**Additional Topics 2004 – Paper 7 Question 16 (AH)**

(relates to lectures 6-8 by F Stajano)

($a$) Source broadcasts a stream of messages.
Sender authentication is required.
Signing every message is too expensive.
Threat model includes active attacker in the network.
Hash unsuitable because attacker can recompute it.
MAC unsuitable for broadcast because every recipient must know the key.

($b$) Overview: Source sends a chain of packets Pi, linked by forward and backward authenticators.

Bootstrap: first packet is authenticated by signing. Steady state: can't change a packet without prior packets revealing it. Diagram: see Stajano textbook p 125.

The attacker can't forge the payload Mi: the MAC will reveal that. The attacker can't recompute the MAC after forging: the key Ki isn't there yet (only revealed in Pi+1).

The attacker can't make up a fake Ki and substitute it in Pi+1: the earlier, genuine packet Pi-1 contains a commitment to (hash of) the correct key Ki.

($c$) ($i$) Main problem: if recipient R receives Pi after Pi+1 has been sent, then R cannot validate Pi. In a broadcast environment, sender S can never be sure about when (whether?) all the recipients got Pi. So S must wait for a long time between packets. Therefore, limitation on packet rate and hence on throughput.

($ii$) If long interpacket delay, there is also a long delay before R can validate the previous packet. The data might be stale by then.

($iii$) If recipient R misses a packet, the forward commitment chain is broken. R can no longer validate any future packets.

($iv$) R must have listened to packet 0 (signed bootstrap) to validate any others. So one can only join the broadcast at the start.

($d$) ($i$) TESLA transmits Ki in Pi+d instead of Pi+1. So there is more time for the authenticator to arrive while still fresh, and there are more packets in transit ie greater throughput.

($ii$) TESLA uses several authenticators per packet: Ki1 revealed in Pi+d1, Ki2 in Pi+d2, Ki3 in Pi+d3 etc. Recipient R can validate packet Pi with any of these that were sent before R received Pi.

(*iii*) TESLA uses a Lamport hash chain to generate the keys, so any previous key can be used to validate (but not predict) any subsequent key.

(*iv*) No solution offered. As in Guy Fawkes, the broadcast stream must be periodically re-bootstrapped with a signature.