Is your braid secure?

An Analysis of security protocols dependent upon Braid Group Cryptography.

By: Wendy Minton

References:

New Key Agreement Protocol in Braid Group Cryptography by Iris Anshel, Michael Anshel, Benji Fisher, and Dorian Goldfied.

A Practical Attack on Some Braid Group Based Cryptographic Primitives by Dennis Hofheinz and Rainer Steinwandt.

Summary of Presentation

- High Level View of the Commutator Key Agreement Protocol
  - Step-by-step exploration of its details
  - Formal protocol
- Conjugacy Problem
- Security of Commutator Key Exchange Protocol
  - Known weaknesses
  - How paper one compensates
- Attack of Commutator Key Exchange Protocol
  - Re-evaluate security

High Level: Keys

- Each agent has a set of data associated with them from which private, public and session keys are derived.
  - An agent's private key is an element of their associated set
  - A pubic key is formed by combining the private key of one agent with an element from the set of another agent
    - A session key is derived from the public keys of the two agents involved in the protocol run.
- As elements of another another agents sets are needed in key derivation, these sets of data must be publicly available to all agents within the system.

Commutator Key Agreement Protocol

The following public information is needed for key derivation:

$$let\ N \in \mathbb{Z},\ p \in \mathbb{Z}\ such\ that\ 6 < N < p$$

$$E : B_{(N+1)} \rightarrow K_N, p$$

$$S_A = \left\{ a_{1,} a_{2,} \ldots, a_m \right\} \in B_{(N+1)}$$

$$S_B = \left\{ b_{1,} b_{2,} \ldots, b_n \right\} \in B_{(N+1)}$$

Commutator Key Agreement Protocol

Secret Keys:

- Alice
  - $x \in S_A$
- Bob
  - $y \in S_B$

Public Keys:

- Alice
  - $PK = \left\{ x^{-1} b_i x \middle| b_i \in S_B \right\}$
- Bob
  - $PK = \left\{ y^{-1} a_i y \middle| a_i \in S_A \right\}$

Shared Information:

$$E(x^{-1} y^{-1} xy)$$

High Level: Protocol

The function E, as shown on the previous slide, is called the key extractor function. This function will be explained in more detail later in the presentation. For know, keep in mind that all data transmitted during the course of the protocol instance is first given as input to the key extractor function. This function alters the state of the data for efficiency,and security purposes.

Commutator Key Agreement Protocol

$$A \rightarrow B : \{PKA\}$$
$$B \rightarrow A : \{PKB\}$$

Alice and Bob share their public keys, which is the set described on the previous slide.

They then, from the keys, are able to extract each others private keys, and now both known the session key.

$$A \rightarrow B : \{M\}_{E(X^{-1}Y^{-1}XY)}$$
$$B \rightarrow A : \{M\}_{E(X^{-1}Y^{-1}XY)}$$

Commutator Key Agreement Protocol

At this point, both A and B have each other's sets and secret keys. They now derive first their public keys, and then, the session key that will be used to encrypt all messages from this point forward.

Concepts to Formalize

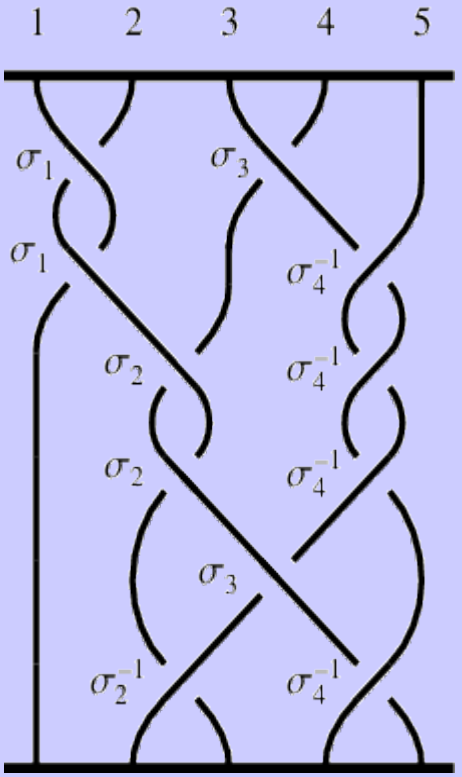
- What's B_{N+1}?
- What's $E : B_{N+1} \rightarrow K_{N,p}$?
- How can anything be kept secret if all the information needed to generate the keys is transmitted unencrypted?
- What is a conjugate?
- How secure is this protocol, that depends entirely upon its encryption scheme?

Math Time!!

Step One: The Artin Braid Group

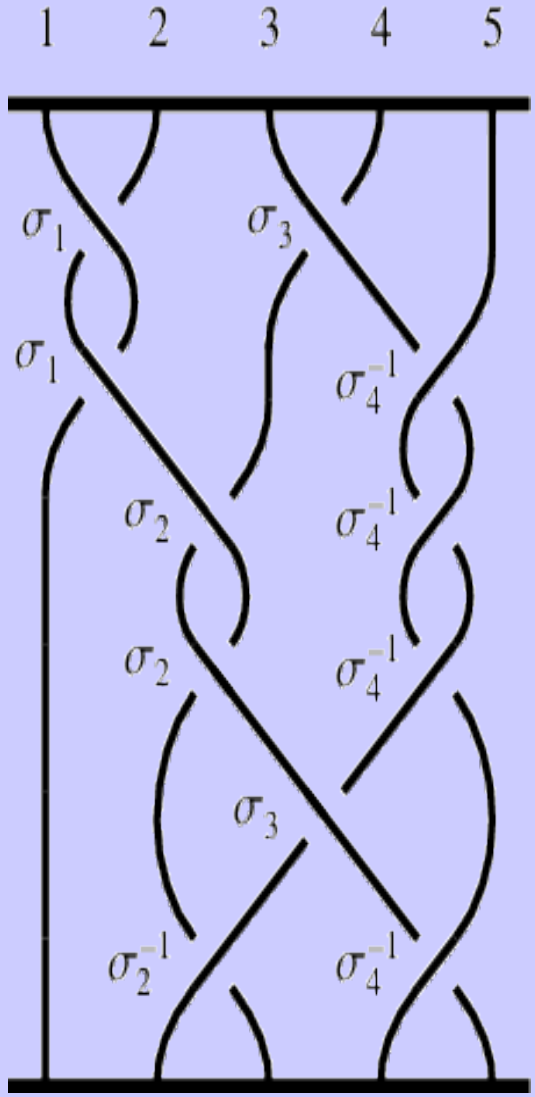
Hair braid:

Artin Braid:



Step One: The Artin Braid Group

A mathematical braid is a collection of *n* strands, each labeled with an index that corresponds to its current positioning from the left-most side of the braid. Note that the index of a particular strand changes as it is “braided”.

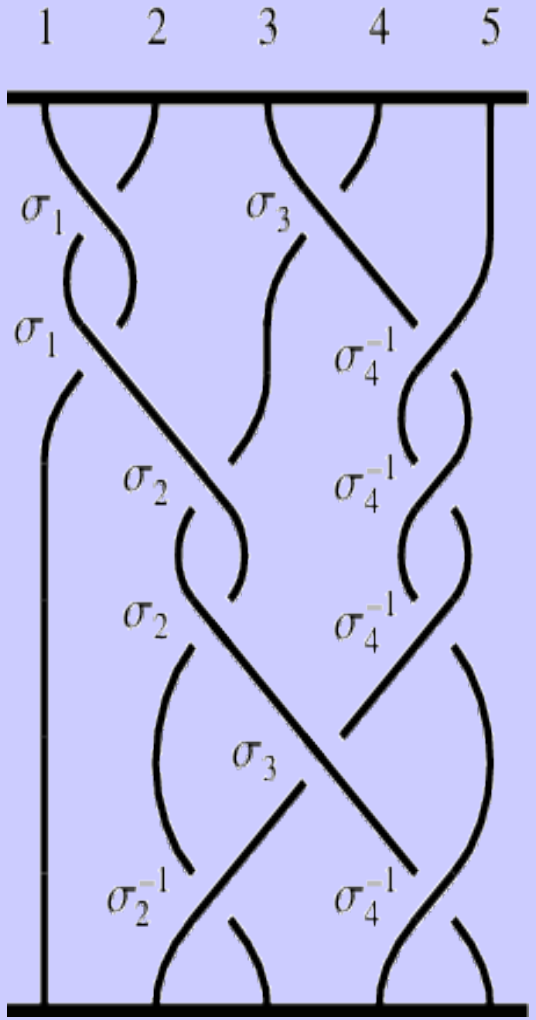


Step One : The Artin Braid Group

Braids are represented by a permutation on their crossings. This representation is called a ***braid word.***

The braid word for the braid on the right is:

$$\sigma_1\sigma_3\sigma_1\sigma_4^{-1}\sigma_2\,\sigma_4^{-1}\sigma_2\sigma_4^{-1}\,\sigma_3\sigma_2^{-1}\sigma_4^{-1}$$



Step One: The Artin Braid Group

Definition: Group

Let G be a non-empty set together with a binary operation. We say G is a group under this operation iff:

1.) Associativity holds
2.) An identity exists
3.) Each element in the set has an inverse

Step One: The Artin Braid Group

For me to claim that these braids are actually a mathematical group, I have to show the following:

1.) A binary operation exists
2.) The operation is associative
3.) An identity exists
4.) Inverses exists

Stay tuned for the dreaded “Proof by Hand Waving” technique!

Binary Operation

I need some chalk...

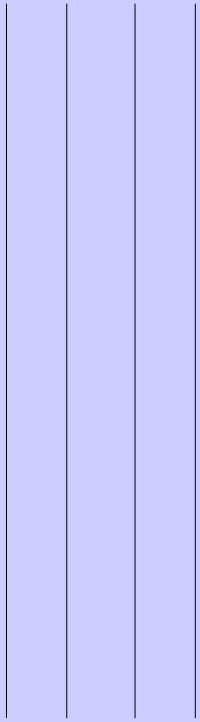
Associativity

So, to “multiply” two braids, you pretty much just stack them one on top of the other. The order in which you stack them has no bearing on the final “product”.

You may be able to create an example where the stacking seems to create different braids, but there exists certain way in which you can alter the representation, and not change the underlying braid. These are called **Markov Moves**. As an FYI, these moves are essentially a formalizing of methods for **planar isotopy** on braids.

Identity of Artin Braid Group

The identity of this group is a set of un-braided strands, as shown:



Identity for B_4

Inverses

The inverse of a braid is another braid, such that when they are stacked on top of each other, and after Markov Moves have been performed, the result is the identity.

I need the chalk again!

Step Two: Key Extractor

The Commutator Key Agreement Protocol's central component is its key generation algorithm, which has been shown to be very efficient. The algorithm is just a function, which takes as its input a braid, and outputs an element of another group.

So, the key extractor algorithm is simply a mapping from one group to another. Or, for all you mathematically savvy people, the key extractor is a **Homomorphism!**

Homomorphism

A homomorphism is an operation preserving mapping from one group to another. Formally:

$$Let\ H : G_1 \rightarrow G_s\ such\ that\ H(ab) = H(a)H(b) \forall\ a, b \in G_1$$

So, for our purposes, the key extractor algorithm takes as its input a braid word, and produces an element of the Colored Burau Group. All elements of the latter group can then be treated pretty much like the elements of the Artin Group, as “multiplying” braids and then performing the key extraction is the same as performing the extraction on the braids, and then “multiplying” them.

Step Three: Colored Burau Group

- In the Commutator Key Agreement Protocol, the Colored Burau Group acts as the **Key Space.**
- Denoted $\mathbf{K_{N,p}}$
  - N = length of the braid words used by the protocol
  - p is the prime chosen by the protocol
- How is this group different from the Artin Braid Group?
  - Increases implementation efficiency
  - Represents braid words as a matrix and a permutation.

Step Three: Colored Burau Group

- Represents a braid word as a set of matrices
  - Each matrix represents one crossing
  - The matrices are very sparse, allowing for fast manipulation
  - This new representation contains all the same information of the braid words, but in a new way. This rewriting of the braid word makes it very difficult to determine the original braids used in the key.
  - The rewriting also reduces braid words to their canonical form which has been shown to be more resilient to attacks.

Step Four : Two Subgroups

For a two-party key agreement, we need two subgroups of the Artin Braid group, one for Alice and one for Bob.

$$\text{Let } SA = \langle a1, a2, \ldots, am \rangle$$

$$SB = \langle b1, b2, \ldots, bn \rangle$$

The braids within these two subgroups are publicly known, and will be used in public-key generation.

Step Four: Subgroups

Definition: If a subset H of a group G is itself a group under the operation of G, we say H is a subgroup of G.

So, for S_A and S_B, we know the following:

1.) They are closed under braid multiplication

2.) Braid multiplication is associative

3.) There exists an identity

4.) All elements within S_A and S_B have an inverse

Commutator Key Agreement Protocol

Public Information:

$$let\ N \in \mathbb{Z},\ p \in \mathbb{Z}\ such\ that\ 6 < N < p$$

$$E : B_{(N+1)} \rightarrow K_N, p$$

$$S_A = \left\{ a_{1,} a_{2,} \ldots, a_m \right\} \in B_{(N+1)}$$

$$S_B = \left\{ b_{1,} b_{2,} \ldots, b_n \right\} \in B_{(N+1)}$$

Commutator Key Agreement Protocol

Secret Keys:

- Alice
  - $x \in S_A$
- Bob
  - $y \in S_B$

Public Keys:

- Alice
  - $PK = \left\{ x^{-1} b_i x \middle| b_i \in S_B \right\}$
- Bob
  - $PK = \left\{ y^{-1} a_i y \middle| a_i \in S_A \right\}$

Shared Information:

$$E(x^{-1} y^{-1} xy)$$

Commutator Key Agreement Protocol

$$A \rightarrow B : \{PKA\}$$
$$B \rightarrow A : \{PKB\}$$

Alice and Bob share their public keys, which is the set described on the previous slide.

They then, from the keys, are able to extract each others private keys, and now both known the session key.

$$A \rightarrow B : \{M\}_{E(X^{-1}Y^{-1}XY)}$$
$$B \rightarrow A : \{M\}_{E(X^{-1}Y^{-1}XY)}$$

Step Five: Conjugates

Definition: Conjugate

$$\forall\, b_i \in B_N \exists\, b_j, x \in B_N$$

$$such\ that\ b_i = x^{-1} b_j x$$

Importance of Conjugacy Problem

- In the protocols, agents share their primary keys, which is a set conjugate braids.
- For example, Alice sends Bob:

$$PK = \left\{ x^{-1} b_i x \middle| b_i \in S_B \right\}$$

- Since Bob knows all b_i in S_B, he can use that knowledge to determine the element *x*
  - When Bob determines what x is, we say that Bob is solving the conjugacy problem.
  - In the protocol, the only way that x can be determined is if this conjugacy problem is solved.

Importance of Conjugacy Problem

- Remember, public keys are not transmitted as is but are "rewritten" into another form before they are transmitted.
  - The key extractor function thus protects the private keys, as an attacker who knows the two agents sets of data could conceivably determine the private keys.
- Once Alice and Bob exchange their public keys, they compute the shared secret. This secret, which serves as the session key, is also a conjugate.

$$x^{-1} y^{-1} xy$$

Step Five: Conjugates

- Transmission of public keys and session key computation, relying upon the **Conjugacy Problem** to preserve the integrity of their interaction**.**
- To determine the session key, an attacker must:
  - Determine both agents' private keys
    - Solve an instance of the conjugacy problem on both agents' public keys
- The difficulty of this problem depends upon the group. For braid groups, the problem has not yet been solved within polynomial time (in relation to the length of the braid word).

Step Five: Conjugates

Anshel presents a brute force attack that can solve the conjugacy problem, but it requires testing every braid of a certain length within a group. He defends his protocol by requiring all braid words used to have length > 6, and for all parties involved to use different sized subgroups. He claims this is sufficient to render the brute force attack infeasible.

Security Analysis

The Commutator Key Agreement Protocol relies upon the computational difficulty of the Conjugacy Problem and Commutator Problem.

While it is not known exactly how difficult these problems are, current research suggests that no polynomial algorithm will be found.

The authors of the paper admit that if such an algorithm where to be found, their protocol's claim to security would collapse, as attackers would easily be able to determine the private keys, and thus gain the session key.

Security Analysis

Some other things to consider:

- A specific braid word can be conjugate to more then one braid.
  - So, even if an attacker is able to solve the conjugacy problem, the resulting braid may or may not be an agents secret key.
- Braids have more then one braid word corresponding to them
  - This is due to the existence of Markov Moves

Attack

Paper Two gives an algorithm that, using a heuristic approach, can attack the Commutator Key Agreement Protocol with a VERY High Success rate.

Success Rate:

99%

Show the Algorithm

Attack

- Paper two exploits the fact that more then one braid can be conjugate to another braid.
  - An algebraist can explain that all braids that are conjugate to a particular braid are of a certain form.
  - Attackers can thus use this common form to significantly reduce the number of cases that must be checked in a brute force attack.
    - That's what paper two does, it exploits what they call “conjugacy classes”.
    - They also simplify things and work with the permutation representation of braids.

Attack

The Attack does not solve the conjugacy problem, but uses brute force. The length of the braid words was the length suggested by the authors of paper one. Paper two goes on to examine the use of longer braid words, but their success rates remain within the 90% range.

A similar protocol, which essentially conjugates the secret key twice, instead of once, did better. Success rates were only around 78%, but that is still much too high for the protocol to be of much use.