Int. J. of Advanced Networking and Applications
Volume: 01, Issue: 05, Pages: 327-330 (2010)

A Secure Key Agreement Protocol Using Braid Groups

Atul Chaturvedi
Department of Mathematics, Pranveer Singh Institute of Technology (PSIT) Kanpur (UP), India
Email : atulibs@gmail.com
Shyam Sundar
Department of Mathematics, Pranveer Singh Institute of Technology (PSIT) Kanpur (UP), India
Email : ssmishra_15@yahoo.co.in

ABSTRACT

This paper presents an authenticated key agreement protocol based on a braid group. It is proved that the proposed protocol meets several security attributes under the assumption that the Root Problem (RP) in braid group is a hard problem.

Keywords : Braid group, Root Problem, key agreement, authentication, security.

Date of Submission: January 27, 2010 Date of Acceptance: May 01, 2010

I. INTRODUCTION

Non commutative groups, specially Braid groups of Artin in recent years have emerged as suitable setting for cryptographic protocols [1,2,8,9,13].The idea of using the braid group as a platform for cryptosystems was first introduced in 1999 by Anshel, Anshel and Goldfeld [2]. The useful feature of Braid groups is that they are more complicated than Abelian groups, but are not too complicated to work with. These two characteristics make braid group a convenient and suitable choice.

However, recent results about the linearity of braid groups and Lawrence-Krammer representations have made these cryptosystems vulnerable to linear algebra based attacks. In particular Hughes [7] has shown that key generation methods discussed in [1] are not secure. This suggests using other problems for cryptographic protocols using Braid groups as platform. Root problem (RP) has been suggested by Sibert, Dehronoy, and Girault in 2003[13]. They also remarked that in open literature there is no cryptographic protocol based on RP. Here we use Root Problem to suggest a new key agreement scheme. Root Problem (RP) in braid groups is algorithmically difficult, and consequently provide one-way functions. We use it to propose a key agreement protocol over a braid group.

Traditional symmetric cryptography is based on the sender and receiver of a message knowing and using the same secret key: the sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. However the main problem in this scheme is in getting the sender and receiver to agree on the secret key without anyone else getting to know it. If they are in separate physical locations, they must trust a courier, or a phone system, or some other transmission medium to prevent the disclosure of the secret key being communicated. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all messages encrypted or authenticated using that key. The generation of such keys is called *key agreement*; and all cryptosystems must deal with key agreement issues. Because all keys in a symmetric cryptosystem must remain secret, secret-key cryptography often has difficulty providing secure key agreement, especially in open systems with a large number of users.

The concept of key agreement was introduced in 1976 by W. Diffie and M.Hellman [6]. In their seminal scheme each person gets a pair of keys, one called the *public* key and the other called the *private* key. Each person's public key is published while the private key is kept secret. The need for the sender and receiver to share secret information is thus eliminated: all communications involve only public keys, and no private key is ever transmitted or shared. No longer is it necessary to trust some communications channel to be secure against eavesdropping or betrayal. The only requirement is that public keys are associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory). Anyone can send a confidential message by just using public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient.

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 mention its desirable attributes. In section 4, we present our protocol, and we give a proof of security for our scheme. The paper ends with conclusion.

Volume: 01, Issue: 05, Pages: 327-330 (2010)

II. BRAID GROUPS

Emil Artin [**3**] in 1925 defined B_n, the braid group of index n, using following generators and relations: Consider the *generators* $s_1, s_2, ..., s_{n-1}$, where s_i represents the braid in which the $(i+1)^{st}$ string crosses over the i^{th} string while all other strings remain uncrossed. The *defininig relations* are

1. $s_i s_j = s_j s_i \; for \left|i-j\right| > 1$,

2. $s_i s_j s_i = s_j s_i s_j \; for \left|i-j\right| = 1$.

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. The braid $\Delta = (s_1 s_2 ..... s_{n-1})(s_1 s_2 ..... s_{n-2}) ..... (s_1 s_2)(s_1)$ is called the *fundamental braid*. Δ nearly commutes with any braid b. In fact $\Delta b = t(b)\Delta$, where $t : B_n \to B_n : t(s_i) = s_{n-i}$ is an automorphism. Since t^2 is the identity map, $?^2$ truly commutes with any braid. A subword of the fundamental braid ? 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 $\{0,1,\ldots,n-1\}$ For example, ? 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* $D(p)$ of a permutation p is defined by $D(p) = \{i \,|\, p(i) > p(i+1)\}$. Any braid b can be written uniquely as $b = \Delta^u p_1 p_2 .. p_l$ where u is an integer, p_i are permutation braids different from Δ and $D(p_{i+1}) \subset D(p_i^{-1})$. 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 $a,b \in B_n$, ab means the left-canonical form of ab and so it is hard to guess its factors a or b from ab.

If b is a non-trivial and $e \geq 2$ is an integer, then b^e is never identity. In other words, the braid groups are torsion-free. The *Root Problem* in B_n is to find, given y and $e \geq 2$, an x such that $y = x^e$. It is proved in [14] that RP is decidable but is computationally infeasible if braids of a sufficient size are considered.

III. 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,[5,8,10,11,12]. A number of desirable attributes of such key agreement protocols have been identified in [5].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 to an adversary he/she can impersonate A, since it is precisely this value that identifies A. This attribute requires that this loss should not enable such an adversary to impersonate other entities to A.

* **Unknown key-share**. It should not be possible to coerce A to share a key with entity B without A's knowledge, i.e., when A believes the key is shared with some entity $C \neq B$, and B correctly believes the key is shared with A.

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

IV. 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 a braid group B_n, where RP 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 $s_1, ..., s_{\frac{n}{2}-1}$, i.e., braids where the $n/2$ lower strands only are braided (resp. in the subgroup generated by $s_{\frac{n}{2}+1}, ..., s_{n-1}$). We know that every element in LB_n commutes with every element in UB_n. We denote by

s	:	sufficiently complicated n-braid
e	:	integer ≥ 2
$a_1, a_2 \in LB_n$	:	A's long term private key pair
$a_1^e s a_2^e = X_a$	:	A's long term public key
$b_1, b_2 \in UB_n$	:	B's long term private key pair
$b_1^e s b_2^e = X_b$	:	B's long term public key
h	:	strong one-way hash function

IV.I KEY AGREEMENT

Here we describe the AKAP following the above notations. The protocol works in the following steps.

1. A randomly chooses x_1, and x_2 in LB_n , computes $x_1^e s x_2^e = Y_a$ If $Y_a = I$ (Identity braid), A terminates the protocol and restarts with new x_1 and x_2.
   A, then sends $h(Y_a)$ to B.

Volume: 01, Issue: 05, Pages: 327-330 (2010)

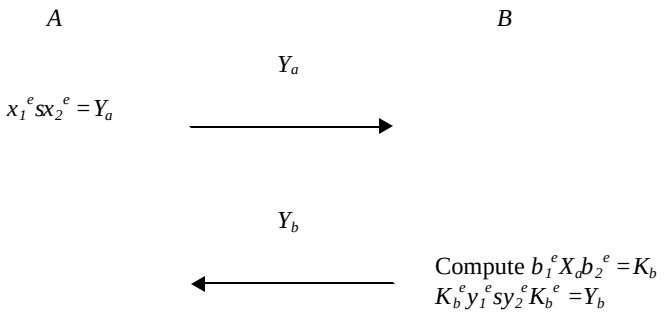

Fig.1. Two-pass AKA Protocol

2. Upon receiving Y_a, B randomly chooses y_1, and y_2 in UB_n , computes $K_b = b_1^e X_a b_2^e$, and $Y_b = K_b^e y_1^e s y_2^e K_b^e$. If K_b or $Y_b = I$, B terminates the protocol and restarts with new y_1 and y_2. B, then sends $h(Y_b)$ to A.
3. Upon receiving Y_b, A computes $K_b = a_1^e X_b a_2^e = K_a$, and the shared key $KEY_a = x_1^e K_a^{-e} Y_b K_a^{-e} x_2^e$.
4. B also computes the shared key $KEY_b = y_1^e Y_a y_2^e$.
5. After regular protocol running, A and B share the secret $K = KEY_a = KEY_b$.

IV.II SECURITY CONSIDERATION

Here we show that 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, because $KEY_a = x_1^e K_a^{-e} Y_b K_a^{-e} x_2^e$
$= x_1^e K_a^{-e} K_b^e y_1^e s y_2^e K_b^e K_a^{-e} x_2^e = x_1^e y_1^e s y_2^e x_2^e = y_1^e x_1^e s x_2^e y_2^e$
$= y_1^e Y_a y_2^e = KEY_b$.

(Perfect) Forward Secrecy: During the computation of the session key K for each entity, the random braids x_1, x_2, y_1, y_2 still act on it. An adversary who captured their private keys (a_1, a_2) or (b_1, b_2) 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 is the very root problem. Hence, under the assumption that the RP is computationally infeasible, AKAP meets the *forward secrecy* requirement.

Key-Compromise Impersonation: Suppose A's long-term private key, (a_1, a_2), is disclosed. Now an adversary who knows this value can clearly impersonate A. Is it possible for the adversary impersonates B to A without knowing the B's long-term private key, (b_1, b_2)? For the success of the impersonation, the adversary must know A's ephermeral key (x_1, x_2) at least. So, also in this case, the adversary should extract (x_1, x_2) from A's ephemeral public value $Y_a = x_1^e s x_2^e$. This also contradicts that RP is hard.

Unknown Key-share: We examine the unknown key-share attack that allows an adversary E to make one party believe K to be shared with E while it is in fact shared with a different party. A common scenario is that E has X_a certified without knowing the private key a of A, and uses it to talk with B as E while she poses as B to A simultaneously. Our protocol is secure against this attack because for E, we have $h(X_a)$ $h(X_e)$ in computing each K

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 the entity B, to make the party, A generate the session key K (KEY_b) which is pre-selected value by B, for example B should solve the following $K = y_1^e Y_a y_2^e$. But this again falls into the problem of RP.

V. CONCLUSION

In this paper we proposed a new authenticated key agreement protocol, called AKAP. Our protocol makes use of the fact that the RP is hard in the braid group It is secure in the sense that it meets some desirable attributes of secure AKA protocol.

REFERENCE

[1] I.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] I. 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**,101-126, 1947.

[4] M.Bellare, P.Rogaway, *Entity authentication and key distribution*, Proceeding of CRYPTO'93, Santa Barbara, USA,1994,341-358.

[5] S. Blake-Wilson, D.Johnson, A.Menezes, *Key agreement protocol and their security analysis*, Proceedings of Sixth IMA International Conference

on Cryptography and Coding, Cirencester, UK,1997, 30- 45.

[6] W.Diffie, & M.Hellman, *New directions in cryptography*, IEEE Trans. Inform. Theory,22 (6), 1976,644-654.

[7] J.Hughes, *A linear algebraic attack on the AAFG1 braid group cryptosystem*, ACISP, 2002, 176-189, Springer Lect.Notes in Comp. Sci.2384.

[8] 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.

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

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

[11] L.Law, AMenezes, M.Qu, J.Solinas, & S.Vanstone, *An efficient protocol for authenticated key agreement*, Design,Codes and Cryptography, 28(2), 2003, 119-134.

[12] A.Menezes, M.Qu, & S.Vanstone, *Key agreement and the need for authentication*, Proceedings of PKS'95, Toranto,Canada,1995.

[13] H.Sibert, P.Dehornoy, & M.Girault, *Entity authentication schemes using braid word reduction*, WCC 2003,to appear;http://eprint.iacr.org/2002/187.

[14] V.B.Styshnev, *The extraction of a root in a braid group (English)*, Math. USSR Izv. **13,** 1979, 405-416.

Authors Biography

Atul Chaturvedi received his M.Sc, M. Phil., and Ph.D. Degrees from Dr. B. R. A University, Agra. He is currently an Associate Professor in the Department of Mathematics, Pranveer Singh Institute of Technology (PSIT), Kanpur. He is a member of Group for Cryptographic Research and Cryptography Research Society of India. His current research interests include Applied Mathematics and Braid Group Cryptography.

Shyam Sundar received his M.Sc Degrees from Kanpur University and Ph.D. from HBTI, Kanpur. He is currently an Assistant Professor in the Department of Mathematics, Pranveer Singh Institute of Technology (PSIT), Kanpur. His current research interests include Applied Mathematics and Cryptography.