# 可证明安全 - 1. 前言

钱宸

网络空间安全学院
山东大学

2025/10/13

# Contents

# 背景

# 形式化证明与密码学

# 密码学的形式化标记

# 形式化标记

- A, B: 用户 A, B

# 形式化标记

- A, B: 用户 A, B
- A $\rightarrow$ B: 用户 A 向用户 B 发送消息

# 形式化标记

- A, B: 用户 A, B
- A $\rightarrow$ B: 用户 A 向用户 B 发送消息
- $\mathsf{pk}(A), \mathsf{sk}(A)$: 用户 A 的公钥和私钥

# 形式化标记

- A, B: 用户 A, B
- $\mathsf{A} \rightarrow \mathsf{B}$: 用户 A 向用户 B 发送消息
- $\mathsf{pk}(\mathsf{A}), \mathsf{sk}(\mathsf{A})$: 用户 A 的公钥和私钥
- $[\![\mathsf{pk}(\mathsf{A})]\!]_x^{\mathsf{a}}$: 对消息 $x$ 使用用户 A 的公钥进行加密

# 形式化标记

- A, B: 用户 A, B
- $\mathsf{A} \to \mathsf{B}$: 用户 A 向用户 B 发送消息
- $\mathsf{pk}(\mathsf{A}), \mathsf{sk}(\mathsf{A})$: 用户 A 的公钥和私钥
- $[\![\mathsf{pk}(\mathsf{A})]\!]_x^{\mathsf{a}}$: 对消息 $x$ 使用用户 A 的公钥进行加密
- $\langle x, y \rangle$: 有序对 $(x, y)$, 表示元素之间的简单连接

# Needham Schnroeder 密钥交换

# 密钥交换 (简略) [Needham and Schroeder, 1978]

$$\begin{array}{lll} 1. & \mathsf{A} \to \mathsf{B}: & [\![\langle \mathsf{A}, \mathsf{N_A} \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(B)}} \\ 2. & \mathsf{B} \to \mathsf{A}: & [\![\langle \mathsf{B}, \mathsf{N_B} \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(A)}} \\ 3. & \mathsf{A} \to \mathsf{B}: & [\![N_b]\!]^{\mathsf{a}}_{\mathsf{pk(B)}} \end{array}$$

**协议执行**

1. 用户 A 向用户 B 发送加密消息 $[\![\langle \mathsf{A}, \mathsf{N_A} \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(B)}}$, 其中 $\mathsf{N_A}$ 是用户 A 生成的随机数.

# 密钥交换 (简略) [Needham and Schroeder, 1978]

$$
\begin{array}{llll}
1. & \mathsf{A} \to \mathsf{B}: & [\![\langle \mathsf{A}, \mathsf{N_A}\rangle]\!]^{\mathsf{a}}_{\mathsf{pk(B)}} \\
2. & \mathsf{B} \to \mathsf{A}: & [\![\langle \mathsf{B}, \mathsf{N_B}\rangle]\!]^{\mathsf{a}}_{\mathsf{pk(A)}} \\
3. & \mathsf{A} \to \mathsf{B}: & [\![N_b]\!]^{\mathsf{a}}_{\mathsf{pk(B)}}
\end{array}
$$

**协议执行**

1. 用户 A 向用户 B 发送加密消息 $[\![\langle \mathsf{A}, \mathsf{N_A}\rangle]\!]^{\mathsf{a}}_{\mathsf{pk(B)}}$, 其中 $\mathsf{N_A}$ 是用户 A 生成的随机数.
2. 用户 B 向用户 A 发送加密消息 $[\![\langle \mathsf{N_A}, \mathsf{N_B}\rangle]\!]^{\mathsf{a}}_{\mathsf{pk(A)}}$, 其中 $\mathsf{N_B}$ 是用户 B 生成的随机数.

# 密钥交换 (简略) [Needham and Schroeder, 1978]

$$
\begin{array}{lll}
1. & \mathsf{A} \to \mathsf{B}: & [\![\langle \mathsf{A}, \mathsf{N_A} \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(B)}} \\
2. & \mathsf{B} \to \mathsf{A}: & [\![\langle \mathsf{B}, \mathsf{N_B} \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(A)}} \\
3. & \mathsf{A} \to \mathsf{B}: & [\![N_b]\!]^{\mathsf{a}}_{\mathsf{pk(B)}}
\end{array}
$$

**协议执行**

1. 用户 A 向用户 B 发送加密消息 $[\![\langle \mathsf{A}, \mathsf{N_A} \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(B)}}$, 其中 $\mathsf{N_A}$ 是用户 A 生成的随机数.
2. 用户 B 向用户 A 发送加密消息 $[\![\langle \mathsf{N_A}, \mathsf{N_B} \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(A)}}$, 其中 $\mathsf{N_B}$ 是用户 B 生成的随机数.
3. 用户 A 向用户 B 发送加密的消息 $[\![\mathsf{N_B}]\!]^{\mathsf{a}}_{\mathsf{pk(B)}}$, 以证明自己是通信的发起者.

# 更严格的交互说明

$$
\begin{array}{lccr}
\mathsf{A} & \xrightarrow{\llbracket\langle \mathsf{A},\mathsf{N_A}\rangle\rrbracket^{\mathsf{a}}_{\mathsf{pk(B)}}} & \xrightarrow{\llbracket\langle x,y\rangle\rrbracket^{\mathsf{a}}_{\mathsf{pk(B)}}} & \mathsf{B} \\
\mathsf{A} & \xleftarrow{\llbracket\langle \mathsf{N_A},z\rangle\rrbracket^{\mathsf{a}}_{\mathsf{pk(A)}}} & \xleftarrow{\llbracket\langle y,\mathsf{N_B}\rangle\rrbracket^{\mathsf{a}}_{\mathsf{pk}(x)}} & \mathsf{B} \\
\mathsf{A} & \xrightarrow{\llbracket z\rrbracket^{\mathsf{a}}_{\mathsf{pk(B)}}} & \xrightarrow{\llbracket \mathsf{N_B}\rrbracket^{\mathsf{a}}_{\mathsf{pk(B)}}} & \mathsf{B}
\end{array}
$$

## 并行运行与攻击

- 协议实际运行在并行环境中，可能会有多个实例同时进行.
- 用变量 $x, y, z$ 来指代接收到，但是无法验证的消息.
- 在并行环境中，是否存在攻击?

# Lowe's 中间人攻击 [Lowe, 1998]

$$A \xrightarrow{[\![\langle A, N_A\rangle]\!]^a_{pk(C)}} C$$

$$C(A) \xrightarrow{[\![\langle A, N_A\rangle]\!]^a_{pk(B)}} B$$

$$C(A) \xleftarrow{[\![\langle N_A, N_B\rangle]\!]^a_{pk(A)}} B$$

$$A \xleftarrow{[\![\langle N_A, N_B\rangle]\!]^a_{pk(A)}} C$$

$$A \xrightarrow{[\![N_B]\!]^a_{pk(C)}} C$$

$$C(A) \xrightarrow{[\![N_B]\!]^a_{pk(B)}} B$$

**Lowe's 中间人攻击**

- C 对 A 扮演 C 自己, 而对 B 扮演 A.
- B 以为自己在与 A 通信, 实际上却是在与 C 通信.

# 一些背景

**[Needham and Schroeder, 1978]**

该论文实际提出的时候要求: 任意参与者都诚实执行协议. 安全性保障针对诚实但好奇的敌手.

- 因此 [Lowe, 1998] 的攻击严格意义上并不是一个针对 NS 协议的攻击.
- 然而, NS 协议中的假设在如今的环境中被普遍认为是不成立的.

# Lowe 的修复

- Lowe 通过在第二个消息中加入 B 的身份消息, 修复了 NS 协议.
- 第二轮消息从 $[\![\langle \mathsf{N_A}, \mathsf{N_B} \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(A)}}$ 变更为 $[\![\langle \mathsf{N_A}, \langle \mathsf{N_B}, \mathsf{B} \rangle \rangle]\!]^{\mathsf{a}}_{\mathsf{pk(A)}}$.
- 往后我们称修复后的协议为 NSL 协议.

# 消息与演绎

# 项

## 定义

- $\mathcal{F}$: 函数集合
  - 密码学算法用函数来表示, 包括签名、加密等
  - **函数签名:** 每个函数都有一个名称和参数类型的列表, 称为函数的**函数签名** (signature)
  - **元数:** 每个函数都有一个固定的参数个数, 称为函数的**元数** (arity)
- $\mathcal{X}$: 变量集合
- $\mathcal{N}$: 原子消息 (常量) 集合
- 项的集合: $\mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$,

# 项

## 定义

- $\mathcal{F}$: 函数集合
  - 密码学算法用函数来表示, 包括签名、加密等
  - **函数签名:** 每个函数都有一个名称和参数类型的列表, 称为函数的**函数签名** (signature)
  - **元数:** 每个函数都有一个固定的参数个数, 称为函数的**元数** (arity)
- $\mathcal{X}$: 变量集合
- $\mathcal{N}$: 原子消息 (常量) 集合
- 项的集合: $\mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$,

## 项 $t$ 中的变量与原子消息

- var$(t)$: 变量通常是外部未定义元素, 例如协议中的未知消息.
- n$(t)$: 原子消息通常是已知的基本元素, 例如密钥, 用户标识符等.

# 定义样例

在安全协议里面，我们首先定义一个标准函数集 $\mathcal{F}_{\mathsf{std}} = \{\mathsf{senc}, \mathsf{aenc}, \langle\cdot,\cdot\rangle, \mathsf{pk}\}$

- senc：对称加密函数，$\mathsf{senc}(x,k)$ 表示使用密钥 $k$ 对消息 $x$ 进行对称加密，记作 $[\![x]\!]_k^{\mathsf{s}}$
- aenc：认证加密函数，$\mathsf{aenc}(x,k)$ 表示使用密钥 $k$ 对消息 $x$ 进行非对称加密，记作 $[\![x]\!]_k^{\mathsf{a}}$
- $\langle\cdot,\cdot\rangle$：有序对构造函数，记作 $\langle x,y\rangle$
- pk：公钥函数，$\mathsf{pk}(k)$ 表示私钥 $k$ 对应的公钥，记作 $\mathsf{pk}\,(k)$

**NSL 中的消息项**

$t_0 = \mathsf{aenc}(\langle a, n_a\rangle, \mathsf{pk}\,(k_a)) = [\![\mathsf{pk}(k_a)]\!]^{\mathsf{a}}_{\langle a,n_a\rangle}$ 其中 $a, n_a, k_a \in \mathcal{N}$.

# 项位置

**定义**

- 项 $t$ 中的位置集合 $\mathsf{Pos}(t)$, 位置 $p$ 是一个字符串, 例如 $\epsilon, 1, 1.1, 1.2, 2, 2.1, \ldots$.

$$\mathsf{Pos}(f(t_1, t_2, \ldots, t_n)) = \{\epsilon\} \cup \bigcup_{i=1}^{n} \{i.p \mid p \in \mathsf{Pos}(t_i)\}$$

- $t|_p$: 项 $t$ 中位置 $p$ 处的子项, $t|_\epsilon = t$
- $t[s]_p$: 用项 $s$ 替换项 $t$ 中位置 $p$ 处的子项
- $\mathsf{st}(t)$: 项 $t$ 的子项集合

$$\mathsf{st}(f(t_1, t_2, \ldots, t_n)) = \{f(t_1, t_2, \ldots, t_n)\} \cup \bigcup_{i=1}^{n} \mathsf{st}(t_i)$$

# NSL 中的消息项

## NSL 中的消息项

$t_0 = [\![\mathsf{pk}(k_a)]\!]^{\mathsf{a}}_{\langle a, n_a \rangle} = \mathsf{aenc}(\langle a, n_a \rangle, \mathsf{pk}\left(k_a\right))$, 则

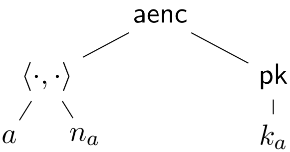
- $\mathsf{Pos}(t_0) = \{\epsilon, 1, 1.1, 1.2, 2, 2.1\}$
- $\mathsf{st}(t_0) = \{t_0, \langle a, n_a \rangle, a, n_a, \mathsf{pk}\left(k_a\right), k_a\}$

# NSL 中的消息项

## NSL 中的消息项

$t_0 = [\![\mathsf{pk}(k_a)]\!]^{\mathsf{a}}_{\langle a, n_a\rangle} = \mathsf{aenc}(\langle a, n_a\rangle, \mathsf{pk}\,(k_a))$, 则

- $\mathsf{Pos}(t_0) = \{\epsilon, 1, 1.1, 1.2, 2, 2.1\}$
- $\mathsf{st}(t_0) = \{t_0, \langle a, n_a\rangle, a, n_a, \mathsf{pk}\,(k_a)\,, k_a\}$




- $t_0|_{1.2} = n_a$
- $t_0|_2 = \mathsf{pk}\,(k_a)$
- $t_0[\mathsf{pk}\,(k_b)]_2 = [\![\mathsf{pk}(k_b)]\!]^{\mathsf{a}}_{\langle a, n_a\rangle}$

# 置换

**定义**

- 置换 $\sigma$: $\mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$
- $\sigma$ 作用在项 $t$ 上, 记作 $t\sigma$, 定义如下:

$$
\begin{aligned}
x\sigma = \sigma(x) &= x && \text{if } x \notin \mathsf{Dom} \\
f(t_1, t_2, \ldots, t_n)\sigma &= f(t_1\sigma, t_2\sigma, \ldots, t_n\sigma)
\end{aligned}
$$

# 置换

**定义**

- 置换 $\sigma$: $\mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$
- $\sigma$ 作用在项 $t$ 上, 记作 $t\sigma$, 定义如下:

$$
\begin{aligned}
x\sigma = \sigma(x) &= x & \text{if } x \notin \mathrm{Dom} \\
f(t_1, t_2, \ldots, t_n)\sigma &= f(t_1\sigma, t_2\sigma, \ldots, t_n\sigma)
\end{aligned}
$$

**归一化:** 如果两个项能够通过置换变得相同, 则称这两个项是**归一化**的. $u, v$ 是可归一的, 如果存在置换 $\sigma$, 使得 $u\sigma = v\sigma$.

# 置换

## 定义

- 置换 $\sigma$: $\mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$
- $\sigma$ 作用在项 $t$ 上, 记作 $t\sigma$, 定义如下:

$$x\sigma = \sigma(x) = x \qquad \text{if } x \notin \mathsf{Dom}$$

$$f(t_1, t_2, \ldots, t_n)\sigma = f(t_1\sigma, t_2\sigma, \ldots, t_n\sigma)$$

**归一化:** 如果两个项能够通过置换变得相同, 则称这两个项是**归一化**的. $u, v$ 是可归一的, 如果存在置换 $\sigma$, 使得 $u\sigma = v\sigma$.

## 定理 (一般归一化置换 [Baader and Snyder, 2001])

如果两个项 $u, v$ 是可归一的, 则存在一个最一般的归一化置换 $\sigma = \mathsf{mgu}(u, v)$, 使得对于任意使得 $u\sigma' = v\sigma'$ 的置换 $\sigma'$, 都存在一个置换 $\theta$, 使得 $\sigma' = \sigma\theta$.

# 消息演绎

- 在密码学协议和算法中，需要满足一些特殊的性质
- 例如：对于密文的解密等于原文
- 这些性质可以通过一些简单的规则来描述

# 推理系统

## 解密

根据密钥 $k$ 与消息 $\mathsf{senc}(m,k)$ 表示消息 $m$ 使用密钥 $k$ 加密, 则有如下推理规则:

$$\frac{\mathsf{senc}(m,k) \quad k}{m}$$

# 推理系统

**解密**

根据密钥 $k$ 与消息 $\mathsf{senc}(m, k)$ 表示消息 $m$ 使用密钥 $k$ 加密, 则有如下推理规则:

$$\frac{\mathsf{senc}(m, k) \quad k}{m}$$

**定义 (推理规则)**

推理规则 $r$ 是一个分数形式的表达式, 形如

$$\frac{u_1 \quad u_2 \quad \ldots \quad u_n}{u}$$

其中 $u_1, u_2, \ldots, u_n$ 是前提, $u$ 是结论. 如果 $n = 0$, 则称 $r$ 为公理.

**推理系统**则表示为一组推理规则的集合.

# Dolev-Yao 模型 [Dolev and Yao, 1981]

- Dolev-Yao 模型是密码学中最著名的推理系统之一.
- 该模型假设密码学算法是完美的, 即没有任何漏洞.
- 该模型定义了一组推理规则, 用于描述攻击者可以执行的操作.

# Dolev-Yao 模型 [Dolev and Yao, 1981]

- Dolev-Yao 模型是密码学中最著名的推理系统之一.
- 该模型假设密码学算法是完美的, 即没有任何漏洞.
- 该模型定义了一组推理规则, 用于描述攻击者可以执行的操作.

**Dolev-Yao 模型中的推理规则 $\mathcal{I}_{\mathrm{DY}}$**

$$\frac{\langle x, y\rangle}{x} \qquad \frac{\langle x, y\rangle}{y} \qquad \frac{x \quad y}{\langle x, y\rangle}$$

$$\frac{\mathsf{senc}(x, k) \quad k}{x} \qquad \frac{\mathsf{aenc}(x, \mathsf{pk}(k)) \quad k}{x}$$

$$\frac{x \quad k}{\mathsf{senc}(x, k)} \qquad \frac{x \quad \mathsf{pk}(k)}{\mathsf{aenc}(x, \mathsf{pk}(k))}$$

# 可推导性

**定义 (可推导性)**

给定一个推理系统 $\mathcal{I}$, 和一个项集合 $S \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$, 如果存在一个推理规则 $r \in \mathcal{I}$, 使得

- 存在一个置换 $\sigma$, 使得 $r$ 的前提 $u_1, u_2, \ldots, u_n$ 和结论 $t$ 满足 $t_i = u_i\sigma \in S$ (对于所有 $1 \leq i \leq n$) 和 $t = u\sigma$

则称 $t$ 是从 $S$ 中**可一步推导**的, 记作 $S \vdash_{\mathcal{I}}^{1} t$.

# 可推导性

**定义 (可推导性)**

给定一个推理系统 $\mathcal{I}$, 和一个项集合 $S \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$, 如果存在一个推理规则 $r \in \mathcal{I}$, 使得

- 存在一个置换 $\sigma$, 使得 $r$ 的前提 $u_1, u_2, \ldots, u_n$ 和结论 $t$ 满足 $t_i = u_i\sigma \in S$ (对于所有 $1 \leq i \leq n$) 和 $t = u\sigma$

则称 $t$ 是从 $S$ 中**可一步推导**的, 记作 $S \vdash^1_{\mathcal{I}} t$.

**Dolev-Yao 模型中的可推导性**

设 $S = \{\mathsf{senc}(a, k), k\}$, 则 $S \vdash^1_{\mathcal{I}_{\mathsf{DY}}} a$, 因为存在推理规则 $\dfrac{\mathsf{senc}(x, k) \quad k}{x}$.

# 可推导性与证明树

**定义 (证明树)**

一个项 $t$ 是从一个项集合 $S$ 中可推导的, 记作 $S \vdash_{\mathcal{I}} t$, 如果存在一棵证明树 $\mathbf{\Pi}$, 满足:

- 叶子节点是 $S$ 中的项
- 如果一个非叶子节点 $t'$ 的子节点是 $t_1, t_2, \ldots, t_n$, 则存在一个推理规则 $r \in \mathcal{I}$, 和一个置换 $\sigma$, 使得 $r$ 的前提 $u_1, u_2, \ldots, u_n$ 和结论 $u$ 满足 $t_i = u_i\sigma$ (对于所有 $1 \leq i \leq n$) 和 $t' = u\sigma$. 即 $\dfrac{u_1 \quad u_2 \quad \ldots \quad u_n}{u}$
- 根节点是 $t$

# 可推导性与证明树

**定义 (证明树)**

一个项 $t$ 是从一个项集合 $S$ 中可推导的, 记作 $S \vdash_{\mathcal{I}} t$, 如果存在一棵证明树 $\mathbf{\Pi}$, 满足:

- 叶子节点是 $S$ 中的项
- 如果一个非叶子节点 $t'$ 的子节点是 $t_1, t_2, \ldots, t_n$, 则存在一个推理规则 $r \in \mathcal{I}$, 和一个置换 $\sigma$, 使得 $r$ 的前提 $u_1, u_2, \ldots, u_n$ 和结论 $u$ 满足 $t_i = u_i\sigma$ (对于所有 $1 \leq i \leq n$) 和 $t' = u\sigma$. 即 $\dfrac{u_1 \quad u_2 \quad \cdots \quad u_n}{u}$
- 根节点是 $t$

**Dolev-Yao 模型中的可推导性**

设 $S_0 = \left\{\langle k_1, k_2\rangle, \langle k_3, a\rangle, [\![n]\!]^{\mathsf{s}}_{\langle k_1, k_3\rangle}\right\}$, 则 $S_0 \vdash_{\mathcal{I}_{\mathsf{DY}}} n$

# 入侵者推理分析

- 在安全协议中, 我们通常假设入侵者可以完全控制网络.
- 入侵者可以拦截、修改、重放消息, 甚至伪造消息.
- 因此, 我们需要分析入侵者在给定初始知识的情况下, 能够推导出哪些消息.

# 入侵者推理分析

- 在安全协议中，我们通常假设入侵者可以完全控制网络.
- 入侵者可以拦截、修改、重放消息，甚至伪造消息.
- 因此，我们需要分析入侵者在给定初始知识的情况下，能够推导出哪些消息.

**定义 (入侵者推理分析)**

给定一个推理系统 $\mathcal{I}$，和一个项集合 $S \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$，入侵者推理分析问题是确定对于一个项 $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{N})$，是否有 $S \vdash_{\mathcal{I}} t$.

- 入侵者推理分析是安全协议分析中的一个重要问题.
- 然而对于一般的推理系统，入侵者推理分析问题是**不可判定的**. [Abadi and Cortier, 2006]
- 如果对于**局部理论**，即如果 $S \vdash t$，则存在一个只用 $S$ 和 $t$ 中子项的证明树，则入侵者推理分析问题是**可判定的**.

# 局部理论

**定义 (局部理论)**

一个推理系统 $\mathcal{I}$ 是**局部的**, 如果对于任意项集合 $S$ 和项 $t$, 如果 $S \vdash_{\mathcal{I}} t$, 则存在一个证明树 $\mathbf{\Pi}$, 使得 $\mathsf{st}(\mathbf{\Pi}) \subseteq \mathsf{st}(S \cup \{t\})$.

# 局部理论

**定义 (局部理论)**

一个推理系统 $\mathcal{I}$ 是**局部的**, 如果对于任意项集合 $S$ 和项 $t$, 如果 $S \vdash_{\mathcal{I}} t$, 则存在一个证明树 $\mathbf{\Pi}$, 使得 $\mathsf{st}(\mathbf{\Pi}) \subseteq \mathsf{st}(S \cup \{t\})$.

**定理**

令 $\mathcal{I}$ 为一个局部推理系统, 则入侵者推理分析问题是在多项式时间内 (PTIME) 可判定的.

# 局部理论

**定义 (局部理论)**

一个推理系统 $\mathcal{I}$ 是**局部的**, 如果对于任意项集合 $S$ 和项 $t$, 如果 $S \vdash_{\mathcal{I}} t$, 则存在一个证明树 $\mathbf{\Pi}$, 使得 $\mathsf{st}(\mathbf{\Pi}) \subseteq \mathsf{st}(S \cup \{t\})$.

**定理**

令 $\mathcal{I}$ 为一个局部推理系统, 则入侵者推理分析问题是在多项式时间内 (PTIME) 可判定的.

我们可以简单给出这个这个算法

- 初始化 $S_0 = S$
- 重复以下步骤, 直到 $S_i$ 不再变化:
  - $S_{i+1} = S_i \cup \left(\{u \mid S_i \vdash^1 u\} \cap \mathsf{st}(S \cup \{t\})\right)$

# Dolev-Yao 模型的局部性

**定理 ([Abadi and Cortier, 2006])**

*Dolev-Yao* 模型 $\mathcal{I}_{DY}$ 是局部的.

# Dolev-Yao 模型的局部性

**定理 ([Abadi and Cortier, 2006])**

*Dolev-Yao 模型 $\mathcal{I}_{DY}$ 是局部的.*

- 考虑最小证明树
- 分别考虑组合和解构两种不同的推理规则

# 练习

**练习**

设 $S = \left\{ [\![k_2]\!]^{\mathsf{s}}_{\langle k_1, [\![k_1]\!]^{\mathsf{s}}_{k_3} \rangle}, \langle k_1, k_1 \rangle, [\![ [\![k_1]\!]^{\mathsf{s}}_{k_3} ]\!]^{\mathsf{s}}_{k_1} \right\}$, 其中 $k_1, k_2, k_3 \in \mathcal{N}$.

1. 证明 $S \vdash_{\mathcal{I}_{\mathsf{DY}}} k_1$, $S \vdash_{\mathcal{I}_{\mathsf{DY}}} k_1$.
2. 证明 $S \nvdash_{\mathcal{I}_{\mathsf{DY}}} k_3$.

# References I


Abadi, M. and Cortier, V. (2006).
Deciding knowledge in security protocols under equational theories.
*Theoretical Computer Science*, 367(1-2):2–32.

Baader, F. and Snyder, W. (2001).
Unification theory.
*Handbook of automated reasoning*, 1:445–532.

Dolev, D. and Yao, A. C.-C. (1981).
On the security of public key protocols (extended abstract).
pages 350–357.

Lowe, G. (1998).
Casper: A compiler for the analysis of security protocols.
*J. Comput. Secur.*, 6(1-2):53–84.

# References II

Needham, R. M. and Schroeder, M. D. (1978).
Using encryption for authentication in large networks of computers.
*Commun. ACM*, 21(12):993–999.