Timestamps in Security Protocols

- One method of handling this kind of problem is timestamps
- Proper use of timestamps can limit the time during which an exposed key is dangerous
- But timestamps have their own problems

Using Timestamps in the Needham-Schroeder Protocol

- The trusted authority includes timestamps in his encrypted messages to Alice and Bob
- Based on a global clock
- When Alice or Bob decrypts, if the timestamp is too old, abort the protocol

Using Timestamps to Defeat Mallory

$\boldsymbol{K_S}$

$E_{K_B}(K_S, \text{Alice}, T_X)$

$\boldsymbol{K_B}$

Mallory

Bob

$\boldsymbol{T_X}$

$E_{K_B}(K_S, \text{Alice}, T_X)$

$\boldsymbol{T_X \ll T_{now}}$

$\boxed{\boldsymbol{T_{now}}}$

Now Bob checks T_X against his clock

So Bob, fearing replay, discards $\boldsymbol{K_S}$

And Mallory's attack is foiled

Problems With Using Timestamps

- They require a globally synchronized set of clocks
  - Hard to obtain, often
  - Attacks on clocks become important
- They leave a window of vulnerability

The Suppress-Replay Attack

- Assume two participants in a security protocol
  - Using timestamps to avoid replay problems
- If the sender's clock is ahead of the receiver's, attacker can intercept message
  - And replay later, when receiver's clock still allows it

Handling Clock Problems

1). Rely on clocks that are fairly synchronized and hard to tamper with

– Perhaps GPS signals

2). Make all comparisons against the same clock

– So no two clocks need to be synchronized

Is This Overkill?

- Some of these attacks are pretty specialized
  - Requiring special access or information
- Some can only achieve certain limited effects
- Do we really care?

Why Should We Care?

- Bad guys are very clever
- Apparently irrelevant vulnerabilities give them room to show that
- Changes in how you use protocols can make vulnerabilities more relevant
- A protocol without a vulnerability is always better
  - Even if you currently don’t care

Something to Bear in Mind

- These vulnerabilities aren’t specific to just these protocols
- They are common and pop up all over
  - Even in cases where you aren’t thinking about a “protocol”
- Important to understand them at a high conceptual level