Tripartite Key Agreement Protocol using Conjugacy Problem in Braid Groups

Atul Chaturvedi
Department of Mathematics
Pranveer Singh Institute of Technology

Varun Shukla
Department of Electronics & Communication
Pranveer Singh Institute of Technology

ABSTRACT

Braid groups were first introduced by Emil Artin in 1925. First cryptosystem, using Braid groups as a platform was discovered by Anshel et al in 2001. After the publication of this paper several cryptosystems on Braid groups had been designed. In this paper we have proposed a tripartite authenticated key agreement protocol using conjugacy problem which works in a braid group. We have proved that our protocol meet the security attributes under the assumption that the Braid Decomposition Problem (BDP) and the Conjugacy Search Problem (CSP) are hard in braid group.

Keywords

Braid group, Braid Decomposition Problem, Conjugacy Search Problem, authentication, tripartite key agreement,

1. INTRODUCTION

Recent years in cryptological research have witnessed several proposals for secure cryptographic schemes using noncommutative groups; in particular Artin's braid groups [1, 2, 5, 7, 9, 10, 11, 12, 13, 14, 15, 18, 19, 20, and 21]. The idea of applying braid group as a platform for cryptosystems was introduced by Anshel *et al* [2]. Braid groups, are more complicated than Abelian groups and, on the other hand, are not too complicated to work with. These two characteristics make braid group a convenient and useful choice to attract the attention of researchers.

We make use of Braid Decomposition Problem (BDP) and Conjugacy Search Problem (*CSP*) to suggest a new tripartite authenticated key agreement scheme. The *BDP* and *CSP* in braid groups are algorithmically difficult and consequently provide one-way functions. We use this characteristic of *BDP* and *CSP* to propose a tripartite authenticated key agreement protocol using braid groups which meets security attributes.

The rest of the paper is organized as follows: We present a brief introduction of braid groups in section 2. In section 3, we define authenticated key agreement protocol. In section 4, we present our protocol, and we give a proof of security for our scheme. The paper ends with conclusion.

2. BRAID GROUPS

Emil Artin [3] in 1925 defined B_n, the braid group of index *n*, using following generators and relations: Consider the *generators* $\sigma_1, \sigma_2, ..., \sigma_{n-1}$, where σ i represents the braid in which the $(i+1)^{st}$ string crosses over the i^{th} string while all other strings remain uncrossed. The *definining relations* are

1. $\sigma_i\sigma_j = \sigma_j\sigma_i \; for |i-j| > 1$,
2. $\sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \; for |i-j| = 1$.

The reader may consult any textbook on braids for a geometrical interpretation of elements of the group B_n by an *n*-strand braid in the usual sense [4]. The braid $\Delta = (\sigma_1\sigma_2 .... \sigma_{n-1})(\sigma_1\sigma_2 .... \sigma_{n-2}) ....(\sigma_1\sigma_2)(\sigma_1)$ is called the *fundamental braid*. Δ nearly commutes with any braid *b*. In fact $\Delta b = \tau(b)\Delta$, where $\tau : B_n \to B_n : \tau(\sigma_i) = \sigma_{n-i}$ is an automorphism. Since τ^2 is the identity map, Δ^2 truly commutes with any braid. A subword of the fundamental braid Δ is called a *permutation braid* and the set of all permutation braids is in one-to-one correspondence with the set $\sum_n$ of permutations on $\{0,1,...,n-1\}$. For example, Δ is the permutation sending *i* to *n-i*. The word length of a permutation *n*-braid is $\leq \frac{n(n-1)}{2}$. The *descant set* $D(\pi)$ of a permutation π is defined by $D(\pi) = \{i | \pi(i) > \pi(i+1)\}$. Any braid *b* can be written uniquely as $b = \Delta^u \pi_1\pi_2 ... \pi_l$ where *u* is an integer, π_i are permutation braids different from Δ and $D(\pi_{i+1}) \subset D(\pi_i^{-1})$. This unique decomposition of a braid *b* is called a *left canonical form*. All the braids in this paper are assumed to be in the *left-canonical form*. For example, for *a,b* ∈ B_n, *ab* means the left-canonical form of *ab* and so it is hard to guess its factors *a* or *b* from *ab*. In B_n, we say that two elements *x* and *y* are *conjugate* to each other if $y = axa^{-1}$ for some *a* in B_n and we write *x* ~ *y*. Here *a* or a^{-1} is called a *conjugator* and the pair (*x,y*) is said to be conjugate. The *Conjugacy Decision Problem* (*CDP*) asks to determine whether *x* ~ *y* for a given (*x*, *y*). Equivalently, we may ask that given two group words *x* and *y* in B_n, can we decide in a finite number of steps whether or not *x* and *y* are conjugate in B_n? In other words, does there exist an element *a* in B_n such that $y = axa^{-1}$? In [8], Garside proved that the *CDP* for braid groups is solvable, but the algorithm he proposed, as well as all improvements proposed thereafter, has a high cost that is exponential in the length of the considered words and the number of strands. The *Conjugacy Search Problem* (*CSP*) asks to find *a* in B_n satisfying $y = ax\,a^{-1}$ for a given instance (*x*, *y*) in B_n such that *x* ~ *y*. In other words, given two elements $x, y \in B_n$ and the information that $y = axa^{-1}$ for some *a* in B_n, *CSP* asks to find at least one particular element *a* like that. It is considered infeasible to solve *CSP* for sufficiently large braids. The probability for a random conjugate of *x* to be equal to *y* is negligible. For B_n, a pair $(x,y) \in B_n \times B_n$ is said to be *CSP-hard* if $x \sim y$ and *CSP* is infeasible for the instance (x,y).If (x,y) is *CSP-hard*, so is clearly (y,x).

Braid Decomposition Problem (BDP): Given two $n-$ braids w, s in B_n , find x, y in B_n such that $w = xsy$.

3. AUTHENTICATED KEY AGREEMENT

A key agreement protocol is a method in which a shared key, (session key), is obtained by two or more entities such that no single entity can presume the resulting key. Most of the times, entities use public channel driven by the adversaries and it may varies with every execution round of the protocol. This secret key can be used subsequently to create a secure communication channel among or between the entities.

Mostly, a key agreement protocol is called authenticated if the protocol is able to ensure that the session key is known only to the intended entities in a protocol run. Without authentication, a key agreement protocol would probably turn out to be insecure since an adversary can easily violate the scheme be using the man-in-the-middle attack as well as other related cryptographic attacks.

Wilson and Menzes [16, 17] have defined a number of desirable security features which are normally used to analyze key agreement protocol in today's era. They are:

Known-key security: A protocol is considered to be known session key secure if it remains unaffected achieving its goal in the face of an adversary who has learned some previous session keys.

(Perfect) forward secrecy: A protocol enjoys forward secrecy if the secrecy of the previous session keys is not affected when the long term private keys of one or more entities are compromised. Perfect forward secrecy refers to the scenario when the long term private keys of all the participating entities are compromised.

Key-compromise impersonation resistance: Suppose A's long term private key is disclosed. Then an adversary who knows this value can now impersonate A since it is precisely the value which identifies A. We can say that a protocol is *key compromise impersonation resistant* if this loss will not enable an adversary to masquerade as other legitimate entities to A as well or obtain other parties secret key.

Unknown key share resistance: In an *unknown-key share attack* an adversary convinces a group of entities that they share a key with the adversary whereas in fact, the key is shared between the group and another party. This situation can be exploited in a number of ways by the adversary when the key is subsequently used to provide encryption of integrity.

Key control resistance: It should not be possible for any of the participants (or an adversary) to force the session key to a presume value or predict the value of the session key.

4. PROPOSED PROTOCOL

4.1 Initial setup

Here we use three subgroups of B_n denoted by B_{n_1} , B_{n_2} , B_{n_3} which are generated as follows: We choose n_1, n_2, n_3, such that $n = n_1 + n_2 + n_3$. Also B_{n_i} (i = 1, 2, 3) is the subgroup of B_n consisting of braids made by braiding n_i - strands from the left among n-strands with the order n_1, n_2, n_3. Thus each $r_m \in B_{n_m}$ commutes with each $r_n \in B_{n_l}$ (m = 1, 2, 3 and n = 1, 2, 3 such that $m \neq$ l). Now we use the following symbols for our protocol:

$s \in B_n$	: sufficiently complicated $n-$braid
A, B, C	: three participants (users) who want to share a secret
$a_1, a_2 \in B_{n_1}$	: $A's$ long term private key pair
$b_1, b_2 \in B_{n_2}$	: $B's$ long term private key pair
$c_1, c_2 \in B_{n_3}$	: $C's$ long term private key pair
$X_1 = a_1sa_2$	: $A's$ long term public key
$X_2 = b_1sb_2$	: $B's$ long term public key
$X_3 = c_1sc_2$	: $C's$ long term public key

4.2 Key Agreement

Step I

- A chooses r_1, t_1 in B_{n_1}
- A sends r_1st_1 to B

Step II

- B chooses r_2, t_2 in B_{n_2}
- B sends r_1st_1 , $r_2r_1st_1t_2$, r_2st_2 to C

Step III

- C choose r_3, t_3 in B_{n_3}, computes $K_{31} = c_1X_1c_2$, $K_{32} = c_1X_2c_2$
- C sends $T_1 = K_{31}r_3r_2st_2t_3K_{31}^{-1}$, $T_2 = K_{32}r_3r_1st_1t_3K_{32}^{-1}$ to A and B respectively.

Step IV

- A computes K_{31} and the shared key : $S\,(A) = r_1K_{31}^{-1}T_1K_{31}t_1$
- B computes K_{32} and the shared key : $S\,(B) = r_2K_{32}^{-1}T_2K_{32}t_2$
- C also computes the shared key : $S\,(C) = r_3r_2r_1st_1t_2t_3$

4.3 Correctness

$$\begin{aligned} S\,(A) &= r_1K_{31}^{-1}T_1K_{31}t_1 \\ &= r_1K_{31}^{-1}\left(K_{31}r_3r_2st_2t_3K_{31}^{-1}\right)K_{31}t_1 \\ &= r_1r_3r_2st_2t_3t_1 \\ &= r_1r_2r_3st_3t_2t_1 \end{aligned}$$

$$\begin{aligned} S\,(B) &= r_2K_{32}^{-1}T_2K_{32}t_2 \\ &= r_2K_{32}^{-1}\left(K_{32}r_3r_1st_1t_3K_{32}^{-1}\right)K_{32}t_2 \\ &= r_2r_3r_1st_1t_3t_2 \\ &= r_1r_2r_3st_3t_2t_1 \end{aligned}$$

$$S\,(C) = r_3r_2r_1st_1t_2t_3$$

$$= r_1 r_2 r_3 s t_3 t_2 t_1$$

4.4 Security Analysis

Known – key security: It is clear that the session key of our protocol varies with every protocol run since it is established according to the values of the entities' ephemeral private key pairs $(r_1, t_1), (r_2, t_2)$, and (r_3, t_3) in that particular session. So, knowledge of past session keys will not allow the attacker to deduce the session keys afterward.

(Perfect) forward secrecy: Suppose the long-term private keys of all the entities are compromised. It permits an adversary to obtain session keys which are previously established between participators. But nobody can compute the previously established session key. In this case, if an adversary has learned that all entities long-term private key pairs, say (a_1, a_2), (b_1, b_2), and (c_1, c_2) at some point in the future, the adversary is not able to compute the previously established session key $S(A)$ without ephemeral private key (r_1, t_1). Similarly $S(B)$ and $S(C)$ cannot be computed without (r_2, t_2) and (r_3, t_3) respectively.

Key-compromise impersonation resistance: Key-compromise impersonation means that compromise of an entity's (say A) long-term private key (a_1, a_2) will allow an adversary E to masquerade as C (or B) to A. In our protocol, even though an adversary who has compromised A's private key could forge the message in the first run and compute the same session key with A, It cannot violate the signature on behalf of C(or B) to A. This key confirmation requirement makes our protocol resistant to key compromise impersonation attack.

Unknown key share resistance: Even though an insider attacker can compute the session key, it can not violate the signature on behalf of the other parties. Without knowing their private key, this key confirmation message makes the protocol secure to unknown key share attack.

Key control resistance: In our protocol, not even a single participant could force the session key to a predicted value since the session key of our protocol is derived by using the long term and ephemeral private keys of all the protocol participants.

5. CONCLUSION

This paper proposes a tripartite authenticated key agreement protocol using Braid decomposition problem and conjugacy problem in braid groups that resists all the security threats and provides key confirmation. It is secure and efficient since no participant can compel the session key to a predefined value.

6. REFERENCES

[1] Anshel, M.Anshel, B.Fisher,and D.Goldfeld, New key agreement protocols in braid group cryptography, Proc.of CT-RSA 2001,LNCS, **2020**,Springer-Verlag, 1-15.

[2] Anshel, M. Anshel and D. Goldfeld, An algebraic method of public-key cryptography, Math. Research Letters, **6** (1999), 287-291.

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

[4] J. Birman, Braids, Links, and Mapping Class Groups, Annals of Math. Studies, Princeton Univ. Press (1975).

[5] Atul Chaturvedi and Sunder Lal, An authenticated key agreement protol using conjugacy problem in braid groups, International Journal of Network security, Vol. 6, No. 2, pp. 181 – 184, Mar. 2008. (http://ijns.nchu.edu.tw/)

[6] M.M. Chowdhury, On the security of new key exchange protocols based on the triple decomposition problem, preprint 2007, http://www.aexiv.org/abs/cs.CR/0611065.

[7] M.M. Chowdhury, An authentication scheme using non – commutative semi groups, ieeexplore.ieee.org,2007.

[8] F. A. Garside, The braid group and other groups, Quart. J. Math. Oxford 20-78(1969) 235-254.

[9] K.H.Ko, D.H.Choi, M.S.Cho, and J.W.Lee, New signature scheme using conjugacy problem, (http://eprint.iacr.org/2002/168).

[10] K.H. KO, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang, and C Park, New public-key cryptosystem using braid groups, Advances in Cryptology, Proceeding of Crypto - 2000, Lecture Notes in Computer Science 1880, ed. M Bellore, springs Verlag (2000), 166-183.

[11] A. G. Myasnikov, V. Shpilrain and A. Ushakov, Group – based cryptography, Birkhauser, 2008.

[12] V. Shpilrain and A. Ushakov, A new key exchange protocol based on the decomposition problem, contemp. Math. 418 (2006) , 161 – 167.

[13] V. Shpilrain and A. Ushakov, An authentication scheme based on the twisted conjugacy problem, preprint (2008), http://arxiv.org/pdf/math.GR/0805.2701.

[14] H. Sibert, P. Dehornoy, & M. Girault, Entity authentication schemes using braid word reduction," in International Workshop on Coding and Cryptography (WCC) 2003, Discrete Applied Mathematics, **154-2**, Elsevier, 420 – 436 (2006). (http://eprint.iacr.org/2002/187).

[15] B. C. Wang, Y. P. Hu, Sinature scheme based on the root extraction problem over braid groups, ieeexplore.ieee.org, 2009

[16] S. B. Wilson, and A. Menezes, " Autenticated Diffie – Hellman key agreement protocols", Proceedings of the 5th Annual Workshop on Selected aeas in Cryptography (SAC' 98), LNCS, pp. 339 – 361, 1999.

[17] S. B. Wilson, D. Johnson and A. Menezes, Key agreement protocol and their security analysis", Proceedings of the 6th IMA International Conference on Cryptography and Coding, Vol. 1355, LNCS, pp. 339 – 361, Springer – Verlag, 1998.

[18] W. Yun, X . Gua – Hua, Z. Xing – Kai , B. Wan. Security analysis and design ofproxy signature schemes over braid groups, http://eprint.iacr.org/2009/458

[19] W. Yun, X . Gua – Hua, Z. Xing – Kai , B. Wan. A strong blind signature scheme over braid groups, http://eprint.iacr.org/2009/622.

[20] W. Yun, G. Xiong, W. Bao, and X. Zhang, A ring signature scheme over braid groups, Journal of Electronics, 27 (4), 522 – 527,2010.