AN OVERVIEW OF BRAID GROUP CRYPTOGRAPHY

KARL MAHLBURG

Abstract. The past several years have seen an explosion of interest in the cryptographic applications of non-commutative groups. Braid groups in particular are especially desirable, as they provide difficult computational problems and can be implemented quite efficiently. Several different groups of researchers have proposed numerous cryptographic protocols that make use of braid groups, but unfortunately, flaws have been found in nearly every one. This expository paper discusses the specifications, attacks, and responses of both the Anshel, Anshel, and Goldfeld Commutator [2, 1] and the Cho et al. Diffie-Hellman Conjugacy [10] key exchange protocols.

1. Introduction

The goal of public-key cryptography is to communicate securely over public channels, so that a malicious interloper cannot obtain any secret data even if he is able to read the transmitted messages. Most of the methods currently in use are based on arithmetic over finite fields, be they systems that rely on modular arithmetic like RSA, or systems that use the group action of elliptic curves like Certicom. The potential advent of quantum computers is very troubling, because all of these cryptosystems are easily broken by such machines. The expository paper by Koblitz and Menezes contains a broader review of the current state of affairs [23].

Many people have investigated non-commutative algebraic structures in hopes of finding a new alternative, and for a time, braid groups seemed to hold a great deal of promise [2, 10, 14]. After closer investigation it has been discovered that braid groups have perhaps too much structure, and many of the same techniques for efficient computations can be used to attack braid-based protocols [8, 17, 21, 22, 25]. The Conjugacy problem in braid groups forms the basis for many proposed cryptosystems, and recent results have shown that the problem is more feasible than many people had expected [6, 7, 19].

The body of the paper begins with an introduction to braid groups in Section 2, followed by the description of two prominent protocols that utilize them in Section 3. Some of the known weaknesses, attacks and modifications for these protocols appear in Section 4.

I would like to acknowledge a brief but beneficial discussion with Andrew Bolstad, who researched implementation aspects of braid group cryptography [8] for a similar report. It should also be recorded that Helger Lipmaa maintains an excellent online clearinghouse of progress in braid group cryptography at http://www.tcs.hut.fi/ helger/crypto/link/public/braid/.

Date: December 13, 2004.

2. Braid Groups

The braid groups are (highly) non-commutative torsion-free groups that were first introduced by Artin [3]. They are of cryptographic interest because computations and data storage can be performed quite efficiently, but they are complex enough that at first glance it seems unlikely that they have any unexpected underlying structure. In this section, they are first defined geometrically in terms of braids, and then the following subsections give algebraic group presentations and describe certain canonical forms that are used in algorithms. The author regrets that he is not more skilled at producing graphics in LaTeX, as the visual representation is a powerful tool in analyzing and understanding braid groups. The beginning reader is strongly encouraged to sketch all of the braid relations throughout this paper, and the graphical interpretation will often be explained for complicated algebraic expressions involving braids.

The name "braid group" is completely natural, as the n-th braid group B_n is defined as a set of "n-braids." Consider a set b of n non-intersecting bands that start at the points $(x, y, z) = (0, i, 0)$ for $1 \leq i \leq n$, and end at the corresponding points $(1, \lambda_b(i), 0)$. The set of equivalence classes under isotopy of all such bands is then precisely the set of n-braids, and it is clear that each class is uniquely given by the ordered set of crossings between bands. Here a crossing is defined to be an intersection between two bands in the projection of a braid to the (x, y) plane (when drawing braids, the top and bottom bands in a crossing are distinguished), and any braid implicitly has finitely many crossings (the trivial braid has zero crossings). The group operation is defined simply by connecting the endpoints and concatenating two braids, and the inverse of a braid is constructed by reversing each crossing sequentially. The well-developed theories of knots and links provide powerful tools for using the complement of a braid in ambient space to describe the possible configurations, and it is the author's understanding that this plays a role in certain classifications of braids [26].

2.1. **Artin presentation.**

In the above description, it is clear that crossings between adjacent bands are the fundamental units in constructing braids. The Artin presentation gives B_n in terms of the generators $\{\sigma_i\}_{1 \leq i \leq n-1}$, where σ_i is a braid with exactly one crossing: band $i+1$ passes over band i. The relation between σ_i and

σ_i^{-1} leads to the designation of a crossing as *positive* if $i+1$ crosses above i, and otherwise it is *negative*.

Definition 2.1. The $(n+1)$-th braid group is given by the presentation

(2.1)
$$B_{n+1} := < \sigma_1, \ldots, \sigma_n \mid \sigma_i \sigma_{i+1} = \sigma_{i+1} \sigma_i,\ \sigma_{i+1} \sigma_i \sigma_{i+1} = \sigma_i \sigma_{i+1} \sigma_i,\ \ 1 \leq i \leq n-1 > .$$

Remark. This definition falls into the more general framework of *Artin* and *Coxeter Groups* (see [5]), which have generators $\{a_i\}_{i \in I}$ and relations of the form

$$< a_i a_j >^{m_{ij}} = < a_j a_i >^{m_{ji}}, \tag{2.2}$$

where the notation $< b_1 b_2 \ldots b_k >^n$ represents the n-term truncation of the infinite periodic word $\overline{b_1 b_2 \ldots b_k}$. For a given Artin group G, the associated Coxeter group is obtained by adding the relation $a_i^2 = 1$ for all i. Those who are familiar with presentations of symmetric groups will recognize that the Coxeter group

of B_n is precisely the Coxeter presentation of the symmetric group S_n. Another generalization of braid groups is the Garside groups [18].

The preceding remark implies that there is a surjective homomorphism ϕ that sends a braid $b \in B_n$ to a permutation $\phi(b) = \pi \in S_n$ defined by $\pi(i) := \lambda_b(i)$. This permutation records only the ending position of each band, and ignores the sign of the crossings. In other words, the crossings σ_i and σ_i^{-1} both map to the same transposition $(i,\ i+1)$. A braid is said to be *pure* if $\phi(b)$ is trivial.

It is helpful to have a canonical inverse for ϕ that defines the set $\widetilde{S}_n := \phi^{-1}(S_n)$ of *permutation braids.* The permutation braids are also referred to as *simple* braids. Any permutation π decomposes into adjacent transpositions as $\pi = (t_1,\ t_1 + 1)\cdots(t_k,\ t_k+1)$, so $\phi^{-1}(\pi) := \sigma_{t_1}\cdots\sigma_{t_k}$. An equivalent description is to construct the simplest possible braid that maps band i to $\pi(i)$ for all i, with every crossing positive. The map ϕ^{-1} is henceforth denoted by a tilde, as $\phi^{-1}(\pi) = \tilde{\pi}$.

The permutation braids are a special subset of the set of *positive* braids B_n^+, which is the monoid formed by restricting to words in σ_i^{+1} under the same relations (2.1) as the original braid group. Garside proved the key property that two positive braids a and b are equivalent in B_n if and only if they are equivalent in B_n^+. This implies that the order relation

$$a \leq b \Leftrightarrow b = ac,\ a, b, c \in B_n^+$$

forms a lattice from the positive braids [18] (Garside is also responsible for much of the following development).

One permutation braid is particularly useful in decomposing general braids into simple factors.

Definition 2.2. The *fundamental braid* in B_n is $\Delta_n := \widetilde{\pi_n}$, where $\pi_{\pi_n}(i) = n+1-i$.

Geometrically, Δ_n is the braid in which all n bands are rotated through a half-turn, with the relative order among the bands preserved. Every pair of bands meets in a single crossing, and thus Δ_n is a word of length $n(n-1)/2$ in the Artin generators.

The fundamental braid has a number of remarkable properties.

Proposition 2.3. *Suppose that* $1 \leq i \leq n-1$.

(1) *There are equivalent formulae for the fundamental braid:*

$$\begin{aligned}\Delta_n =& (\sigma_i)(\sigma_{i+1}\sigma_i)\cdots(\sigma_{n-1}\cdots\sigma_i)(\sigma_{i-1}\sigma_i\cdots\sigma_{n-1})\cdots(\sigma_1\cdots\sigma_{n-1}) \\ =& (\sigma_1\cdots\sigma_{n-1})\cdots(\sigma_1\cdots\sigma_{i+1})(\sigma_i\cdots\sigma_1)\cdots(\sigma_i\sigma_{i-1})(\sigma_i).\end{aligned} \tag{2.3}$$

(2) *Commutation is described by* $\Delta_n\sigma_i = \phi_n(\sigma_i)\Delta_n$, *where* $\phi_n(\sigma_i) := \sigma_{n-i}$.
(3) *If* $n \geq 3$, *then the center of* B_n *is* $< \Delta_n^2 >$.
(4) *Every permutation braid* $\tilde{\pi}$ *is a left factor of* Δ_n *in* B_n^+, *so that* $\Delta_n = \tilde{\pi}c$ *for some simple* $c \in B_n^+$.

The explicit formulae and the commutation relation are readily apparent from the geometric perspective, even though they are cumbersome in the algebraic group presentation. For example, $\Delta_n\sigma_i$ is a braid in which all of the bands are rotated a half-turn, followed by band $i+1$ crossing over band i. If the half-turn is undone before the crossing, then the crossing moves to $n-i+1$ over $n-i$, with the half-turn following, which is just the braid $\sigma_{n-i}\Delta_n$. But this same fact requires several lines

to prove algebraically,

$$\begin{aligned}\sigma_i\Delta_n &= \sigma_i(\sigma_{n-1})(\sigma_{n-2}\sigma_{n-1})\cdots(\sigma_1\cdots\sigma_{n-1}) \\ &= (\sigma_{n-1})\cdots\sigma_i(\sigma_{i+1}\cdots\sigma_{n-1})(\sigma_i\cdots\sigma_{n-1})\cdots(\sigma_1\cdots\sigma_{n-1}) \\ &= (\sigma_{n-1})\cdots(\sigma_i\cdots\sigma_{n-1})\sigma_{n-1}(\sigma_{i-1}\cdots\sigma_{n-1})\cdots(\sigma_1\cdots\sigma_{n-1}) \\ &= \Delta_n\sigma_{n-i},\end{aligned} \tag{2.4}$$

where the last line follows from inductively applying the fact that

$$\sigma_{n-k}(\sigma_{i-k}\cdots\sigma_{n-1}) = (\sigma_{i-k}\cdots\sigma_{n-1})\sigma_{n-k-1}. \tag{2.5}$$

The properties of the fundamental braid allow for the construction of a canonical decomposition of an arbitrary braid into a sequence of simple braids.

Definition 2.4. A sequence $A_1\cdots A_p$ of simple braids $A_i \in \widetilde{S_n}$ is *normal* if for every adjacent pair, A_i is the maximal simple braid that appears on the left of any equivalent form of A_iA_{i+1}.

Note that this definition can also be stated in terms of the lattice ordering. Again, the condition seems rather technical symbolically, but geometrically is easy to understand. A simple braid is one in which each pair of bands crosses at most once, and every crossing is positive. Thus every braid starts with a unique, maximal simple braid that is obtained by pushing to the left as many crossings as possible without introducing a second twist between any two bands.

Proposition 2.3 implies that for any i, a negative crossing can be written as

$$\sigma_i^{-1} = \Delta_n^{-1}A \tag{2.6}$$

for some simple word A. Therefore all of the negative crossings in an arbitrary word can be converted to negative powers of Δ_n, which are then easily pushed to the left by the commutation relation. The representation of a braid in the following theorem is referred to as the *normal* (left-weighted) form of a braid [16].

Theorem 2.5. *Any $b \in B_n$ is equivalent to a unique word of the form*

$$b = \Delta_n^u A_1A_2\cdots A_p,$$

where u is an integer, and $A_1\cdots A_p$ is a normal sequence.

The index p in the normal form of a braid b is known as the *length* of the braid, which is an important invariant for cryptological purposes. Note that a braid may be encoded as a data set containing an integer and a finite number of permutations. The normal form is very useful for computations, as there are polynomial-time algorithms for converting an arbitrary word into normal form, and in fact, for a generic sequence of simple words, many of the factors will remain unchanged!

2.2. **Band generator presentation.** Although the rest of this paper will focus mainly on the Artin presentation, a new efficient implementation of the braid group uses the band generators, which serve as a sort of dual to the Artin generators [6, 7, 13]. For every pair $s > r$, the element a_{sr} represents a braid in which band s and r are swapped, with band s crossing on top, and band r crossing over every other band apart from s. Thus

$$a_{sr} = \sigma_{s-1}\cdots\sigma_{r+1}\sigma_r\sigma_{r+1}^{-1}\cdots\sigma_{s-1}^{-1} \tag{2.7}$$

and $a_{i+1,i} = \sigma_i$, so that the Artin and band generators clearly generate the same group. The presentation is

$$B_n := < a_{sr} \mid a_{ts}a_{rq} = a_{rq}a_{ts},\ a_{ts}a_{sr} = a_{tr}a_{ts} = a_{sr}a_{tr} >, \tag{2.8}$$

where the relations are defined only for the non-trivial cases, when the values of distinct subscripts q, r, s, and t are all different and satisfy the proper inequalities.

Just as braids written with the Artin generators reduce to the nice canonical form of Theorem 2.5, the band generators also allow a decomposition for general braids. The fundamental word is now given by

$$\delta_n := a_{n,n-1} \cdots a_{32}a_{21}, \tag{2.9}$$

and the *simple braids* are similarly defined to be the set of left factors over all positive decompositions of δ_n. The combinatorics of such words can be described, and turn out to be indexed by the Catalan numbers for nested parentheses. Note that the center is generated by $\delta_n^n = \Delta_n^2$, as it can be seen that every strand makes one complete rotation.

The fundamental word δ_n also satisfies many properties analogous to Proposition 2.3 for Δ_n, and a normal sequence is still defined to be a sequence of simple braids that are maximally weighted to the left. The theorem is the analogous decomposition result.

Theorem 2.6. *Any $b \in B_n$ is equivalent to a unique word of the form*

$$b = \delta_n^u A_1 A_2 \cdots A_p,$$

where u is an integer, and $A_1 \cdots A_p$ is a normal sequence in the band generators.

Now the benefit of the band generators is apparent, as the n-th Catalan number is far smaller than the number of permutations in S_n, and thus there are fewer different simple words that may appear in the normal form. The band-generator form has been used for greater efficiency in some practical implementations of braid groups [8]. In particular, the smaller number of potential factors allows the Conjugacy problem to be solved much more quickly in this form, although it is still exponential. See section 2.4 for more discussion.

2.3. **Representations of the Braid Group.** The reducible Burau representation $\rho : B_n \to GL(n-1, \mathbb{Z}[t, t^{-1}])$ was discovered shortly after the definition of the braid groups, and is can be naturally described topologically. For the present context, an algebraic description is more helpful, and it is given in terms of the Artin generators by

$$\rho(\sigma_i)(t) := \begin{cases} \begin{pmatrix} -t & 1 \\ 0 & 1 \end{pmatrix} \oplus I_{n-3} & \text{if } i = 1, \\ I_{i-2} \oplus \begin{pmatrix} 1 & 0 & 0 \\ t & -t & 1 \\ 0 & 1 & 0 \end{pmatrix} \oplus I_{n-i-2} & \text{if } 1 < i < n-1, \\ I_{n-3} \oplus \begin{pmatrix} 1 & 0 \\ t & -t \end{pmatrix}. \end{cases} \tag{2.10}$$

These matrices satisfy the braid relations (2.1), and the image is the set of *Burau matrices.*

For many years it was believed that this representation might be faithful, but it was recently showed by Moody, Long and others that it is unfaithful for $n \geq 5$ (see [4, 24]). This is still a useful representation, as the kernel (and thus the probability of collisions) is quite small, the matrices are sparse, and their size does not grow very quickly relative to n. These properties have been used in several proposed cryptosystems and also in some attacks.

A generalized version of the Burau representation is used for cryptography in [1], where each band in a braid is associated with a distinct color that is encoded by t_i. The crossing σ_i is mapped to $\rho(\sigma_i)(t_i)$, and for a product of braids, a permutation is required to uniquely track the relative positions and colors of the crossings.

Definition 2.7. The colored Burau group CB_n is the set

$$S_n \times GL(n-1, \mathbb{Z}[t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}]$$

together with the multiplication

$$(\pi_1, M_1)(\pi_2, M_2) := (\pi_1\pi_2, (\pi_2^{-1}(M_1))M_2), \tag{2.11}$$

where a permutation π acts on each element of a matrix M by mapping $t_i \mapsto t_{\pi(i)}$.

The colored Burau representation $\rho : B_n \to CB_n$ is then given by

$$\rho(\sigma_i) = ((i, i+1), \rho(\sigma_i)(t_i)). \tag{2.12}$$

This representation also preserves the braid relations, and is again a homomorphism with a small kernel.

The other important representation of the braid groups is due to Lawrence and Krammer. Two of the more celebrated results in algebra of recent years are the independent proofs by Bigelow [4] and Krammer [24] that this representation

$$\mathcal{K} : B_n \to GL(n(n-1)/2, \mathbb{Z}[q, t, q^{-1}, t^{-1}] \tag{2.13}$$

is faithful. If the basis elements for the module over $\mathbb{Z}[q, t, q^{-1}, t^{-1}]$ are denoted by x_{ij} for $1 \leq i < j \leq n$, then the action of the matrix linear transformation $\mathcal{K}(\sigma_i)$ is

(2.14)

$$\mathcal{K}(\sigma_i) : x_{ij} \mapsto \begin{cases} tq^2 x_{k,k+1} & \text{if} i = k,\ j = k+1, \\ (1-q)x_{i,k} + qx_{i,k+1} & \text{if} j = k,\ i < k, \\ x_{ik} + tq^{k-i+1}(q-1)x_{k,k+1} & \text{if} j = k+1,\ i < k, \\ tq(q-1)x_{k,k+1} + qx_{k+1,j} & \text{if} i = k,\ k+1 < j, \\ x_{kj} + (1-q)x_{k+1,j} & \text{if} i = k+1,\ k+1 < j, \\ x_{ij} & \text{if} i < j < k \text{ or } k+1 < i < j, \\ x_{ij} + tq^{k-i}(q-1)^2 x_{k,k+1} & \text{if} i < k,\ k+1 < j. \end{cases}$$

Such a matrix with respect to the basis x_{ij} is called the *Krammer matrix* of a braid. Many useful bounds for the powers of t and the size of the coefficients appear in [9], and are derived from basic facts about the representation proven in [24]. See section 4.3 for an explanation of why the most effective attacks on the Diffie-Hellman braid cryptosystem use this representation.

2.4. **The Word Problem and the Conjugacy Problem.** The word problem in braid groups asks whether two given words in σ_i (or any other set of generators) are equivalent as braids. This problem is easily solved, and the normal forms of sections 2.1 and 2.2 was developed to allow for the easy comparison of two arbitrary words. Each word can be converted to normal form in polynomial time, and then the uniqueness of the decomposition provides the test.

The Conjugacy problem is to find an element c such that $x = cx'c^{-1}$ when given two elements x, x' that are known to be conjugate. This problem is unsettled for braid groups, but there has been great progress in recent years in extending Garside's summit set method [18], and experimental data suggests that the problem is feasible. As of yet, there are still no provably efficient algorithms. This topic returns later in section 4.2.

3. Braid Group Cryptosystems

This section gives the basic definitions for two of the most prominent braid-based cryptosystems. Both systems were designed to take advantage of the difficulty of the Conjugacy problem, but the subtle differences between the protocols and the idealized problem make huge differences in the security, as the attacks of Section 4 demonstrate. The braid group Conjugacy problem itself may also be severely threatened by some recent methods.

3.1. **Commutator Protocols.** This key agreement protocol was proposed by Anshel, Anshel and Goldfeld in [2], and then later extended to make use of a certain key extractor in the final step [1]. In this system, an integer N is fixed, and any two parties A, B who reside in the network each have a public subgroup in B_n,

$$S =< s_1, s_2, \ldots, s_m >, \tag{3.1}$$
$$T =< t_1, t_2, \ldots, t_n >,$$

where s_i, t_j are arbitrary elements. To establish a shared secret, A first chooses a secret $a = s_{i_1} \cdots s_{i_k} \in S$, and B chooses a secret $b = t_{j_1} \cdots t_{j_\ell} \in T$. Then A and B send the sets of pairs

$$\{(t_1, at_1a^{-1}), \ldots, (t_n, at_na^{-1})\}, \tag{3.2}$$
$$\{(s_1, bs_1b^{-1}), \ldots, (s_n, bs_nb^{-1})\},$$

respectively. Now A can compute

$$(bs_{i_1}b^{-1}) \cdots (bs_{i_k}b^{-1})a^{-1} = (bab^{-1})a^{-1}, \tag{3.3}$$

and B can compute the commutator $bab^{-1}a^{-1}$ similarly. This shared secret is now used to generate a key.

The key extractor introduced in the second paper [1] makes use of the colored Burau representation (a slight modification of the original) by mapping a braid b to a pair (M, π), where M is the image of B in the Burau representation, and π is the image of b under the map $\phi : B_n \to S_n$. The shared secret commutator is calculated as above, and then mapped as described. The key is finally obtained by reducing the matrix M modulo some prime p, and plugging in the result to a key hash function.

Note that the original version of this protocol is secure in any setting where the apparent "multiple conjugacy search problem" is difficult, which regrettably does not include the braid groups! The second version was developed because of

a specific weakness in the first due to the structure of the braid groups, but the second cannot be as easily adapted to a general setting.

3.2. **Diffie-Hellman Conjugacy Protocols.** In their original paper on public-key cryptography [15], Diffie and Hellman proposed the now famous key exchange protocol that is based on the difficulty of the discrete log problem in finite fields. If g is some element of large order, then two parties A and B pick secret values a and b respectively, and transmit g^a and g^b publicly. Then both compute the shared value g^{ab}, which is protected from eavesdroppers as long as the "Diffie-Hellman problem" (which is often expected to be equivalent to the Discrete Log problem) is difficult.

Cheon and others in Ko's research group [10] had the insight that a similar procedure can be used in braid groups by taking conjugates from disjoint, commutative subgroups. For the braid group B_{2n}, consider the subgroups of lower and upper braids

$$LB_{2n} :=< \sigma_1, \ldots, \sigma_{n-1} >, \tag{3.4}$$
$$UB_{2n} :=< \sigma_{n+1}, \ldots, \sigma_{2n-1} > .$$

It is clear from (2.1) that $LB_{2n} \cong UB_{2n} \cong B_n$, and that the two subgroups commute with each other. A relatively complicated public braid $x \in B_{2n}$ is made available prior to running this protocol. Then A chooses a secret braid $a \in LB_{2n}$, and sends $y_a = axa^{-1}$ to B. Similarly, B sends $y_b = bxb^{-1}$ to A. Now both A and B compute

$$ay_ba^{-1} = abxb^{-1}a^{-1} = baxa^{-1}b^{-1} = by_ab^{-1}. \tag{3.5}$$

Many other protocols that are related to the original Diffie-Hellman procedure can also be recast in this manner; for example, the ElGamal signature scheme in braid groups has been considered [10].

4. Attacks on Braid Cryptosystems

The braid cryptosystems of the previous section ignited a great deal of research and excitement in the subject, and it quickly became apparent that they were insecure. Many attempts to modify them have also failed, and it is unclear if any braid-based systems can be safely used. This section describes the vulnerabilities of the Commutator and Diffie-Hellman protocols, and also discusses the best known attacks on the general Conjugacy problem for braid groups.

Originally it was believed that the Conjugacy problem was difficult enough in the braid groups that secure cryptosystems could be constructed. Later it was discovered that refinements to the summit set method greatly weaken the Conjugacy problem. However, even without this development there were other problems, as the given protocols rely on variant problems that are related but not necessarily equivalent to the Conjugacy problem. The Diffie-Hellman braid protocol can be attacked by using the Lawrence-Krammer representation, even though the generic Conjugacy problem cannot be in this manner. And the Commutator protocol is vulnerable to length attacks, which again are not effective on the general Conjugacy problem.

4.1. **Length attacks on the Commutator protocol.** The Commutator protocol releases multiple instances of pairs (x, axa^{-1}) for a secret value a that is taken from a set with a finite number of known generators. This set of data can be exploited in a simple way to determine a, as first observed in [22]. The following attack is not as effective for the general Conjugacy problem, as the set of generators of B_n are a much more varied set.

Recall from Theorem 2.5 and the ensuing discussion that the length function is a well-defined property of braids.

Definition 4.1. The *distance* between two braids b, c is defined by

$$d(b, c) := \text{length}(bc^{-1}).$$

This gives a nice metric on braids that satisfies the important triangle inequality

$$d(b, e) + d(c, e) \geq d(b, c). \tag{4.1}$$

Observe that the generators of the set s in the Commutator protocol should be relatively *tangled*, which means that none of them should share very many factors or combine to form simpler words. One then expects that unless s_j^{-1} is an initial factor of a, then the length of $s_j(at_ia^{-1})s_j^{-1}$ is larger than the length of at_ia^{-1} with non-zero probability, as

$$\text{length}(s_j(at_ia^{-1})s_j^{-1}) \leq 2 \cdot \text{length}(s_j) + \text{length}(at_ia^{-1}). \tag{4.2}$$

In response to this attack, the Commutator protocol was amended to use generators of small length, and to use the associated matrices from the colored Burau representation instead of just braids [1]. Then the protocol is run as before, with secret values a, b leading to a shared secret $bab^{-1}a^{-1}$. This braid is then converted to its colored Burau matrix and permutation pair $\rho(bab^{-1}a^{-1})$, which is then run through a *key extractor* E. The key extractor sends a braid b to

$$E(b) := (\pi_b, M_b(\tau_1, \ldots, \tau_n) \ / \ \mathbb{F}_p), \tag{4.3}$$

where $\tau_i \in \mathbb{F}_p^*$ are fixed. substitutes values in a finite field $\mathbb{F}_p$ for the color variables t_i, yielding a large, finite keyspace. The purpose of this procedure is to mask the commutator through a sort of hash function so that the generators can have small length without being easily observed in the final key. See section 4.3 for more about the security of this protocol.

4.2. **The summit set and the Conjugacy Problem.** The basis for all of the present algorithms for the Conjugacy problem is in Garside's initial work [18], where he associated a finite set of conjugates with every braid b. This set is known as the *summit set*, which was then improved (made smaller) to the *super summit set* [16], and finally the *ultra summit set* [19]. This latter is just the union of the cyclic parts of the orbits in the super summit set, which is smaller still. The super summit set $SSS(b)$ is defined to be the set of all braids $c = aba^{-1}$ for every a such that length(c) minimal. The key property is that this set is computable, which solves the Conjugacy problem, as

$$SSS(b) = SSS(c) \quad \Longleftrightarrow \quad c = aba^{-1}.$$

To compute the super summit set, use the following easily computable conjugates.

Definition 4.2. If $b = \Delta_n^u A_1 \cdots A_p \in B_n$ is in normal form, then the *cycle* and *decycle* of b are

$$\partial_+(b) := \Delta_n^u A_2 \cdots A_p \phi_n^u(A_1), \quad \partial_-(b) := \Delta_n^u \phi_n^u(A_p) A_1 \cdots A_{p-1},$$

respectively, where $\phi_n(\sigma_i) = \sigma_{n-i}$ for all i.

Remark. The cycle and decycle of b are both conjugate to b, as $\partial_+(b) = sbs^{-1}$ for $s = \Delta_n^u A_2 \cdots A_n$.

.

The algorithm of [16] relies on some simple results about the effects of repeated cycling and taking conjugates of simple words.

Proposition 4.3. *Suppose that $b \in B_n$.*

1. *If $b \notin SSS(b)$, then cycling or decycling at most $n(n-1)/2$ times yields a braid of smaller length.*
2. *If $b \in SSS(b)$, then*

$$SSS(b) \subset \{\tilde{\pi} b \tilde{\pi}^{-1} \mid \pi \in S_n\}.$$

Hence checking for the conjugacy of two braids b, b' is achieved by finding the set $SSS(b)$ and testing the membership of b'. Gebhardt showed [19] that the ultra summit set $USS(b)$ is sufficient for the same test, and is typically much smaller than $SSS(b)$ in practice, although it is not provably so. Whatever security remains in the Conjugacy problem is due to the fact that the size of the summit sets may be exponential in general, and it is difficult to obtain good bounds on the cardinality.

A related approach to the Conjugacy problem utilizes ideas from the length attack from section 4.1 (see [25] for example). Given two conjugates braids b and b', one computes a braid $cb'c^{-1}$ of minimal length, and tests whether this is equal to b, or in a slight improvement, whether the new braid differs from b by conjugation with a simple braid. Again, the length function lends itself to quite powerful methods for finding conjugates.

4.3. **Linear representations.** The linear representations of the braid groups illustrate the divide between the general Conjugacy problem and the problems that actually arise in protocol security. The previous section contains some of the best attacks for the Conjugacy problem, and while they have become increasingly effective, there is still some remaining security. There are potential uses of linear representations of braid groups to construct algorithms for this problem [11, 9, 13], but a difficulty arises in that the matrix found may not be easily invertible to a braid. This is in sharp contrast with the problems underlying the Commutator and Diffie-Hellman Conjugacy protocols, which are highly vulnerable to such attacks.

In section 4.1 the revised Commutator protocol was defined, which uses the colored Burau representation to transform braids into matrices. However, the added structure of the Burau matrices allows an attacker to solve a straightforward linear system of equations over a finite field to obtain the key [25]. The authors of the protocol were aware of this possibility, but did not realize that the special form of the matrices would make such an attack feasible. This linear algebraic attack is especially effective for the recommended short generators (which are necessary to avoid the length attack!), and to the author's knowledge there have not been any additional updated versions of the Commutator protocol.

Recall the Diffie-Hellman Conjugacy protocol from section 3.2. The secret, commuting values a, b are combined with a public braid x to compute a shared secret $abxb^{-1}a^{-1}$. The protocol is compromised if one can find the secret value from $y_a = axa^{-1}$ and $y_b = bxb^{-1}$. Cheon and Jun [9] solve this problem by using the Lawrence-Krammer representation $\mathcal{K}$ of B_n.

Let $Y_a = \mathcal{K}(y_a), Y_b = \mathcal{K}(y_b)$ be the images in the representation. Working modulo some prime p and certain irreducible polynomials in t and q, solve for the matrix A in the equations

$$\begin{aligned} Y_a A &= A Y_b \\ \mathcal{K}(\sigma_i) A &= A \mathcal{K}(\sigma_i) \quad \text{for} \quad \sigma_i \in UB_{2n}. \end{aligned} \tag{4.4}$$

The key insight in this attack is that even though the solution to the linear equations may not be $\mathcal{K}(a)$, the commutativity conditions in (4.4) allow it to serve the same purpose, as

$$A Y_b A^{-1} = \mathcal{K}(b) Y_a \mathcal{K}(b)^{-1} = \mathcal{K}(abxb^{-1}a^{-1}), \tag{4.5}$$

which can be lifted to the desired braid. This attack is effective because the matrices that arise in the representation satisfy very restrictive bounds.

4.4. **Randomness and other concerns.** Dehornoy observed [13] that great care must be taken in choosing random braids, for otherwise the normal form of a product ab may yield too much information about the individual factors a and b. A particular problem lies in picking a random sequence of simple words; then it is very possible that many of the factors are already left-weighted. Some possible solutions are to use randomizing techniques to change the presentation of a word into an equivalent form, or to insert small permutations into the middle of the word, which can greatly alter the braid. There are also other ways to rewrite braids into different forms [12].

There is also a general worry that seemingly contradictory security requirements arise from different attacks. For example, the length attack implies that the generators in the Commutator protocol should have small length [22, 1], but that makes the Conjugacy problem easier. As explained above, the Conjugacy problem can be too easy if random sequences of simple words are chosen, but it is difficult to guarantee a large length otherwise. The fact that no one has found a good balance between the different attacks should give one pause before relying on braid group cryptography.

5. Conclusion

The braid groups have seen heavy interest in recent years, and many have raised them as a new hope for future cryptography. There are good algorithms for storing elements and performing group operations, and the groups are aesthetically very appealing. The cryptosystems that have been proposed thus far have relied on the difficulty of the conjugacy problem, but the use of the super (ultra) summit sets has greatly reduced the security of these systems. Even worse, the Lawrence-Krammer representation solves the Diffie-Hellman Braid Problem in polynomial time, and the normal form techniques that were developed by Garside to solve the word problem have also found use in attacking the Commutator protocol.

One alternative is to use a root-finding problem instead of conjugacy. If $b = c^e \in B_n$, then it is known that c is a unique root [20], and it seems to be a difficult

problem to find this root. However, it is not yet known how to design a protocol that isolates this problem, and the example of [14] combines a Square-root problem with the Conjugacy problem.

After the initial optimism, there are still not any trusted protocols involving braid groups, and many simulations that have been run suggest that the Conjugacy problem is simply not difficult enough. Regardless, even if the braid groups are ultimately abandoned for cryptographic purposes, their study has lead to a great deal of progress on decision problems and protocols involving more general Artin groups, and perhaps even other non-commutative groups may find their place in cryptography.

References

[1] I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, *New key agreement protocols in braid group cryptography,* CT-RSA 2001 (San Francisco), Springer Lect. Notes in Comp. Sci., 2020 (2001) 1–15.

[2] I. Anshel, M. Anshel, and D. Goldfeld, *An algebraic method for public-key cryptography,* Math. Res. Letters **6** (1999) 287–291.

[3] E. Artin, *Theory of Braids,* Ann. of Math. **48** (1947) 101–126.

[4] S. Bigelow, *Braid groups are linear,* J. Amer. Math. Soc. **14** (2001) 471–486.

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

[6] J. Birman, K. Ko, and S. Lee, *A new approach to the word problem in the braid groups,* Adv. in Math. **139** (1998) 322–353.

[7] J. Birman, K. Koo, and S. Lee, *The infimum, supremum, and geodesic length of a braid conjugacy class,* Adv. in Math. **164** (2001) 41–56.

[8] J. Cha, J. Cheon, J. Han, K. Ko, and S. Lee, *An efficient implementation of braid groups,* AsiaCrypt 2001, 144–156, Springer Lect. Notes in Comp. Sci. 2048, 2001.

[9] J. Cheon, and B. Jun, *A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem,* CRYPTO 2003, 212–225, Springer Lecture Notes in Comput. Sci. 2729, 2003.

[10] J. Cheon, J. Han, J. Kang, K. Ko, S. Lee, and C. Park, *New public-key cryptosystem using braid groups,* CRYPTO 2000, 166–184, Springer Lect. Notes in Comp. Sci. 1880, 2000.

[11] M. Cho, D. Choi, K. Ko, and J. Lee, *New signature scheme using conjugacy problem,* online http://eprint.iacr.org/2002/168/.

[12] P. Dehornoy, *A fast method for comparing braids,* Adv. in Math. **123** (1997) 205–235.

[13] P. Dehornoy, *Braid-based cryptography,* Contemp. Math. **360** (2004) 5–33.

[14] P. Dehornoy, M. Girault, and H. Sibert, *Entity authentication schemes using braid word reduction,* Proc. Internat. Workshop on Coding and Cryptography, 153–164, Versailles, 2003.

[15] W. Diffie, and M. Hellman, *New directions in cryptography,* IEEE Trans. on Inf. Theory **22** (1976) 644–654.

[16] E. Elrifai, and H. Morton, *Algorithms for positive braids,* Quart. J. Math. Oxford **45** (1994) 479–497.

[17] N. Franco and J. Gonzales-Meneses, *Conjugacy problem for braid groups and Garside groups,* J. Algebra **266** (2003) 112–132.

[18] F. Garside, *The braid group and other groups,* Quart. J. Math. Oxford **20** (1969) 235–254.

[19] V. Gebhardt, *A new approach to the conjugacy problem in Garside groups,* preprint, online http://www.arxiv.org/abs/math.GT/0306199 (2003).

[20] J. Gonzales-Meneses, *The n-th root of a braid is unique up to conjugacy,* Alg. and Geom. Topology **3** (2003) 1103–1118.

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

[22] J. Hughes and A. Tannenbaum, *Length-based attacks for certain group based encryption rewriting systems,* Inst. for Math. and its Applic. 2000, online http://www.ima.umn.edu/preprints/apr2000/1696.pdf.

[23] N. Koblitz, A. Menezes *A survey of public-key cryptosystems,* SIAM Review **46** (2004) 599–634.

[24] D. Krammer, *Braid groups are linear,* Ann. of Math. **151** (2002) 131–156.
[25] S. Lee, and E. Lee, *Potential weakness of the commutator key agreement protocol based on braid groups,* Eurocrypt 2002, Springer Lect. Notes in Comp. Sci. 2332, (2002) 14–28.
[26] W. Thurston, *On the geometry and dynamics of diffeomorphisms of surfaces,* Bull. Amer. Math. Soc. **19** (1988), 417–431.

University of Wisconsin-Madison
E-mail address: mahlburg@math.wisc.edu