ON AN AUTHENTICATION SCHEME BASED ON THE ROOT PROBLEM IN THE BRAID GROUP

BOAZ TSABAN

ABSTRACT. Lal and Chaturvedi proposed two authentication schemes presumably based on the difficulty of the Root Problem in the braid group. We describe a deterministic linear time algorithm to crack the first scheme, and show that the second scheme is not more secure than schemes based on the Conjugacy Search Problem, and can therefore be cracked by existing heuristic attacks with very good success probability, as long as the parameters are practical.

1. THE FIRST AUTHENTICATION SCHEME

Lal and Chaturvedi propose in [6] two authentication schemes based on the difficulty of the Root Problem in the braid group. The basic definitions are given in [6]. Their first scheme is defined as follows. We work in the braid group B_n where n is even. In the sequel, multiplication of elements of B_n means concatenation and reduction to left canonical form. Let $LB_n = \langle \sigma_1, \dots, \sigma_{n/2-1} \rangle$ and $UB_n = \langle \sigma_{n/2+1}, \dots, \sigma_n \rangle$.

Key Generation. Alice chooses integers $r, s \geq 2$, $a \in LB_n$, and $b \in UB_n$. The public key is $(X = a^r b^s, r, s)$, and the secret key is (a, b).

Authentication. Bob chooses $c \in UB_n$ and $d \in LB_n$, and sends Alice the challenge $Y = c^r d^s$. Alice responds with (a hash image of) $Z = a^r Y b^s$. Bob verifies that $Z = c^r X d^s$.

It is argued in [6] that the scheme is secure if the Root Problem of finding x given x^r ($r \geq 2$ fixed) in B_n is difficult.

The first observation is that extraction of roots is not necessary in order to crack this scheme.

Claim 1. *If one can, given xy where $x \in LB_n$, and $y \in UB_n$, find (x, y), then one can authenticate as Alice.*

Proof. Take $x = a^r$ and $y = b^s$. Then xy is known. Find $(x, y) = (a^r, b^s)$, and note that this suffices for the authentication. $\square$

Supported by the Koshland Center for Basic Research.

Claim 1 together with the following proposition implies that the scheme is insecure.

Proposition 2. *Given xy where $x \in LB_n$, and $y \in UB_n$, there is an efficient algorithm to find (x, y).*

Proof (sketch). The approach adopted here was suggested to us by Shmuel Kaplan. We merely had to prove that it works.

Given a braid $w \in B_n$ containing a strand starting at position i, there is a well-defined braid $r_i(w) \in B_{n-1}$ obtained by *removing that strand* from the braid w (and enumerating the starting an ending positions in the unique order-preserving manner). Note that $r_i(w)$ can be computed in time polynomial in n and the number of Artin generators used to write w.

Similarly, for a set $I \subseteq \{1, \dots, n\}$, we can define $r_I(w) \in B_{n-k}$ to be the braid obtained by removing all strands starting at positions which are members of I (and then re-enumerating the starting and ending positions in an order-preserving manner).

Assume that $x \in LB_n$ and $y \in UB_n$, and we are given a representative a of the homotopy class of xy. a is homotopically equivalent to the braid b which is the side-by-side concatenation of (represetatives of the homotopy classes of) x and y. Let $I = \{n/2+1, \dots, n\}$. Then $r_I(a)$ is homotopically equivalent to $r_I(b)$ (the same homotopy works). But $r_I(b) = x$.

In summary, given xy, compute $z = r_I(xy)$. Then (in B_n) $z = x$, and we can compute $x^{-1}(xy) = y$. □

The attack works for any group G with LB_n and UB_n replaced by any two commuting subgroups L, U of G with $L \cap U = \{e\}$, provided that for each $x \in L$ and each $y \in U$, x and y can be efficiently recovered from their product xy (note that the condition $L \cap U = \{e\}$ guarantees that such a decomposition is unique). This is the case, for example, with the subgroups A_s, B_s of Thompson's group F, defined in Shpilrain-Ushakov's paper [9] (see the proof of Proposition 1 of [9]).

González-Vasco has pointed out to us that if L, U are as above, and in addition either L or U is a *normal* subgroup of G, then for each known $w \in G$, xwy can be efficiently (and uniquely) decomposed: If L is a normal subgroup of G, then $w^{-1}xw \in L$, and therefore we can decompose $w^{-1}xwy$ into $w^{-1}xw$ and y (and then recover x from $w^{-1}xw$). The case that U is a normal subgroup of G is treated similarly.

2. The second authentication scheme

Lal and Chaturvedi also propose a second scheme in [6], *Scheme II*:

Key Generation. Alice chooses integers $r, s \geq 2$, $a \in LB_n$, and $c \in B_n$. The public key is $(X = a^r ca^s, c, r, s)$, and the secret key is a.

Authentication. Bob chooses $b \in UB_n$, and sends Alice the challenge $Y = b^r cb^s$. Alice responds with (a hash value of) $Z = a^r Y a^s$. Bob verifies that $Z = b^r X b^s$.

The attack of Section 1 does not apply to Scheme II. To crack this scheme, it suffices to solve the following *Decomposition Problem*:

> Given xcy where $x, y \in LB_n$ are unknown and $c \in B_n$ is known, find $\tilde{x}, \tilde{y} \in LB_n$ such that $\tilde{x}c\tilde{y} = xcy$.

In principle, the generic attack described in [3, 8] applies to this problem, and it seems that for practical parameters required to make the system usable, its success probability will not be negligible. However, the generic attack is much more time consuming than the one suggested in Section 1, and to evaluate its feasibility, it must be tested against practical parameters, which so far have not been suggested.

We therefore consider again the original Scheme II. Here too, the powers are irrelevant for our discussion, so we consider instead the following *Generalized Scheme II*: Fix a group G and subgroups A, B of G such that A, B commute elementwise.

Key Generation. Alice chooses $a_1, a_2 \in A$, and $c \in G$. The public key is $(X = a_1 ca_2, c)$.

Authentication. Bob chooses $b_1, b_2 \in B$, and sends Alice the challenge $Y = b_1 cb_2$. Alice responds with (a hash value of) $Z = a_1 Y a_2$. Bob verifies that $Z = b_1 X b_2$.

In order to crack Generalized Scheme II, it suffices to solve the following.

Problem 3. *Given c, $X = a_1 ca_2$, and $Y = b_1 cb_2$ such that $a_1, a_2 \in A$ and $b_1, b_2 \in B$, find $Z = a_1 Y a_2 = b_1 X b_2$.*

More precisely, the elements a_1, a_2, b_1, b_2, c are chosen according to known distributions on the relevant spaces (A, B, and G), and one has to find Z with a significant probability. Similar probabilistic adaptations can be made to all assertions in the sequel, but for clarity we often omit those.

Lemma 4. *Consider an instance of Problem 3. If either b_1 or b_2 is known to commute with c, then Z can be computed efficiently by anyone.*

Proof. If $b_1c = cb_1$, then $cb_1b_2 = b_1cb_2 = Y$ is known, and therefore so is b_1b_2. It follows that

$$Z = a_1Ya_2 = a_1b_1cb_2a_2 = a_1cb_1b_2a_2 = a_1ca_2b_1b_2 = X(b_1b_2)$$

is known. The case where $b_2c = cb_2$ is similar. □

Remark 5. Note that in the original Scheme II, b_1, b_2 are both powers of the same element $b \in B$, and if b commutes with c, then both b_1 and b_2 commute with c. It could, however, be the case that b^r commutes with c, but b^s does not: In B_n, the fundamental element Δ does not commute with all elements, but its square Δ^2 does.

As the roles of (a_1, a_2, Y) and (b_1, b_2, X) in Problem 3 are symmetric, Lemma 4 implies the following.

Lemma 6. *Consider an instance of Problem 3. If either a_1 or a_2 is known to commute with c, then Z can be computed efficiently by anyone.* □

Assume now that Bob generates b_1, b_2 in a way that with a nontrivial probability p, either b_1 or b_2 commute with c. Then, in about $2/p$ tries, false identification is possible: In each try, the pretender flips a coin to guess whether b_1 or b_2 commutes with c, and uses Lemma 4. This will succeed with probability $p/2$.

But actually, one could heuristically check whether b_1 or b_2 commute with c. By Lemma 6, if c commutes with all elements of A (or commutes, with probability close to 1, with the elements of A generated in the protocol, a fact that can be verified experimentally), then the system is insecure. Thus, we may assume that it is easy to generate elements $a \in A$ which do not commute with c. Fix such an element a. Compute

$$\begin{aligned} W &= (b_1cb_2)a(b_1cb_2)^{-1} = b_1cb_2ab_2^{-1}c^{-1}b_1^{-1} = \\ &= b_1cab_2b_2^{-1}c^{-1}b_1^{-1} = b_1cac^{-1}b_1^{-1}. \end{aligned}$$

If $b_1c = cb_1$, then

$$W = cb_1ac^{-1}b_1^{-1} = cab_1c^{-1}b_1^{-1} = cac^{-1}b_1b_1^{-1} = cac^{-1},$$

which can be verified as we know a and c. And if not, then it is unlikely that $W = cac^{-1}$, that is, that $b_1db_1^{-1} = d$, where $d = cac^{-1}$. The last assertion just tells that cac^{-1} commutes with b_1.

A similar argument applies for b_2: Computing

$$U = (b_1cb_2)^{-1}a(b_1cb_2) = (b_1cb_2)^{-1}a(b_1cb_2) = b_2^{-1}c^{-1}acb_2,$$

we have that if $b_2c = cb_2$, then $U = c^{-1}ac$, and otherwise, this is unlikely, as $U = c^{-1}ac$ if, and only if, b_2 commutes with $c^{-1}ac$.

We arrive at the following.

Lemma 7. *In instances of Problem 3, if it is easy to find elements $a \in A$ such that with a high probability, cac^{-1} (respectively, $c^{-1}ac$) does not commute with b_1 (respectively, b_2), then it can be checked with high certainty whether $b_1c = cb_1$ (respectively, $b_2c = cb_2$).* □

In practical settings where c does not commute with all elements of A, it is likely that any generic enough element of A will have the properties required in Lemma 7. Moreover, since we can repeat the test of commutation for many distinct a's, the "high probability" in Lemma 7 does not seem necessary.

By symmetry, if c commutes with any of a_1, a_2, then this can also be detected heuristically, and by Lemmas 6 and 4, false identification is possible in these cases, either.

We may therefore assume that none of the generated elements commutes with c. In particular, there are many $a \in A$ which do not commute with c.

The following approach is inspired by the beautiful observations used by Chowdhury in a different context [2]. For the elements a which do not commute with c, we obtain as above the equation

$$W = b_1 d b_1^{-1}$$

with $d = cac^{-1}$ being a rather generic element of G. Similarly, we can obtain conjugacy equations for b_2, a_1, and a_2. This reduces the problem to the (strict) *Simultaneous Conjugacy Search Problem*:

> Given many equations $W = xdx^{-1}$ where $x \in B$ is unknown and $d \in G$ is known, find x (modulo the center of G).

To see that this suffices, consider an instance of Problem 3, and assume that $h, g \in G$ are central (i.e., commute with all elements of G), and that hb_1, gb_2 are known. Then

$$(hb_1)c(gb_2) \cdot Y^{-1} = (hg)(b_1cb_2) \cdot Y^{-1} = (hg)Y \cdot Y^{-1} = hg$$

can be computed, and therefore so can

$$(hg)^{-1}(hb_1)X(gb_2) = g^{-1}b_1Xgb_2 = g^{-1}gb_1Xb_2 = b_1Xb_2 = Z.$$

Moreover, it suffices to know either of b_1 or b_2 modulo the center of G in order to find the other modulo the center of G. Indeed, if h is in the center of G and b_1h is known, then

$$c^{-1}(b_1h)^{-1}Y = c^{-1}(b_1h)^{-1}b_1cb_2 = c^{-1}h^{-1}cb_2 = c^{-1}ch^{-1}b_2 = h^{-1}b_2$$

can be computed, and is equal to b_2 modulo the center of G. The case that b_2 is known modulo the center of G is treated similarly.

In principle, there could exist a solution to the given equations $W = xdx^{-1}$ which is not equal to x modulo the center of G, but this seems to be unlikely in nontrivial scenarios. Indeed, if $\tilde{x} \in B$ is another solution to all of these equations, then for each d used in the equations, $xdx^{-1} = \tilde{x}d\tilde{x}^{-1}$, and therefore $\tilde{x}^{-1}x$ commutes with d. Since the attacker can generate as many such equations as desired and the elements d look rather generic, it follows that $\tilde{x}^{-1}x$ is likely to belong to the center G, and therefore $\tilde{x}$ is equal to x modulo the center of G.

This shows, heuristically, that Generalized Scheme II is not likely to be more secure than schemes based on the simultaneous conjugacy search problem.

We now move back to the original setting, where $G = B_n$ is the braid group. There are a variety of efficient heuristic algorithms for the simultaneous conjugacy search problem in B_n, which have very good success rates [5, 7]. Moreover, since we can choose many elements $a \in A$, we can produce as many families of conjugacy equations for b_1 (or for b_2, a_1 or a_2) as we need and it suffices to solve correctly (modulo the group's center) one such family of equations, the probability of success of the mentioned methods should get very close to 1.

This suggests that in practical settings, Scheme II is likely to be insecure, either.

Acknowledgments. We thank María Isabel González-Vasco and Dima Ruinskiy for their useful suggestions. For a treatment of the *standard* root problem in the braid group, see [4].

References

[1] J. S. Birman, V. Gebhardt, and J. Gonzalez-Meneses, *Conjugacy in Garside Groups III: Periodic braids*, eprint `http://arxiv.org/abs/math.GT/0609616` (2006).

[2] M. Chowdhury, *On the security of new key exchange protocols based on the triple decomposition problem*, eprint `http://arXiv.org/abs/cs/0611065` (2006).

[3] D. Garber, S. Kaplan, M. Teicher, B. Tsaban, and U. Vishne, *Probabilistic solutions of equations in the braid group*, Advances in Applied Mathematics **35** (2005), 323–334.

[4] A. Groch, D. Hofheinz, and R. Steinwandt, *A practical attack on the root problem in braid groups*, Contemporary Mathematics **418** (2006), 121–132.

[5] D. Hofheinz and R. Steinwandt, *A Practical Attack on Some Braid Group Based Cryptographic Primitives*, Proceedings of PKC 2003, Lecture Notes in Computer Science **2567** (2003), 187–198.

[6] S. Lal and A. Chaturvedi, *Authentication schemes using braid groups*, eprint http://arXiv.org/cs.CR/0507066 (2005).

[7] S. Maffre, *A weak key test for braid based cryptography*, Designs, Codes and Cryptography **39** (2006), 347–373.

[8] D. Ruinskiy, A. Shamir, and B. Tsaban, *Length-based cryptanalysis: The case of Thompson's Group*, Journal of Mathematical Cryptology, to appear.

[9] V. Shpilrain and A. Ushakov, *Thompson's group and public key cryptography*, ACNS 2005, Lecture Notes in Computer Science **3531** (2005), 151–164.

Department of Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel

E-mail address: boaz.tsaban@weizmann.ac.il

URL: http://www.cs.biu.ac.il/~tsaban