Cryptanalysis of the Shpilrain-Ushakov protocol in $F$

Francesco Matucci

Cornell University

June 28, 2007
1. The protocol
   - Problem and key exchange
   - The platform group and choice of parameters

2. Cryptanalysis of the protocol
   - Other representations of $F$
   - The attack and generalizations
Decomposition Problem
Decomposition Problem

The protocol is based on the Decomposition Problem:
Decomposition Problem

The protocol is based on the **Decomposition Problem**: 

Given a group \( G \), a subset \( X \subseteq G \) and \( w_1, w_2 \in G \)
Decomposition Problem

The protocol is based on the **Decomposition Problem**: 

Given a group $G$, a subset $X \subseteq G$ and $w_1, w_2 \in G$ find $a, b \in X$ such that 

$$aw_1b = w_2$$
Key Exchange Protocol
Public Data.
Key Exchange Protocol

**Public Data.** A group $G$, an element $w \in G$ and two subgroups $A, B$ of $G$ such that

$$ab = ba, \quad \forall a \in A, b \in B$$
Key Exchange Protocol

Public Data. A group $G$, an element $w \in G$ and two subgroups $A, B$ of $G$ such that

$$ab = ba, \quad \forall a \in A, b \in B$$

Private Keys.
Key Exchange Protocol

Public Data. A group $G$, an element $w \in G$ and two subgroups $A, B$ of $G$ such that

$$ab = ba, \quad \forall a \in A, b \in B$$

Private Keys.

- Alice selects $a_1 \in A, b_1 \in B$ and sends $u_1 = a_1 wb_1$ to Bob
Key Exchange Protocol

Public Data. A group $G$, an element $w \in G$ and two subgroups $A, B$ of $G$ such that

$$ab = ba, \quad \forall a \in A, b \in B$$

Private Keys.

- Alice selects $a_1 \in A, b_1 \in B$ and sends $u_1 = a_1 wb_1$ to Bob
- Bob selects $b_2 \in B, a_2 \in A$ and sends $u_2 = b_2 wa_2$ to Alice
Key Exchange Protocol

**Public Data.** A group $G$, an element $w \in G$ and two subgroups $A, B$ of $G$ such that

$$ab = ba, \quad \forall a \in A, b \in B$$

**Private Keys.**

- Alice selects $a_1 \in A, b_1 \in B$ and sends $u_1 = a_1 wb_1$ to Bob
- Bob selects $b_2 \in B, a_2 \in A$ and sends $u_2 = b_2 wa_2$ to Alice
- Alice computes $K_A = a_1 u_2 b_1 = a_1 b_2 wa_2 b_1$
Key Exchange Protocol

**Public Data.** A group $G$, an element $w \in G$ and two subgroups $A, B$ of $G$ such that

$$ab = ba, \quad \forall a \in A, b \in B$$

**Private Keys.**

- Alice selects $a_1 \in A, b_1 \in B$ and sends $u_1 = a_1 wb_1$ to Bob
- Bob selects $b_2 \in B, a_2 \in A$ and sends $u_2 = b_2 wa_2$ to Alice
- Alice computes $K_A = a_1 u_2 b_1 = a_1 b_2 wa_2 b_1$
- Bob computes $K_B = b_2 u_1 a_2 = b_2 a_1 wb_1 a_2$
Key Exchange Protocol
Key Exchange Protocol

Since $A$ and $B$ commute elementwise

$$K_A = a_1 b_2 w a_2 b_1 = b_2 a_1 w b_1 a_2 = K_B = K$$

becomes their shared secret key.
Key Exchange Protocol

Since $A$ and $B$ commute elementwise

$$K_A = a_1 b_2 w a_2 b_1 = b_2 a_1 w b_1 a_2 = K_B = K$$

becomes their shared secret key.

Eve’s Data.
Since $A$ and $B$ commute elementwise

$$K_A = a_1 b_2 w a_2 b_1 = b_2 a_1 w b_1 a_2 = K_B = K$$

becomes their shared secret key.

**Eve’s Data.** She has all the public data and the two elements $u_1, u_2$, observed during Alice and Bob’s exchange.
Thompson’s group $F$
Thompson’s group $F$

Combinatorial group theory approach:
Thompson’s group $F$

Combinatorial group theory approach:

$$F = \langle x_0, x_1, x_2, \ldots \mid x_i^{-1}x_nx_i = x_{n+1}, \forall i < n \rangle$$
Thompson’s group $F$

Combinatorial group theory approach:

$$F = \langle x_0, x_1, x_2, \ldots | x_i^{-1}x_nx_i = x_{n+1}, \forall i < n \rangle$$

Advantage:
Thompson’s group $F$

Combinatorial group theory approach:

$$F = \langle x_0, x_1, x_2, \ldots | x_i^{-1}x_nx_i = x_{n+1}, \forall i < n \rangle$$

Advantage: there are normal forms and they are fast to compute.
Normal Forms in $F$
Normal Forms in $F$

$$F = \langle x_0, x_1, x_2, \ldots | x_k^{-1} x_n x_k = x_{n+1}, \forall k < n \rangle$$
Normal Forms in $F$

\[ F = \langle x_0, x_1, x_2, \ldots | x_k^{-1} x_n x_k = x_{n+1}, \forall k < n \rangle \]

\[ x_n x_k \rightarrow x_k x_{n+1} \text{ (smaller subscripts first)} \]
Normal Forms in $F$

$$F = \langle x_0, x_1, x_2, \ldots | x_k^{-1} x_n x_k = x_{n+1}, \forall k < n \rangle$$

$$x_n x_k \rightarrow x_k x_{n+1} \text{ (smaller subscripts first)}$$

$$x_k^{-1} x_n \rightarrow x_{n+1} x_k^{-1} \text{ (positive before negative)}$$
Normal Forms in $F$

\[ F = \langle x_0, x_1, x_2, \ldots | x_k^{-1} x_n x_k = x_{n+1}, \forall k < n \rangle \]

\[ x_n x_k \rightarrow x_k x_{n+1} \quad \text{(smaller subscripts first)} \]

\[ x_k^{-1} x_n \rightarrow x_{n+1} x_k^{-1} \quad \text{(positive before negative)} \]

\[ x_n^{-1} x_k \rightarrow x_k x_{n+1}^{-1} \quad \text{(positive before negative)} \]
Normal Forms in $F$

$$F = \langle x_0, x_1, x_2, \ldots | x_k^{-1} x_n x_k = x_{n+1}, \forall k < n \rangle$$

- $x_n x_k \rightarrow x_k x_{n+1}$ (smaller subscripts first)
- $x_k^{-1} x_n \rightarrow x_{n+1} x_k^{-1}$ (positive before negative)
- $x_n^{-1} x_k \rightarrow x_k x_{n+1}^{-1}$ (positive before negative)
- $x_k^{-1} x_n^{-1} \rightarrow x_{n+1}^{-1} x_k^{-1}$ (smaller subscripts last)
Normal Forms in $F$

$$F = \langle x_0, x_1, x_2, \ldots | x_k^{-1}x_nx_k = x_{n+1}, \forall k < n \rangle$$

$$x_nx_k \rightarrow x_kx_{n+1} \text{ (smaller subscripts first)}$$

$$x_k^{-1}x_n \rightarrow x_{n+1}x_k^{-1} \text{ (positive before negative)}$$

$$x_n^{-1}x_k \rightarrow x_kx_{n+1}^{-1} \text{ (positive before negative)}$$

$$x_k^{-1}x_{n^{-1}} \rightarrow x_{n+1}^{-1}x_k^{-1} \text{ (smaller subscripts last)}$$

Normal forms:
Normal Forms in $F$

\[ F = \langle x_0, x_1, x_2, \ldots | x_k^{-1} x_n x_k = x_{n+1}, \forall k < n \rangle \]

\[ x_n x_k \rightarrow x_k x_{n+1} \text{ (smaller subscripts first)} \]

\[ x_k^{-1} x_n \rightarrow x_{n+1} x_k^{-1} \text{ (positive before negative)} \]

\[ x_n^{-1} x_k \rightarrow x_k x_{n+1}^{-1} \text{ (positive before negative)} \]

\[ x_k^{-1} x_n^{-1} \rightarrow x_{n+1}^{-1} x_k^{-1} \text{ (smaller subscripts last)} \]

Normal forms:

\[ f = x_{i_1} x_{i_2} \ldots x_{i_u} x_{j_v}^{-1} \ldots x_{j_2}^{-1} x_{j_1}^{-1} \ (i_1 \leq \ldots \leq i_u, j_1 \leq \ldots \leq j_v) \]
Normal Forms in $F$
Normal Forms in $F$

Unique, if $reduced$: if $x_i$ and $x_i^{-1}$, then so does $x_{i+1}$ or $x_{i+1}^{-1}$. 
Normal Forms in $F$

Unique, if reduced: if $x_i$ and $x_i^{-1}$, then so does $x_{i+1}$ or $x_{i+1}^{-1}$.

$$x_0x_1x_1x_3x_5^{-1}x_4^{-1}x_1^{-1}x_0^{-1} = x_0x_1x_2x_4^{-1}x_3^{-1}x_0^{-1}$$
Normal Forms in $F$

Unique, if reduced: if $x_i$ and $x_i^{-1}$, then so does $x_{i+1}$ or $x_{i+1}^{-1}$.

\[
x_0x_1x_3x_5^{-1}x_4^{-1}x_1^{-1}x_0^{-1} = x_0x_1x_2x_4^{-1}x_3^{-1}x_0^{-1}
\]

Theorem (Shpilrain-Ushakov, 2005)
Normal Forms in $F$

Unique, if reduced: if $x_i$ and $x_i^{-1}$, then so does $x_{i+1}$ or $x_{i+1}^{-1}$.

$$x_0x_1x_1^{-1}x_3x_5^{-1}x_1^{-1}x_0^{-1} = x_0x_1x_2x_4^{-1}x_3^{-1}x_0^{-1}$$

Theorem (Shpilrain-Ushakov, 2005)

If $|\cdot|$ denotes the word length, the normal form an element $g$ can be computed in time $O(|g|\log |g|)$. 
Parameters and Key Generation
Parameters and Key Generation

The proposed commuting subgroups of $F$ are defined from the previous presentation. Choose an $s \in \mathbb{N}$:
The proposed commuting subgroups of $F$ are defined from the previous presentation. Choose an $s \in \mathbb{N}$:

$$A_s = \langle x_0, x_1^{-1}, \ldots, x_0x_s^{-1} \rangle$$
The proposed commuting subgroups of $F$ are defined from the previous presentation. Choose an $s \in \mathbb{N}$:

$$A_s = \langle x_0 x_1^{-1}, \ldots, x_0 x_s^{-1} \rangle$$

$$B_s = \langle x_{s+1}, \ldots, x_{2s} \rangle$$
Choice of the parameters
Choice of the parameters

- Select (randomly) $s \in [3, 8]$ and an even $M \in [256, 320]$. 
Choice of the parameters

- Select (randomly) $s \in [3, 8]$ and an even $M \in [256, 320]$.
- Choose a random $w \in \langle x_0, x_1, \ldots, x_{s+2} \rangle$, with $|w| = M$. 
Choice of the parameters

- Select (randomly) $s \in [3, 8]$ and an even $M \in [256, 320]$.
- Choose a random $w \in \langle x_0, x_1, \ldots, x_{s+2} \rangle$, with $|w| = M$.
- Alice chooses random $a_1 \in A_s$, $b_1 \in B_s$, with $|a_1| = |b_1| = M$. 
Choice of the parameters

- Select (randomly) $s \in [3, 8]$ and an even $M \in [256, 320]$.
- Choose a random $w \in \langle x_0, x_1, \ldots, x_{s+2} \rangle$, with $|w| = M$.
- Alice chooses random $a_1 \in A_s, b_1 \in B_s$, with $|a_1| = |b_1| = M$.
- Bob chooses random $a_2 \in A_s, b_2 \in B_s$, with $|a_2| = |b_2| = M$. 
Choice of the parameters

- Select (randomly) $s \in [3, 8]$ and an even $M \in [256, 320]$.
- Choose a random $w \in \langle x_0, x_1, \ldots, x_{s+2} \rangle$, with $|w| = M$.
- Alice chooses random $a_1 \in A_s, b_1 \in B_s$, with $|a_1| = |b_1| = M$.
- Bob chooses random $a_2 \in A_s, b_2 \in B_s$, with $|a_2| = |b_2| = M$.

They both compute

$$K = a_1 b_2 w a_2 b_1$$
Choice of the parameters

- Select (randomly) \( s \in [3, 8] \) and an even \( M \in [256, 320] \).
- Choose a random \( w \in \langle x_0, x_1, \ldots, x_{s+2} \rangle \), with \(|w| = M\).
- Alice chooses random \( a_1 \in A_s, b_1 \in B_s \), with \(|a_1| = |b_1| = M\).
- Bob chooses random \( a_2 \in A_s, b_2 \in B_s \), with \(|a_2| = |b_2| = M\).

They both compute

\[
K = a_1 b_2 w a_2 b_1
\]

The key space increases exponentially in \( M \), i.e. \( |A_s(M)| \geq \sqrt{2}^M \).
$F$ as piecewise-linear homeomorphisms
$F$ as piecewise-linear homeomorphisms

$F$ is the group $PL_2(I)$, with respect to composition, of all piecewise-linear homeomorphisms of the unit interval $I = [0, 1]$ with a finite number of breakpoints, such that
$F$ as piecewise-linear homeomorphisms

$F$ is the group $PL_2(I)$, with respect to composition, of all piecewise-linear homeomorphisms of the unit interval $I = [0, 1]$ with a finite number of breakpoints, such that

- all slopes are integral powers of 2,
$F$ as piecewise-linear homeomorphisms

$F$ is the group $PL_2(I)$, with respect to composition, of all piecewise-linear homeomorphisms of the unit interval $I = [0, 1]$ with a finite number of breakpoints, such that

- all slopes are integral powers of 2,
- all breakpoints have dyadic rational coordinates.
As piecewise-linear homeomorphisms

$F$ is the group $PL_2(I)$, with respect to composition, of all piecewise-linear homeomorphisms of the unit interval $I = [0, 1]$ with a finite number of breakpoints, such that:

- all slopes are integral powers of 2,
- all breakpoints have dyadic rational coordinates.

Here is the first generator $x_0$ of $F$. 
$F$ as piecewise-linear homeomorphisms

$F$ is the group $PL_2(I)$, with respect to composition, of all piecewise-linear homeomorphisms of the unit interval $I = [0, 1]$ with a finite number of breakpoints, such that

- all slopes are integral powers of 2,
- all breakpoints have dyadic rational coordinates.

Here is the first generator $x_0$ of $F$:
$F$ as piecewise-linear homeomorphisms

$F$ is the group $PL_2(I)$, with respect to composition, of all piecewise-linear homeomorphisms of the unit interval $I = [0, 1]$ with a finite number of breakpoints, such that

- all slopes are integral powers of 2,
- all breakpoints have dyadic rational coordinates.

Here is the first generator $x_0$ of $F$.
Generators of $F$ as PL-homeomorphisms
Generators of $F$ as PL-homeomorphisms

The previous infinite generating set is given by:
Generators of $F$ as PL-homeomorphisms

The previous infinite generating set is given by:

![Diagram showing generators as PL-homeomorphisms]
Generators of $F$ as PL-homeomorphisms

The previous infinite generating set is given by:

$x_s$ acts non-trivially on the domain $[\varphi_{s-1}, 1]$, where

$$\varphi_s := 1 - \frac{1}{2s+1}$$
$A_s$ and $B_s$ as groups of homeomorphisms
$A_s$ and $B_s$ as groups of homeomorphisms

The subgroups $A_s$ and $B_s$ assume the following form:
$A_s$ and $B_s$ as groups of homeomorphisms

The subgroups $A_s$ and $B_s$ assume the following form:
The subgroups $A_s$ and $B_s$ assume the following form:

Their supports live in different squares, divided by $\varphi_s$. 
$A_s$ and $B_s$ as groups of homeomorphisms

The subgroups $A_s$ and $B_s$ assume the following form:

Their supports live in different squares, divided by $\varphi_s$

Observe that $B_s = PL_2([\varphi_s, 1])$. 
Tree diagrams for $F$
Tree diagrams for $F$

Elements of $F$ send a dyadic partition of $[0, 1]$ into another such partition.
Elements of $F$ send a dyadic partition of $[0,1]$ into another such partition. This can be represented by means of tree pairs.
Elements of $F$ send a dyadic partition of $[0, 1]$ into another such partition. This can be represented by means of tree pairs.

The element $x_0$ has the following diagram:
Elements of $F$ send a dyadic partition of $[0, 1]$ into another such partition. This can be represented by means of tree pairs.

The element $x_0$ has the following diagram:
Many tree pairs for the same element
Many tree pairs for the same element

It is possible to get a \textit{reduced} tree pair, by repeated application of the following reduction:
Many tree pairs for the same element

It is possible to get a *reduced* tree pair, by repeated application of the following reduction:
Many tree pairs for the same element

It is possible to get a *reduced* tree pair, by repeated application of the following reduction:
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient

To multiply fastly, we need to modify the diagram:
To multiply fastly, we need to modify the diagram:
To multiply fastly, we need to modify the diagram:
To multiply fastly, we need to modify the diagram:
Multiplication of diagrams is efficient (digression)
Multiplication of diagrams is efficient (digression)

These new diagrams have an input, an output, merges and splits
Multiplication of diagrams is efficient (digression)

These new diagrams have an input, an output, merges and splits.
Multiplication of diagrams is efficient (digression)

These new diagrams have an input, an output, merges and splits

They also have a set of reductions
Multiplication of diagrams is efficient (digression)

These new diagrams have an input, an output, merges and splits.

They also have a set of reductions.
Multiplication of diagrams is efficient (digression)

These new diagrams have an input, an output, merges and splits

![split diagram](image)

![merge diagram](image)

They also have a set of reductions

![Type I reduction](image)

![Type II reduction](image)
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient
Multiplication of diagrams is efficient
We need to cut the directed diagram back into a tree pair:
We need to cut the directed diagram back into a tree pair:
Multiplication of diagrams is efficient

We need to cut the directed diagram back into a tree pair:
Multiplication of diagrams is efficient

We need to cut the directed diagram back into a tree pair:
Multiplication of diagrams is efficient

We need to cut the directed diagram back into a tree pair:

[Diagram showing three parts with a cut indicated]
Multiplication of diagrams is efficient

All of the previous steps can performed fastly.

[Diagram showing multiplication of diagrams]
Outline of the attack
Outline of the attack

Recall: $A_s, B_s, w, u_1 = a_1 w b_1, u_2 = b_2 w a_2$ are public, and that

$$\varphi_s := 1 - \frac{1}{2^{s+1}}$$

separates the supports of $A_s$ and $B_s$. 

Outline of the attack

Recall: $A_s, B_s, w, u_1 = a_1 wb_1, u_2 = b_2 wa_2$ are public, and that

$$\varphi_s := 1 - \frac{1}{2^{s+1}}$$

separates the supports of $A_s$ and $B_s$.

1. Compute $w(\varphi_s)$ and see if $w(\varphi_s) \leq \varphi_s$ or $w(\varphi_s) > \varphi_s$. 
Outline of the attack

Recall: $A_s, B_s, w, u_1 = a_1 wb_1, u_2 = b_2 wa_2$ are public, and that

$$\varphi_s := 1 - \frac{1}{2^s+1}$$

separates the supports of $A_s$ and $B_s$.

1. Compute $w(\varphi_s)$ and see if $w(\varphi_s) \leq \varphi_s$ or $w(\varphi_s) > \varphi_s$.
2. If $w(\varphi_s) \leq \varphi_s$, attack Bob’s keys:
Recall: $A_s, B_s, w, u_1 = a_1w b_1, u_2 = b_2 w a_2$ are public, and that

$$
\varphi_s := 1 - \frac{1}{2^{s+1}}
$$

separates the supports of $A_s$ and $B_s$.

1. Compute $w(\varphi_s)$ and see if $w(\varphi_s) \leq \varphi_s$ or $w(\varphi_s) > \varphi_s$.
2. If $w(\varphi_s) \leq \varphi_s$, attack Bob’s keys:
   - compute the $A_s$-part $\bar{a}_2$ of $w^{-1}u_2 \in AB$,
Outline of the attack

Recall: $A_s, B_s, w, u_1 = a_1 wb_1, u_2 = b_2 wa_2$ are public, and that

$$\varphi_s := 1 - \frac{1}{2s+1}$$

separates the supports of $A_s$ and $B_s$.

1. Compute $w(\varphi_s)$ and see if $w(\varphi_s) \leq \varphi_s$ or $w(\varphi_s) > \varphi_s$.
2. If $w(\varphi_s) \leq \varphi_s$, attack Bob’s keys:
   - compute the $A_s$-part $\bar{a}_2$ of $w^{-1}u_2 \in AB$,
   - compute $\bar{b}_2 := u_2(\bar{a}_2)^{-1}w^{-1}$. 
Outline of the attack

Recall: $A_s, B_s, w, u_1 = a_1 wb_1, u_2 = b_2 wa_2$ are public, and that

$$\varphi_s := 1 - \frac{1}{2s+1}$$

separates the supports of $A_s$ and $B_s$.

1. Compute $w(\varphi_s)$ and see if $w(\varphi_s) \leq \varphi_s$ or $w(\varphi_s) > \varphi_s$.

2. If $w(\varphi_s) \leq \varphi_s$, attack Bob’s keys:
   - compute the $A_s$-part $\overline{a}_2$ of $w^{-1}u_2 \in AB$,
   - compute $\overline{b}_2 := u_2(\overline{a}_2)^{-1}w^{-1}$.

3. If $w(\varphi_s) > \varphi_s$, attack Alice’s keys:
Outline of the attack

Recall: $A_s, B_s, w, u_1 = a_1 wb_1, u_2 = b_2 wa_2$ are public, and that

$$\varphi_s := 1 - \frac{1}{2^{s+1}}$$

separates the supports of $A_s$ and $B_s$.

1. Compute $w(\varphi_s)$ and see if $w(\varphi_s) \leq \varphi_s$ or $w(\varphi_s) > \varphi_s$.
2. If $w(\varphi_s) \leq \varphi_s$, attack Bob’s keys:
   - compute the $A_s$-part $\overline{a}_2$ of $w^{-1}u_2 \in AB$,
   - compute $\overline{b}_2 := u_2(\overline{a}_2)^{-1}w^{-1}$.
3. If $w(\varphi_s) > \varphi_s$, attack Alice’s keys:
   - compute the $B_s$-part $\overline{b}_1$ of $w^{-1}u_1 \in AB$,
Outline of the attack

Recall: $A_s, B_s, w, u_1 = a_1 wb_1, u_2 = b_2 wa_2$ are public, and that

$$
\varphi_s := 1 - \frac{1}{2^{s+1}}
$$

separates the supports of $A_s$ and $B_s$.

1. Compute $w(\varphi_s)$ and see if $w(\varphi_s) \leq \varphi_s$ or $w(\varphi_s) > \varphi_s$.

2. If $w(\varphi_s) \leq \varphi_s$, attack Bob’s keys:
   - compute the $A_s$-part $\overline{a}_2$ of $w^{-1}u_2 \in AB$,
   - compute $\overline{b}_2 := u_2(\overline{a}_2)^{-1}w^{-1}$.

3. If $w(\varphi_s) > \varphi_s$, attack Alice’s keys:
   - compute the $B_s$-part $\overline{b}_1$ of $w^{-1}u_1 \in AB$,
   - compute $\overline{a}_1 := u_1(\overline{b}_1)^{-1}w^{-1}$. 
Outline of the attack

Recall: \( A_s, B_s, w, u_1 = a_1 w b_1, u_2 = b_2 w a_2 \) are public, and that

\[
\varphi_s := 1 - \frac{1}{2^{s+1}}
\]

separates the supports of \( A_s \) and \( B_s \).

1. Compute \( w(\varphi_s) \) and see if \( w(\varphi_s) \leq \varphi_s \) or \( w(\varphi_s) > \varphi_s \).
2. If \( w(\varphi_s) \leq \varphi_s \), attack Bob’s keys:
   - compute the \( A_s \)-part \( \overline{a}_2 \) of \( w^{-1} u_2 \in AB \),
   - compute \( \overline{b}_2 := u_2 (\overline{a}_2)^{-1} w^{-1} \).
3. If \( w(\varphi_s) > \varphi_s \), attack Alice’s keys:
   - compute the \( B_s \)-part \( \overline{b}_1 \) of \( w^{-1} u_1 \in AB \),
   - compute \( \overline{a}_1 := u_1 (\overline{b}_1)^{-1} w^{-1} \).

The pair \( (\overline{a}_i, \overline{b}_i) \) allows us to recover the shared key \( K \).
Explanation of the case $\varphi_s$
Explanations of the case $\omega(\varphi_s) \leq \varphi_s$

On $[0, \varphi_s]$ we have $b_2 = id$, and so

$$u_2(t) = b_2w_2(t) = w_2(t) \quad t \in [0, \varphi_s]$$
Explanation of the case $w(\varphi_s) \leq \varphi_s$

On $[0, \varphi_s]$ we have $b_2 = id$, and so

$$u_2(t) = b_2 wa_2(t) = wa_2(t) \quad t \in [0, \varphi_s]$$

Thus we have

$$a_2(t) = w^{-1}u_2(t) \quad t \in [0, \varphi_s].$$
Explanation of the case $w(\varphi_s) \leq \varphi_s$

On $[0, \varphi_s]$ we have $b_2 = id$, and so

$$u_2(t) = b_2 wa_2(t) = wa_2(t) \quad t \in [0, \varphi_s]$$

Thus we have

$$a_2(t) = w^{-1}u_2(t) \quad t \in [0, \varphi_s].$$

But $a_2 = id$ on $[\varphi_s, 1]$ and so

$$a_2(t) = \begin{cases} 
  w^{-1}u_2(t) & t \in [0, \varphi_s] \\
  t & t \in [\varphi_s, 1]
\end{cases}$$
Explanation of the case $w(\varphi_s) \leq \varphi_s$

On $[0, \varphi_s]$ we have $b_2 = id$, and so

$$u_2(t) = b_2 w a_2(t) = w a_2(t) \quad t \in [0, \varphi_s]$$

Thus we have

$$a_2(t) = w^{-1} u_2(t) \quad t \in [0, \varphi_s].$$

But $a_2 = id$ on $[\varphi_s, 1]$ and so

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

Notice $w^{-1} u_2(\varphi_s) = \varphi_s$ so $w^{-1} u_2 \in AB$. 
Explanation of the case $w(\varphi_s) \leq \varphi_s$

On $[0, \varphi_s]$ we have $b_2 = id$, and so

$$u_2(t) = b_2 w a_2(t) = wa_2(t) \quad t \in [0, \varphi_s]$$

Thus we have

$$a_2(t) = w^{-1} u_2(t) \quad t \in [0, \varphi_s].$$

But $a_2 = id$ on $[\varphi_s, 1]$ and so

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

Notice $w^{-1} u_2(\varphi_s) = \varphi_s$ so $w^{-1} u_2 \in AB$. So $a_2$ is given by the $A_s$-part of $w^{-1} u_2$. 
Explanation of the case $w(\varphi_s) \leq \varphi_s$
Explaination of the case $\nu(\varphi_s) \leq \varphi_s$

We want to recover the $A_s$-part of the element $\nu^{-1}u_2 \in AB$ in an efficient way.
Explanation of the case $w(\varphi_s) \leq \varphi_s$

We want to recover the $A_s$-part of the element $w^{-1}u_2 \in AB$ in an efficient way. We write the tree diagram of $w^{-1}u_2$. 

\[
\text{Diagram here}
\]
Explanation of the case $w(\varphi_s) \leq \varphi_s$

We want to recover the $A_s$-part of the element $w^{-1}u_2 \in AB$ in an efficient way. We write the tree diagram of $w^{-1}u_2$. 

![Tree Diagram]

Francesco Matucci  Cryptanalysis of the Shpilrain-Ushakov protocol in $F$
Explanation of the case \( w(\varphi_s) \leq \varphi_s \)

We want to recover the \( A_s \)-part of the element \( w^{-1}u_2 \in AB \) in an efficient way. We write the tree diagram of \( w^{-1}u_2 \).
Explanation of the case $w(\varphi_s) \leq \varphi_s$

We want to recover the $A_s$-part of the element $w^{-1}u_2 \in AB$ in an efficient way. We write the tree diagram of $w^{-1}u_2$.

\[ x_1x_0^{-1} \]
Explanation of the case $w(\varphi_s) \leq \varphi_s$

We want to recover the $A_s$-part of the element $w^{-1}u_2 \in AB$ in an efficient way. We write the tree diagram of $w^{-1}u_2$. 

Francesco Matucci  
Cryptanalysis of the Shpilrain-Ushakov protocol in $F$
Explanation of the case $w(\varphi_s) \leq \varphi_s$

We want to recover the $A_s$-part of the element $w^{-1}u_2 \in AB$ in an efficient way. We write the tree diagram of $w^{-1}u_2$. 

\[
x_1x_0^{-1}x_2
\]
Explanation of the case $w(\varphi_s) \leq \varphi_s$

We want to recover the $A_s$-part of the element $w^{-1}u_2 \in AB$ in an efficient way. We write the tree diagram of $w^{-1}u_2$.

![Tree diagram of $w^{-1}u_2$]

From the diagram of $a_2 \in A_s$ there is a fast algorithm to write it with the generators of $F$. 

Francesco Matucci  Cryptanalysis of the Shpilrain-Ushakov protocol in $F$
Attacking the other secret word.
Attacking the other secret word.

Depending on $w(\varphi_s)$, we chose to attack either Alice or Bob.
Attacking the other secret word.

Depending on $w(\varphi_s)$, we chose to attack either Alice or Bob.

We can also look for the other keys.
Attacking the other secret word.

Depending on $w(\varphi_s)$, we chose to attack either Alice or Bob.

We can also look for the other keys.

Similar techniques and the fact that

\[ A_s = PL_2([0, \varphi_s]) \]
\[ B_s = PL_2([\varphi_s, 1]) \]

allow us to recover an approximation for the other key.
Sketch of the attack to the other word
Sketch of the attack to the other word

We attack Alice’s word, for \( w(\varphi_s) \leq \varphi_s \):
Sketch of the attack to the other word

We attack Alice’s word, for $w(\varphi_s) \leq \varphi_s$:

$$u_1(t) = a_1 w(t) \quad t \in [0, \varphi_s]$$
Sketch of the attack to the other word

We attack Alice’s word, for $w(\varphi_s) \leq \varphi_s$:

$$u_1(t) = a_1w(t) \quad t \in [0, \varphi_s]$$

so that

$$a_1(t) = u_1w^{-1}(t) \quad t \in [0, w(\varphi_s)].$$
We attack Alice’s word, for \( w(\varphi_s) \leq \varphi_s \):

\[
u_1(t) = a_1 w(t) \quad t \in [0, \varphi_s]
\]

so that

\[
a_1(t) = u_1 w^{-1}(t) \quad t \in [0, w(\varphi_s)].
\]

This is the only requirement for \( a_1 \).
We attack Alice’s word, for $w(\phi_s) \leq \phi_s$:

$$u_1(t) = a_1 w(t) \quad t \in [0, \phi_s]$$

so that

$$a_1(t) = u_1 w^{-1}(t) \quad t \in [0, w(\phi_s)].$$

This is the only requirement for $a_1$.

Since $A_s = PL_2([0, \phi_s])$, we can find an $a_\sigma \in A_s$ such that

$$a_\sigma = a_1 \quad t \in [0, w(\phi_s)].$$
Sketch of the attack to the other word

We attack Alice’s word, for \( w(\varphi_s) \leq \varphi_s \):

\[
u_1(t) = a_1 w(t) \quad t \in [0, \varphi_s]\]

so that

\[
a_1(t) = u_1 w^{-1}(t) \quad t \in [0, w(\varphi_s)].
\]

This is the only requirement for \( a_1 \).

Since \( A_s = PL_2([0, \varphi_s]) \), we can find an \( a_\sigma \in A_s \) such that

\[
a_\sigma = a_1 \quad t \in [0, w(\varphi_s)].
\]

Then continue as before.
Changing the subgroups $A$ and $B$
Changing the subgroups $A$ and $B$

Theorem (Guba-Sapir, 1997-Kassabov-M, 2006)

$C_F(g) \cong F^m \times \mathbb{Z}^n, \forall g \in F.$
Changing the subgroups $A$ and $B$

**Theorem (Guba-Sapir, 1997-Kassabov-M, 2006)**

$$C_F(g) \cong F^m \times \mathbb{Z}^n, \forall g \in F.$$
Theorem (Guba-Sapir, 1997-Kassabov-M, 2006)

\[ C_F(g) \cong F^m \times \mathbb{Z}^n, \forall g \in F. \]

The \( F \)-terms correspond to the intervals where \( g \) is trivial.
Changing the subgroups $A$ and $B$

Theorem (Guba-Sapir, 1997-Kassabov-M, 2006)

\[ C_F(g) \cong F^m \times \mathbb{Z}^n, \forall g \in F. \]

The $F$-terms correspond to the intervals where $g$ is trivial.
The $\mathbb{Z}$-terms correspond to the intervals where $g$ is non-trivial.
Changing the subgroups $A$ and $B$

**Theorem (Guba-Sapir, 1997-Kassabov-M, 2006)**

$$C_F(g) \cong F^m \times \mathbb{Z}^n, \forall g \in F.$$  

The $F$-terms correspond to the intervals where $g$ is trivial. The $\mathbb{Z}$-terms correspond to the intervals where $g$ is non-trivial.

If $A$ is a subgroup, and $b \in F$ commutes with $A$ elementwise, the support of $A$ and $b$ must be “disjoint.”
Choosing a different group
Choosing a different group

If instead of $F$ we consider a larger group of PL-homomorphisms of the unit interval, then two commuting subgroups still must have “disjoint” support.
Choosing a different group

If instead of $F$ we consider a larger group of PL-homomorphisms of the unit interval, then two commuting subgroups still must have “disjoint” support.

What requires attention is an “extension problem”.
Choosing a different group

If instead of $F$ we consider a larger group of PL-homomorphisms of the unit interval, then two commuting subgroups still must have “disjoint” support.

What requires attention is an “extension problem”.

Example: given $a_1$ on $[0, w(\varphi_s)]$, find $a_\sigma \in A$ with $a_\sigma = a_1$. 
Choosing a different group

If instead of $F$ we consider a larger group of PL-homomorphisms of the unit interval, then two commuting subgroups still must have “disjoint” support.

What requires attention is an “extension problem”.

Example: given $a_1$ on $[0, w(\varphi_s)]$, find $a_\sigma \in A$ with $a_\sigma = a_1$.

More generally, if we choose a group $G$ acting on some space, and have $A, B$ commuting elementwise so that their support is disjoint, a similar technique may apply.
Conclusions
Conclusions

**Good:** we are always able to recover the secret key.
Conclusions

**Good:** we are always able to recover the secret key.

**Limits:** Our methods depend strongly on the fact that commuting subgroups have disjoint supports.
Conclusions

**Good:** we are always able to recover the secret key.

**Limits:** Our methods depend strongly on the fact that commuting subgroups have disjoint supports.

They still apply using the same protocol (or some variation of it) on other groups, but they cannot be used in a general context where no other representation is given.
Related work
In 2006, Ruisnkiy-Shamir-Tsaban have developed some more general length-based attacks which recover the secret key in most instances.
In 2006, Ruisnkiy-Shamir-Tsaban have developed some more general length-based attacks which recover the secret key in most instances.

In May 2007, Runskiy-Shamir-Tsaban have uploaded a paper on the arXiv with new general type of attacks based on the “subgroup distance function” and they tested it yet again on this protocol.