The trapdoor oneway functions in braid groups

Sangjin Lee

sjlee@cmi.univ-mrs.fr

http://www.cmi.univ-mrs.fr/∼sjlee

CMI, Université de Provence, Marseille, France

Nov. 9, 2001

Overview

1. Braid groups and conjugacy problem
2. Cryptosystems using braid gorups
3. Hardness of multiple conjugacy problem

Braid Groups and conjugacy problem

Braid groups have rich theories.

1. geometric objects
   (conjugacy problem in braid groups is important in low dimension topology)

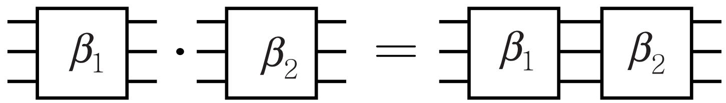
2. combinatorial group theoretic approach
   (automatic group structure, positive braid monoid)

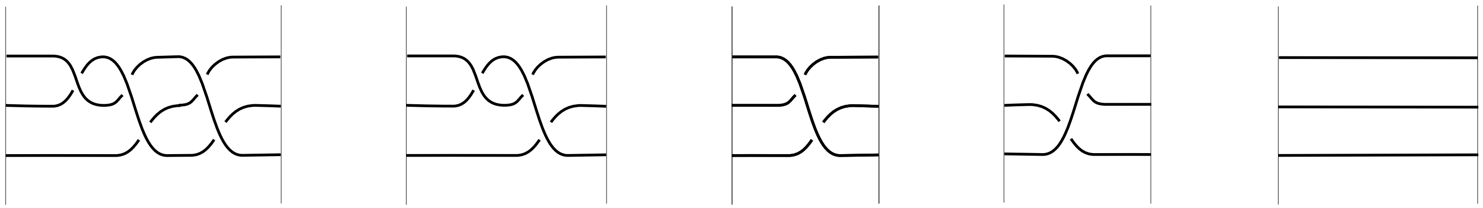
3. topological dynamics

4. representations into matrix groups
   (residually finite, faithful representation)

Geometric definition

$B_n = \{\text{collection of } n \text{ intertwining strings}\}/\text{isotopy}$

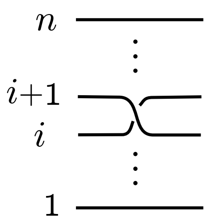
B_n is a group under the multiplication 



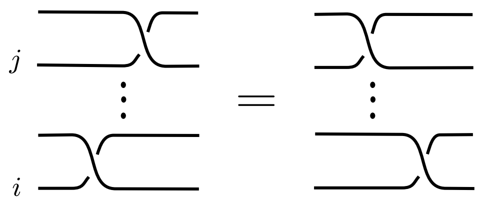
$\beta_1\beta_2$ β_1 β_2 β_2^{-1} identity

Artin presentation

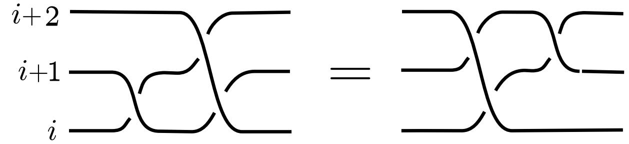
$$B_n = \left\langle \sigma_1, \ldots, \sigma_{n-1} \;\middle|\; \begin{array}{cc} \sigma_i\sigma_j = \sigma_j\sigma_i, & (|i-j|\geq 2) \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}, & (i=1,\ldots,n-2) \end{array} \right\rangle$$



σ_i



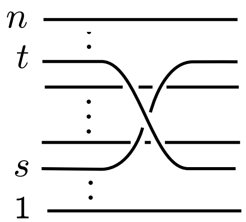
$\sigma_i\sigma_j=\sigma_j\sigma_i$



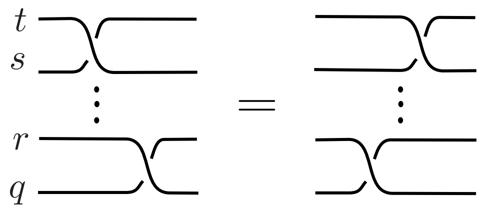
$\sigma_i\sigma_{i+1}\sigma_i=\sigma_{i+1}\sigma_i\sigma_{i+1}$

Band generator presentation

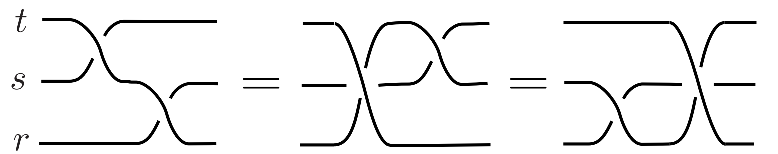
$$B_n = \left\langle \begin{array}{c} a_{ts} \\ (n\geq t>s\geq 1) \end{array} \;\middle|\; \begin{array}{cc} a_{ts}a_{rq} = a_{rq}a_{ts}, & (t>s>r>q \text{ or } t>r>q>s) \\ a_{ts}a_{sr} = a_{tr}a_{ts} = a_{sr}a_{tr}, & (t>s>r) \end{array} \right\rangle$$



a_{ts}



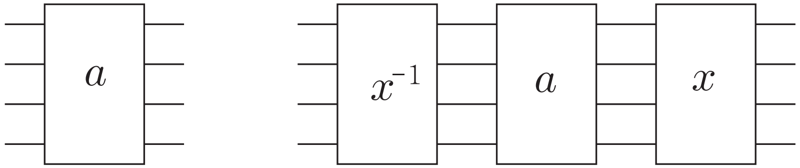
$a_{ts}a_{rq}=a_{rq}a_{ts}$



$a_{ts}a_{sr}=a_{tr}a_{ts}=a_{sr}a_{tr}$

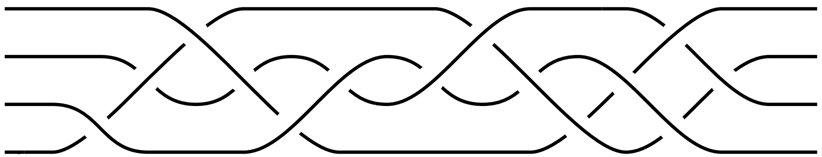
Conjugacy search problem

Given $(a, b) \in B_n^2$, where $b = x^{-1}ax$ for some secret $x \in B_n$.
Find $x \in B_n$ such that $b = x^{-1}ax$.

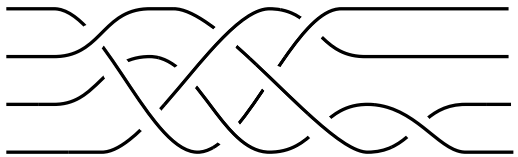


1. Conjugacy search problem seems to be intractable.
2. We need a trapdoor.

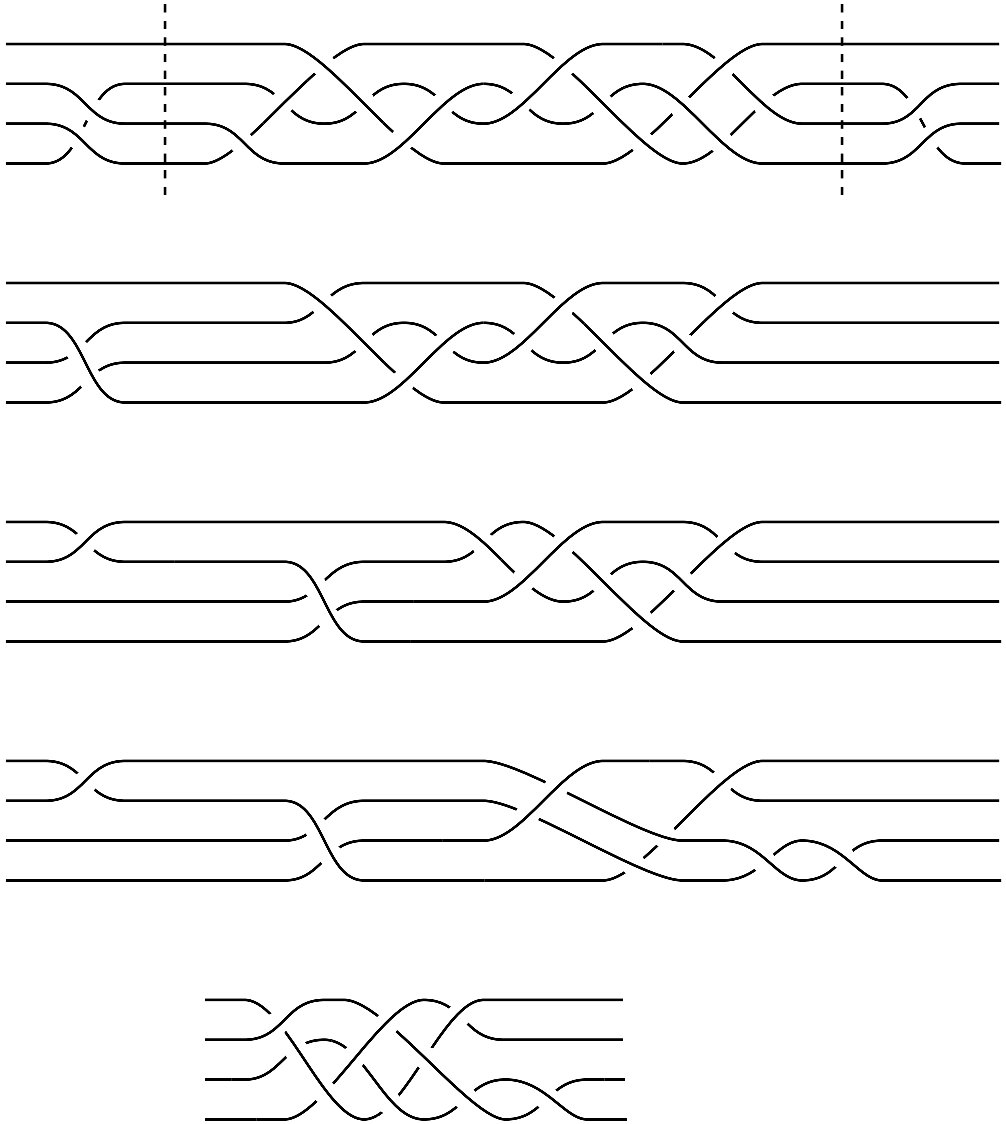
Example: $b = (\sigma_1\sigma_2)a(\sigma_1\sigma_2)^{-1}$



$a = \sigma_1\sigma_2^{-1}\sigma_3\sigma_2\sigma_1^{-1}\sigma_2^{-2}\sigma_3^{-1}\sigma_2\sigma_1\sigma_2\sigma_1\sigma_3^{-1}$



$b = \sigma_3^{-1}\sigma_2\sigma_1\sigma_2^{-1}\sigma_3^{-1}\sigma_1\sigma_2\sigma_3^{-1}\sigma_1^2$



These figures show the isotopy from $(\sigma_1\sigma_2)a(\sigma_1\sigma_2)^{-1}$ to b. This example shows that $x^{-1}ax$ looks very different from a even with simple x.

Nice features of braid groups

1. There is an efficient encoding of braids.
   n-braid can be encoded as $(u, \pi_1, \ldots, \pi_\ell)$, $u \in \mathbf{Z}$, π_i n-permutation.
   It can be encoded using '$\ell n \log n$' bits.
   The number of such braids are roughly $2^{\ell n \log n}$.

2. There are efficient algorithms to do group operations.
   If the sizes of braids used in a cryptosystem are reasonable in practical use, then the cryptosystem would be efficient.
   Parallel processing is possible.

3. The conjugacy problem is intractable. But we need a trapdoor.

4. There are rich theories which can be used to make braid cryptosystem reliable.

Cryptosystems using braid groups

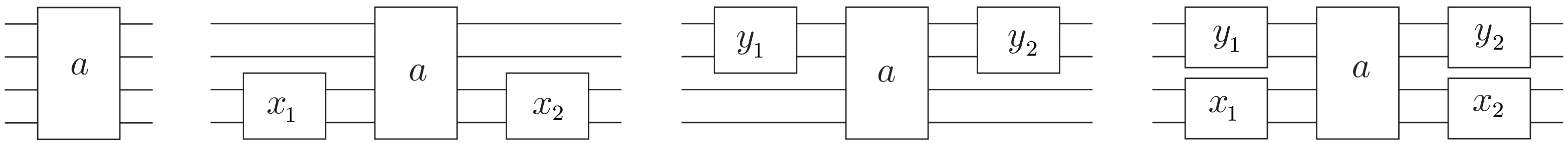
1. Cryptosystems using braid DH-problem
2. Commutator key agreement protocol of Anshel-Anshel-Fisher-Goldfeld

Cryptosystems using braid DH-problem

Let $n = 2m$, $LB_n = \langle \sigma_1, \ldots, \sigma_{m-1} \rangle$ and $UB_n = \langle \sigma_{m+1}, \ldots, \sigma_{2m} \rangle$.

Decomposition Problem

Given $(a, b) \in B_n^2$, where $b = x_1 a x_2$ for some $x_1, x_2 \in LB_n$,
Find $x_1, x_2 \in LB_n$ such that $b = x_1 a x_2$.



Braid DH-problem

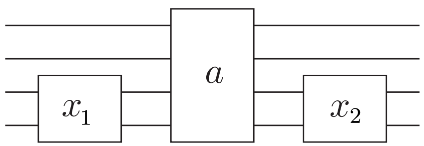
Given $(a, b_1, b_2) \in B_n^3$, where $b_1 = x_1 a x_2$ and $b_2 = y_1 a y_2$
for some $x_1, x_2 \in LB_n$ and $y_1, y_2 \in UB_n$.
Find $y_1 b_1 y_2$ $(= x_1 b_2 x_2 = x_1 y_1 a x_2 y_2)$.

Key Agreement Scheme

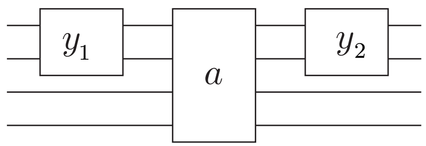
$a \in B_n$ is published. (Assume that Braid DH-problem is hard for a.)

A		B
chooses secret $x_1, x_2 \in LB_n$. computes $b_1 = x_1 a x_2$.	$\xrightarrow{b_1}$ $\xleftarrow{b_2}$	chooses secret $y_1, y_2 \in UB_n$. computes $b_2 = y_1 a y_2$.
computes $K_A = x_1 b_2 x_2$.		computes $K_B = y_1 b_1 y_2$.

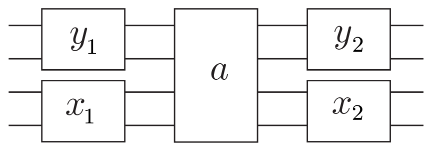
Here, $K_A = x_1 b_2 x_2 = x_1 (y_1 a y_2) x_2 = y_1 (x_1 a x_2) y_2 = y_1 b_1 y_2 = K_B$.



b_1



b_2



$K_A = K_B$

Some remarks

1. We can make a PKC using the key agreement scheme.

2. Current security level is approximately $(n/2)^l$.
   Even with this level, braid PKC is practical.
   – Weakness arising from the trapdoor property.
   – Security level will be improved.

3. BAD NEWS: It is difficult to find a secure public key a.

4. GOOD NEWS: After choosing a secure public key a, it is easy to make a private key (x_1, x_2) and a session key (y_1, y_2).

Commutator key agreement protocol

- **Multiple Simultaneous Conjugacy Search Problem with Restricted Conjugator(MSCP)**

Given $(a_1, \ldots, a_s), (b_1, \ldots, b_s) \in (B_n)^s$.
Find $x \in B_n$ such that $b_i = x^{-1}a_i x$ *simultaneously* for all i.
$(x \in \langle c_1, \ldots, c_t \rangle$, publicly known subgroup.)

$$\begin{cases} x^{-1}a_1 x = b_1 \\ x^{-1}a_2 x = b_2 \\ \quad\cdots \\ x^{-1}a_s x = b_s \end{cases} \qquad \text{and} \qquad x \in \langle c_1, \ldots, c_t \rangle$$

1. The conjugation is simultaneous.
2. The conjugator x is in the subgroup generated by $\{c_1, \ldots, c_t\}$.
   (cf. Length attack of J. Hughes and A. Tannenbaum.)

- **Commutator Conjugacy Problem**

Given $(a_1,\ldots,a_s)$, $(b_1,\ldots,b_s)$, $(c_1,\ldots,c_t)$, $(d_1,\ldots,d_t)$.   (It is known that $x^{-1}a_ix=b_i$ and $y^{-1}c_iy=d_i$ for some $x\in\langle c_1,\ldots,c_t\rangle$ and $y\in\langle a_1,\ldots,a_s\rangle$.)   Find the commutator $[x,y]=x^{-1}y^{-1}xy$.

$$\begin{cases} x^{-1}a_1x=b_1 \\ x^{-1}a_2x=b_2 \\ \cdots \\ x^{-1}a_sx=b_s \end{cases} \qquad \begin{cases} y^{-1}c_1y=d_1 \\ y^{-1}c_2y=d_2 \\ \cdots \\ y^{-1}c_ty=d_t \end{cases}$$

$$x\in\langle c_1,\ldots,c_t\rangle \qquad y\in\langle a_1,\ldots,a_s\rangle$$

Key Agreement Scheme

A publishes $(a_1, \ldots, a_s) \in B_n^s$. B publishes $(c_1, \ldots, c_t) \in B_n^t$.

A		B
chooses $x = W(a_1, \ldots, a_s)$. computes $d_i = x^{-1} c_i x$.	$\xrightarrow{(d_1, \ldots, d_t)}$ $\xleftarrow{(b_1, \ldots, b_s)}$	chooses $y = V(c_1, \ldots, c_t)$. computes $b_i = y^{-1} a_i y$.
computes $K_A = x^{-1} W(b_1, \ldots, b_s)$.		computes $K_B = (V(d_1, \ldots, d_t))^{-1} y$.

$$\left\{ \begin{array}{l} W(b_i) = W(y^{-1} a_i y) = y^{-1} W(a_i) y = y^{-1} x y \\ V(d_i) = V(x^{-1} c_i x) = x^{-1} V(c_i) x = x^{-1} y x \end{array} \right\} \Rightarrow \quad K_A = K_B.$$

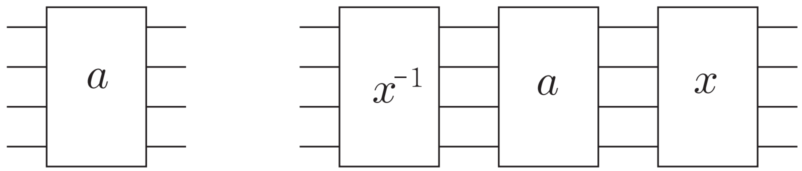
Hardness of multiple conjugacy problem

(joint work with Eonkyung Lee)

1. Algorithm for original conjugacy problem
2. Algorithm for multiple simultaneous conjugacy problem(MSCP)

Solution of Conjugacy Problem

Given $(a, b) \in B_n^2$, where $b = x^{-1}ax$ for some secret $x \in B_n$.
Find $x \in B_n$ such that $b = x^{-1}ax$.



Strategy

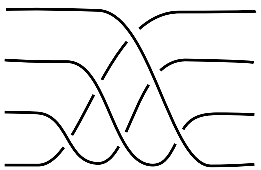
1. Simplify the instance: find $a' \underset{\text{conj}}{\sim} a$ and $b' \underset{\text{conj}}{\sim} b$ which are simplest.
2. Find the conjugator: find a sequence
   $a' = a_0 \to a_1 \to \cdots \to a_N = b'$ such that
   (i) a_i's are also simplest, and
   (ii) $a_i = H_i^{-1} a_{i-1} H_i$ for some simple braid H_i.

Left-canonical(left-greedy, normal) form

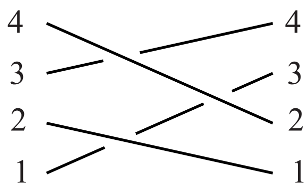
Every braid $\alpha \in B_n$ is uniquely expressed as

$$\alpha = \Delta^u A_1 A_2 \cdots A_k,$$

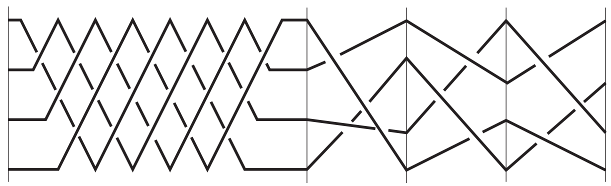
where $\Delta = (\sigma_1 \cdots \sigma_{n-1}) \cdots (\sigma_1\sigma_2)(\sigma_1)$ is the *fundamental braid*, A_i's are *permutation braids*, $A_i \notin \{e, \Delta\}$, and each $A_i A_{i+1}$ is *left-weighted*. ($\Rightarrow$ α can be encoded as $(u, \pi_1, \ldots, \pi_k)$, $u \in \mathbf{Z}$, π_i's n-permutations.)



Δ in B_4



permu. braid
corres. to 3142



Left canonical form
$\Delta^{-3}(\sigma_3\sigma_1\sigma_2\sigma_1)(\sigma_2\sigma_1\sigma_3)(\sigma_1\sigma_3\sigma_2)$
$(-3, 3241, 2413, 3142)$

NOTE: We can say that α is simple if $k = \mathrm{len}(\alpha)$ small
(in conjugacy class, if $u = \inf(\alpha)$ large, or $u + k = \sup(\alpha)$ small).

Super summit set

- Let $a = \Delta^u A_1 \cdots A_k$ LCF, $(\tau(a) = \Delta^{-1} a \Delta \Rightarrow \tau(\sigma_i) = \sigma_{n-i},\ \tau^2 = \mathrm{id})$

$$\text{cycling, } \mathbf{c}(a) := \Delta^u A_2 \cdots A_k \tau^u(A_1),$$
$$\text{decycling, } \mathbf{d}(a) := \Delta^u \tau^u(A_k) A_1 \cdots A_{k-1}.$$

  RHS is *not* in canonical form, and must be rearranged.

- Cycling Theorem
  (i) $\inf(a)$ not maximal $\Rightarrow \inf \mathbf{c}^l(a) > \inf(a)$ for some $l \leq \frac{n(n-1)}{2}$.
  (ii) $\sup(a)$ not minimal $\Rightarrow \sup \mathbf{d}^l(a) < \sup(a)$ for some $l \leq \frac{n(n-1)}{2}$.
  $\Rightarrow$ 'maximal inf' and 'minimal sup' can be achieved simultaneously.

- Convexity theorem: $x = \Delta^u H_1 \cdots H_i \cdots H_k$ LCF and $b = x^{-1} a x$.
  Let $a_i = (\Delta^u H_1 \cdots H_i)^{-1} a (\Delta^u H_1 \cdots H_i)$. Then
$$\inf(a_i) \geq \min\{\inf(a), \inf(b)\}, \quad \sup(a_i) \leq \max\{\sup(a), \sup(b)\}.$$

- We can compute super summit set: $a' \in \mathrm{SSS}(a)$ iff $a' \underset{\text{conj}}{\sim} a$ and $\inf(a')$ maximal and $\sup(a')$ minimal in conjugacy class.

$$
\begin{array}{llllllll}
a & & & & & & & b \\
\downarrow (a'=P^{-1}aP) & & & & & & & \downarrow (b=Q^{-1}b'Q) \\
a' = a_0 \rightarrow & a_1 & \rightarrow & a_2 & \rightarrow \cdots \rightarrow & a_N & = b' & \\
 & \| & & \| & & \| & & \\
 & H_1^{-1}a_0H_1 & & H_2^{-1}a_1H_2 & & H_N^{-1}a_{N-1}H_N & &
\end{array}
$$

$$
\begin{aligned}
a &= Pa'P^{-1} = Pa_0P^{-1} = (PH_1)a_1(H_1^{-1}P^{-1}) \\
&\quad \cdots\cdots \\
&= (PH_1 \cdots H_N)a_N(H_N^{-1} \cdots H_1^{-1}P^{-1}) \\
&= (PH_1 \cdots H_N)b'(H_N^{-1} \cdots H_1^{-1}P^{-1}) \\
&= (PH_1 \ldots H_NQ)b(PH_1 \ldots H_NQ)^{-1}
\end{aligned}
$$

Multiple simultaneous conjugacy problem

Given $(a_1, \ldots, a_s), (b_1, \ldots, b_s) \in (B_n)^s$.

Find $x \in B_n$ such that $b_i = x^{-1} a_i x$ *simultaneously* for all i.

($x \in \langle c_1, \ldots, c_t \rangle$, publicly known subgroup.)

$$\left\{ \begin{array}{c} x^{-1} a_1 x = b_1 \\ x^{-1} a_2 x = b_2 \\ \cdots \\ x^{-1} a_s x = b_s \end{array} \right. \qquad \left\{ \begin{array}{l} 1.\ n \geq 80, \quad s = t = 20 \\ 2.\ a_i:\ \text{product of 5 to 10 } \sigma_j\text{'s.(simple)} \\ 3.\ x:\ \text{product of 100 } c_j\text{'s.(complicated)} \\ 4.\ b_i = x^{-1} a_i x.\ \text{(complicated)} \end{array} \right.$$

1. x is unique up to Center$(B_n) = \langle \Delta^2 \rangle$ for generic instance.

2. The strategy for the ordinary conjugacy problem works.
   – simplifying the instance: easy
   – finding the conjugator: hard

Cycling theorem

Theorem. $b = x^{-1}ax$. $a = P_1 A_1 \Delta^v$ RCF. $b = \Delta^w B_1 Q_1$ LCF.

1. If $\inf(b) < \inf(a)$, $x = x_1 C_1$ for some $x_1 \in B_n$ and a permutation braid C_1 such that $\inf(x) = \inf(x_1)$ and $C_1 = \Delta\tau^w(B_1^{-1})$.
2. If $\inf(b) > \inf(a)$, $x = C_1 x_1$ for some $x_1 \in B_n$ and a permutation braid C_1 such that $\inf(x) = \inf(x_1)$ and $C_1 = \tau^v(B_1^{-1})\Delta$.

* Note that C_1 depend only on (a, b).

Proof of 1. Let $x = \Delta^u R$, $\inf(x) = u$. From '$xb = ax$' and '$v > w$',

$$
\begin{aligned}
&(\Delta^u R)(\Delta^w B_1 Q_1) = (\Delta^v A_1 P_1)(\Delta^u R) \\
&\Rightarrow R\tau^w(B_1 Q_1) = \Delta^{v-w}\tau^{u-w}(A_1 P_1)\tau^w(R), \\
&\Rightarrow \inf(R\tau^w(B_1 Q_1)) \geq 1 \Rightarrow \inf(R\tau^w(B_1)) \geq 1 \\
&\Rightarrow R = R_1 C_1 \text{ such that } C_1\tau^w(B_1) = \Delta \text{ } (C_1 \text{ permutation braid}) \\
&\Rightarrow C_1 = \Delta\tau^w(B_1)^{-1}
\end{aligned}
$$

$$
\begin{array}{ccccccc}
(a_1,\ldots,a_s) & & & & & & (b_1,\ldots,b_s) \\
\downarrow {\scriptstyle (a_i'=P^{-1}a_iP)} & & & & & & {\scriptstyle (b_i'=Qb_iP^{-1})} \downarrow \\
(a_1',\ldots,a_s') & \to & \underset{a_i''=H_1^{-1}a_i'H_1}{(a_1'',\ldots,a_s'')} & \to & \underset{a_i'''=H_2^{-1}a_i''H_2}{(a_1''',\ldots,a_s''')} & \to\cdots\to & (b_1'\ldots,b_s')
\end{array}
$$

Remarks

- a_i''s and b_i''s are as simple as a_i's for each i in terms of 'inf'.
- We can find P, Q immediately.
  And $x = Px_1Q$ for some $x_1 \in B_n$ such that $\inf(x) = \inf(x_1)$.
- In some cases, $x = PQ$ or $x = P\Delta Q$(ex. all $a_i \in B_n^+$ for all i.)

Finding the remaining conjugator

1. The difficulty is that the number of permutation braids, $n! \sim 2^{n \log n}$, is an infeasible number. ($18! \geq 2^{52}$, $80! \geq 2^{394}$) (in band-gen. pres. $\mathcal{C}_n = \frac{(2n)!}{n!(n+1)!} \sim 4^n$. $\mathcal{C}_{30} \geq 2^{51}$, $\mathcal{C}_{80} \geq 2^{149}$.)

2. In some cases, we can avoid the infeasible numbers $n!$ and $\mathcal{C}_n$.
   (a) If $a_i = \sigma_j$ for some i, then the induced permutation π_1 of H_1 should satisfy $|\pi_1(j) - \pi_1(j+1)| = 1$.
   (b) If $a_i = \sigma_j \sigma_k$, $|j-k| \geq 2$, then the induced permutation π_1 of H_1 should satisfy $|\pi_1(j) - \pi_1(j+1)| = |\pi_1(k) - \pi_1(k+1)| = 1$.

3. If we cannot avoid avoid the infeasible numbers $n!$ and $\mathcal{C}_n$, then the AAG scheme would be secure. But the parameter recommendation that a_i's are product of small number of Artin generators, introduces some degree of security reduction.

Further works on braid cryptosystems

- Find a trapdoor which weakens the conjugacy problem less.
  We must take the following things into account: the information on the induced permutations, the representations into matrix groups, the combinatorial group theoretic approach and the topological dynamics.
- Make other applications.
- What is random braid?
  Because braid groups are infinite group, we need a discussion on the randomness in braid groups.

Yamamura's scheme and conjugacy problem in B_3

B_3 is closely related to $PSL_2(\mathbf{Z})$ and share the normal form.

- $\bar{\rho} : B_3 \to PSL_2(Z)$ defined by $\rho(\sigma_1) = AB$, $\rho(\sigma_2^{-1}) = A^2B$, where $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, with $\ker \bar{\rho} = \langle \Delta^2 \rangle$.
- (Murasugi) Every $a \in B_3$ is conjugate to $\Delta^{2k} P(\sigma_1, \sigma_2^{-1})$ for some k and a positive word P (unique as a cyclic word).
- $V \in PSL_2(\mathbf{Z})$ is in normal form iff $V = P(AB, A^2B)$ for a positive word P, by moving B from the left to right if necessary.

We can compute the secret information of Yamamura's scheme.

- PK : $(W_1 = M^{-1}V_1M, W_2 = M^{-1}V_2M, q = M^{-1}p)$ and SK : p.
- By using the (quadratic) conjugacy algorithm in B_3, we can find, from (W_1, W_2), the hidden matrices (V_1, V_2, M), and hence p from (M, q).

— THANK YOU —