Access Control and Privacy Policies (5)

Email: christian.urban at kcl.ac.uk
Office: S1.27 (1st floor Strand Building)

Slides: KEATS (also homework is there)

Protocols

Some examples where "over-the-air" protocols are used:

- wifi
- card readers (you cannot trust the terminals)
- RFID (passports)
- car transponders

Protocols

Some examples where "over-the-air" protocols are used:

- wifi
- card readers (you cannot trust the terminals)
- RFID (passports)
- car transponders

The point is that we cannot control the network: An attacker can install a packet sniffer, inject packets, modify packets, replay messages...fake pretty much everything.

Keyless Car Transponders

- There are two security mechanisms: one remote central locking system and one passive RFID tag (engine immobiliser).
- How can I get in? How can thieves be kept out? How to avoid MITM attacks?

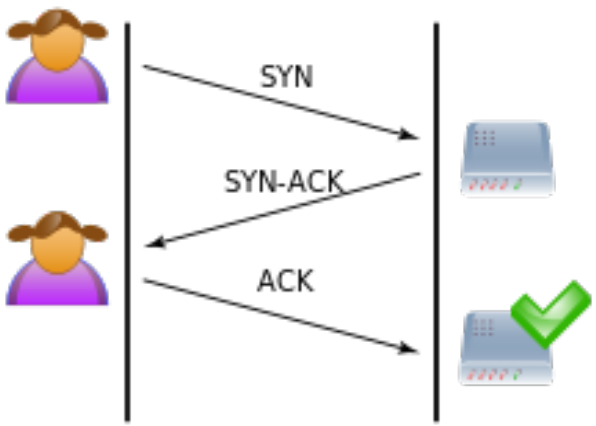
Papers: Gone in 360 Seconds: Hijacking with Hitag2, Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer

HTTPS / GSM

- I am sitting at Starbuck. How can I be sure I am really visiting Barclays? I have no control of the access point.
- How can I achieve that a secret key is established in order to encrypt my mobile conversation? I have no control over the access points.

Handshakes

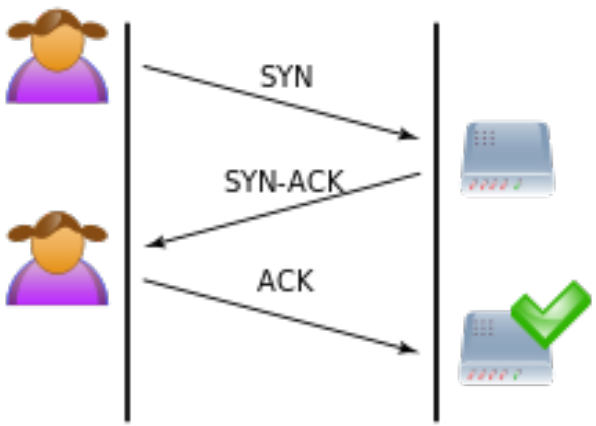
- starting a TCP connection between a client and a server initiates the following three-way handshake protocol:



Alice: Hello server!
Server: I heard you
Alice: Thanks

Handshakes

- starting a TCP connection between a client and a server initiates the following three-way handshake protocol:

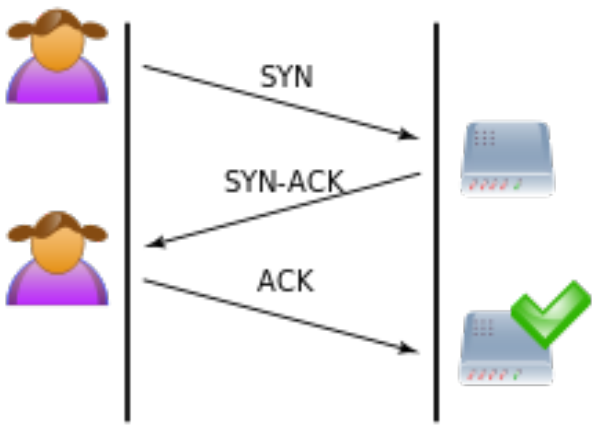


Alice: Hello server!
Server: I heard you
Alice: Thanks

$A \rightarrow S$: SYN
$S \rightarrow A$: SYN-ACK
$A \rightarrow S$: ACK

Handshakes

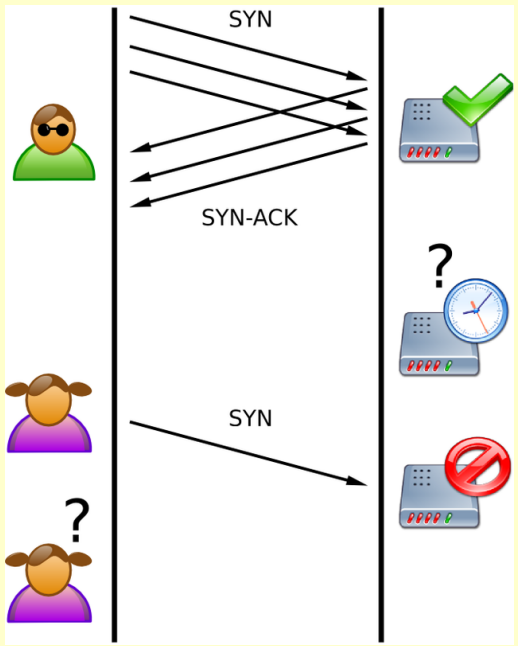
- starting a TCP connection between a client and a server initiates the following three-way handshake protocol:



Alice:
Server:
Alice:

$A \rightarrow S$: SYN
$S \rightarrow A$: SYN-ACK
$A \rightarrow S$: ACK

SYNflood attacks:



Authentication

"On the Internet, nobody knows you're a dog."

Knock Knock!
Who's there?
Alice.
Alice who?

Authentication Protocols

Alice (A) and Bob (B) share a secret key K_{AB}

Password transmission:

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

Authentication Protocols

Alice (A) and Bob (B) share a secret key K_{AB}

Password transmission:

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

Problems: Eavesdropper can capture the secret and replay it; B cannot confirm the identity of A

Authentication Protocols

Alice (A) and Bob (B) share a secret key K_{AB}

Simple Challenge Response (solving the replay problem):

$$
\begin{array}{lll}
A \rightarrow B: & \text{Hi I am A} & \\
B \rightarrow A: & N & \text{(challenge)} \\
A \rightarrow B: & \{N\}_{K_{AB}} &
\end{array}
$$

Authentication Protocols

Alice (A) and Bob (B) share a secret key K_{AB}

Simple Challenge Response (solving the replay problem):

$$\begin{array}{lll} A \rightarrow B: & \text{Hi I am A} & \\ B \rightarrow A: & N & \text{(challenge)} \\ A \rightarrow B: & \{N\}_{K_{AB}} & \end{array}$$

- cannot be replayed since next time will be another challenge N
- B authenticates A, but A does not authenticate B (Eve can intercept messages from A, send random challenge and ignore last)

Authentication Protocols

Alice (A) and Bob (B) share a secret key K_{AB}

Mutual Challenge Response:

$$\begin{aligned} A \rightarrow B: &\quad N_A \\ B \rightarrow A: &\quad \{N_A, N_B\}_{K_{AB}} \\ A \rightarrow B: &\quad N_B \end{aligned}$$

Authentication Protocols

Alice (A) and Bob (B) share a secret key K_{AB}

Mutual Challenge Response:

$$\begin{aligned} A \rightarrow B: &\quad N_A \\ B \rightarrow A: &\quad \{N_A, N_B\}_{K_{AB}} \\ A \rightarrow B: &\quad N_B \end{aligned}$$

But requires shared secret key.

Nonces

1. I generate a nonce (random number) and send it to you encrypted with a key we share
2. you increase it by one, encrypt it under a key I know and send it back to me

I can infer:

- you must have received my message
- you could only have generated your answer after I send you my initial message
- if only you and me know the key, the message must have come from you

$$
\begin{array}{ll}
A \to B: & N_A \\
B \to A: & \{N_A, N_B\}_{K_{AB}} \\
A \to B: & N_B
\end{array}
$$

The attack (let A decrypt her own messages):

$$
\begin{array}{ll}
A \to E: & N_A \\
E \to A: & N_A \\
A \to E: & \{N_A, N'_A\}_{K_{AB}} \\
E \to A: & \{N_A, N'_A\}_{K_{AB}} \\
A \to E: & N'_A \ \ (= N_B)
\end{array}
$$

$$
\begin{aligned}
A \to B&: \quad N_A \\
B \to A&: \quad \{N_A, N_B\}_{K_{AB}} \\
A \to B&: \quad N_B
\end{aligned}
$$

The attack (let A decrypt her own messages):

$$
\begin{aligned}
A \to E&: \quad N_A \\
E \to A&: \quad N_A \\
A \to E&: \quad \{N_A, N'_A\}_{K_{AB}} \\
E \to A&: \quad \{N_A, N'_A\}_{K_{AB}} \\
A \to E&: \quad N'_A \ \ (= N_B)
\end{aligned}
$$

Solutions: $K_{AB} \neq K_{BA}$ or include an id in the second message

Encryption to the Rescue?

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

Encryption to the Rescue?

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

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

Trusted Third Party

Simple protocol for establishing a secure connection via a mutually trusted 3rd party (server):

$$A \rightarrow S : A, B$$
$$S \rightarrow A : \{K_{AB}\}_{K_{AS}} \text{ and } \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}}$$
$$A \rightarrow B : \{K_{AB}\}_{K_{BS}}$$
$$A \rightarrow B : \{m\}_{K_{AB}}$$

Public-Key Infrastructure

- the idea is to have a certificate authority (CA)
- you go to the CA to identify yourself
- CA: "I, the CA, have verified that public key P^{pub}_{Bob} belongs to Bob"

- CA must be trusted by everybody
- What happens if CA issues a false certificate? Who pays in case of loss? (VeriSign explicitly limits liability to $100.)

Person-in-the-Middle

"Normal" protocol run:

- A sends public key to B
- B sends public key to A
- A sends a message encrypted with B's public key, B decrypts it with its private key
- B sends a message encrypted with A's public key, A decrypts it with its private key

Person-in-the-Middle

Attack:

- A sends public key to B — C intercepts this message and send his own public key to B
- B sends public key to A — C intercepts this message and send his own public key A
- A sends a message encrypted with C's public key, C decrypts it with its private key, re-encrypts with B's public key
- similar the other way

Person-in-the-Middle

Prevention:

- *A* sends public key to *B*
- *B* sends public key to *A*
- *A* encrypts message with *B*'s public key, send's **half** of the message to *B*
- *B* encrypts message with *A*'s public key, send's **half** of the message back to *A*
- *A* sends other half, *B* can now decrypt entire message
- *B* sends other half, *A* can now decrypt entire message

Person-in-the-Middle

Prevention:

- *A* sends public key to *B*
- *B* sends public key to *A*
- *A* encrypts message with *B*'s public key, send's **half** of the message to *B*
- *B* encrypts message with *A*'s public key, send's **half** of the message back to *A*
- *A* sends other half, *B* can now decrypt entire message
- *B* sends other half, *A* can now decrypt entire message

C would have to invent a totally new message

Car Transponder (HiTag2)

1. C generates a random number r
2. C calculates $(F, G) = \{r\}_K$
3. $C \rightarrow T$: r, F
4. T calculates $(F', G') = \{r\}_K$
5. T checks that $F = F'$
6. $T \rightarrow C$: r, G'
7. C checks that $G = G'$

Car Transponder (HiTag2)

1. C generates a random number r
2. C calculates $(F, G) = \{r\}_K$
3. $C \rightarrow T$: r, F
4. T calculates $(F', G') = \{r\}_K$
5. T checks that $F = F'$
6. $T \rightarrow C$: r, G'
7. C checks that $G = G'$

This process means that the transponder believes the car knows the key K, and the car believes the transponder knows the key K. They have authenticated themselves to each other.

Person-in-the-Middle

- Border Gateway Protocol (BGP) — routers believe their neighbours
- it is possible to advertise bad routes
- can be done over continents

http://www.renesys.com/2013/11/mitm-internet-hijacking/

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

- (social engineering attacks)

Best Practices

Principle 1: Every message should say what it means: the interpretation of a message should not depend on the context.

Best Practices

Principle 1: Every message should say what it means: the interpretation of a message should not depend on the context.

Principle 2: If the identity of a principal is essential to the meaning of a message, it is prudent to mention the principal's name explicitly in the message (though difficult).

Best Practices

Principle 3: Be clear about why encryption is being done. Encryption is not wholly cheap, and not asking precisely why it is being done can lead to redundancy. Encryption is not synonymous with security.

Possible Uses of Encryption

- Preservation of confidentiality: $\{X\}_K$ only those that have K may recover X.
- Guarantee authenticity: The partner is indeed some particular principal.
- Guarantee confidentiality and authenticity: binds two parts of a message — $\{X, Y\}_K$ is not the same as $\{X\}_K$ and $\{Y\}_K$.

Best Practices

Principle 4: The protocol designers should know which trust relations their protocol depends on, and why the dependence is necessary. The reasons for particular trust relations being acceptable should be explicit though they will be founded on judgment and policy rather than on logic.

Example Certification Authorities: CAs are trusted to certify a key only after proper steps have been taken to identify the principal that owns it.

Formal Methods

Ross Anderson about the use of Logic:

> *Formal methods can be an excellent way of finding bugs in security protocol designs as they force the designer to make everything explicit and thus confront difficult design choices that might otherwise be fudged.*

Mid-Term

- homework, handouts, programs...

Any Questions?