Braid Group Cryptography - Liam Hardiman

Background

A **finitely presented group** is specified by the following data.

1. **Generators** $x_1, x_2, \ldots, x_n$. Just a set of symbols.
2. **Relators** $r_1 = e$, $r_2 = e$, $\ldots$, $r_m = e$. More on these in a moment.

For each generator x_i there is a corresponding inverse, x_i^{-1}. A **word** is just a finite string made of the symbols x_i and x_i^{-1}. The empty string e is a word and will be the identity element in G. The relators are words.

G consists of all *equivalence classes* of words, where two words u and v are equivalent if u can be transformed into v by a finite sequence of cancellations or eliminating/introducing relators.

Equivalently, G is a quotient of the free group on the generators modulo the normal closure of the relators.

The Word Problem

The **word problem** is the decision problem that asks whether two words u and v are equivalent in G. In some finitely presented groups this is **undecidable** – it is provably impossible to give an algorithm that always outputs a correct answer[1].

The Conjugacy Problem

The **conjugacy search problem** asks, given two conjugate words u and v in G, for a word w such that $u = w^{-1}vw = v^w$. Like the word problem, this is undecidable in some finitely presented groups.

The Braid Group

Symbolically, the braid group on n strands has the following presentation

$$B_n = \langle \sigma_1, \sigma_2, \ldots, \sigma_{n-1} \mid \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ if } |i-j| = 1,\ \sigma_i\sigma_j = \sigma_j\sigma_i \text{ if } |i-j| > 1 \rangle.$$

Visually, imagine two sets of n items arranged in vertical lines on either side of the page. Attach one end of a string to each item on the left side of the page. To each item on the right side attach the other end of one string. This connection is a **braid**. The strings might cross over or under one another any number of times and we say each such configuration gives a *different* braid.

The generator σ_i represents the braid formed by crossing strand i under strand $i+1$ and leaving the other strings fixed.

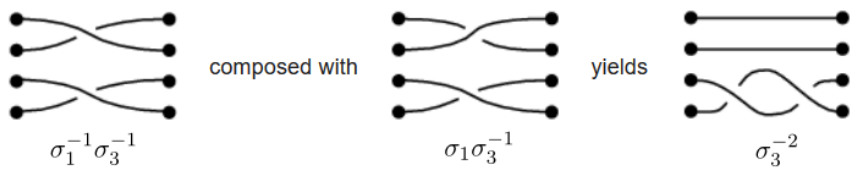

Figure 1: Composing two braids in B_4

[1] Novikov, P. S. (1955), "On the algorithmic unsolvability of the word problem in group theory", Proceedings of the Steklov Institute of Mathematics (in Russian)

Two connections that can be made to look the same by tightening the strings are considered the *same* braid. Composing two braids consists of drawing them next to one another, gluing the points in the middle, and connecting the strands.

Theorem[2]**:** Every braid, w, in B_n has a unique representation called the left-canonical form,

$$w = \Delta^u A_1 A_2 \cdots A_p, \quad \text{for some } u \in \mathbb{Z}$$

where Δ is a particular braid, the fundamental braid, and the A_i's are braids of a particular form that come from a *finite* subset of B_n. Moreover, if w is given as a word of length ℓ in the generators $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$, then this canonical form is computable in time $O(\ell^2 n \log n)$. In particular, the word problem is easy in B_n.

Diffie-Hellman with Braids[3]

1. Alice and Bob publicly agree on subgroups of B_n, $A = \langle a_1, \ldots, a_k \rangle$ and $B = \langle b_1, \ldots b_m \rangle$.
2. Alice picks a secret word x in $a_1, \ldots, a_k$, $x = x(a_1, \ldots, a_k)$. Bob picks a secret word y in $b_1, \ldots, b_m$, $y = y(b_1, \ldots, b_m)$.
3. Alice sends $b_1^x, \ldots, b_m^x$ to Bob and Bob sends $a_1^y, \ldots, a_k^y$ to Alice.
4. Alice computes $x(a_1^y, \ldots, a_k^y) = x^y = y^{-1}xy$. Bob computes $y(b_1^x, \ldots, b_m^x) = y^x = x^{-1}yx$.
5. Alice multiplies on the left by x^{-1}, obtaining $x^{-1}y^{-1}xy$. Bob multiplies on the left by y^{-1} and inverts, obtaining $(y^{-1}x^{-1}yx)^{-1} = x^{-1}y^{-1}xy$. The shared secret is the commutator $[x, y] = x^{-1}y^{-1}xy$.

Security[4]

An eavesdropper knows $a_1, \ldots, a_k, b_1, \ldots, b_m$ and $a_1^y, \ldots, a_k^y, b_1^x, \ldots, b_m^x$. It might appear that if they can solve the simultaneous conjugacy search problem then they can obtain the shared secret. But that might not be enough.

- If $a_i^{y'} = a_i^y$ for all i, it need not be the case that $y' = y$, just that $y' = c_a y$ for some c_a in the centralizer of A. Similarly, solving the simultaneous conjugacy problem in the b_j^x's gives $x' = c_b x$ for some c_b centralizing B.
- The commutator of x' and y' is

  $$[x', y'] = (x')^{-1}(y')^{-1}x'y' = x^{-1}c_b^{-1}y^{-1}c_a^{-1}c_b x c_a y,$$

  which need not equal $[x, y]$.
- However, if x' is in A and y' is in B, then $c_b = x^{-1}x$ and $c_a = y^{-1}y'$ implies that $c_b \in A$ and $c_a \in B$. Consequently, c_b commutes with y and c_a and c_a commutes with x and c_b, so we have the equality $[x', y'] = [x, y]$.

So evidently, the eavesdropper needs to not only solve the simultaneous conjugacy problem, but they need these conjugating elements to be words in A and B.

---

[2] Ko, K., Lee, S., Cheon, J., Han, J., Kang, J., Park, C.. New Public-Key Cryptosystem Using Braid Groups. CRYPTO 2000, 166-184. Springer Lecture Notes in Computer Science 1880 (2000).

[3] Anshel, I., Anshel, M., Goldfeld, D.

[4] Shpilrain, V. & Ushakov, A. "The Conjugacy Search Problem in Public Key Cryptography: Unnecessary and Insufficient." In: Applicable Algebra in Engineering, Communication and Computing (2006) 17:285