Authentication Specifications

IT 862
What is Authentication?

- **Publication**: A Hierarchy of Authentication Specifications by Gavin Lowe. Available at http://web.comlab.ox.ac.uk/oucl/work/gavin.lowe/Publications.html,
  - Paper #10
Notations for Authentication

- The Initiator (A): begins the protocol
- The responder (B): responds to the request
- Attacker (C):
- Agents: initiator, responder, server, attacker
- Role of an agent: their function in the protocol
Agents and Roles

- An agent may be an initiator in one protocol, a responder in another and a server in a third.
Many definitions of authentication

- A runs a protocol apparently with B
- What can A deduce about B?
  - B has been running the same protocol?
  - That B thinks it is running the same protocol with A? (as opposed to some other C?)
  - Is there a one-to-one relationship between A’s run and B’s run?
  - Did A and B agree on some value of an attribute?
Aliveness

- The Weakest Version: A protocol guarantees the aliveness of B to A, whenever A completes a run apparently with B, then B has been running a protocol previously.

- Notes:
  - B may not have believed that (s)he ran the same protocol with A.
  - Perhaps not recently!
Failures in achieving Aliveness

- An intruder launches a mirror attack.
- An attacker runs two versions of the same protocol with A and uses data from the first to respond to the second.
- Other attacks:
  - A completes a run, B was present but running another/same protocol with someone else
  - B was running a different protocol with A
Weak Agreement

- A protocol guarantees to A a week agreement with B whenever A completes the run apparently with B, B has previously been running the protocol apparently with A.

Notes:
- B may not have participated necessarily as a responder!
Liveliness but not weak agreement

- Attacker imitates B to A by using B as an oracle – adopts B’s identity
- A believes it is running the protocol with B
- But B does not believe it is running protocol with A
- Example: Low’s attack on Needham-Schroeder protocol.
Non-injective agreement

- A has a \textcolor{blue}{non-injective agreement with B on a set of data items \textcolor{red}{ds} (a set of free variables)} iff whenever A completes a protocol run apparently with B, the B has been previously running the protocol apparently with A,
- B acts as a responder
- A and B agree to all data values corresponds to variables in \textcolor{red}{ds}. 
(injective) Agreement

- A protocol guarantees to A an agreement with B on a set of data items \( ds \) iff whenever A completes a run apparently with B, then B as a responder has previously ran a protocol apparently with A and
- The two agents agreed on the data value corresponding to all variables in \( ds \)
- Each run of A corresponds to a unique run of B
non-injective but non-full agreement

- Agent A is tricked into believing that B wants to establish two sessions with A
- In the original Kerberos protocol, the freshness was only guaranteed by a time stamp.
- Did not check that all timestamps were different from previous ones.
Recentness

- What does recentness mean?
- Usually, an implementation dependent timeout – orthogonal to previous definitions.
- So we can have recent aliveness, recent non-injective agreement etc.
- Example: meeting on-injective agreement without being recent.

Message 1: \( A \rightarrow B : \{A, k\}_{k \_ \{AB\}} \)
Intensional Specifications

- Bill Roscoe’s: No node can believe a protocol run has completed unless a correct series of messages have occurred up to and including the last

- Stronger than full agreement!
  - If A runs a protocol apparently with B, then A’s version of the run must agree with that of B, including data values
  - Hence full agreement
Intensional > full agreement

- A receives encrypted msg from B, passes to D.
- Attacker: replaces one encrypted msg with another
  - Not an attack under full agreement, but under intensional correctness
- Server sends msg 1 and 2 in order. Attacker swaps them.
  - Not an attack under full agreement, but under intensional correctness
Intensional does not guarantee recentness

- **Consider** Message1: $A \rightarrow B : \{A,k\}_{k\{AB\}}$
- **Does not guarantee recentness but is intensionally correct!**
- **Sometimes, recentness is indirectly guaranteed by ordering.**
  - If second msg arrived then first must have.
  - Indirect timeout built into intensional correctness!
What is the point?

- Need to precisely specify the security objectives of the protocol.
- That is, need to specify the security objectives in a formal language.
- Need to prove that the protocol satisfy its security objectives.
- Lowe used CSP to specify authentication goals.
- Switch to an introduction to CSP.