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 sends B : . . .
B sends A : . . .
:

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

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)

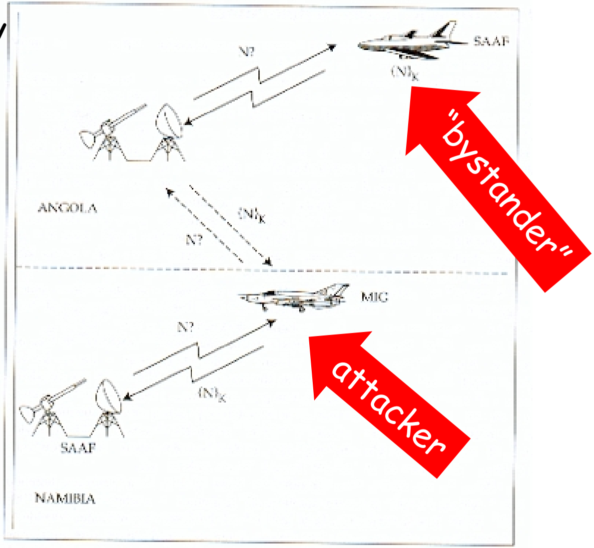


Figure 2.2 The MIG-in-the middle attack.

Identify Friend or Foe

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

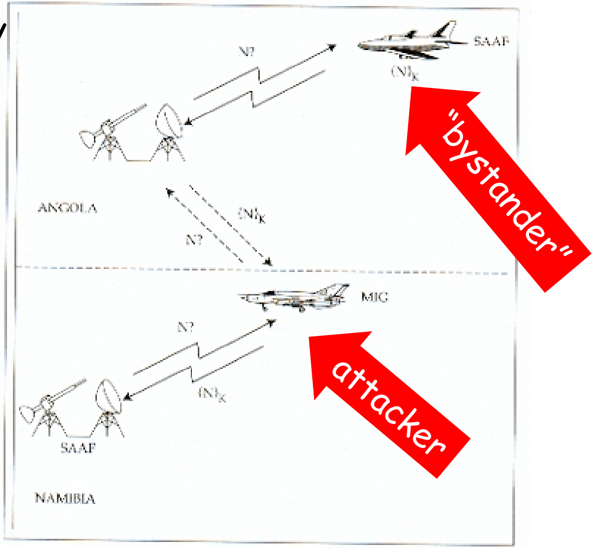


Figure 2.2 The MIG-in-the middle attack.

being outsmarted by Angola/Cuba ended SA involvement

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:

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

Replay Attacks

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

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

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

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

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

compromise K_{AB}

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

compromise K_{AB}

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

$A \to S : A, B, N_A$

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

$A \to B : \{K_{AB}, A\}_{K_{BS}}$

$B \to A : \{N_B\}_{K_{AB}}$

$A \to B : \{N_B - 1\}_{K_{AB}}$

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

$A \rightarrow S : A, B, N_A$

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

$A \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$

$B \rightarrow A : \{N_B\}_{K_{AB}}$

$A \rightarrow B : \{N_B - 1\}_{K_{AB}}$

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

$B \rightarrow I(A) : \{N'_B\}_{K_{AB}}$

$I(A) \rightarrow B : \{N'_B - 1\}_{K_{AB}}$

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

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
$B \to I(A) : \{N'_B\}_{K_{AB}}$
$I(A) \to B : \{N'_B - 1\}_{K_{AB}}$

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

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

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

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

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

“Real-World” Attacks

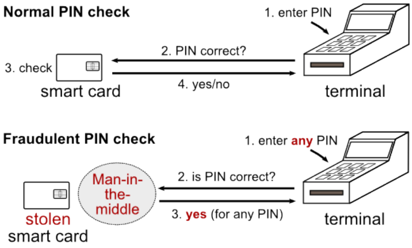
EMV (Europay, MasterCard, Visa) is a standard for payments by credit cards

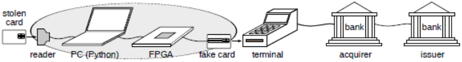
It consists of three phases:

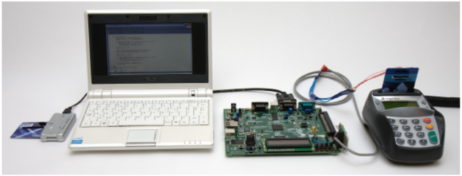
1. card authentication phase (the terminal reads the information; signs it with a public key and verifies the signed information)
2. cardholder authentication (PIN; terminal sends PIN to card which verifies it; it can also verify it online with the bank)
3. transaction authorisation (the terminal asks the card to provide an authentication code for the transaction; the code is sent to the bank for verification)

A Man-in-the-middle attack

- 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