Braid Group Cryptography

Madsen, Reese
madsenar@clarkson.edu

Pawlaczyk, Tyler
pawlactb@clarkson.edu

Klee, Bryan
kleebm@clarkson.edu

December 18, 2019

1 Braids

In this section we will explain the mathematics behind a braid group. A braid group has braids as the set and concatenation as the group operation written as $< B_n, || >$ where n is the number of strands and

Definition 1.1. $B_n = \{\sigma_1, ..., \sigma_{n-1} : \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{ if } |i-j| = 1 \text{ and } \sigma_i\sigma_j = \sigma_j\sigma_i \text{ if } |i-j| > 1\}$

A braid group is infinite and nonabelian meaning that the elements do not commute such that: $a, b \in B_n : ab \neq ba$. Note that for n strands there are $n-1$ generators represented by σ. A braid is the concatenation of generators. A positive generator, σ_i^+, corresponds to crossing left over right and a negative generator corresponds to crossing right over left, σ_i^-.

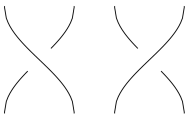

Figure 1: σ_i^+,σ_i^-

For example the braid $b \in B_3 :\ b = \sigma_2^+\sigma_1^+\sigma_2^-\sigma_1^-$ is the following:

2 Braids as Permutations

There is no unique way to write a braid as a concatenation of generators. There is no unique form because you can simply use the rules of the braid group to switch generators around. Therefore there is a connection defined between braid groups and permutation groups so that we can define a uniqueness. There exists a endomorphism: $\phi : B_n \to \Sigma_n$. σ_i represents the switch of the i-th strand with the (i+1) strand. Denote the permutation by $\pi \in \Sigma_n, \pi(i) = b_i$. The trick is to then draw straight lines for the permutations on the braid diagram. For example the permutation $\pi = (1234) \to (4213)$ maps the 1st strand to the 4th strand and so forth. This corresponds to the following braid $A = \sigma_1\sigma_2\sigma_1\sigma_3$. The fundamental braid $\Delta_n = (\sigma_1...\sigma_{n-1})(\sigma_1...\sigma_{n-2})...\sigma_1$ corresponds to the permutation $\Omega_n = n(n-1), ..., (2)(1) = n!$. Now we can define the left canonical form which is used for a unique representation on braids to be utilized in the crypto system.

Definition 2.1. For any $w \in B_n$ $\exists$ a unique representation called the left canonical form.
$w = \Delta^u A_1 A_2 ... A_l, u \in Z', A \in \Sigma_n \backslash \{e, \Delta\}$ where $A_i A_{i+1}$is left weighted for $1 \leq i \leq l-1$.

3 Hard problem associated with braids.

As we have seen in RSA and El-Gamal there are associated hard problems with these crypto systems. For RSA there is the factoring problem where for:

- p,q are 2 distinct k-bit primes.
- n=pq
- Multiplying is computationally easy while the inverse function, factoring, is computationally hard.

Similarly El-Gamal has the discrete logarithm which is the associated computationally hard problem. For braid group cryptography there are several computationally hard problems. One in particular that is widely used is the conjugacy search problem. Conjugacy is defined as:

Definition 3.1. $G -$ group. $a, x, y \in G$. If $y = axa^{-1},$ then y is *conjugate* to x via a.

Conjugacy Search Problem:
Given $x, y \in B_n$ such that $y = axa^{-1}$ for some $a \in B_n$.
Find $b \in B_n$ such that $y = b^{-1}xb$.

There are several types of conjugacy search problems and one we will focus on is the Diffie-Hellman type generalized conjugacy search problem. First we need to define 2 commuting subgroups of B_n.

Definition 3.2. $LB_n, UB_n < B_n$.
$LB_n = \{\sigma_1, ..., \sigma_{[n/2]-1}\}, UB_n = \{\sigma_{[n/2]+1}, ..., \sigma_{n-1}\}$

We know use the fact by definition of a braid group that generators commute if and only if the generators do not share a common strand. Since the $\sigma_{n/2}$ generator is missing, elements in each subgroup will commute with each other such that: $a \in LB_n, b \in UB_n, ab = ba$. We can know look at the Diffie-Hellman generalized conjugacy search problem.

Diffie-Hellman type Generalized Conjugacy Search Problem:
Given $x, y_A, y_B \in B_n$ such that $y_A = axa^{-1}$ and $y_B = bxb^{-1}$ for some $a \in LB_n$ and $b \in UB_n$.
Find $by_Ab^{-1} = ay_Ba^{-1} = abxb^{-1}a^{-1}$

This takes advantage of the fact that we commute a and b which results in the following: $abxb^{-1}a^{-1} = baxa^{-1}b^{-1}$ so both Alice and Bob can decode the message.