

浅谈 RSA 算法原理及其破解与防御

姓名: 侯力广 学号: 521070910043

摘要: 作为非对称加密算法的代表, RSA 算法凭借其强大的安全性, 成为了目前公认的最为完善的公钥加密算法之一。本文详细介绍了 RSA 算法实现的数学原理及实现步骤, 并从实际应用的角度讨论了因子分解攻击、小指数攻击、共模攻击三种破解 RSA 的方法, 进而针对性提出了相应的防御措施。

关键词: RSA 算法; 原理; 破解; 防御;

一、引言

数据加密技术, 根据加密和解密使用的密钥是否相同, 可以分为对称加密和非对称加密。传统的对称加密算法, 加密和解密使用同一个密钥, 密钥一旦泄露密文便会公开, 通信的安全性难以保障。而非对称加密算法使用一对密钥进行加密与解密, 私钥由用户保存, 公钥可以自由分发, 公钥加密的信息只能由私钥解密, 大大提高了通信的安全性。RSA 算法作为非对称加密算法的代表, 由美国三位科学家 Rivest、Shamir 和 Adleman 于 1976 年提出, 该算法基于数论中大数的分解机制, 利用大数分解的困难性进行加密, 进一步实现了数据加密和数字签名的功能。时至今日人们仍然无法有效地破解 RSA 加密算法, 但随着计算机性能的提升, 依据其加密原理, 在一些特殊情况下, 仍然存在理论或实际上破解的可能。^[2]

二、RSA 算法

(一) RSA 的数学基础及证明

1. 欧拉函数

对于正整数 n , $\phi(n)$ 表示小于等于 n 的正整数中与 n 互质的数的个数

2. 完全剩余系

若 m 是一个给定的正整数, 则全部整数可分成 m 个集合, 记作 K_0, K_1, \dots, K_{m-1} , 其中 $K_r (r = 0, 1, \dots, m-1)$ 是由一切形如 $qm + r (q = 0, \pm 1, \pm 2, \dots)$ 的整数所组成的, K_0, K_1, \dots, K_{m-1} 叫做模 m 的剩余类

若 a_0, a_1, \dots, a_{m-1} 是 m 个整数, 并且其中任何两数都不同再一个剩余类里, 则 a_0, \dots, a_{m-1} 叫做模 m 的一个完全剩余系

3. 简化剩余系

如果一个模 m 的剩余类里面的数与 m 互质, 则称之为一个与模 m 互质的剩余类。在与模 m 互质的全部剩余类中, 从每一类各任取一数所作成的数的集合, 叫做模 m 的一个简化剩余系

4. 欧拉函数的积性

引理 1 若整数 a, b 与正整数 m 满足 $a \equiv b \pmod{m}, d \mid m, d > 0$, 则 $a \equiv b \pmod{d}$

证明 1 整数 a, b 对模 m 同余的充分与必要条件是 $a = b + mt$, 而 $a = b + mt = b + kdt = b + d(kt)$, 故 $a \equiv b \pmod{d}$

引理 2 若 $a \equiv b \pmod{m}, a = a_1d, b = b_1d, (d, m) = 1$, 则 $a_1 \equiv b_1 \pmod{m}$

证明 2 $a \equiv b \pmod{m}$, 则 $m \mid a - b$, 但 $a - b = d(a_1 - b_1), (d, m) = 1$, 故 $m \mid a_1 - b_1$, 即 $a_1 \equiv b_1 \pmod{m}$

引理 3 若 $a \equiv b \pmod{m}$, 则 $(a, m) = (b, m)$, 因而若 d 能整除 m 及 a, b 两数之一, 则 d 必能整除 a, b 中的另一个

证明 3 $a \equiv b \pmod{m}$, 则 $a = b + mt$, 故 a, m, b, m 具有相同的公因数, 因而 $(a, m) = (b, m)$

引理 4 若 m_1, m_2 是互质的两个正整数, 而 x_1, x_2 分别通过模 m_1, m_2 的完全剩余系, 则 $m_2x_1 + m_1x_2$ 通过模 m_1m_2 的完全剩余系

证明 4 x_1, x_2 分别通过 m_1, m_2 个整数, 因此 $m_2x_1 + m_1x_2$ 通过 m_1m_2 个整数, 由完全剩余系的定义可知, 只需证明这 m_1m_2 个整数对模 m_1m_2 两两不同余

假定 $m_2x'_1 + m_1x'_2 \equiv m_2x''_1 + m_1x''_2 \pmod{m_1m_2}$, 其中 x'_1, x''_1 是 x_1 所通过的完全剩余系中的整数, x'_2, x''_2 是 x_2 所通过的完全剩余系中的整数

由引理 1 得: $m_2x'_1 \equiv m_2x''_1 \pmod{m_1}, m_1x'_2 \equiv m_1x''_2 \pmod{m_2}$

由引理 2 及 $(m_1, m_2) = 1$ 得: $x'_1 \equiv x''_1 \pmod{m_1}, x'_2 \equiv x''_2 \pmod{m_2}$

从而 $x'_1 = x''_1, x'_2 = x''_2$, 故 $m_2x_1 + m_1x_2$ 通过的 m_1m_2 个整数对模 m_1m_2 两两同余。

引理 5 若 m_1, m_2 是两个互质的正整数, x_1, x_2 分别通过模 m_1, m_2 的简化剩余系,

则 $m_2x_1 + m_1x_2$ 通过模 m_1m_2 的简化剩余系

证明 5 由于简化剩余系是一个完全剩余系中一切与模互质的数组成的, 只需证明 $m_2x_1 + m_1x_2$ 通过模 m_1m_2 的一个完全剩余系中一切与模 m_1m_2 互质的整数

由引理 4 得: x_1, x_2 分别通过模 m_1, m_2 的完全剩余系, $m_2x_1 + m_1x_2$ 通过模 m_1m_2 的完全剩余系。若 $(x_1, m_1) = (x_2, m_2) = 1$, 由 $(m_1, m_2) = 1$ 得 $(m_2x_1, m_1) = (m_1x_2, m_2) = 1$, 故 $(m_2x_1 + m_1x_2, m_1) = 1, (m_2x_1 + m_1x_2, m_2) = 1$, 故 $(m_2x_1 + m_1x_2, m_1m_2) = 1$

反之, 若 $(m_2x_1 + m_1x_2, m_1m_2) = 1$, 则 $(m_2x_1 + m_1x_2, m_1) = (m_2x_1 + m_1x_2, m_2) = 1$ 由引理 3 得: $(m_2x_1, m_1) = (m_1x_2, m_2) = 1$, 而 $(m_1, m_2) = 1$, 故 $(x_1, m_1) = (x_2, m_2) = 1$

综上 $(m_2x_1 + m_1x_2, m_1m_2) = 1 \iff (m_2x_1, m_1) = (m_1x_2, m_2) = 1$, 故 $m_2x_1 + m_1x_2$ 通过模 m_1m_2 的一个完全剩余系中一切与模 m_1m_2 互质的整数

定理 6 若 m_1, m_2 是两个互质的正整数, 则 $\varphi(m_1m_2) = \varphi(m_1) \cdot \varphi(m_2)$

证明 6 由引理 5 知, 若 x_1, x_2 分别通过模 m_1, m_2 的简化剩余系, 则 $m_2x_1 + m_1x_2$ 通过模 m_1m_2 的简化剩余系, 即 $m_2x_1 + m_1x_2$ 通过 $\varphi(m_1m_2)$ 个整数. 另一方面由于 x_1 通过 $\varphi(m_1)$ 个整数, x_2 通过 $\varphi(m_2)$ 个整数, 因此 $m_2x_1 + m_1x_2$ 通过 $\varphi(m_1)\varphi(m_2)$ 个整数. 故 $\varphi(m_1m_2) = \varphi(m_1)\varphi(m_2)$

5. 欧拉定理

引理 7 若 $(a, m) = 1$, x 通过模 m 的简化剩余系, 则 ax 通过模 m 的简化剩余系

证明 7 ax 通过 $\varphi(m)$ 个整数, 由于 $(a, m) = 1, (x, m) = 1$, 故 $(ax, m) = 1$. 反证, 若 $ax_1 \equiv ax_2 \pmod{m}$, 由引理 2, $x_1 \equiv x_2 \pmod{m}$, 这与 x 通过模 m 的简化剩余系矛盾, 故 ax 通过模 m 的简化剩余系成立

定理 8 设 m 是大于 1 的整数, $(a, m) = 1$, 则 $a^{\varphi(m)} \equiv 1 \pmod{m}$

证明 8 设 $r_1, r_2, \dots, r_{\varphi(m)}$ 是模 m 的简化剩余系, 则由引理 7, $ar_1, ar_2, \dots, ar_{\varphi(m)}$ 也是模 m 的简化剩余系, 故 $(ar_1) \cdots (ar_{\varphi(m)}) \equiv r_1r_2 \cdots r_{\varphi(m)} \pmod{m}$

整理可得: $a^{\varphi(m)} (r_1r_2 \cdots r_{\varphi(m)}) \equiv r_1r_2 \cdots r_{\varphi(m)} \pmod{m}$

而 $(r_1, m) = (r_2, m) = \cdots = (r_{\varphi(m)}, m) = 1$, 故 $(r_1r_2 \cdots r_{\varphi(m)}, m) = 1$

再用引理 2 即得: $a^{\varphi(m)} \equiv 1 \pmod{m}$

6. 费马定理

引理 9 设 $a = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$, 则 $\varphi(a) = a \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$

证明 9 由定理 6 可知 $\varphi(a) = \varphi(p_1^{a_1}) \varphi(p_2^{a_2}) \cdots \varphi(p_k^{a_k})$

往证: $\varphi(p^\alpha) = p^\alpha - p^{\alpha-1}$. 由 $\varphi(a)$ 的定义知 $\varphi(p^\alpha)$ 等于从 p^α 减去 $1, \dots, p^\alpha$ 中与 p^α

不互质的数的个数,亦即等于从 p^α 减去 $1, \dots, p^\alpha$ 中与 p 不互质的数的个数. 由于 p 是质数, 故 $\varphi(p^\alpha)$ 等于从 p^α 减去 $1, \dots, p^\alpha$ 中被 p 整除的数的个数.

由函数 $[x]$ 的性质可知 $1, \dots, p^\alpha$ 中被 p 整除的数的个数是 $\left[\frac{p^\alpha}{p}\right] = p^{a-1}$

故可得: $\varphi(p^\alpha) = p^\alpha - p^{a-1}$

综上 $\varphi(a) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1}) \cdots (p_k^{a_k} - p_k^{a_k-1}) = a \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$

定理 10 若 p 是素数, 则 $a^p \equiv a(\text{mod } p)$.

证明 10 若 $(a, p) = 1$, 由引理 9 得 $\varphi(p) = p(1 - \frac{1}{p}) = p - 1$. 又由定理 8 可知 $a^{p-1} \equiv 1(\text{mod } p)$, 因而 $a^p \equiv a(\text{mod } p)$.

若 $(a, p) \neq 1$, 则 $p \mid a$, 故 $a^p \equiv a \equiv 0(\text{mod } p)$.

(二) RSA 的实现及证明

1.RSA 实现步骤

(1) 取两个尽量大且不相等的质数 p 和 q , 计算 $N = pq$

(2) 计算 $\varphi(N) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$

(3) 选取正整数 e 满足 $1 < e < \varphi(N)$ 并且 e 与 $\varphi(N)$ 互质, 即 $(e, \varphi(N)) = 1$

(4) 计算 e 关于 $\varphi(N)$ 的模逆元 d , 即求 d 满足 $ed \equiv 1(\text{mod } \varphi(N))$

至此, RSA 算法的密钥已经产生, (e, N) 为公钥, (d, N) 为私钥 (也可互换)

(5) 加密变换: 对明文 a , 可利用私钥 (e, N) , 加密后变换为密文 b

$$b \equiv a^e(\text{mod } N)$$

(6) 解密变换: 对加密后的密文 b , 可利用私钥 (d, N) , 解密变换后还原为 a

$$b^d \equiv a^{ed} \equiv a^{1+k\varphi(N)} \equiv a(\text{mod } N)$$

实际操作中 e, N 被公开, 而 d 被秘密保管. 若有攻击者想获得私钥 d , 就必须知道 $\varphi(N)$. 而由 (2) 知, 求 $\varphi(N)$ 就需要知道 N 的质因子 p, q . 当 p, q 的位数很大时, 按照现有的数学方法, 即使加上现有的超级电子计算机, 也不可能在限定时间内知道 $\varphi(N)$ 的值, 因而不可能知道 d . 这就是说, RSA 的保密性能是很好的.^[1]

2.RSA 实现的证明

往证: $a^{1+k\varphi(N)} \equiv a(\text{mod } N)$

(1) 若 $(a, N) = 1$, 则由定理 8 可知 $a^{\varphi(N)} \equiv 1(\text{mod } N)$, 进而 $a^{k\varphi(N)} \equiv 1^k \equiv 1(\text{mod } N)$, 故 $a^{1+k\varphi(N)} \equiv a(\text{mod } N)$

(2) 若 $(a, N) \neq 1$, 只需证明 $a^{1+k\varphi(N)} \equiv a(\text{mod } p), a^{1+k\varphi(N)} \equiv a(\text{mod } q)$

<2.1> 如果 p 整除 a , 则 $p \mid a$, 故 $a^{1+k\varphi(N)} \equiv a \equiv 0(\text{mod } p), a^{1+k\varphi(N)} \equiv a \equiv 0(\text{mod } q)$.

<2.2> 如果 p 不整除 a , 则 $(p, a) = 1, \varphi(N) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$

由定理 10 可知: $a^{(p-1)} \equiv 1(\text{mod } p)$, 则 $a^{k(p-1)(q-1)} \equiv 1^{k(q-1)} \equiv 1(\text{mod } p)$

故可得: $a^{1+k\varphi(N)} \equiv a^{1+k(q-1)(p-1)} \equiv a(\text{mod } p)$, 同理可证: $a^{1+k\varphi(N)} \equiv a(\text{mod } q)$

综上: $a^{1+k\varphi(N)} \equiv a(\text{mod } p), a^{1+k\varphi(N)} \equiv a(\text{mod } q)$

三、RSA 算法的破解与防御

(一) RSA 的破解方法

1. 因子分解攻击

由于该算法的安全性是建立在大整数因子分解的困难上, 分解 N 与求 $\varphi(N)$ 等价, 若分解出 N 的因子, 则 RSA-N 是不安全的, 因此分解 N 是互联网中最明显的攻击方法. 通过计算 $\varphi(N) = (p-1)(q-1)$ 从而确定 $d \equiv e^{-1}(\text{mod } \varphi(N))$.

伴随着计算机技术的快速发展, 国外已经具备了对 130 位十进制整数进行因子分解的能力, 而且采用数域筛选法可分解比 RSA-129 更大的数.^[3]

2. 小指数攻击

为了保证 RSA 算法的安全性, 必须使得选择的大素数位数极大, 从而导致 RSA 算法的加解密速度较慢. 所以一些使用 RSA 算法进行加密的机构采用了一种提升 RSA 速度并且能使加密易于实现的解决方案——令公钥 e 取较小的值.

由于 $b \equiv a^e(\text{mod } N)$, 同时也可以表示为 $a^e = kN + b$. 在实现加密时, 如果需要加密的明文比较小并且选取的 e 的值也很小, 则可能出现上式中 k 的值很小, 甚至当 $a^e < N$ 时出现 $k = 0$ 的情况.

$k = 0$ 时, $b = a^e$, 并且因为 e 的值不大, 可以直接对于所有可能范围内的 e 值对已知的明文 b 进行开 e 次方的操作解出明文 a 的值.

$k \neq 0$ 时, 亦可对可能满足条件的 k 的值进行遍历, 同样可以求解出明文 a 的值.^[2]

3. 共模攻击

在使用 RSA 算法加密时, 如果不同的用户使用的公钥中的模值 n 相同时, 算法体系存在共模攻击的可能性, 可以在不需要对 n 进行分解的情况下破解^[4].

若两个不同的用户使用公钥 $(N, e_1)(N, e_2)$ 对同一信息进行加密得到不同的密文 b_1, b_2 , 并且 e_1, e_2 互质, 则根据最大公因数的性质, 必存在两整数 s, t 使得 $s * e_1 + t * e_2 = 1$

故: $a = a^1 = a^{se_1 + te_2}$, 再两边同时对模 N 取余

$$\begin{aligned}
a \bmod N &= a = a^{se_1 + te_2} \bmod N \equiv (a^{se_1} \bmod N)(a^{te_2} \bmod N) \bmod N \\
&\equiv ((a^{e_1} \bmod N)^s \bmod N)((a^{e_2} \bmod N)^t \bmod N) \bmod N \\
&= (b_1^s \bmod N)(b_2^t \bmod N) \bmod N
\end{aligned}$$

可以表示为: $m = (b_1^s \bmod N)(b_2^t \bmod N) \bmod N^{[2]}$

由上式可以看出, 当使用相同的模 N 以及互质的两个 e_1, e_2 对同一明文进行加密时, 可以利用求得的满足 $se_1 + te_2 = 1$ 的 s 和 t 以及密文 b_1, b_2 来求解出明文 m , 而不需进行运算量巨大的大数分解.

利用扩展欧几里得算法可以求解出满足上述条件的 s, t , 详细过程可见参考文献 [4].

(二) 防御方法

1. 针对因子分解攻击

(1) p, q 选择差异较大的大素数, 以确保分解 N 的困难性.

(2) p, q 选择强素数, 以确保因子分解攻击在有效时间内不会实现.^[3]

(3) 必须使 N 也达到足够大 (600bit 以上), 这样密钥长度一定会大于 2048 位, 无法反求出私钥 d 而实施攻击.^[3]

2. 针对小指数攻击

(1) e, d 都应取较大的值, 以降低暴力破解的可能性.

(2) e, d 用随机数填充, 使得密钥的长度大于 2048 位, 增大破解私钥 d 的难度.

3. 针对共模攻击

用户不要共用同一个模数 N , 使得只有通过推导密钥对才能恢复明文 a , 从而大大提高破译难度.

四、结语

本文分析了 RSA 加密算法的实现原理, 并提出了几种可能的攻击手段及对应的防御措施。这些攻击方式有的需要极大的运算时间, 在有效的时间段内很难实现; 有的需要满足比较苛刻的条件才有可能完成。在实际操作中利用 RSA 算法进行加密时, 对常见的几种攻击方式所需的条件进行注意和避免的话, 仍然可以保证 RSA 算法的可靠性。

很遗憾笔者个人能力有限, 本文对于 RSA 算法的攻击、防御、改进的讨论并不是十分深入。对于 RSA 的攻击方法, 如: 选择密文攻击、计时攻击、群体暴力破解等^[3]; 对

RSA 算法的改进，如多素数加密原理、快速分块模幂算法、双重 RSA 算法等^[5] 笔者也在学习。但笔者认为，安全与平衡永远是处于一种相互制衡、此消彼长的状态，没有绝对的安全，也没有绝对安全的加密算法。伴随密码学与计算机技术的迅速发展，必然会出现更多、更安全的加密算法，也会出现更丰富的针对新加密算法的破解方案。

参考文献

- [1] 闵嗣鹤. 初等数论 [M]. 北京: 高等教育出版社,2020.
- [2] 张文博, 冯梅, 李青, 江波. 基于 Python 的 RSA 加密算法及其几种破解方法的研究 [J]. 信息系统工程. 2020.12 . 132-134
- [3] 周绯菲. RSA 算法攻击手段与防御方案的探讨 [J]. 交通部管理干部学院学报. 2008.09 . 第 18 卷第 4 期. 38-41
- [4] 邹慧, 余梅生, 王建东. 有效解决 RSA 共模攻击的素数生成方案 [J]. 计算机工程与应用,2004,27:88-89+153.
- [5] 蒋翔, 胡静. 基于 RSA 算法的改进方法研究 [J]. 发展与创新. 2018.10 . 251-254