A Theory of Dictionary Attacks and its Complexity

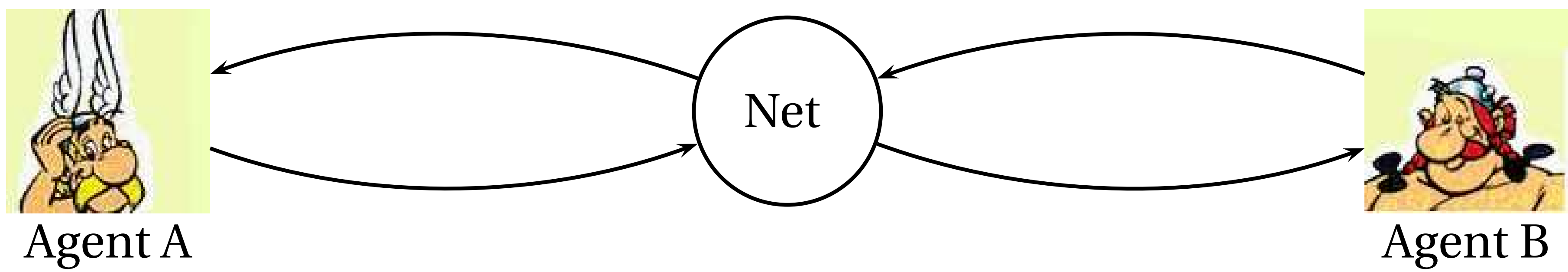
Stéphanie Delaune[1,2], **Florent Jacquemard**[1,3]

[1] LSV, CNRS UMR 8643, ENS de Cachan

[2] France Telecom R&D

[3] INRIA, Research Unit Futurs, Project SECSI

Cryptographic Protocols



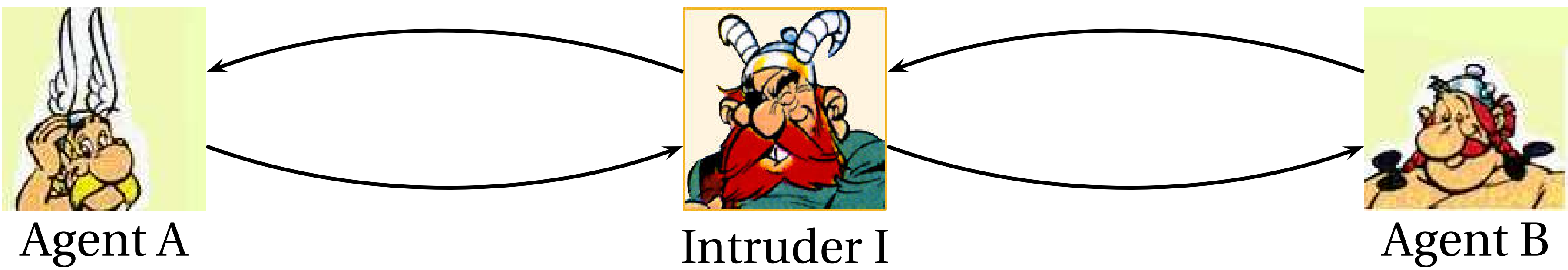
Protocol

↪ rules of message exchanges

Goal

↪ secure communications: *secrecy, authentication, anonymity* ...

Cryptographic Protocols



Messages $\rightsquigarrow$ ground terms

$\hookrightarrow$ pairing $\langle m_1, m_2 \rangle$

$\hookrightarrow$ encryption $\{m\}_k$ (symmetric / asymmetric)

$\hookrightarrow$ $\mathsf{pub}(a)$, $\mathsf{priv}(a)$, $\mathsf{pw}(a, b)$ ...

Intruder $\rightsquigarrow$ deductions modeled by an inference system

Dolev-Yao Model for Intruder's Deductions

Let be u, v terms and T a finite set of terms *(intruder's knowledge)*

Axiom (A) $$\frac{u \in T}{T \vdash u}$$

Pairing (P) $$\frac{T \vdash u \quad T \vdash v}{T \vdash \langle u, v \rangle}$$

Unpairing (UL) $$\frac{T \vdash \langle u, v \rangle}{T \vdash u}$$

Unpairing (UR) $$\frac{T \vdash \langle u, v \rangle}{T \vdash v}$$

Encryption (E) $$\frac{T \vdash u \quad T \vdash v}{T \vdash \{u\}_v}$$

Decryption (D) $$\frac{T \vdash \{u\}_v \quad T \vdash v^{-1}}{T \vdash u}$$

$\hookrightarrow$ **Perfect Cryptography Assumption**

no way to obtain knowledge about u from $\{u\}_v$ without knowing v^{-1}

Results in the Dolev-Yao Model

Passive Attacker *intruder deduction problem*

Given a set of messages T and a message s, does there exist a proof of $T \vdash s$ **?**

$\hookrightarrow$ PTIME–complete

Active Attacker *insecurity problem for a finite number of sessions*

Given $\mathcal{P}$, s, $\mathcal{S}_0$ and T_0, does there exist a running sequence of protocol rules such that:

- at the end, the intruder's knowledge is T,
- there exists a proof of $T \vdash s$ **?**

$\hookrightarrow$ NP–complete [Rusinowitch&Turuani, 01]

Example 1 - Vote Protocol

Consider this exchange:

$$A \quad \rightarrow S : \quad \{vote\}_{\mathsf{pub}(S)}$$

with $vote \in \mathcal{D} = \{yes, no\}$.

Example 1 - Vote Protocol

Consider this exchange:

$$A \quad \rightarrow S : \quad \{vote\}_{\mathsf{pub}(S)}$$

with $vote \in \mathcal{D} = \{yes, no\}$.

Description of the dictionary attack:

Example 1 - Vote Protocol

Consider this exchange:

$$A \quad \rightarrow S : \quad \{vote\}_{\mathsf{pub}(S)}$$

with $vote \in \mathcal{D} = \{yes, no\}$.

Description of the dictionary attack:

- guess the value chosen by A: yes,

Example 1 - Vote Protocol

Consider this exchange:

$$A \quad \to S: \quad \{vote\}_{\mathsf{pub}(S)}$$

with $vote \in \mathcal{D} = \{yes, no\}$.

Description of the dictionary attack:

- guess the value chosen by A: yes,
- build the verifier: $\{yes\}_{\mathsf{pub}(S)}$,

Example 1 - Vote Protocol

Consider this exchange:

$$A \quad \rightarrow S : \quad \{vote\}_{\mathsf{pub}(S)}$$

with $vote \in \mathcal{D} = \{yes, no\}$.

Description of the dictionary attack:

- guess the value chosen by A: yes,
- build the verifier: $\{yes\}_{\mathsf{pub}(S)}$,
- compare it with the intercepted message: $\{vote\}_{\mathsf{pub}(S)}$.

Example 1 - Vote Protocol

Consider this exchange:

$$A \quad \rightarrow S : \quad \{vote\}_{\mathsf{pub}(S)}$$

with $vote \in \mathcal{D} = \{yes, no\}$.

Description of the dictionary attack:

- guess the value chosen by A: yes,
- build the verifier: $\{yes\}_{\mathsf{pub}(S)}$,
- compare it with the intercepted message: $\{vote\}_{\mathsf{pub}(S)}$.

Result:

the intruder knows $vote$, the presumed secret vote of A.

Example 2 - Nonce Challenge

Consider the two-messages handshake transaction: [Gong et al, 93]

$$\begin{aligned} A &\to B : \quad \{n\}_{\mathsf{pw}(A,B)} \\ B &\to A : \quad \{n+1\}_{\mathsf{pw}(A,B)} \end{aligned}$$

where $\mathsf{pw}(A, B) \in \mathcal{D} = \{p_1, \ldots, p_n\}$ is a poorly-chosen symmetric key and n is a nonce.

Example 2 - Nonce Challenge

Consider the two-messages handshake transaction: [Gong et al, 93]

$$\begin{aligned} A &\to B: \quad \{n\}_{\mathsf{pw}(A,B)} \\ B &\to A: \quad \{n+1\}_{\mathsf{pw}(A,B)} \end{aligned}$$

where $\mathsf{pw}(A,B) \in \mathcal{D} = \{p_1, \ldots, p_n\}$ is a poorly-chosen symmetric key and n is a nonce.

Description of the dictionary attack:

- guess the value of $\mathsf{pw}(A,B)$: p_i,
- decrypt the two intercepted messages $\{n\}_{\mathsf{pw}(A,B)}$ and $\{n+1\}_{\mathsf{pw}(A,B)}$ with the password p_i,
- verify if the second obtained value is the increment of the first one.

Example 2 - Nonce Challenge

Consider the two-messages handshake transaction: [Gong et al, 93]

$$\begin{aligned} A &\rightarrow B: \quad \{n\}_{\mathsf{pw}(A,B)} \\ B &\rightarrow A: \quad \{n+1\}_{\mathsf{pw}(A,B)} \end{aligned}$$

where $\mathsf{pw}(A, B) \in \mathcal{D} = \{p_1, \ldots, p_n\}$ is a poorly-chosen symmetric key and n is a nonce.

Description of the dictionary attack:

- guess the value of $\mathsf{pw}(A, B)$: p_i,
- decrypt the two intercepted messages $\{n\}_{\mathsf{pw}(A,B)}$ and $\{n+1\}_{\mathsf{pw}(A,B)}$ with the password p_i,
- verify if the second obtained value is the increment of the first one.

Result:

the intruder knows the password $\mathsf{pw}(A, B)$, and the challenge n.

Example 3 - Enhanced Kerberos Protocol

Consider the following protocol: [Gong et al, 93]

$$
\begin{array}{llll}
0. & A & \rightarrow S: & \{A, B, N1, N2, Ca, \{Ta\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\
1. & S & \rightarrow A: & \{N1, K \oplus N2\}_{\mathsf{pw}(A,S)}, \{A, K, Ts\}_{\mathsf{pw}(B,S)} \\
2. & A & \rightarrow B: & \{A, K, Ts\}_{\mathsf{pw}(B,S)}
\end{array}
$$

Goal:

distributed a secret symmetric key K to be shared between the user A and the ticket-granting service B

Where:

- $\mathsf{pw}(A, S)$ et $\mathsf{pw}(B, S)$ are symmetric passwords shared between A (resp. B) and S
- $N1$, $N2$, Ca are nonces
- Ta, Ts are timestamps

Example 3 - Dictionary Attack

$$\begin{array}{lllll} \alpha.0. & A & \rightarrow & S: & \{A, B, N1, N2, Ca, \{Ta\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\ \alpha.1. & S & \rightarrow & A: & \{N1, K \oplus N2\}_{\mathsf{pw}(A,S)}, \{A, K, Ts\}_{\mathsf{pw}(B,S)} \end{array}$$

Example 3 - Dictionary Attack

$$
\begin{array}{llcll}
\alpha.0. & A & \rightarrow & S: & \{A, B, N1, N2, Ca, \{Ta\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\
\alpha.1. & S & \rightarrow & A: & \{N1, K \oplus N2\}_{\mathsf{pw}(A,S)}, \{A, K, Ts\}_{\mathsf{pw}(B,S)} \\
\\
\beta.0. & I(A) & \rightarrow & S: & \{A, B, N1, N2, Ca, \{Ta\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\
\beta.1. & S & \rightarrow & I(A): & \{N1, K' \oplus N2\}_{\mathsf{pw}(A,S)}, \{A, K', Ts'\}_{\mathsf{pw}(B,S)}
\end{array}
$$

Example 3 - Dictionary Attack

$$\begin{array}{llcll}
\alpha.0. & A & \rightarrow & S: & \{A, B, N1, N2, Ca, \{Ta\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\
\alpha.1. & S & \rightarrow & A: & \{N1, K \oplus N2\}_{\mathsf{pw}(A,S)}, \{A, K, Ts\}_{\mathsf{pw}(B,S)} \\
 & & & & \\
\beta.0. & I(A) & \rightarrow & S: & \{A, B, N1, N2, Ca, \{Ta\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\
\beta.1. & S & \rightarrow & I(A): & \{N1, K' \oplus N2\}_{\mathsf{pw}(A,S)}, \{A, K', Ts'\}_{\mathsf{pw}(B,S)}
\end{array}$$

I guesses $\mathsf{pw}(A, S)$

Example 3 - Dictionary Attack

$$
\begin{array}{llcll}
\alpha.0. & A & \to & S: & \{A, B, N1, N2, Ca, \{Ta\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\
\alpha.1. & S & \to & A: & \{N1, K \oplus N2\}_{\mathsf{pw}(A,S)}, \{A, K, Ts\}_{\mathsf{pw}(B,S)} \\
 & & & & \\
\beta.0. & I(A) & \to & S: & \{A, B, N1, N2, Ca, \{Ta\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\
\beta.1. & S & \to & I(A): & \{N1, K' \oplus N2\}_{\mathsf{pw}(A,S)}, \{A, K', Ts'\}_{\mathsf{pw}(B,S)}
\end{array}
$$

I guesses $\mathsf{pw}(A, S)$

$$
\begin{array}{llcll}
\gamma.0. & I(A) & \to & S: & \{A, B, M1, M2, Ci, \{Ti\}_{\mathsf{pw}(A,S)}\}_{\mathsf{pub}(S)} \\
\gamma.1. & S & \to & I(A): & \{M1, K'' \oplus M2\}_{\mathsf{pw}(A,S)}, \{A, K'', Ts''\}_{\mathsf{pw}(B,S)} \\
\gamma.2. & I(A) & \to & B: & \{A, K'', Ts''\}_{\mathsf{pw}(B,S)}
\end{array}
$$

Summary

Goal:

Capture such «dictionary attacks» in a formal framework

Summary

Goal:

Capture such «dictionary attacks» in a formal framework

Related Work:

- In [Gong et al, 93][Tsudik&Van Herreweghen, 93]:
  $\hookrightarrow$ examples of «guessing attacks»
  $\hookrightarrow$ countermeasures to design protocols resistant to this kind of attacks
- In [Lowe, 02][Corin et al, 03]:
  $\hookrightarrow$ definitions of «guessing attacks»
  $\hookrightarrow$ implemented procedure to find automatically such attacks

Summary

Goal:

Capture such «dictionary attacks» in a formal framework

Related Work:

- In [Gong et al, 93][Tsudik&Van Herreweghen, 93]:
  $\hookrightarrow$ examples of «guessing attacks»
  $\hookrightarrow$ countermeasures to design protocols resistant to this kind of attacks
- In [Lowe, 02][Corin et al, 03]:
  $\hookrightarrow$ definitions of «guessing attacks»
  $\hookrightarrow$ implemented procedure to find automatically such attacks

Results:

- An inference system extending the classical Dolev-Yao model
- Decidability and Complexity results in the extended model

Dolev-Yao' Model

- A set $\mathcal{G}$ of guessable data which take their value in a finite dictionary $\mathcal{D}$ known by the intruder.
  $vote \in \{yes, no\}$, $\mathsf{pw}(a, b) \in \{p_1, \ldots, p_n\}$

Dolev-Yao' Model

- A set $\mathcal{G}$ of guessable data which take their value in a finite dictionary $\mathcal{D}$ known by the intruder.
  $vote \in \{yes, no\}$, $\mathsf{pw}(a, b) \in \{p_1, \ldots, p_n\}$

- A new form of sequent

$$T,\ T' \vdash'\ v$$

Intruder's knowledge

Guessable values

Built term

Dolev-Yao' Model

- A set $\mathcal{G}$ of guessable data which take their value in a finite dictionary $\mathcal{D}$ known by the intruder.
  $vote \in \{yes, no\}$, $\mathsf{pw}(a, b) \in \{p_1, \ldots, p_n\}$

- A new form of sequent

$$T,\ T' \vdash'\ v$$

Intruder's knowledge

Guessable values

Built term

- A new inference system called DY'-Model

Guessing Rule (G) $\dfrac{g \in \mathcal{G}}{T/\{g\} \vdash' g}$

Dolev-Yao' Model

(A′) $$\frac{u \in T}{T/\emptyset \vdash' u}$$

(P′) $$\frac{T/T_1' \vdash' u \quad T/T_2' \vdash' v}{T/T_1' \cup T_2' \vdash' \langle u, v\rangle}$$

(UL′) $$\frac{T/T' \vdash' \langle u, v\rangle}{T/T' \vdash' u}$$

(UR′) $$\frac{T/T' \vdash' \langle u, v\rangle}{T/T' \vdash' v}$$

(E′) $$\frac{T/T_1' \vdash' u \quad T/T_2' \vdash' v}{T/T_1' \cup T_2' \vdash' \{u\}_v}$$

(D′) $$\frac{T/T_1' \vdash' \{u\}_v \quad T/T_2' \vdash' v^{-1}}{T/T_1' \cup T_2' \vdash' u}$$

Guessing (G) $$\frac{g \in \mathcal{G}}{T/\{g\} \vdash' g}$$

Compare Rule (1^{st} version)

$$
\dfrac{
P_1 \left\{ \dfrac{\dfrac{\ldots}{\vdash' u_1} \quad \ldots \quad \dfrac{\ldots}{\vdash' u_{n_1}}}{T/T_1' \vdash' v} (R_1) \right.
\qquad
P_2 \left\{ \dfrac{\dfrac{\ldots}{\vdash' v_1} \quad \ldots \quad \dfrac{\ldots}{\vdash' v_{n_2}}}{T/T_2' \vdash' v} (R_2) \right.
}{T \vdash g} (C)
$$

where:

(i). $g \in T_1' \cup T_2'$

(ii). $R1 \neq R2$ or $\{u_1, .., u_{n_1}\} \neq \{v_1, ..., v_{n_2}\}$

(iii). P_1 and P_2 are normal DY'–proofs

Compare Rule

$$P_1 \left\{ \dfrac{\dfrac{\cdots}{\vdash' u_1} \quad \ldots \quad \dfrac{\cdots}{\vdash' u_{n_1}}}{T/T_1' \vdash' u} (R_1) \right. \quad P_2 \left\{ \dfrac{\dfrac{\cdots}{\vdash' v_1} \quad \ldots \quad \dfrac{\cdots}{\vdash' v_{n_2}}}{T/T_2' \vdash' v} (R_2) \right. \over T \vdash g} (C)$$

where:

(i). $g \in T_1' \cup T_2'$

(ii). R1 $\neq$ R2 or $\{u, u_1, .., u_{n_1}\} \neq \{v, v_1, ..., v_{n_2}\}$

(iii). P_1 and P_2 are normal DY'–proofs

(iv). $R(u, v)$ where $R = Id \cup \{(k, k^{-1}) | k$ is a key$\}$

Example 1 - Vote Protocol

$A \to S : \{v\}_{\mathsf{pub}(S)}$

$T = \{\{v\}_{\mathsf{pub}(S)}, \mathsf{pub}(S)\}$

$v \in \mathcal{G}$

Example 1 - Vote Protocol

$$A \to S : \{v\}_{\mathsf{pub}(S)}$$

$$T = \{\{v\}_{\mathsf{pub}(S)}, \mathsf{pub}(S)\}$$
$$v \in \mathcal{G}$$

We have the following derivation:

$$\dfrac{\dfrac{\dfrac{v \in \mathcal{G}}{T/\{v\} \vdash v}\,(G) \qquad \dfrac{\mathsf{pub}(S) \in T}{T/\emptyset \vdash' \mathsf{pub}(S)}\,(A')}{T/\{v\} \vdash' \{v\}_{\mathsf{pub}(S)}}\,(E') \qquad \dfrac{\{v\}_{\mathsf{pub}(S)} \in T}{T/\emptyset \vdash' \{v\}_{\mathsf{pub}(S)}}\,(A')}{T \vdash v}\,(C)$$

Example 2 - Nonce Challenge

$$\begin{array}{lll} A & \rightarrow B: & \{n\}_{\mathsf{pw}(A,B)} \\ B & \rightarrow A: & \{n+1\}_{\mathsf{pw}(A,B)} \end{array}$$

$$T = \{\{n\}_{\mathsf{pw}(A,B)}, \{n+1\}_{\mathsf{pw}(A,B)}\}$$
$$\mathsf{pw}(A,B) \in \mathcal{G}$$

Example 2 - Nonce Challenge

$$\begin{aligned} A &\to B: \quad \{n\}_{\mathsf{pw}(A,B)} \\ B &\to A: \quad \{n+1\}_{\mathsf{pw}(A,B)} \end{aligned}$$

$$T = \{\{n\}_{\mathsf{pw}(A,B)}, \{n+1\}_{\mathsf{pw}(A,B)}\}$$
$$\mathsf{pw}(A,B) \in \mathcal{G}$$

We have the following derivation ($\mathsf{p} = \mathsf{pw}(A,B)$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\{n\}_{\mathsf{p}} \in T}{T/\emptyset \vdash' \{n\}_{\mathsf{p}}}(A') \qquad \dfrac{\mathsf{p} \in \mathcal{G}}{T/\{\mathsf{p}\} \vdash' \mathsf{p}}(G)}{T/\{\mathsf{p}\} \vdash' n}(D')}{T/\{\mathsf{p}\} \vdash' n+1} \qquad \dfrac{\dfrac{\{n+1\}_{\mathsf{p}} \in T}{T/\emptyset \vdash' \{n+1\}_{\mathsf{p}}}(A') \qquad \dfrac{\mathsf{p} \in \mathcal{G}}{T/\{\mathsf{p}\} \vdash' \mathsf{p}}(G)}{T/\{\mathsf{p}\} \vdash' n+1}(D')}{T \vdash \mathsf{p}}(C)$$

Results in the Extended Model (1)

Passive Attacker *– intruder deduction problem*

Given a set of messages T, a set $\mathcal{G}$ of guessable values and a message s, does there exist a proof of $T \vdash s$?
$\hookrightarrow$ PTIME–complete

Results in the Extended Model (1)

Passive Attacker – *intruder deduction problem*

Given a set of messages T, a set $\mathcal{G}$ of guessable values and a message s, does there exist a proof of $T \vdash s$?
$\hookrightarrow$ PTIME–complete

Proof (sketch)

1. Compute $\mathcal{G}' = \{g \in \mathcal{G} \mid T \vdash g\}$.

Results in the Extended Model (1)

Passive Attacker *– intruder deduction problem*

Given a set of messages T, a set $\mathcal{G}$ of guessable values and a message s, does there exist a proof of $T \vdash s$?
$\hookrightarrow$ PTIME–complete

Proof (sketch)

1. Compute $\mathcal{G}' = \{g \in \mathcal{G} \mid T \vdash g\}$.
2. Solve the problem of the existence of a proof of $T \cup \mathcal{G}' \vdash s$ in the **standard** Dolev-Yao model.

Results in the Extended Model (1)

Passive Attacker *– intruder deduction problem*

Given a set of messages T, a set $\mathcal{G}$ of guessable values and a message s, does there exist a proof of $T \vdash s$?
$\hookrightarrow$ PTIME–complete

Proof (sketch)

1. Compute $\mathcal{G}' = \{g \in \mathcal{G} \mid T \vdash g\}$.

   Given a set of messages T, a set $\mathcal{G}$ of guessable values and $g \in \mathcal{G}$:
   - Lemma 1: a guessing-proof of $T \vdash g$ contains only subterms of $T \cup \mathcal{G}$,
   - Lemma 2: the existence of a guessing-proof can be decided in polynomial time.

2. Solve the problem of the existence of a proof of $T \cup \mathcal{G}' \vdash s$ in the **standard** Dolev-Yao model.

Results in the Extended Model (2)

Active Attacker *insecurity problem for a finite number of sessions*
Given $\mathcal{P}$, s, $\mathcal{S}_0$, T_0 and $\mathcal{G}$, does there exist a running sequence of protocol rules such that:

- at the end, the intruder's knowledge is T,
- there exists a proof of $T \vdash s$ **?**

$\hookrightarrow$ NP–complete

Results in the Extended Model (2)

Active Attacker *insecurity problem for a finite number of sessions*
Given $\mathcal{P}$, s, $\mathcal{S}_0$, T_0 and $\mathcal{G}$, does there exist a running sequence of protocol rules such that:

- at the end, the intruder's knowledge is T,
- there exists a proof of $T \vdash s$?

$\hookrightarrow$ NP–complete

Proof (sketch)

1. Choose an interleaving $\mathcal{I}$ containing rules of the protocol and guesses.

Results in the Extended Model (2)

Active Attacker *insecurity problem for a finite number of sessions*
Given $\mathcal{P}$, s, $\mathcal{S}_0$, T_0 and $\mathcal{G}$, does there exist a running sequence of protocol rules such that:

- at the end, the intruder's knowledge is T,
- there exists a proof of $T \vdash s$?

$\hookrightarrow$ NP–complete

Proof (sketch)

1. Choose an interleaving $\mathcal{I}$ containing rules of the protocol and guesses.
2. Build the corresponding symbolic constraints system $\mathcal{C}$.

Results in the Extended Model (2)

Active Attacker *insecurity problem for a finite number of sessions*
Given $\mathcal{P}$, s, $\mathcal{S}_0$, T_0 and $\mathcal{G}$, does there exist a running sequence of protocol rules such that:

- at the end, the intruder's knowledge is T,
- there exists a proof of $T \vdash s$?

$\hookrightarrow$ NP–complete

Proof (sketch)

1. Choose an interleaving $\mathcal{I}$ containing rules of the protocol and guesses.
2. Build the corresponding symbolic constraints system $\mathcal{C}$.
3. Choose for each $x_i \in vars(\mathcal{C})$, a term $t_i \in st(\mathcal{C})$, and let σ be a mgu of $\mathcal{E} = \{x_1 \approx t_1, \ldots, x_m \approx t_m\}$.

Results in the Extended Model (2)

Active Attacker *insecurity problem for a finite number of sessions*
Given $\mathcal{P}$, s, $\mathcal{S}_0$, T_0 and $\mathcal{G}$, does there exist a running sequence of protocol rules such that:

- at the end, the intruder's knowledge is T,
- there exists a proof of $T \vdash s$?

$\hookrightarrow$ NP–complete

Proof (sketch)

1. Choose an interleaving $\mathcal{I}$ containing rules of the protocol and guesses.
2. Build the corresponding symbolic constraints system $\mathcal{C}$.
3. Choose for each $x_i \in vars(\mathcal{C})$, a term $t_i \in st(\mathcal{C})$, and let σ be a mgu of $\mathcal{E} = \{x_1 \approx t_1, \ldots, x_m \approx t_m\}$.
4. if σ is ground, check whether σ is a solution of $\mathcal{C}$, if so return Yes.

Example 2 – Nonce Challenge

Input

- $$\mathcal{P} = \begin{array}{lllll} 1. & A & \rightarrow B: & \{N\}_{\mathsf{pw}(A,B)} \\ 2. & B & \rightarrow A: & \{N+1\}_{\mathsf{pw}(A,B)} \end{array}$$
- $\mathcal{S}_0 = \{s\}$ where $s = (A = a, B = b, N = n)$
- $T_0 = \emptyset$
- $secret = \mathsf{pw}(a, b)$

Example 2 – Nonce Challenge

Input

- $$\mathcal{P} = \begin{array}{llll} 1. & A & \rightarrow B: & \{N\}_{\mathsf{pw}(A,B)} \\ 2. & B & \rightarrow A: & \{N+1\}_{\mathsf{pw}(A,B)} \end{array}$$
- $\mathcal{S}_0 = \{s\}$ where $s = (A = a, B = b, N = n)$
- $T_0 = \emptyset$
- $secret = \mathsf{pw}(a, b)$

Output

1. Interleaving $\mathcal{I} = [s.1; s.2\ ]$

Example 2 – Nonce Challenge

Input

- $$\mathcal{P} = \begin{array}{llll} 1. & A & \to B: & \{N\}_{\mathsf{pw}(A,B)} \\ 2. & B & \to A: & \{N+1\}_{\mathsf{pw}(A,B)} \end{array}$$
- $\mathcal{S}_0 = \{s\}$ where $s = (A = a, B = b, N = n)$
- $T_0 = \emptyset$
- $secret = \mathsf{pw}(a, b)$

Output

1. Interleaving $\mathcal{I} = [s.1; s.2]$
2. Constraints system $\mathcal{C}$

Example 2 – Nonce Challenge

Input

- $$\mathcal{P} = \begin{array}{lll} 1.\ A & \rightarrow B: & \{N\}_{\mathsf{pw}(A,B)} \\ 2.\ B & \rightarrow A: & \{N+1\}_{\mathsf{pw}(A,B)} \end{array}$$
- $\mathcal{S}_0 = \{s\}$ where $s = (A = a, B = b, N = n)$
- $T_0 = \emptyset$
- $secret = \mathsf{pw}(a, b)$

Output

1. Interleaving $\mathcal{I} = [s.1; s.2\ ]$
2. Constraints system $\mathcal{C}$

$s.1$ $\qquad T_1 = T_0, \{n\}_{\mathsf{pw}(a,b)}$

Example 2 – Nonce Challenge

Input

- $$\mathcal{P} = \begin{array}{llll} 1. & A & \to B: & \{N\}_{\mathsf{pw}(A,B)} \\ 2. & B & \to A: & \{N+1\}_{\mathsf{pw}(A,B)} \end{array}$$
- $\mathcal{S}_0 = \{s\}$ where $s = (A = a, B = b, N = n)$
- $T_0 = \emptyset$
- $secret = \mathsf{pw}(a, b)$

Output

1. Interleaving $\mathcal{I} = [s.1; s.2\ ]$
2. Constraints system $\mathcal{C}$

$$\begin{array}{lll} s.1 & & T_1 = T_0, \{n\}_{\mathsf{pw}(a,b)} \\ s.2 & T_1 \vdash_{\mathsf{dy}} \{x\}_{\mathsf{pw}(a,b)} & T_2 = T_1, \{x+1\}_{\mathsf{pw}(a,b)} \end{array}$$

Example 2 – Nonce Challenge

Input

- $$\mathcal{P} = \begin{array}{llll} 1. & A & \rightarrow B: & \{N\}_{\mathsf{pw}(A,B)} \\ 2. & B & \rightarrow A: & \{N+1\}_{\mathsf{pw}(A,B)} \end{array}$$
- $\mathcal{S}_0 = \{s\}$ where $s = (A = a, B = b, N = n)$
- $T_0 = \emptyset$
- $secret = \mathsf{pw}(a, b)$

Output

1. Interleaving $\mathcal{I} = [s.1; s.2\ ]$
2. Constraints system $\mathcal{C}$

$$\begin{array}{lll} s.1 & & T_1 = T_0, \{n\}_{\mathsf{pw}(a,b)} \\ s.2 & T_1 \vdash_{\mathsf{dy}} \{x\}_{\mathsf{pw}(a,b)} & T_2 = T_1, \{x+1\}_{\mathsf{pw}(a,b)} \\ secret & T_2 \vdash_{\mathsf{g}} \mathsf{pw}(a,b) & \end{array}$$

Example 2 – Nonce Challenge

Input

- $$\mathcal{P} = \begin{array}{lll} 1.\ A & \rightarrow B: & \{N\}_{\mathsf{pw}(A,B)} \\ 2.\ B & \rightarrow A: & \{N+1\}_{\mathsf{pw}(A,B)} \end{array}$$
- $\mathcal{S}_0 = \{s\}$ where $s = (A = a, B = b, N = n)$
- $T_0 = \emptyset$
- $secret = \mathsf{pw}(a,b)$

Output

1. Interleaving $\mathcal{I} = [s.1;\ s.2\ ]$
2. Constraints system $\mathcal{C}$

$$\begin{array}{lll} s.1 & & T_1 = T_0, \{n\}_{\mathsf{pw}(a,b)} \\ s.2 & T_1 \vdash_{\mathsf{dy}} \{x\}_{\mathsf{pw}(a,b)} & T_2 = T_1, \{x+1\}_{\mathsf{pw}(a,b)} \\ secret & T_2 \vdash_{\mathsf{g}} \mathsf{pw}(a,b) & \end{array}$$

3. $\mathcal{E} = \{x \approx n\}$
4. $\sigma = \{x \mapsto n\}$ is a solution of $\mathcal{C}$.

Algorithm – Completeness

Key Lemma:
Let σ be a minimal solution of constraints system $\mathcal{C}$. For all $x \in vars(\mathcal{C})$, there exists $t \in st(\mathcal{C}) \setminus vars(\mathcal{C})$ such that $t\sigma = x\sigma$.

Algorithm – Completeness

Key Lemma:
Let σ be a minimal solution of constraints system $\mathcal{C}$. For all $x \in vars(\mathcal{C})$, there exists $t \in st(\mathcal{C}) \setminus vars(\mathcal{C})$ such that $t\sigma = x\sigma$.

Proof (sketch) – *by contradiction* –
Assume that $\exists x$ such that $\forall t$ with $t\sigma = x\sigma$, we have $t \notin st(C)$

Let be $\delta = \{x\sigma \mapsto 0\}$.

We show for each constraint $T_i \vdash_{\mathsf{x}} v_i \in \mathcal{C}$ that $\sigma' = \sigma\delta$ is also a solution, *i.e.*:
- if $\mathsf{x} = \mathsf{dy}$ then $\exists$ a proof of $T_i\sigma' \vdash v_i\sigma'$ (in DY)
- if $\mathsf{x} = \mathsf{g}$ then $\exists$ a proof of $T_i\sigma' \vdash v_i\sigma'$ (in DY'+ Compare)

Hence σ' is a solution of $\mathcal{C}$. (contradiction)

Conclusion & Future Work

Conclusion

- An inference system capturing «dictionary attacks» in a formal framework

As in the Dolev-Yao model

- The intruder deduction problem is PTIME–complete
- The protocol insecurity problem for a bounded number of sessions is NP–complete

Conclusion & Future Work

Conclusion

- An inference system capturing «dictionary attacks» in a formal framework

As in the Dolev-Yao model

- The intruder deduction problem is PTIME–complete
- The protocol insecurity problem for a bounded number of sessions is NP–complete

Future Work

Other extensions of the DY–model:

- on–line guessing attacks
- chosen plaintext attacks