta^k x_1 x_2 \cdots x_\ell$ where x_i is a permutation braid which is not the fundamental braid. Let δ be the number of Artin generators in x. Then we have the following bounds for $\mathcal{K}'(x)$:*

(a) The degree in t is bounded below by k and above by $\max(k+\ell, k)$.
(b) The coefficients of each entry inside $\mathcal{K}'(x)$ is given by a ratio of two integers. The absolute values of numerators and denominators are bounded by $2^{\delta - 2(n-1)k}$ and $2^{2(n-1)\max(k+\ell,k)}$, respectively.

3.2 Inverting the Lawrence-Krammer representation

Here we develop a way to recover a braid from its image matrix under the Lawrence-Krammer representation. As mentioned earlier, the faithfulness of the Lawrence-Krammer representation of B_n in a linear group has been proven in several ways by different authors. Moreover, it has been known to be so easy that it takes a polynomial time of low degree in braid length and the index but we haven't found any reference with an explicit complexity available at hand.

The proof of faithfulness was due to Krammer [11], which enables us to construct an algorithmic way to recover the original braid from a matrix of the representation.

From Fact 1 we can easily obtain the matrix of Δ as tA, for a matrix A whose entries are from $\mathbb{Z}[q^{\pm 1}]$. Together with Fact 2, it suffices to recover the original braid x' of the matrix $(tA)^{-d_0}\mathcal{K}(x)$. Note that x' lies in $B_n^+ - \Delta B_n^+$, which corresponds to the nontrivial part in the normal form of x. x' has obviously smaller Charney length than x.

Suppose now x is a positive braid. Let us take $\{v_{ij}\}$ as the basis of V_0. The Lawrence-Krammer representation $\mathcal{K}$ yields a natural action of the monoid B_n^+ over V_0. Let A be the subset of Ref, $\{(i,j) \in \mathrm{Ref} | (x(1,\ldots,1))_{(i,j),t=0} \neq 0\}$. This A corresponds to a permutation y in S_n which corresponds to the braid ry in Ω. It makes the left most factor of x, so one has $x = yx'$. Applying the same steps to $\mathcal{K}(x')$ recursively, we obtain the greedy form of x after all, as it decreases the Charney length.

In this way, given $\mathcal{K}(x) = \sum_{i=d_t}^{\ell} A_i(q)t^i$, we can recover $x \in B_n$ in polynomial time. We shall describe the algorithm roughly as follows:

Algorithm 1 *Invert the Lawrence-Krammer representation.*
Input: *A matrix* $\mathcal{K}(x) \in GL_m(t^{\pm 1}, q^{\pm 1})$ *where* $m = n(n-1)/2$
Output: *A braid* $x \in B_n$.

1. *Compute* $\mathcal{K}(x') = \mathcal{K}(\Delta)^{-d_t}\mathcal{K}(x)$
2. *Perform the basis change from* $(v_{ij})_{ij}$ *to* $(x_{ij})_{ij}$.
3. *For* $k = 1$ *to* ℓ *do*
   2.1 *Take a nonzero element* $y \in D_\phi$ *and compute*

$$A = \{c_{ij} | \mathcal{K}(x')y \text{ has a nonzero coefficient at the } ij \text{ coordinate}\}$$

   (For the definition of the set D_ϕ *one can refer to [11].)*
   2.2 *Compute the maximal element* $\tau_k \in S_n$ *such that* $L(z) \subset A$ *as follows.*
   - *Find all reflections* $\sigma_i = (i, i+1)$ *with* $1 \le i < n$ *such that* $L(\sigma_i) \subset A$
   - *Given* $\tau\sigma_i$ *and* $\tau\sigma_j$, *a greater element is selected from* $\tau\sigma_i\sigma_j$ *or* $\tau\sigma_i\sigma_j\sigma_i$
   - *Find the maximal element* $\tau_k \in S_n$ *by the recursive use of the above method*

   2.3 *Compute the positive braid* x_k *corresponding to* τ_k
   2.4 *Replace* $\mathcal{K}(x')$ *by* $\mathcal{K}(x_k)^{-1}\mathcal{K}(x')$
4. *Output* $x = \Delta^{d_t} x_1 x_2 \cdots x_k$

The complexity of this algorithm is about d_t power of a matrix and $n^2\ell$ multiplication of permutations, which is dominated by d_t power of a matrix.

Theorem 2. *Given* $\mathcal{K}(x) = \sum_{i=d_t}^{\ell} A_i(q)t^i$, *we can recover* $x \in B_n$ *in* $O(2m^3 \log d_t)$ *multiplications of entries.*

Note that it works even when a (nonzero) constant multiple of $\mathcal{K}'(x)$ is given since we only check whether the coefficient is zero in each stage. Hence we may deal with integer coefficients instead of rational coefficients.

4 Cryptanalysis of Braid Cryptosystems

4.1 An equivalent Key

The security of the key exchange scheme and the encryption scheme in braids are based on the DHCP. The DHCP asks to find $baua^{-1}b^{-1}$ from $u, v = aua^{-1}, w = bub^{-1}$ given two commuting subgroups LB_n and RB_n of B_n, $a \in LB_n$, $b \in RB_n$ and $u \in B_n$. In this section, firstly, we will show that we don't need the original key a but a "fake" key A to solve the DHCP. The DHCP on a linear group is equivalent to a system of linear equations, whose solutions acts as the fake key. Note that it breaks the encryption scheme and key agreement scheme, but does not solve the original conjugacy problem to the bottom. The conjugacy problem in a general non-commutative group is, nevertheless, still difficult.

Without solving the problem in B_n, we try to solve it in $\mathrm{GL}_m(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$ for $q = 1/2$ and $m = n(n-1)/2$ via the modified Lawrence-Krammer representation. Denote by A, B, U, V, and W the image of a, b, u, v, and w under this representation

$\mathcal{K}'$, respectively. We will compute a matrix A from $\mathrm{GL}_m(\mathbb{Z}[t])$ satisfying the following equations:

$$UA = AV \tag{5}$$

$$A\mathcal{K}'(\sigma_i) = \mathcal{K}'(\sigma_i)A, \quad n/2 < i < n. \tag{6}$$

The solutions in $\mathbb{Z}[t]^{m^2}$ make a nontrivial module $\mathcal{N}$ over $\mathbb{Z}[t]$, since we have already a nontrivial solution $\mathcal{K}(a)$. As the set of invertible matrices in $\mathcal{N}$ is dense under Zariski topology, we can take an invertible matrix over $\mathbb{Q}(t)$ from $\mathcal{N}$ with overwhelming probability. Let A' be an invertible matrix solution. Using A', one can compute $\mathcal{K}'(baua^{-1}b^{-1})$ in the matrix ring as follows:

$$A'WA'^{-1} = A'BUB^{-1}A'^{-1} = BA'UA'^{-1}B^{-1} = BVB^{-1} = \mathcal{K}'(baua^{-1}b^{-1}). \tag{7}$$

That is, the matrix A' plays the same role that the key a does. Thus we call such A' a *pseudo-key.*

4.2 A System of Linear Equations

We are able to change the above into an overdetermined system of linear equations of A. That is, we obtain the system of equations of the following form:

$$T_0 N = \begin{bmatrix} K \\ L_{n/2+1} \\ \vdots \\ L_{n-1} \end{bmatrix} X = 0, \tag{8}$$

where X is the column vector $[a_{11}, \ldots, a_{1m}; a_{21}, \ldots, a_{2m}; \ldots; a_{m1} \ldots, a_{mm}]^t$ made from $A = [a_{ij}]$ and K, L_i's are the $m^2 \times m^2$ matrix of the linear relations in Equation (5) and (6), respectively.

The system (8) has m^2 variables and $(n/2)m^2$ equations. However, by precise analysis of Krammer matrices, we can reduce the number of variables and equations as follows:

Theorem 3. *Equation (8) has at most $\frac{1}{7}n^4$ nontrivial variables and $\frac{1}{4}n^4$ nontrivial equations.*

Proof. Define V_k to be a subspace of V_0 generated by $\{x_{ij} | (i,j) \notin I_k\}$ where $I_k = \{(i,j) | 1 \le i < j < k \quad \text{or} \quad k+1 < i < j \le n\}$. From Equation (1), we see that the Krammer matrix $\mathcal{K}(\sigma_k)$ transforms V_k to itself and acts as the identity on the basis element x_{ij} when $(i,j) \in I_k$. Thus it can be written as $\begin{bmatrix} M_k & 0 \\ 0 & I \end{bmatrix}$ by reordering of the basis, where M_k is a square matrix of size $k(n-k)+n$ $(= \binom{n}{2} - \binom{k-1}{2} - \binom{n-k-1}{2})$.

Since $\cap_{1 \le k < n/2} I_k = \{(i,j) | n/2 \le i < j \le n\}$, a Krammer matrix of any left-braid $a \in LB_n$ can be written as $\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}$ where M is a square matrix of size $\frac{1}{8}(3n^2 - 2n - 8)$ $(= \binom{n}{2} - \binom{n/2+1}{2})$. Therefore only $\frac{1}{8^2}(3n^2 - 2n - 8)^2$ entries of A in Equation (5) are unknown.

This property of A reduces the number of equation in Equation (5) to $\binom{n}{2}^2 - \binom{n/2}{2}^2 \approx \frac{15}{64}n^4$. Also each equation in Equation (6) has only $k(n-k)+n$ non-trivial equations, whose sum for $n/2 \le k < n$ is about $\frac{1}{12}n^3$. Hence the total number of non-trivial equations are at most $\frac{1}{4}n^4$.

4.3 Estimate the Diffie-Hellman key

Theorem 4. *Let $u \in B_n$, $a \in LB_n$, and $b \in RB_n$ with ℓ canonical factors. Then $abub^{-1}a^{-1}$ can be written as a product of at most ℓ number of Δ^{-1} and at most 3ℓ number of canonical factors. Further each entry inside $\mathcal{K}'(abub^{-1}a^{-1})$ is a Laurent polynomial of t*

$$\sum_{d=-\ell}^{4\ell} \frac{a_i}{b_i} t^d \quad \textit{with } |a_i| \le 2^{\delta+2n\ell} \textit{ and } |b_i| \le 2^{8n\ell},$$

where δ is the number of Artin generators in $abub^{-1}a^{-1}$ bounded by $2\ell n(n-1)$.

Proof. Denote by $\mathrm{len}(x)$ the Charney length of x. Observe that $\mathrm{len}(xy) \le \mathrm{len}(x) + \mathrm{len}(y)$ for $x, y \in B_n$ and $\mathrm{len}(ab) \le \max(\mathrm{len}(a), \mathrm{len}(b))$ for $a \in LB_n$ and $b \in RB_n$. Also the inverse of x for $x \in B_n$ with r canonical factors is written as a product of at most r number of Δ^{-1} and at most r number of canonical factors. Since ab consists of at most ℓ canonical factors, we get the first assertion. The second assertion follows from Theorem 1.

Since u, v, and σ_k are positive braids, the entries of corresponding Krammer matrices are polynomial with rational coefficients. By multiplying the appropriate scalars to the both sides of Equations (5) and (6), we can consider $U, V, \mathcal{K}'(\sigma_i)$, and even A as matrices whose entries are polynomials with integer coefficients.

Let p be a prime with $p > 2^{\delta+10n\ell+1}$ and $f(t)$ an irreducible polynomial of degree 5ℓ over $\mathbb{Z}/p$. Since each entry of $\mathcal{K}(abub^{-1}a^{-1})$ is a polynomial of degree 5ℓ and with coefficient $< p$, we know that

$$\mathcal{K}'(baua^{-1}b^{-1}) = t^{-\ell}2^{-8n\ell}\{t^{\ell}2^{8n\ell}\mathcal{K}'(baua^{-1}b^{-1}) \mod (p, f(t))\} \tag{9}$$

if we take a representative of a residue class for coefficients from the interval $(-p/2, p/2)$. Therefore we are enough to compute $A \mod (p, f(t))$ in Equation (5) and (6). From the famous Bertrand's postulate below, it is guaranteed that $p < 2^{\delta+10n\ell+2}$.

Fact 4 (Bertrand's postulate) *[7] There exists a prime between n and $2n$.*

4.4 Algorithm and Complexity

The proposed algorithm to solve the braid Diffie-Hellman problem is described roughly as follows:

Algorithm 2 *Find an equivalent key using Gaussian Elimination.*

Input: *$u \in B_n$, $a \in LB_n$, $b \in RB_n$, $m = n(n-1)/2$, a prime p, and an irreducible polynomial $f(t)$ of the degree d satisfying Equation (9).*

Output: *$\mathcal{K}'(baua^{-1}b^{-1})$.*

1. *Compute the images of u and $v = aua^{-1}$ in $GL_m(k)$ via $\mathcal{K}'$, where k is the residue field $k = \mathbb{Z}[t]/(p, f(t))$.*
2. *Induce a system $\frac{1}{4}n^4$ linear equations in $\frac{1}{7}n^4$ variables from the simultaneous equations $\mathcal{K}'(v)A = A\mathcal{K}'(u)$ and $\mathcal{K}'(\sigma_i)A = A\mathcal{K}'(\sigma_i)$ for $n/2 < i \leq n$ over k*
3. *Apply Gaussian elimination for the system in order to compute A. We may multiply an appropriate integer to the both side of each equation to get integer coefficients.*
4. *If A is nonsingular, compute A^{-1}. Otherwise, go back to the above step and take another solution.*
5. *Compute $\mathcal{K}'(w)$ for $w = bub^{-1}$ and output $A\mathcal{K}'(w)A^{-1} = \mathcal{K}'(baua^{-1}b^{-1})$*
6. *Use Algorithm 1 to compute $baua^{-1}b^{-1}$.*

To evaluate the complexity of Gaussian elimination step, we need the following two facts:

Fact 5 *[17, p.15] The Gaussian elimination of an $m \times m$ matrix takes $\frac{1}{3}m^\tau$ for $\tau = \log_2 7$, which can be reduced to 2.376 theoretically.*

Fact 6 *[16] One multiplication or one inversion in a finite field with cardinality p^d takes $O(\log^\epsilon p^d)$ bit operations where $\epsilon > log_2 3$.*

Using the above facts, we can estimate the complexity of our algorithm as follows:

Theorem 5. *Assume LB_n and RB_n are two commuting subgroups of the n-braid group B_n. Given $u \in B_n, a^{-1}ua, b^{-1}ub$ for $a \in LB_n$ and $b \in RB_n$, $b^{-1}a^{-1}uab$ can be computed in about $2^{-5}n^{4\tau}(\ell\delta)^\epsilon$ (or $2^{-3}n^{4\tau+2\epsilon}\ell^{2\epsilon}$) bit operations where δ is the maximum word length of $abub^{-1}a^{-1}$ bounded by $2\ell n^2$.*

Proof. First, evaluate the complexity of Step 3. Since $p < 2^{\delta+10n\ell+2}$ and $d < 5\ell$, it is

$$\frac{1}{3}(\frac{1}{4}n^4)^\tau \log^\epsilon p^d \leq 2^{-5}n^{4\tau}\ell^\epsilon(\delta + 10n\ell + 2)^\epsilon \approx 2^{-5}n^{4\tau}(\ell\delta)^\epsilon. \tag{10}$$

The inverse of A can be computed in $O(n^3 \log^\epsilon(p^d))$. From Theorem 2, we know that recovering the braid awa^{-1} takes $O(2m^3 \log \ell)$ multiplications in k, which is about $O(n^6(\ell\delta)^\epsilon)$. The remainder takes very little. Hence the complexity of this algorithm is dominated by that of Gaussian elimination.

If we take $\tau = 2.8$ and $\epsilon = 1.6$, the complexity is $O(n^{14.4}\ell^{3.2})$. Theoretically, we can take $\tau = 2.376$ so that the complexity is $O(n^{9.5}\ell^{3.2})$.

In Table 1, we compare the attack complexity of braid encryption scheme, where n is the braid index and ℓ is the canonical length of a, b and u. The column [9] shows the complexity of the brute force attack with complexity $(\frac{n}{2}!)^\ell$ (the first three numbers were cited from [9] and the remainder was computed by $2^{n\ell}$ roughly since it is enough for this large number.) and the column [4] shows the super-summit attack with complexity $(n/2)^\ell$. The complexity of the proposed algorithm is evaluated by $2^{-3}n^{4\tau+2\epsilon}\ell^{2\epsilon}$ for $\tau = \log_2 7$ and $\epsilon \approx \log_2 3$. The column for ECC means the key size of elliptic curve cryptography with corresponding complexity (which was estimated roughly by square-root attacks such as Pollard ρ).

n	ℓ	[9]	[4]	Proposed Alg.	Key size of corr. ECC
50	5	2^{251}	2^{13}	2^{86}	172
70	7	2^{665}	2^{35}	$2^{95.4}$	191
90	12	2^{1863}	2^{66}	2^{103}	206
200	30	2^{6000}	2^{199}	2^{125}	250
1000	100	2^{10^5}	2^{900}	2^{162}	324
10000	1000	2^{10^7}	2^{12330}	2^{221}	442
100000	10000	2^{10^9}	$2^{1566666}$	2^{278}	556

Table 1. The performance of the attack algorithm

Note that the super-summit attack [4] is efficient for small n, but the proposed attack is efficient for large n since it has a *polynomial* complexity. The table shows that it is very hard to increase the complexity of braid encryption scheme, for example, in order to obtain similar complexity to 556 bit elliptic curve cryptography, the braid index should be about 10^5 (huge!!). Also in this case one cipher text must be about $10^9 \approx 2^{30}$ bits.

5 Conclusion

In this paper we proposed a polynomial time algorithm to solve the DHCP in braid groups. Though the complexity is too large to break the encryption scheme with the proposed parameters in [9] in real time, the braid encryption scheme is considered to be insecure since increasing the key size increase the attack complexity only a little. For example, to get the same complexity with 556 bit elliptic curve cryptography, the braid index should be about 10^5, which is impossible since one cipher text must be more than 10^9 bits.

We expect that the complexity can be reduced by more precise analysis on the Lawrence-Krammer representation. Further this analysis can be applied even to the generalized scheme based the decomposition problem [4] with the similar complexity since only the number of variables in the system of equations is doubled in the generalized version.

Since this cryptanalysis is based on the faithfulness of the Krammer representation, losing the group structure would be a possible way to avoid this kind of attacks. Currently, the key agreement scheme in [2] or the first key agreement scheme in [1] resists against this attack since it loses the group structure through the extractor map, so we cannot directly apply the same steps to obtain a pseudo-key [9].

Acknowledgements The authors would like to thank Ki Hyoung Ko, Sang Jin Lee, Eonkyung Lee, Jae Choon Cha, and Sang Geun Hahn for initial discussions on this problem, and the anonymous referees for their helpful comments. This work was supported by KISA (*R&D* 2002-S-073) and IRIS.

References

1. I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, *New Key Agreement Protocols in Braid Group Cryptography*, Proc. of CT-RSA 2001, Lecture Notes in Computer Science, Vol. 2020, Springer-Verlag, pp 13-27, 2001.

2. I. Anshel, M. Anshel, and D. Goldfeld, *An Algebraic Method for Public-Key Cryptography*, Math. Res. Lett., Vol. 6, No. 3-4, pp. 287-291, 1999.
3. J. Birman, K. Ko and S. Lee, *A New Approach to the Word and Conjugacy Problem in the Braid Groups*, Advances in Mathematics, Vol. 139, pp. 322-353, 1998.
4. J. Cha, K. Koh, S. Lee, J. Han, and J. Cheon, *An Efficient Implementations of Braid Groups*, Proc. of Asiacrypt 2001, Lecture Notes in Computer Science, Vol. 2248, Springer-Verlag, pp. 144-156, 2001.
5. R. Gennaro and D. Micciancio, *Cryptanalysis of a Pseudorandom Generator Based on Braid Groups*, Proc. of Eurocrypt 2002, Lecture Notes in Computer Science, Vol. 2332, Springer-Verlag, pp. 1-13, 2002.
6. D. Hofheinz and R. Steinwandt, *A Practical Attack on Some Braid Group Based Cryptography Primitives*, Proc. of PKC 2003, Lecture Notes in Computer Science, Vol. 2567, Springer-Verlag, pp. 187-198, 2003.
7. G. H. Hardy, E. M. Wright, *An introduction to the Theory of Numbers*, Oxford Univ. Press, 1978
8. K. Koh *et. al New Signature Scheme Using Conjugacy Problem*, Preprint, 2002.
9. K. Ko, S. Lee, J. Cheon, J. Han, J. Kang, C. Park, *New Pulic-key Cryptosystem using Braid Groups*, Proc. of Crypto 2000, Lecture Notes in Computer Science, Vol. 1880, Springer-Verlag, pp. 166-183, 2000
10. D. Krammer, *The Braid group B_4 is Linear*, Inventiones Mathematics, Vol. 142, pp. 451-486, 2002.
11. D. Krammer, *Braid groups are Linear*, Annals of Mathematics, Vol. 155, pp. 131-156, 2002.
12. S. Lee, *The Trapdoor Oneway Functions in Braid Groups*, Workshop on Algbraic Methods in Cryptography, Slides are available in http://knot.kaist.ac.kr/ sjlee.
13. S. Lee and E. Lee, *Potential Weaknesses of the Commutator Key Agreement Protocol Based on Braid Groups*, Proc. of Eurocrypt 2002, Lecture Notes in Computer Science, Vol. 2332, Springer-Verlag, pp. 14-28, 2002.
14. E. Lee and J. Park, *Cryptanalysis of the Public-key Encryption based on Braid Groups*, Proc. of Eurocrypt 2003, Lecture Notes in Computer Science, Vol. 2656, Springer-Verlag, pp. 477-490, 2003.
15. E. Lee, S. J. Lee and S. G. Hahn, *Pseudorandomness from Braid Groups*, Proc. of Crypto 2001, Lecture Notes in Computer Science, Vol. 2139, Springer-Verlag, pp. 486-502, 2001.
16. A. Menezes, P. Oorschot, and S. Vanston, *Handbook of Applied Cryptography*, CRC Press, 1997.
17. G. Strang, *Linear Algebra and its Applications*, Harcourt, 1986.