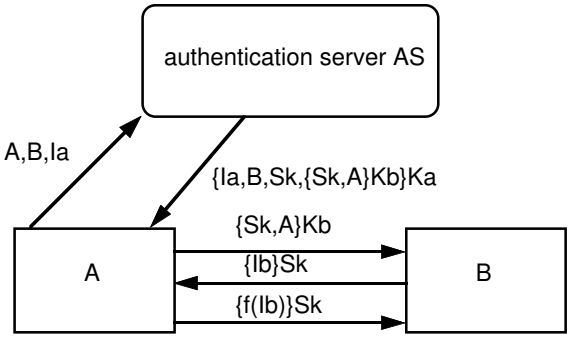
Needham-Schroeder Protocols

- Used for authentication in a network
- Require an *authentication server* that can supply requested information from a secret key
- Protocols for both classical, public-key cryptosystems
- Minor bug, subsequently fixed by Denning and Sacco

(*CACM* **21**(12) pp. 993-999, Dec. 1978)

NS: Classical Cryptosystem



NS: Classical Cryptosystem (2)

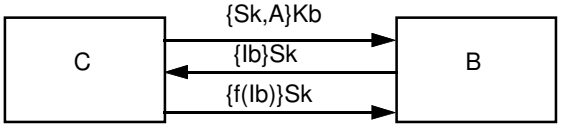
Ia is an identifier used by A exactly once (a *nonce*)

- B knows message came via AS as it is encrypted in a key known only to AS and B
- B knows session key Sk was produced by AS for the same reason
- B knows AS is vouching that the other party is A
- B knows this is not a replay due to thew Ib handshake

As B trusts AS, A knows communication came from A, and B can use the session key Sk

NS: Flaw in Classical

Suppose C obtains Sk and records {SK,A}Kb. Then



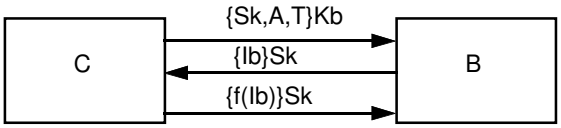
and B now thinks it is talking to A!

Fix: assign f to pairs and keep secret.

How: if done via AS, same as key problem; if done directly, swap keys directly and forget about this protocol

NS: Denning and Sacco Modification

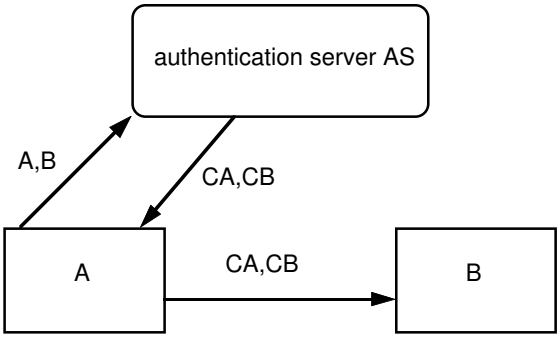
All encrypted messages include a timestamp T. Then



Now B checks that T is “acceptably close” to its current time (this being no more than the server's clock drift wit respect to the local clock, plus any network delay)

Note: AS also sends A {T, B, Sk, {Sk, A, T}Kb}Ka

NS: Public Key Cryptosystem



NS: Public Key Cryptosystem (2)

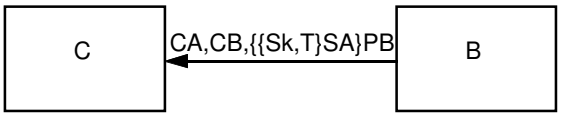
CA is A's certificate, signed by the authentication server

- CA contains an identity (A), a timestamp, and a public key, and is encrypted using AS's secret key; all components are visible, though
- To validate, B sends certificate in question to AS

This distributes public keys

NS: Session Key Transmission

Assume AS has obtained B's public key PB



Now B can decrypt {{Sk,T}SA}PB as it has SB, B's secret key, and PA, A's public key; no-one else can, though

This does *not* fix flaws, but merely simplifies original Needham-Schroeder protocol

X.509 Authentication Protocols

- Used for authentication in a network
- Three types:
  - one-way authentication (one transmission)
  - two-way authentication (transmission + reply)
  - three-way authentication (transmission + reply + ack)
- Assume Alice has Bob's certificate (public key) and has validated it; Bob has a certification path to Alice

(*Blue Book Volume VIII – Data Communications Networks Directory Recommendation X.509*)

X.509: General Token Structure

All involve a token of the form

A{T, IA, B, IB, sgnData, {encData}PB}

where:

- T is a timestamp, and IA, IB are nonces
- sgnData is authentication checks depending only on the data
- {encData}PB is encryption of confidential parameters with B's public key
- A{…} is a digital signature of what is between {…} appended to the token

X.509: Token Structure Problems

Suppose A sends the following token to B, with the data encrypted and the key in the encData field and an integrity check in the sgnData field:

A{T, IA, B, IB, sgnData, {encData}PB}data

C intercepts the message, sends to B:

C{T',IC,B, IB, sgnData, {encData}PB}data

B believes data came from C, and will reply to C; from the reply, C may be able to derive A's original request!

Problem: contents of *token* not integrity-checked

X.509: Token Structure Fix

Obvious solution: sign the token contents:

A{T, IA, B, IB, sgnData, {encData}PB, {hash}SA}data

where {hash}SA is the one-way hash of the rest of the token contents, signed using A's private key

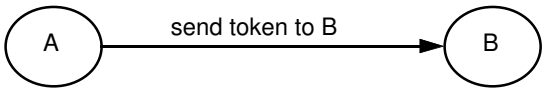
- C cannot replace the first two fields as he cannot recompute the last token field, not knowing A's secret key

X.509: One-Way Authentication

Goal of one-way authentication protocol: to establish …

- identity of A
- A generated authentication token
- identity of B
- authentication token intended for B
- integrity of authentication token
- authentication token is not a replay

X.509: One-Way Authentication (2)



- A sends B the token

  A,A{T,IA,B,sgnData,{encData}PB}

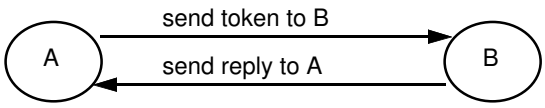
- B validates it as follows:
  - obtains A's public key being sure it's not expired)
  - validates the signature of the token
  - checks B is the intended recipient
  - checks the timestamp T is current
  - checks that the message is not a replay (see if IA is a repeat)

X.509: Two-Way Authentication

Goal of two authentication protocol: to establish …

- everything for one-way protocol, plus:
- authentication token in reply generated by B
- authentication token in reply intended for A
- integrity of authentication token in reply
- authentication token in reply is not a replay
- (if desired) secrecy of part of the authentication tokens

X.509: Two-Way Authentication (2)



- A sends B the token

  A,A{T,IA,B,sgnData,{encData}PB}
- B validates it as for the one-way case
- B sends A the token

  B{T',IB,A,IA,sgnData,{encData}PA}
- A validates it as for the one-way case

X.509: Three-Way Authentication

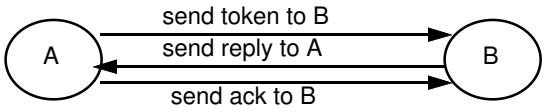
Goal of two authentication protocol: to establish …

- everything for two-way protocol

but also eliminate the need for time stamp checking

Why? Useful if clocks drift a lot, or clock synchronization is a problem

X.509: Three-Way Authentication (2)



- A sends B the token (possibly setting T to 0)

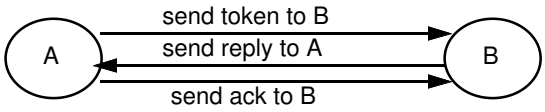
  A,A{T,IA,B,sgnData,{encData}PB}

- B validates it as for the one-way case (possibly omitting timestamp check)
- B sends A the token (possibly setting T' to 0)

  B{T',IB,A,IA,sgnData,{encData}PA}

- A validates it as for the one-way case (possibly omitting timestamp check)

X.509: Three-Way Authentication (3)



- A checks that the IA sent initially is the same as the IA received in the reply
- A sends B the token

  A{IB}

- B checks the validity of the ack token as follows:
  - validate the signature, and thus the integrity, of the ack token
  - check that the IB sent in the reply token matches the IB received in the ack token

X.509: Three-Way Flaw

Suppose C wishes to impersonate A to B, and has recorded one interchange of the three-way protocol (*ie*, all three tokens):

- C sends to B the first token

  A,A{0,IA,B,sgnData,{encData}PB}

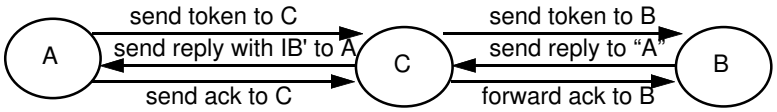
- B responds (thinking it is talking to A, but really talking to C) with a new nonce IB'

  B{0,IA,B,IB',sgnData,{encData}PA}

- C starts talking to A, and A sends

  A,A{0,IA',C,sgnData,{encData}PC}

X.509: Three-Way Flaw (2)



- C responds to A using IB'

  C{0,IA',C,IB',sgnData,{encData}PA}

- A responds to C by

  A{IB'}

- C sends B

  A{IB'}

X.509: Three-Way Fix

Problem: last token (the ack one) has no *signed* indication of sender, so anyone can simply forward it to B.

Solution: add the origin in the signed token.

Now:

- A sends B the token

  A{IB, B}

- B validates as before, but also:
  - check that B is the intended recipient of the ack token