可证明安全 - 3. 程序建模

钱宸
网络空间安全学院
山东大学

2025/10/15

Contents

引言与动机

引言与动机

- 在前面的章节中, 我们介绍了密码学中的一些基本概念, 包括密码原语、协议以及等式理论和静态等价.
- 然而, 要形式化地分析密码协议的安全性, 我们需要一个精确的程序模型来描述协议的执行过程和攻击者的能力.
- 本章将介绍用于建模密码协议的程序模型, 主要包括进程代数.

语法

语法与语义

- 程序模型通常使用进程代数来描述协议的行为. 进程代数提供了一种形式化的方式来表示并发系统中的进程及其交互.
- π-演算允许我们描述进程之间的通信和同步 [Milner, 1992](图灵奖'91).
- 此外, 我们还将讨论程序模型的语义, 包括操作语义和标记语义, 以便理解进程的执行过程.

进程模拟

定义 (进程语法)

进程的语法定义如下:

$$P, Q ::= 0 \mid P \| Q \mid !P \mid \nu n.P \mid \textbf{if } t_1 = t_2 \textbf{ then } P \textbf{else } \ Q \mid \textbf{in } (u, x).P \mid \textbf{out } (u, t).P$$

其中, 0 表示空进程, $P \| Q$ 表示并发执行的进程, $!P$ 表示无限复制的进程, $\nu n.P$ 表示名称 n 的限缩, $\textbf{if } t_1 = t_2 \textbf{ then } P \textbf{else } \ Q$ 表示条件分支, $\textbf{in } (u, x).P$ 表示从通道 u 接收消息并绑定到变量 x, $\textbf{out } (u, t).P$ 表示向通道 u 发送消息 t.

扩展进程

定义 (扩展进程)

$$A, B, C ::= P \mid A \| B \mid \nu n.A \mid \nu x.A \mid \{t/x\}$$

扩展进程包括普通进程 P, 并发组合 $A \| B$, 名称限缩 $\nu n.A$, 变量限缩 $\nu x.A$, 以及替换 $\{t/x\}$. 其中 **let** $x = t$ **in** P 表示将表达式 t 的值绑定到变量 x, 是 $\nu x.(P \| \{t/x\})$ 的缩写.

将协议建模成为进程

- 我们可以将密码协议中的各个角色表示为进程, 并使用进程代数的操作来描述它们之间的交互.
- 例如, 发送消息可以表示为输出操作, 接收消息可以表示为输入操作, 条件判断可以表示为条件分支等.
- 通过这种方式, 我们可以精确地描述协议的执行过程, 并分析其安全性.

NSL 协议的建模

- 以 Needham-Schroeder 公钥协议为例, 我们可以将其建模为以下进程:

$$
\begin{aligned}
P_A(\mathsf{sk}_i, \mathsf{pk}_r) := \nu n_a.\mathbf{out}\ (c, \mathsf{aenc}(\langle \mathsf{pk}(\mathsf{sk}_i), n_a\rangle, \mathsf{pk}_r)).\mathbf{in}\ (c, x).\\
&\mathbf{if}\ \mathsf{fst}(\mathsf{adec}(x, \mathsf{sk}_i)) = n_a\ \ \mathbf{then}\\
&\mathbf{let}\ \ x_{nb} = \mathsf{snd}(\mathsf{adec}(x, \mathsf{sk}_i))\ \mathbf{in}\\
&\mathbf{out}\ (c, \mathsf{aenc}(x_{nb}, pk_r)).0\\
P_B(\mathsf{sk}_r) := \mathbf{in}\ (c, y).\\
&\mathbf{let}\ \ \mathsf{pk}_i = \mathsf{fst}(\mathsf{adec}(y, n_B))\mathbf{in}\\
&\mathbf{let}\ \ y_{na} = \mathsf{snd}(\mathsf{adec}(y, \mathsf{sk}_r))\mathbf{in}\\
&\nu n_b.\mathbf{out}\ (c, \mathsf{aenc}(\langle y_{na}, n_b\rangle, \mathsf{pk}_i))\\
&\mathbf{in}\ (c, z).\\
&\mathbf{if}\ \mathsf{adec}(z, \mathsf{sk}_r) = n_b\ \ \mathbf{then}\ Q
\end{aligned}
$$

协议执行

最终, 我们可以将 NSL 协议的整体执行表示为以下进程:

$$P_1 := \nu \mathsf{sk}_a, \mathsf{sk}_b.(P_A(\mathsf{sk}_a, \mathsf{pk}(\mathsf{sk}_b)) \| P_B(\mathsf{sk}_b) \| \mathbf{out}\ (c, \mathsf{pk}(\mathsf{sk}_a)) \| \mathbf{out}\ (c, \mathsf{pk}(\mathsf{sk}_b)))$$

协议执行

然而, 上述建模并不能反应中间敌手的执行状态, 上述过程只满足了协议的正常执行过程. 为了更好地分析协议的安全性, 我们需要引入中间敌手的模型, 并将其与协议进程进行组合.

$$P_2 := \nu \mathsf{sk}_a, \mathsf{sk}_b.(P_A(\mathsf{sk}_a, \mathsf{pk}(\mathsf{sk}_b)) \| P_A(sk_a, \mathsf{pk}(\mathsf{sk}_c)) \| P_B(\mathsf{sk}_b) \| \\ \mathbf{out}\ (c, \mathsf{pk}(\mathsf{sk}_a)) \| \mathbf{out}\ (c, \mathsf{pk}(\mathsf{sk}_b)))$$

协议执行

$$P_2 := \nu \mathsf{sk}_a, \mathsf{sk}_b.(\mathbf{in}\ (c, x_{\mathsf{pk}}).P_A(\mathsf{sk}_a, x_{\mathsf{pk}}) \| P_B(\mathsf{sk}_b) \| \mathbf{out}\ (c, \mathsf{pk}(\mathsf{sk}_a)) \| \mathbf{out}\ (c, \mathsf{pk}(\mathsf{sk}_b))$$

语义

语义: 结构等价

定义 (结构等价)

进程 P 和 Q 结构等价, 记作 $P \equiv Q$, 如果它们可以通过以下规则相互转换:

- 交换律: $P \| Q \equiv Q \| P$
- 结合律: $(P \| Q) \| R \equiv P \| (Q \| R)$
- 单位元: $P \| 0 \equiv P$
- 复制展开: $!P \equiv P \| !P$
- 单位限缩: $\nu n.0 \equiv 0$
- 限缩交换: $\nu n.\nu m.P \equiv \nu m.\nu n.P$
- 限缩与并发: 如果 $n \notin \mathsf{n}(Q)$, 则 $\nu n.(P \| Q) \equiv (\nu n.P) \| Q$
- 链接: $\nu x.\{t/x\} \equiv 0$
- 子集: $\{t/x\} \| A \equiv \{t/x\} \| A\{t/x\}$
- 重写: 如果 $t =_E t'$, 则 $\{t/x\} \equiv \{t'/x\}$

样例

样例

证明在 $t_1 \equiv_E t_2$ 的情况下, 有 $\mathbf{out}\ (u, t_1).P \equiv \mathbf{out}\ (u, t_2).P$.

内部规则

定义 (内部规则)

进程 P 可以通过以下内部规则进行转换:

- 通信: $\mathbf{in}\ (u, x).P \| \mathbf{out}\ (u, t).Q \to P\{t/x\} \| Q$
- 条件判断: 如果 $t_1 =_E t_2$, 则 $\mathbf{if}\ t_1 = t_2\ \mathbf{then}\ P \mathbf{else}\ \ Q \to P$; 否则, $\mathbf{if}\ t_1 = t_2\ \mathbf{then}\ P \mathbf{else}\ \ Q \to Q$.

其中, α 表示动作标签, 可以是输入或输出操作.

References I


Milner, R. (1992).
Functions as processes.
Math. Struct. Comput. Sci., 2(2):119–141.