GROUP THEORY IN CRYPTOGRAPHY

Shreshtha Chaturvedi

MPhil in Mathematics

Ambedkar University Delhi

Introduction

- The word "Cryptography" stems from ancient Greek words *kryptós* (English: "hidden"), and *graphein* (English: "to write").
- The field of cryptography has been dominated by Number Theory since years, and the use of Groups in Cryptography is relatively new.
- Many papers have proposed cryptosystems based on group theoretic concepts in the last few years.
- The quest for good candidate groups which can serve cryptography is still on.
- Braid groups appear to be good candidates.

Some Hard Problems in Groups

There are some difficult group-theoretic problems which can be exploited for cryptographic purposes. Some of them are listed below. (In this poster, the publicly known elements are in blue and the private elements are in red.)

- *The factorisation problem* Let H, K be subgroups of a group G and let $w \in G$. Find elements $h \in H$ and $k \in K$ such that $hk = w$.
- *The word problem* Let G be a finitely generated group. Let W and W' be two words in G. Determine if W and W' represent the same element.
- *Conjugacy decision problem* Let G be a group and $g, g' \in G$. Determine whether g and g' are conjugate.
- *Conjugacy search problem* Let G be a group and $g, g' \in G$. If g and g' are known to be conjugate, find $a \in G$ such that $g = ag'a^{-1}$.
- *Generalised conjugacy search problem* Let G be a group, $g, g' \in G$, and $H \leq G$. Find $a \in H$ such that $g = ag'a^{-1}$.
- *The isomorphic decision problem* Let G and G' be two groups with finite presentation in terms of generators and defining relations. Find out if G and G' are isomorphic.

Stickel's Key Exchange

Let G be a non-abelian finite group. Let $x, y \in G$ be such that $xy \neq yx$. Let n_1 and n_2 be orders of x and y respectively. The key between twp parties, say Alice and Bob, can be shared in the following manner.

- Bob picks natural numbers r and s such that $0 < r < n_1$ and $0 < s < n_2$ and sends $x^r y^s$ to Alice.
- Alice picks natural numbers u and v such that $0 < u < n_1$ and $0 < v < n_2$ and sends $x^u y^v$ to Bob.
- Bob computes $x^r(x^u y^v)y^s = x^{r+u}y^{v+s} = K_b$ and Alice computes $K_a = x^u(x^r y^s)y^v = x^{u+r}y^{s+v} = K_b$. Hence, they both share the same key $K_a = K_b$.

Remarks

- If a group G is to be used in a cryptographic protocol based on one-way functions, it must satisfy the following general requirements [1].
  1. G should be well known, or well studied, or both.
  2. There should be an efficiently computable normal form for the elements of G.
  3. By inspection, it should be impossible to compute the elements g_1 and g_2 from the product $g_1 g_2$ where $g_1, g_2 \in G$.
  4. The number of words of length n in G should grow faster than any polynomial in n.
- The interest in infinite non-abelian groups has increased and many of the suggested protocols are in need of such infinite non-abelian groups whose elements have efficient normal forms.
- Efficiency is a huge concern at present, as most of the group theoretic protocols seem to face implementation issues.

The Braid Group B_n

The braid groups are infinite groups that arise naturally from geometric braids. They were explicitly introduced by **Emil Artin**. The braid group on n-strings, denoted by B_n, is defined by the presentation

$$B_n = \left\langle b_1, \ldots, b_{n-1} : \begin{array}{rcll} b_i b_j b_i & = & b_j b_i b_j; & |i-j| = 1 \\ b_i b_j & = & b_j b_i; & |i-j| \geq 2 \end{array} \right\rangle$$

Each element of B_n is called an n - braid. Here, n is said to be the braid index.

Geometrically, a generator $b_i \in B_n$ can be visualised as an n - braid in which the i^{th} string goes under the $(i+1)^{th}$ string to occupy the lower $(i+1)^{th}$ position, while the $(i+1)^{th}$ string occupies the lower i^{th} position. Figure 1 is an example of b_2 in B_4.

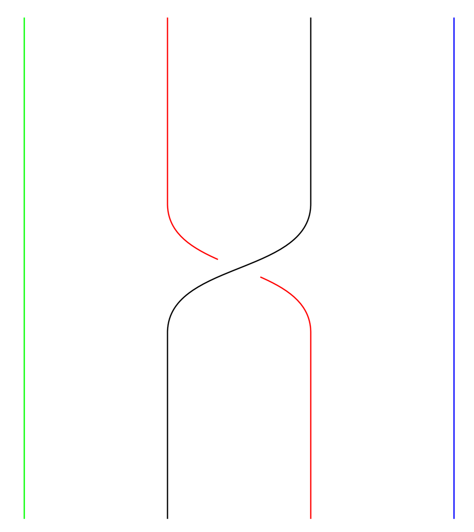

Figure 1: The generator b_2 in B_4

Why braid groups enrich cryptography

- There is an efficiently computable **unique canonical form of a braid** which can be written as an ordered tuple $(m, \sigma_1, \sigma_2, \ldots, \sigma_k)$ where $m \in \mathbb{Z}$, $\sigma_i \in S_n$.
- Braid groups have interesting hard problems which can be exploited for cryptographic purposes. Some of them are the Generalised conjugacy search problem, the Conjugacy search, decision & decomposition problem, the Cycling problem and the Markov problem.

Canonical form of a braid

- Let $\sigma \in S_n$ such that $\sigma(i) = a_i$. Denote σ by $\sigma = a_1 a_2 \cdots a_n$. Define the surjective homomorphism $h : B_n \to S_n$ by
  $$h(a) = \sigma = a_1 a_2 \cdots a_n$$
  where $a \in B_n$ is a braid in which the string at the upper i^{th} position ends at the lower a_i^{th} position.
- We obtain an n-braid, say A_σ, corresponding to σ in which the upper i^{th} string is connected to the lower a_i^{th} string with each crossing positive. Such a braid A_σ is said to be the *permutation braid*. We denote the set of all such braids by S_n^+.
- The permutation braid corresponding to the permutation $\tau_n = n(n-1)\cdots(2)1$ is called the *fundamental braid* and is denoted by Δ_n. Figure 2 is an example of Δ_4.

- Every word B in B_n has a unique left weighted factorisation $B = \Delta^m A_1 A_2 \cdots A_t$ where $A_i \in S_n^+ \setminus \{I, \Delta\}$ for all $i = 1, 2, \ldots, t$, m is an integer and Δ is the fundamental braid. This factorisation is called the **left canonical (or Garside's normal) form** of B, and t said to be its canonical length.
- Let B be a word in B_n with word length k. Then the left canonical form of B can be computed in time $O(k^2 n\log n)$. ([6])
- Let $B = \Delta^m A_1 A_2 \cdots A_t$ be an n braid of canonical length t. The total number of such n braids, that is, of canonical length t is at least $(\lfloor \frac{n-1}{2} \rfloor !)^t$. ([6])

The fundamental braid

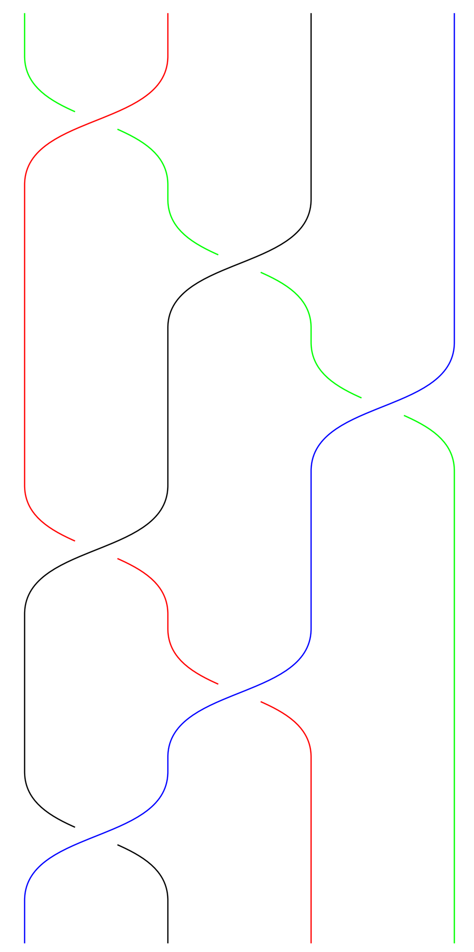

Figure 2: Example of the fundamental braid Δ_4 in B_4

The Ko et al. Protocol

Let $A = LB_l$ and $B = RB_r$ denote the subgroup of B_{l+r} obtained by braiding the left l strands and the right r strands respectively. Thus, $A = < b_1, b_2, \ldots, b_{l-1} >$ and $B = < b_{l+1}, \ldots, b_{l+r-1} >$. It follows from braid relations that every element of A commutes with every element of B.

Alice and Bob can share a key in the following manner.

- An $l + r$ braid, say $x \in B_{l+r}$ is made public.
- Alice selects $a \in LB_l$ and sends $y_1 = axa^{-1}$ to Bob.
- Bob selects $b \in RB_r$ and sends $y_2 = bxb^{-1}$ to Alice.
- Alice computes $K_a = ay_2a^{-1}$, Bob computes $K_b = by_1b^{-1}$. As $ab = ba$ we have $K_a = K_b$

This key exchange protocol depends on the difficulty of solving the **generalised conjugacy search problem** in braid groups.

Bibliography

[1] Alexei Miasnikov, Vladimir Shpilrain and Alexander Ushakov. *Group-based Cryptography*. Birkhäuser Verlag, 2008.

[2] E. Artin. Theory of braids. *Annals of Mathematics*, 48(1):101–126, 1947.

[3] Eberhard Stickel. A new method for exchanging secret keys. *Proceedings - 3rd International Conference on Information Technology and Applications, ICITA 2005*, 2:426–430, 2005.

[4] Elsayed Elrifai and Hugh Morton. Algorithms for positive braids. *Quarterly Journal of Mathematics*, 45(4):479–497, 1994.

[5] F. Garside. The braid group and other groups. *Quarterly Journal of Mathematics*, 20:235–254, 1969.

[6] Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-Sung Kang and Choonsik Park. New public-key cryptosystem using braid groups. *Advances in Cryptology - CRYPTO 2000, Lecture Notes in Computer Science 1880*, (*Springer, Berlin, 2000*), pages 166–183, 2000.