Access Control and Privacy Policies (8)

Email: christian.urban at kcl.ac.uk
Office: S1.27 (1st floor Strand Building)
Slides: KEATS (also homework is there)

Last Week

Andrew Secure RPC Protocol: A and B share a key K_{AB} and want to identify each other

- A sends B : A, N_A
- B sends A : $\{N_A, K'_{AB}\}_{K_{AB}}$
- A sends B : $\{N_A\}_{K'_{AB}}$

Protocols

A sends B : ...

- by convention A, B are named principals Alice... but most likely they are programs, which just follow some instructions

Protocols

$$A \text{ sends } B : \ldots$$
$$B \text{ sends } A : \ldots$$
$$\vdots$$

- by convention A, B are named principals Alice... but most likely they are programs, which just follow some instructions

- indicates one "protocol run", or session, which specifies some order in the communication
- there can be several sessions in parallel (think of wifi routers)

Last Week

A and B share the key K_{AB} and want to identify each other

- A sends B : A, N_A
- B sends A : $\{N_A, K'_{AB}\}_{K_{AB}}$
- A sends B : $\{N_A\}_{K'_{AB}}$

Defeating Challenge-Response

A reflection attack: an intruder I impersonates B.

A sends $I : A, N_A$

Defeating Challenge-Response

A reflection attack: an intruder I impersonates B.

A sends $I : A, N_A$ I sends $A : B, N_A$

Defeating Challenge-Response

A reflection attack: an intruder I impersonates B.

A sends I : A, N_A

I sends A : B, N_A
A sends I : $\{N_A, K'_{AB}\}_{K_{AB}}$

Defeating Challenge-Response

A reflection attack: an intruder I impersonates B.

A sends I : A, N_A	I sends A : B, N_A
I sends A : $\{N_A, K'_{AB}\}_{K_{AB}}$	A sends I : $\{N_A, K'_{AB}\}_{K_{AB}}$

Defeating Challenge-Response

A reflection attack: an intruder I impersonates B.

A sends I : A, N_A
I sends A : $\{N_A, K'_{AB}\}_{K_{AB}}$
A sends I : $\{N_A\}_{K'_{AB}}$

I sends A : B, N_A
A sends I : $\{N_A, K'_{AB}\}_{K_{AB}}$

Defeating Challenge-Response

A reflection attack: an intruder I impersonates B.

A sends I : A, N_A	I sends A : B, N_A
I sends A : $\{N_A, K'_{AB}\}_{K_{AB}}$	A sends I : $\{N_A, K'_{AB}\}_{K_{AB}}$
A sends I : $\{N_A\}_{K'_{AB}}$	I sends A : $\{N_A\}_{K'_{AB}}$

Defeating Challenge-Response

A reflection attack: an intruder I impersonates B.

A sends I : A, N_A	I sends A : B, N_A
I sends A : $\{N_A, K'_{AB}\}_{K_{AB}}$	A sends I : $\{N_A, K'_{AB}\}_{K_{AB}}$
A sends I : $\{N_A\}_{K'_{AB}}$	I sends A : $\{N_A\}_{K'_{AB}}$

Sounds stupid: "...answering a question with a counter question"

was originally developed at CMU for terminals to connect to workstations (e.g. file servers)

Identify Friend or Foe

198?: war between Angola (supported by Cuba) and Namibia (supported by SA)

Identify Friend or Foe

198?: war between Angola (supported by Cuba) and Namibia (supported by SA)

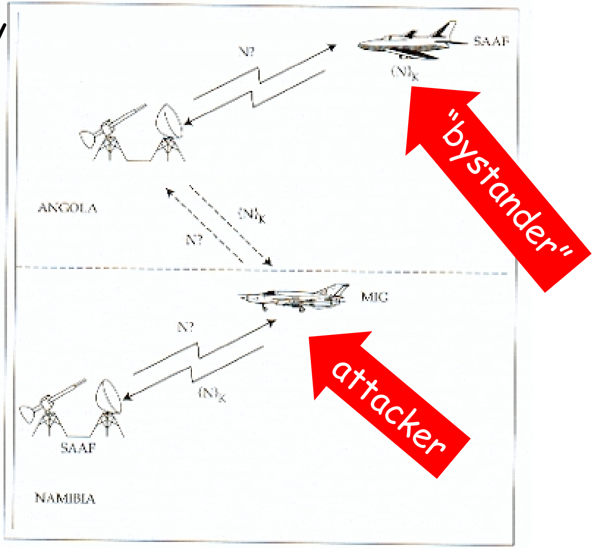


Figure 2.2 The MIG-in-the middle attack.

Identify Friend or Foe

198?: war between Angola (supported by Cuba) and Namibia (supported by SA)

being outsmarted by Angola/Cuba ended SA involvement

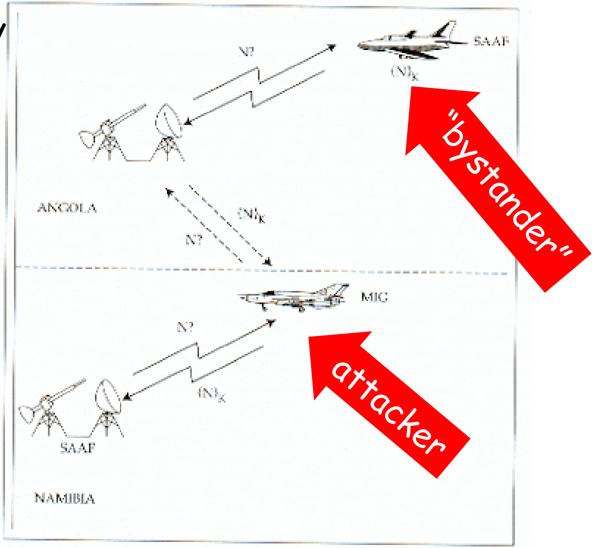


Figure 2.2 The MIG-in-the middle attack.

Identify Friend or Foe

198?: war between Angola (supported by Cuba) and Namibia (supported by SA)

being outsmarted by Angola/Cuba ended SA involvement

IFF opened up a nice side-channel attack

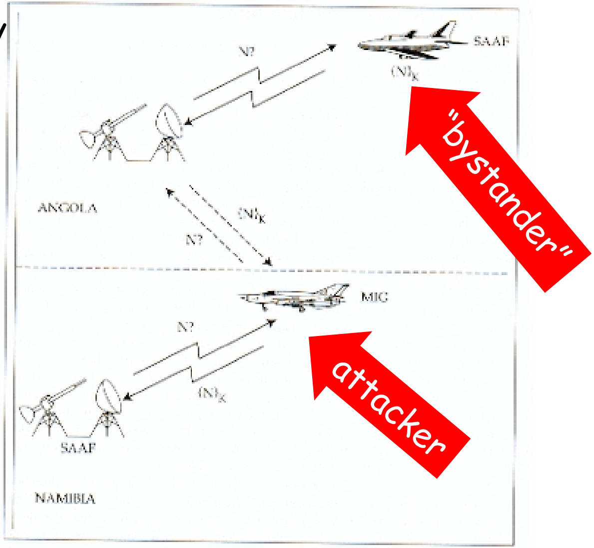


Figure 2.2 The MIG-in-the middle attack.

Encryption to the Rescue?

- A sends B : $\{A, N_A\}_{K_{AB}}$ encrypted
- B sends A : $\{N_A, K'_{AB}\}_{K_{AB}}$
- A sends B : $\{N_A\}_{K'_{AB}}$

Encryption to the Rescue?

- A sends B : $\{A, N_A\}_{K_{AB}}$ encrypted
- B sends A : $\{N_A, K'_{AB}\}_{K_{AB}}$
- A sends B : $\{N_A\}_{K'_{AB}}$

means you need to send a separate "Hello" signal (bad), or worse share a single key between many entities

Protocol Attacks

- replay attacks
- reflection attacks
- man-in-the-middle attacks
- timing attacks
- parallel session attacks
- binding attacks (public key protocols)
- changing environment / changing assumptions

Replay Attacks

Schroeder-Needham protocol: exchange of a symmetric key with a trusted 3rd-party S:

$$
\begin{array}{l}
A \rightarrow S : A, B, N_A \\
S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \\
A \rightarrow B : \{K_{AB}, A\}_{K_{BS}} \\
B \rightarrow A : \{N_B\}_{K_{AB}} \\
A \rightarrow B : \{N_B - 1\}_{K_{AB}}
\end{array}
$$

Replay Attacks

Schroeder-Needham protocol: exchange of a symmetric key with a trusted 3rd-party S:

$$
\begin{array}{l}
A \rightarrow S : A, B, N_A \\
S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \\
A \rightarrow B : \{K_{AB}, A\}_{K_{BS}} \\
B \rightarrow A : \{N_B\}_{K_{AB}} \\
A \rightarrow B : \{N_B - 1\}_{K_{AB}}
\end{array}
$$

at the end both A and B should be in the possession of the secret key K_{AB} and know that the other principal has the key

$$A \rightarrow S : A, B, N_A$$
$$S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$
$$A \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$$
$$B \rightarrow A : \{N_B\}_{K_{AB}}$$
$$A \rightarrow B : \{N_B - 1\}_{K_{AB}}$$

$$A \rightarrow S : A, B, N_A$$
$$S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$
$$A \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$$
$$B \rightarrow A : \{N_B\}_{K_{AB}}$$
$$A \rightarrow B : \{N_B - 1\}_{K_{AB}}$$

compromise K_{AB}

$$A \to S : A, B, N_A$$
$$S \to A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$
$$A \to B : \{K_{AB}, A\}_{K_{BS}}$$
$$B \to A : \{N_B\}_{K_{AB}}$$
$$A \to B : \{N_B - 1\}_{K_{AB}}$$

compromise K_{AB}

$$A \to S : A, B, N'_A$$
$$S \to A : \{N'_A, B, K'_{AB}, \{K'_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$

$A \rightarrow S : A, B, N_A$
$S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
$A \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$
$B \rightarrow A : \{N_B\}_{K_{AB}}$
$A \rightarrow B : \{N_B - 1\}_{K_{AB}}$

compromise K_{AB}

$A \rightarrow S : A, B, N'_A$
$S \rightarrow A : \{N'_A, B, K'_{AB}, \{K'_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
$I(A) \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$ replay of older run

$$A \to S : A, B, N_A$$
$$S \to A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$
$$A \to B : \{K_{AB}, A\}_{K_{BS}}$$
$$B \to A : \{N_B\}_{K_{AB}}$$
$$A \to B : \{N_B - 1\}_{K_{AB}}$$

compromise K_{AB}

$$A \to S : A, B, N'_A$$
$$S \to A : \{N'_A, B, K'_{AB}, \{K'_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$
$$I(A) \to B : \{K_{AB}, A\}_{K_{BS}}$$ replay of older run
$$B \to I(A) : \{N'_B\}_{K_{AB}}$$
$$I(A) \to B : \{N'_B - 1\}_{K_{AB}}$$

$$A \rightarrow S : A, B, N_A$$
$$S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$
$$A \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$$
$$B \rightarrow A : \{N_B\}_{K_{AB}}$$
$$A \rightarrow B : \{N_B - 1\}_{K_{AB}}$$

compromise K_{AB}

$$A \rightarrow S : A, B, N'_A$$
$$S \rightarrow A : \{N'_A, B, K'_{AB}, \{K'_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$
$$I(A) \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$$ replay of older run
$$B \rightarrow I(A) : \{N'_B\}_{K_{AB}}$$
$$I(A) \rightarrow B : \{N'_B - 1\}_{K_{AB}}$$

B believes it is following the correct protocol, intruder I can form the correct response because it knows K_{AB} and talk to B masquerading as A

Replay Attacks

Andrew Secure RPC protocol: exchanging a new key between A and B

$$
\begin{aligned}
A \rightarrow B &: A, \{N_A\}_{K_{AB}}\\
B \rightarrow A &: \{N_A + 1, N_B\}_{K_{AB}}\\
A \rightarrow B &: \{N_B + 1\}_{K_{AB}}\\
B \rightarrow A &: \{K_{AB}^{new}, N_B^{new}\}_{K_{AB}}
\end{aligned}
$$

Replay Attacks

Andrew Secure RPC protocol: exchanging a new key between A and B

$$
\begin{array}{l}
A \rightarrow B : A, \{N_A\}_{K_{AB}} \\
B \rightarrow A : \{N_A + 1, N_B\}_{K_{AB}} \\
A \rightarrow B : \{N_B + 1\}_{K_{AB}} \\
B \rightarrow A : \{K_{AB}^{new}, N_B^{new}\}_{K_{AB}}
\end{array}
$$

Assume nonces are represented as bit-sequences of the same length

$$
\begin{array}{ll}
A \rightarrow B : A, \{N_A\}_{K_{AB}} & \\
B \rightarrow A : \{N_A + 1, N_B\}_{K_{AB}} & \\
A \rightarrow I(B) : \{N_B + 1\}_{K_{AB}} & \text{intercepts} \\
I(B) \rightarrow A : \{N_A + 1, N_B\}_{K_{AB}} & \text{resend 2nd msg}
\end{array}
$$

Time-Stamps

The Schroeder-Needham protocol can be fixed by including a time-stamp (e.g., in Kerberos):

$$
\begin{array}{l}
A \rightarrow S : A, B, N_A \\
S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A, T_S\}_{K_{BS}}\}_{K_{AS}} \\
A \rightarrow B : \{K_{AB}, A, T_S\}_{K_{BS}} \\
B \rightarrow A : \{N_B\}_{K_{AB}} \\
A \rightarrow B : \{N_B - 1\}_{K_{AB}}
\end{array}
$$

Time-Stamps

The Schroeder-Needham protocol can be fixed by including a time-stamp (e.g., in Kerberos):

$$
\begin{aligned}
A &\rightarrow S : A, B, N_A \\
S &\rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A, T_S\}_{K_{BS}}\}_{K_{AS}} \\
A &\rightarrow B : \{K_{AB}, A, T_S\}_{K_{BS}} \\
B &\rightarrow A : \{N_B\}_{K_{AB}} \\
A &\rightarrow B : \{N_B - 1\}_{K_{AB}}
\end{aligned}
$$

but nothing is for free: then you need to synchronise time and possibly become victim to timing attacks

It can also be fixed by including another nonce:

$$
\begin{array}{l}
A \rightarrow B : A \\
B \rightarrow A : \{A, N_B\}_{K_{BS}} \\
A \rightarrow S : A, B, N_A, \{A, N_B\}_{K_{BS}} \\
S \rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A, N_B\}_{K_{BS}}\}_{K_{AS}} \\
A \rightarrow B : \{K_{AB}, A, N_B\}_{K_{BS}} \\
B \rightarrow A : \{N_B\}_{K_{AB}} \\
A \rightarrow B : \{N_B - 1\}_{K_{AB}}
\end{array}
$$

It can also be fixed by including another nonce:

$$
\begin{aligned}
A &\rightarrow B : A \\
B &\rightarrow A : \{A, N_B\}_{K_{BS}} \\
A &\rightarrow S : A, B, N_A, \{A, N_B\}_{K_{BS}} \\
S &\rightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A, N_B\}_{K_{BS}}\}_{K_{AS}} \\
A &\rightarrow B : \{K_{AB}, A, N_B\}_{K_{BS}} \\
B &\rightarrow A : \{N_B\}_{K_{AB}} \\
A &\rightarrow B : \{N_B - 1\}_{K_{AB}}
\end{aligned}
$$

but nothing is for free: then you need to synchronise time and possibly become victim to timing attacks

Binding Attacks

with public-private keys it is important that the public key is bound to the right owner (verified by a certification authority CA)

$$A \rightarrow CA : A, B, N_A$$
$$CA \rightarrow A : CA, \{CA, A, N_A, K_B^{pub}\}_{K_A^{pub}}$$

A knows K_A^{prig} and can verify the message came from CA in response to A's message and trusts K_B^{pub} is B's public key

Binding Attacks

$$A \to I(CA) : A, B, N_A$$
$$I(A) \to CA : A, I, N_A$$
$$CA \to I(A) : CA, \{CA, A, N_A, K_I^{pub}\}_{K_A^{pub}}$$
$$I(CA) \to A : CA, \{CA, A, N_A, K_I^{pub}\}_{K_A^{pub}}$$

Binding Attacks

$$A \to I(CA) : A, B, N_A$$
$$I(A) \to CA : A, I, N_A$$
$$CA \to I(A) : CA, \{CA, A, N_A, K_I^{pub}\}_{K_A^{pub}}$$
$$I(CA) \to A : CA, \{CA, A, N_A, K_I^{pub}\}_{K_A^{pub}}$$

A now encrypts messages for B with the public key of I (which happily decrypts them with its private key)

There are plenty of other protocols and attacks.
This could go on “forever”.

There are plenty of other protocols and attacks.
This could go on "forever".

attacks because of changing environment

Changing Environment Attacks

- all protocols rely on some assumptions about the environment (e.g., cryptographic keys cannot be broken)

Changing Environment Attacks

- all protocols rely on some assumptions about the environment (e.g., cryptographic keys cannot be broken)

- in the "good olden days" (1960/70) rail transport was cheap, so fraud was not worthwhile

Changing Environment Attacks

- all protocols rely on some assumptions about the environment (e.g., cryptographic keys cannot be broken)

- when it got expensive, some people bought cheaper monthly tickets for a suburban station and a nearby one, and one for the destination and a nearby one
- a large investment later all barriers were automatic and tickets could record state

Changing Environment Attacks

- all protocols rely on some assumptions about the environment (e.g., cryptographic keys cannot be broken)

- But suddenly the environment changed: rail transport got privatised creating many companies cheating each other
- revenue from monthly tickets was distributed according to a formula where the ticket was bought

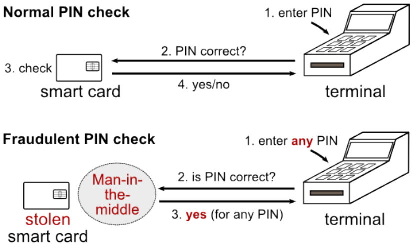
Changing Environment Attacks

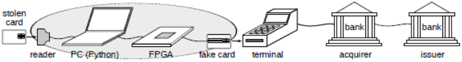
- all protocols rely on some assumptions about the environment (e.g., cryptographic keys cannot be broken)

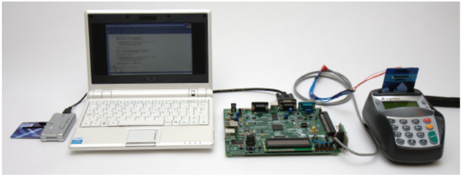
- apart from bad outsiders (passengers) you also had bad insiders (rail companies)
- chaos and litigation ensued

A Man-in-the-middle attack in real life:

- the card only says yes or no to the terminal if the PIN is correct
- trick the card in thinking transaction is verified by signature
- trick the terminal in thinking the transaction was verified by PIN







Problems with EMV

- it is a wrapper for many protocols
- specification by consensus (resulted unmanageable complexity)
- its specification is 700 pages in English plus 2000+ pages for testing, additionally some further parts are secret
- other attacks have been found
- one solution might be to require always online verification of the PIN with the bank

Good Practices

- explicit principles (you authenticate all data you might rely on)
- the one who can fix a system should also be liable for the losses

Privacy et al

Some terminology:

- secrecy is the mechanism used to limit the number of principals with access to information (eg, cryptography or access controls)
- confidentiality is the obligation to protect the secrets of other people or organizations (secrecy for the benefit of an organisation)
- anonymity is the ability to leave no evidence of an activity (eg, sharing a secret)
- privacy is the ability or right to protect your personal secrets (secrecy for the benefit of an individual)