第 2 章：私钥密码
Private-Key Encryption

赵俊舟

西安交通大学网安学院
junzhou.zhao@xjtu.edu.cn

2025 年 12 月 20 日

目录

目录

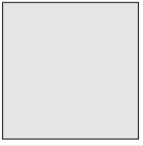
绝对安全的问题

- 敌手无法由密文获知关于明文的任何额外信息，即使敌手具有无限计算能力。这个要求太过苛刻（unnecessarily strong）。
- 如果一个密码体制能够被计算能力有限的敌手以非常小的概率破解，这种密码在实际中仍可能具有实用意义。

例 (实际中依然安全的密码)

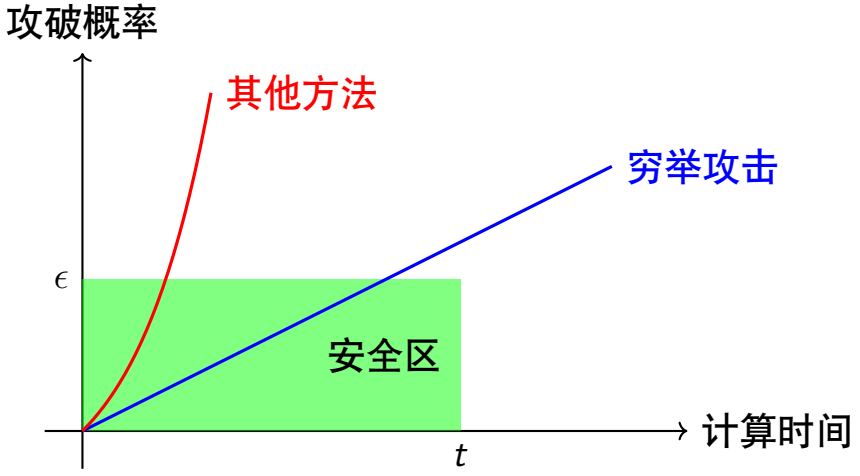
某密码即使让敌手使用当前最先进的计算机连续计算 200 年，成功破解密文的概率仍不超过 2^{-60}。

probability	*equivalent*
2^{-10}	*full house in 5-card poker*
2^{-20}	*royal flush in 5-card poker*
2^{-28}	*you win this week's Powerball jackpot*
2^{-40}	*royal flush in 2 consecutive poker games*
2^{-60}	*the next meteorite that hits Earth lands in this square* →



- 计算安全是设计现代密码的事实标准之一。

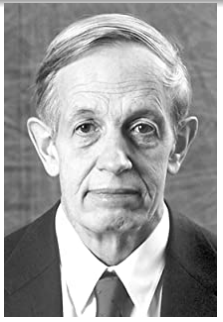
计算安全：具体安全（Concrete Secure）



- (t, ϵ)-安全：敌手计算时间最多为 t，成功攻破密码的概率不超过 ϵ。

John Nash 与计算安全

- 约翰 · 纳什（John Nash, 1928 – 2015），美国数学家，博弈论创建者。
- 1955 年，纳什在一封给 NSA 的信中提出了计算安全的思想。
- 遗憾的是，纳什的信一直处于机密状态，直到 2012 年才公开。



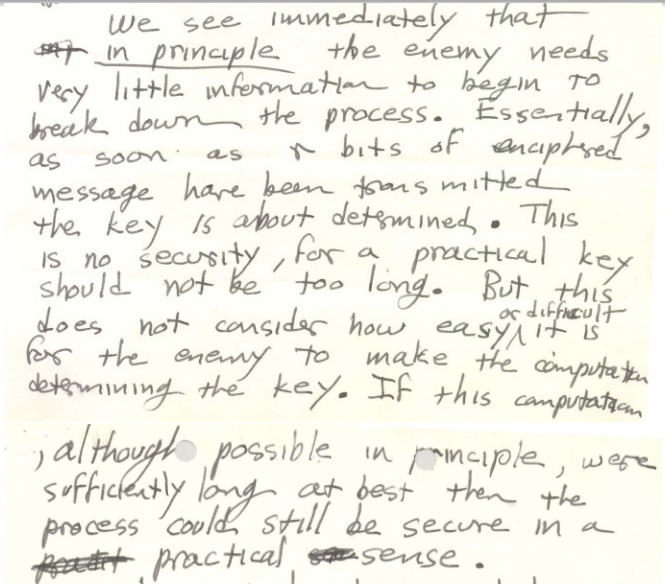
约翰 · 纳什

计算安全的思想

真正重要的是攻击是否在计算上不可行，而不是攻击是否完全不可能。

It doesn't really matter whether attacks are impossible, only whether attacks are computational infeasible.

纳什给 NSA 的信件



We see immediately that in principle the enemy needs very little information to begin to break down the process. Essentially, as soon as r bits of enciphered message have been transmitted the key is about determined. This is no security, for a practical key should not be too long. But this does not consider how easy, or difficult it is for the enemy to make the computation determining the key. If this computation, although possible in principle, were sufficiently long at best then the process could still be secure in a practical sense.

https://www.nsa.gov/Portals/70/documents/news-features/declassified-documents/nash-letters/nash_letters1.pdf

计算安全的渐进定义

- **安全参数** $n \in \mathbb{Z}$：一个整数，描述一个密码体制的安全性，n 越大越安全。通常，n 表示密钥的比特数。
- **多项式敌手**：指敌手的计算时间是关于 n 的多项式，即 $p(n)$。
- **概率多项式（PPT）敌手**：如果还允许敌手的计算有随机性。
- **可忽略概率**：对于任意多项式 p，当 n 足够大时，敌手攻破密码的概率小于 $1/p(n)$。

例 (可忽略函数，negligible function)

- 类似于 $2^{-n}, 2^{-\sqrt{n}}, n^{-\log n}$ 等函数都是可忽略的。
- 例如，对于任意 $c > 0$，解 $2^{-n} < n^{-c}$，得 $n > c \log n$；当 $c = 5$ 时，$n > 23$。

私钥密码

定义 (私钥密码，Private-Key Encryption Scheme)

一个私钥密码体制记为 $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$，其中三个算法都是 PPT 算法，并且满足：

- **密钥生成算法**：$k = \mathsf{Gen}(1^n)$ 并且 $|k| \geq n$
- **加密算法**：$c = \mathsf{Enc}_k(m)$ 其中 $m \in \{0,1\}^*$
- **解密算法**：$m = \mathsf{Dec}_k(c)$

密码体制 Π 满足**正确性要求**：$\forall n, k = \mathsf{Gen}(1^n), m \in \{0,1\}^*$，有

$$\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$$

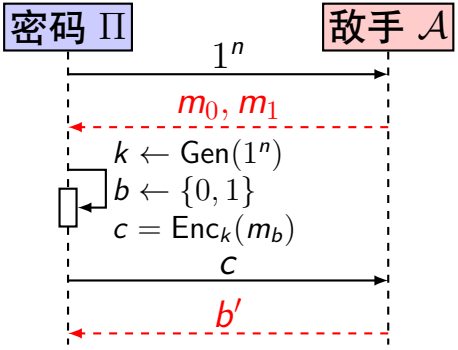
如果明文 $m \in \{0,1\}^{\ell(n)}$，则称 Π 为**定长私钥密码**或**定长对称密码**。

窃听不可区分与窃听安全（EAV-Security）

- 唯密文攻击
- 概率多项式敌手
- 敌手只观察到一条密文
- 如果对于任意 PPT 敌手 $\mathcal{A}$，都存在可忽略函数 negl，使

$$\Pr[\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)=1] \le \frac{1}{2}+\mathsf{negl}(n)$$

则称密码体制 Π 满足窃听不可区分或窃听安全（EAV-security）。



当 $b'=b$ 时，敌手成功，记为 $\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)=1$

目录

目录

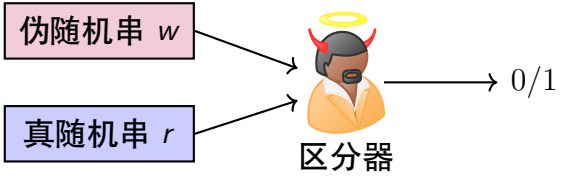
伪随机的基本概念

- 绝对安全（或 OTP）要求密钥长度至少等于消息长度，能否使用更短的密钥？
- 伪随机生成器：使用短随机数（也称为种子）来高效生成一个长的、看起来像随机数的确定性算法。
- 是否存在真随机？存在。例如，random.org 利用大气噪声产生真随机，并向外提供收费服务。

- 生成真随机数成本高、效率低，依然需要伪随机生成器。

如何严格定义伪随机？

- 早期做法（Golomb 随机性公设）：串的任一比特是 1 的概率接近 $1/2$、伪随机串中 $0/1$ 的数量差别不大、游程数量分布情况、异自相关性、通过统计学中的分布假设检验，等等。
- 问题：难以穷举所有统计检验方法
- 利用不可区分性来定义伪随机：
  - 考虑一个高效的区分器 D: $\{0,1\}^{\ell(n)} \mapsto \{0,1\}$
  - 分别输入一个伪随机串和一个真随机串，如果区分器 D 输出 1 的概率近似相等，则说明伪随机性好。



伪随机生成器的定义

定义 (伪随机生成器，Pseudorandom Generator, PRG)

如果一个确定性多项式算法 $G\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ 满足：

- **扩展性**：$\forall n, \ell(n) > n$
- **伪随机性**：对于任意 PPT 区分器 D，随机串 $r \leftarrow \{0,1\}^{\ell(n)}$，随机种子 $s \leftarrow \{0,1\}^n$，有

$$\left|\Pr[D(r)=1] - \Pr[D(G(s))=1]\right| \leq \mathsf{negl}(n)$$

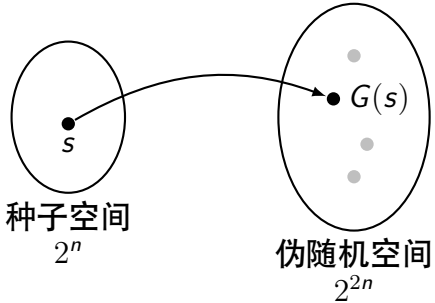
则称 G 为伪随机生成器，称 $\ell(n)$ 为扩展因子。

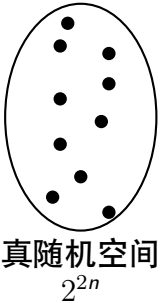
例 (判断 G 是否是伪随机生成器？)

- 定义 $G(s) \triangleq s \| \oplus_{i=1}^{n} s_i$，$G$ 是否是伪随机生成器？
- 定义区分器 $D(w) = \mathbf{1}(w_0 = \oplus_{i=1}^{n} w_i)$
- 则 $\Pr[D(G(s))=1] = 1$，对于随机串 r，$\Pr[D(r)=1] = 1/2$

对伪随机生成器的一种穷举攻击

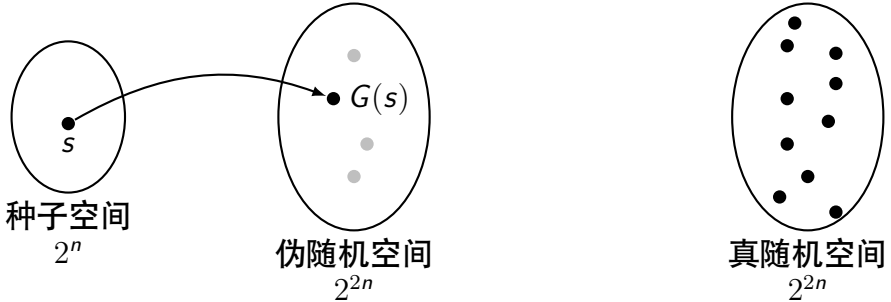
- 当 $\ell(n) = 2n$ 时，随机串空间为 $\{0,1\}^{2n}$，包含 2^{2n} 个串。
- 在真随机条件下，每个串被采到的概率为 2^{-2n}。
- 在伪随机条件下，由于种子长度为 n，PRG 值域最多有 2^n 个串，占总空间的比例不超过 $2^n/2^{2n} = 2^{-n}$。
- 即大部分串被采到的概率为 0。





对伪随机生成器的一种穷举攻击

- 考虑一个指数时间复杂度的区分器 D：对于任意串 w，当且仅当存在 s 且 $G(s) = w$ 时，$D(w) = 1$，否则 $D(w) = 0$
- 如果 w 由 PRG 产生，则 $\Pr[D(w) = 1] = 1$
- 如果 w 完全随机，则 $\Pr[w \in Range(G)] \leq 2^{-n}$
- 因此 $|\Pr[D(r) = 1] - \Pr[D(G(s)) = 1]| \geq 1 - 2^{-n}$

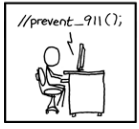


充分种子空间原则与伪随机生成器的存在性

- 指数时间复杂度的区分器其实是一种针对伪随机生成器的穷举攻击。
- 为抵抗穷举攻击，要求种子 s 足够随机，以便 $G(s)$ 足够随机，同时种子的长度 n 要足够大。
- 伪随机生成器的存在性假设：
  - 无法严格证明伪随机生成器是否必然存在，因此通常只能假设伪随机生成器存在。
  - 如果假设单向函数存在，那么可以由单向函数构建伪随机生成器。

真实案例（CVE-2008-0166）

- 2008 年，为避免一个编译警告，Debian 的一个发布版本中误删了一行代码，引起 OpenSSL 中关于伪随机生成器的漏洞。



IN THE RUSH TO CLEAN UP THE DEBIAN-OPENSSL FIASCO, A NUMBER OF OTHER MAJOR SECURITY HOLES HAVE BEEN UNCOVERED:

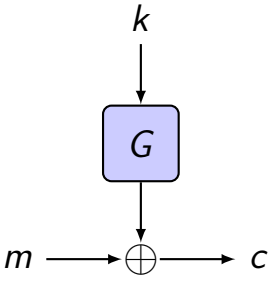
AFFECTED SYSTEM	SECURITY PROBLEM
FEDORA CORE	VULNERABLE TO CERTAIN DECODER RINGS
XANDROS (EEE PC)	GIVES ROOT ACCESS IF ASKED IN STERN VOICE
GENTOO	VULNERABLE TO FLATTERY
OLPC OS	VULNERABLE TO JEFF GOLDBLUM'S POWERBOOK
SLACKWARE	GIVES ROOT ACCESS IF USER SAYS ELVISH WORD FOR "FRIEND"
UBUNTU	TURNS OUT DISTRO IS ACTUALLY JUST WINDOWS VISTA WITH A FEW CUSTOM THEMES

目录

基于伪随机生成器构造窃听安全的私钥密码

设计 (满足窃听安全的私钥密码)

- G 是一个扩展因子为 $\ell(n)$ 的伪随机生成器，定义密码 $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 为：
- **密钥生成函数**：输入 1^n，输出随机密钥 $k \leftarrow \{0,1\}^n$
- **加密函数**：$\mathsf{Enc}_k(m) \triangleq G(k) \oplus m$
- **解密函数**：$\mathsf{Dec}_k(c) \triangleq G(k) \oplus c$



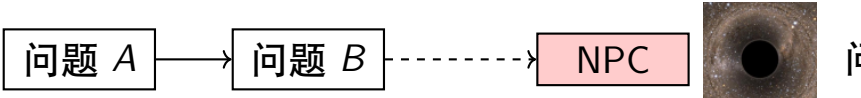
定理

使用伪随机生成器构造的私钥密码满足窃听安全。

❓ 如何证明？

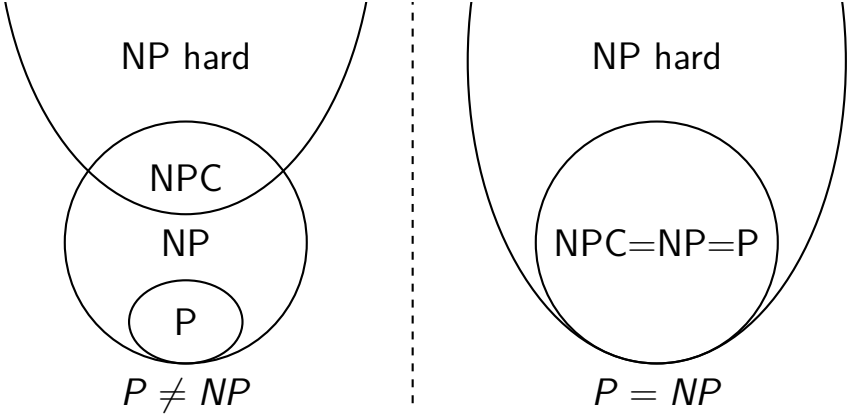
回顾：P 与 NP 问题

- P 问题：可以在多项式时间内被验证和求解的问题。
- NP 问题：可以在多项式时间内被验证的问题。
- 归约（Reduction）：**一个问题可以转化为另一个问题**，例如
  - 长方形面积计算问题可归约为梯形面积计算问题；
  - 一元一次方程求解问题可归约为一元二次方程求解问题。
- 通过归约可以建立问题之间的难易关系。
- NP 完全问题（NPC）：NP 问题中最难的一类问题，所有 NP 问题都可以归约为 NPC 问题。1972 年 Richard Karp 给出 21 个 NPC 问题（K21），并证明这些问题可以彼此归约。



问题黑洞

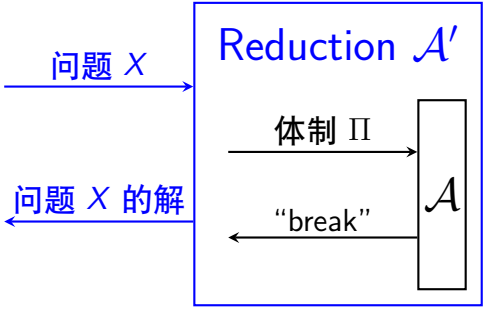
回顾：$P \overset{?}{=} NP$



- 如果能解决任意一个 NPC 问题，那就解决了所有 NPC 问题。
- 等价于证明了 $P = NP$，即所有 NP 问题都可高效求解。
- 目前，大多数科学家相信 $P \neq NP$，但还没有给出证明。

基于归约证明密码体制的安全性

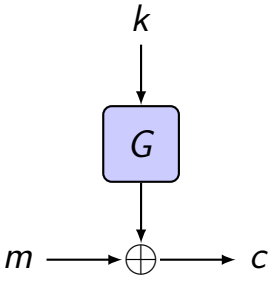
- **假设**：如果一个 PPT 敌手 $\mathcal{A}$ 可以攻破密码体制 Π。
- **构造**：问题 X 是一个很难的问题，例如 NPC；
- **归约**：构造一个 PPT 算法 $\mathcal{A}'$，$\mathcal{A}'$ 通过调用算法 $\mathcal{A}$ 可以解决问题 X。
- **矛盾**：因为问题 X 没有高效解法，所以这样的敌手 $\mathcal{A}$ 不存在，从而证明体制 Π 的安全性。



基于伪随机生成器构造窃听安全的私钥密码

设计 (满足窃听安全的私钥密码)

- G 是一个扩展因子为 $\ell(n)$ 的伪随机生成器，定义密码 $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ 为：
- **密钥生成函数**：输入 1^n，输出随机密钥 $k \leftarrow \{0,1\}^n$
- **加密函数**：$\mathrm{Enc}_k(m) \triangleq G(k) \oplus m$
- **解密函数**：$\mathrm{Dec}_k(c) \triangleq G(k) \oplus c$

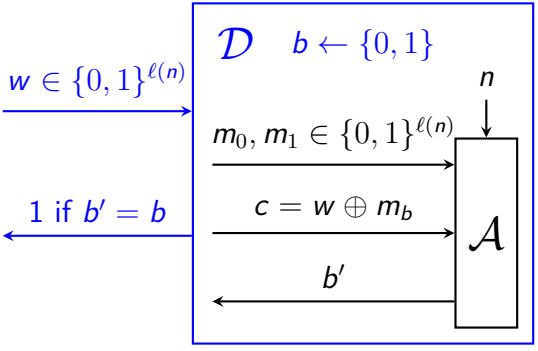


定理

使用伪随机生成器构造的私钥密码满足窃听安全。

证明思路

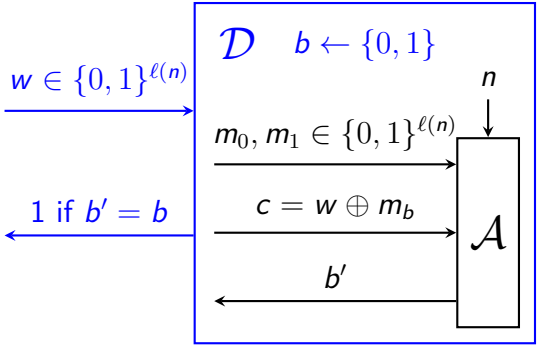
- 如果敌手 $\mathcal{A}$ 可以攻破密码体制 Π，那么可以用 $\mathcal{A}$ 构造一个区分器 $\mathcal{D}$，用来区分串 w 是由 PRG 产生还是完全随机。
- 但是实际中区分器 $\mathcal{D}$ 无法区分伪随机和真随机，所以敌手 $\mathcal{A}$ 无法攻破密码体制 Π。



- 如果 w 完全随机，则 Π 实为 OTP，不可攻破，$\mathcal{A}$ 输出 1 的概率为 $1/2$。
- 如果 w 由 PRG 产生，假设 $\mathcal{A}$ 可攻破，那么输出一个大于 $1/2$ 的概率。
- 那么 $\mathcal{A}$ 可以被用来区分一个串是 PRG 产生的还是完全随机串，违背了 PRG 假设。

构造区分器 $\mathcal{D}$

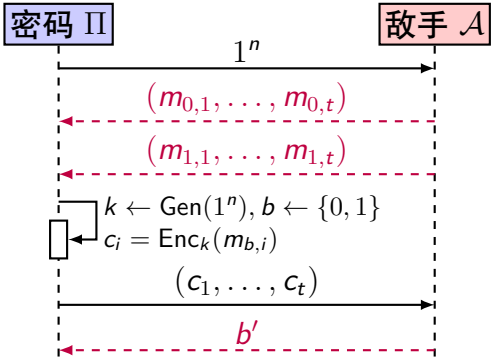
- **输入**：串 $w \in \{0,1\}^{\ell(n)}$
- 随机选择两个消息 $m_0, m_1 \in \{0,1\}^{\ell(n)}$
- 随机选择比特 $b \leftarrow \{0,1\}$
- 令 $c = w \oplus m_b$
- 将 c 发送给 $\mathcal{A}$，返回 b'
- **输出**：当 $b' = b$ 时，输出 1，否则输出 0



目录

目录

多密文窃听不可区分与多密文窃听安全



- 实际中敌手可能获得多条使用相同密钥加密的密文。
- **多密文窃听博弈**：当 $b' = b$ 时，敌手攻击成功，记为

$$\mathsf{PrivK}^{mult}_{\mathcal{A},\Pi}(n) = 1$$

- 如果

$$\Pr[\mathsf{PrivK}^{mult}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

称密码 $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 满足**多密文窃听不可区分**或**多密文窃听安全**。

窃听安全与多密文窃听安全

结论

窃听安全的密码不一定是多密文窃听安全的。

- OTP 满足窃听不可区分，但不满足多密文窃听不可区分。
- 假设 $\mathcal{A}$ 输出 $\vec{M}_0 = (0^\ell, 0^\ell), \vec{M}_1 = (0^\ell, 1^\ell)$
- 当 $\mathcal{A}$ 收到 (c_1, c_2) 时，如果 $c_1 = c_2$，$\mathcal{A}$ 输出 0，否则输出 1
- 可以看到 $\mathcal{A}$ 成功的概率为 1，原因在于 OTP 是确定性密码。

定理

如果密码体制 Π 的加密函数 Enc 是确定性函数，则 Π 不可能是多密文安全的。

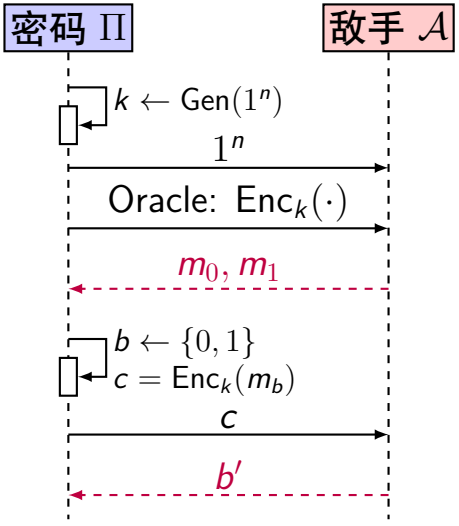
若要实现多密文窃听安全，则密码必须为随机密码。

目录

选择明文攻击（Chosen-Plaintext Attack, CPA）

- 选择明文攻击假设敌手可以控制通信双方发送的消息，通过观察这些消息在信道中的密文，从而破解密码。
- 选择明文攻击是一种比窃听攻击更强的攻击方式，是实际中真实存在的攻击手段。
- 二战期间，英军通过在特定位置的海域布置水雷，观察这些明文对应的德军密文，从而破解德军密码。
- 中途岛战役中，美军获得日军即将进攻 AF 的情报，但不知道 AF 具体代表哪个地方。美军猜测 AF 可能代表中途岛，于是发送假消息“中途岛缺少淡水”，然后观察日军随后的通信，发现日军密文中出现“AF 缺少淡水”这样的信息，于是确定 AF 代表中途岛。

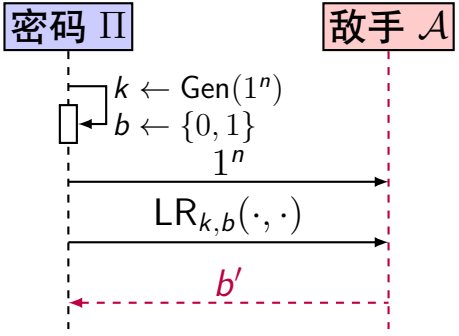
选择明文安全（CPA-Security）



- 当 $b' = b$ 时，敌手攻击成功，记为
$$\mathrm{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1$$
- 如果
$$\Pr[\mathrm{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathrm{negl}(n)$$
称密码 Π 满足CPA 不可区分或CPA 安全。
- CPA 安全是现代密码安全性的最低要求。

多密文 CPA 安全 (CPA-Security for Multiple Encryptions)

- 定义一个 Left-or-Right Oracle $\mathsf{LR}_{k,b}$：输入一对消息 m_0, m_1，计算 $c = \mathsf{Enc}_k(m_b)$ 并返回 c。
- 将 $\mathsf{LR}_{k,b}$ 交给敌手 $\mathcal{A}$，$\mathcal{A}$ 可以反复调用该 Oracle 得到多个密文，因此属于多密文窃听；同时 $\mathcal{A}$ 可以通过 $\mathsf{LR}_{k,b}(m,m)$ 得到 m 的密文，因此属于 CPA 攻击。



- 当 $b' = b$ 时，$\mathcal{A}$ 攻击成功，记
$$\mathsf{PrivK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1$$
- 如果
$$\Pr[\mathsf{PrivK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$
称密码 Π 是多密文 CPA 不可区分或多密文 CPA 安全。

多密文 CPA 安全 (CPA-Security for Multiple Encryptions)

- 当密码 Π 是多密文 CPA 安全时，Π 也是 CPA 安全的。
- 当密码 Π 是多密文 CPA 安全时，Π 也是多密文窃听安全的。
- CPA 安全等价于多密文 CPA 安全。
- CPA 安全是现代密码安全性的最低要求。

定理 (CPA 安全与多密文 CPA 安全的等价性)

任何一个私钥密码，如果它是 *CPA* 安全的，那么它也是多密文 *CPA* 安全的。

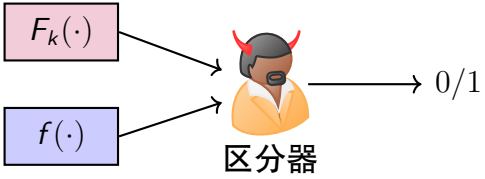
目录

目录

伪随机函数（Pseudorandom Functions, PRFs）

- 带密钥函数（keyed function）$F: \{0,1\}^* \times \{0,1\}^* \mapsto \{0,1\}^*$
- 给定密钥 k 和输入 x，能高效计算输出 $F(k,x)$。
- F 是一个二元函数，维度：$\ell_{key}(n), \ell_{in}(n), \ell_{out}(n)$。简化情况：$\ell_{key}(n) = \ell_{in}(n) = \ell_{out}(n) = n$。
- 一元函数表示：$F_k: \{0,1\}^n \mapsto \{0,1\}^n$，其中 $F_k(x) \triangleq F(k,x)$
- 选择不同的密钥 k，F_k 代表不同的映射。
- Func_n：把 n 比特串映射为 n 比特串的函数的集合。
- 如果随机选择一个密钥 k 得到函数 F_k，与从集合 Func_n 中随机选择一个函数 f，对于一个 PPT 区分器 D 来说不可区分，则称带密钥函数 F 是一个伪随机函数。

伪随机函数严格定义的一种尝试：利用区分器



- 集合 Func_t 中有多少个函数？
  - 每个函数看作一个映射表，表的第 x 行表示 $f(x)$；
  - 这个表共有 2^n 行，每行 n 比特，所以这个表可以用 $n \cdot 2^n$ 比特表示，所以映射表可以看作一个 $n \cdot 2^n$ 长的比特串；
  - 这样的串共有 $2^{n \cdot 2^n}$ 个，所以 $|\mathsf{Func}_n| = 2^{n \cdot 2^n}$。
- 将 F_k 和 f 表示为两个比特串，交给区分器辨别。
- 问题：两个串是指数长的 $(n \cdot 2^n)$，区分器根本无法在多项式时间内读完输入，何况还要区分！

伪随机函数的严格定义

- 将函数表示为神谕（Oracle），记作 $\mathcal{O}$，输入 x，输出 $\mathcal{O}(x)$。
- 神谕可以是 F_k 或 f，如果在这两种情况下，区分器无法区分，则称 F 为伪随机函数。

定义 (伪随机函数，Pseudorandom Functions, PRFs)

给定一个带密钥函数 $F: \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^n$。随机选择 $k \leftarrow \{0,1\}^n$ 得到函数 F_k，随机选择函数 $f \leftarrow \mathsf{Func}_n$。如果存在一个 PPT 区分器 D 和可忽略函数 negl 满足

$$\left|\Pr[D^{F_k(\cdot)}(1^n)=1] - \Pr[D^{f(\cdot)}(1^n)=1]\right| \leq \mathsf{negl}(n)$$

则称带密钥函数 F 是伪随机函数。

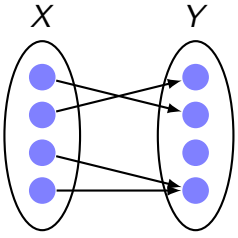
伪随机函数举例

例 (判定函数 F 是否为伪随机函数)

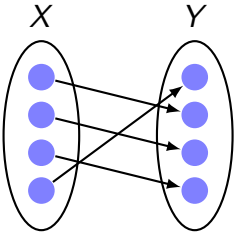
- 定义带密钥函数为 $F(k,x)=k\oplus x$，随机选择 k，$F_k(x)$ 是均匀分布的。
- 定义区分器 D:
  - 在两个不同输入 x_1 和 x_2 上访问神谕 $\mathcal{O}$，得到 $y_1=\mathcal{O}(x_1)$ 和 $y_2=\mathcal{O}(x_2)$。
  - 如果 $y_1\oplus y_2=x_1\oplus x_2$，则输出 1，否则输出 0。
- 当 $\mathcal{O}=F_k$ 时，D 始终输出 1。
- 当 $\mathcal{O}=f$ 时，利用 $x_1,x_2,f(x_2)$ 的独立性，得到
  $\Pr[f(x_1)\oplus f(x_2)=x_1\oplus x_2]=\Pr[f(x_1)=x_1\oplus x_2\oplus f(x_2)]=2^{-n}$
- 所以 F 不是伪随机函数。

伪随机置换（Pseudorandom Permutations）

- 置换是从一个集合到集合自身的双射函数。
- 带密钥置换函数：F_k 是双射，$\forall k$，因此存在逆置换 F_k^{-1}。
- 所有置换构成的集合 $\mathrm{Perm}_n \subset \mathrm{Func}_n$，只包含集合 Func_n 中各行互不相同的映射表。



伪随机函数



伪随机置换

伪随机置换（Pseudorandom Permutations）

定义（强伪随机置换）

给定一个带密钥函数 $F\colon \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^n$。随机选择 $k \leftarrow \{0,1\}^n$ 得到函数 F_k，随机选择函数 $f \leftarrow \mathrm{Perm}_n$。如果存在一个 PPT 区分器 D 和可忽略函数 negl 满足

$$\left|\Pr[D^{F_k(\cdot),F_k^{-1}(\cdot)}(1^n)=1] - \Pr[D^{f(\cdot),f^{-1}(\cdot)}(1^n)=1]\right| \leq \mathrm{negl}(n)$$

则称带密钥函数 F 是强伪随机置换。

定理（伪随机置换与伪随机函数的关系）

如果 F 是一个伪随机置换，且满足 $\ell_{in}(n) \geq n$，那么 F 也是一个伪随机函数。

目录

基于伪随机函数的 CPA 安全私钥密码

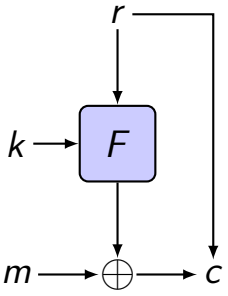
设计 (满足 CPA 安全的私钥密码)

- F 为伪随机函数，按如下方式定义密码体制 $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$：
- **密钥生成函数**：输入 1^n，输出 $k \leftarrow \{0,1\}^n$
- **加密**：输入密钥 k 和消息 $m \in \{0,1\}^n$，采样一个随机数 $r \leftarrow \{0,1\}^n$，输出密文

$$\mathsf{Enc}_k(m) = (r, F_k(r) \oplus m)$$

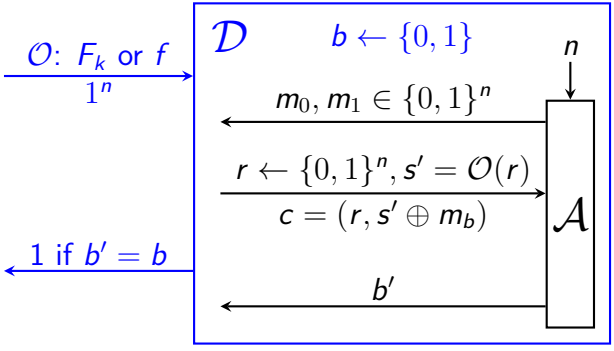
- **解密**：输入密钥 k 和密文 $c = (r, s)$，输出明文

$$\mathsf{Dec}_k(c) = F_k(r) \oplus s$$



证明思路

- First, analyze the security in an idealized world where f is used in $\widetilde{\Pi}$.
- Next, claim that if Π is insecure when F_k was used, then this would imply F_k is not PRF by reduction.



小结

1. 计算安全的定义
2. 构建窃听安全密码
3. 更强的计算安全定义
4. 构建 CPA 安全密码