

braided group cryptography

First Approach (Commutator Protocol)

Let s_1, \dots, s_n and t_1, \dots, t_n be braids which Alice and Bob choose publicly.

Alice chooses a secret word in Σ

$a = s_{i_1} \dots s_{i_k}$ and sends Bob $a^{-1}t_1a, a^{-1}t_2a, \dots, a^{-1}t_na$ in order, using greedy normal form.

Bob chooses a secret $b = t_{j_1}t_{j_2}\dots t_{j_\ell}$ and sends Alice

$b^{-1}s_1b, \dots, b^{-1}s_nb$ similarly.

Both Alice and Bob may now compute $a^{-1}b^{-1}ab$ efficiently, for instance Alice computes

$$a^{-1} \cdot (b^{-1}s_{i_1}b, b^{-1}s_{i_2}b, \dots, b^{-1}s_{i_k}b) = a^{-1}b^{-1}ab$$

However, an eavesdropper, Eve, must solve the multiple conjugator search problem

2nd Approach (Diffie-Hellman Conjugator Protocol)

Original method:

Let $g \in F$, a finite field, be chosen publicly.

Alice chooses $a \in \mathbb{N}$ and sends Bob g^a .

Bob chooses $b \in \mathbb{N}$ and sends g^b .

Both sides then compute g^{ab} .

To find g^{ab} , Eve must solve the Diffie Hellman problem.

Braid version:

Alice and Bob choose a long braid in B_{2n} .

Let $L = \langle \sigma_1, \dots, \sigma_{n-1} \rangle$

$U = \langle \sigma_{n+1}, \dots, \sigma_{2n-1} \rangle$

These subgroups of B_{2n} commute.

Alice chooses $a \in L$ and sends Bob $a^{-1}xa$.

Bob chooses $b \in U$ and sends $b^{-1}xb$.

Both sides compute $a^{-1}b^{-1}xba$.

Last time

Commutator Protocol

s_1, \dots, s_m

t_1, \dots, t_n

Alice: $a = s_{i_1} \dots s_{i_k}$, sends $a^{-1}t_1a, a^{-1}t_2a, \dots, a^{-1}t_na$

Bob: $b = t_{j_1} \dots t_{j_\ell}$, sends $b^{-1}s_1b, \dots, b^{-1}s_mb$

shared secret: $a^{-1}b^{-1}ab$

Length Attack: (problem: B_n is a metric space)

Def: define $d: B^n \times B^n \rightarrow \mathbb{N}$ by $d(x, y) = \text{length}(\text{GNF}(xy^{-1}))$

Then 1) $d(x, y) \geq 0$, with $d(x, y) = 0 \Leftrightarrow x = y$

2) $d(x, y) + d(y, z) \leq d(x, z)$

(not symmetric, but $d'(x, y) := d(x, y) + d(y, x)$ is.)

Now, with some probability, we get

$d(s_i^{-1}a^{-1}tas_i, t) \leq d(a^{-1}ta, t)$ if s_i is the last generator in a ,

$d(s_i^{-1}a^{-1}tas_i, t) > d(a^{-1}ta, t)$ otherwise.

The more tangled the s_i are, the more robust this effect is. (that is, the lower the probability that $\text{length}(\text{GNF}(s_1s_2))$ is short, the higher the probability to get the inequalities above).

This allows a probabilistically guided search for secret a (or b).

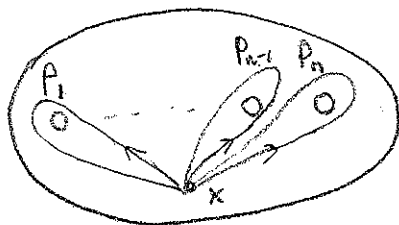
Proposed Solution:

- 1) use short words to make length attack harder
- 2) pass to a representation to "encrypt" the braids.
(Want an easily computable representation that hides word length)

The Burau representation:

Consider a disk with n punctures D_n .
Let $x \in D_n$.

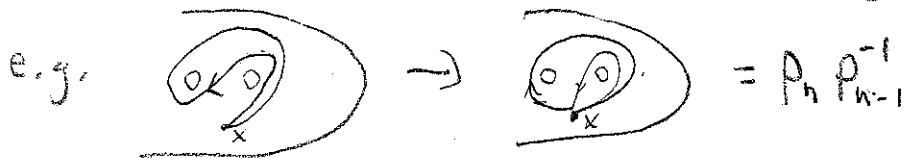
Draw the following paths P_1, \dots, P_n from x to x .



$$P_i: [0, 1] \rightarrow D_n$$

$$P_i(0) = P_i(1) = x$$

Every path from x to x can be continuously deformed into a sequence of paths P_1, \dots, P_m and their reverses.
(The length zero sequence is the constant path).



These deformations are called homotopies.

Then $\{ \text{paths based at } x \text{ in } D_n \} / \text{homotopies}$

forms a group with reverse = inverse, concatenation = product

This group is $\pi_1(D_n)$, the first homotopy group on D_n . B_n acts on D_n (by sliding the punctures), which gives an action on $\pi_1(D_n)$.

The Burau representation records information about the action of B_n on $\pi_1(D_n)$.

Problem: A representation is a Group action on an F -module (i.e. vector space), but $\pi_1(D_n)$ is not abelian.

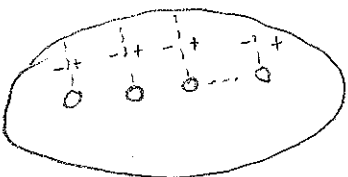
One could try to abelianize $\pi_1(D_n)$. One gets

$\text{Ab}(\pi_1(D_n)) = H_1(D_n)$, the first homology group of D_n .

It works: $H_1(D_n) \cong \mathbb{Z}^n$ and B_n acts on \mathbb{Z}^n (and we can extend this to an action on \mathbb{C}^n).

However, this rep'n is boring (factors through a rep'n of S_n).

Solution: cleverly encode non-abelian information in an abelian way, as follows:

- ① make \mathbb{Z} -many copies of D_n
- ② cut as follows: 

- ③ Glue the "+" part of each cut in the n 'th copy to the "-" part of that cut in the $n+1$ 'st copy. Call the resulting space \tilde{D}

Now if we trace the path p_n in \tilde{D}_n it is not a loop; it goes from x in the n -th copy of D_n to x in the next copy.

A set of generators for $\pi_1(\tilde{D}_n)$ can be expressed as $b_j = p_j^k \cdot p_{j+1} p_j^{-k-1}$ for all $k \in \mathbb{Z}$, $j \in 2, \dots, n$.

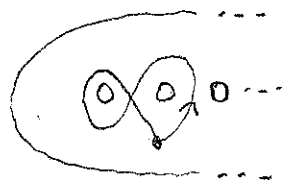
Then $Ab(\pi_1(\tilde{D}_n)) = H_1(\tilde{D}_n)$.

Let the formal variable t act on elements of $\pi_1(\tilde{D}_n)$ via

$$t \cdot p_{j+1} p_j^{-1} = p_j p_{j+1} p_j^{-2}$$

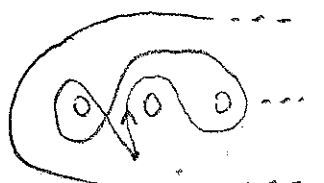
Then $H_1(\tilde{D}_n)$ is a $\mathbb{Z}[t, t^{-1}]$ module.
(with basis elts $b_j = p_{j+1} p_j^{-1}$)

e.g.



$$= p_2 p_1^{-1} \rightarrow b_1$$

↓
action by σ_2



$$= p_2^{-1} p_3 p_2 p_1^{-1} = (p_2^{-1} p_3) (p_2 p_1^{-1})$$

$$\rightarrow t^{-1} \cdot b_2 + b_1$$

B_n acts on $H_1(\tilde{D}_n)$ via the (reduced) Burau representation

One then obtains the following:

$$P(\sigma_i)(t) := \begin{cases} \begin{pmatrix} -t & 0 \\ t & 1 \end{pmatrix} \oplus I_{n-3} & \text{if } i=1 \\ I_{i-2} \oplus \begin{pmatrix} 1 & 1 & 0 \\ 0 & -t & 0 \\ 0 & t & 0 \end{pmatrix} \oplus I_{n-i-2} & \text{if } 1 < i < n-1 \\ I_{n-3} \oplus \begin{pmatrix} 1 & 1 \\ 0 & -t \end{pmatrix} & \text{otherwise.} \end{cases}$$

Note that setting $t=1$ gives a representation of S_n (the bary one mentioned earlier), so the Burau rep. can be thought of as a deformation of a rep of S_n .

Prop: The Burau representation at $t=1$ factors through the standard representation of S_n .

PF (sketch): Establish a bijection by noticing that ρ permutes the vertices of an $(n-1)$ -simplex in \mathbb{C}^{n-1} , with vertices given by

$$v_1 = (1, 0)$$

$$v_i = (0, \dots, 0, \underset{\substack{\uparrow \\ i\text{th} \\ \text{column}}}{-1}, 1, 0, \dots, 0) \text{ for } 1 < i < n$$

$$v_n = (0, \dots, 0, -1)$$

Then σ_i acts by permuting v_i and v_{i+1}

The Burau representation is

- 1) sparse
- 2) almost faithful (long believed to be faithful, still open when $n=4$)
- 3) small dimension

Cryptography again:

Unfortunately, Burau representations have too much structure. Using this structure, it is possible to extract the shared key in the commutator protocol by solving a system of linear equations.

This attack works best when the s_i are small (which we wanted in order to thwart the length attack).

One might expect that if B_n has a faithful representation, one could derive information about braids using linear algebra in the representation. A group which has a faithful representation is called a linear group.

In fact, B_n has a faithful representation, called the Lawrence-Krammer Representation, which can be used to attack the Diffie-Hellman conjugator protocol.

One finds a matrix A s.t., $A P_{b^{-1}xb} A^{-1} = P_{a^{-1}b^{-1}xba}$ in polynomial time. From $P_{a^{-1}b^{-1}xba}$ one can use special properties of the L-K representation to recover $a^{-1}b^{-1}xba$. Interestingly, in this solution $A \neq P_{b^{-1}}$ in general, so linearity is being used in a fundamental way.