An Authenticated Key Agreement Protocol Using Conjugacy Problem in Braid Groups

Atul Chaturvedi and Sunder Lal

(Corresponding author: Atul Chaturvedi)

Department of Mathematics Institute of Basic Science, Dr. B. R. Ambedkar (Agra) University
Khandari, Agra-282002(UP), India (Email: atulibs@gmail.com)

(Received Mar. 6, 2006; revised and accepted Apr. 29, 2006 & Dec. 21, 2006)

Abstract

In this paper we propose an authenticated key agreement, which works in a braid group. We prove that our protocol meet the security attributes under the assumption that the Conjugacy Search Problem (CSP) is hard in braid group.

Keywords: Authenticated key agreement, braid group, conjugacy search problem

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, 10, 9, 14]. The idea of applying braid group as a platform for cryptosystems was introduced by Anshel et al [2]. Braid groups, on the one hand, 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.

In [10], Ko et al. proposed a braid group version of Diffie-Hellman key agreement [6]. Man-in-the-middle attack works on this protocol, which sets ground for our work, presented in this paper. We improve the above scheme by proposing a new authenticated key agreement protocol based on CSP in braid groups. We make use of Conjugacy Search Problem (CSP) to suggest a new key agreement scheme. The CSP in braid groups is algorithmically difficult and consequently provides one-way functions. We use this characteristic of CSP to propose a key agreement protocol which is resistant to Man-in-the-middle attack.

The rest of the paper is organized as follows: We present a brief introduction of braid groups in Section 2. In Section 3, we give Ko's key agreement protocol. In Section 4, we define authenticated key agreement protocol. In Section 5, 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 Bn, the braid group of index n, using following generators and relations: Consider the generators $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$, 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 defining relations are $1.\sigma_i\sigma_j = \sigma_j\sigma_i \text{for} |i-j| > 1$, $2.\sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \text{for} |i-j| = 1$.

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 [5]. The braid $\Delta = (\sigma_1\sigma_2 \ldots\ldots \sigma_{n-1})(\sigma_1\sigma_2 \ldots\ldots \sigma_{n-2}) \ldots\ldots (\sigma_1\sigma_2)(\sigma_1)$ is called the *fundamental braid*. Δ nearly commutes with any braid b. In fact $\Delta b = \tau(b)\Delta$, where $\tau : B_n \longrightarrow B_n : \tau(\sigma_i) = \sigma_{n-1}$ 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 Σ_n of permutations on$\{0, 1, \ldots, n-1\}$. 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 $b = \Delta^u\pi_1\pi_2\ldots\pi_j$ where u is an integer, π_j are permutation braids different from Δ and $D(\pi_{j+1}) \subset D(\pi_J^{-1})$. This unique decomposition of a braid b is called a *left canonical form.* For example, for $a, b \in B_n, ab$ means the left-canonical form of ab and so it is hard to guess its factors a in B_n and we write $x \sim y$. 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 $x \sim y$ for a given (x, y). Equivalently, we can 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 such that $y = axa^{-1}$? In [7], Garside proves 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 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 $(x, y) \in B_n$ det $\times B_n$ 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).

3 Diffie-Hellman Key Agreement (DHKA)

3.1 DHKA for Finite Field

Suppose that A and B want to agree on a shared secret key using the Diffie-Hellman key agreement protocol [6]. They proceed as follows: First, A generates a random private value a and B generates a random private value b. Then they derive their public values using parameters p and g and their private values. A's public value is g^a mod p and B's public value is g^b mod p. They then exchange their public values. Finally, A computes $k_{ab} = (g^b)^a$ mod p, and B computes $k_{ba} = (g^a)^b$ mod p. Since $k_{ab} = k_{ba} = k, A$ and B now have a shared secret key k.

3.2 Braid Group Version of DHKA Using Conjugacy Problem

Ko et al. [10] proposed a braid group version of Diffie-Hellman key agreement protocol. Let B_n be a braid group where CSP is infeasible. As mentioned earlier, all the braids in B_n are assumed to be in the left canonical form. Thus for a, b in B_n, it is hard to guess a or b from ab. We assume that n is even, and denote by LB_n (resp.UB_n) the subgroup of B_n generated by $\sigma_1, \ldots, \sigma_{\frac{n}{2}-1}$, i.e., braids where the $n/2$ lower strands only are braided (resp. in the subgroup generated by $\sigma_{\frac{n}{2}-1}, \ldots, \sigma_{n-1}$). We know that every element in LB_n commutes with every element in UB_n.

Initial set up: A sufficiently complicated n-braid $x \in B_n$ for a large n is selected and is known to both the parties A and B.

Key agreement: (a) A chooses a random secret braid $a \in LB_n$ computes axa^{-1} and sends it to B. (b) B chooses $b \in UB_n$ computes bxb^{-1} and sends to A. (c) A receives bxb^{-1} and computes $a(bxb^{-1})a^{-1}$. (d) B receives axa^{-1} and computes $b(axa^{-1})b^{-1}$.

3.3 Man-in-the Middle Attack

Above protocol 3.2 is vulnerable to a middle-person attack. In this attack, an opponent, C, does the following

1) C intercepts A, s public value axa^{-1} and sends cxc^{-1} to B.

2) When B transmits his public value bxb^{-1}, C substitutes it with cxc^{-1} and sends it to A.

3) C and A thus agree on one shared key $K_{AC} = acxc^{-1}a^{-1}$ and C and B agree on another shared key $K_{BC} = bcxc^{-1}b^{-1}A$.

4) After this exchange, C simply decrypts any messages sent out by A or B, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the correct party. This vulnerability is due to the fact that Diffie-Hellman key agreement does not authenticate the participants.

To remove this attack we propose a new authenticated key agreement protocol.

4 Authenticated Key Agreement Protocol (AKAP)

In a key agreement protocol two or more distributed entities need to share some key in secret, called *session* key. This secret key can then be used to create a confidential communication channel amongst the entities. Since the path breaking work of Diffie-Hellman [6] in 1976, several key agreement protocols have been proposed over the years [10, 11, 12, 13, 16]. A number of desirable attributes of such key agreement protocols have been identified in [16]. Nowadays most protocols are analyzed with such attributes. These are listed as under:

- **Known-key security**. Each run of a key agreement protocol between two entities A and B should produce a unique secret key. Independent of previous session keys, if any. Thus a protocol should still achieve its goal even if an adversary has learned some other session keys.

- **Perfect forward secrecy**. If long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities should not be affected.

- **Key-compromise impersonation**. Suppose A's long-term private key is disclosed. Clearly an adversary that knows this value can now impersonate A, since it is precisely this value that identifies A. However, it may be desirable in some circumstances that this loss does not enable the adversary to impersonate other entities to A.

- **Unknown key-share**. Entity B cannot be coerced into sharing a key with entity A without B's knowledge, i.e., when B believes the key is shared with some entity $C \neq A$, and A (correctly) believes the key is shared with B.

- **Key control**. Neither entity should be able to force the session key to be a pre-selected value.

5 The Proposed Scheme

In this section we describe our two-pass Authenticated Key Agreement Protocol (AKAP) between two entities A and B, and consider its security. For our scheme, the initial setup known to both A and B is same as in the previous Scheme 3.2: We denote by

$$\begin{array}{rl} x: & \text{Sufficiently complicated } n\text{-braid;} \\ r \in LB_n: & A\text{'s long term private key;} \\ X_a = rxr^{-1}: & A\text{'s long term public key;} \\ s \in UB_n: & B\text{'s long term private key;} \\ X_b = sxs^{-1}: & B\text{'s long term public key.} \end{array}$$

5.1 Key Agreement

Following the above mentioned notations, we describe the AKAP below. The protocol works in the following steps.

$$\begin{array}{ccc} A & & B \\ & Y_a & \\ Y_a = cxc^{-1} & & \\ & \longrightarrow & \\ & Y_b & \\ & & \text{Compute } K_b = sX_as^{-1} \\ & \longleftarrow & Y_b = KbdY_ad^{-1}K_b^{-1} \end{array}$$

Figure 1: Two-pass AKA protocol

1) A choose $c \in LB_n$, computes $Y_a = cxc^{-1}$. If $Y_a = I$ (Identity braid),terminates the protocol run with failure. Otherwise A sends it to B.

2) Upon receiving Y_a, B choose $d \in UB_n$, computes $K_b = sX_as^{-1}$, and ${}_{Y}b = K_bdY_ad^{-1}K_b^{-1}$.

3) If K_b or $Y_b = I$, B terminates the protocol run with failure. Otherwise B sends it to A.

4) Upon receiving Y_b, A computes $K_a(= K_b) = rX_br^{-1}$, and the shared key $KEY_a = cK_a^{-1}Y_bK_ac^{-1}$.

5) B also computes the shared key $KEY_b = dY_ad^{-1}$.

6) In each step 4 and 5, if KEY_a or KEY_b is I, then the protocol run is terminated with failure.

7) After regular protocol running, A and B share the secret $K = KEY_a = KEY_b$.

5.2 Security Consideration

Here we prove our protocol meets the following desirable attributes under the assumption that the root problem is hard.

Known-Key Security: If A and B execute the regular protocol run, they clearly share their unique session key K as above.

(Perfect) Forward Secrecy: During the computation of the session key K for each entity, the random braids c and d still act on it. An adversary who may have captured their private keys r or s should extract K_a or K_b from the information Y_a and Y_b to know the previous or next session keys between them. However, this contradicts that CSP is hard. Hence, under the assumption that the CSP is secure, AKAP meets the *forward secrecy.*

Key-Compromise Impersonation: Suppose A's long-term private key, r, is disclosed. Now an adversary who knows this value can clearly impersonate A. Is it possible for the adversary to impersonate B to A without knowing B's long-term private key, s? For the success of the impersonation, the adversary must know A's **ephemeral** key c at least. So, also in this case, the adversary should extract c from A's ephemeral public value $Y_a = cxc^{-1}$. This also contradicts that CSP is hard.

Unknown Key-share: Suppose an adversary E now try to make A believe that the session key is shared with B, while B believes that the session key is shared with E. To launch the unknown key-share attack, the adversary E should set his public key to be certified even though he does not know his correct private key. For this, E makes it by utility the public values X_a, X_b and x. With some simple calculations, we see that the unknown key-share attack fails.

Key Control: As the same argument in the above, the key-control is clearly impossible for the third party. The only possibility of *key-control* attack may be brought out by the participant of the protocol, B. But for participant B, in order to make him a party, A generate the session key $K(KEY_b)$ which is pre-selected value by B. For example B should solve the following $K = dY_ad^{-1}$. But this again falls into the problem of CSP.

6 Conclusion

In this paper we proposed a new authenticated key agreement protocol, called AKAP. Our protocol makes use of the fact that the CSP is hard in the braid group. We prove that our scheme is secure against many well know attacks on protocols.

References

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

[2] I. Anshel, M. Anshel, and D. Goldfeld, “An algebraic method of public-key cryptography,” *Mathmetics Research Letters*, vol. 6, pp. 287-291, 1999.

[3] E. Artin, “Theory of braids,” *Annals of Mathematics*, vol. 48, pp. 101-126, 1947.

[4] M. Bellare and P. Rogaway, “Entity Authentication and Key Distribution,” in *Advances in Cryptology (Crypto'93*, pp. 341-358, 1994.

[5] J. Birman, “Braids, links, and mapping class groups,” *Annals of Mathematics Studies* , Princeton University Press, 1975.

[6] W. Diffie and M.Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[7] F. A. Garside, “The braid group and other groups,” *Quarterly Journal of Mathematics*, Oxford 20-78, pp. 235-254, 1969.

[8] J. Hughes,“A Linear Algebraic Attack on the AAFG1 Braid Group Cryptosystem,” in *ACISP'02*, LNCS 2384, pp. 176-189, Springer-Verlag, 2002.

[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,” in *Advances in Cryptology (Crypto'00)*, LNCS 1880, pp. 166-183, Springer-Verlag, 2000.

[11] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, *An Efficient Protocol for Authenticated Key Agreement*, Technical Report CORR98-05, Department of CO, University of Waterloo, 1998.

[12] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, “An efficient protocol for authenticated key agreement,” *Design, Codes and Cryptography*, vol. 28, no. 2, pp. 119-134, 2003.

[13] A. Menezes, M. Qu, and S. Vanstone, “Key agreement and the need for authentication,” in *Proceedings of PKS'95*, pp. 34-42, 1995.

[14] H. Sibert, P. Dehornoy, and M. Girault, “Entity authentication schemes using braid word reduction,” in *Discrete Applied Mathematics*, vol. 154, no. 2, pp. 420-436, 2006.

[15] V. B. Styshnev, “The extraction of a root in a braid group (English),” *Mathematics of the USSR Izvestija*, vol. 13, pp. 405-416, 1979.

[16] S. B. Wilson, D.Johnson, and A. Menezes,“Key agreement protocol and their security analysis,” in *Proceedings of Sixth IMA International Conference on Cryptography and Coding*, pp. 30-45, 1997.

Sunder Lal is currently Professor and Head of Department of Mathematics. At present he is a Dean, Faculty of Science, IBS Khandari, Dr. B. R. A. University, Agra, INDIA. He is member of Indian Mathematical Society, Group for Cryptographic Research and Cryptography Research Society of India. His current research interests include Cryptography, Number theory and Mathematics Education.

Atul Chaturvedi received his M.Sc and M. Phil. Degrees from Dr. B. R. A University, Agra. He is currently a Doctoral candidate under the instruction of Prof. Sunder Lal. He is a lecturer in the Department of Mathematics, Dr. B. R. A. University, Agra. He is a member of Group for Cryptographic Research and Cryptography Research Society of India. His current research interests include Braid Group Cryptography.