

# Braid Group Cryptography - Liam Hardiman

---

## The Word Problem

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

## The Conjugacy Problem

The **conjugacy search problem** asks, given two conjugate words  $u$  and  $v$  in  $G$ , find 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

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

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  $\sigma_i$  represents the braid formed by crossing strand  $i$  under strand  $i + 1$  and leaving the other strings fixed.

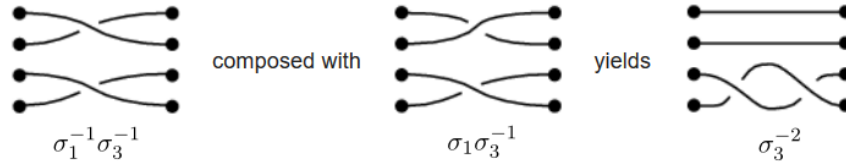


Figure 1: Composing two braids in  $B_4$

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<sup>2</sup>:** 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  $\Delta$  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  $\ell$  in the generators  $\sigma_1, \sigma_2, \dots, \sigma_{n-1}$ , then this canonical form is computable in time  $O(\ell^2 n \log n)$ . In particular, the word problem is easy in  $B_n$ .

---

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

<sup>2</sup>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).

## Diffie-Hellman with Braids<sup>3</sup>

1. Alice and Bob publicly agree on subgroups of  $B_n$ ,  $A = \langle a_1, \dots, a_k \rangle$  and  $B = \langle b_1, \dots, b_m \rangle$ .
2. Alice picks a secret word  $x$  in  $a_1, \dots, a_k$ ,  $x = x(a_1, \dots, a_k)$ . Bob picks a secret word  $y$  in  $b_1, \dots, b_m$ ,  $y = y(b_1, \dots, b_m)$ .
3. Alice sends  $b_1^x, \dots, b_m^x$  to Bob and Bob sends  $a_1^y, \dots, a_k^y$  to Alice.
4. Alice computes  $x(a_1^y, \dots, a_k^y) = x^y = y^{-1}xy$ . Bob computes  $y(b_1^x, \dots, b_m^x) = y^x = x^{-1}yx$ .
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<sup>4</sup>

An eavesdropper knows  $a_1, \dots, a_k, b_1, \dots, b_m$  and  $a_1^y, \dots, a_k^y, b_1^x, \dots, b_m^x$ . It might appear that if they can solve the simultaneous conjugacy search problem (search SCP) 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  $c_b = x'x^{-1}$  and  $c_a = y'y^{-1}$  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 search SCP, but they need these conjugating elements to be words in  $A$  and  $B$ . This combination of problems is called the simultaneous conjugacy separation search problem (SCSSP).

## How Hard Is This to Crack?

It is currently unknown whether computing the centralizers of  $A$  and  $B$  reduces to the search SCP. In either case, as of 2014, there is no known efficient algorithm for computing centralizers of arbitrary subsets of braid groups.<sup>5</sup> Kotov et al. describe an attack on the SCSSP that works experimentally but they don't appear to provide a bound on the complexity.<sup>6</sup>

<sup>3</sup>Anshel, I., Anshel, M., Goldfeld, D. "An algebraic method for public-key cryptography." *Mathematical Research Letters*. 1999. vol. 6. 287-291.

<sup>4</sup>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

<sup>5</sup>Kalka, A., Tsaban, B., Vinokur, G.. "Complete simultaneous conjugacy invariants in Artin's braid groups." arXiv. Submitted 2014, last revised 2018.

<sup>6</sup>Kotov, M., Menshov, A., Myasnikov, D., Panteleev, D., Ushakov, A.. "Conjugacy Separation Problem in Braids: an Attack on the Original Colored Burau Key Agreement Protocol." IACR Cryptology ePrint Archive. 2018.