Analysis and design of cryptographic protocols

Main topics

- The BAN logic
- Design principles
- Case studies
  - Needham-Schroeder $\Rightarrow$ Kerberos
  - Otway-Rees
  - SSL (an old version)
  - GSM
The problem

Security protocols are three-line programs that people still manage to get wrong.

*Roger M. Needham*

The BAN logic

- After its inventors: **Burrows, Abadi, Needham**
- **Belief and action**
  - The logic cannot prove that a protocol is wrong
  - However, if you cannot prove a protocol correct, then consider that protocol with great suspicion
Formalism

\[ P \equiv X \quad \text{P believes } X. \text{ P behaves as if } X \text{ were true} \]

\[ P \bowtie X \quad \text{P sees } X: \]

\[ P \vdash X \quad \text{P once said } X: \]

\[ P \Rightarrow X \quad \text{P controls } X. \]

\[ \#(X) \quad X \text{ is fresh} \]

\[ P^K \quad K \text{ is a shared key between } P \text{ e } Q \]

\[ P \leftrightarrow Q \quad K \text{ is a shared secret between } P \text{ e } Q \]

\[ K \rightarrow P \quad K \text{ is P’s public key} \]

\[ \langle X \rangle_y \quad X \text{ is a combined with } Y \]

\[ \{X\}_K \quad X \text{ has been encrypted with } K \]
Examples

\[ A \equiv \#(N_a) \]  A believes that \( N_a \) is fresh

\[ A \equiv A \leftrightarrow B \]  A believes \( K \) to be a shared key with \( B \)

\[ T \equiv A \leftrightarrow B \]  \( T \) believes that \( K \) is a shared key between \( A \) and \( B \)

\[ A \equiv T \Rightarrow A \leftrightarrow B \]  A believes \( T \) an authority on generating session keys

\[ A \equiv T \Rightarrow \#(A \leftrightarrow B) \]  A believes that \( T \) is competent in generating fresh session keys

Preliminaries

- BAN logic considers **two epochs**: the present and the past
  - The present begins with the start of the protocol
- Beliefs achieved in the present are **stable** for all the protocol duration
- If \( P \) says \( X \) then \( P \) believes \( X \)
- Beliefs of the past may not hold in the present
### Postulates: message meaning rule

<table>
<thead>
<tr>
<th>Postulate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P \equiv Q \leftrightarrow P, P \triangleleft {X}_K$</td>
<td>If $K$ is a shared key between $P$ and $Q$, a $P$ sees a message encrypted by $K$ containing $X$ (and $P$ did not send that message), then $P$ believes that $X$ was sent by $Q$.</td>
</tr>
<tr>
<td>$P \equiv Q \sim X$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Postulate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P \equiv \rightarrow Q, P \triangleleft {X}^{-1}$</td>
<td>If $K$ is $Q$’s public key, and $P$ sees a message signed by $con K^{-1}$ containing $X$, then $P$ believes that $X$ was sent by $Q$.</td>
</tr>
<tr>
<td>$P \equiv Q \sim X$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Postulate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P \equiv Q \equiv X$</td>
<td>If $Y$ is a shared secret between $P$ and $Q$, and $P$ sees a message where $Y$ is combined with $X$ (and $P$ did not send the message), then $P$ believes that $X$ was sent by $Q$.</td>
</tr>
<tr>
<td>$P \equiv Q \sim X$</td>
<td></td>
</tr>
</tbody>
</table>

### Postulates: nonce verification rule

<table>
<thead>
<tr>
<th>Postulate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P \equiv #(X), P \equiv Q \sim X$</td>
<td>$P \equiv Q \equiv X$</td>
</tr>
</tbody>
</table>

- If $P$ believes $Q$ said $X$ and $P$ believes $X$ is fresh, then $P$ believes $Q$ believes $X$ (now, in this protocol execution).
- If $P$ believes $X$ was sent by $Q$, and $P$ believes $X$ is fresh, then $P$ believes $Q$ has sent $X$ in this protocol execution instance.
Postulates: jurisdiction rule

\[ P \models Q \equiv X, P \models Q \Rightarrow X \]
\[ P \models X \]

- If \( P \) believes \( Q \) believes \( X \) and \( P \) believes \( Q \) is an authority on \( X \), then \( P \) believes \( X \) too
- If \( P \) believes \( Q \) says \( X \) and \( P \) trusts \( Q \) on \( X \), then \( P \) believes \( X \) too

Altri postulati

\[ P \models X, P \models Y \]
\[ P \models (X, Y) \]
\[ P \models Q \models (X, Y) \]
\[ P \models Q \models X \]
\[ P \models Q \models \neg X \]
\[ P \models \neg (X, Y) \]
\[ P \models \neg X \]
\[ P \models \neg \neg X \]
\[ P \models \neg \neg (X, Y) \]

\[ P \models #(X) \]
\[ P \models #(X, Y) \]

\[ P \models (X, Y) \]
\[ P \models X \]
\[ P \models (X) \]
\[ P \models X \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
\[ P \models k \]
Protocollo idealizzato

Each protocol step is represented as

\[ A \rightarrow B : \text{messaggio} \]

For example:

\[ A \rightarrow B : \{ A, K_{ab} \}_{K_{ba}} \]

This notations is ambiguous. Thus the protocol is idealized

\[ A \rightarrow B : \{ K_{ba} \}_{K_{ba}} \]

The resulting specification is more clear and you can resume the formula

\[ K_{ba} \]

\[ B \triangleleft A \leftrightarrow B \]

Protocol analysis

- Protocol analysis consists in the following steps
  1. Derive the idealized protocol from the real one
  2. Determine assumptions
  3. Apply postulates to each protocol step and determine beliefs achieved by principals at the step
  4. Draw conclusions
Protocol analysis

[assumption] \( S_1 \)  [assertion 1]
....
[assertion \(i-1\)] \( S_i \)  [assertion \(i\)]
....
[assertion \(n-1\)] \( S_n \)  [conclusions]

\[
\text{Assertion } i-1 \quad A \equiv A \leftrightarrow B \\
\text{Step } i \quad A \rightarrow B : \{X\}_K \\
\text{Assertion } i \quad A \equiv A \leftrightarrow B, B \equiv A \sim X
\]

By applying the message meaning postulate

Objectives of a protocol

Objectives depend on the context

- Typical objectives

  \[
  A \models A \leftrightarrow B \\
  B \models A \leftrightarrow B
  \]

  (key authentication)

  often

  \[
  A \models B \models A \leftrightarrow B \\
  B \models A \models A \leftrightarrow B
  \]

  (key confirmation)

  also

  \[
  A \models # A \leftrightarrow B \\
  B \models # A \leftrightarrow B
  \]

  (key freshness)

- Interaction with a certification authority

  \[
  A \models a \quad B
  \]
### Needham-Schroeder (1978)

**Real protocol**

<table>
<thead>
<tr>
<th>Step</th>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>$A \rightarrow T$</td>
<td>$A, B, N_a$</td>
</tr>
<tr>
<td>M2</td>
<td>$T \rightarrow A$</td>
<td>$E_{K_a}(N_a, B, K_{ab}, E_{K_b}(K_{ab}, A))$</td>
</tr>
<tr>
<td>M3</td>
<td>$A \rightarrow B$</td>
<td>$E_{K_b}(K_{ab}, A)$</td>
</tr>
<tr>
<td>M4</td>
<td>$B \rightarrow A$</td>
<td>$E_{K_{ab}}(N_b)$</td>
</tr>
<tr>
<td>M5</td>
<td>$A \rightarrow B$</td>
<td>$E_{K_{ab}}(N_b - 1)$</td>
</tr>
</tbody>
</table>

---

**Idealized protocol**

<table>
<thead>
<tr>
<th>Step</th>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2</td>
<td>$T \rightarrow A$</td>
<td>${N_a, A \leftrightarrow B} #({A \leftrightarrow B}, {A \leftrightarrow B}<em>{K_b})</em>{K_a}$</td>
</tr>
<tr>
<td>M3</td>
<td>$A \rightarrow B$</td>
<td>${A \leftrightarrow B}_{K_b}$</td>
</tr>
<tr>
<td>M4</td>
<td>$B \rightarrow A$</td>
<td>${N_b, A \leftrightarrow B}<em>{K</em>{ab}}$ from $B$</td>
</tr>
<tr>
<td>M5</td>
<td>$A \rightarrow B$</td>
<td>${N_b, A \leftrightarrow B}<em>{K</em>{ab}}$ from $A$</td>
</tr>
</tbody>
</table>

---

*Implicit statement, not explicitly derived from the real protocol*

- The idealized protocol may contain implicit statements.
Needham-Schroeder

Principle 1. We have to specify the meaning of each message; specification must depend on the message contents; it must be possible to write a sentence describing such a meaning.

Assumptions

\[
\begin{align*}
A &\models A \leftrightarrow T \\
B &\models B \leftrightarrow T \\
T &\models A \leftrightarrow T \\
T &\models B \leftrightarrow T \\
T &\models A \leftrightarrow B \\
A &\models \left( T \Rightarrow A \leftrightarrow B \right) \\
B &\models \left( T \Rightarrow A \leftrightarrow B \right) \\
A &\models \left( T \Rightarrow \# \left( A \leftrightarrow B \right) \right) \\
B &\models \# \left( N_a \right) \\
T &\models \# \left( A \leftrightarrow B \right) \\
B &\models \# \left( N_b \right) \\
T &\models \# \left( A \leftrightarrow B \right) \\
B &\models \# \left( A \leftrightarrow B \right)
\end{align*}
\]

Objectives

\[
\begin{align*}
A &\models A \leftrightarrow B \\
B &\models B \leftrightarrow B \\
A &\models B \equiv A \leftrightarrow B \\
A &\models B \equiv A \leftrightarrow B \\
B &\models A \equiv A \leftrightarrow B \\
B &\models A \equiv A \leftrightarrow B
\end{align*}
\]

 Principle 2. Designer must know the trust relationships upon which the protocol is based. He/she must know why they are necessary. Such reasons must be made explicit.
Needham-Schroeder

After M2
message meaning e
nonce verification
\[ A \models T \models a \leftarrow b \]
\[ A \models T \models \# \leftarrow b \]
jurisdiction rule
\[ A \models a \leftarrow b \]
\[ A \models \# \leftarrow b \]
Principle 3. A key may have been used recently to encrypt a nonce but it may be old or compromised. The recent use of a key does not make it more secure.

After M3
message meaning
\[ B \models T \models a \leftrightarrow b \]
nonce verification
\[ B \models T \models a \leftrightarrow b \]
jurisdiction rule
\[ B \models a \leftarrow b \]
Dopo M5
message meaning
\[ B \models A \models a \leftarrow b \]
nonce verification
\[ B \models A \models a \leftarrow b \]

Otway-Rees protocol

Real protocol
M1. \( A \rightarrow B \): \( M, A, B, E_{K_A}(N_A, M, A, B) \)
M2. \( B \rightarrow T \): \( M, A, B, E_{K_A}(N_A, M, A, B), E_{K_B}(N_B, M, A, B) \)
M3. \( T \rightarrow B \): \( M, E_{K_A}(N_A, K_{ab}), E_{K_B}(N_B, K_{ab}) \)
M4. \( B \rightarrow A \): \( M, E_{K_A}(N_A, K_{ab}) \)

Idealized protocol
M1. \( A \rightarrow B \): \( \{ N_A, M, A, B \}_{K_A} \)
M2. \( B \rightarrow T \): \( \{ N_A, M, A, B \}_{K_A}, \{ N_B, M, A, B \}_{K_B} \)
M3. \( T \rightarrow B \): \( \{ N_a, A \leftrightarrow B, B \models A \models M \}_{K_A}, \{ N_b, A \leftrightarrow B, A \models M \}_{K_B} \)
M4. \( B \rightarrow A \): \( \{ N_b, A \leftrightarrow B, A \models M \}_{K_A} \)
The protocol presents two strange features

- $N_a$ and $N_b$ are nonces. They are supposed to prove freshness. Then, why are they encrypted in messages M1 and M2?

- Why do we need $M$ in addition to $N_a$ and $N_b$?
  - Actually it disappears after M2

---

**Otway-Rees**

M1. $A \rightarrow B$:  \( \left\{ N_a, M, A, B \right\}_{K_a} \)

M2. $B \rightarrow T$:  \( \left\{ N_a, M, A, B \right\}_{K_a}, \left\{ N_b, M, A, B \right\}_{K_b} \)

M3. $T \rightarrow B$:  \( \left\{ N_a, A \leftrightarrow B, B \sim M \right\}_{K_a}, \left\{ N_b, A \leftrightarrow B, A \sim M \right\}_{K_b} \)

M4. $B \rightarrow A$:  \( \left\{ N_a, A \leftrightarrow B, B \sim M \right\}_{K_a} \)

**M1:** Alice says that $M$ is a transaction with Bob and $N_a$ is another name of Alice in $M$

**M2:** Bob says that $M$ is a transaction with Bob and $N_b$ is another name of Bob in $M$

**M3:** After receiving $N_b$, $T$ says that $K_{ab}$ is good and that Alice believed to be in $M$

**M4:** After receiving $N_a$, $T$ says that $K_{ab}$ is good and that Bob believed to be in $M$
Protocollo di Otway-Rees

Ipotesi

Risultati

After M2

Given Bob’s belief in $N_b$ freshness, then

Given Bob’s trust in T about keys and its capability to relay, then

After M3

After M4

Given Alice’s belief in $N_a$, then

Given Alice’s trust in T about keys and its capability to relay and given Alice’s belief in M freshness

Ban logic
Otway-Rees

- Nonces $N_a$ and $N_b$ are for freshness but also to link messages M1 and M2 to messages M3 and M4, respectively
  - Nonce $N_a (N_b)$ is a reference to Alice (Bob) within $M$, or equivalently,
  - nonce $N_a (N_b)$ is another name for Alice (Bob) in $M$
- In M1 (M2), encryption is not for secrecy but to indissolubly link Alice (Bob), $N_a (N_b)$ and $M$ together

**Principle 4.** Properties required to nonces must be clear. What it is fine to guarantee freshness might not be to guarantee an association between parts

**Principles 5.** The reason why encryption is used must be clear

---

Otway-Rees modified

- If nonces have to guarantee freshness only, then messages M1 and M2 could be modified as follows

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M1.</td>
<td>$A \rightarrow B$: $M, A, B, N_A, E_{K_A}(M, A, B)$</td>
</tr>
<tr>
<td>M2.</td>
<td>$B \rightarrow T$: $M, A, B, N_A, E_{K_A}(M, A, B), N_B, E_{K_B}(M, A, B)$</td>
</tr>
</tbody>
</table>

  - M1 and M3 (M2 and M4) are not linked anymore
  - The resulting protocol is subject to a man-in-the-middle attack
    - An adversary may impersonate Bob (Alice) with respect to Alice (Bob)
Otway-Rees modified

- The resulting protocol is subject to a man-in-the-middle attack
  - An adversary may impersonate Bob (Alice) with respect to Alice (Bob)
- Let us suppose that Carol (the adversary)
  - has already carried out a protocol instance with Alice
  - holds an "old" ciphertext $E_{K_a}(M', A, C)$

The Attack

M1. $A \rightarrow B[C]$: $M', A, B, N_a, E_{K_a}(M, A, B)$
M2. $C \rightarrow T$: $M', A, C, N_a, E_{K_a}(M', A, C), N_e, E_{K_e}(M', A, C)$
M3. $T \rightarrow C$: $M', E_{K_a}(N_a, K_{ab}), E_{K_e}(N_e, K_{ab})$
M4. $[C]B \rightarrow A$: $E_{K_a}(N_a, K_{ab})$
Protocollo di Otway-Rees "migliorato"

- If we need to insert references to Alice and Bob in M3 and M4, then the protocol can be modified as follows:

M1. \( A \rightarrow B : A, B, N_a \)
M2. \( B \rightarrow T : A, B, N_a, N_b \)
M3. \( T \rightarrow B : E_{K_A}(N_a, A, B, K_{ab}), E_{K_B}(N_b, A, B, K_{ab}) \)
M4. \( B \rightarrow A : E_{K_A}(N_a, A, B, K_{ab}) \)

**Principle 6.** If an identifier is necessary to complete the meaning of a message, it is prudent to explicitly mention such an identifier in the message.

---

**SSL (old version)**

Protocol objectives:
- establish a shared key \( K_{ab} \)
- mutual authentication

| M1. | \( A \rightarrow B : \{K_{ab}\}_{K_b} \) |
| M2. | \( B \rightarrow A : \{N_b\}_{K_{ab}} \) |
| M3. | \( A \rightarrow B : \{C_A, \{N_b\}_{K^{-1}_a}\}_{K_{ab}} \) |

M1: Bob sees key \( K_{ab} \)
M2: After receiving it, Bob says he saw \( K_{ab} \)
M3: After receiving it, Alice says she saw \( N_b \)

In the protocol there is no link between A and key \( K_{ab} \)
Adversary Mallet plays an MIM attack and impersonates \( A \) with respect to \( B \)

\[
\begin{align*}
M1': \{K_{am}\}_{K_m} & \quad \rightarrow \\
M2': \{N_b\}_{K_{am}} & \quad \rightarrow \\
M3': \{C_A, \{N_b\}_{K^{-1}_a}\}_{K_{am}} & \quad \rightarrow \\
\end{align*}
\]

After M3, Bob believes he is talking to Alice

SSL (old version)

The attack may be avoided by modifying M3 as follows

\[
M3 \quad A \rightarrow B : \{C_A, \{A, B, K_{ab}, N_b\}_{K^{-1}_a}\}_{K_{ab}}
\]

after receiving \( N_b \), Alice says that \( K_{ab} \) is a good key to communicate with Bob

- Important
  - It’s necessary to introduce identifiers A and B in message M3 because, otherwise, the attack would be still possible by setting \( K_{am} = K_{bm} \)
Sign encrypted data

**Principle 7.**
- If an entity signs an encrypted message, it is not possible to infer that such an entity knows the message contents.
- In contrast, if an entity signs a message and then encrypts it, then it is possible to infer that the entity knows the message contents.

**Esempio: X.509**

\[ A \rightarrow B : \{ T_a , N_a , B , X_a , \{ Y_a \}_{K_{Y_a}} \}_{K_{X_a}} \]

The message contains no proof that the sender (Alice) knows \( Y_a \).

---

On hash functions

For efficiency, we sign the hash of a message rather than the message itself.

\[ A \rightarrow B : \{ X \}_{K_{Y_a}} , \{ h(X) \}_{K_{X_a}} \]

- The message does not contain any proof that the signer Alice actually knows \( X \).
- However, the signer Alice expects that the receiver Bob behaves as if the sender Bob knew the message.
- Therefore, unless the signer Alice is unwary*, signing the hash is equivalent to sign the message.

* Metaphore: a manager who signs without reading.
Postulates for hash functions

\[
P \equiv Q \sim h(X), \quad P \preceq X
\]

The postulate can be generalized to composite messages

\[
P \equiv Q \sim h(X_1, \ldots, X_n), \quad P \preceq X_1, \ldots, P \preceq X_n
\]

\[
P \equiv Q \sim (X_1, \ldots, X_n)
\]

Notice that \( P \) may receive \( X_i \) from different channels in different moments.

The GSM case

Real protocol

M1. \( C \rightarrow S : C \)
M2. \( C \leftarrow S : \rho \)
M3. \( C \rightarrow S : \sigma \)

- \( \rho \) random challenge generated by \( S \)
- \( <\sigma, K> = h(K_C, \rho) \)

Assumptions

\[
S \equiv C \leftrightarrow S \quad C \equiv S \leftrightarrow C
\]

\[
S \equiv \#(\rho)
\]

Idealized protocol

M3. \( C \rightarrow S : \left\langle C \leftrightarrow S, \rho \right\rangle_{K_C} \)

Results

\[
S \equiv C \equiv S \leftrightarrow C
\]
Predictable nonces

**Principle 8.** A predictable quantity can be used as a nonce in a challenge-response protocol. In such a case, the nonce must be protected by a replay attack.

**Example: Alice receives a time stamp from a Time Server**
(ex. Alice uses the time stamp to synchronize her clock)

\[
\begin{align*}
M1 & \quad A \rightarrow S \quad A, N_a \\
M2 & \quad S \rightarrow A \quad \{T_s, N_a\}_{K_M}
\end{align*}
\]

- \(N_a\): predictable nonce
- (M2): After receiving \(N_a\), S said \(T_s\)

**Ipotesi**

\[
A \equiv S \leftrightarrow A \\
A \equiv S \Rightarrow T_s \\
A \equiv \#(N_a)
\]

**Risultati**

\[
A \equiv S \sim T_s \\
A \equiv S \Rightarrow T_s \\
A \equiv T_s
\]

An attack

\(M\) predicts the next value of \(N_a\)

\[
\begin{align*}
M1 & \quad M \rightarrow S \quad A, N_a \\
M2 & \quad S \rightarrow M \quad \{T_s, N_a\}_{K_M} \quad (S \text{ receives } M2 \text{ at time } T_s)
\end{align*}
\]

At time \(T_s' > T_s\), Alice initiates a protocol instance

\[
\begin{align*}
M1 & \quad A \rightarrow S[M] \quad A, N_a \\
M2 & \quad S[M] \rightarrow A \quad \{T_s, N_a\}_{K_M}
\end{align*}
\]

Alice is led to believe that the current time is \(T_s\) and not \(T_s'\)

Since \(N_a\) is predictable then it must be protected

\[
\begin{align*}
M1 & \quad A \rightarrow S \quad A, \{N_a\}_{K_M} \\
M2 & \quad S \rightarrow A \quad \{T_s, \{N_a\}_{K_M}\}_{K_M}
\end{align*}
\]
Nonce: timestamp

Principle 9. If freshness is guaranteed by time stamp, then the difference between the local clock and that of other machines must be largely smaller than the message validity. Furthermore, the clock synchronization mechanisms is part of the Trusted Computing Base (TCB)

Example

- Kerberos. If the server clock can be set back, then authenticators can be reused
- Kerberos. If the server clock can be set ahead, then it is possible to generate post-dated authenticators

On coding messages

Principle 10. The contents of a message must allow us to determine: (i) the protocol the message belongs to, (ii) the execution instance of the protocol, (iii) the number of the message within the protocol

Example Needham-Schroeder

\[
\begin{align*}
M_4 & \quad B \rightarrow A \quad E_{K_{ab}} (N_b) \\
M_5 & \quad A \rightarrow B \quad E_{K_{ab}} (N_b - 1)
\end{align*}
\]

It would be more clear

\[
\begin{align*}
M_4 & \quad B \rightarrow A \quad E_{K_{ab}} (\text{N-S Message 4, } N_b) \\
M_5 & \quad A \rightarrow B \quad E_{K_{ab}} (\text{N-S Message 5, } N_b)
\end{align*}
\]