Strand Spaces

Guttman, Thayer, Herzog

Strand Spaces Summary

• Definitions
• Architecture
• An Example

Strands

• Represent the data/actions in a protocol that one principal may use/take
• Depicted by sending and receiving terms
• May, but need not, be alternating sends and receives
An abstract Strand

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN</td>
<td>pseudo-Strand A</td>
</tr>
<tr>
<td>A-&gt;B:</td>
<td>snd([na]k)</td>
</tr>
<tr>
<td>B-&gt;A:</td>
<td>rcv([nb]k)</td>
</tr>
<tr>
<td>A-&gt;B:</td>
<td>snd([na]k)</td>
</tr>
</tbody>
</table>

Terms

- Represent protocol data
- Denoted by a pair
  - "+" or "-" indicating sent or received data
  - The ID of the data transmitted
- The terms for the statement "A->B: msg" are:
  - Alice's strand: +msg
  - Bob's strand: -msg

A Strand

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN</td>
<td>Strand A</td>
</tr>
<tr>
<td>A-&gt;B:</td>
<td>+[na]k</td>
</tr>
<tr>
<td>B-&gt;A:</td>
<td>-[nb]k</td>
</tr>
<tr>
<td>A-&gt;B:</td>
<td>+nb</td>
</tr>
</tbody>
</table>
Nodes

- **Node**: Pair \(<s,i>\) where:
  - \(s\) is a strand and
  - \(1 \leq i \leq \text{length}(s)\).
- Node \(<s,i>\) belongs to strand \(s\)
- \(n = <s,i> \in N_s\) if \(n\) is a node in strand \(s\)
- if \(n = <s,i> \in N_s\), then:
  - \(\text{index}(n) = i\) and
  - \(\text{strand}(n) = s\)
- Each node \(n\) is associated with a term that can be extracted by the operation:
  - \(\text{term}(n)\)

Two Strands With Nodes Labeled

<table>
<thead>
<tr>
<th>SN</th>
<th>(n_i)</th>
<th>Strand A</th>
<th>Strand B</th>
<th>(n'_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-&gt;B: {na}k</td>
<td>(&lt;A,1&gt;)</td>
<td>+{na}k</td>
<td>-{na}k</td>
<td>(&lt;B,1&gt;)</td>
</tr>
<tr>
<td>B-&gt;A: {nb}k</td>
<td>(&lt;A,2&gt;)</td>
<td>-{nb}k</td>
<td>+{nb}k</td>
<td>(&lt;B,2&gt;)</td>
</tr>
<tr>
<td>A-&gt;B: nb</td>
<td>(&lt;A,3&gt;)</td>
<td>+nb</td>
<td>-nb</td>
<td>(&lt;B,3&gt;)</td>
</tr>
</tbody>
</table>

Connecting Strands

- The connection between strands is straightforward, corresponding nodes of the strands are connected by send and receive operators
- These send/receive operators are represented by horizontal arrows in strand notation
- The horizontal arrow denotes causal precedence of terms in different strands. Intuitively, the RECEIVE must occur after the SEND. In that sense, the RECEIVE is "caused" by the SEND.
Two Connected Strands

<table>
<thead>
<tr>
<th>SN</th>
<th>n_i</th>
<th>Strand A</th>
<th>Strand B</th>
<th>n'_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>A→B: {na}k</td>
<td>&lt;A,1&gt;</td>
<td>+{na}k</td>
<td>→</td>
<td>-{na}k</td>
</tr>
<tr>
<td>B→A: {nb}k</td>
<td>&lt;A,2&gt;</td>
<td>-{nb}k</td>
<td>←</td>
<td>+{nb}k</td>
</tr>
<tr>
<td>A→B: nb</td>
<td>&lt;A,3&gt;</td>
<td>+nb</td>
<td>→</td>
<td>-nb</td>
</tr>
</tbody>
</table>

Top Eight Headlines

1. Chef Throws His Heart into Helping Feed Needy
2. Red Tape Holds Up New Bridges
3. Typhoon Rips Through Cemetery; Hundreds Dead
4. Man Struck By Lightning Faces Battery Charge
5. New Study of Obesity Looks for Larger Test Group
6. Kids Make Nutritious Snacks
7. Local High School Dropouts Cut in Half
8. Hospitals are Sued by 7 Foot Doctors

Bundles (C)

• A bundle is a set of edges C with the following properties
  – C is finite
  – One and only one sender for each receiver
  – All actions in each strand are in the bundle
  – No cycles
A Bundle

\[ S_1 \quad \rightarrow \quad +_{\{na\}k} \quad \rightarrow \quad -_{\{na\}k} \]

\[ -_{\{nb\}k} \quad \leftarrow \quad +_{\{nb\}k} \]

\[ +_{nb} \quad \rightarrow \quad -_{nb} \]

Edges

- \( n_1 \rightarrow n_2 \) means that \( \text{term}(n_1) = +a \) and \( \text{term}(n_2) = -a \) for some message 'a'
  - Said another way,
    - \( \text{unterm}(n_1) = \text{unterm}(n_2) \)
- \( n_1 \Rightarrow n_2 \) means that
  - \( n_1 \) and \( n_2 \) are on the same strand and
  - \( \text{index}(n_1) = \text{index}(n_2) - 1 \)

Term and Key Composition

- \( T \) = set of texts
- \( K \) = set of keys disjoint from \( T \)
- \( a \in t \) means \( a \) is a subterm of \( t \)
  - \( a \in t \) for \( t \in T \) iff \( a = t \)
  - \( a \in K \) for \( K \) iff \( a = K \)
  - \( a \in \{g\}_k \) iff \( a \in g \) or \( a = \{g\}_k \)
  - \( a \in g \) iff \( a \in g \), \( a \in h \) or \( a = g \in h \)

Keys never occur in the term space

Text terms have only one subterm

Keys have only one subterm

The only subterms of an encrypted value are:
1. The encrypted value itself
2. The subterms of the value before it was encrypted

In order to be a subterm of a join, 'a' must be the join itself, or a subterm of one of the joined terms.
Lemma 2.6

- If:
  - C is a bundle
  - \( \preceq \) is a partial ordering relation on C
- Then: Every non-empty subset of the nodes in C has \( \preceq \) minimal members.

\[\begin{array}{c|c|c}
\hline
n_i & \text{Strand A} & \text{Strand B} \\
\hline
\langle A,1 \rangle & +\text{msg1} & \rightarrow & -\text{msg1} & \langle B,1 \rangle \\
\langle A,2 \rangle & -\text{msg2} & \leftarrow & +\text{msg2} & \langle B,2 \rangle \\
\langle A,3 \rangle & +\text{msg3} & \rightarrow & -\text{msg3} & \langle B,3 \rangle \\
\hline
\end{array}\]

\[\begin{array}{c|c|c}
\hline
n_i & \text{Strand A} & \text{Strand B} \\
\hline
\langle A,2 \rangle & -\text{msg2} & \leftarrow & +\text{msg2} & \langle B,2 \rangle \\
\langle A,3 \rangle & +\text{msg3} & \rightarrow & -\text{msg3} & \langle B,3 \rangle \\
\hline
\end{array}\]
Lemma 2.7

- If:
  1. C is a bundle
  2. S is a set of nodes where for all nodes m,m',
     \(\text{unterm}(m) = \text{unterm}(m') \iff m \in S \Rightarrow m' \in S\)

- Then: If \(n\) is a _minimal_ member of \(S\), then the sign of \(n\) is positive.

- **Proof:** Assume \(n\) is negative. Then, \(\exists n'\mid n' \rightarrow n\)
  (by bundle property, there must be a + for each -).
- By assumption (2), \(n' \in S\).
- Since \(n' \rightarrow n\) and \(n' \in S\), \(n\) cannot be minimal.
- So, \(n\) cannot be minimal and negative

Lemma 2.8

If:
- C is a bundle and
- \(t \in A\) (\(t\) is a term in a node in the bundle \(C\)) and
- \(n \in C\) is a _minimal_ element of \(\{m \in C : t \text{ term}(m)\}\)
  i.e. \(n\) is the minimal node of the subset of nodes in \(C\) that contain the term \(t\) as a subterm.

Then \(n\) is an originating occurrence for \(t\)

- **Proof:**
  - For any \(n' \in C\) where \(n' \in C\),
  - By the minimality property of \(n\) \(\text{term}(n')\).
  - Thus, \(n\) is originating for \(t\).

The Strand Approach

1. Define the operations that penetrators can perform
2. Prove properties of penetrator strands
## Penetrator Atomic Actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M. Text message</td>
<td><code>&lt;+a&gt;</code>, where <code>a ∈ T</code></td>
</tr>
<tr>
<td>F. Flushing</td>
<td><code>&lt;-g&gt;</code></td>
</tr>
<tr>
<td>T. Tee</td>
<td><code>&lt;-g, +g, +g&gt;</code></td>
</tr>
<tr>
<td>C. Concatenation</td>
<td><code>&lt;-g, -h, +g h&gt;</code></td>
</tr>
<tr>
<td>S. Separation into components</td>
<td><code>&lt;-g h, +g, +h&gt;</code></td>
</tr>
<tr>
<td>K. Key</td>
<td><code>&lt;+K&gt;</code> where <code>K ∈ K</code></td>
</tr>
<tr>
<td>E. Encryption</td>
<td><code>&lt;-K, -h, +{h}K&gt;</code></td>
</tr>
<tr>
<td>D. Decryption</td>
<td><code>&lt;-K⁻¹, -{h}K, +h&gt;</code></td>
</tr>
</tbody>
</table>

## Proving a Proposition

1. Let `C` be any arbitrary bundle and
2. `K ∈ K \ K_ρ` and
3. If `K` never originates on a regular node
4. Then `K □ term(p)` for any penetrator node `p ∈ C`.

## Proving a Proposition

1. Let `C` be a bundle and `K ∈ K \ K_ρ` and
2. Assume `K` never originates on a regular node in `C`
3. Then `K □ term(p)` for any penetrator node `p ∈ C`.
4. Consider `S = {n ∈ C: K □ term(n)}`.
   - **Suppose `S` is non-empty.**
   - Then `S` has members that are:
     5. minimal (L2.8)
     6. Originating nodes for `K` (L2.6)
     7. Penetrator nodes (by (2))
   - Examine possible penetrator node cases.
Possible Cases for Penetrator Nodes from Def 3.1

- M: Not possible since $K \Box t$.
- F: No positive nodes
- T: No value originates on positive nodes
- C: No value originates on positive nodes
- S: No value originates on positive nodes
- K: This form is a Key originating from a penetrator, but the key must be in $K_p$, which is false by assumption
- E: Close, but no key can occur in the positive node that did not occur in an earlier node
- D: Close, but no key can occur in the positive node that did not occur in an earlier node

The Contradiction

- We assume that there exists a penetrator node in C where K occurs
- By the form of the penetrator actions, we showed that no such occurrence can happen
- So, the assumption that such a node exists must be false

Review

- Strand Space Model
  - Terms, Edges, Strands, Bundles
- Strand Space Lemmas
- Penetrator Actions
- Proof Example