CISC 611-90-O-2019/Late Spring - Network Operating Systems
Homework - 3

Youwei Lu

Exercise 19.15: show how to construct a send-constrained channel from a receive-constrained channel, and vice versa. Hint: use a trusted node connected to the given channel.

The following nomenclatures are used in the solution:

- c: Any channel connecting the parties that to be able to communicate
- h: Hash
- m: Message
- N: Trusted node
- t: Time by N's clock
- rc: Receive-constrained channel
- sc: Send-constrained channel

With the nomenclatures used, the problem can be rephrased as: given a rc, show how to construct a $s(rc)$.

Use a N. when it receives a m, it uses the received-constrained channel rc to return to the sender a signed hash $\text{sig}\{h(m,t)\}$. Then construct $s(rc)$ from c and N. The rules for sending and receiving on $s(rc)$ are as follow:

To send m on $s(rc)$:

send m to N

receive $< t$, $\text{sig}\{h(m,t)\} >$ from N over rc

send $< m, t,$ sig$\{h(m,t)\} >$ on c.

To receive m on $s(rc)$:

receive $< m, t, h >$ on c
verify $h =$ sig$\{h(m,t)\}$
verify currency of t and freshness of h
Discard m if verification fails, else receive m.

It is assumed that the receivers' clocks are synchronized to N's clock. The timestamp (and hence state of the sender) is deemed current if it is within a given bound of the time on the receiver's clock. To prevent replay attacks, the receiver need remember the hashes for only a limited time: older messages will have a non-current timestamp.

In a similar fashion, a receive-constrained channel $r(sc)$ can be implemented from a send-constrained channel sc. All messages are sent (over any channel) to a trusted node, which stores them. Receivers must use a particular send-constrained channel to reach that node, which responds with the next message for them.