可证明安全 - 引言

钱宸
网络空间安全学院
山东大学

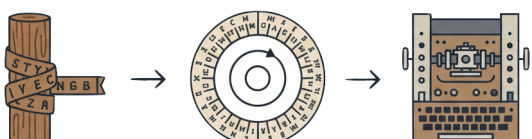
2025.09.15

Contents

Contents

前言

古典密码学



从古典密码到现代密码

古典密码与现代密码的区别

- 古典密码: 替换、置换
- 现代密码: 数学基础、计算复杂性

密码学

常见的密码学应用

- 机密性: 私钥加密、公钥加密
- 完整性: 消息认证码、数字签名
- 认证: 认证协议
- 密钥交换: 密钥交换协议
- 零知识证明
- ...

密码学

常见的密码学应用

- 机密性: 私钥加密、公钥加密
- 完整性: 消息认证码、数字签名
- 认证: 认证协议
- 密钥交换: 密钥交换协议
- 零知识证明
- ...

密码学

常见的密码学应用

- 机密性: 私钥加密、公钥加密
- 完整性: 消息认证码、数字签名
- 认证: 认证协议
- 密钥交换: 密钥交换协议
- 零知识证明
- ...

怎么论证安全性?

- 经验主义方法
- 数学证明方法

如何严格定义安全性?

敌手 $\mathcal{A}$

- 敌手 $\mathcal{A}$ 是一个试图攻击密码学方案的算法 (图灵机).
- 安全性定义通常将 $\mathcal{A}$ 的能力限制在多项式时间内的概率算法.

如何严格定义安全性?

敌手 $\mathcal{A}$

- 敌手 $\mathcal{A}$ 是一个试图攻击密码学方案的算法 (图灵机).
- 安全性定义通常将 $\mathcal{A}$ 的能力限制在多项式时间内的概率算法. (为什么呢?)

如何严格定义安全性?

安全性目标

- 机密性: 敌手 $\mathcal{A}$ 无法在多项式时间内区分两条消息的加密结果.
- 完整性: 敌手 $\mathcal{A}$ 无法在多项式时间内伪造合法的消息认证码.
- 认证: 敌手 $\mathcal{A}$ 无法在多项式时间内冒充合法用户.
- 密钥交换: 敌手 $\mathcal{A}$ 无法在多项式时间内计算出通信双方的会话密钥.

如何严格定义安全性?

上述所有目标都是说明 $\mathcal{A}$ 不能达成某个目标. 通常使用" 安全性游戏 $\mathbf{G}$" 来形式化这个过程. 更加严谨的定义:

安全性定义

- 使用 $\mathbf{G}_{\mathcal{A}} \Rightarrow 1$ 表示敌手 $\mathcal{A}$ 在安全性游戏 $\mathbf{G}$ 中获胜.
- 对于一个密码学方案 Π, 给定任意敌手 $\mathcal{A}$, 我们使用 $\mathsf{Adv}^{\mathbf{G}}_{\Pi,\mathcal{A}}$ 表示敌手 $\mathcal{A}$ 在安全性游戏 $\mathbf{G}$ 中获胜的概率, 即需要证明

$$\forall \mathcal{A}, \quad \mathsf{Adv}^{\mathbf{G}}_{\Pi,\mathcal{A}} = \Pr[\mathbf{G}_{\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda).$$

  这里需要对于所有多项式时间的敌手 $\mathcal{A}$ 成立, 其中 $\mathsf{negl}(\lambda)$ 是一个忽略函数.

如何严格定义安全性?

安全性证明可以看作以下断言的正确性证明:

$$\forall \mathcal{A}, \quad \mathsf{Adv}^{\mathbf{G}}_{\Pi,\mathcal{A}} = \Pr[\mathbf{G}_{\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda).$$

如何严格定义安全性?

安全性证明可以看作以下断言的正确性证明:

$$\forall \mathcal{A}, \quad \mathsf{Adv}^{\mathbf{G}}_{\Pi,\mathcal{A}} = \Pr[\mathbf{G}_{\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda).$$

安全性证明

对于一个协议 Π, 一个安全性游戏 $\mathbf{G}$, 给出一个上述断言的正确性证明.

如何严格定义安全性?

安全性证明可以看作以下断言的正确性证明:

$$\forall \mathcal{A}, \quad \mathsf{Adv}^{\mathbf{G}}_{\Pi,\mathcal{A}} = \Pr[\mathbf{G}_{\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda).$$

安全性证明

对于一个协议 Π, 一个安全性游戏 $\mathbf{G}$, 给出一个上述断言的正确性证明.

一些复杂度碎碎念

- 多项式时间: $O(n^k)$, 其中 k 是一个常数.
- 忽略函数: $\mathsf{negl}(\lambda)(n)$, 满足 $\forall c > 0, \exists n_c, \forall n > n_c, \mathsf{negl}(n) < \frac{1}{n^c}$.
- 安全参数: λ, 其中 1^λ 表示 1^λ, 即 λ 个 1 组成的字符串. 通常用 1^λ 使算法的输入长度为 λ.

例子：签名安全性

签名安全性定义 (不严谨)

- 签名安全性要求敌手 $\mathcal{A}$ 无法在多项式时间内伪造合法的签名.

例子：签名安全性

签名安全性定义 (不严谨)

- 签名安全性要求敌手 $\mathcal{A}$ 无法在多项式时间内伪造合法的签名.

UF-KOA(Unforgeability under Key-Only Attack)

- 敌手 $\mathcal{A}$ 只能获得签名者的公钥 (KOA)，并试图伪造签名 (UF-).

  $\mathbf{Exp}_{\Pi,\mathcal{A}}^{\text{UF-KOA}}(1^\lambda)$

  - $\mathsf{KGen} \xleftarrow{\$} (\mathsf{vk}, \mathsf{sk})$
  - $(\mathsf{m}^\star, \sigma^\star) \xleftarrow{\$} \mathcal{A}(\mathsf{vk})$
  - **if** $\mathsf{Ver}(\mathsf{vk}, \mathsf{m}^\star, \sigma^\star) = 1$ **then**
  - **return** 1
  - **else return** 0

例子：签名安全性

签名安全性定义 (不严谨)

- 签名安全性要求敌手 $\mathcal{A}$ 无法在多项式时间内伪造合法的签名.

UF-KOA(Unforgeability under Key-Only Attack)

- 敌手 $\mathcal{A}$ 只能获得签名者的公钥 (KOA)，并试图伪造签名 (UF-).

$\mathbf{Exp}^{\text{UF-KOA}}_{\Pi,\mathcal{A}}(1^\lambda)$

- $\mathsf{KGen} \xleftarrow{\$} (\mathsf{vk}, \mathsf{sk})$
- $(\mathsf{m}^\star, \sigma^\star) \xleftarrow{\$} \mathcal{A}(\mathsf{vk})$
- **if** $\mathsf{Ver}(\mathsf{vk}, \mathsf{m}^\star, \sigma^\star) = 1$ **then**
- **return** 1
- **else return** 0

- 需要证明 $\forall \mathcal{A} : \mathsf{Adv}^{\text{UF-KOA}}_{\Pi,\mathcal{A}}(1^\lambda) = \Pr\left[\mathbf{Exp}^{\text{UF-KOA}}_{\Pi,\mathcal{A}}(1^\lambda) \Rightarrow 1\right] \leq \mathsf{negl}(\lambda)$.
- 这里需要对于所有多项式时间的敌手 $\mathcal{A}$ 成立.

安全性证明

本课程目的

- 什么样的证明是一个严谨的安全性证明?
- 如何构造一个严谨的安全性证明?
- 如何使用形式化验证工具来辅助安全性证明?

本课程结构

1. 密码学中的安全性证明
   - 基于规约的证明
   - 基于安全游戏的证明
   - 基于模拟的证明

本课程结构

1. 密码学中的安全性证明
   - 基于规约的证明
   - 基于安全游戏的证明
   - 基于模拟的证明
3. 自动形式化验证工具
   - 消息与演绎推理
   - 等式论与静态等价性
   - 密码学计算模型
   - 安全性质建模
   - 自动化验证

参考资料

课程参考书目

- 基于安全游戏的证明 [Bellare and Rogaway, 1994]
- 基于模拟的证明 [Lindell, 2016]
- 形式化验证 [Cortier et al., 2014]

References


Bellare, M. and Rogaway, P. (1994).
Entity authentication and key distribution.
In Stinson, D. R., editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 232–249.
Springer, Berlin, Heidelberg.


Cortier, V., Kremer, S., et al. (2014).
Formal models and techniques for analyzing security protocols: A tutorial.
Foundations and Trends® in Programming Languages, 1(3):151–267.


Lindell, Y. (2016).
How to simulate it - A tutorial on the simulation proof technique.
Cryptology ePrint Archive, Report 2016/046.