十、RSA问题与加密

哈尔滨工业大学

张宇

2024春

概览

1. RSA问题（数论基础）
2. 攻击"书本上的RSA"
3. 实践中的RSA加密方案

RSA简介

- RSA: Ron Rivest, Adi Shamir and Leonard Adleman, 三位作者于1977年发表RSA加密方案。
- RSA问题: 给定 $N = pq$ (两个不同的大质数的乘积) 并且 $y \in \mathbb{Z}^*_N$, 计算 y^{-e}, 即y模N下的e次方根。
- 开放问题： RSA问题比分解 N 更容易吗?
- RSA相关标准: PKCS\#1 (RFC3447/8017), ANSI X9.31, IEEE 1363
- 密钥长度： 1,024 到 4,096 比特
- 已知最强的公开密码学分析：768比特密钥已经被破解
- RSA挑战赛： 破解 RSA-2048 来赢得 \$200,000 USD
- 密钥长度比较 ： 3072比特RSA密钥安全强度相当于128比特对称密钥

Symmetric	RSA
80 bits	1024 bits
128 bits	3072 bits
256 bits	15360 bits

书本上的RSA

- 构造：
  - Gen: 输入 1^n 运行 GenRSA(1^n) 产生 N, e, d。$pk = \langle N, e\rangle$ 和 $sk = \langle N, d\rangle$ 。
  - Enc: 输入 pk 和 $m \in \mathbb{Z}_N^*$，获得密文 $c := [m^e \bmod N]$.
  - Dec: 输入 sk 和 $m \in \mathbb{Z}_N^*$，获得明文 $m := [c^d \bmod N]$.
- 不安全性：由于“书本上的RSA”是确定性的，在我们已经提出的任何安全定义下都是不安全的。
- 下面学习问题：如何产生 N, e, d? 什么是 $\mathbb{Z}_N^*$? 如何计算 $m^e \bmod N$? 这个难题是TDP? 为什么很难?
- 参考教材：《A Computational Introduction to Number Theory and Algebra》(Version 2) Victor Shoup。

质数与模运算

- 整数集合 $\mathbb{Z}$, $a, b, c \in \mathbb{Z}$。
- a 整除 b: $a \mid b$ 如果 $\exists c, ac = b$ (否则 $a \nmid b$). b 是 a 的倍数。如果 $a \notin \{1, b\}$，那么 a 是 b 的因子。
- $p > 1$ 是质数（素数），如果其没有因子；否则，是合数。
- $\forall a, b$, $\exists$ 商 q, 余数 r: $a = qb + r$, 且 $0 \le r < b$。
- 最大公因子 $\gcd(a, b)$ 是最大的整数 c 使得 $c \mid a$ 且 $c \mid b$。 $\gcd(0, b) = b$, $\gcd(0, 0)$未定义。
- a 和 b 是互质，如果 $\gcd(a, b) = 1$。
- 余数 $r = [a \bmod N] = a - b\lfloor a/b \rfloor$ 并且 $r < N$. N 称为模。
- $\mathbb{Z}_N = \{0, 1, \ldots, N-1\} = \{a \bmod N | a \in \mathbb{Z}\}$.
- a 是模 N 下可逆的 $\iff$ $\gcd(a, N) = 1$。如果 $ab \equiv 1 \pmod{N}$，那么 $b = a^{-1}$是模 N 下 a 的乘法逆。

Primes and Modular Arithmetic

- The set of **integers** $\mathbb{Z}$, $a, b, c \in \mathbb{Z}$.
- $p > 1$ is **prime** if it has no factors; otherwise, **composite**.
- **Greatest common divisor** $\gcd(a, b)$ is the largest integer c such that $c \mid a$ and $c \mid b$. $\gcd(0, b) = b$, $\gcd(0, 0)$ undefined.
- Remainder $r = [a \bmod N] = a - b\lfloor a/b \rfloor$ and $r < N$. N is called **modulus**.
- $\mathbb{Z}_N = \{0, 1, \ldots, N-1\} = \{a \bmod N | a \in \mathbb{Z}\}$.
- a is **invertible modulo** $N \iff \gcd(a, N) = 1$. If $ab \equiv 1 \pmod{N}$, then $b = a^{-1}$ is **multiple inverse** of a **modulo** N.

课堂练习

- 欧几里德算法（辗转相除法）： $\gcd(a,b)=\gcd(b,[a \bmod b])$.
  - $\gcd(12, 27)$
- 扩展欧几里德算法：给定 a,N，寻找 X,Y 使得 $Xa+YN=\gcd(a,N)$ （贝祖定理）
  - 例子，求11 (mod 17)下的逆元，$a=11$，$N=17$，$Xa+YN=r$

```
r       X      Y   m
17      0      1
11      1      0   1
 6     -1      1   1
 5      2     -1   1
 1     -3      2
```

- 求余然后相加/乘
  - 计算 $193028 \cdot 190301 \bmod 100$
- 消去律：如果 $\gcd(a,N)=1$ 且 $ab\equiv ac \pmod{N}$，那么 $b\equiv c \pmod{N}$.
  - $a=3, c=10, b=2, N=24$

群

- $\mathbb{Z}_N^* \stackrel{\text{def}}{=} \{a \in \{1, \ldots, N-1\} | \gcd(a, N) = 1\}$
- 群是一个集合 $\mathbb{G}$ 带有一个二元操作 $\circ$:
  - 闭包：$\forall g, h \in \mathbb{G}, g \circ h \in \mathbb{G}.$
  - 单位元：$\exists$ 单位元 $e \in \mathbb{G}$ 使得 $\forall g \in \mathbb{G}, e \circ g = g = g \circ e.$
  - 逆元：$\forall g \in G, \exists\, h \in \mathbb{G}$ 使得 $g \circ h = e = h \circ g$. h 是 g 的逆元.
  - 结合律：$\forall g_1, g_2, g_3 \in \mathbb{G}, (g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3).$
- $\mathbb{G}$ with $\circ$ 是阿贝尔群，如果有交换律：$\forall g, h \in \mathbb{G}, g \circ h = h \circ g.$
- 逆元的存在意味着消去律
- 当 $\mathbb{G}$ 是有限群，$|\mathbb{G}|$ 是群的阶。
- 问题：$\mathbb{Z}_N^*$ 是乘法下的群吗？$\mathbb{Z}_N$ 在乘法下呢？$\mathbb{Z}_{15}^* =?$ $\mathbb{Z}_{13}^* =?$

Group

$$\mathbb{Z}_N^* \stackrel{\text{def}}{=} \{a \in \{1, \ldots, N-1\} | \gcd(a, N) = 1\}$$

A **group** is a set $\mathbb{G}$ with a binary operation $\circ$:

- (**Closure**:) $\forall g, h \in \mathbb{G}$, $g \circ h \in \mathbb{G}$.
- (**Existence of an Identity**:) $\exists$ **identity** $e \in \mathbb{G}$ such that $\forall g \in \mathbb{G}, e \circ g = g = g \circ e$.
- (**Existence of Inverses**:) $\forall g \in G$, $\exists\, h \in \mathbb{G}$ such that $g \circ h = e = h \circ g$. h is an **inverse** of g.
- (**Associativity**:) $\forall g_1, g_2, g_3 \in \mathbb{G}$, $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$.

$\mathbb{G}$ with $\circ$ is **abelian** if

- (**Commutativity**:) $\forall g, h \in \mathbb{G}, g \circ h = h \circ g$.

Existence of inverses implies **cancellation law**.
When $\mathbb{G}$ is a **finite group** and $|\mathbb{G}|$ is the **order** of group.

$\mathbb{Z}_{15}^* =?$ $\mathbb{Z}_{13}^* =?$ Is $\mathbb{Z}_N^*$ a group under '$\cdot$'?

群指数

- $g^m \stackrel{\text{def}}{=} \underbrace{g \circ g \circ \cdots \circ g}_{m \text{ times}}.$
- 欧拉定理：$\mathbb{G}$ 是有限群。那么，$\forall g \in \mathbb{G}, g^{|\mathbb{G}|} = 1.$
- 注：课上证明，将群中每个元素与 g 相乘后连乘等于群中元素连乘。
- 例子：计算 $3 \in \mathbb{Z}_7^*$ 的所有幂。
- 费马小定理：$\forall g \in \mathbb{G}$ and i, $g^i \equiv g^{[i \bmod |\mathbb{G}|]}$.
- 注：这是欧拉定理的推论。
- 例子：计算 $3^{78} \in \mathbb{Z}_7^*$

群上算法

- 加/减：线性时间 $O(n)$.
- 乘：最初 $O(n^2)$。
  - Karatsuba (1960，当时23岁): $O(n^{\log_2 3})$ $(2^b x_1 + x_0) \times (2^b y_1 + y_0)$ 使用3个乘法。
  - 注：因为 $x_1 \cdot y_0 + x_0 \cdot y_1 = (x_1 + x_0) \cdot (y_1 + y_0) - x_1 \cdot y_1 - x_0 \cdot y_0$。
  - 最佳渐进算法: $O(n \log n)$。
- 除/求余：$O(n^2)$。
- 指数：$O(n^3)$，平方指数法，例如计算8次幂并不需要乘8次，而是计算4次幂的平方，而4次幂来自2次幂平方。

Algorithm 1: Exponentiating by Squaring

input : $g \in G$; exponent $x = [x_n x_{n-1} \ldots x_2 x_1 x_0]_2$

output: g^x

```
y ← g; z ← 1
for i = 0 to n do
    if x_i == 1 then z ← z × y
    y ← y^2
return z
```

欧拉phi函数

- 欧拉phi函数：$\phi(N) \stackrel{\text{def}}{=} |\mathbb{Z}_N^*|$. *注：整数乘法群的阶*
- 算法基本定理：$N = \prod_i p_i^{e_i}$, $\{p_i\}$ 是不同的质数， $\phi(N) = \prod_i p_i^{e_i - 1}(p_i - 1)$。
- 例题：$N = pq$ 其中 p, q 是不同质数。$\phi(N) =? \ \phi(12) =? \ \phi(30) =?$
- 欧拉定理与费马小定理：$a \in \mathbb{Z}_N^*$. $a^{\phi(N)} \equiv 1 \pmod{N}$. *注：前面证明过*
- 如果 p 是质数并且 $a \in \{1, \ldots, p-1\}$，那么 $a^{p-1} \equiv 1 \pmod{p}$. *注：因为质数$p$乘法群的阶为$p-1$*
- 例题：$3^{43} \bmod 49 =?$

基于群指数函数的排列

- 指数函数 $f_e \ : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ by $f_e(x) = [x^e \bmod N]$.
- 对指数函数求逆：y 的 e 次方根: $x^e \equiv y$, $x \equiv y^{1/e}$.
- 推论：如果 $\gcd(e, \phi(N)) = 1$，那么 f_e 是排列。
- 证明：令 $d = [e^{-1} \bmod \phi(N)]$，那么 f_d 是 f_e 的逆函数。
  $y \equiv x^e; \quad f_d(y) \equiv y^d \equiv x^{ed} \equiv x.$
- 例题：在 $\mathbb{Z}_{10}^*$ 中, $e = 3$, $d =?$, $f_e(3) =?$, $f_d(f_e(3)) =?$, $9^{\frac{1}{3}} =?$
- 问题：如果对于某些特别的N无法计算 $\phi(N)$，那么会如何？如果不能分解 N 呢?

整数分解是难题

- 分解 $N = pq$. p, q 长度相同为 n.
- 尝试分解: $\mathcal{O}(\sqrt{N} \cdot \mathsf{polylog}(N))$.
- Pollard's $p-1$ 方法: 当 $p-1$ 具有小质数因子时有效。
- Pollard's rho 方法: $\mathcal{O}(N^{1/4} \cdot \mathsf{polylog}(N))$.
- 二次筛法 [Carl Pomerance]: 亚指数时间 $\mathcal{O}(\exp(\sqrt{n \cdot \log n}))$.
- 已知最优算法为通用数域筛法 [Pollard]： $\mathcal{O}(\exp(n^{1/3} \cdot (\log n)^{2/3}))$.

RSA问题是难题

- 思路：分解难 $\Longrightarrow$ 对于 $N = pq$, 找到 p, q 难 $\Longrightarrow$ 计算 $\phi(N) = (p-1)(q-1)$ 难

  $\Longrightarrow$ 无法模 $\phi(N)$ 计算

  $\Longrightarrow$ 计算 $e^{-1} \bmod \phi(N)$ 难

  这里存在一段空白

  $\Longrightarrow$ RSA 问题难：给定 $y \in \mathbb{Z}_N^*$, 计算 y^{-e} modulo N.

- 开放问题：RSA 比分解容易?

生成一个RSA问题

- 令 GenModulus(1^n) 为一个概率多项式时间算法，输入 1^n, 输出 (N, p, q) ，其中 $N = pq$, 并且 p, q 是 n 比特质数，除了有可忽略的概率失败。
- 产生RSA问题算法简述：
  1. 由GenModulus(1^n) 产生 (N, p, q) ；
  2. 计算$\phi(N) := (p-1)(q-1)$；
  3. 寻找一个e，使得$\gcd(e, \phi(N)) = 1$；
  4. 计算$d := [e^{-1} \bmod \phi(N)]$；
  5. 返回 N, e, d

RSA难题假设

- RSA实验 $\mathsf{RSAinv}_{\mathcal{A},\mathsf{GenRSA}}(n)$:
  1. 运行 $\mathsf{GenRSA}(1^n)$ 来产生 (N, e, d)。
  2. 选择 $y \leftarrow \mathbb{Z}_N^*$。
  3. 敌手 $\mathcal{A}$ 给定 N, e, y, 并输出 $x \in \mathbb{Z}_N^*$.
  4. $\mathsf{RSAinv}_{\mathcal{A},\mathsf{GenRSA}}(n) = 1$，实验成功，如果 $x^e \equiv y \pmod N$，否则实验失败 0 。
- 定义：RSA问题相对于GenRSA是难的，如果 $\forall$ PPT算法 $\mathcal{A}$, $\exists$ negl 使得，$\Pr[\mathsf{RSAinv}_{\mathcal{A},\mathsf{GenRSA}}(n) = 1] \leq \mathsf{negl}(n)$.

构造陷门排列

- 用 GenRSA 来定义一个排列族：
  - Gen: 输入 1^n, 运行 GenRSA(1^n) 来产生 (N, e, d) 并且 $I = \langle N, e \rangle$, td $= d$, 令 $\mathcal{D}_I = \mathcal{D}_{\text{td}} = \mathbb{Z}_N^*$.
  - Samp: 输入 I, 挑选一个随机元素 x of $\mathbb{Z}_N^*$.
  - $f_I(x) = [x^e \bmod N]$.
  - 确定性求逆算法 $\text{Inv}_{\text{td}}(y) = [y^d \bmod N]$.
- 将RSA问题规约到陷门排列求逆问题。

攻击e较小的书本上的RSA

- 小 e 和 小 m 令模算术失去作用，不再是难题。
  - 如果 $e = 3$ 并且 $m < N^{1/3}$，那么 $c = m^3$ 并且 $m = ?$
  - 在混合加密中，1024比特 RSA 与 128比特 AES。
- 当小e 被使用时通用攻击：
  - $e = 3$, 同一个消息 m 被发送给 3 个不同的接收者。
  - $c_1 = [m^3 \bmod N_1]$, $c_2 = [m^3 \bmod N_2]$, $c_3 = [m^3 \bmod N_3]$.
  - N_1, N_2, N_3 互质, 并且 $N^* = N_1 N_2 N_3$，使用中国剩余定理可知，∃ 唯一的 $\hat{c} < N^*$:
  - $\hat{c} \equiv c_1 \pmod{N_1}$, $\hat{c} \equiv c_2 \pmod{N_2}$, $\hat{c} \equiv c_3 \pmod{N_3}$.
  - $\hat{c} \equiv m^3 \pmod{N^*}$. 由于 $m^3 < N^*$, $m = \hat{c}^{1/3}$.

共模攻击

- 共模攻击使用相同的模数 N.
- 情况1：多个用户带有自己的密钥。每个用户可以以自己的 e, d 计算 $\phi(N)$，然后找到其他人的 d.
- 情况2：用两个公钥为同一个消息加密。
  - 假设 $\gcd(e_1, e_2) = 1$, $c_1 \equiv m^{e_1}$ and $c_2 \equiv m^{e_2} \pmod{N}$. $\exists X, Y$ 使得 $Xe_1 + Ye_2 = 1$（贝祖定理）.
  - $c_1^X \cdot c_2^Y \equiv m^{Xe_1} m^{Ye_2} \equiv m^1 \pmod{N}$.
  - $N = 15, e_1 = 3, e_2 = 5, c_1 = 8, c_2 = 2, m = ?$

CCA攻击

- 使用CCA恢复消息：敌手 $\mathcal{A}$ 选择一个随机数 $r \leftarrow \mathbb{Z}_N^*$ 并计算 $c' = [r^e \cdot c \bmod N]$，使用CCA获得 m' 。那么，$m =?$
- 在拍卖中讲价格翻倍：$c = [m^e \bmod N]. c' = [2^e c \bmod N]$.

RSA实现问题

- 将二进制串编码为 $\mathbb{Z}_N^*$ 中元素： $\ell = \|N\|$。任意长度为$\ell - 1$ 的二进制串 m 可以被看作是 Z_N 中元素。尽管 m 不在 Z_N^* 中，RSA 仍工作。
- e 的选择： $e = 3$ 或小 d 都是坏选择。 推荐 $e = 65537 = 2^{16} + 1$
- 使用中国剩余定理来加速解密： $[c^d \bmod N] \leftrightarrow ([c^d \bmod p], [c^d \bmod q])$.
- 假设一个 n 比特整数指数预算需要 n^3 操作。RSA 解密花费 $(2n)^3 = 8n^3$，其中使用中国剩余定理需要 $2n^3$。

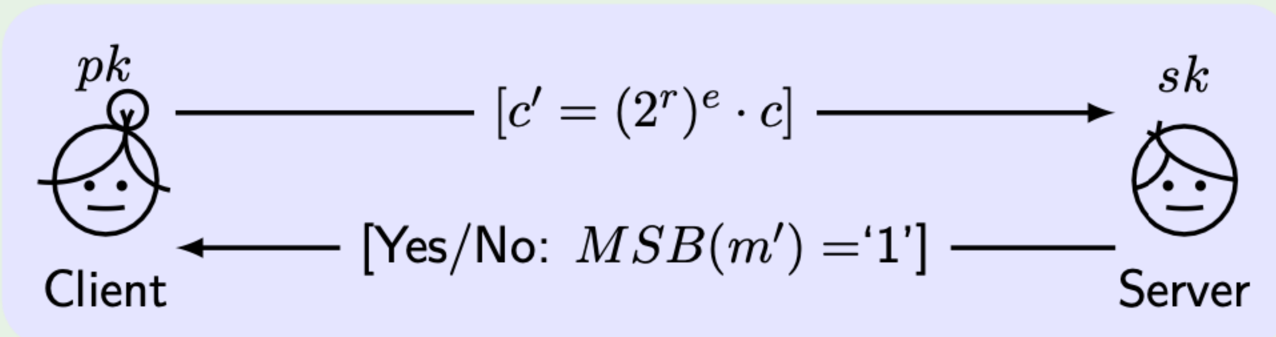
Padded RSA

- 思路：添加随机性来改进安全
- 构造：
  - 令 ℓ 为一个函数，对所有 n， $\ell(n) \leq 2n-2$，为被加密的消息长度。
  - Gen: 输入 1^n, 运行 GenRSA(1^n) 来产生 (N, e, d). 输出 $pk = \langle N, e\rangle$ 和 $sk = \langle N, d\rangle$。
  - Enc: 输入 $m \in \{0,1\}^{\ell(n)}$, 选择随机串 $r \leftarrow \{0,1\}^{\|N\|-\ell(n)-1}$. 输出 $c := [(r\|m)^e \bmod N]$。*注：填充随机串后加密*
  - Dec: 计算 $\hat{m} := [c^d \bmod N]$, 并输出 $\hat{m}$ 中的低$\ell(n)$个比特。*注：这部分为明文*
- ℓ 不应该太大 (理论上的 r 太小) 也不应该太小 (实践中的 m 太小)。
- 定理：如果RSA问题相对于GenRSA 是难的，那么基于 $\ell(n) = \mathcal{O}(\log n)$ 的构造是CPA安全的。
- 证明：与对称加密中CPA安全方案类似。

真实案例

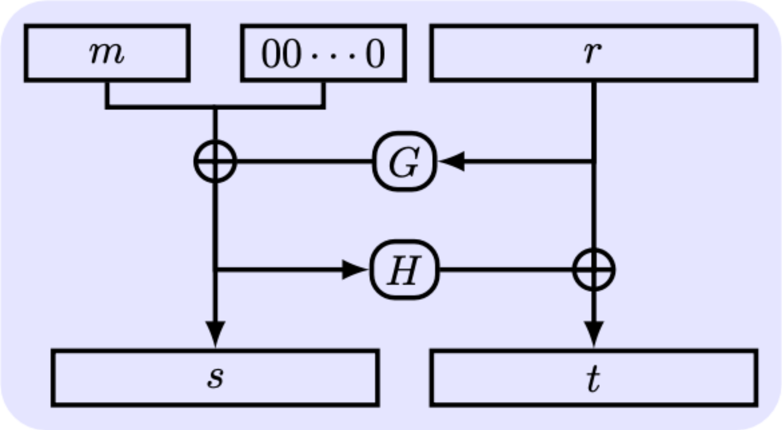
CCA on PKCS#1 v1.5 in HTTPS [Bleichenbacher 1998]

The message is padded in a format "$(00\|02\|s\|0\|m)$", where "02" means version 1. Here we simplfy $00\|02$ as the MSB of plaintext.



Defense: treating incorrectly formatted message blocks in a manner ("return a random string as the message") indistinguishable from correctly formatted blocks. See [RFC 5246]

OAEP



- 最优非对称加密填充（Optimal Asymmetric Encryption Padding，OAEP): 将长度 $n/2$ 的 m 编码为长度 $2n$ 的消息 $\hat{m}$ 。G, H 是随机预言机。
- RSA-OAEP在ROM下是CCA安全的。（当RO实例化后可能不安全）
- CPA攻击下，敌手不知道r，则m被完美保护；若要知道 r，则必须知道s，这不可能。
- CCA攻击下，无法有效进行解密查询，因为在应答前会检查明文中“00…0”。
- 局限性：这个方案对RSA是安全的，但对其他TDP可能不是。

实现攻击

- 计时攻击：[Kocher et al. 1997] 计算 c^d 所消耗的时间可能泄漏 d。（需要高解析时钟）
- 能耗攻击：[Kocher et al. 1999] 为计算c^d 智能卡消耗的能量可能泄漏d。
- 防御：将密文和随机数 r 绑定，解密 $r^{e}\cdot c$。
- 密钥生成问题：(在 OpenSSL RSA 密钥生成过程中):
- 相同的 p 由多个设备产生(源自启动时的低熵)，但是不同的 q (源自额外的随机性).
  - 问题: 不同设备的 N_1,N_2 , $\gcd(N_1,N_2) = ?$
  - 实验结果: 可分解 0.4% 的公开的HTTPS密钥。

故障攻击

- 故障攻击：在解密过程中 $c^d \bmod N$ 发生的计算机故障可能泄漏 d 。
- 之前提到过使用中国剩余定理来加速解密:

  $[c^d \bmod N] \leftrightarrow ([m_p \equiv c^d \pmod p], [m_q \equiv c^d \pmod q])$
- 假设在计算 m_q 时发生错误，但在计算 m_p 时没有错误。
- $m' \equiv c^d \pmod p, m' \not\equiv c^d \pmod q$。
- $(m')^e \equiv c \pmod p, (m')^e \not\equiv c \pmod q$
- $\gcd((m')^e - c, N) = \ ?$
- 防御：检查输出 (但减慢 10%)。

本节小结

- RSA问题是TPD，但书本上RSA加密不安全，RSA-OAEP在ROM下是CCA安全的。