An Authenticated Group Key Agreement Protocol on Braid groups

HO-KYU LEE[1], HYANG-SOOK LEE[2], YOUNG-RAN LEE[3]

Department of Mathematics, Ewha Womans University, Seoul, Korea[1,2,3]

e-mail : hokyu@dreamwiz.com[1], *hsl@ewha.ac.kr*[2], *panic@ewha.ac.kr*[3]

Abstract. In this paper, we extend the 2-party key exchange protocol on braid groups to the group key agreement protocol based on the hardness of Ko-Lee problem. We also provide authenticity to the group key agreement protocol.

Keywords : braid groups, conjugacy problem, key agrement protocol, authentication

1. Introduction

In 2000, Ko et al. [13] proposed a new public key cryptosystem on braid groups based on the hardness of the conjugacy problem. The foundation of this system is quite different from widely used cryptosystems on number theory, even if there are some similarities in design. The key exchange scheme on braid groups is based on the hardness of the Ko-Lee problem which is a Diffie-Hellman version of the conjugacy problem. There are many group key agreement protocol using Diffie-Hellman key exchange [2, 3, 7, 14]. The motivation to the common group shared key is caused by the growing importance of secure group communications on open network such as distributed simulation, multi-user games, audio/video conferencing, interactive chat and collaborative applications of all kinds. In this paper, we propose the group shared key protocol on braid groups based on the hardness Ko-Lee problem. We also provide the authenticated group key agreement protocol and show the security properties of the scheme.

This paper is organized as follows.

In Section 2, we give the background of braid groups and computationally hard problems regarding the conjugacy. Based on the Ko-Lee assumption, we introduce the 2-party key agreement protocol. In Section 3, we construct the group key agreement protocol on braid groups based on the hardness Ko-Lee problem and improve the protocol by authentication. We also prove that the authenticated protocol is contributory, perfect forward secret, resistant to known key attacks.

2. Preliminaries

In this section, we give the basic definitions of braid groups and discuss some hard problems on those groups. For more information of braid groups, word problem and conjugacy problem, refer to the papers [1, 4, 5, 8, 9, 10].

For each integer $n \geq 2$, the *n-braid group* B_n is the group generated by $\sigma_1, \sigma_2, \cdots, \sigma_{n-1}$ with the relations

(i) $\sigma_i \sigma_j = \sigma_j \sigma_i$ where $\mid i - j \mid \geq 2$,

(ii) $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$.

The integer n is called the *braid index* and each element of B_n is called an $n-braid$. An n-braid has the following geometric interpretation: It is a set of disjoint n-strands all of which are attached to two horizontal bars at the top and at the bottom such that each strands always heads downward as one walks along the strand from the top to the bottom. In this geometric interpretation, each generator σ_i represents the process of swapping the i-th strand with the next one (with i-th strand going under the $(i+1)$-th one). Two braids are equivalent if one can be deformed to the other continuously in the set of braids. B_n is the set of all equivalence classes of geometric n-braids with a natural group structure. The multiplication ab of two braids a and b is the braid obtained by positioning a on the top of b. The identity e is the braid consisting of n straight vertical strands and the inverse of a is the reflection of a with respect to a horizontal line. So σ^{-1} can be obtained from σ by switching the over-strand and under-strand.

We describe some mathematically hard problems in braid groups. We say that x and y are conjugate if there is an element a such that $y = axa^{-1}$. For $m < n, B_m$ can be considered as a subgroup of B_n generated by $\sigma_1, \sigma_2, \cdots, \sigma_{m-1}$.

1. CONJUGACY DECISION PROBLEM (CDP)
Instance : $(x, y) \in B_n \times B_n$ such that $y = axa^{-1}$ for some $a \in B_n$.
Objective : Determine whether x and y are conjugate or not.

2. CONJUGACY SEARCH PROBLEM (CSP)
Instance : $(x, y) \in B_n \times B_n$ such that $y = axa^{-1}$ for some $a \in B_n$.
Objective : Find $b \in B_n$ such that $y = bxb^{-1}$.

3. GENERALIZED CONJUGACY SEARCH PROBLEM (GCSP)
Instance : $(x, y) \in B_n \times B_n$ such that $y = axa^{-1}$ for some $a \in B_m, m \leq n$.
Objective : Find $b \in B_m$ such that $y = bxb^{-1}$.

4. CONJUGACY DECOMPOSITION PROBLEM (CDP)
Instance : $(x, y) \in B_n \times B_n$ such that $y = axa^{-1}$ for some $a \in B_m, m < n$.
Objective : Find $b_1, b_2 \in B_m$ such that $y = b_1 x b_2$.

The public key system on braid groups in [13] is based on the generalized conjugacy search problem. We consider two subgroups LB_l and RB_r of B_{l+r} for some appropriate pair of integers (l, r). LB_l(resp.RB_r) is the subgroup of B_{l+r} consisting of braids made by braiding left l (resp. right r)-strands among $l + r$ strands. LB_l is generated by $\sigma_1, \cdots, \sigma_{l-1}$ and RB_r is generated by $\sigma_{l+1}, \cdots, \sigma_{l+r-1}$. For any $a \in LB_l$ and $b \in RB_r$, $ab = ba$. We choose a sufficiently complicated $(l + r)$-braid $\alpha \in B_{l+r}$. Then the following is the one-way function.

$$f : LB_l \times B_{l+r} \to B_{l+r} \times B_{l+r}, \quad f(a, x) = (axa^{-1}, x).$$

For a given a pair (a, x), it is easy to compute axa^{-1} but the all the known attacks need exponential time to compute a from (axa^{-1}, x). This one way function is based on the difficulty of the generalized conjugacy search problem. The key agreement scheme is based on the following Ko-Lee problem. The Ko-Lee problem is the Diffie-Hellman type of a generalized conjugacy search problem.

5. <u>KO-LEE PROBLEM</u> (KLP)
Instance : The triple (x, y_1, y_2) of elements in B_{l+r} such that $y_1 = axa^{-1}$ and $y_2 = bxb^{-1}$ for some hidden $a \in LB_l$ and $b \in RB_r$.

Objective : Find $by_1b^{-1}(= ay_2a^{-1} = abxa^{-1}b^{-1})$. Here, $ab = ba$ for any $a \in LB_l$ and $b \in RB_r$.

We say that the *computational Ko-Lee assumption* if no efficient algorithm can compute the shared key $abxb^{-1}a^{-1}$. We also say that the *decisional Ko-Lee assumption* if it is hard to distinguish the shared key $abxb^{-1}a^{-1}$ from a random conjugate of x of the form wxw^{-1}. The GCSP and the computational Ko-Lee problem have no polynomial-time solving algorithm yet. However, it turns out to be the decisional Ko-Lee assumption is false [12]. Now we introduce the 2-party key agreement protocol on braid groups [13].

Key Agreement Protocol :

(i) *Preparation step* : Suppose A and B want to share a common secret key. An appropriate pair of integers (l, r) and a sufficiently complicated $(l + r)$-braid $\alpha \in B_{l+r}$ is selected and published.

(ii) *Key agreement scheme* :

(a) A chooses a random secret braid $r_1 \in LB_l$ and sends $y_1 = r_1\alpha{r_1}^{-1}$ to B.

(b) B chooses a random secret braid $r_2 \in RB_r$ and sends $y_2 = r_2\alpha{r_2}^{-1}$ to A.

(c) A receives y_2 and computes the shared key $k = r_1y_2{r_1}^{-1}$.

(d) B receives y_1 and computes the shared key $k = r_2y_1{r_2}^{-1}$.

Since $r_1 \in LB_l$ and $r_2 \in RB_r,\ r_1r_2 = r_2r_1$. This implies $k = r_1y_2{r_1}^{-1} = r_2y_1{r_2}^{-1}$. Therefore A and B obtain the common secret k.

The security of this protocol is based on the hardness of Ko-Lee problem. The shared secret key k must be derived by applying a suitable key derivation function to the quantity $r_1r_2\alpha r_2^{-1}r_1^{-1}$. For otherwise, an attacker might be able to get partial information about common secret keys even if KLP is hard.

3. Authenticated group key agreement on braid groups

Our interest is to design the authenticated key agreement protocol on braid groups. This protocol requires the following desirable properties.

- Perfect Forward Secrecy (PFS)
- Resistance to Known-Key Attacks
- Key Authentication
- Key Confirmation and Key Integrity

All of these are necessary to achieve resistance to active adversaries where an adversary additionally subverts the communications by injecting, deleting, altering or replaying messages. We give some definitions and terminology regarding authenticated key agreement protocol.

A *key agreement protocol* is a key establishment technique whereby a shared secret key is derived by two(or more) parties as a function of information contributed, or associated with, each of these, such that no party can predetermine the resulting value. A key agreement protocol is *contributory* if each party equally contributes to the key and guarantees its freshness. Let A and B be two honest parties i.e. legitimate who execute the steps of a protocol correctly. A key agreement protocol is said to provide *implicit key authentication*(of B to A) if the party A is assured that no other party aside from a specially identified second party B can possibly learn the value of a particular secret key. A protocol provides *key confirmation* if a party is assured that its peer(or a group thereof) actually has possession or a particular secret key. A contributory key agreement protocol provides *key integrity* if a party is assured that its particular secret key is a function of only the individual contributions of all protocol parties. In particular, extraneous contribution(s) to the group key cannot be tolerated even if it does not afford the attacker(s) with any additional knowledge. A key agreement protocol which provides implicit key authentication to both participating parties is called an *authenticated key agreement protocol* (A-KA). A protocol is said to have *perfect forward secrecy* if compromise of long-term keys does not compromise past session keys. A protocol is said to be vulnerable to *known-key attack* if compromise of past session keys allows either a passive adversary to compromise future session keys, or an active adversary to impersonate one of the protocol parties.

3.1. Group key agreement protocol

In this subsection we construct the group key agreement(GKA) protocol on braid groups by extending the 2-party key agreement.

The following notation is used in this section.

n : number of group members
i, j : index of group members

M_i : i-the group member
B_l : l-th braid group
α : sufficiently complicated l-braid
x_i : long-term secret key of M_i in B_{l_i}
r_i : random secret key of M_i in B_{l_i}
$k_{i,j}$: long-term common secret key shared by M_i and M_j for $i \neq j$
S_n : group key shared by all n-members
$S_n(M_i)$: M_i's view on a group key

We consider n subgroups $B_{l_1}, B_{l_2}, \cdots, B_{l_n}$ of l-braid group B_l where $l = l_1 + l_2 + \cdots + l_n$ for some appropriate integers $l_1, l_2, \cdots, l_n$. Each B_{l_i} is the subgroup of B_l consisting of braids made by braiding l_i-strands from the left among l-strands with the order $l_1, l_2, \cdots, l_n$. Thus each B_{l_i} is generated by

$$< \sigma_{\sum_{j=0}^{i-1} l_j+1},\ \sigma_{\sum_{j=0}^{i-1} l_j+2},\ \cdots,\ \sigma_{\sum_{j=0}^{i} l_j-2},\ \sigma_{\sum_{j=0}^{i} l_j-1} >$$

where $i = 1, 2, \cdots, n$ and $l_0 = 0$ by convention. For any $r_m \in B_{l_m}$ and $r_n \in B_{l_n}$ with $m \neq n$, $r_m r_n = r_n r_m$. Let $\alpha \in B_l$ be a sufficiently complicated l-braid. We suppose $\{M_i | i = 1, \cdots, n\}$ is the set of members wishing to share a key. We construct a shared group key by performing the following steps.

GKA Protocol on Braid groups

Round i, $(i = 1, 2, \cdots, n-1)$
(i) M_i selects a random $r_i \in B_{l_i}$.
(ii) $M_i \longrightarrow M_{i+1}$: $\{r_i \cdots \hat{r}_j \cdots r_1 \alpha r_1^{-1} \cdots \hat{r}_j^{-1} \cdots r_i^{-1} \mid j = 1, 2, \cdots, i\}$ and $r_i r_{i-1} \cdots r_1 \alpha r_1^{-1} \cdots r_{i-1}^{-1} r_i^{-1}$, where $\hat{r}_j$ means that r_j does not appear.
Round n
(i) M_n selects a random $r_n \in B_{l_n}$.
(ii) M_n computes $r_n \cdots \hat{r}_i \cdots r_1 \alpha r_1^{-1} \cdots \hat{r}_i^{-1} \cdots r_n^{-1}$ for each $i = 1, \cdots, n-1$.
$M_n \longrightarrow M_i$ for all $i = 1, \cdots, n-1$: $r_n \cdots \hat{r}_i \cdots r_1 \alpha r_1^{-1} \cdots \hat{r}_i^{-1} \cdots r_n^{-1}$.

Then each participant M_i obtains the shared key by computing

$$\begin{aligned} S_n(M_i) &= r_i(r_n \cdots \hat{r}_i \cdots r_1 \alpha r_1^{-1} \cdots \hat{r}_i^{-1} \cdots r_n^{-1}) r_i^{-1} \\ &= r_n \cdots r_{i+1} r_i r_{i-1} \cdots r_1 \alpha r_1^{-1} \cdots r_{i-1}^{-1} r_i^{-1} r_{i+1}^{-1} \cdots r_n^{-1}. \end{aligned}$$

M_n also computes the shared key

$$S_n(M_n) = r_n(r_{n-1} \cdots r_1 \alpha r_1^{-1} \cdots r_{n-1}^{-1}) r_n^{-1}. \quad \square$$

Our protocols are based on distributively computing a subset of $\{S\alpha S^{-1} | S \subset \{r_1, \cdots r_n\}\}$. From $r_n \cdots \ \hat{r}_i \ \cdots \ r_1 \alpha r_1^{-1} \cdots \hat{r}_i^{-1} \cdots r_n^{-1}$, each member M_i can easily computes the shared key $S_n = r_n \cdots r_1 \alpha r_1^{-1} \cdots r_n^{-1}$.

3.2. Authenticated group key agreement protocol

In this subsection, we construct the authenticated group key agreement(A-GKA) protocol on braid groups.

A-GKA Protocol on Braid groups

Initialization : Let α be a sufficiently complicated l-braid in B_l and $M_1, \cdots, M_n$ be n participants wishing to share a key. Each M_i chooses a secret $x_i \in B_{l_i}$ and computes $x_i \alpha x_i^{-1}$. Let $\{(x_i, x_i \alpha x_i^{-1}) | i = 1, \cdots, n\}$ be the set of long-term secret and public keys of $M_i's$. Thus $(l_1, \cdots, l_n, \alpha, x_1 \alpha x_1^{-1}, \cdots, x_n \alpha x_n^{-1})$ are the public values of the system.

Round i, $(i = 1, 2, \cdots, n-1)$

(i) M_i selects a random $r_i \in B_{l_i}$.

(ii) $M_i \longrightarrow M_{i+1} : \{r_i \cdots \hat{r}_j \cdots r_1 \alpha r_1^{-1} \cdots \hat{r}_j^{-1} \cdots r_i^{-1} \mid j = 1, 2, \cdots, i\}$ and $r_i r_{i-1} \cdots r_1 \alpha r_1^{-1} \cdots r_{i-1}^{-1} r_i^{-1}$.

Round n

(i) M_n selects a random $r_n \in B_{l_n}$ and M_n computes $k_{in} = x_n x_i \alpha x_i^{-1} x_n^{-1}$ for each $i = 1, \cdots, n-1$.

(ii) $M_n \longrightarrow M_i$ for all $i = 1, \cdots, n-1 : \sigma_i = k_{in} r_n \cdots \hat{r}_i \cdots r_1 \alpha r_1^{-1} \cdots \hat{r}_i^{-1} \cdots r_n^{-1} k_{in}^{-1}$.

When each M_i receives σ_i, compute k_{in} and $S_n(M_i) = r_i k_{in}^{-1} \sigma_i k_{in} r_i^{-1}$. Therefore the shared key for all M_i is

$$\begin{aligned} S_n(M_i) &= r_i k_{in}^{-1} \sigma_i k_{in} r_i^{-1} \\ &= r_i r_n \cdots \hat{r}_i \cdots r_1 \alpha r_1^{-1} \cdots \hat{r}_i^{-1} \cdots r_n^{-1} r_i^{-1} \\ &= r_n \cdots r_i \cdots r_1 \alpha r_1^{-1} \cdots r_i^{-1} \cdots r_n^{-1}. \end{aligned}$$

Also M_n computes the shared key

$$S_n(M_n) = r_n(r_{n-1} \cdots r_1 \alpha r_1^{-1} \cdots r_{n-1}^{-1}) r_n^{-1}. \quad \square$$

THEOREM 3.1. *A-GKA is a contributory authenticated key agreement protocol.*

Proof. From the construction of the above protocol, it is evident that the protocol is contributory. Let C be an active adversary who can modify, delay or inject messages. The goal of the adversary is to share a key with either M_i for $i \in \{1, \ldots, n-1\}$ or M_n by masquerading as some M_i.

Attack on M_n : Let $S_n(M_n)$ be the key computed by M_n and $S_n(M_n) = r_n c_n \alpha c_n^{-1} r_n^{-1}$ where c_n is possibly known to C and $c_n r_n = r_n c_n$. Computing $r_n c_n \alpha c_n^{-1} r_n^{-1}$ requires C to compute $r_n \alpha r_n^{-1}$. But the only expression containing $r_n \alpha r_n^{-1}$ is $\sigma_i = k_{in}(\frac{c_n}{r_i}) r_n \alpha r_n^{-1} (\frac{c_n}{r_i}) k_{in}^{-1}$. Hence it is intractable to compute $r_n \alpha r_n^{-1}$ without the knowledge of k_{in} for any $i = 1, \cdots, n-1$.

Attack on M_i for some i : Let $S_n(M_i)$ be the key computed by M_i and $S_n(M_i) = r_i k_{in}^{-1} c_i \alpha c_i^{-1} k_{in} r_i^{-1}$ where c_i is possibly known to C. First, suppose $c_i = k_{in}\bar{c}_i$ where $\bar{c}_i$ is polynomially independent of k_{in} and known to C. Then

$$S_n(M_i) = r_i k_{in}^{-1} (k_{in} \bar{c}_i \alpha \bar{c}_i^{\,-1} k_{in}^{-1}) k_{in} r_i^{-1} = r_i \bar{c}_i \alpha \bar{c}_i^{\,-1} r_i^{-1}.$$

However computing $k_{in} \bar{c}_i \alpha \bar{c}_i^{\,-1} k_{in}^{-1}$ is intractable without the knowledge of k_{in}. Therefore it is difficult to compute $S_n(M_i)$. Next, we assume c_i is polynomially independent of k_{in}. Then $r_i k_{in}^{-1} c_i \alpha c_i^{-1} k_{in} r_i^{-1}$ is still a function of k_{in}^{-1} and k_{in}, hence computing $S_n(M_i)$ is intractable by C. $\square$

THEOREM 3.2. *A-GKA protocol provides perfect forward security.*

Proof. Suppose that all long term keys $\{k_{in} \mid i = 1, \ldots, n-1\}$ are compromised. Then the adversary is able to compute a subset of $\{S\alpha S^{-1} \mid S \subset \{r_1, r_2, \cdots, r_n\}\}$ where $S\alpha S^{-1}$ means $r_{i_k} \cdots r_{i_1} \alpha r_{i_1}^{-1} \cdots r_{i_k}^{-1}$ for $S = \{r_{i_1}, \cdots, r_{i_k}\}$. However, by the direct extension of 2-party key exchange scheme, it is intractable to find the group key for the given set $\{S\alpha S^{-1} \mid S \subset \{r_1, r_2, \cdots, r_n\}\}$. $\square$

THEOREM 3.3. *A-GKA is resistant to the known key attacks.*

Proof. The protocol A-GKA is resistant to passive known-key attacks since the session keys do not contain any information of long-term keys. Let $S_n(M_i)$ be the session key computed by each M_i, $S_n(M_i) = r_i k_{in} c_i \alpha c_i^{-1} k_{in}^{-1} r_i^{-1}$ for $i = 1, \cdots, n-1$ and $S_n(M_n) = r_n c_n \alpha c_n^{-1} r_n^{-1}$ where each c_i is a quantity possibly known to the adversary C. C also knows a subset of $\{S\alpha S^{-1} | S \subset \{r_1, \cdots, r_n\}\}$. Using these information, it is difficult to find $k_{in} \alpha k_{in}^{-1}$ or $k_{in}^{-1} \alpha k_{in}$. Therefore it is resistant to the active known-key attacks. $\square$

References

1. E. Artin, *Theory of braids*, Annals of Math. 48 (1947), 101-126.

2. G. Atenies, M. Steiner, G. Tsudik. Authenticated group key agreement and friends, ACM Conference on Computer and Communications Security, 1998.

3. C. Becker and U. Willie, communication complexity of group key distribution, ACM conference on Computer and Communication Society, 1998.

4. J. S. Birman, *Braids, links and mapping class groups*, Annals of Math. Study, no. 82, Princeton University Press(1974).

5. J. S. Birman, K. H. Ko and S. J. Lee, *A new approach to the word and conjugacy problems in the braid groups*, Advances in Math. 139 (1998), 322-353.

6. D. Boneh and A. Silverberg, Applications of Multi-linear forms to Cryptography, http://eprint.iacr.org,2002.

7. M. Burmester and Y. Desmedt. A Secure and Efficient Conference key Distribution System, Advances in Cryptology-Eurocrypto'94, LNCS, Springer Verlag, 275-286, 1995.

8. W. Diffie and M. Hellman. New direction In Cryptography, IEEE Transactions on Information Theory, IT-22(6):644-654, 1976.

9. E. A. Elrifai and H. R. Morton, *Algotithms for positive braids*, Quart. J. Math. Oxford 45 (1994), no. 2, 479-497.

10. D. Epstein, J. Cannon, D. Holt, S. Levy, M. Pasterson and W. Thurston, *Word processing in groups*, Jones & Bartlett, 1992.

11. F. A. Garside, *The braid group and other groups*, Quart. J. Math. Oxford 20 (1969), no. 78, 235-254.

12. R. Gennaro and D. Micciancio, *cryptanalysis of a pseudorandom generator based on braid groups*, Euro Crypto'2002, LNCS , pp 1-13, Springer 2002.

13. K. Ko, S. Lee, J. Cheon, J. Han, J. kang C. Park. *New public key cryptosystem using braid groups*, Crypto'2000, LNCS 1880, pp.166-183, Springer 2000.

14. M. Stein, G. Tsudik, M. Waidner. Diffie Hellman Key Distribution Extended to Group Communication, ACM conference on computer and communication security, 1996.