CSC 774 Advanced Network Security

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.

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
- \( h() \): hash function
- \( (\text{key}, \text{comm}) = \text{commit}(\text{item}) \)
  - 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’)

- Choose \( y_O \) (recovery authenticator)
- \( r_O \) (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 \( y_R \) (recovery authenticator)
- \( r_R \) (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)

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{expect}_O(\text{descr}_O, \text{descr}_R) )</td>
<td>( m_3 := \text{item}_O, \text{key}_O )</td>
<td>( \text{fits}(\text{descr}_O, \text{item}_O) )</td>
</tr>
<tr>
<td></td>
<td>( m_4 := \text{item}_R, r_R, \text{key}_R )</td>
<td>( \text{open}(\text{item}_O, \text{key}_O, \text{com}_O) )</td>
</tr>
<tr>
<td>If fits(\itemR, descrR) and open(\itemR, keyR, comR) and [no timeout] then</td>
<td></td>
<td>If [timeout] then [Recovery for R]</td>
</tr>
<tr>
<td></td>
<td>( m_5 := r_O )</td>
<td>else [Recovery for O]</td>
</tr>
</tbody>
</table>

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

### Output:
- **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 [timeout] then

open (item_0, key_0, com_0)?
fits ( descr_0, item_0)?

retransmit \( m_4 \), observable by T

open (item_R, key_R, com_R)?
fits ( item_R, descr_R)?

else

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

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 $m_4$, observable by T

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

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

else

$m_5 := m_1, m_2, y_R$

$\rightarrow$

$m_T := sign_T(h(m))$

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></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

<table>
<thead>
<tr>
<th>O</th>
<th>T</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{In: } \text{contract}_O</td>
<td>\text{In: } \text{contract}_R</td>
<td>\text{In: } \text{contract}_R</td>
</tr>
</tbody>
</table>

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

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

Choose \( y_R \) randomly; \( \text{contract}_R \equiv \text{contract}_o? \)

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

\( m_3 := o_o \)
If \([\text{timeout}]\) then
\[ m := m_1, m_2, y_R \]

If [the received messages fit together] then

\[ m_2 \]

If [response] then

\[ m_3 \]

else

\[ m_2 := \text{sign}_R (h(m)) \]

Output:
\[ \text{contract}_R, (m_1, m_2) \]

Output:
\[ \text{contract}_R, (m_1, \alpha_0) \]

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