Solution to the Third COSC 6360 Quiz for Fall 2013

Jehan-François Pâris
jfparis@uh.edu
First question

- In the Kerberos system, which entities share these secret keys or passwords? (5 points per correct line, no partial credit)
### Answer

<table>
<thead>
<tr>
<th>Secret</th>
<th>User's WS</th>
<th>Kerberos</th>
<th>TGS</th>
<th>Server S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**User's password**

| X |   |   |   |   |

**Secret key of TGS**

|   |   |   |   |   |

**Secret key of server S**

<p>| | | | | |
|   |   |   |   |   |</p>
<table>
<thead>
<tr>
<th>Secret</th>
<th>User’s WS</th>
<th>Kerberos</th>
<th>TGS</th>
<th>Server S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User’s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secret key of TGS</td>
<td>_____</td>
<td><strong>X</strong></td>
<td><strong>X</strong></td>
<td></td>
</tr>
<tr>
<td>Secret key of server S</td>
<td>_____</td>
<td>_____</td>
<td>_____</td>
<td></td>
</tr>
</tbody>
</table>
### Answer

<table>
<thead>
<tr>
<th>Secret</th>
<th>User’s WS</th>
<th>Kerberos</th>
<th>TGS</th>
<th>Server S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **User’s password**: 
  - Secret: __X__
  - User’s WS: __X__
  - Kerberos: __X__
  - TGS: ___
  - Server S: ___

- **Secret key of TGS**: 
  - Secret: _____
  - User’s WS: __X__
  - Kerberos: __X__
  - TGS: ___
  - Server S: ___

- **Secret key of server S**: 
  - Secret: _____
  - User’s WS: ___
  - Kerberos: ___
  - TGS: ___
  - Server S: __X__ __X__
Second question

- What is the function of the *i-node map* in a log-structured file system? (10 points)
- Where and how is it stored on the disk? (5 points)
The i-node map contains the **addresses** of the i-node blocks.
- Required because i-nodes do not reside at fixed positions on the disk.

The i-node map is stored **on the log** along with the data blocks, the directory blocks and the i-node blocks.
- **Not at a fixed location!**
What it means

Fixed location but out-of-date

Checkpoint Area

I-node map blocks spread on the log

Log

I-node blocks also spread on the log

Log
Third Question

- Consider a RAID-5 array having four data blocks, namely, $b_0$, $b_1$, $b_2$, and $b_3$, and one parity block $p$ per stripe.

- Assuming that block $b_3$ suddenly becomes unavailable, how could you reconstruct its contents?
$b_3 = b_0 \oplus b_1 \oplus b_2 \oplus p$
Fourth question

- It would allow intruders to \textit{replay} tickets of legitimate users
Fifth question

- What is the purpose of the BitTorrent *rarest first* rule? (10 points)
- When does it *not* apply? (5 points)
Answer

- The rarest first policy ensures that each downloader fetches first the pieces that most of its peers want.
- It does not apply to downloaders that have not yet downloaded their first piece.
Sixth question

- What is the purpose of ticket transfers in lottery scheduling? (10 points)

- Which problem do they solve? (5 points)
Ticket transfers provide explicit transfers of tickets from one client to another

- When a client waits for a reply from a server, it can temporarily transfer its tickets to that server

They eliminate priority inversions
Seventh question

- According to Shah et al., what is the main motivation for their *randomized tit-for-tat* policy? (10 points)
Answer

- Randomized tit-for tat lets each peer select neighbors at random at the beginning of every playback
  - Results in faster diffusion of new chunks among peers **OR**
  - Gives more free tries to a larger number of peers in the swarm to download chunks
Eighth question

- What are the main property and the main use of SHA-1 signatures? (10 points)
Answer

- SHA-1 is a cryptographic hash function
- It guarantees that any change to the hashed data will (with very high probability) change the hash value
- It is used to verify the *integrity* of SSH packets