# Length-Based Attacks for Certain Group Based Encryption Rewriting Systems

James Hughes
Storage Technology Corporation
7600 Boone Avenue North
Minneapolis, MN 55428

Allen Tannenbaum
Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, MN 55455

March 29, 2000

**Abstract**

In this note, we describe a probabilistic attack on cryptosystems based on the word/conjugacy problems for finitely presented groups of the type proposed in [1]. The attack is based on having a canonical representative of each string relative to which a length function may be computed. Hence the term *length attack*.

## 1 Introduction

Recently, a novel approach to public key encryption based on the algorithmic difficulty of solving the word and conjugacy problems for finitely presented groups has been proposed in [1]. The method is based on having a canonical

minimal length form for words in a given finitely presented group, which can be computed rather rapidly, and in which there is no corresponding fast solution for the conjugacy problem. A key example is the Braid group. In this note, we will indicate a possible probabilistic attack on such a system, using the length function on the set of conjugates defining the public key. Note that since each word has a canonical representative, the length function is well-defined and *can be computed*. The attack may be relevant to more general types of string rewriting cryptosystems, and so we give some of the relevant background. Thus this note will also have a tutorial flavor.

The contents of this paper are as follows. In Section 2, we make some general remarks are rewriting systems, and the notion of "length" of a word. In Section 3, we define the Artin and Coxeter groups. In Section 4, we discuss the classical word and conjugacy problems for finitely presented groups. In Section 5, the Braid cryptosystem of [1] is described. In Section 6, we give the length attack for possibly compromising such a cryptosystem, and finally in Section 7 we draw some general conclusions, and directions for further research for group rewriting based encryption systems.

The authors would like to thank Professor Paul Garrett for a number of very helpful conversations about cryptography.

# 2 Background on Monoid and Group Based Rewriting Systems

In this section, we review some of the relevant concepts from group theory for rewriting based encryption. We work in this section over a monoid, but similiar remarks hold for group based rewriting systems as well.

Let $k$ be an arbitary field, and $S = \{a_1, \ldots, a_n\}$ a finite set. Let $S^*$ be the finite monoid generated by $S$, that is,

$$S^* = \{a_{\sigma(1)}^{i_1} \cdots a_{\sigma(n)}^{i_n}\}.$$

Elements of $S^*$ are called *words*. We then define the *free algebra* generated by $S$ to be

$$A = k[S^*] = k < S >= \{\sum k_{i_1 \ldots i_n} a_{\sigma(1)}^{i_1} \cdots a_{\sigma(n)}^{i_n}\}.$$

We are now ready to define precisely the key notion of *rewriting system*. Let $R \subset S^* \times S^*$. We call $R$ the set of replacement rules. Many times the pair

$(u, v) \in R$ is denoted by $u \to v$. The idea is that when the word $u$ appears inside a larger word, we replace it with $v$. More precisely, for any $x, y \in S^*$, we write

$$xuy \to xvy,$$

and say that the word $xuy$ has been *re-written* or *reduced* to $xuy$. $x$ is *irreducible* or *normal* if it cannot be rewritten.

We will still need a few more concepts. We say that the *rewriting system* $(S, T)$ is *terminating* if there is no infinite chain $x \to x_1 \to x_2 \to \cdots$ of re-writings. We then say that the partial ordering $x \geq y$ defined by $x \to \cdots \to y$ is *well-founded.* $R$ is *confluent* if a word $x$ which can be re-written in two different ways $y_1$ and $y_2$, the re-writings $y_1$ and $y_2$ can be re-written to a common word $z$.

Note that if $R$ is terminating, confluence means that there exists a unique irreducible word, $x_{red}$ representing each element of the monoid presented by the re-writing system. Such a system is called *complete.* Given a word $x \in S^*$, we define the *length of* $x$, to be the number of generators in $x_{red}$.

**Remark:**
In the case of groups, the basic outline just given is valid. A key example of a group in which one can assign a length function is the Braid group via the results in [4]. This is the basis of the cryptosystem proposed in [1].

# 3 Artin and Coxeter Groups

In this section, we review some of the pertinent background on Artin and Coxeter groups. An excellence reference for this material in [3], especially for the Braid groups.

Let $G$ be a group. For $a, b \in G$ we define

$$< ab >^q := aba \ldots, \quad \text{product with } q \text{ factors.}$$

For example,

$$< ab >^3 := aba, \; < ab >^4 := abab, \; < ba >^5 := babab.$$

An *Artin group* is a group $G$ which admits a set of generators $\{a_i\}_{i \in I}$ with $I$ a totally ordered index set, and with relations of the form

$$< a_i a_j >^{m_{ij}} = < a_j a_i >^{m_{ji}},$$

for any $i, j \in I$ and with $m_{ij}$ non-negative integers. The matrix $M := [m_{ij}]_{i,j \in I}$ is called the *Coxeter matrix.*

The *Braid group*, $B_n$, is defined by taking the indexing set $I := \{1, \ldots, n\}$, and

$$\begin{aligned} m_{ij} &= 2 \quad \text{for } |i-j| > 1, \\ m_{i,i+1} &= m_{i+1,i} = 3. \end{aligned}$$

Thus the *Braid group* $B_n$ is a special case of an Artin group defined by the generators $\sigma_1, \ldots, \sigma_n$, with the relations

$$\begin{aligned} \sigma_i \sigma_j &= \sigma_j \sigma_i \quad |i-j| > 1,\ i, j \in I, \\ \sigma_i \sigma_{i+1} \sigma_i &= \sigma_{i+1} \sigma_i \sigma_{i+1}. \end{aligned}$$

Given an Artin group $G$ with Coxeter matrix $M := [m_{ij}]_{i,j \in I}$ the assoicated *Coxeter group* is defined by adding the relations $a_i^2 = 1$, for $i \in I$. One can easily show them that a Coxeter group is equivalently defined by the relations

$$(a_i a_j)^{m_{ij}} = 1, \quad i, j \in I, \text{ with } m_{ii} = 2.$$

Artin groups and their associated Coxeter groups have some nice properties which could make them quite useful in potential rewriting based systems as we will now see.

# 4 Word and Conjugacy Problems for Finitely Presented Groups

Let

$$G = < s_1, s_2, \cdots, s_n : r_1, \cdots, r_k >$$

be a finitely presented group. Let $U$ be the free monoid generated by $s_i$ and $s_i^{-1}$. Then the *word problem* is given two strings (words), $u, v \in U$, decide if $u = v$ in $G$. The *conjugacy problem* is to decide if there exists $a \in G$ such that $u = ava^{-1}$, i.e., $u$ and $v$ are conjugates.

It is well-known that both these problems are algorithmically unsolvable for general finitely presented groups. However, for some very important groups they are solvable, e.g., for Artin groups with finite Coxeter groups. In

fact, Brieskorn and Saito [2] give an explicit solution to the word and conjugacy problems for this class of groups. Their algorithm runs in exponential time however. See also [7] and the references therein for some recent results on the word and conjugacy problems for Coxeter groups.

In some recent work, Birman-Ko-Lee [4] show that for the Braid group, the word problem is solvable in polynomial time (in fact, it is quadratic in the word length). For another solution to this problem see [6].

At this point, there is no known polynomial time algorithm known for the conjugacy problem for the Braid group with $n \geq 6$ strands; see [4]. This remark is essential to the security of the Braid cryptosystem in [1].

## 4.1 Distance Between Words

In this section, we observe that if one can find a unique irreducible word from which one can derive a length function, then one can give a natural distance between words in a given group $G$. We will see that this is the case for the Braid group via Birman-Ko-Lee [4].

Let $\alpha, \beta, \gamma$ denote words relative to a finite presentation of the group $G$. Let $\ell$ denote the length function which we assume exists. Then we define the *distance* $d$ between the words $\alpha, \beta$ as

$$d(\alpha, \beta) := \ell(\alpha\beta^{-1}).$$

We can easily check that $d$ is indeed a distance function. We have the following:

**Lemma 1** *$d$ has the following properties:*

*(1)* $d(\alpha, \beta) = d(\beta, \alpha)$.

*(2)* $d(\alpha, \beta) = 0$ *if and only if* $\alpha = \beta$.

*(3)* $d(\alpha, \gamma) + d(\gamma, \beta) \geq d(\alpha, \beta)$.

*Thus $d$ defines a distance function.*

**Proof.** Property (1) follows from the fact that for any $g \in G$, $\ell(g) = \ell(g^{-1})$. Property (2) follows since

$$d(\alpha, \beta) = \ell(\alpha\beta^{-1}) = 0$$

if and only if $\alpha\beta^{-1}$ is equivalent to null string, i.e., $\alpha = \beta$. Property (3) (the triangle inequality) follows from the facts that for any elements $g, h \in G$,

$$\ell(g) + \ell(h) \geq \ell(gh),$$

and

$$(\alpha\gamma^{-1})(\gamma\beta^{-1}) = \alpha\beta^{-1}.$$

# 5 Braid Cryptosystem

In some very interesting recent work, Anschel et al. [1] propose a new twist to rewriting systems for public key encryption. We will first state their approach over a general group. We should first note however that the use of the word and conjugacy problems for public-key cryptosystems is not new. An early reference is [9].

The general idea is as follows: Alice ($A$) and Bob ($B$) have as their public keys subgroups of a given group $G$,

$$S_A =< s_1, \ldots, s_n >, \quad < t_1, \ldots t_m > .$$

$A$ chooses a secret element $a \in S_A$ and $B$ chooses a secret element $b \in S_B$. $A$ transmits the set of elements $a^{-1}t_1a, \ldots, a^{-1}t_ma$ and $B$ transmits the set of elements $b^{-1}s_1b, \ldots, b^{-1}s_nb$. (The elements are rewritten is some fashion before transmission.)

Now suppose that

$$a = s_{\sigma(1)}^{i_1} \cdots s_{\sigma(n)}^{i_n}.$$

Then note that

$$\begin{aligned} b^{-1}ab &= b^{-1}s_{\sigma(1)}^{i_1} \cdots s_{\sigma(n)}^{i_n} b \\ &= b^{-1}s_{\sigma(1)}^{i_1} b \; b^{-1}s_{\sigma(2)}^{i_2} b \cdots b^{-1}s_{\sigma(n)}^{i_n} b \\ &= (b^{-1}s_{\sigma(1)}b)^{i_1} \cdots (b^{-1}s_{\sigma(n)}b)^{i_n}. \end{aligned}$$

(The conjugate of the product of two elements is the product of the conjugates.) Thus $A$ can compute $b^{-1}ab$, and similarly $B$ can compute $a^{-1}ba$. The common key then is

$$a^{-1}b^{-1}ab = [a, b],$$

the commutator of the two secret elements.

Note that since the two users have the common key written in different forms, in order to extract the message, it must be reduced to an identical group element. For the Braid group, this can be accomplished by reducing the commutator to the Birman-Ko-Lee canonical form [4]. In [1], the method of Dehornoy [6] for comparing Braids is also proposed for rewriting public elements.

The Braid group is particularly attractive for this protocol since one has a quadratic time solution for the word problem, and the only known solution to the conjugacy problem is exponential.

# 6 The Length Attack

In this section, we describe the length attack on word/conjugacy based encryption systems in which one can associate a canonical representative, and therefore a length function of the type described above. We assume that the group $G$ has generators $g_1, \ldots, g_N$ subject to certain relations. For example, for the Braid group one can take the standard Artin generators.

The idea is that group elements with long lengths have a higher probability of forming noncommutative *tangles* which have a smaller probability of combining with other factors. The particular probability will obviously be a function of the group and its presentation. Clearly, the significant probability of factor commutation and factor annihilation seem to be a basis for a well-constructed cryptosystem.

The remainder of this discussion will be a way of using substantial tangles in a *length attack* , and calculating an upper bound for the actual difficulty of this attack. It is important to emphasize that the ability of removing large tangles is not a general solution to the Braid Conjugacy problem. It is a specific attack on word/conjugates encryption systems of the type defined above. Indeed, for such crytosystems one has the some key information about the secret elements, namely, the factors are known and their number bounded.

Let

$$a \in S_A =< s_1, \ldots, s_n >,$$

be the secret element. Recall that in the above protocol, $a^{-1}t_r a$ and $t_r$ $(r = 1, \ldots m)$ are publicly given. We also assume that the factors $s_i$ have

lengths large relative to $a$. For given $r$, set

$$u_r = a^{-1} t_r a.$$

Then the idea is to compute

$$\ell(s_j^{\pm 1} u_r s_j^{\mp 1}),$$

repeatly. If $\ell(s_j^{\pm 1} u_r s_j^{\mp 1})$ decreases, one has found a correct factor of $a$ with a certain probability which depends of the lengths $\ell(s_i)$ for $i = 1, \ldots, n$. The key is that the lengths $\ell(s_i)$ should be large. In this case, there is the greatest probability of a tangle being formed which can be used to glean information about $a$.

We can estimate the workload in carrying out such a procedure. Without loss of generality we can assume that $a$ is made up of $n$ distinct factors combined in $d$ ways. If the length of the $s_i$ is large, then one join a small number of these factors together to create a substantial tangle. If we include the inverses of the generators, we should consider $2n$ factors. Let us call the number of factors necessary to make a tangle $k$. Thus we can create $(2n)^k$ tangled factors to try.

By trying all tangles, a pattern that there are certain factors which annihilate better than others should be observed. One can do this on a single public conjugate in $(2n)^k$ operations. This pattern can be significantly reinforced by repeating this $n$ times on each public conjugate $at_ra^{-1}$. Combining all the steps above brings us to $n(2n)^k$ operations.

Relative to the lengths $\ell(s_i)$ of the generators $s_i$ (and the specific group chosen), we conjecture that in a number of cases this will be sufficiently reliable to removing a given $s_i$, so that backtracking will not be necessary. We can now do this $dn$ times bringing the total to $dn(2n)^k$ operations.

This is polynomial to the number of different factors, and linear to the number of factors in the public keys. This is the basis of the length attack.

In some sense, the length attack is reminiscent of the "smoothness" attack for the Diffie-Hellman public key exchange system based on the discrete logarithm [8]. In this case, the protocol may be vulnerable when all of the prime factors of $q - 1$ (where the base field for the discrete logarithm has $q$ elements) are small. (Such a number is called *smooth*.)

# 7 Conclusions

We have made a computation which indicates that a length attack on a conjugacy/word problem cryptosystem of the type defined above has difficulty bounded above by $dn(2n)^k$. Given this conjecture, the only exponential aspect is the number of factors necessary to form a reliable tangle. To make this secure, $k$ needs to be 100 or larger.

In addition, as described, this attack does not use many tricks that one can use in order to speed up this length algorithm by several orders of magnitude. These include randomized and/or genetic algorithms which lead to more probabilistic solutions.

It is important to note that this attack does not solve the general conjugacy problem for the Braid group. Indeed, in this case the factors of $a$ are known and bounded. In the general conjugacy problem, the number of possible factors of $a$ is infinite. Consequently, the the conjugacy problem seems to be much harder and not amenable to this technique. The key exchange of the type proposed in [1] requires the factors be known and communicated, and give the attacker far more information than is known to the general conjugacy problem.

At this point, we are planning on statistically testing the length attack on several groups including the Braid group and affine Coxeter groups. These latter groups are very geometric, and may lead to much sharper bounds.

# References


[1] I. Anschel, M. Anschel, and D. Goldfeld, "An algberaic method for public-key cryptography," *Mathematical Research Letters* **6** (1999), pp. 1-5.

[2] E. Brieskorn and K. Saito, "Artin Gruppen und Coxeter Gruppen," *Invent. Math.* **17** (1972), pp. 245-271.

[3] J. Birman, *Braids, Links, and Mapping Class Groups*, Annals of Mathematics Studies, Princeton University Press, Princeton, New Jersey, 1975.

[4] J. Birman, K. Ko, and S. Lee, "A new approach to the word and conjugacy problems in the braid groups," *Advances in Math.* **139** (1998), pp. 322-353.

[5] D. Cox, J. Little, D. O'Shea, *Using Algebraic Geometry*, Springer-Verlag, New York, 1998.

[6] P. Dehornoy, "A fast method for comparing braids," *Advances in Math.* **127** (1997), pp. 200-235.

[7] H. Eriksson, *Computational and Combinatorial Aspects of Coxeter Groups,* Doctoral Dissertation, NADA, KTH, Sweden, September 1994.

[8] B. Schneier, *Applied Cryptography (Second Edition)*, Wiley Publishing, New York, 1996.

[9] N. Wagner and M. Magyarik, "A public key cryptosystem based on the word problem," *Advances in Cryptology: Proceedings of Crypto'84* edited by G. Blakely and D. Chaum, *Lecture Notes in Computer Science* **196** (1985), pp. 19-36, Springer, New York.