Thompson’s group $F$ and group-based cryptography

Jennifer Taback
Bowdoin College

May 2, 2008
Attribution

This talk is based on work of

1. Vladimir Shpilrain and Alexander Ushakov, City University of New York
2. Alexei Myasnikov, McGill University
3. Francesco Matucci, Cornell University
4. Sean Cleary, Murray Elder, Jennifer Taback and Andrew Reichnitzer
The goal of this talk is to introduce some of the main ideas of group-based cryptography, and to highlight one of my favorite groups, Thompson’s group $F$, and it’s 15 minutes of fame in this field.

Specifically I will touch on:

1. How decision problems are used in cryptography.
2. How to analyze the security of a group-based cryptosystem, in particular, why studying the probability that a randomly chosen subgroup has a particular form is important.
3. How to make precise that notion of choosing a random subgroup of a group.
4. Thompson’s group $F$, and why it is so interesting.
5. Some examples of group based cryptosystems. I will introduce one due to Shpilrain and Ushakov which is implemented using Thompson’s group $F$.
6. How one might attempt to attack a cryptosystem. I will discuss a length-based attack given by Shpilrain and Ushakov to illustrate how this can be done. Then I will describe an alternate approach to breaking this system.
The Key Distribution Problem

The central problem in cryptography is the key distribution problem.

Suppose that Alice wants to send a secret message to Bob.

They must securely agree on a secret key which will allow them to encrypt their messages.

Their algorithm must ensure that if Eve intercepts their communication, she cannot recover the common key and decrypt the message.
Diffie-Hellman key exchange protocol

The main idea

Use a “one-way function”, that is, a function which is easy to calculate and hard to undo.

A simple one-way function

Given a positive integer $m$ and integers $a$ and $x$, it is easy to compute

$$a^x \pmod{m}$$

The discrete logarithm problem

Given $a^x \pmod{m}$, as well as $a$ and $m$, the problem of computing $x$ is called the discrete logarithm problem. There is no efficient algorithm to do this.

Diffie-Hellman, 1976

The Diffie-Hellman key exchange protocol will allow Alice and Bob to securely exchange a least residue mod $p$ (i.e. an element of $\mathbb{Z}_p$), which they can use as the key.

Group based cryptography asks: why is this group different from all other groups? Answer: it’s not. Can we make a key exchange protocol which allows Alice and Bob to securely exchange a group element?
Diffie-Hellman key exchange protocol

Public data
Choose $m \in \mathbb{Z}^+$ and $y \in \mathbb{Z}$.

Alice
Alice picks a private key $A$ and sends $y^A(\text{mod } m)$ to Bob.

Bob
Bob picks a private key $B$ and sends $y^B(\text{mod } m)$ to Alice.

The common key
Alice computes $(y^B)^A = y^{AB}(\text{mod } m)$ and Bob computes $(y^A)^B = y^{AB}(\text{mod } m)$. 
Decision problems in group theory

Question
Can we replace the discrete logarithm problem with a sufficiently “hard” problem form geometric group theory and increase security?

The word problem
Given a finitely generated group $G$ and a word $w \in G$, decide whether or not $w$ is the identity word in $G$.

The conjugacy problem
Given two elements $a, b \in G$, decide whether there is some $x \in G$ so that $a^x = xax^{-1} = b$.
Solvable conjugacy problem implies solvable word problem.
Using conjugacy in the key exchange protocol

**Conjugacy Search Problem**

Let $G$ be a group. Given $a, b \in G$ and the information that $a^x = xax^{-1} = b$ for some $x \in G$, find at least one such element $x$.

This problem can replace the discrete logarithm problem in standard key exchange protocols.

Two protocols based on the conjugacy search problem.

2. Anshel et al, may be vulnerable to length based attacks, but a group theoretic attack via the conjugacy search problem actually relies on a much more difficult problem.

Let's see how the Ko-Lee protocol works.
The Ko-Lee key exchange protocol relying on the conjugacy search problem

Let \( G \) be a group with a solvable word problem.

**Public data**
Choose an element \( a \in G \).

**Private data**
Alice picks a private key \( x \) and sends \( a^x = xax^{-1} \) to Bob.
Bob picks a private key \( y \) and sends \( a^y = yay^{-1} \) to Alice.

**The common key**
Alice computes \((a^y)^x = a^{yx}\) and Bob computes \((a^x)^y = a^{xy}\). Choose \( x \) and \( y \) from a set of commuting elements to generate a common key.

Goal: Find a group with a large set of commuting elements, solvable word problem and hard conjugacy problem.

**Proposed Platform**
The Braid Groups \( B_n \) on \( n \) strands.
Choosing a group for implementation

Are the braid groups a good choice for implementing the Ko-Lee protocol?

A practical attack on the Ko-Lee protocol implemented on the braid groups was presented by Myasnikov, Shpilrain and Ushakov.

Using a combination of algorithms specific to braid groups, they ran 2466 experiments, each allotted 150 minutes (deemed a “reasonable” length of time) and had success in 2378 of them, yielding a 96.43% success rate.

Should we implement other protocols using braid groups?
Choosing a group for implementation

Are the braid groups a good choice for implementing the Ko-Lee protocol?

A practical attack on the Ko-Lee protocol implemented on the braid groups was presented by Myasnikov, Shpilrain and Ushakov.

Using a combination of algorithms specific to braid groups, they ran 2466 experiments, each allotted 150 minutes (deemed a “reasonable” length of time) and had success in 2378 of them, yielding a 96.43% success rate.

Should we implement other protocols using braid groups?

Bad Choices

- Groups with fast algorithms for computations within the group.
- Groups with unsolvable word problem (required for decryption by legitimate parties).

Good choices

- Groups which are not “similar” to any groups on the bad list.
- Perhaps the braid groups? (often proposed)
### Random subgroups of groups

<table>
<thead>
<tr>
<th>What is the probability that a randomly chosen subgroup of a finitely generated group has a particular form?</th>
</tr>
</thead>
<tbody>
<tr>
<td>What does this mean?</td>
</tr>
<tr>
<td>How do we formalize the idea of choosing a random subgroup?</td>
</tr>
<tr>
<td>We will consider our subgroups up to isomorphism.</td>
</tr>
</tbody>
</table>

### Consequences for group-based cryptography

If with probability one, a randomly chosen subgroup is the free group, then we can just pretend that our group is a free group, and apply free group algorithms to attack group-based cryptosystems.

Thus good platforms for these group-based cryptosystems are groups in which:

- a randomly chosen isomorphism class of subgroup is not free with probability one.
- perhaps a randomly chosen isomorphism class of subgroup is not free with probability $p$ for some fixed $p < 1$.
- there is no isomorphism class of subgroup which is chosen with probability one.
Making the notion of a random subgroup of $G$ precise

Choose a set $X$ of representatives for elements of $G$, and define a notion of integer size. For example, the size of a representative might be its length with respect to a fixed finite generating set.

Let $X_k$ be the set of unordered $k$-tuples of representatives $x \in X$. Then each member of $X_k$ corresponds to a $k$-generated subgroup of $G$.

Use the notion of size of an element to define the size of a $k$-tuple. For example, the size of the $k$-tuple could be the maximum size of the sizes of its components, or the sum of the sizes of its components.

Divide $X_k$ into spheres, letting the sphere $\text{Sph}_k(n)$ of radius $n$ consist of all members of $X_k$ of size $n$. This is called a stratification of $X_k$. 

Thompson’s Group $F$ and group-based cryptography
Making the notion of a random subgroup of $G$ precise

Choose a set $X$ of representatives for elements of $G$, and define a notion of integer size. For example, the size of a representative might be its length with respect to a fixed finite generating set.

Let $X_k$ be the set of unordered $k$-tuples of representatives $x \in X$. Then each member of $X_k$ corresponds to a $k$-generated subgroup of $G$.

Use the notion of size of an element to define the size of a $k$-tuple. For example, the size of the $k$-tuple could be the maximum size of the sizes of its components, or the sum of the sizes of its components.

Divide $X_k$ into spheres, letting the sphere $\text{Sph}_k(n)$ of radius $n$ consist of all members of $X_k$ of size $n$. This is called a stratification of $X_k$.

The asymptotic density of a subset $T$ in $X_k$ is defined to be the limit

$$\lim_{n \to \infty} \frac{|T \cap \text{Sph}_k(n)|}{|\text{Sph}_k(n)|}$$

if this limit exists, where $|T|$ denotes the size of the set $T$. 
Making the notion of a random subgroup of $G$ precise

The \textit{asymptotic density} of a subset $T$ in $X_k$ is defined to be the limit

$$
\lim_{n \to \infty} \frac{|T \cap \text{Sph}_k(n)|}{|\text{Sph}_k(n)|}
$$

if this limit exists, where $|T|$ denotes the size of the set $T$.

Let $T_H$ be the set of $k$-tuples that generate a subgroup of $G$ isomorphic to some particular subgroup $H$.

1. If the density of $T_H$ is positive we say that $H$ is \textit{visible} in the space of $k$-generated subgroups of $G$.

   We call the set of all visible $k$-generated subgroups of $G$ the \textit{$k$-subgroup spectrum}, denoted by $\text{Spec}_k(G)$.

2. If the density of $T_H$ is one, we say that $H$ is \textit{generic}.

3. If this density is zero we say that $H$ is \textit{negligible}.
Random subgroups of braid groups

Group Presentation
Generators: \( \{\sigma_1, \cdots, \sigma_{n-1}\} \) where in \( \sigma_i \) the only crossing is of the \( i \) \( th \) strand over the next strand.
Relations:
1. \( \sigma_i \sigma_j = \sigma_j \sigma_i \) for \( |i - j| \geq 2 \)
2. \( \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \).

Random subgroups of \( B_n \)
The pure braid group \( PB_n \) is the kernel of the canonical map \( B_n \to S_n \). It has finite index in \( B_n \).
A \( k \)-tuple of group elements has the free basis property in \( G \) if it freely generates a free subgroup of \( G \).

Theorem[Myasnikov-Ushakov]: The free basis property is generic in the pure braid group \( PB_n \) for \( n \geq 3 \).
They conjecture that this is true in \( B_n \) as well.
Thompson’s group $F$

### Algebraic Interpretation

**Infinite Presentation**

\[
\langle x_0, x_1, \cdots \mid x_i^{-1} x_j x_i = x_{j+1}, \ i < j \rangle
\]
Thompson’s group $F$

**Algebraic Interpretation**

Infinite Presentation

\[ \langle x_0, x_1, \cdots | x_i^{-1}x_jx_i = x_{j+1}, \ i < j \rangle \]

**Geometric Interpretation**

Pairs of finite rooted binary trees
**Algebraic Interpretation**

Infinite Presentation
\[
\langle x_0, x_1, \cdots | x_i^{-1} x_j x_i = x_{j+1}, \ i < j \rangle
\]

**Geometric Interpretation**

Pairs of finite rooted binary trees

**Analytic Interpretation**

Piecewise-linear homeomorphisms of \([0, 1]\), such that

1. all slopes are powers of 2
2. all break points have coordinates in the set of dyadic rationals
The generators $x_0$ and $x_k$ of $F$ as piecewise-linear homeomorphisms.
Note that Thompson’s group $F$ has no free subgroups.

**Theorem**

For any fixed $k$, the subgroup spectrum $\text{Spec}_k(G)$ (with respect to the “Max” stratification) contains many isomorphism classes of subgroups with small, but positive, asymptotic density.

Thompson’s group $F$ provides the first example of

1. a group without a generic type of subgroup.

2. a group with *persistent* subgroups, that is, they appear in $\text{Spec}_k(G)$ for all sufficiently large $k$. Persistent subgroups of $F$ include $F^n \times \mathbb{Z}^m$, for $n, m \geq 0$ and $F^n \wr \mathbb{Z}$.

3. a group with subgroups that disappear from the subgroup spectrum as $k$ increases.

Does this make Thompson’s group a good candidate for a group-based cryptosystem?

This is joint work with Sean Cleary, Murray Elder and Andrew Reichnitzer.
The decomposition Problem

Replaces the conjugacy search problem in key exchange protocols.

Problem

Given an element \( w \) of a group \( G \), a subset \( A \subseteq G \) and an element \( x \cdot w \cdot y \), find elements \( x', y' \in A \) so that \( x' \cdot w \cdot y' = x \cdot w \cdot y \).

Special case of the decomposition problem

\( w = 1 \) and \( G = \mathbb{Z}_p^* \) \( \Rightarrow \) RSA factorization problem.

Revised decomposition problem

Given an element \( w \) of a group \( G \), and subsets \( A, B \subseteq G \) and an element \( x \cdot w \cdot y \), find elements \( x' \in A \) and \( y' \in B \) so that \( x' \cdot w \cdot y' = x \cdot w \cdot y \).

Shpilrain-Ushakov Key Exchange Protocol

Goal: Impose additional restrictions on the subgroups \( A \) and \( B \) of \( G \) needed for the decomposition problem to ensure a common key.
Shpilrain-Ushakov Key Exchange Protocol

Public Data

A group $G$, an element $w \in G$, and two subgroups $A, B$ of $G$ with the property that $ab = ba$ for all $a \in A$ and $b \in B$.

Private Data

Alice chooses $a_1 \in A$ and $b_1 \in B$.
Bob chooses $a_2 \in A$ and $b_2 \in B$.

Computing the shared key

Alice sends $a_1 wb_1$ to Bob.
Bob sends $b_2 wa_2$ to Alice.
### Shpilrain-Ushakov Key Exchange Protocol

#### Public Data
A group $G$, an element $w \in G$, and two subgroups $A, B$ of $G$ with the property that $ab = ba$ for all $a \in A$ and $b \in B$.

#### Private Data
- Alice chooses $a_1 \in A$ and $b_1 \in B$.
- Bob chooses $a_2 \in A$ and $b_2 \in B$.

#### Computing the shared key
- Alice sends $a_1wb_1$ to Bob.
- Bob sends $b_2wa_2$ to Alice.
- The key is then $K = b_2(a_1wb_1)a_2 = a_1(b_2wa_2)b_1$.
## Shpilrain-Ushakov Key Exchange Protocol

### Public Data

A group $G$, an element $w \in G$, and two subgroups $A, B$ of $G$ with the property that $ab = ba$ for all $a \in A$ and $b \in B$.

### Private Data

Alice chooses $a_1 \in A$ and $b_1 \in B$.

Bob chooses $a_2 \in A$ and $b_2 \in B$.

### Computing the shared key

Alice sends $a_1 wb_1$ to Bob.

Bob sends $b_2 wa_2$ to Alice.

The key is then $K = b_2(a_1 wb_1)a_2 = a_1(b_2 wa_2)b_1$.

### Implementation

How do we choose a group $G$ to implement this protocol? We need a simple way of finding commuting subgroups $A$ and $B$ of $G$. Is this protocol vulnerable to length based attack?
For $k \in \mathbb{Z}^+$, define $\phi_k = 1 - \frac{1}{2^{k+1}}$

From the definition of $x_k$, we see that $x_k^{-1}([\phi_k, 1]) = [\phi_{k+1}, 1] \subset [\frac{3}{4}, 1]$.

For $t \in [\phi_k, 1]$, we have
$$\frac{d}{dt} x_0 x_k^{-1}(t) = x_0'(x_k^{-1}(t))(x_k^{-1})'(t) = 2 \cdot \frac{1}{2} = 1$$

Figure: Elements of the form $x_0$ and $x_k$. 
Implementing the protocol with Thompson’s group $F$

**Elements of the form $x_0 x_k^{-1}$**

For $t \in [\phi_k, 1]$, the element $x_0 x_k^{-1}$ is the identity.

**The group $A$**

Let $A_s$ be the group generated by the elements $\{x_0 x_1^{-1}, x_0 x_2^{-1}, \cdots, x_0 x_s^{-1}\}$.

**The group $B$**

Let $B_s$ be the group generated by the elements $\{x_{s+1}, x_{s+2}, \cdots, x_{2s}\}$.

**Figure:** An example of an element of $A_s$ and one of $B_s$. 
The Shpilrain-Ushakov Lemma

Lemma (Shpilrain-Ushakov)

For every fixed \( s \in \mathbb{N} \), and elements \( a \in A_s \) and \( b \in B_s \), we have \( ab = ba \).

Proof:

Figure: An example of an element of \( A_s \) and one of \( B_s \).

Figures drawn by Francesco Matucci.
Shpilrain and Ushakov suggest the following parameters for practical key generation:

1. Select randomly and uniformly: $s \in [3, 8]$ and $M \in \{256, 258, \ldots, 318, 320\}$

2. Select the word $w$ as a product of length $M$ of generators in the set $\{x_0, x_1, x_2, \ldots, x_{s+2}\}$ and their inverses

3. Select $a_1, a_2 \in A_s$ as products of length $M$ of the generators $\{x_0x_1^{-1}, x_0x_2^{-1}, \ldots, x_0x_s^{-1}\}$ and their inverses

4. Select $b_1, b_2 \in B_s$ as products of length $M$ of the generators $\{x_{s+1}, x_{s+2}, \ldots, x_{2s}\}$ and their inverses
Is length-based attack a concern for this protocol, using Thompson’s group?

**Recall: Revised decomposition problem**

Given an element $w$ of a group $G$, and subsets $A, B \subseteq G$ and an element $x \cdot w \cdot y$, find elements $x' \in A$ and $y' \in B$ so that $x' \cdot w \cdot y' = x \cdot w \cdot y$.

**Construct a directed graph $Γ$**

The vertices of $Γ$ are the elements of Thompson’s group $F$. There is an edge from $v_1$ to $v_2$ labeled $(w_1, w_2)$ if $v_2 = w_1 v_1 w_2$ where

1. $(w_1, w_2) = (w_1, 1)$ and $w_1 \in A_s$, or
2. $(w_1, w_2) = (1, w_2)$ and $w_2 \in B_s$
Constructing a length-based attack

Is length-based attack a concern for this protocol, using Thompson’s group?

Recall: Revised decomposition problem

Given an element \( w \) of a group \( G \), and subsets \( A, B \subseteq G \) and an element \( x \cdot w \cdot y \), find elements \( x' \in A \) and \( y' \in B \) so that \( x' \cdot w \cdot y' = x \cdot w \cdot y \).

Construct a directed graph \( \Gamma \)

The vertices of \( \Gamma \) are the elements of Thompson’s group \( F \). There is an edge from \( v_1 \) to \( v_2 \) labeled \((w_1, w_2)\) if \( v_2 = w_1v_1w_2 \) where

1. \((w_1, w_2) = (w_1, 1)\) and \( w_1 \in A_s \), or
2. \((w_1, w_2) = (1, w_2)\) and \( w_2 \in B_s \)

Breaking Alice’s key is equivalent to finding a path from \( w \) (the initial base word) to \( w' = a_1wb_1 \) in \( \Gamma \).
Algorithm for a length-based attack

Input: The original public word $w$ and the word $w' = a_1 wb_1$ transmitted by Alice.
Output: A pair of words $x_1 \in A_s$ and $x_2 \in B_s$ so that $w' = x_1 wx_2$.

Initializations for Algorithm: Let $S_w = \{w\}$, $S_{w'} = \{w'\}$, $M_w = \emptyset$, $M_{w'} = \emptyset$.
Let $S_A$ and $S_B$ denote the sets of generators for $A_s$ and $B_s$ respectively.
Algorithm for a length-based attack

Input: The original public word $w$ and the word $w' = a_1 wb_1$ transmitted by Alice.
Output: A pair of words $x_1 \in A_s$ and $x_2 \in B_s$ so that $w' = x_1wx_2$.

Initializations for Algorithm: Let $S_w = \{w\}$, $S_{w'} = \{w'\}$, $M_w = \emptyset$, $M_{w'} = \emptyset$.
Let $S_A$ and $S_B$ denote the sets of generators for $A_s$ and $B_s$ respectively.

A. Find a shortest word $u \in S_w - M_w$.
B. Multiply $u$ by elements of $S_A^{\pm 1}$ on the left and by elements of $S_B^{\pm 1}$ on the right, and add each result into $S_w$ with edges labelled accordingly.
C. Add $u$ into $M_w$.
D. Perform steps A-C with $S_w$ and $M_w$ replaced by $S_{w'}$ and $M_{w'}$.
E. If $S_w \cap S_{w'} = \emptyset$, then goto step A.
F. If there exists $\overline{w} \in S_w \cap S_{w'}$, then find a path in $S_w$ from $w$ to $\overline{w}$ and a path in $S_{w'}$ from $\overline{w}$ to $w'$. Concatenate and output the label of the result.

In trials, the success rate of the length-based attack was zero.
Algorithm for a length-based attack

Input: The original public word $w$ and the word $w' = a_1 wb_1$ transmitted by Alice.
Output: A pair of words $x_1 \in A_s$ and $x_2 \in B_s$ so that $w' = x_1 wx_2$.

Initializations for Algorithm: Let $S_w = \{w\}$, $S_{w'} = \{w'\}$, $M_w = \emptyset$, $M_{w'} = \emptyset$. Let $S_A$ and $S_B$ denote the sets of generators for $A_s$ and $B_s$ respectively.

A. Find a shortest word $u \in S_w - M_w$.
B. Multiply $u$ by elements of $S_A^{\pm 1}$ on the left and by elements of $S_B^{\pm 1}$ on the right, and add each result into $S_w$ with edges labelled accordingly.
C. Add $u$ into $M_w$.
D. Perform steps A-C with $S_w$ and $M_w$ replaced by $S_{w'}$ and $M_{w'}$.
E. If $S_w \cap S_{w'} = \emptyset$, then goto step A.
F. If there exists $\bar{w} \in S_w \cap S_{w'}$, then find a path in $S_w$ from $w$ to $\bar{w}$ and a path in $S_{w'}$ from $\bar{w}$ to $w'$. Concatenate and output the label of the result.

In trials, the success rate of the length-based attack was zero.
Recovering the shared secret key

Evil Eve, knowing the elements \( w, u_1 = a_1wb_1 \) and \( u_2 = b_2wa_2 \), can always recover one of Alice and Bob’s private keys.

If the graph of \( w \) is

1. below \((\phi_s, \phi_s)\) ⇒ recover Bob’s secret key.
2. above \((\phi_s, \phi_s)\) ⇒ recover Alice’s secret key.

Figure: An example of an element of \( A_s \) and one of \( B_s \).

Recall that for \( k \in \mathbb{Z}^+ \), define \( \phi_k = 1 - \frac{1}{2^{k+1}} \).
Recovering Bob’s private key

Suppose that \( w(\phi(s)) \leq \phi(s) \). In particular, \( w(t) \leq \phi_s \) for all \( t \in [0, \phi_s] \). Thus:

\[
u_2(t) = b_2 wa_2(t) = wa_2(t) \text{ for all } t \in [0, \phi_s].\]

Since Eve knows the element \( w \), she can obtain \( a_2(t) \) for all \( t \in [0, \phi_s] \). She then knows that

\[
a_2(t) = \begin{cases} w^{-1}u_2(t) & t \in [0, \varphi_s] \\ t & t \in [\varphi_s, 1]. \end{cases}
\]

Now Eve has the elements \( a_2, w \) and \( u_2 = b_2 wa_2 \) and she computes

\[
b_2 = u_2 a_2^{-1} w^{-1}
\]

thereby detecting Bob’s private keys and the shared secret key \( K \).

Using slightly more mathematics, we can recover Alice’s secret key even when \( w(\phi(s)) \leq \phi(s) \).
What type of groups should be proposed for cryptographic protocols based on the conjugacy search and related problems?

The conjugacy search problem in $G$ should be well-studied or related to a different well-known problem in mathematics.

The word problem in $G$ should have a fast (linear- or quadratic-time) solution by a deterministic algorithm. This is required for efficient common key extraction by legitimate parties.

The conjugacy search problem (or other appropriate problem) should not have a subexponential-time solution by a deterministic algorithm.

Proposing a better platform: (Less crucial, but useful.) There should be a way of “disguising” elements of $G$ via the relators so that it would be impossible to recover $x$ from $xax^{-1}$ by inspection.
References


