



# *Cryptographic Key Exchange Protocol Based On The Conjugacy Search Problem<sup>a</sup>*

Salah A. Aly

salah@cs.tamu.edu

Department of Computer Science,  
Texas A&M University

02/01/2006

---

<sup>a</sup>Joint work with Dr. Andreas Klappenecker and Pradeep K. Sarvepalli

# Outline

- Introduction
- Key Exchange Protocols Survey
- Hard Problems in Group theory
- Key Exchange Protocols using Decomposition Search Problem
- Braid Groups
- Attacks & Cryptoanalysis
- Conclusions

# Key Exchange Protocols, Overview

Many approaches for KEP:

- The well-known problems in number theory have been used widely to construct cryptosystems, i.e. they depend on hardness of
  - Factoring a large integer.
  - Discrete logarithm problem.
- New hard problems from combinatorial group theory are considered, including
  - Conjugacy search problem.
  - Decomposition search problem.
  - Subgroup membership search problem.
- Other Ideas in quantum cryptography and Johnson Noise and Kirchoff's Law.

# Group Theory

- Definition of a group  $G$ , a set of elements equipped with a binary operation such that
  - For any  $g, h \in G$ ,  $gh \in G$ .
  - A unique identity  $e \in G$ , s.t.  $eg = ge = g \forall g \in G$ .
  - An inverse for every element  $g \in G$ , s.t.  
 $gg^{-1} = gg^{-1} = e$ .
  - Associative, for any  $g, x, y \in G$ ,  $g(xy) = (gx)y$ .
- $G$  is an abelian (commutative) group, if  
 $gh = hg, \forall g, h \in G$ .
- A subset  $H$  is a subgroup of  $G$  if  $H$  is also a group.

# Group Theory (Cont.)

- Let  $g \in G$ , the centralizer of  $g$  in  $G$ ,

$$C_G(g) = \{h \in G \mid gh = hg\}.$$

- We say that  $g$  is a conjugate to  $h$  if there is an element  $a \in G$  s.t.  $g = aha^{-1}$  where  $g, h \in G$ .
- A finitely generated group is a group where there is a set of elements generates the entire group, i.e.  
 $G = \langle g_1, g_2, \dots, g_n \rangle$ .

# Hard Problems in Groups

Some problems in group theory are believed to be hard.

- **Conjugacy Search Problem (CP):**

Instance:  $(x, y) \in G \times G$  such that  $x$  and  $y$  are conjugate.

Goal: Find some  $g \in G$  such that  $y = gxg^{-1}$ .

- **Conjugacy Decomposition Search Problem (DSP):**

Instance:  $(x, y) \in G \times G$  such that  $y = gxg^{-1}$  for some  $g \in H \subseteq G$ .

Goal: Find some  $a, b \in H$ , such that  $y = axb$ .

- **Root Problem (RP):**

Instance:  $(y, p) \in G \times \mathbb{Z}$  such that  $y = x^p$  for some  $x \in G$ .

Goal: Find some  $g \in G$  such that  $y = g^p$ .

These problems are hard only for some groups, ex. braid groups, etc.

- Let a non-commutative group  $G$ , and two subgroups  $A, B \subseteq G$  s.t.  $ab = ba$  for  $a \in A$  and  $b \in B$ , and a public element (value)  $g \in G$ .
- Alice chooses privately  $a \in A$  and sends  $aga^{-1}$  to Bob.
- Bob chooses privately  $b \in B$ , and sends  $bgb^{-1}$  to Alice.
- Alice computes  $K_1 = abgb^{-1}a^{-1}$ .
- Bob computes  $K_2 = barga^{-1}b^{-1}$ .
- Claim the common key is  $K_1 = K_2 = K$ .

Later he proposed an attack to break this protocol in polynomial time for a special representation of braid groups. (Cheon 2003).

## Algebraic Method for PKC (Anshel 1999, 2001) [1], [2]

- Let a non-commutative group  $G$ , and two subgroups  $A, B \subseteq G$  s.t.  $ab = ba$  for  $a \in A$  and  $b \in B$ , and a public element (value)  $g \in G$ .
- Alice chooses privately  $a_1, a_2 \in A$  and send  $a_1ga_2$  to Bob.
- Bob chooses privately  $b_1, b_2 \in B$ , and sends  $b_1gb_2$  to Alice.
- Alice computes  $K_1 = a_1b_1gb_2a_2$ .
- Bob computes  $K_2 = b_1a_1ga_2b_2$ .
- Claim  $K_1 = K_2 = K$ .

- The security of this protocol depends on the search problem in  $G$ .
  - Let  $g \in G$ , and two subgroups  $A, B \subseteq G$ .
  - Find two elements  $a \in A$  and  $b \in B$  such that  $w = agb$ .
  - By looking at the key formalism, the groups  $A$  and  $B$  have to be large so it is hard to compute  $a$  and  $b$ .

Let a public group  $G$  and arbitrary  $g \in G$ ,

1. Alice chooses an element  $a_1 \in G$  of length  $l$ , computes  $C_G(a_1)$ , and publishes  $A = C_G(a_1)$ , or its generators.
2. Bob chooses an element  $b_2 \in G$  of length  $l$ , computes  $C_G(b_2)$ , and publishes  $B = C_G(b_2)$ , or its generators.
3. Alice chooses randomly  $a_2 \in B$  and sends  $S_A = a_1 g a_2$  to Bob.

## The Protocol Scenario (Shpilrain and Ushakov) (Cont.)

4. Bob chooses randomly  $b_1 \in A$  and sends  $S_B = b_1 g b_2$  to Alice.
5. Alice computes her key as  $K_A = a_1 S_B a_2$ .
6. Bob computes his key as  $K_B = b_1 S_A b_2$ .

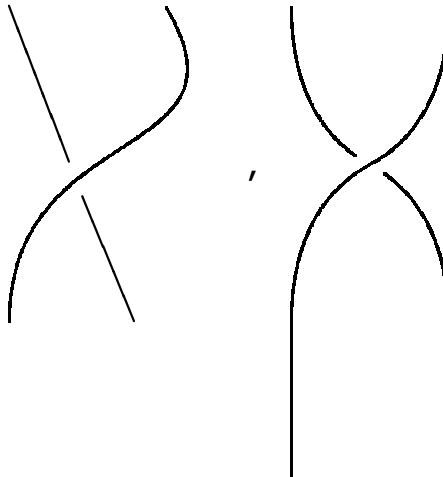
Note that:

- In this protocol the groups A and B do not necessarily need to commute, it is the main modification of the original work by Anshel.
- As a matter of fact the security of the protocol depends on the hardness of the decomposition search problem mentioned earlier.

# Braid Groups

A braid group  $B_n$  can be defined as follows.

- A braid is attained by setting down a number of parallel strands and intertwining them so that they run in the same direction.
- The braid index is the number of  $n$  strands.
- Braid groups have been used in algebraic geometry, operator and knot theory, robotics, public key cryptography, etc.



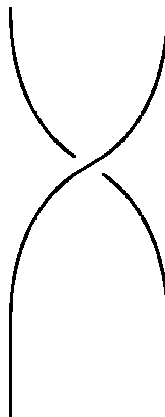
# Braid Groups (Cont.)

The braid group  $B_n$  is defined by the Artin presentation

$$\langle \sigma_1, \dots, \sigma_{n-1} \mid \begin{array}{l} \sigma_i \sigma_j = \sigma_j \sigma_i \text{ for } |i - j| \geq 2, \\ \sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j \text{ for } |i - j| = 1 \end{array} \rangle$$

- identity  $\epsilon \in B_n$
- every element  $\sigma$  has a unique inverse  $\sigma^{-1} \in B_n$
- multiplication of two elements  $\sigma_i \sigma_j \in B_n$

It has been shown that  $S_n$  is a homomorphic image of a braid group  $B_n$ .



# Braid Groups

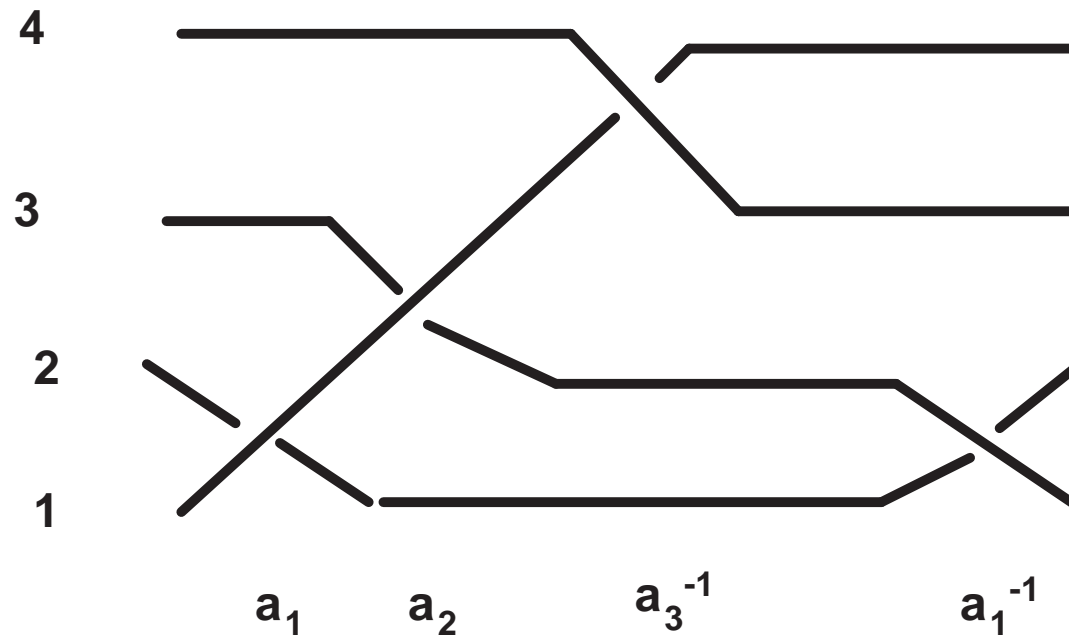


Figure 1: Braid Group

# Braid Groups

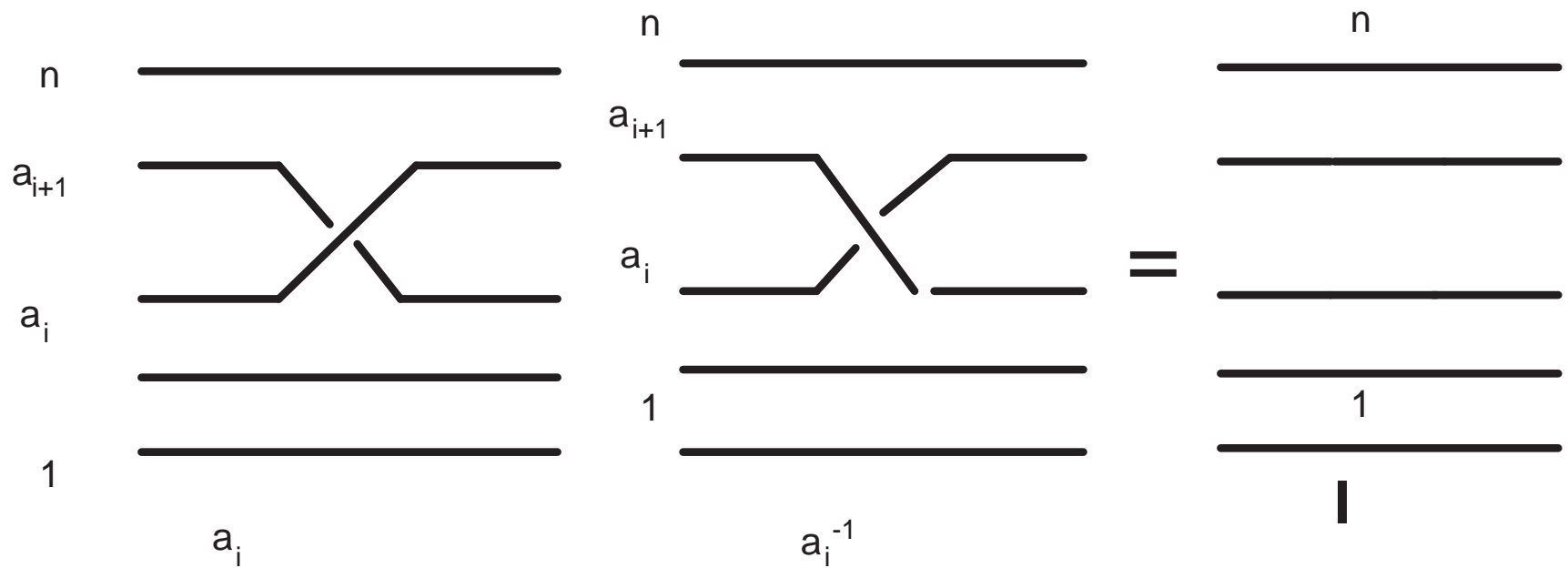
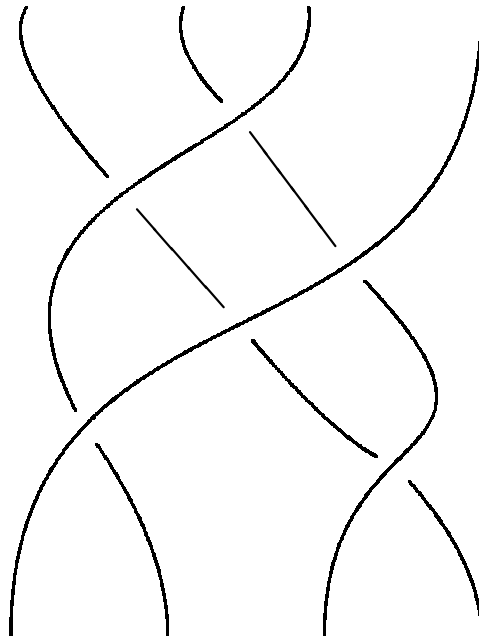


Figure 2: Braid Group

# Braid Groups



## So, Why Using Braid Groups (Shpilrain and Ushakov, 2005)

- Braid groups are infinite non-commutative groups and increase exponentially with the number of strands  $n$ .
- It is efficiently computable in quadratic time algorithms, i.e. multiplication of elements in  $B_n$ .
- It is not easy to compute the centralizer of an element  $g \in B_n$ . Also, computing  $C_G(A)$  is even much harder since this set is huge.
- There is no theoretical solution to the CP, DSP, DHCP, etc.

# Encrypting and Decrypting using Brai

Now, we present Ko, Lee, Cheon, et. (Crypto 2000) [] method to encrypt and decrypt a message using the previous key establishment. Let us choose two subgroups of  $B_n$ , for even  $n$ , as follows.

$$LB_n = \langle \sigma_1, \sigma_2, \dots, \sigma_{n/2-1} \rangle.$$

$$UB_n = \langle \sigma_{n/2+1}, \sigma_{n/2+2}, \dots, \sigma_{n-1} \rangle.$$

Indeed, for  $a \in LB_n$  and  $b \in UB_n$ ,  $ab = ba$ .

## Encrypting and Decrypting using Braids (Cont.)

- Bob chooses a random braid  $b$  in  $UB_n \subseteq B_n$  and he sends the encrypted text  $m_1 = m \oplus h(bS_A b^{-1})$  together with the auxiliary value  $S_B = bgb^{-1}$ , where  $S_A = aga^{-1}$ .
- Alice computes  $m_A = m_1 \oplus h(aS_B a^{-1})$ . Alice retrieves Bob's original message.
- Indeed  $m_A = m$ , because the braids  $b$  and  $a$  commute, we have  $aS_B a^{-1} = abgb^{-1}a^{-1} = barga^{-1}b^{-1} = bS_A b^{-1}$ .

# Breaking the protocol & Attacks

There have been many approaches to break the *DHCP* or *DSP* in braid groups. The main three strategies of the attacks are :

1. Solving the Conjugacy Problem (CP).
2. Finding a probabilistic approach inside  $B_n$ .
3. Using auxiliary linear representation of braids.

## ● Cryptoanalysis of Shpilrain and Ushakov Scheme (2006):

- Attacks on Alice's private elements. Find an element  $a'$  that commutes with every element of  $\langle A \rangle$ , and  $b'$  that commutes with every element of  $\langle B \rangle$ . Such that  $w = a'gb'$ . So, the pair  $(a', b')$  is equivalent to  $(a_1, a_2)$ .
- Similarly attacks on Bob's private elements. Find an element  $a''$  that commutes with every element of  $\langle A \rangle$ , and  $b''$  that commutes with every element of  $\langle B \rangle$ . Such that  $w = a''gb''$ . So, the pair  $(a'', b'')$  is equivalent to  $(b_1, b_2)$ .

# Breaking the protocol & Attacks (Con

- A polynomial time algorithm proposed by Cheon and Jun (Crypto 2003) attacked braid groups. They solved CP in special representation of braids, called Lawrence-Krammer representation, but the CP problem is still hard for Artin braid group representation.
- Supper Summit Set ( $SSS$ ) Attack proposed by Franco and Meneses (J. Algebra 2001) and Ultra Summit Set (USS) attack proposed by Gebhardt (preprint 2003).
  - Let  $g$  be a braid, The  $SSS(g)$  is the set of all possible conjugates of  $b$  with minimal possible complexity. It is believed that this set is finite and computable.
- The attacks based on length: The attacks depend on the length of the key, and use probabilistic heuristic methods (preprint Nov. 2005).

These attacks have not been proved theoretically and they do not seem to work with large braids (see Dehornoy, Contemporary Math. 2004).

# Further Research Problems

Whether or not a combinatorial group theory is applicable to build cryptosystems, the question is how secure they are?.

- Key generation: It is not sufficiently obvious how to choose the keys from arbitrary braid groups, see (Dehornoy 2004). Choosing the braid keys by a smart way prevents the mentioned attacks, since non of them breaks the conjugacy problem in the general case, little is known about braid structures.
- Construct new cryptosystems based on braid groups, i.e. schemes for signature, secret sharing, etc.
- Other braid problems: replace the conjugacy search problem by any other hard problem in braid group, i.e.
  - the root problem.
  - minimal length problem (SSS) has been proved to be hard (NP-complete), see (Dehornoy 2004).

Amazing question: Is there a connection between quantum computing and braid groups?

- It looks that the decomposition search problem in group theory is hard as factoring integers.
- P. Shor (94) proposed an algorithm to factor integers in poly. time.
- The goal is to find a quantum algorithm to solve the conjugacy search problem.

# *Questions !!!*

24-1