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

Construct a send-constrained channel from a receive-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 receive-constrained channel rc to return to the sender a signed hash $\text{sig}\{h(m,t)\}$. Then construct $s(rc)$ from c and N.

Assuming the receivers' and N's clocks are synchronized, so the timestamp can be treated as current within a certain bound of the receiver's time. The followings are the steps on sending and receiving m on $s(rc)$:

To send m on $s(rc)$:

send m to N
receive $< t$, 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 for verification fails, otherwise receive m.

Construct a receive-constrained channel from a send-constrained channel

Using the same way, a receive-constrained channel can be implemented from a send-constrained channel. Again, the problem may be rephrased as: given a sc, show how to construct a $r(sc)$. Use a trusted node N to receive and store all messages from any channels. Then the receivers use a send-constrained channel to reach N, and receive the next message from N.