Braid Cryptosystem Notes

December 17, 2019

1 Braid Cryptographic System

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

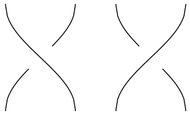
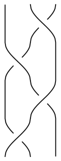


Figure 1: σ_i^+,σ_i^-

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



1.1.1 Left and Right Canonical Forms

For any $w \in B_n$ $\exists$ a unique representation called the left canonical form.

$w = \Delta^u A_1 A_2 ... A_l, u \in Z', A \in \Sigma_n$ without the following elements $\{e, \Delta\}$
where $A_i A_{i+1}$is left weighted for $1 \leq i \leq l-1$
where Σ_n is the set of all permutation braids.

1.2 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 1.2. $G-$group. $a, x, y \in G$. If $y = axa^{-1},$ then y is *conjugate* to x via a.

1.2.1 Sub-Groups of the Braid Group

There are two commuting subgroups of B_n.

$$\begin{aligned} LB_n < B_n &\text{ generated by } \{\sigma_1, ..., \sigma_{[n/2]}\} \\ UB_n < B_n &\text{ generated by } \{\sigma_{n/2+1}, ..., \sigma_{n-1}\} \\ a \in B_n &\text{ commutes w/} b \in UB_n : ab = ba \end{aligned}$$

Notice how σ_3 is missing, we do this in order to be able to commute the upper and lower group. We do this using the second part of the braid definition

1.3 Braid Cryptographic System

Let's define the Braid Cryptographic System.

$$\begin{aligned} n &: \text{the Braid index} \\ l &: \text{the Canonical Index} \end{aligned}$$

1.3.1 Commuter-based Key Agreement

There are many variants of the conjugacy search problem.

1.3.2 Generalized Conjugacy Search

Given: $x, y \in B_n$ s.t. $y = a^{-1}xa$ for some $a \in LB_n$
Find: $b \in LB_n$ s.t. $y = b^{-1}xb$
(note: can replace LB_n w/ UB_n)

Deliverables

11/21/2019

1. Finish Notes (TP) (COMPLETE)
2. Install/Demo CBraid (reference 6 of Anandam) (JL, BK, TP) (COMPLETE)
3. Learn Cryptosystem part (RM) (COMPLETE)