文章编号:1007-0834(2004)04-0062-03

几种加密算法安全性的概率分析

刘晓真,王 瑗

(河南财经学院 计算机科学系,河南 郑州 450002)

摘要:DES、RSA与杂凑函数是现代加密技术的标志性成果,这些算法的安全性除了取决于密钥的保密措施外,还取决于算法本身.针对上述几种算法的安全性从概率分析的角度作一个比较,对于防止敌手攻击,保证加密信息的安全有着直接的意义.

关键词:加密技术;加密算法;安全杂凑算法

中图分类号:TP309 **文献标识码**:A

0 引言

密码编码学是密码体制的设计学;密码分析学是在未知密钥的情况下从密文推演出明文或密钥的技术.密码编码学与密码分析学合起来即为密码学.如果不论截取者获得了多少密文,但在密文中都没有足够的信息来唯一地确定出对应的明文,则这一密码体制称为无条件安全的,或称为理论上是不可破的.在无任何限制的条件下,目前几乎所有实用的密码体制均是可破的.因而,我们关心的是要研制出在计算上(不是在理论上)是不可破的密码体制.如果一个密码体制中的密码不能被可以使用的计算资源破译,则这一密码体制称为在计算上是安全的.信息论创始人香农(C.E.Shannon)在1949年发表著名文章,论证了一般经典加密方法得到的密文几乎都是可破的.但从20世纪60年代起,随着电子技术、计算技术等相关学科的迅速发展,密码学进入了一个新的发展时期.20世纪70年代后期,美国的数据加密标准DES和公开密钥密码体制的出现,使其成为近代密码学发展史上的两个重要里程碑.下面分别加以讨论.

1 DES

DES(Date Encryption Standerd)[1]是迄今为止世界上最为广泛使用和流行的一种分组密码算法,其分组长度为64 bit,密钥长度为56bit,由美国IBM公司研制.由于DES实际56bit的密钥长度,密钥量有$2^{56}\approx10^{17}$个,不足以抵御穷举攻击,于是,人们就尝试用DES进行多次加密,最简单的是用两个密钥进行两次加密.已给一个明文P和两个加密密钥K_1、K_2,密文为$C=E_{K2}(E_{K1}(P))$,解密是用相反的次序使用,$P=D_{K1}(D_{K2}(C))$,直到1992年已经有人证明了用56 bit的一次加密不可能等价于两重或多重加密.对于一个任意给定的明文P,双重DES产生的密文值有2^{64}种可能,双重DES实际使用了一个112bit的密钥,因此有2^{112}种可能的密钥,这样平均来说对于一个给定的明文P,将产生一个给定密文C的不同的112bit的密钥个数是$\frac{2^{112}}{2^{64}}=2^{48}$.因而上述过程对于第一对明文、密文有$2^{48}$次加、解密结果相等.如果再加上一对已知明、密文,误报率为$\frac{2^{48}}{2^{64}}=2^{-16}$.因此,已知两对明、密文,实施攻击检测到正确密钥的概率为$1\sim2^{-16}$,攻击的工作量为2^{56}.这与攻击DES的工作量2^{55}相差无几见表1.

表1 使用穷举法的平均破译时间

密钥长度(bit)	密钥总数	破译时间(搜索1次/μs)	破译时间(搜索10^6次/μs)
32	$2^{32}=4.3\times10^9$	$2^{31}\mu s=35.8$分	2.15毫秒
56	$2^{56}=7.2\times10^{16}$	$2^{55}\mu s=1\,142$年	10.01小时
128	$2^{128}=3.4\times10^{38}$	$2^{127}\mu s=5.4\times10^{24}$年	5.4×10^{18}年

可见,如果每微秒搜索10^6次,128位的密钥是

收稿日期:2004-09-16

作者简介:刘晓真(1963—),女,河南开封人,河南财经学院计算机科学系副教授,华中科技大学计算机学院2001级计算机理论与应用专业硕士研究生.

很安全的.

2 RSA

2.1 RSA 的加密算法

引理 1(欧几里得(Euclid)算法)

(1)求最大公约数

对任意非负整数 a 和正整数 b,有 $\gcd(a,b)=\gcd(b,a \bmod b)$.

(2)求乘法逆元

如果 $\gcd(a,b)=1$,则 a 在 mod b 下有乘法逆元,即存在一 $x(x<b)$,使得 $ax=1 \bmod b$.

引理 2

如果 p 为大于 2 的素数,则方程 $x^2 \equiv 1 (\bmod p)$ 的解只有 $x \equiv 1$ 和 $x \equiv -1$. 即只要存在一个 $x_0 \neq 1$ 且 $x_0 \neq -1$,满足 $x_0^2 \equiv 1 (\bmod p)$,即可知 p 不为素数.

引理 3(费尔玛(Fermat)定理)

若 p 是素数,a 是正整数且 $\gcd(a,b)=1$,则 $a^{p-1} \equiv 1 \bmod p$.

素数的寻找与判定

RSA 算法是建立在"大数分解和素数检测"的理论基础上的.首先需要测试一个整数为素数.对于大数的素性检验来说没有直接简单的方法,一些基本的测试方法如 Solovag - Strassen 方法、Lehmann 方法等都属于概率测试方法,下面介绍的是 Rabin - Miller 方法,这个算法是由 Rabin 发明的.事实上,它是在 NIST(国家标准和技术研究所)的 DSS 建议中推荐算法的一个简化版.素数检测就是判定一个给定的正整数是否为素数.

首先选择一个待测的随机数 p,计算 b,b 是 2 整除 $p-1$ 的 2 的次数(即 2^b 是能整除 $p-1$ 的 2 的最大幂数).然后计算 m,使得 $n=1+2^b m$.

检测程序如下[2]:随机选择大奇数 n,随机选择 $a<n-1$,

(1) 将 $n-1$ 表示为二进制形式 $b_k b_{k-1} \cdots b_0$;

(2) $d=1$

for i = k down to 0 do {

$x=d$;

$d=d^2 \bmod n$;

if $d=1$ and $(x \neq 1)$and$(x \neq n-1)$ then return TRUE;

if $b_i=1$ then $d=a\,d \bmod n$}

if $d \neq 1$ then return TRUE ;

return FALSE.

其中,n 是待检验的数.如果算法的返回值为 TRUE,则 n 肯定不是素数;如果返回值为 FALSE,则 n 有可能是素数.在实际测试中要多次返回 FALSE 才能以较大的概率判断 n 是一个素数.由引理 3 知,若 n 为素数,则 d 为 1;若 $d \neq 1$,则 n 不是素数.对 s 个不同的 a 运行该算法,如果算法每次返回都为 FALSE,则 n 是素数的概率为:$1-\frac{1}{2^s}$.

2.2 RSA 的安全性

RSA 的速度

非对称密钥密码系统与对称密钥密码系统相比较,确实有其不可替代的优点,但它的运算量远大于后者,超过几百倍、几千倍甚至上万倍,复杂得多.由于 RSA 涉及到大数的高次幂运算,所以用软件实现速度较慢,尤其在加密大量数据时.一般用硬件实现 RSA,速度较快,大约是 DES 的 1 500 分之一.

RSA 的安全性取决于大素数分解的困难性.两个大素数相乘在计算上是容易实现的,但将该乘积分解为两个大素数因子的计算量却相当巨大.如 $n=pq$被因数分解,RSA 便被攻破.因为若 p、q 已知,则 $\varphi(n)=(p-1)(q-1)$就可算出,由引理 1 可算出解密密钥 d.

一个 b 位二进制数 n 的因数分解大约需要的机器周期数为 $\exp\{[\ln n \ln\ln(n)]^{1/2}\}$,若机器周期为 1$\mu$s,则平均破译时间见表 2:

表 2 密钥长与平均破译时间的对比

密钥长度(bit)	分解时间
100	30 秒
200	3 天
300	9 年
500	100 万年
1 000	6×10^{15}年

因此,基于对 RSA 安全性的考虑,在设计 RSA 系统的时候,p、q 应满足以下几点:

(1)p、q 要足够大,一般应在 $10^{100} \sim 10^{125}$之间,这样可以基本保证不会在有效时间内密码被破解.

(2)如果 $p>q$,要求 $p-q$ 不宜太小,最好与 p、q 位数接近.如果 p 和 q 位数足够接近,则

$\frac{1}{2}(p+q) \approx n$,并且$\frac{1}{2}(p+q)$是一个相当小的数,于是,下式的右端是一个相当小的平方数,

$$\left(\frac{p+q}{2}\right)^2 - n = \left(\frac{p-q}{2}\right)^2$$

这样可以利用引理 3 的因数分解法将 n 分解因数.

(3)$(p-1)$与$(q-1)$的最大公约数应尽可能地小,否则,将有 d^2 个整数 b,使得 n 对基 b 是伪素数,这将增大 n 因数分解的可能性.

随着计算能力的持续增长和因式分解算法的不断改善,现在大数分解已经不像过去那么难了.

1977 年,RSA 的三个发明者在《科学美国人》的数学游戏专栏留了一个 129 位十进数(426bit),并悬赏 100 美元奖励分解该数的读者.当时,他们估计至少在 4 亿亿年后才能得到破译结果.然而,1994 年4 月,由 Atkins 等人在 Internet 上动用了 1 600 台计算机,仅仅工作了 8 个月之后就领到了这笔奖金.现在,人们已能分解 140 多个十进制位的大素数[3].

随着硬件资源的迅速发展和因数分解算法的不断改进,为保证 RSA 非对称密钥密码体制的安全性,最实际的做法是不断增加模 n 的位数.可能不久以后,1 024bit 甚至 2 048 *bit* 的密钥长度将比较合理.除此之外,研究者还建议采用一些其他限制,如:p 和 q 的长度接近;$p-1$ 和 $q-1$ 都应包含大的质因子等.

3 杂凑函数

杂凑函数是为 $h=H(M)$ 的函数,其中 M 是可变长消息,$H(M)$ 是固定长杂凑值.已知 h,求使得和 $H(x)=h$ 的 x 在计算上是不可行的[4].

称为 $H(x)$ 是一个强单向杂凑函数是指,对于任意的 $x_1 \neq x_2, f(x_1)=f(x_2)$ 在计算上是不可行的.

杂凑函数的安全问题是所谓的生日攻击,具体分为下列情形:

3.1 已知杂凑函数 f 有 n 个可能的输出值,其中 $f(x)$ 是一特定的输出值,现随机输入 k 个值 $E=\{x_1,x_2\cdots,x_n\}$,则至少存在一个 $x_0\in E$,使

$P[f(x_0)=f(x)]\geqslant 0.5$

问 k 有多大?

(如果 $n=365$ 时,该问题相当于问 k 有多大时,至少有一个人的生日是在某一个特定日子(如元旦).)

$1-\frac{(n-1)^k}{n^k}=1-\left(1-\frac{1}{n}\right)^k\geqslant 0.5,\left(1-\frac{1}{n}\right)^k\leqslant 0.5\because\left(1-\frac{1}{n}\right)^k\approx\frac{k}{n}\Rightarrow 1-\frac{k}{n}\leqslant 0.5,\frac{k}{n}\geqslant 0.5\therefore k\approx\frac{n}{2}$

若 n 是一 m bit 长的数,则 $k=\frac{2^m}{2}=2^{m-1}$

3.2 已知杂凑函数 f 有 n 个可能的输出值,其中 $f(x)$ 是一特定的输出值,现随机输入 k 个值 $E=\{x_1,x_2,\cdots,x_n\}$,则至少存在两个 $x_i、x_j\in E(1\leqslant i,j\leqslant n)$,使 $[f(x_i)=f(x_j)]\geqslant 0.5$

问 k 有多大?

(如果 $n=365$ 时,该问题相当于问 k 有多大时,k 个人中至少两人生日相同的概率大于 0.5?)假定 $k\leqslant 365$,则所有人生日不同的概率为

$$p=\frac{\frac{c_{365}^k}{k!}}{365^k}=\frac{365!}{(365-k)!\ 365^k}$$

当 $k=23$ 时,代入上式中可得,$p=0.507\ 3$ 即为所求.

顺便指出,当 $k=100$ 时,代入上式可得 p 的值要大得多,$p=0.999\ 999\ 7$.此即所谓的生日悖论.一般地,密码杂凑函数(如 MD5)的软件实现快于分组密码(如 DES)的软件实现.

4 小结

密码技术是保证网络、信息安全的核心技术,实现网络安全,需要评估风险及网络的安全操作代价,即网络安全最终是一个折中的方案.

参 考 文 献

[1] 谢希仁.计算机网络[M].北京:电子工业出版社,2001.

[2] 杨波.网络安全理论与应用[M].北京:电子工业出版社,2001.

[3] 沈金龙.计算机通信与网络[M].北京:北京邮电大学出版社,2002.

[4] 盛骤,谢式千,潘承毅.概率论与数理统计[M].北京:高等教育出版社,2003.

The Probability Comparative about Security of Several Encryption Algorithm

LIU Xiao-zhen, WANG Yuan

(*Department of Computer Science, Henan Institute of Finance and Economics, Zhengzhou* 450002, *China*)

Abstract: DES, RSA and Hash function are the symbolized results of modern encryption technique. The security of these algorithm depend on both the security steps of key and the algorithm itself. Next, the security of above algorithm will be compared from the view of probability analysis. It has the direct meaning to prevent enemies from attacking and guarantee the security of encryption information.

Key words: encryption technical; encryption algorithm; secure hash algorithm