J2ME Version of the Tuple Board

Travis Thomas
M.S. Project Defense
Committee:
Associate Professor Alan Kaminsky
Assistant Professor Minseok Kwon
Professor Edith Hemaspanndra
Outline

- Tuple Board Background
- J2ME Tuple Board
- Secure Card Deck
- War Game
- Texas Hold 'em
- Performance
- Demo
Background

• Ad Hoc Networks
  – No concept of client/server
  – Self organizing
  – Very dynamic

• Tuple Space
  – Developed by Professor David Gelernter.
  – Basic idea is to post tuples into a space where all devices can see them.
  – Devices can interact with tuples to solve problems.
  – Basic operations: write, read, notify and take
Background Con't

- **Tuple Board**
  - Developed by Professor Alan Kaminsky and students.
  - Extends the idea of tuple space to ad-hoc networking.
  - Basic operations are post, withdraw, read, notify.
Tuple Space vs Board

• Analogy:
  – Tuple Space is like a tack board:
    • Anybody can add content
    • Anybody can remove content.
  – Tuple Board is like display cases:
    • Each device can post things in their display case
    • Only posting device can take content down
  – Both allow sharing of information.
Back Ground Con't

- J2ME
  - Micro version of Java designed to run on devices less powerful than desktops
  - More limited api allows it to cut back on resources required.
Idea

- Tuple board could reach a wider audience if it could be deployed on J2ME devices.
- Card games that could be played on phones versus other people also playing on phone could be a killer application.
  - People like to play cards
  - May not have cards in pocket
  - Very likely they will have their cell phone
  - Allows for testing concept of ad-hoc Secure Deck of Cards
Sony Ericsson 990i

- Smart phone
  - Runs J2ME with some key packages that are relied on heavily by Tuple Board:
    - Java.net
    - Java.reflect
  - Wifi adapter
J2ME Tuple Board

- Very similar to design of original tuple board
- TupleBoardProxy
  - Uses Sockets in J2ME instead of channels.
- ByteBuffer
  - Had to be written from scratch for J2ME version.
- No generics or for each loops.
J2ME Tuple Board

- Very similar to use as J2SE tuple board.
- Tuple board always returns base tuple type requires explicit programmer cast.
Secure Card Deck

• Why?
  – No trusted server
  – All players want to play
    • Can't be trusted not to collude
Secure Card Deck

- All players agree on a large prime $p$
- Each player chooses random secret exponent $E$
- Each player computes the corresponding decryption exponent $D$
  - Such that

$$ED = 1 \pmod{\phi(p)}, \text{where } \phi(p) = p - 1 \text{ since } p \text{ is prime}$$
Secure Card Deck Encryption

• Each player uses their encryption exponent $E$ on the card $X$

$$X^E \pmod{p} = XE$$

- SecondPlayer encrypts value player one already encrypted.

$$XE^{E_2} \pmod{p} = XEE_2$$
Secure Card Deck Dealing

- Cards are dealt by assigning the index in the list of encrypted cards after all devices have encrypted and shuffled.

<table>
<thead>
<tr>
<th>EncryptedCard1</th>
<th>EncryptedCard2</th>
<th>EncryptedCard3</th>
<th>EncryptedCard4</th>
<th>EncryptedCard5</th>
</tr>
</thead>
</table>

P1
- Gets 1
- And 3

P2
- Gets 2
- And 4
Secure Card Deck Decryption

- Device gets other devices to remove their E using their D

\[ X^{EE^D (\text{mod } p)} = X^E \]

- Now device can remove its encryption and view the card.

\[ (X^E)^D (\text{mod } p) = X \]
Secure Deck Setup

- Create Prime Modulus and private keys
- Create original BigInteger Cards
- Encrypt and shuffle cards
- Read() DeckInfoTuple
- Return DeckInfoTuple
- Create e and d based on DeckInfo prime modulus
Secure Deck Setup

Main Secure Deck → TupleBoard

Read() → FinalDeckTuple

Return finalDeckTuple

Secure Deck

Clean up some tuples

Clean up some tuples
War Game

- Game usually played by two people using a deck of cards.
- Each participant is dealt 26 cards.
- Place 1 card face up on table; player with highest card gets both.
- Both players have same card; war commences.
  - Three cards face down, another face up winner takes all.
- Game ends when one player runs out of cards.
War Game (WG) Phone

oppnent
Diamond 3#26

You have won the round!

Diamond Queen # 26

Take Theirs First  Play Card  Take Yours First
WG Initialization Communication

- WarGame (1) adds a SessionTuple Listener.
- SecureDeck posts a SessionTuple.
- TupleBoard is notified of the SessionTuple.
- WarGame (2) adds a SessionTuple Listener.
- WarGame (2) posts a PlayerTuple.
- TupleBoard gets an iterator with PlayerTuple Template.
- Use iterator: Player tuples are returned.
- Knows opponent has joined.
- Create SecureDeck with makedeck flag true.
- Create SecureDeck with makedeck flag false.
WG Initialization Con't
WG Game Play

1. WarGame (1) requests 26 cards for each player.
3. TupleBoard reads associated HandTuple.
4. PlayCard button enabled user will press button to play.
WG Game Play Con't

Post() WarGame-StateTuple
Read() Opponents WarGame-StateTuple
Return opponents WarGame-StateTuple
Get BigInteger representations of both played cards.
Return BigIntegers
Have SecureDeck Remove Decryption
Return BigIntegers with only opponent lock in place.

Get BigInteger representations of both played cards.
Return BigIntegers
Have SecureDeck Remove Decryption
Return BigIntegers with only opponent lock in place.

Read() Opponents WarGame-StateTuple
Return opponents WarGame-StateTuple
**WG Game Play End**

**Sequence Diagram**

- **WarGame (1)**
  - Post a `DecryptedTuple` with both cards ready for opponent.
  - Return `DecryptedTuple`
  - Calculates winner of round. Process
  - Repeats until one player runs out of cards.

- **SecureDeck**
  - Read()
  - `DecryptedTuple` posted by opponent.

- **TupleBoCard**
  - Read()
  - `DecryptedTuple` posted by opponent.
  - Return `DecryptedTuple`

- **WarGame (2)**
  - Post a `DecryptedTuple` with both cards ready for opponent.
  - Calculates winner of round.
  - Process
  - Repeats until one player runs out of cards.
WG Classes Con't
Texas Hold 'em Game

- Community card game
- Each player is dealt two cards (bet)
- Flop 3 community cards (bet)
- Turn community card (bet)
- River community card (bet)
- Showdown using everyone's best 5 cards.
Texas Hold 'em
Texas Hold 'em Game

- Opp Cards: H 4, H 9, C Ace, H 2, D Queen, H 3, D 10,
- Opp Cards:
- My cards: H 4, H 9, C Ace, H 2, D Queen, D 9, H 8,

Stop Playing  Keep Playing

Someone seems to have lost connection. Unless all devices can reconnect you will have to Quit and start a new session. Sorry.
Texas Hold 'em Initialization

Request 2 Cards be dealt to all players

Post OrderTuple

Read HandTuple

Return HandTuple

Post GetDecrypted Tuple for card

Post GetDecrypted Tuple

Read OrderTuple

Read OrderTuple

Return OrderTuple

Return OrderTuple

start getting card decrypted
Texas Hold 'em Decrypting Hand

```
Request
GetDecrypted Tuple Iterator
Return iterator
iterator.read()
Return GetDecrypted Tuple
Use Decryption Key
Return card with key removed
Post GetDecrypted Tuple

Request
GetDecrypted Tuple Iterator
Return iterator
iterator.read()
Return GetDecrypted Tuple
Use Decryption Key
Return card with key removed
Post GetDecrypted Tuple
```
Texas Hold 'em Decrypting Hand
Con't
Texas Hold 'em Game Play
Texas Hold 'em Getting Bets
Betting Con't

If opponent did not bet as much
Post TellPlayerTurn Tuple (2)
Read TotalBet Tuple
Return TotalBet Tuple

Read TellPlayerTurn Tuple
Return TellPlayerTurn Tuple
Post TotalBet Tuple
Community Cards

Diagram:

1. **Poker (1)**
   - Post **TellPlayerTurn**
   - Post **CommunityCard**
   - Post **CommunityCard**
   - Post **CommunityCard**

2. **TupleBoard**
   - Read **CommunityCardTuple**
   - Read **CommunityCard**
   - Read **CommunityCard**

3. **Poker (2)**
   - Return **CommunityCard**
   - Return **CommunityCard**

4. **SecureDeck**

Instructions:

- **Go through the same decryption that was used for the cards in HandTuple**
- **Read in all 3 community cards like (2)**
Hold 'em

- Game continues
- Betting
- Turn like flop
- Betting
- River like flop
- Betting
Finding Winner

Poker (1) -> TupleBoard
  Post TellPlayerTurn Tupel (Hands)
  Post HandResult Tupel
  Read HandResult Tupel
  Read opponents HandResult Tupel
  Return Tupel

TupleBoard -> Poker (2)
  Read TellPlayerTurn Tupel
  Return TellPlayerTurn Tupel
  Post TellPlayerTurn Tupel (Hands)
  Post HandResult Tupel
  Read opponents HandResult Tupel
  Return Tupel

Poker (2) -> SecureDeck

SecureDeck
Dealer Transition

- **PokerStart (1)**
- **Poker (1)**
- **TupleBoard**
- **PokerStart (2)**
- **Poker (2)**

1. **Post KeepPlaying Tuple**
   - Read KeepPlaying Tuple
   - Return Tuple

2. **Create New HoldEm() object should not be dealer this time**

3. **Read Dealer Tuple**
   - Return Tuple
   - **Create New HoldEm() object should be dealer this time**
Hold 'em Classes Initialization

GameSelFrame
- +create(): GameSelFrame
- Extends Frame
- Extends GameSelFrame

SessionSelectUI
- +create(): SessionSelectUI
- addListener(listener: SessionSelectUIListener)
- addSessionToList(session: String)
- removeSessionFromList(session: String)
- updateUser(msg: String)

PokerStart
- +createSessionClicked(sessionName: String, password: String, list: List)
- +joinSessionClicked(sessionToJoin: String, sessionPassword: String)
- +joinSetupOkButton(money: int)
- +keepPlayingClicked(turn: int, myMoney: int, lowPoint: Frame, keepPlayingFrame)
- +posted(tuple: Tuple)
- +setupOkButtonClicked(numPlayers: int, defaultMoney: int, smallBlind: int)
- +stopPlayingClicked(turn: int, frame: KeepPlayingFrame)
- +withdrawn(tuple: Tuple)

SessionSelectUIListener

PokerSetupUIListener

PostWithdrawListener

ActionListener

JoinSetupUIListener

KeepPlayingFrameListener

TupleBoard

create(): Calls PostWithdrawListener

Initialization Con't

[Diagram of a UML sequence diagram showing interactions between entities such as 'PokerStart', 'JoinSetupUI', 'HoldEmGame', '等方面的交互逻辑和事件触发，例如'createSessionClosed', 'joinSessionClosed', 'createSession', 'setupUIClosed', 'createSetupUI', 'setupUI', 'withdrawalListener', 'click', 'getAmount', 'getPosition', 'getRound', 'isPlayer', 'run', etc.]}
## CPU Stats

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell</td>
<td>Intel Dual Core 1.66 GHz</td>
<td>1.5 GB</td>
</tr>
<tr>
<td>IBM</td>
<td>Pentium 3 866 MHz</td>
<td>384 MB</td>
</tr>
<tr>
<td>iMac</td>
<td>Mac G4 800 Mhz</td>
<td>256 MB</td>
</tr>
<tr>
<td>Phone</td>
<td>ARM 208 Mhz</td>
<td>61 MB (16.6 Free after startup)</td>
</tr>
</tbody>
</table>
## Performance

- **War times in seconds**

<table>
<thead>
<tr>
<th>Round</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell vs Dell</td>
<td>Instant</td>
<td>Instant</td>
<td>Instant</td>
<td>Instant</td>
<td>Instant</td>
<td>Instant</td>
<td>Instant</td>
<td>Instant</td>
</tr>
<tr>
<td>Dell vs IBM</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Phone vs Phone</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Phone vs Dell</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
## Performance

- **Poker times in seconds**

<table>
<thead>
<tr>
<th>Round</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone and Dell</td>
<td>161</td>
<td>168</td>
<td>171</td>
<td>225</td>
<td>277</td>
<td>Phone Froze</td>
</tr>
<tr>
<td>Dell and Dell</td>
<td>53</td>
<td>16</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Dell and IBM</td>
<td>55</td>
<td>43</td>
<td>45</td>
<td>43</td>
<td>45</td>
<td>42</td>
</tr>
<tr>
<td>Dell, iMac Phone</td>
<td>322</td>
<td>Phone Froze</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dell Imac IBM</td>
<td>88</td>
<td>80</td>
<td>69</td>
<td>81</td>
<td>79</td>
<td>80</td>
</tr>
<tr>
<td>Phone Phone</td>
<td>Phone Froze</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question/Demo

• Questions?
• Demo
  - War Emulator vs Computer
  - War Phone vs Phone
  - War Phone vs Computer
  - Hold 'Em Computer vs Computer, Emulator
  - Hold 'Em Emulator vs Phone
  - Hold 'Em Phone vs Phone