CSC 774 -- Network Security

Topic 7.3: Optimistic Fair Exchange

Outline

• Overview of Fair Exchange
• Optimistic Fair Exchange
  – A General Protocol
  – Optimized Protocol
    • Contract signing
• Take-home reading
  – Optimized Protocols
    • Certified mail
    • Payment for receipt
    • Fair purchase
Fair Exchange

- A fair exchange should guarantee that at the end of the exchange
  - Either each party has received what it expects to receive,
  - Or no party has received anything

- Examples
  - Certified mail
  - Contract signing
  - Payment

Traditional Fair Exchange

- ISO proposals
  - Use a TTP to ensure fairness

- Limitations
  - TTP is heavily involved
  - Bottleneck
  - Single point of failure

[Diagram of Trusted Third Party (TTP) connected to Originator and Responder]
**Optimistic Fair Exchange**

- **Assumptions**
  - Most participants are honest
- **Allow participants to exchange without TTP**
- **Fall back to TTP when there are failures**
  - Dishonest participants, communication failures, etc.

![Diagram of Optimistic Fair Exchange]

**Three Phases of Optimistic Fair Exchange**

- **Phase 1**
  - The parties try to exchange items without a TTP
- **Phase 2**
  - The parties try to exchange items through a TTP
- **Phase 3**
  - Each computer outputs all evidence and any participant may visit a court
Degree of Fairness

- **Strong (true) fairness**
  - If the TTP is able to
    - Undo a transfer of an item (revocability)
      - Example: revoke a signed contract
    - Produce a replacement for it (Generatability)
      - Example: generate a replacement of a receipt
  - Weak fairness
    - If the TTP can only produce affidavits
    - Requires an external dispute resolution system
      - Example: court

Generic Exchange Protocol

- Two stages
  - Stage 1 (Two flows)
    - The originator O and the recipient R promise each other an exchange of items
  - Stage 2 (Three flows)
    - Exchange the items along with non-repudiation tokens
Notations

- item\(_X\): the item X wants to send
- descr\(_X\): a description of item\(_X\)
- expect\(_X\)(descr\(_X\), descr\(_Y\)):
  - Evaluate to true if X is satisfied with exchanging item\(_X\) with item\(_Y\).
- fits(descr, item)
  - Evaluate to true if the description fits the item
- h(): hash function
- (key, comm) = commit(item)
  - Generate a commitment comm to item, and also generate a key, without which it’s impossible to get the item.
  - Verifiable encryption.
- open(item, key, comm)
  - Use key to open the item whose commitment is comm.

---

Generic Exchange Protocol (Cont’)

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>In: item(_O), descr(_O), expect(_O)</td>
<td></td>
<td>In: item(_R), descr(_R), expect(_R)</td>
</tr>
</tbody>
</table>

Choose \(y_0\) (recovery authenticator) \(r_0\) (NRR authenticator) randomly; determine T
(\(key_0, com_0\) := commit (item\(_O\)))

\[ m_1 := \text{sign}_O(T, R, h(y_0), h(r_0), t, com_0, descri_0) \]

If not \(\text{expect}_R(\text{descr}_R, \text{descr}_O)\) then Abort;
Choose \(y_R\) (recovery authenticator) \(r_R\) (NRR authenticator) randomly;
(\(key_R, com_R\) := commit (item\(_R\)))

\[ m_2 := \text{sign}_R(O, h(m_1), h(y_R), h(r_R), com_R, descri_R) \]
Generic Exchange Protocol (Cont’d)

\[
\begin{align*}
\text{expect}_O(\text{descr}_O, \text{descr}_R) \\
m_1 := \text{item}_O, \text{key}_O
\end{align*}
\]

\[
\begin{align*}
\text{open}(\text{item}_O, \text{key}_O, \text{com}_O) \\
m_2 := \text{item}_R, \text{key}_R, \text{com}_R
\end{align*}
\]

If fits(\text{item}_R, \text{descr}_R) and opens(\text{item}_R, \text{key}_R, \text{com}_R) and [no timeout] then

\[
m_3 := \text{item}_O
\]

else [Recovery for O]

\[
m_3 := \text{r}_O
\]

If [timeout] then

[Recovery for R]

\[
\begin{align*}
\text{Output:} \\
\text{item}_R \\
\text{NRO Token: } (m_2, \text{key}_R, \text{com}_R) \\
\text{NRR Token: } (m_1, m_2, \text{r}_R)
\end{align*}
\]

\[
\begin{align*}
\text{Output:} \\
\text{item}_O \\
\text{NRO Token: } (m_2, \text{key}_O, \text{com}_O) \\
\text{NRR Token: } (m_1, m_2, \text{r}_O)
\end{align*}
\]

• Question:
  – Why can these tokens guarantee NRO or NRR?
### Recovery for O

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m := m_1, m_2, y_O$</td>
<td>If [the received messages fit together] then</td>
<td>retransmit $m_3$, observable by $T$</td>
</tr>
<tr>
<td></td>
<td>If [retransmit invalid] then abort not [timeout] then</td>
<td>retransmit $m_4$, observable by $T$</td>
</tr>
<tr>
<td></td>
<td>open $(\text{item}_O, \text{key}_O, \text{com}_O)$?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fits $(\text{descr}_O, \text{item}_O)$?</td>
<td>$m_5$</td>
</tr>
<tr>
<td></td>
<td>else</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$m_T = \text{sign}_T(h(m))$ or $\text{sign}_T(&quot;\text{Cancel&quot;}, h(m))$</td>
<td></td>
</tr>
</tbody>
</table>

### Question

- Can this recovery protocol guarantee
  - Strong fairness for $O$?
    - ____
  - Weak fairness for $O$?
    - ____
Recovery for R

\[ m := m_1, m_2, y_R \]

If [the received messages fit together] then

retransmit \( m_4 \), observable by \( T \)

open (item\(_R\), key\(_R\), com\(_R\))?
fits (item\(_R\), descr\(_R\))?

\[ m_5 \]

If [retransmit invalid] then abort
if not [timeout] then

else

\[ m_T := \text{sign}_T(h(m)) \]

Question

- Can this recovery protocol guarantee
  - Strong fairness for \( R \)?
    - ______
  - Weak fairness for \( R \)?
    - ______
Types of items

- Confidential data
  - Data that will be released during the protocol
  - Example: Software
- Public data
  - Data that will be released even if the protocol execution fails
  - Purpose: fair exchange of non-repudiation tokens.
  - Example: contract
- Payments
  - A payment sub-protocol that is executed to transfer value from payer to payee
  - Example: PayWords

Types of Items (Cont’d)

- Generatable
  - The TTP can produce a replacement of the item
- Revocable
  - The TTP can undo the transfer of the item

<table>
<thead>
<tr>
<th>Public Data</th>
<th>Conf. Data</th>
<th>Payment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generatable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revocable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exchange Types

<table>
<thead>
<tr>
<th>Public Data</th>
<th>Conf. Data</th>
<th>Payment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Data</td>
<td>Contract Signing</td>
<td>Payment with Receipt</td>
</tr>
<tr>
<td>Conf. Data</td>
<td>Exchange of Goods</td>
<td>Fair Purchase</td>
</tr>
<tr>
<td>Payment</td>
<td></td>
<td>Currency Exchange</td>
</tr>
</tbody>
</table>

Optimized Protocol -- Contract Signing

\[ \text{O} \quad \text{T} \quad \text{R} \]

\[ \text{In: } \text{contract}_O \quad \text{In: } \text{contract}_R \]

Choose \( o_O \) randomly; determine \( T \)

\[ m_1 := \text{sign}_O(T, R, h(o_O), t, \text{contract}_O) \]

Choose \( y_R \) randomly.

\[ \text{contract}_R = \text{contract}_O ? \]

\[ m_2 := \text{sign}_R(h(m_1), h(y_R)) \]

\[ m_3 := o_O \]
Contract Signing (Cont’d)

\[
\text{If [timeout] then} \quad m := m_1, m_2, y_R
\]

\[
\text{If [the received messages fit together] then}
\]

\[
\text{If [response] then}
\]

\[
\text{else}
\]

\[
m_r := \text{sign}_r(h(m))
\]

Output: contractR, (m1, m2)

Output: contractR, (m1, o_0)

• Question:
  – Why can these tokens guarantee NRO or NRR?