Last Time (I)

Lecture

- Dolev Yao Model
- Terms and Messages
- Notion of Locality
- Undecidability Result
Last Time (I)

Exercise

- Properties of Syntactic Subterms
- Locality Result
- Security against Passive Intruder
- Logical Passive Attack on Shamir 3-Pass Protocol
Outline of Today

1. Active Intruder: Security Problem
2. Bounded Number of Sessions
3. NP-Hardness for Bounded Number of Sessions
4. Unbounded number of sessions
5. Conclusion
Outline

1 Active Intruder: Security Problem

2 Bounded Number of Sessions

3 NP-Hardness for Bounded Number of Sessions

4 Unbounded number of sessions

5 Conclusion
The Intruder is the Network (Worst Case)

Passive: Intruder deduction problem
The Intruder is the Network (Worst Case)

Passive: Intruder deduction problem

Active Intruder Security problem

- intercept messages (add messages to his knowledge)
- Play messages from his knowledge
- Start new sessions

Execution tree has:
- infinite branching (size of messages is not bounded)
- infinite depth (number of sessions is not bounded)
Active Intruder with bounded number of sessions

- Theoretically: *decidable*
- Interesting *practically*:
  - Find flaws
  - Usually attacks use *few sessions*!
### Dolev-Yao Deduction System

**Deduction System**: $T_0 \vdash ? \ s$

- **(A)** \[
\frac{u \in T_0}{T_0 \vdash u}
\]
- **(P)** \[
\frac{T_0 \vdash u \quad T_0 \vdash v}{T_0 \vdash \langle u, v \rangle}
\]
- **(C)** \[
\frac{T_0 \vdash u \quad T_0 \vdash v}{T_0 \vdash \{u\}_v}
\]
- **(UL)** \[
\frac{T_0 \vdash \langle u, v \rangle}{T_0 \vdash u}
\]
- **(UR)** \[
\frac{T_0 \vdash \langle u, v \rangle}{T_0 \vdash v}
\]
- **(D)** \[
\frac{T_0 \vdash \{u\}_v \quad T_0 \vdash v}{T_0 \vdash u}
\]
Model: actions, roles and protocol

**Definition (Action)**

An action is a couple \( (recv(u), send(v)) \) such that \( u \in T(F, X) \cup \{init\} \), \( v \in T(F, X) \cup \{stop\} \). Denoted \( (u \rightarrow v) \).
Model: actions, roles and protocol

**Definition (Action)**

An action is a couple \((\text{recv}(u), \text{send}(v))\) such that 
\(u \in \mathcal{T}(F, X) \cup \{\text{init}\}, \ v \in \mathcal{T}(F, X) \cup \{\text{stop}\}\). Denoted \((u \rightarrow v)\).

**Definition (Role)**

A role is a finite sequence of actions:

\[(u_1 \rightarrow v_1), \ldots, (u_n \rightarrow v_n)\]

such that \(\text{vars}(v_i) \subseteq \bigcup_{1 \leq j \leq i} \text{vars}(u_j)\).
Model: actions, roles and protocol

Definition (Action)

An action is a couple \((recv(u), send(v))\) such that \(u \in T(\mathcal{F}, \mathcal{X}) \cup \{\text{init}\}\) and \(v \in T(\mathcal{F}, \mathcal{X}) \cup \{\text{stop}\}\). Denoted \((u \rightarrow v)\).

Definition (Role)

A role is a finite sequence of actions:

\[(u_1 \rightarrow v_1), \ldots, (u_n \rightarrow v_n)\]

such that \(\text{vars}(v_i) \subseteq \bigcup_{1 \leq j \leq i} \text{vars}(u_j)\).

Definition (Protocol)

A protocol \(P\) is a finite set of roles: \(P = \{R_1, \ldots, R_k\}\)
1st Example

**Example (Needham-schroeder)**

1. $A \rightarrow B : \{N_a, A\}_{pk(B)}$
2. $B \rightarrow A : \{N_a, N_b\}_{pk(A)}$
3. $A \rightarrow B : \{N_b\}_{pk(B)}$

Write down each agent’s role description.
1st Example

Example (Needham-schroeder)

1. $A \rightarrow B : \{N_a, A\}_{pk(B)}$
2. $B \rightarrow A : \{N_a, N_b\}_{pk(A)}$
3. $A \rightarrow B : \{N_b\}_{pk(B)}$

Write down each agent’s role description.

$R_A = (init \rightarrow \{N_a, A\}_{pk(B)}),$
$\quad (\{N_a, y_b\}_{pk(A)} \rightarrow \{y_b\}_{pk(B)}),$
$R_B = (\{x_a, z\}_{pk(B)} \rightarrow \{x_a, N_b\}_{pk(z)})$
$\quad (\{N_b\}_{pk(B)} \rightarrow stop)$
Scyther Notation

A:  const Na: Nonce;
    var Nb: Nonce;

    send(A,B, {Na,A}pk(B));
    recv(B,A, {Na,Nb}pk(A));
    send(A,B, {Nb}pk(B));

B:  const Nb: Nonce;
    var Na: Nonce;

    recv(A,B,{Na,A}pk(B));
    send(B,A,{Na,Nb}pk(A));
    recv(A,B,{Nb}pk(B));
Exercise

Denning-Sacco Protocol

1. \( A \rightarrow S : \langle A, B \rangle \)
2. \( S \rightarrow A : \{\langle \langle B, N_{AB} \rangle, \langle N_s, \langle \langle N_{AB}, \langle A, N_s \rangle \rangle \rangle \rangle_{K_{BS}} \rangle \rangle \} \rangle_{K_{AS}} \)
3. \( A \rightarrow B : \{\langle N_{AB}, \langle A, N_s \rangle \rangle \} \rangle_{K_{BS}} \)
4. \( B \rightarrow A : \{S_{AB} \} \rangle_{N_{AB}} \)

\( P_{DS} = \{R_A, R_B, R_S\} \) models one session of \( A, B \) and \( S \).
Exercise

Denning-Sacco Protocol

1. \( A \rightarrow S : \langle A, B \rangle \)
2. \( S \rightarrow A : \{ \{ \langle B, N_{AB} \rangle, \langle N_s, \{ \langle N_{AB}, \langle A, N_s \rangle \rangle \} K_{BS} \} \} \} K_{AS} \)
3. \( A \rightarrow B : \{ \langle N_{AB}, \langle A, N_s \rangle \rangle \} K_{BS} \)
4. \( B \rightarrow A : \{ S_{AB} \} N_{AB} \)

\( P_{DS} = \{ R_A, R_B, R_S \} \) models one session of \( A, B \) and \( S \).

\[ R_A = \text{ (init } \rightarrow \langle A, B \rangle , \) \]
\[ (\{ \{ \langle B, x_A \rangle, \langle y_A, z_A \rangle \} \} K_{AS} \rightarrow z_A ) , \]
\[ (\{ w_A \} x_A \rightarrow \text{ stop} ) \]

\[ R_B = \text{ (} \{ x_B, \langle a, y_B \rangle \} \} K_{BS} \rightarrow \{ S_{AB} \} x_B \) \]

\[ R_S = \text{ (} \langle A, B \rangle \rightarrow \{ \langle B, N_{AB}, \langle N_s, \langle A, N_s \rangle \rangle \} K_{BS} \} \) \} K_{AS} ) \]
Semantic

Definition (States and Transitions)

A state is a couple \((T, P)\) where \(T\) is a set of ground terms (intruder knowledge) and \(P\) a protocol. We define a transition relation between states \((T, P) \rightarrow (T', P')\) by:

- \(R_i \in P, R_i = (u \rightarrow v), R'_i\)
- \(T \vdash u\sigma \ (\text{dom}(\sigma) = \text{vars}(u))\)
- \(T' = T \cup \{v\sigma\}\)
- \(P' = (P \setminus \{R_i\}) \cup R'_i\sigma\)
Example

Let \( T = \{ a, b, k_I \} \) and \( P = \{ R \} \) where
\[
R = (\langle x, y \rangle \rightarrow \langle \{ y \}_k, x \rangle), (z \rightarrow \langle x, \langle y, z \rangle \rangle).
\]

- \((T, P) \rightarrow (T \cup \{ \langle \{ b \}_k, a \rangle \}, \{(z \rightarrow \langle a, \langle b, z \rangle \rangle)\})\)
- \((T, P) \rightarrow (T \cup \{ \langle \{ a \}_{k_I}, a \rangle \}, \{(z \rightarrow \langle a, \langle \{ a \}_{k_I}, z \rangle \rangle)\})\)
- \((T, P) \not\rightarrow (T \cup \{ \langle \{ a \}_k, a \rangle \}, \{(z \rightarrow \langle a, \langle \{ a \}_k, z \rangle \rangle)\})\)
Example

Let $T = \{a, b, k_I\}$ and $P = \{R\}$ where
$R = (\langle x, y \rangle \rightarrow \langle \{y\}_k, x \rangle), (z \rightarrow \langle x, \langle y, z \rangle \rangle)$.

- $(T, P) \rightarrow (T \cup \{\langle \{b\}_k, a \rangle\}, \{(z \rightarrow \langle a, \langle b, z \rangle \rangle)\})$
- $(T, P) \rightarrow (T \cup \{\langle \{a\}_k^l, a \rangle\}, \{(z \rightarrow \langle a, \langle \{a\}_k^l, z \rangle \rangle)\})$
- $(T, P) \not\rightarrow (T \cup \{\langle \{a\}_k, a \rangle\}, \{(z \rightarrow \langle a, \langle \{a\}_k, z \rangle \rangle)\})$

Each branch has a finite depth, but possibly an infinite branching.
### Preservation of the secrecy

#### Definition (Secrecy)

Let $T_1$ be a ground set of terms (Initial knowledge of the intruder). A protocol $P$ does not preserve the secrecy of a ground term $s$ for $T_1$ if there exist a state $(T', P')$, such that

- $T' \vdash s$
- $(T_1, P) \rightarrow^* (T', P')$

where $\rightarrow^*$ is the reflexive and transitive closure of $\rightarrow$.

If there does not exist a such state $(T', P')$ we say that $P$ preserves the secrecy of $s$ for the initial intruder knowledge $T_1$. 
Interleaving

Definition (Partial Order $<_P$)

A protocol $P$ define a partial order $<_P$ on actions of $P$, s.t

$$(u_i \rightarrow v_i) <_P (u_j \rightarrow v_j)$$

if $R \in P$, $R = (u_1 \rightarrow v_1) \ldots (u_i \rightarrow v_i) \ldots (u_j \rightarrow v_j) \ldots (u_n \rightarrow v_n)$ $(1 \leq i \leq j \leq n)$. 
Interleaving

Definition (Partial Order $<_P$)

A protocol $P$ define a partial order $<_P$ on actions of $P$, s.t

$$(u_i \rightarrow v_i) <_P (u_j \rightarrow v_j)$$

if $R \in P$, $R = (u_1 \rightarrow v_1) \ldots (u_i \rightarrow v_i) \ldots (u_j \rightarrow v_j) \ldots (u_n \rightarrow v_n)$ ($1 \leq i \leq j \leq n$).

Definition (Execution Order $<_E$)

An execution order $<_E$ of $P$ is a total order on the subset $A$ of actions of $P$, compatible with $<_P$ and stable by predecessor, i.e.

$$\text{if } b \in A \text{ et } a <_P b \text{ then } a \in A \text{ and } a <_E b$$

It corresponds to an interleaving of roles.
Secrecy

**Definition (Secrecy over \( \prec_E \))**

Let an execution order \( \prec_E \) of \( P \). We assume that

\[
(u_1 \rightarrow v_1) \prec_E \ldots \prec_E (u_n \rightarrow v_n)
\]

\( \prec_E \) does not preserve the secrecy of \( s \), given \( T_1 \) if there exists \( \sigma_1, \ldots, \sigma_n \) such that

\[
(P, T_1) \rightarrow (P_1, T_1 \cup \{v_1 \sigma_1\}) \rightarrow \ldots \rightarrow (P_n, T_1 \cup \{v_1 \sigma_1, \ldots, v_n \sigma_n\})
\]

and \( T_1 \cup \{v_1 \sigma_1, \ldots, v_n \sigma_n\} \vdash s \).
Outline

1. Active Intruder: Security Problem
2. Bounded Number of Sessions
3. NP-Hardness for Bounded Number of Sessions
4. Unbounded number of sessions
5. Conclusion
Constraint System

Symbolic representation of execution tree by constraint system.

**Definition (Constraint System)**

A constraint is an expression $T \models u$ where $T$ is a set of terms and $u$ a term.

A constraint system $C$ is a finite set of constraints $\bigcup_{1 \leq i \leq n} T_i \models u_i$ such that

- $T_i \subseteq T_{i+1}$ $(1 \leq i \leq n)$
- if $T_i \models u_i \in C$ and $x \in \text{vars}(T_i)$ then $T_j = \min\{T' \mid T' \models v \in C, x \in \text{vars}(v)\}$ exists and $j < i$

A substitution $\sigma$ is a solution of $C$ if $T\sigma \models u\sigma$ for all $T \models u \in C$.

We denote by $\perp$ a constraint system unsatisfiable.
From Protocols to Constraint system

Let $P$ a protocol, $<_E$ an execution order of $P$ and $s$ a secret term.

$$(u_1 \rightarrow v_1) <_E (u_2 \rightarrow v_2) <_E \ldots <_E (u_n \rightarrow v_n)$$

We associate $C$:

$$
\begin{align*}
T_1 &\models u_1 \\
T_2 &= T_1 \cup \{v_1\} \models u_2 \\
&\vdots \\
T_n &= T_{n-1} \cup \{v_{n-1}\} \models u_n \\
T_{n+1} &= T_n \cup \{v_n\} \models s
\end{align*}
$$

We show that $C$ has a solution iff $<_E$ does not preserve the secret of the term $s$. 

\[20 / 70\]
Exercises

Exercise 1

\begin{align*}
A \rightarrow B &: \langle A, N_A \rangle \\
B \rightarrow A &: \{ \langle N_A, N_B \rangle \}_{K_{ab}} \\
A \rightarrow B &: N_B \\
B \rightarrow A &: \{ \langle K, N_B \rangle \}_{K_{ab}} \\
A \rightarrow B &: \{ s \}_K
\end{align*}

Intruder knows only identities of $A$ and $B$.

- Give role specification of this protocol of an instance of execution between $A$ and $B$.
- Give a constraint system associated to this protocol between $A$ and $B$. 
Solution

\[ A \rightarrow B : \langle A, N_A \rangle \]
\[ B \rightarrow A : \{\langle N_A, N_B \rangle\}_{K_{ab}} \]
\[ A \rightarrow B : N_B \]
\[ B \rightarrow A : \{\langle K, N_B \rangle\}_{K_{ab}} \]
\[ A \rightarrow B : \{s\}_K \]

\[ T_1 = \{ A, B, \langle A, N_A \rangle, \{\langle N_A, N_B \rangle\}_{K_{ab}}, N_B, \{\langle K, N_B \rangle\}_{K_{ab}}, \{s\}_K, \text{init, stop} \} \]

Roles

\[ R_A = (\text{init} \rightarrow \langle A, N_A \rangle), \]
\[ (\{\langle N_A, X_{N_B} \rangle\}_{K_{(A,X_B)}} \rightarrow X_{N_B}), \]
\[ (\{\langle X_K, X_{N_B} \rangle\}_{K_{(A,X_B)}} \rightarrow \{s\}_K X_K) \]

\[ R_B = (\langle X_A, X_{N_A} \rangle \rightarrow \{\langle X_{N_A}, N_B \rangle\}_{K_{(X_A,B)}}), \]
\[ (N_B \rightarrow \{\langle K, N_B \rangle\}_{K_{(X_A,B)}}), \]
\[ (\{X_s\}_{K} \rightarrow \text{stop}) \]
Solution

\[
\begin{align*}
  A & \rightarrow B : \langle A, N_A \rangle \\
  B & \rightarrow A : \{ \langle N_A, N_B \rangle \} K_{ab} \\
  A & \rightarrow B : N_B \\
  B & \rightarrow A : \{ \langle K, N_B \rangle \} K_{ab} \\
  A & \rightarrow B : \{ s \} K
\end{align*}
\]

\[
T_1 = \{ A, B, \langle A, N_A \rangle, \{ \langle N_A, N_B \rangle \} K_{ab}, N_B, \{ \langle K, N_B \rangle \} K_{ab}, \{ s \} K, \text{init}, \text{stop} \}
\]

Constraint System

\[
\begin{align*}
  T_1 & \\
  T_2 & = T_1 \cup \{ \langle A, N_A \rangle \} \quad \vdash \text{init} \\
  T_3 & = T_2 \cup \{ \{ \langle X_{N_A}, N_B \rangle \} K_{(X_A, B)} \} \quad \vdash \{ \langle N_A, X_{N_B} \rangle \} K_{(A, X_B)} \\
  T_4 & = T_3 \cup \{ X_{N_B} \} \quad \vdash N_B \\
  T_5 & = T_4 \cup \{ \{ \langle K, N_B \rangle \} K_{(X_A, B)} \} \quad \vdash \{ \langle X_K, X_{N_B} \rangle \} K_{(A, X_B)} \\
  T_6 & = T_5 \cup \{ \{ s \} X_K \} \quad \vdash \{ X_s \} K \\
  T_7 & = T_6 \cup \{ \text{stop} \} \quad \vdash s
\end{align*}
\]
## Resolution of Constraint systems

### Definition (Rules of simplification: $C \rightsquigarrow_\sigma C'$)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>$C \cup {T \vdash u} \rightsquigarrow C$ if $T \cup {x \mid T' \vdash x \in C, T' \subset T} \vdash u$</td>
</tr>
<tr>
<td>$R_2$</td>
<td>$C \cup {T \vdash u} \rightsquigarrow_\sigma C \sigma \cup {T \sigma \vdash u \sigma}$ $\sigma = \text{mgu}(t, u)$, $t \in \text{st}(T)$, $t, u$ no variables</td>
</tr>
<tr>
<td>$R_3$</td>
<td>$C \cup {T \vdash u} \rightsquigarrow_\sigma C \sigma \cup {T \sigma \vdash u \sigma}$ $\sigma = \text{mgu}(t_1, t_2)$, $t_1, t_2 \in \text{st}(T)$, $t_1, t_2$ no variables</td>
</tr>
<tr>
<td>$R_4$</td>
<td>$C \cup {T \vdash {u}_v} \rightsquigarrow C \cup {T \vdash u, T \vdash v}$</td>
</tr>
<tr>
<td>$R_5$</td>
<td>$C \cup {T \vdash \langle u, v\rangle} \rightsquigarrow C \cup {T \vdash u, T \vdash v}$</td>
</tr>
<tr>
<td>$R_6$</td>
<td>$C \cup {T \vdash u} \rightsquigarrow \bot$ if $T = \emptyset$ or $\text{var}(T) = \text{var}(u) = \emptyset$ and $T \not\vdash u$</td>
</tr>
</tbody>
</table>
Properties of simplification rules

Lemma (Preservation)

Simplification rules transform a constraint system into a constraint system.
## Properties of simplification rules

<table>
<thead>
<tr>
<th>Lemma (Preservation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Simplification rules transform a constraint system into a constraint system.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lemma (Correctness)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $C \leadsto_{\sigma} C'$ then if $\theta$ is a solution of $C'$, $\sigma \theta$ is also a solution of $C$.</td>
</tr>
</tbody>
</table>
Properties of simplification rules

Lemma (Preservation)

*Simplification rules transform a constraint system into a constraint system.*

Lemma (Correctness)

*If* $C \leadsto_\sigma C'$ *then if* $\theta$ *is a solution of* $C'$, *$\sigma \theta$ is also a solution of* $C$.

Lemma (Termination)

*Simplification rules always terminate: There does not exist infinite chain* $C \leadsto_{\sigma_1} C_1 \leadsto_{\sigma_2} C_2 \leadsto_{\sigma_3} \ldots$.
Properties

Definition (Solved Form)

A constraint system $C$ is in solved form if $C = \perp$ or if each constraint is of the following form $T \models x$ where $x$ is a variable $T \neq \emptyset$.

Lemma

*All constraint systems in solved form different of $\perp$ has at least one solution.*
## Properties

### Definition (Solved Form)

A constraint system $C$ is in **solved form** if $C = \bot$ or if each constraint is of the following form $T \models x$ where $x$ is a variable $T \neq \emptyset$.

### Lemma

*All constraint systems in solved form different of $\bot$ has at least one solution.*

### Lemma (Completeness)

*If $C$ is a constraint system not in solved form and if $\sigma$ is a solution of $C$ then there exists $\theta, \tau$ such that $C \rightsquigarrow_\theta C'$, $\sigma = \theta \tau$ and $\tau$ is a solution of $C'$.*
Decidability

**Theorem**

*Preservation of the secrecy for protocol with bounded number of sessions is decidable.*

- Guess an interleaving and build constraint system associated.
- Using previous lemma $C$ has a solution iff there exists $C'$ in solved form such that $C' \neq \bot$ and $C \leadsto_{\tau} C'$
- Using termination lemma to conclude.

We also can show that the problem is in co-NP.
Exercises

Exercise 1

\[A \rightarrow B : \langle A, N_A \rangle\]
\[B \rightarrow A : \{\langle N_A, N_B \rangle\}_{K_{ab}}\]
\[A \rightarrow B : N_B\]
\[B \rightarrow A : \{\langle K, N_B \rangle\}_{K_{ab}}\]
\[A \rightarrow B : \{s\}_K\]

Intruder knows only identities of \(A\) and \(B\).

- Use simplification rules to transform the system in solved form.
- There exists an easy attack, can you find it?
Solution

\[ T_1 = \{A, B, \langle A, N_A \rangle, \{\langle N_A, N_B \rangle\}, K_{ab}, N_B, \{\langle K, N_B \rangle\}, \{s\}, init, stop\} \]

\begin{align*}
C_1 & \quad T_1 \quad \models init \\
C_2 & \quad T_2 = T_1 \cup \{\langle A, N_A \rangle\} \quad \models \langle X_A, X_{N_A} \rangle \\
C_3 & \quad T_3 = T_2 \cup \{\langle X_{N_A}, N_B \rangle\}_{K_{(X_A, B)}} \quad \models \{\langle N_A, X_{N_B} \rangle\}_{K_{(A, X_B)}} \\
C_4 & \quad T_4 = T_3 \cup \{X_{N_B}\} \quad \models N_B \\
C_5 & \quad T_5 = T_4 \cup \{\langle K, N_B \rangle\}_{K_{(X_A, B)}} \quad \models \{\langle X_K, X_{N_B} \rangle\}_{K_{(A, X_B)}} \\
C_6 & \quad T_6 = T_5 \cup \{\{s\}_{X_K}\} \quad \models \{X_s\}_{K} \\
C_7 & \quad T_7 = T_6 \cup \{stop\} \quad \models s
\end{align*}

Road book


\begin{align*}
R_2 & \quad C \cup \{T \models u\} \looparrowright_{\sigma} C\sigma \cup \{T\sigma \models u\sigma\} \quad \sigma = \text{mgu}(t,u), t \in \text{st}(T), \\
& \quad \text{t, u no variables} \\
& \quad \bullet \text{ Apply nothing on } C_1, \text{ already in resolved form.} \\
& \quad \bullet \text{ Apply } R_2 \text{ on } C_2 \text{ give } \sigma_1 = \{X_{N_A} \rightarrow N_A, X_A \rightarrow A\} \text{ and } R_1
\end{align*}
Solution

\[ T_1 = \{A, B, \langle A, N_A \rangle, \{\langle N_A, N_B \rangle\}_{K_{ab}}, N_B, \{\langle K, N_B \rangle\}_{K_{ab}}, \{s\}_K, init, stop\} \]

\[ C_3\sigma_1 \quad T_3 = T_2 \cup \{\{\langle N_A, N_B \rangle\}_{K_{(A,B)}}\} \quad \vdash \quad \{\langle N_A, X_{N_B} \rangle\}_{K_{(A,X_B)}} \]

\[ C_4\sigma_1 \quad T_4 = T_3 \cup \{X_{N_B}\} \quad \vdash \quad N_B \]

\[ C_5\sigma_1 \quad T_5 = T_4 \cup \{\{\langle K, N_B \rangle\}_{K_{(A,B)}}\} \quad \vdash \quad \{\langle X_K, X_{N_B} \rangle\}_{K_{(A,X_B)}} \]

\[ C_6\sigma_1 \quad T_6 = T_5 \cup \{\{s\}_K \} \quad \vdash \quad \{X_s\}_K \]

\[ C_7\sigma_1 \quad T_7 = T_6 \cup \{stop\} \quad \vdash \quad s \]

Road book \( \sigma_1 = \{X_{N_A} \rightarrow N_A, X_A \rightarrow A\} \)

- Apply \( R_2 \) on \( C_3 \) give \( \sigma_2 = \{X_{N_B} \rightarrow N_B, X_B \rightarrow B\} \) (or \( N_A \)) and \( R_1 \)
Solution

\[ T_1 = \{A, B, \langle A, N_A \rangle, \{\langle N_A, N_B \rangle \}_{K_{ab}}, N_B, \{\langle K, N_B \rangle \}_{K_{ab}}, \{s\}_K, \text{init, stop}\} \]

\[ C_5 \sigma_1 \sigma_2 \quad T_5 = T_4 \cup \{\{\langle K, N_B \rangle \}_{K(A,B)}\} \quad \models \quad \{\langle X_K, N_B \rangle \}_{K(A,B)} \]

\[ C_6 \sigma_1 \sigma_2 \quad T_6 = T_5 \cup \{\{s\}_X \}_{K} \quad \models \quad \{X_s\}_K \]

\[ C_7 \sigma_1 \sigma_2 \quad T_7 = T_6 \cup \{\text{stop}\} \quad \models \quad \text{s} \]

Road book \( \sigma_1 = \{X_{N_A} \rightarrow N_A, X_A \rightarrow A\} \quad \sigma_2 = \{X_{N_B} \rightarrow N_B, X_B \rightarrow B\} \)

- Apply \( R_2 \) on \( C_5 \sigma_1 \sigma_2 \) give \( \sigma_3 = \{X_K \rightarrow N_A\} \)
- Apply \( R_2 \), on \( \sigma_1 \sigma_2 \sigma_3 C_6 \) give \( \sigma_4 = \{X_s \rightarrow s\} \)
Solution

1. $A \rightarrow B : \langle A, N_A \rangle$
2. $B \rightarrow A : \{\langle N_A, N_B \rangle\}_{K_{ab}}$
3. $A \rightarrow B : N_B$
4. $B \rightarrow A : \{\langle K, N_B \rangle\}_{K_{ab}}$
5. $A \rightarrow B : \{s\}_K$

The resolution of constraint system gives the following attack:
Send 2nd message $\{\langle N_A, N_B \rangle\}_{K_{ab}}$ instead of the 4th message $\{\langle K, N_B \rangle\}_{K_{ab}}$. Hence $A$ will replay $\{s\}_{N_A}$ because intruder knows $N_A$.
Exercises

Exercise 2

\[ A \rightarrow B : \quad \{ \langle A, K \rangle \}^{K_{ab}} \]
\[ B \rightarrow A : \quad \{ s \}^{K_{ab}} \]

Intruder knows only identities of A and B. Show that the secret data \( s \) is preserved by one single session between A and B.
Solution

\[ A \rightarrow B : \{\langle A, K \rangle\}_{K_{ab}} \]
\[ B \rightarrow A : \{s\}_{K_{ab}} \]

\[ T_1 = \{A, B, \{\langle A, K \rangle\}_{K_{ab}}, \{s\}_{K_{ab}}\} \]

Constraint System

<table>
<thead>
<tr>
<th>Constraint</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 )</td>
<td>( T_1 ) ||= {\langle A, X_K \rangle}<em>{K</em>{ab}}</td>
</tr>
<tr>
<td>( C_2 )</td>
<td>( T_2 = T_1 \cup {\langle A, X_K \rangle}<em>{K</em>{ab}} ) ||= {s}<em>{X</em>{K_{ab}}}</td>
</tr>
<tr>
<td>( C_3 )</td>
<td>( T_3 = T_2 \cup {s}<em>{X</em>{K_{ab}}} ) ||= s</td>
</tr>
</tbody>
</table>
Solution

\[ C_1 \ T_1 \quad \vdash \quad \{ \langle A, X_K \rangle \} X_{K_{ab}} \]
\[ C_2 \ T_2 = T_1 \cup \{ \langle A, X_K \rangle \} X_{K_{ab}} \quad \vdash \quad \{ s \} X_{K_{ab}} \]
\[ C_3 \ T_3 = T_2 \cup \{ s \} X_{K_{ab}} \quad \vdash \quad s \]

\[ T_1 = \{ A, B, \{ \langle A, K \rangle \}_{K_{ab}}, \{ s \}_{K_{ab}} \} \]

Road book

- Apply nothing or \( R_4 \) or \( R_5 \) and \( R_2 \) on \( C_1 \) give
  \[ \sigma_0 = \{ X_K \rightarrow K, X_{K_{ab}} \rightarrow K_{ab} \} \]

- Apply \( R_5 \) or nothing and \( R_2 \), on \( \sigma_0 C_2 \) give \( \sigma_1 = \{ X_{N_B} \rightarrow N_B \} \)
  (or \( N_A \))

Each time you meet a solved form of the form \( \bot \) with \( R_6 \).
Outline

1. Active Intruder: Security Problem
2. Bounded Number of Sessions
3. NP-Hardness for Bounded Number of Sessions
4. Unbounded number of sessions
5. Conclusion
NP-hardness

Theorem

Decide if a protocol $P$ does not preserve the secrecy of a ground term $s$ from an initial knowledge $T_1$ is NP-difficult.
Recall 3-SAT Problem

Definition

Input: set of propositional variables \( \{x_1, \ldots, x_n\} \) and a conjunction of clauses with 3 literals.

\[
f(\vec{x}) = \bigwedge_{1 \leq i \leq l} (x_{i,1}^{\epsilon_i,1} \lor x_{i,2}^{\epsilon_i,2} \lor x_{i,3}^{\epsilon_i,3})
\]

where \( \epsilon_{i,j} \in \{+, -\} \) and \( x^+ = x, x^- = \neg x \).

Question: Does exist a valuation \( V \) of \( \{x_1, \ldots, x_n\} \), such that \( V(f(\vec{x})) = \top \).

Theorem

3-SAT problem is NP-complete.
NP-difficulty

We build a protocol such that an intruder can deduce $s$ \iff $f(\vec{x})$ is satisfiable.

\[
g(x_{i,j}^{\epsilon_i,j}) = \begin{cases} 
  x_{i,j} & \text{if } \epsilon_{i,j} = + \\
  \{x_{i,j}\} & \text{if } \epsilon_{i,j} = - 
\end{cases}
\]

\[
\forall 1 \leq i \leq l : \quad f_i(\vec{x}) = \langle g(x_{i,1}^{\epsilon_i,1}), g(x_{i,2}^{\epsilon_i,2}), g(x_{i,3}^{\epsilon_i,3}) \rangle
\]

We suppose Initial intruder knowledge is $\{\bot, \top\}$.

\[
A : \quad \langle x_1, \langle \ldots, x_n \rangle \rangle \rightarrow \{\langle f_1(\vec{x}), \langle f_2(\vec{x}), \langle \ldots, \langle f_n(\vec{x}), \text{end}\rangle \ldots \rangle \rangle \}_p
\]

\[
\forall 1 \leq i \leq l :
\begin{align*}
B_i & : \quad \{\langle \langle \top, \langle x, y \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p \\
\overline{B}_i & : \quad \{\langle \langle \bot \rangle^K, \langle x, y \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p \\
C_i & : \quad \{\langle \langle x, \langle \top, y \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p \\
\overline{C}_i & : \quad \{\langle \langle x, \langle \bot \rangle^K, y \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p \\
D_i & : \quad \{\langle \langle x, \langle y, \top \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p \\
D_i & : \quad \{\langle \langle x, \langle y, \langle \bot \rangle^K \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p \\
E & : \quad \{\text{end}\}_p \rightarrow s
\end{align*}
\]
Outline

1. Active Intruder: Security Problem
2. Bounded Number of Sessions
3. NP-Hardness for Bounded Number of Sessions
4. Unbounded number of sessions
5. Conclusion
Recall: Horn Clauses

Definition

A Horn clause is a formula of the following form
\[ p_1 \land \ldots \land p_n \rightarrow p \]

Definition (Horn-SAT problem)

**Inputs:** a set of Horn clauses \( H \)

**Question:** Does exist a valuation \( V \) such that

\[ \forall \phi \in H. \ V \models \phi \]

Theorem (Horn-SAT)

*Horn-SAT problem is decidable in linear time* \(|H|\).
Horn Clauses

A Horn clause is a logical formula of the form

\[
\frac{L_1, \ldots, L_n}{L} \quad (\equiv \neg L_1 \lor \ldots \lor \neg L_n \lor L)
\]

Formalism simple and homogeneous for
- modeling intruder capabilities
- modeling protocol rules
- checking an unbounded number of sessions

This formalism is used like intermediary representation (translation from high level language “Pi-calculus like”) in the Tool ProVerif
[Blanchet2001]

http://www.di.ens.fr/~blanchet/crypto.html
Syntactic representation of protocols

\[ T ::= \text{term} \]
\[ \quad \mid x \quad \text{variable } x \]
\[ \quad \mid a[T_1, \ldots, T_n] \quad \text{name } a \]
\[ \quad \mid f(T_1, \ldots, T_k) \quad \text{application of symbol } f \in \Sigma \]
\[\quad \quad \quad (\text{Arity}(f) = k) \]

\[ F ::= \text{facts} \]
\[ \quad \mid p(M_1, \ldots, M_n) \quad \text{application of predcat } p \]

\[ R ::= \text{rule} \]
\[ \quad \mid F_1 \land \ldots \land F_n \rightarrow F \quad \text{implication} \]
Modeling of cryptographic primitives

Cryptographic primitives are represented by functions.

Example:
Symmetric encryption of a message $m$ by the key $k$ is represented by a function of arity 2 $\text{encrypt}(m, k)$.

Let $\Sigma$ the signature containing a set of functions. We can split this set into two sets: constructors and destructors.

- ** Constructors** are functions which are explicitly in terms.
- ** Destructors** change terms.

A destructor $g$ is defined by an equation $g(T_1, \ldots, T_k) = T$ where $T_1, \ldots, T_k, T$ have only constructors and variables.
Examples

Symmetric Encryption
is defined with one constructor $\text{encrypt}(m, k)$ and one destructor $\text{decrypt}(\text{encrypt}(m, k)), k) = m$

Signature
is modeled by two constructors $\text{sign}(m, sk)$ and $\text{pk}(sk)$ and one destructor $\text{getmsg}(\text{sign}(m, sk), \text{pk}(sk)) = m$

Hash Function
is represented by one constructor $\text{h}(m)$
Intruder capabilities with Horn clauses

- Predicate $I(m)$ models **intruder knowledge**
- $I(m)$ is true iff intruder knows the message $m$

### Intruder Capabilities

<table>
<thead>
<tr>
<th>Rule</th>
<th>Horn Clause</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I(m), I(n) \over I(pair(m, n))$</td>
<td>(pair)</td>
</tr>
<tr>
<td>$I(pair(m, n)) \over I(m)$</td>
<td>(UL)</td>
</tr>
<tr>
<td>$I(pair(m, n)) \over I(n)$</td>
<td>(UR)</td>
</tr>
<tr>
<td>$I(m), I(pubk) \over I(enc(m, pubk))$</td>
<td>(encrypt)</td>
</tr>
<tr>
<td>$I(enc(m, pk(x))), I(x) \over I(m)$</td>
<td>(decrypt)</td>
</tr>
</tbody>
</table>
Others Intruder Capabilities (2)

Let $f$ a constructor with arity $n$:

$$I(x_1), \ldots, I(x_n) \over I(f(x_1, \ldots, x_n))$$

Let $g$ a destructor defined by the equation $g(T_1, \ldots, T_n) = T$

$$I(T_1), \ldots, I(T_n) \over I(T)$$

Remark:
Symbol of the destructor $g$ does not appear in the rules.
Initial knowledge of the Intruder

Let $T$ a ground term.

$$\rightarrow I(T)$$

Example:
Intruder knows public key corresponding to his secret key $sA[]$ of $A$:

$$I(pk(sA[]))$$
Protocols rules by Horn clauses

Needham-Schroeder is modeled by the following Horn clauses

\[
\begin{align*}
A & \rightarrow B : \{N_a, A\}_{\text{pub}(B)} \\
B & \rightarrow A : \{N_a, N_b\}_{\text{pub}(A)} \\
A & \rightarrow B : \{N_b\}_{\text{pub}(B)}
\end{align*}
\]

\[
I(pk(x)) \\
\overline{I(\text{enc}((Na[pk(x)], pk(sA[])), pk(x)))}
\]

\[
I(encrypt((x, y), pk(sB[]))) \\
\overline{I(encrypt((x, Nb[x, y]), y))}
\]

\[
I(pk(x)), I(encrypt((Na[pk(x)], y), pk(sA[]))) \\
\overline{I(encrypt(y, pk(x)))}
\]

Modeling of first message

We assume that intruder chooses with whom A plays the protocol.
Protocols rules by Horn clauses

Needham-Schroeder is modeled by the following Horn clauses

\[
\begin{align*}
A \rightarrow B & : \{N_a, A\}_{pub(B)} \\
B \rightarrow A & : \{N_a, N_b\}_{pub(A)} \\
A \rightarrow B & : \{N_b\}_{pub(B)}
\end{align*}
\]

\[
\begin{align*}
I(pk(x)) \\
I(encrypt((x, y), pk(sB[]))) & \Rightarrow I(encrypt((x, Nb[x, y]), y)) \\
I(pk(x)), I(encrypt((Na[pk(x)], y), pk(sA[]))) & \Rightarrow I(encrypt(y, pk(x)))
\end{align*}
\]

Modeling of Nonces

Nonces are modeled by functions of parameters of the protocols.
Protocols rules by Horn clauses

Needham-Schroeder is modeled by the following Horn clauses

\[
\begin{align*}
A & \rightarrow B : \{N_a, A\}_{pub(B)} \\
B & \rightarrow A : \{N_a, N_b\}_{pub(A)} \\
A & \rightarrow B : \{N_b\}_{pub(B)} \\
I(pk(x)) & \\
I(enc((Na[pk(x)], pk(sA[])), pk(x))) & \\
I(encrypt((x, y), pk(sB[]))) & \\
I(encrypt((x, Nb[x, y]), y)) & \\
I(pk(x)), I(encrypt((Na[pk(x)], y), pk(sA[]))) & \\
I(encrypt(y, pk(x))) &
\end{align*}
\]

Intruder controls the network

We assume that all messages are exchange via the intruder.
Protocols rules by Horn clauses

Needham-Schroeder is modeled by the following Horn clauses

\[ A \rightarrow B : \{N_a, A\} \text{pub}(B) \]
\[ B \rightarrow A : \{N_a, N_b\} \text{pub}(A) \]
\[ A \rightarrow B : \{N_b\} \text{pub}(B) \]

\[
\begin{align*}
I(pk(x)) \\
I(\text{enc}((Na[pk(x)], pk(sA[])), pk(x))) \\
I(\text{encrypt}((x, y), pk(sB[]))) \\
I(\text{encrypt}((x, Nb[x, y]), y)) \\
I(pk(x)), I(\text{encrypt}((Na[pk(x)], y), pk(sA[]))) \\
I(\text{encrypt}(y, pk(x)))
\end{align*}
\]

Intruder controls the network

We assume that all messages are exchange via the intruder.
Protocols rules by Horn clauses

Needham-Schroeder is modeled by the following Horn clauses

\[ A \rightarrow B : \{ N_a, A \}_{pub(B)} \]
\[ B \rightarrow A : \{ N_a, N_b \}_{pub(A)} \]
\[ A \rightarrow B : \{ N_b \}_{pub(B)} \]

\[ I(pk(x)) \]
\[ I(\text{enc}((Na[pk(x)], pk(sA[])), pk(x))) \]
\[ I(\text{encrypt}((x, y), pk(sB[]))) \]
\[ I(\text{encrypt}((x, Nb[x, y]), y)) \]

\[ I(pk(x)), I(\text{encrypt}((Na[pk(x)], y), pk(sA[]))) \]
\[ I(\text{encrypt}(y, pk(x))) \]

Intruder controls the network
We assume that all messages are exchange via the intruder.
Approximations

- **Nonces** are modeled by functions of previous received messages. If an intruder sends the same messages, then the same nonces will be used.

- One step of the protocol can be executed several times if previous steps are executed at least once.

Example:

1. Intruder sends to A the message $M_1$
2. A answers by $M_2$
3. Intruder sends to A the message $M_3$
4. A answers by $M_4$
5. Intruder sends to A the message $M'_3$ (without executing the 2 first steps)
6. A replies with $M'_4$
Correct Approximations

Approximations can lead to false attacks

On studied protocols, we find few false attacks.

Approximations are correct

If we prove correctness of a protocol in Horn clauses, then the protocol is also correct in a model more precise.
Exercise

\[
A \rightarrow B : \quad \{\langle A, K \rangle\}_{K_{ab}}
\]

\[
B \rightarrow A : \quad \{s\}_{K_{ab}}
\]

Intruder knows only identities of \( A \) and \( B \). Show that the secret data \( s \) is preserved by one single session between \( A \) and \( B \).

Give a modeling of this protocol in Horn clauses, with capabilities of the intruder.
Solution

\[ A \rightarrow B : \{ \langle A, K \rangle \}_{K_{ab}} \]
\[ B \rightarrow A : \{ s \}_{K_{ab}} \]

Modeling

\[ I(pk(sk(Y))) \]
\[ I(enc((pk(sk(A)), K[pk(sk(Y)]), sk(AB)))] \]

\[ I(enc((X, Z), sk(AB))) \]
\[ I(enc(s, sk(AB))) \]

\[ I(m), I(n) \]
\[ I(pair(m, n)) \]

\[ I(pair(m, n)) \]
\[ I(m) \]
\[ I(n) \]

\[ I(m), I(pubk) \]
\[ I(enc(m, pubk)) \]

\[ I(m), I(pk(x)) \]
\[ I(x) \]

\[ I(pair(m, n)) \]
\[ I(m) \]
\[ I(n) \]
Derivability

Implication between rules

\[(H_1 \rightarrow C_1) \Rightarrow (H_2 \rightarrow C_2)\] iff there exists a substitution \(\sigma\) such that \(C_1\sigma = C_2\) and \(H_1\sigma = H_2\) \((H_1\) and \(H_2\) are sets of hypothesis)

Definition (Derivability)

Let \(F\) a ground fact and \(B\) a set of rules. \(F\) is derivable from \(B\) iff there exists a finite tree such that:

1. All nodes (except the root) are labelled by rule \(R \in B\)
2. edges are labelled by facts
3. if a tree has a node labelled by a rule \(R\) with an input edge, labelled \(F_0\) and \(n\) output edges labelled \(F_1, \ldots, F_n\) then \(R \Rightarrow \{F_1, \ldots, F_n\} \rightarrow F_0\)
4. The root has an output edge labelled by \(F\)
Secrecy property

Definition

A ground term $S$ is secret if it is not possible to derive $I(S)$ from rules modeling protocol and intruder capabilities.

Example

\[ I(x) \land I(y) \rightarrow I((x, y)) \] (1) \hspace{1cm} I(m[]) \] (4)
\[ I(x) \land I(y) \rightarrow I(encrypt(x, y)) \] (2) \hspace{1cm} I(n[]) \] (5)
\[ I(pk(sA[])) \] (3)

Is it possible to derive $I(encrypt((m[], n[]), pk(sA[])))$?
Solution

Example

\[ I(x) \land I(y) \rightarrow I((x, y)) \quad (1) \quad I(m[]) \quad (4) \]
\[ I(x) \land I(y) \rightarrow I(\text{encrypt}(x, y)) \quad (2) \quad I(n[]) \quad (5) \]
\[ I(\text{pk}(sA[])) \quad (3) \]

We can derive \( I(\text{encrypt}((m[], n[]), \text{pk}(sA[]))) \):

![Diagram](image-url)
Automatisation ?

Is it possible to derive a fact $F$ from a given set of rules (Horn clauses)

It is the same problem solve by Prolog

But: algorithms used in Prolog does not terminate for rules usually used in cryptographic protocols.

In [Blanchet2001], Bruno Blanchet presents a new algorithm for the resolution which “guides” the resolution and which is adapted for cryptographic protocols.
Some Definitions ...

Combination of simplified rules

Let \( R = H \rightarrow C \) and \( R' = H' \rightarrow C' \) two rules. If \( C \in H' \) then

\[
R \circ R' = H \cup (H' \setminus C) \rightarrow C'
\]

Definition (Combination of rules)

Let \( R = H \rightarrow C \) and \( R' = H' \rightarrow C' \) two rules. Suppose that there exists a fact \( F_0 \in H' \), such that \( F_0 \) and \( C \) are unifiable and \( \sigma \) i the mgu for \( C \) and \( F_0 \). Then

\[
R \circ_{F_0} R' = (H \cup (H' \setminus F_0))\sigma \rightarrow C'\sigma
\]
Example:

\[ R = I(pk(x)) \rightarrow I(encrypt(sign(msg[], skA[]), pk(x))) \]

\[ R' = I(encrypt(m, pk(sk))) \land I(sk) \rightarrow I(m) \]

Consider \( F_0 = I(encrypt(m, pk(sk)) \). Then

\[ R \circ_{F_0} R' = I(pk(x)) \land I(x) \rightarrow I(sign(msg[], skA[])) \]

with \( \sigma = \{sk = x, m = sign(msg[], skA[])\} \)
Heuristic for the algorithm

Let $S$ be a finite set of facts. We define $F \in_r S$ iff there exists a substitution $\sigma$ of variables by some others variables such that $F\sigma \in S$.

In the algorithm, $S$ is used to guide the choice of rules combinations: we do not combine $R$ and $R'$ by $R \circ F_0 R'$ if $F_0 \in_r S$.

We take $S = \{I(x)\}$ to avoid the following situation.

$$I(x) \rightarrow I(pk(x)) \quad \text{(1)}$$
$$I(pk(x)) \land I(y) \rightarrow \text{encrypt}(y, pk(x)) \quad \text{(2)}$$

If we apply the combination $(2) \circ I(x)$ (1) we get

$$I(pk(x)) \land I(y) \rightarrow I(pk(\text{encrypt}(y, pk(x)))) \quad \text{(3)}$$
Heuristic for the algorithm

\[ I(x) \rightarrow I(pk(x)) \quad (1) \]
\[ I(pk(x)) \land I(y) \rightarrow encrypt(y, pk(x)) \quad (2) \]
\[ I(pk(x)) \land I(y) \rightarrow I(pk(encrypt(y, pk(x)))) \quad (3) \]

Then we can apply the combination \((3) \circ I(x) (1)\) and get

\[ I(pk(x)) \land I(y) \rightarrow I(pk(pk(encrypt(y, pk(x)))))) \quad (4) \]

We clearly see that successive combinations does not terminate. Similarly if we chose \(I(y) \in S\) as \(F_0\) we have a loop on encryption.
Resolution Algorithm: phase 1

Let \( add(R, B) = \begin{cases} B \\ \{R\} \cup \{R' \in B \mid R \nRightarrow R'\} \end{cases} \)

if \( \exists R' \in B, R' \Rightarrow R \)

otherwise

Let \( B_0 \) the set of rules describing the protocol and the intruder

1. For all \( R \in B_0 \) \( B = add(R, B) \)

2. Let \( R \in B, R = H \rightarrow C \) and \( R' \in B, R' = H' \rightarrow C' \). Suppose that there exists \( F_0 \in H' \) such that
   
   (a) \( R \circ_{F_0} R' \) is defined
   (b) \( \forall F \in H, F \in_r S \)
   (c) \( F_0 \notin_r S \)

   then \( B = add(R \circ_{F_0} R', B) \)

   Execute step 2. until reach a fix point.

3. \( B' = \{(H \rightarrow C) \in B \mid \forall F \in H, F \in_r S\} \)

After the execution of phase 1., we have a ground fact \( F \) which can be derived from \( B' \) iff \( F \) can be derived by \( B_0 \).
Resolution Algorithm: phase 2

derivable rec \((R, B''')\)

1. \( \text{derivablerrec}(R, B''') = \emptyset \)
   \( \text{if } \exists \ \exists \ R' \in B'''.R' \Rightarrow R \)  \# loop: backtrack

2. otherwise, \( \text{derivablerrec}(\emptyset \rightarrow C, B''') = \{C\} \)  \# proof of C

3. otherwise,
   \( \text{derivablerrec}(R, B''') = \bigcup \{ \text{derivablerrec}(R' \circ F_0 R, \{R\} \cup B''') \mid R' \in B', F_0 \text{ is such that } R' \circ F_0 R \text{ us defined } \} \)

\( \text{derivable}(F) = \text{derivablerrec}(\{F\} \rightarrow F, \emptyset) \)

Intuitively

- Hypothesis of \( R \) contains fact that we try at one time
- Conclusion of \( R \) contains fact that we try to derive
- Set \( B''' \) is the set of rules already meet

We have \( F \) is derivable from \( B_0 \) iff \( F \in \text{derivable}(F) \)
Remarks on the algorithm

Fix point of the first phase does not always terminate

In practice, on examples of protocols, this phase terminates.

We can show that if \( S = \{ I(x) \} \) and if \( F \) is a ground fact, then \( \text{derivable}(F) \) terminates

When Proverif terminates ...

For a tagged protocol Proverif always terminates.

Idea: adding a constant inside all “cryptographic” functions.

ProVerif tools has many extensions et optimizations ...

**Verification of Cryptographic Protocols: Tagging Enforces Termination**, by Bruno Blanchet and Andreas Podelski.
One Example of Tagged Protocol

Yahalom Protocol

1. \( A \rightarrow B : (A, N_a) \)
2. \( B \rightarrow S : (B, \{A, N_a, N_b\}_{Kbs}) \)
3. \( S \rightarrow A : (\{B, Kab, N_a, N_b\}_{Kas}, \{A, Kab\}_{Kbs}) \)
4. \( A \rightarrow B : (\{A, Kab\}_{Kbs}\{N_b\}_{Kab}) \)
One Example of Tagged Protocol

**Yahalom Protocol**

1. \( A \rightarrow B : (A, N_a) \)
2. \( B \rightarrow S : (B, \{A, N_a, N_b\}_{Kbs}) \)
3. \( S \rightarrow A : (\{B, Kab, N_a, N_b\}_{Kas}, \{A, Kab\}_{Kbs}) \)
4. \( A \rightarrow B : (\{A, Kab\}_{Kbs}, \{N_b\}_{Kab}) \)

**Tagged Yahalom Protocol**

1. \( A \rightarrow B : (A, N_a) \)
2. \( B \rightarrow S : (B, \{c_1, A, N_a, N_b\}_{Kbs}) \)
3. \( S \rightarrow A : (\{c_2, B, Kab, N_a, N_b\}_{Kas}, \{c_3, A, Kab\}_{Kbs}) \)
4. \( A \rightarrow B : (\{c_3, A, Kab\}_{Kbs}, \{c_4, N_b\}_{Kab}) \)
One Example of Tagged Protocol

Yahalom Protocol

1. $A \rightarrow B : (A, N_a)$
2. $B \rightarrow S : (B, \{A, N_a, N_b\}_{Kbs})$
3. $S \rightarrow A : (\{B, Kab, N_a, N_b\}_{Kas}, \{A, Kab\}_{Kbs})$
4. $A \rightarrow B : (\{A, Kab\}_{Kbs}\{N_b\}_{Kab})$

Tagged Yahalom Protocol

1. $A \rightarrow B : (A, N_a)$
2. $B \rightarrow S : (B, \{c_1, A, N_a, N_b\}_{Kbs})$
3. $S \rightarrow A : (\{c_2, B, Kab, N_a, N_b\}_{Kas}, \{c_3, A, Kab\}_{Kbs})$
4. $A \rightarrow B : (\{c_3, A, Kab\}_{Kbs}\{c_4, N_b\}_{Kab})$

This protocol is secure!
Outline

1. Active Intruder: Security Problem
2. Bounded Number of Sessions
3. NP-Hardness for Bounded Number of Sessions
4. Unbounded number of sessions
5. Conclusion
Summary

Today

- Active Intruder
- Bounded Number of Sessions
- NP-Hardness
- Unbounded Number of Sessions
Next Time

- Playing with Tools:
  - Scyther
  - Avispa: OFMC, CI-Atse, SATMC, TA4SP
  - Proverif
Thank you for your attention

Questions ?