The Modelling and Analysis of Security Protocols
Notes for Lectures 3 & 4

Tom Chothia  T.Chothia@cwi.nl

I. PROTOCOL GOALS

Common goals for security protocol include:

- Key Freshness: the key established is new.
- Key Exclusivity: the key is only known to the principals in the protocol.
- Good Key: the key is both fresh and exclusive.
- Far-end Operative: A knows that B is currently active.
- Once Authentication: A knows that B wishes to communicate with A.
- Entity Authentication: A knows that B is currently active and wants to communicate with A.
- Key Confirmation of A to B: if B can be sure that K is a good key to communicate with A and that A knows K.
- Mutual Belief in Key: is provided for B if K is a good key, A can be sure that K is a good key for B, and A wishes to communicate with B using K.

“Good Key” is the best you can do without the principals sharing a key or secret to start off with, or having each other’s public keys. “Key Confirmation” is often enough to ensure that there are no useful attacks on the protocol.

A. Example protocol: Diffie-Hellman

A and B pick random numbers \( r_A \) and \( r_B \) and calculates \( t_A = g^{r_A} \) and \( t_B = g^{r_B} \):

The protocol just exchanges these numbers:

1) \( A \rightarrow B : t_A \)
2) \( B \rightarrow A : t_B \)

A calculates the key as \( t_A r_A \mod p \) and B calculates it as \( t_A r_B \mod p \). This results in the same key: \( K = g^{r_A r_B \mod p} = t_A r_A \mod p = t_A r_B \mod p \).

An observer cannot work out \( r_A \) and \( r_B \) from \( t_A \) and \( t_B \) therefore the attacker cannot calculate the key. The values of \( g \) and \( p \) must be big enough to make it intractable to try all possible combinations (often 1024 bits and 160 bits).

Diffie-Hellman results in a “Good Key” but provides no information about which principals have this good key.

II. TYPES OF ATTACK

Some common types of attacks on protocols include:

Eavesdropping: An Eavesdropping attack only passively observes messages

Modification: A Modification attack alters or generates some messages

Replay / Preplay: Attacker sends a message from another part of the protocol

Man-in-the-middle: The attacker acts in between a real run of the protocol and repeating the challenge back.

Reflection: Reflection attacks prove authentication by starting another run of the protocol and repeating the challenge back.

Denial of Service: A DoS attack tries to use up all of a severs resources.

Typing Attack: In a typing attack the attacker passes off one type of message as being another e.g. a nonce for a key.

III. GOOD PROTOCOL DESIGN

The 11 principles of Abardi and Needham are:

- Principle 1: Every message should say what it means: the interpretation of the message should depend only on its contents.
- Principle 2: The conditions for a message to be acted upon should be clearly set out.
- Principle 3: If the identity of a principal is essential to the meaning of a message, it is prudent to mention the principal’s name in the message.
- Principle 4: Be clear about why encryption is being done.
- Principle 5: When a principal signs material that has already been encrypted, it should not be inferred that the principal knows the content of the message.
- Principle 6: Be clear what properties you are assuming about nonces.
- Principle 7: The use of a predictable quantity can serve in guaranteeing newness, through a challenge-response exchange. But it should be protected so that an intruder cannot simulate a challenge and later replay the response.
- Principle 8: If timestamps are used as freshness guarantees then the difference between local clocks at various machines must be less than the allowable age of a message deemed to be valid.
- Principle 9: A key may have been used recently yet be quite old, and possibly compromised.
- Principle 10: If an encoding is used to present the meaning of a message, then it should be possible to tell which encoding is being used.
- Principle 11: The protocol designer should know which trust relations his protocol depends on, and why the dependence is necessary.

and not forgetting “Principle 0”: The protocol must be efficient