15/Nov/2005

Cryptanalysis of the Shpilrain-Ushakov Thompson group cryptosystem

Dima Ruinskiy, Adi Shamir and Boaz Tsaban

Weizmann Institute of Science, Rehovot, Israel

Thompson's group

Thompson's group is an infinite non-abelian group, defined, in terms of generators and relations, as follows:

$$F = \left\langle x_0, x_1, x_2, ... \mid x_i^{-1} x_k x_i = x_{k+1} \, (k > i) \right\rangle$$

It is well known that each element of Thompson's group has a unique normal form

$$z = x_{i_1} \cdots x_{i_s} x_{j_1}^{-1} \cdots x_{j_t}^{-1} ,$$

where:

1) $i_1 \le \cdots \le i_s$ and $j_1 \le \cdots \le j_t$.
2) If x_i and x_i^{-1} both occur, then either x_{i+1} or x_{i+1}^{-1} occurs as well.

Using this, one can define a natural length function on the elements of Thompson's group:

$$\ell(z) = s + t ,$$

where z, s, t are as above. In other words, $\ell(z)$ is the number of generators in the normal form of z.

The Shpilrain-Ushakov key agreement protocol

0) Let s be some positive integer. Let $S_A = \left\{x_0 x_1^{-1}, ..., x_0 x_s^{-1}\right\}$, $S_B = \left\{x_{s+1}, ..., x_{2s}\right\}$ and $S_W = \left\{x_0, ..., x_{s+2}\right\}$. Let A_s, B_s and W_s be the subgroups of Thompson's group, generated by the sets S_A, S_B and S_W, respectively. Then for each $a \in A_s$ and each $b \in B_s$ $ab = ba$ [1].

1) Two positive integers s and L are fixed, as well as a word $w \in W_s$, chosen so that $\ell(w) = L$.

2) Alice selects at random elements $a_1 \in A_s$ and $b_1 \in B_s$, such that $\ell(a_1) = \ell(b_1) = L$, computes $u_1 = a_1 w b_1$ and sends u_1 to Bob.

3) Bob selects at random elements $a_2 \in A_s$ and $b_2 \in B_s$, such that $\ell(a_2) = \ell(b_2) = L$, computes $u_2 = b_2 w a_2$ and sends u_2 to Alice.

4) Alice computes $K_A = a_1 u_2 b_1 = a_1 b_2 w a_2 b_1$, whereas Bob computes $K_B = b_2 u_1 a_2 = b_2 a_1 w b_1 a_2$. Because $a_1 b_2 = b_2 a_1$ and $a_2 b_1 = b_1 a_2$, $K_A = K_B$, and so the parties share the same secret key.

The attack

This protocol is insecure if an eavesdropper can use the known elements w and $u_i = a_i w b_i$ to obtain a_i and b_i. Note that finding either a_i or b_i is sufficient for that, because $b_i = w^{-1} a_i^{-1} u_i$, and similarly $a_i = u_i b_i^{-1} w^{-1}$.

Since a_i and b_i are generated as products of generators from a known set, we used a length-based attack, similar to the one described in [2]. In each step of this attack we look for the leftmost generator of the element we try to recover. The algorithm tries each of the possible generators and keeps in memory the M sequences generators that yield the shortest length. M is a pre-defined constant number, which depends on the available computational power. See [2] for details.

Parameters and experimental results

In [1], it is suggested to select s from the interval $[3,8]$ and L as any even number from the interval $[256,320]$.

We generated random elements as described in the protocol, and then attempted to find either a_i or b_i. An experiment was declared successful if either one of them was successfully recovered.

Following are the results of the attack we obtained for various values of L. In all experiments we used $s=5$ and $M=512$. The experiments were performed on a single-core Intel ® Pentium M ® processor, running at 1.8GHz, under Microsoft ® Windows XP Professional ®. 200 iterations of the experiment were run for each value of L and the success probability was calculated among these 200 iterations. The time per iteration was computed with respect to this machine.

L	Success probability	Time per iteration (sec)
2	100%	0.005
4	100%	0.01
8	99%	0.23
16	72%	1.66
32	40%	6.58
64	18%	32.5
128	4%	203.1

As expected, when L increases, the running time of the algorithm increases and its success probability decreases. However, even for $L=128$ the algorithm managed to break the cryptosystem with non-negligible probability using standard computational power and a small running time.

When L=256

In this case we increased M to 1024 and ran several thousand iterations of the experiment. To speed things up, the experiment was run in parallel on 10 Intel ® Xeon ® processors. After less than 24 hours of work, about 2000 iterations were completed, *14 of which were successful*, indicating a success rate of 0.7%. This suggests that L must be increased much beyond the suggestion in [1] in order to make the protocol resistant against the attack presented here.

We are currently working on finding better length functions on Thompson's group and on alternative approaches that will increase the success probability of the attack.

References

[1] V. Shpilrain, A. Ushakov, *Thompson's group and public key cryptography*, http://arxiv.org/abs/math/0505487

[2] D. Garber, S. Kaplan, M. Teicher, B. Tsaban, U. Vishne, *Probabilistic solutions of equations in the braid group*, http://arxiv.org/abs/math.GR/0404076