

RSA

Crazy_13754

2024 年 1 月 16 日

摘要

写了些关于rsa的东西

目录

1 引言	1
2 理论基础	3
2.1 欧拉定理	3
3 算法描述	3

1 引言

实际上，写这篇的时候发现有论文写的很清楚了^[1]。此外，网上各种博客内容已经一应俱全，本来把它们丢上来就可以了。你问我为什么还要写这篇文章？请看这个图 1：



图 1: This is an inserted jpg graphic

如果你还是没有明白我想说什么，请再看看这个表 1：

对写奇奇怪怪东西的看法	可以理解	不可理喻
支持	0.1%	0.0%
不支持	0.2%	99.7%

表 1: Table to test captions and labels

如果（虽然几乎是当然的）你还不理解，那就看看这些东西：



图 2: 冬の花



图 3: RickT

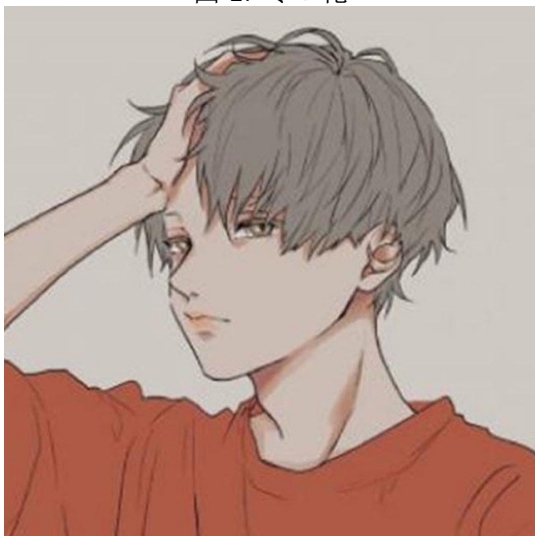


图 4: 朗格拉日

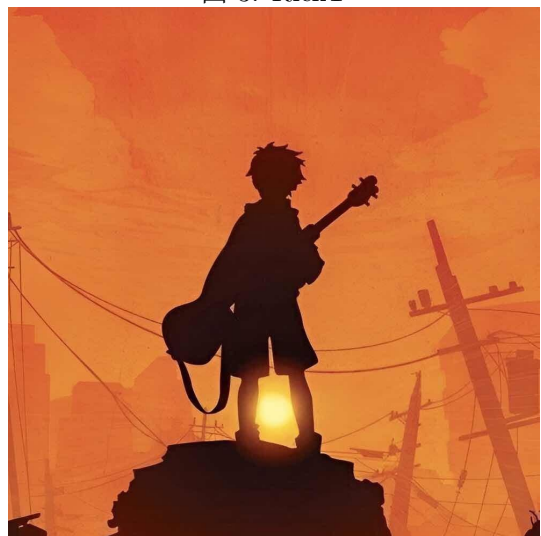


图 5: dx3906

这显然把事情弄得更糟。好吧，只是我在学 \LaTeX ，而你—我的朋友—浪费了你生命中的宝贵时间来看刚刚的内容，而且你接下来看到的东西也都可以在网上找到。不知道这篇文章所帮助你减少的在浏览器上搜索的时间能否弥补它所浪费的时间。或许，你需要的是及时止损，现在就放弃阅读？

公钥密码系统的观点是由Diffie和Hellman在1976年首次提出的,它是密码学发展史上具有里程碑意义的一件大事。与传统对称密码体制（即加、解密密钥相同）相比, 公钥系统使用两个密钥：加密密钥可以公开,称为公钥；解密密钥保密,为私钥。产生公钥体制的内在动力有两个：

- (1) 传统对称体制下密钥的存储和分配问题；
- (2) 消息鉴别问题,就是指用来检验消息来自于声称的来源并且没有被修改。

公钥体制的基础是陷门（单向函数）,即某种实际处理过程的不可逆性。目前的公钥思想基于两种：一是依赖于大数的因数分解的困难性；二是依赖于求模 p 离散对数的困难性。RSA密码算法就是基于大数的因数分解的困难性。

RSA加密算法是一种非对称加密算法，在公开密钥加密和电子商业中被广泛使用。RSA是由罗纳德·李维斯特(Ron Rivest)、阿迪·萨莫尔(Adi Shamir)和伦纳德·阿德曼(Leonard Adleman)在1977年一起提出的。当时他们三人都在麻省理工学院工作。RSA 就是他们三人姓氏开头字母拼在一起组成的。1973年，在英国政府通讯总部工作的数学家克利福德·柯克斯(Clifford Cocks)在一个内部文件中提出了一个与之等效的算法，但该算法被列入机密，直到1997年才得到公开。

这篇文章介绍了因子分解的相关算法，然而，这其中的很多方法对RSA来说并不适用。对于RSA的“暴力”攻击来说，普通数域筛法(GNFS)应该是最优的。如果我能看懂这个算法的话，会加上对它的介绍的……

2 数学基础

2.1 欧拉函数与欧拉定理

对于 $n \in \mathbb{Z}^*$ ，欧拉函数 $\varphi(n)$ 表示小于等于 n 的正整数中与其互质的数字个数。其可表示为：

$$\varphi(n) = \begin{cases} 1, & n = 1 \\ p^k(1 - \frac{1}{p}), & n = p^k, p \text{ 为素数}, k \in \mathbb{Z}^* \\ n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_r}), & n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}, \text{符号同上理} \end{cases}$$

n 一定属于以上某条件，因为大于1的正整数都可以写成一系列质数的积：若 n 是质数，则显然成立；若 n 不是质数，那它一定是个合数，用归纳法可以证明。

当 $n = p^k$ 时，不包含 p 的数才能与其互质。而包含 p 的数字有 $p, 2 \cdot p, \dots, p^{k-1} \cdot p$ 共 p^{k-1} 个，因此此时 $\varphi(n) = p^k - p^{k-1} = p^k(1 - \frac{1}{p})$ 。

当 $n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$ 时，先考虑 $n = p_1^{k_1} p_2^{k_2}$ 的情况。此时，我们有：

而欧拉定理指的则是，如果两个正整数 a 和 n 互质，则 n 的欧拉函数 $\varphi(n)$ 可以让下面的等式成立：

3 加密与解密过程

以下是算法描述：

- 取两个非常大的素数 p 和 q 。

- 令 $N = p \cdot q$, $\varphi(N) = (p - 1) \cdot (q - 1)$ 。
- 找到一个素数 $e < \varphi(N)$, 且要求 e 与 $\varphi(N)$ 互素, 即有 $\gcd(e, \varphi(N)) = 1$ 。
- 计算 e 在模 $\varphi(N)$ 上的逆元 d , 即求 d 满足 $e \cdot d \bmod \varphi(N) = 1$ 。
- (N, e) 作为公钥(pk, Public key)发给对方, (N, d) 作为私钥(pt, Private key)保存。
- 对方接受公钥后, 将原文(pt, Plaintext)通过预先设定好的方式转换成数字, 记为 p , 则密文(ct, Ciphertext)满足 $ct = p^e \bmod N$ (也就是 $ct \equiv p^e \pmod{N}$), 对方将密文发回。
- 接收密文, 并使用私钥解密: $p = ct^d \bmod N$, 也就是 $p \equiv ct^d \pmod{N}$ 。

参考文献

- [1] 陈传波 and 祝中涛. Rsa 算法应用及实现细节. 计算机工程与科学, 28(9):13–14, 2006.