Topic 3.3: Fair Exchange

Outline

• Overview of Fair Exchange
• Optimistic Fair Exchange
  – A General Protocol
  – Optimized Protocol
    • Contract signing
• Self-study
  – 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
**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.

**Trusted Third Party (TTP)**

**Originator**

**Responder**

**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

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

Generic Exchange Protocol (Cont’)

\begin{tabular}{|c|c|c|}
\hline
O & T & R \\
\hline
In: \text{item}_O, \text{descr}_O, \text{expect}_O & & In: \text{item}_R, \text{descr}_R, \text{expect}_R \\
\hline
\end{tabular}

Choose \text{y}_O (recovery authenticator) \text{r}_O (NRR authenticator) randomly; determine T \text{(key}_O, \text{com}_O) := \text{commit} (\text{item}_O)

\[ m_1 := \text{sign}_O(T, R, h(y)_O, h(r)_O, t, \text{com}_O, \text{descr}_O) \]

If not \text{expect}_R(\text{descr}_R, \text{descr}_O) then Abort;
Choose \text{y}_R (recovery authenticator) \text{r}_R (NRR authenticator) randomly;
\text{(key}_R, \text{com}_R) := \text{commit} (\text{item}_R)

\[ m_2 := \text{sign}_R(O, h(m_1), h(y)_R, h(r)_R, \text{com}_R, \text{descr}_R) \]
Generic Exchange Protocol (Cont’d)

\[
\begin{align*}
\text{expect}_O(\text{descr}_O, \text{descr}_R) \\
m_3 := \text{item}_O, \text{key}_O \\
\text{open (item}_O, \text{key}_O, \text{com}_O)? \\
m_2 := \text{item}_R, r_R, \text{key}_R \\
\text{fits(descr}_O, \text{item}_O)? \\
\text{If fits (item}_R, \text{descr}_R) \text{ and open (item}_R, \text{key}_R, \text{com}_R) \\
\text{and [no timeout] then} \\
m_4 := r_O \\
\text{else [Recovery for O]} \\
\text{If [timeout] then} \\
\text{[Recovery for R]}
\end{align*}
\]

Output:
- \(\text{item}_R\)
- NRO Token: \((m_2, \text{key}_R, \text{com}_R)\)
- NRR Token: \((m_1, m_2, r_R)\)

Output:
- \(\text{item}_O\)
- NRO Token: \((m_1, \text{key}_O, \text{com}_O)\)
- NRR Token: \((m_1, m_2, r_O)\)

- Question:
  - Why can these tokens guarantee NRO or NRR?
Recovery for O

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

If [the received messages fit together] then
- retransmit \( m_3 \), observable by T

If [retransmit invalid] then abort else if not [timeout] then
- retransmit \( m_4 \), observable by T

\[ m_T = \text{sign}_T(h(m)) \text{ or } \text{sign}_T("Cancel", h(m)) \]

Question

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

If [the received messages fit together] then

retransmit $m_\sim$, observable by T

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

open (item$_{R}$, key$_{R}$, com$_{R}$)?
fits (item$_{R}$, descr$_{R}$)?

$m_5$

else

$m_T := sign_T(h(m_\sim))$

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

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

Optimized Protocol -- Contract Signing

In: contract_{O}

Choose o_o randomly; determine T;

\[ m_1 := \text{sign}_{o_o}(T, R, h(o_o), t, \text{contract}_o) \]

Choose y_R randomly; contract_R = contract_o?

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

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

If [timeout] then

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

If [the received messages fit together] then

\[
m_2
\]

If [response] then

\[
m_3
\]

else

\[
m_p := \text{sign}_P(h(m))
\]

Output:

contract_R, (m_1, m_2)

Output:

contract_R, (m_1, o_0)

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