Tutorials:

Braid Group Cryptography

Singapore, June 2007

David Garber

Department of Applied Mathematics, School of Sciences
Holon Institute of Technology
Holon, Israel

Contents

- Basic definitions
- Normal forms
- Cryptosystems based on the braid group
- Attacks on the these cryptosystems
- Future directions

Braid Group B_n

Algebraic Definition

Artin generators: $\sigma_1, \ldots, \sigma_{n-1}$

Relations:

$$\begin{aligned} \sigma_i \sigma_{i+1} \sigma_i &= \sigma_{i+1} \sigma_i \sigma_{i+1}, \\ \sigma_i \sigma_j &= \sigma_j \sigma_i \text{ when } |i-j| > 1 \end{aligned}$$

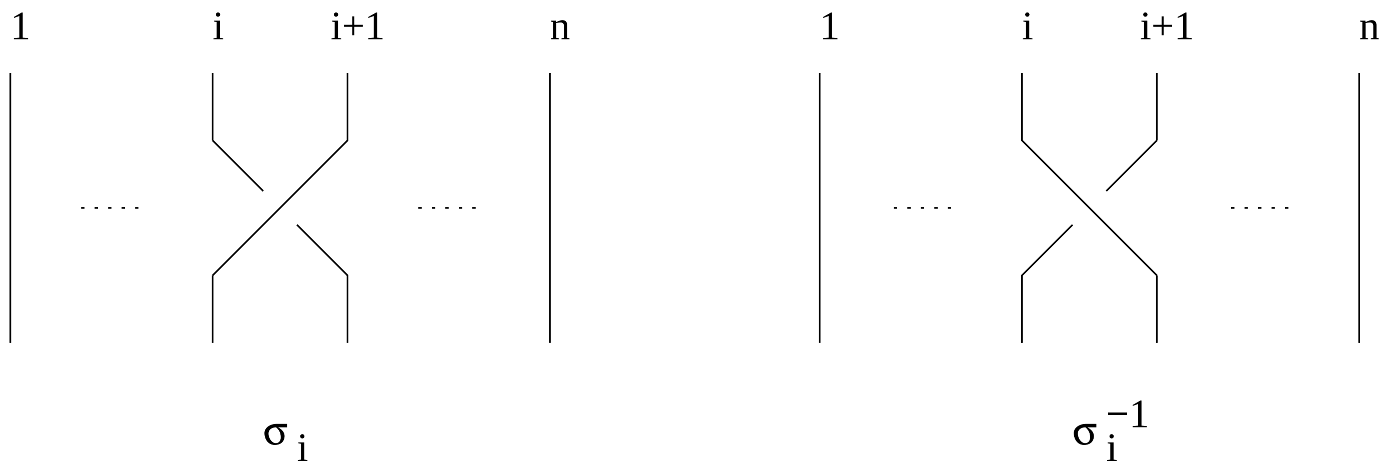
$$B_2 \cong \mathbb{Z}$$

B_n is not commutative for $n \geq 3$.

$$Z(B_n) \cong \mathbb{Z}$$

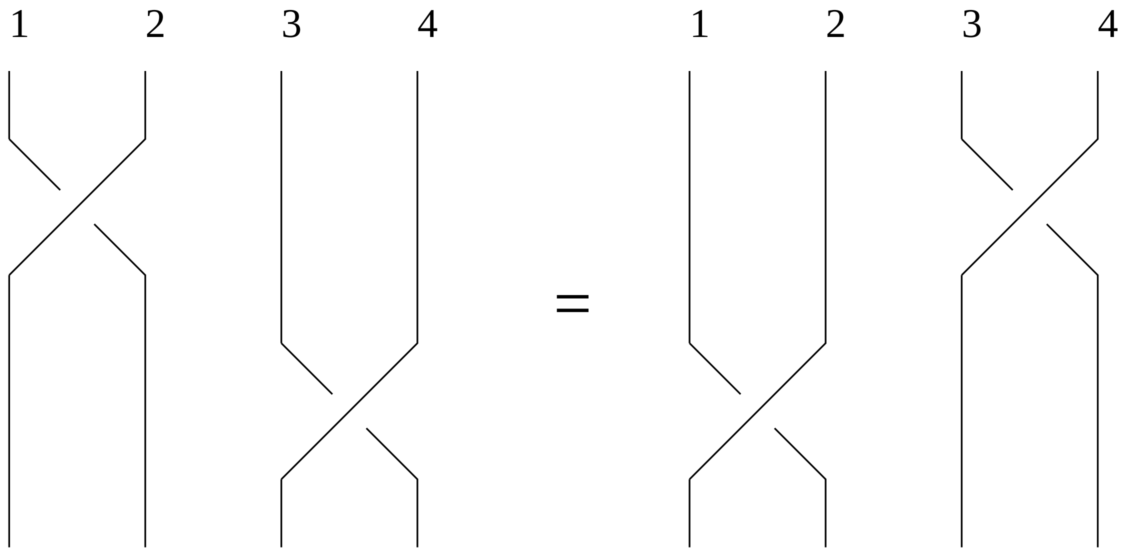
Geometric presentation of the braid group:

The elements of B_n can be interpreted as geometric n strand braids.

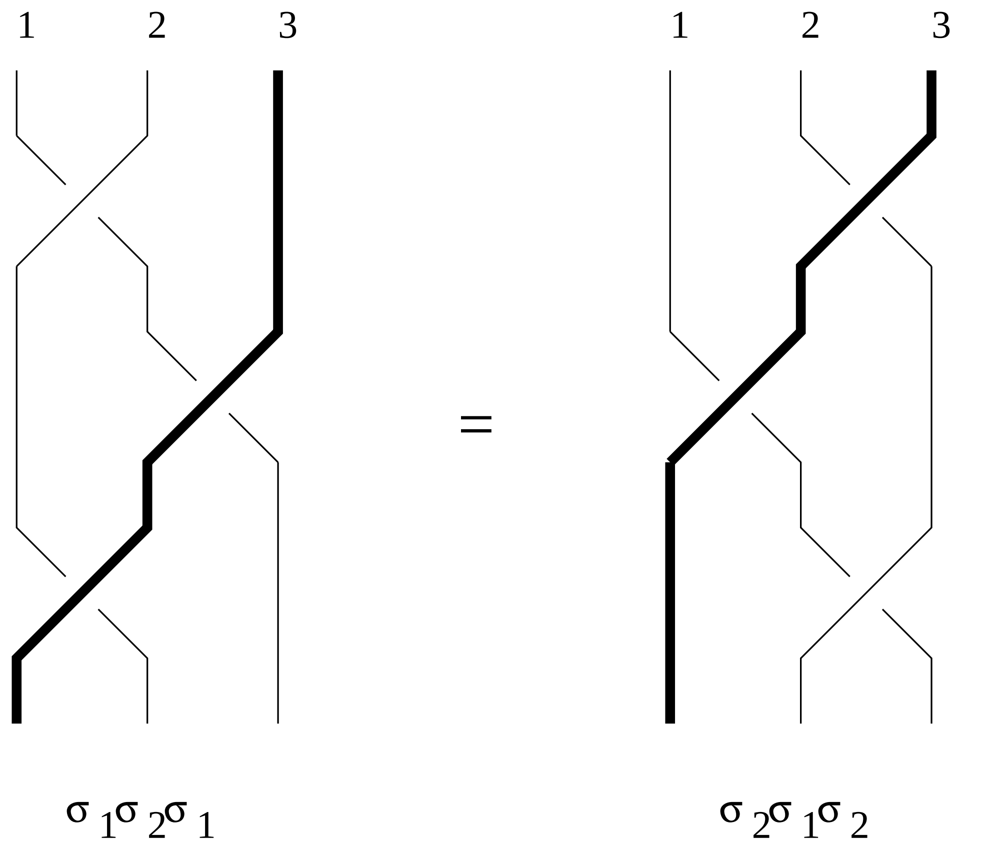


A braid can be seen as induced by a three-dimensional figure consisting on n disjoint curves.

Braid relations in the geometric presentation:

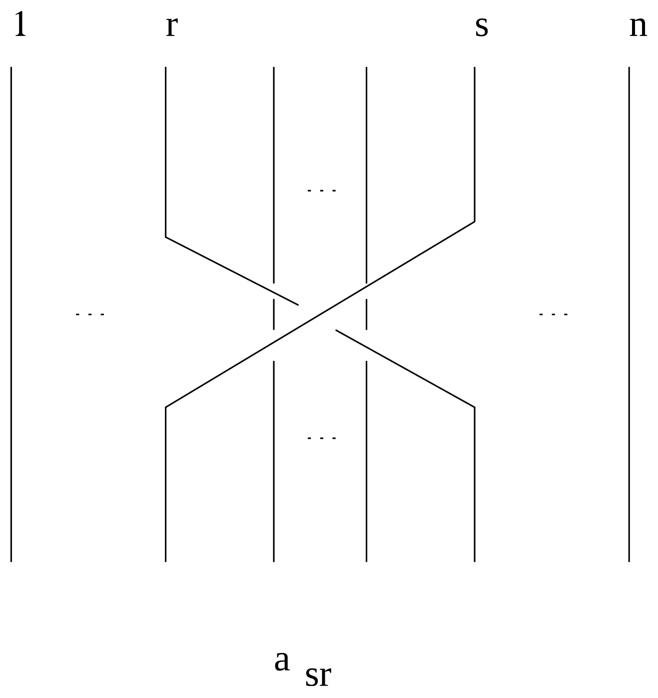


$\sigma_1\sigma_3 \qquad = \qquad \sigma_3\sigma_1$


1
2
3
1
2
3
=
σ 1σ 2σ 1
σ 2σ 1σ 2

Birman-Ko-Lee presentation (1998)

Band generators:

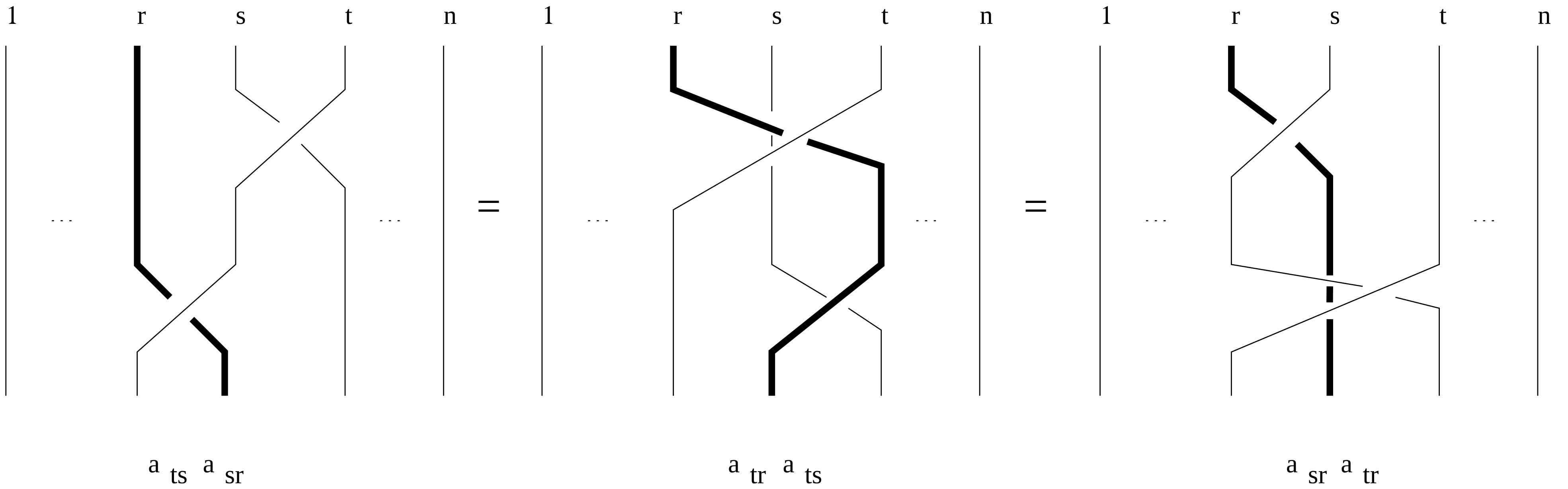


$$\sigma_t = a_{t+1,t}$$

The band generators satisfies the following relations:

- $a_{ts}a_{rq} = a_{rq}a_{ts}$ if $[s,t] \cap [q,r] = \emptyset$ or $[s,t] \subset [q,r]$ or $[q,r] \subset [s,t]$.

- $a_{ts}a_{sr} = a_{tr}a_{ts} = a_{sr}a_{tr}$ for $1 \leq r < s < t \leq n$.



Normal forms of elements in the braid group

Normal form: a unique presentation to each element in the group.

Let ε be the empty word. Having a normal form, solve the word problem:

Word Problem: Given a braid w, does $w \equiv \varepsilon$ hold?

Equivalently:
Problem: Given two braids w, w', does $w \equiv w'$ hold?

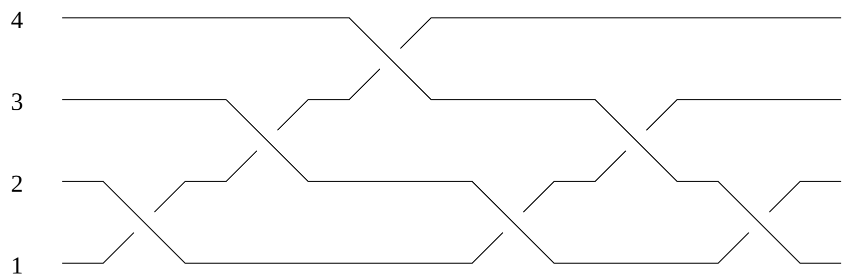
Since: $w \equiv w'$ is equivalent to $w^{-1}w' \equiv \varepsilon$.

Garside normal form

Positive braid: can be written as a product of positive powers.
B_n^+ is the monoid of positive braids.

Fundamental braid $\Delta_n \in B_n^+$:

$$\Delta_n = (\sigma_1 \cdots \sigma_{n-1})(\sigma_1 \cdots \sigma_{n-2}) \cdots \sigma_1$$



$\Delta_4 = \sigma_1 \sigma_2 \sigma_3 \sigma_1 \sigma_2 \sigma_1$

Geometrically, Δ_n is a braid on n strands, where any two strands cross positively **exactly** once.

Properties:

- For any generator σ_i, we can write $\Delta_n = \sigma_i A = B\sigma_i$ for $A, B \in B_n^+$.

- $\sigma_i \Delta_n = \Delta_n \sigma_{n-i}$.

- Δ_n^2 is the generator of the center of B_n.

Partial order on B_n: for $A, B \in B_n$, $A \preceq B$ where $B = AC$ for some $C \in B_n^+$.

Properties:

- $B \in B_n^+ \Leftrightarrow \varepsilon \preceq B$
- $A \preceq B \Leftrightarrow B^{-1} \preceq A^{-1}$.

P is a permutation braid if

$$\varepsilon \preceq P \preceq \Delta_n$$

Geometrically, a permutation braid is a braid on n strands, where any two strands cross positively **at most** once.

Given a permutation braid P:

$$S(P) = \{i | P = \sigma_i P' \text{ for some } P' \in B_n^+\}$$

$$F(P) = \{i | P = P' \sigma_i \text{ for some } P' \in B_n^+\}$$

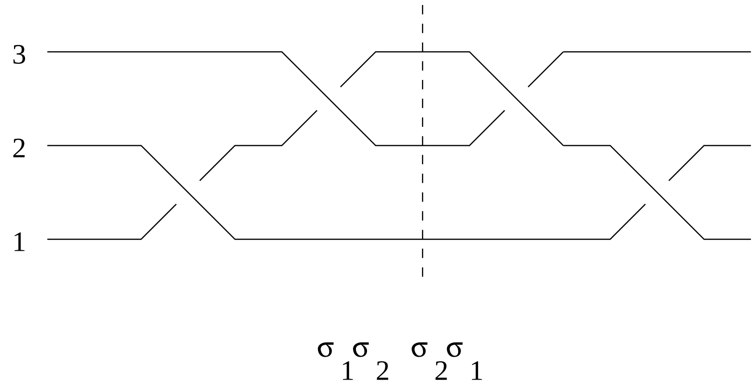
Properties:
1. $i \in S(P)$ if and only if strands i and $i+1$ are exchanged in P.
2. $F(P) = S(\text{rev}(P))$ where rev(P) reverses the order of letters in P.

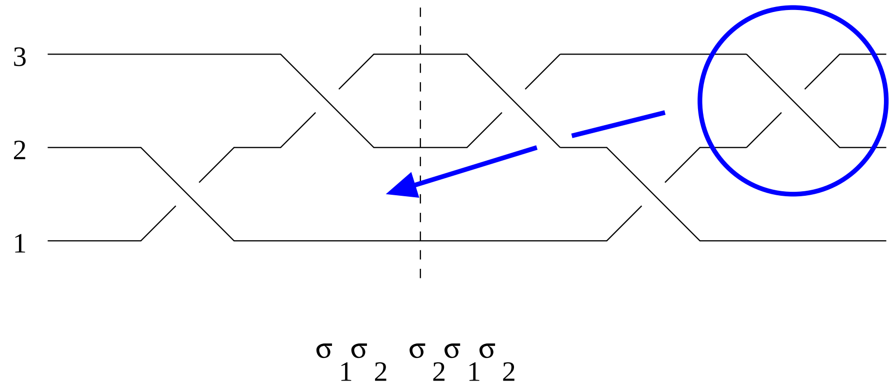
Example: $S(\Delta_n) = F(\Delta_n) = \{1, \ldots, n-1\}$.

Left-weighted decomposition of a positive braid $A \in B_n^+$:

$$A = P_1 P_2 \cdots P_k \text{ where } S(P_{i+1}) \subset F(P_i).$$

Example:





$$\sigma_1\sigma_2 \cdot \sigma_2\sigma_1\sigma_2 = \sigma_1\sigma_2 \cdot \sigma_1\sigma_2\sigma_1 = \sigma_1\sigma_2\sigma_1 \cdot \sigma_2\sigma_1$$

Theorem (Garside): For every braid $w \in B_n$, there is a unique presentation (called **Garside normal form**) given by:

$$w = \Delta_n^r P_1 P_2 \cdots P_k$$

where $r \in \mathbb{Z}$ is maximal, P_i are permutation braids, $P_k \neq \varepsilon$ and $P_1 P_2 \cdots P_k$ is a left-weighted decomposition.

Converting a given braid w into its Garside normal form:
1. Replace σ_i^{-1} by $\Delta_n^{-1} B_i$ where B_i is a permutation braid.
2. Move any appearance of Δ_n to the left. So we get: $w = \Delta_n^{r'} A$ where A is a positive braid.
3. Write A as a left-weighted decomposition of permutation braids, by computing the starting sets and finishing sets.

Complexity: $O(|W|^2 n \log n)$ where $|W|$ is the length of the word in B_n.

Example:

$$w = \sigma_1 \sigma_3^{-1} \sigma_2 \in B_4$$

Since $\Delta_4 = \sigma_3\sigma_2\sigma_1\sigma_3\sigma_2 \cdot \sigma_3$, replace σ_3^{-1} by: $\Delta_4^{-1}\sigma_3\sigma_2\sigma_1\sigma_3\sigma_2$.
So:

$$w = \sigma_1 \cdot \Delta_4^{-1}\sigma_3\sigma_2\sigma_1\sigma_3\sigma_2 \cdot \sigma_2$$

$$w = \Delta_4^{-1} \cdot \sigma_3\sigma_3\sigma_2\sigma_1\sigma_3\sigma_2\sigma_2$$

Left-weighted decomposition:

$$w = \Delta^{-1} \cdot \sigma_2\sigma_1\sigma_3\sigma_2\sigma_1 \cdot \sigma_1\sigma_2$$

Infimum and Supremum:

$$\mathsf{inf}(w) = \mathsf{max}\{r : \Delta^r \preceq w\}$$

$$\mathsf{sup}(w) = \mathsf{min}\{s : w \preceq \Delta^s\}$$

If

$$w = \Delta_n^m P_1 P_2 \cdots P_k$$

then:

$$\mathsf{inf}(w) = m, \mathsf{sup}(w) = m + k$$

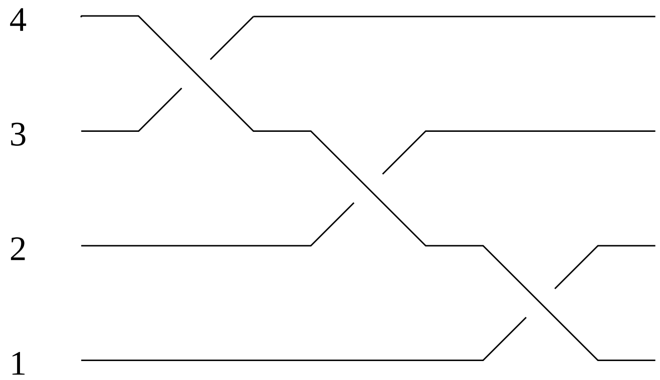
Canonical length of w (or Complexity):

$$\mathsf{len}(w) = \mathsf{sup}(w) - \mathsf{inf}(w) = \#\text{permutation braids}$$

Birman-Ko-Lee's normal form

Fundamental word:

$$\delta_n = a_{n,n-1}a_{n-1,n-2}\cdots a_{2,1} = \sigma_{n-1}\sigma_{n-2}\cdots\sigma_1$$



$\delta_4 = \sigma_3\sigma_2\sigma_1$

Properties: $\delta_n = a_{sr}A = Ba_{sr}$ for A, B positive;

$a_{sr}\delta_n = \delta_n a_{s+1,r+1}$; $\Delta_n^2 = \delta_n^n$

Theorem (Birman-Ko-Lee): $w \in B_n$ has the following unique form:

$$w = \delta_n^j A_1 A_2 \cdots A_k,$$

where $A = A_1 A_2 \cdots A_k$ is positive, j is maximal and k is minimal.

There are $C_n = \frac{(2n)!}{n!(n+1)!}$ (the nth Catalan number) different canonical factors.

Complexity: $O(|W|^2 n)$, where $|W|$ is the length of the word.

More normal forms: Bressaud, Dehornoy, Dynnikov-Wiest.

Public Key Cryptography
(Diffie-Hellman 1976)

Idea: use a one-way function for encryption, which remains one-way only if some information is kept secret.

Purposes for applications of public-key cryptography:

- Confidential message transmission.
- Key exchange.
- Authentication.
- Digital signature.

Diffie-Hellman key-exchange protocol (1976)

Discrete Logarithm Problem: Given α and α^X (mod q), find X.

Protocol:
Public keys: prime q and a primitive element α.
Private keys: Alice: a; Bob: b.

Alice: Sends Bob publicly: $a' = \alpha^a$ (mod q).
Bob: Sends Alice publicly: $b' = \alpha^b$ (mod q)

Shared secret key: $K_{ab} = \alpha^{ab}$ (mod q)

K_{ab} is shared key: Alice computes $K_{ab} = (b')^a$ (mod q).
Bob computes $K_{ab} = (a')^b$ (mod q).

An additional famous Public-Key Cryptosystem: **RSA**.

The underlying (apparently hard) problems

The Conjugacy Problem: Given $u, w \in B_n$, determine whether they are conjugate, i.e., there exists $v \in B_n$ such that

$$w = v^{-1}uv$$

Conjugacy Search Problem: Given conjugate elements $u, w \in B_n$, find $v \in B_n$ such that

$$w = v^{-1}uv$$

Decomposition Problem: $u \notin G \leq B_n$. Find $x, y \in G$ such that $w = xuy$.

Key-agreement protocol
Anshel-Anshel-Goldfeld (1999)

$G = \langle g_1, g_2, \ldots, g_n \rangle \leq B_N$ publicly known.

Secret keys: Alice: $a \in G$. Bob: $b \in G$.

Alice's public key: $ag_1a^{-1}, ag_2a^{-1}, \ldots, ag_na^{-1}$.
Bob's public key: $bg_1b^{-1}, bg_2b^{-1}, \ldots, bg_nb^{-1}$.

Bob knows $b = g_{k_1}^{i_1} g_{k_2}^{i_2} \cdots g_{k_m}^{i_m} \quad \Rightarrow \quad aba^{-1} \quad \Rightarrow \quad K = (aba^{-1})b^{-1}$.

Similarly, Alice knows $bab^{-1} \quad \Rightarrow \quad ba^{-1}b^{-1} \quad \Rightarrow \quad K = a(ba^{-1}b^{-1})$.

Parameters: B_{80} with $m = 20$ and g_i of length 5 or 10 Artin generators.

Diffie-Hellman-type key-exchange protocol
Ko-Lee-Cheon-Han-Kang-Park (2000)

$LB_n = \langle \sigma_1, \ldots, \sigma_{m-1} \rangle; \quad UB_n = \langle \sigma_{m+1}, \ldots, \sigma_{n-1} \rangle$ where $m = \lfloor \frac{n}{2} \rfloor$

Protocol:
Public key: one braid $p \in B_n$.
Private keys: Alice: $s \in LB_n$; Bob: $r \in UB_n$.

Alice: Sends Bob publicly: $p' = sps^{-1}$.
Bob: Sends Alice publicly: $p'' = rpr^{-1}$

Shared secret key: $K = srpr^{-1}s^{-1}$

K shared: Alice: $K = sp''s^{-1} = srpr^{-1}s^{-1}$.
Bob: $K = rp'r^{-1} = rsps^{-1}r^{-1}$.

Parameters: B_{80}, with braids of canonical length 12.

Encryption and decryption
Ko-Lee-Cheon-Han-Kang-Park (2000)

$h : B_n \to \{0,1\}^{\mathbb{N}}$ is a collision-free one-way hash function.

K is a shared secret key.

Bob has a message $m_B \in \{0,1\}^{\mathbb{N}}$:

Bob: sends Alice publicly: $m''_B = m_B \oplus h(K)$.

Alice: computes $m_A = m''_B \oplus h(K)$, and we have $m_A = m_B$, since:

$$m_A = m_B \oplus h(K) \oplus h(K) = m_B.$$