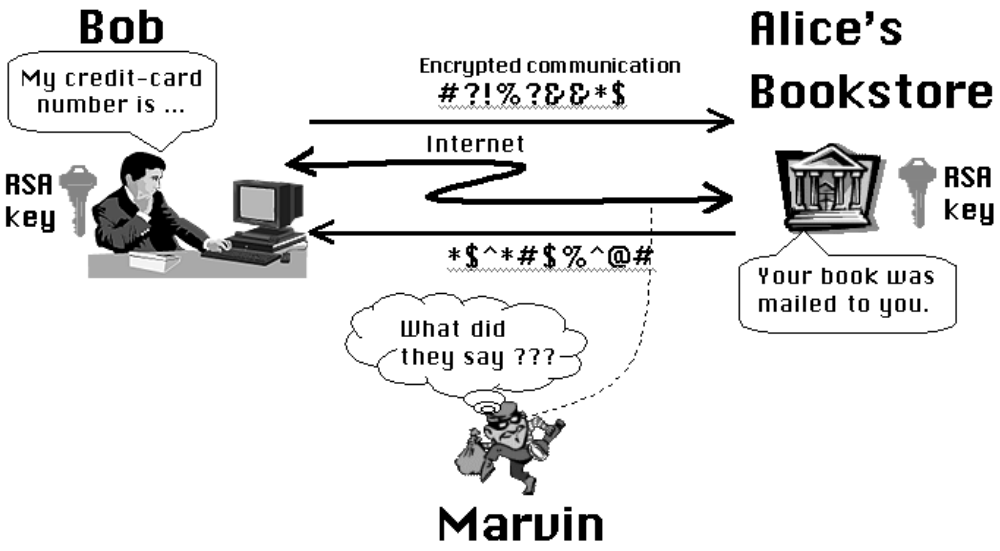
二十年以来对 RSA 密码系统攻击综述

原文：Twenty Years of Attacks on the RSA Cryptosystem
作者：Dan Boneh(dabo@cs.stanford.edu)
译者：Jing Ling(admin@hackfun.org)

1 介绍

由 Ron Rivest, Adi Shamir 和 Len Adleman 发明的 RSA 密码系统首次在 1977 年 8 月的“科学美国人”杂志上发表（译者注：本文于 1999 年 2 月在美国数学学会的 Notices 杂志首次发布）。密码系统最常用于提供隐私保护和确保数字数据的真实性。目前，RSA 部署在许多商业系统中。Web 服务器和浏览器使用它来保护 Web 流量，它可以用于保障电子邮件的隐私和真实性，还可以用于保护远程登录会话，同时它也是电子信用卡支付系统的核心。简而言之，RSA 常用于需要考虑数字数据安全性的应用中。

自最初发布以来，RSA 系统已被许多研究人员分析认为是易受攻击的。尽管二十年的研究出了许多引人入胜的攻击，但其中没有一个攻击是毁灭性的。这些攻击主要说明了不当使用 RSA 的危险，可见安全地实施 RSA 确实也是一项非常重要的任务。我们的目标是使用基础数学的知识分析其中的一些攻击，在整个分析过程中，我们遵循标准命名约定，使用 Alice 和 Bob 表示希望彼此通信的两个正常通信方，使用 Marvin 来表示希望窃听或篡改 Alice 和 Bob 之间通信的恶意攻击者。



我们首先介绍一下 RSA 加密的简化版本。令$N = pq$是比特位长度相同($n/2$位)的两个大素数的乘积。常见N的长度大小是n=1024 位（即：309 个十进制数字），质因子p, q=512 位。令e, d为满足$ed = 1 \bmod \varphi(N)$ 的两个整数，其中$\varphi(N) = (p-1)(q-1)$是乘法群$\mathbb{Z}_N^*$的阶数。我们称N为 RSA 模数，e为加密指数，d为解密指数。$\langle N, e\rangle$为公钥。顾名思义，公钥是公开的，并用于加密消息。$\langle N, d\rangle$称为密钥或私钥，一般情况下只有加密消息的接收者知道，私钥能够解密密文。

消息M是一个整数且满足$M \in \mathbb{Z}_N^*$，要加密M，计算$C = M^e \bmod N$，要解密密文合法接受者计算$M = C^d \bmod N$。（译者注：下面是译者添加的$M = C^d \bmod N$的证明）

欧拉定理：若a,n为正整数，且a,n互素（即$\gcd(a,n)=1$），则$a^{\varphi(n)}=1 \bmod n$。

已知$1= ed \bmod \varphi(n)$，$m<n$，$c=m^e \bmod n$，$\gcd(m,n)=1$，求证$m=c^d \bmod n$。

证明:
等式左边为m
等式右边为$c^d \bmod n$

$$\because c=m^e \bmod n$$
$$\therefore c^d \bmod n=(m^e \bmod n)^d \bmod n=m^{ed} \bmod n$$
$$\because 1= ed \bmod \varphi(n)$$
$$\therefore ed=k\varphi(n)+1$$
$$\therefore c^d \bmod n=m^{ed} \bmod n=m^{k\varphi(n)+1} \bmod n=m(m^{\varphi(n)})^k \bmod n$$
$$\because \gcd(m,n)=1$$
$$\therefore m^{\varphi(n)}=1 \bmod n$$
$$\therefore c^d \bmod n=m(m^{\varphi(n)})^k \bmod n=m(1 \bmod n)^k \bmod n=m(1)^k \bmod n=m \bmod n$$
$$\because m<n$$
$$\therefore m \bmod n=m$$
$$\therefore c^d \bmod n=m \bmod n=m$$

等式右边等于等式左边，证毕。

定义一个 RSA 函数$x \mapsto x^e mod\ N$，如果d已知，很容易使用等式$m=c^d \bmod n$求出x，我们称d为求解函数的陷门。在本次课题研究在没有陷门d的情况下求解 RSA 函数，更准确的说，给一个三元组$\langle N,e,C\rangle$，我们知道在不知道N的因子想计算C模N的e根是非常困难的。因为$\mathbb{Z}_N^*$是有限集，因此可以枚举$\mathbb{Z}_N^*$的所有元素直到找到M。遗憾的是，这将导致算法具有N阶的运行时间，即其输入大小的指数，其为$\log_2 N$的阶数。我们对运行时间更低的算法感兴趣，即n^c的阶数（其中$n=\log_2 N$）或者c是一些小的常数（比如说小于 5）。实践表明这些算法通常在所讨论的输入情况表现良好，在整篇论文中，我们将此类算法称为高效算法。

此次课题我们主要研究 RSA 函数而不是 RSA 密码系统。笼统地说，随机输入上求解 RSA 函数的应该是非常困难的，也就是意味着给定$\langle N,e,C\rangle$攻击者无法计算出明文M。这还不够，密码系统必须抵御更微妙的攻击。如果给出$\langle N,e,C\rangle$，想从M中计算出任何信息应该是非常难的，这被称为语义安全。我们不讨论这些

微妙的攻击，但是必须指出的是如上所述的 RSA 在语义上是不安全的：给定$\langle N, e, C\rangle$，可以容易地推导出关于明文M的一些信息（例如，可以容易地从C推导出N上的M的雅可比符号）。通过向加密过程添加随机处理流程，可以保障 RSA 在语义上的安全性。

RSA 函数$x \mapsto x^e mod\ N$是一个单向陷门函数，正向它可以很容易地计算，但是（据我们所知）除非在特殊情况下，否则在没有陷门d的情况下不能有效地反向求解的。单向陷门函数可用于数字签名，数字签名可以保障电子文件的真实性和不可否认性。例如，它们用于签署数字支票或电子采购订单。为了使用 RSA 对消息$M \in \mathbb{Z}_N^*$进行签名，Alice 使用其私钥$\langle N, d\rangle$签名M并获得签名$S = M^d \bmod N$。给定$\langle M, S\rangle$之后任何人都可以验证M上的 Alice 签名通过$M = S^e \bmod N$。因为只有 Alice 可以生成S，人们可能会认为攻击者无法伪造 Alice 的签名。然而事情并非如此简单，为了保障签名的安全还需要一些额外的措施。数字签名是 RSA 的重要应用，此次课题我们也会研究一些针对 RSA 数字签名的攻击。

RSA 密钥对可以这样生成，选取两个随机$n/2$位素数并将它们相乘以获得N来生成。然后，对于给定的加密指数$e < \varphi(N)$，使用扩展欧几里德算法计算$d = e^{-1} \bmod \varphi(N)$。由于素数集是足够密集的，因此可以通过重复选择随机$n/2$位整数并使用概率素性测试对每个素数进行素性测试来快速生成随机$n/2$位素数。

1.1 大数分解

给了 RSA 公钥$\langle N, e\rangle$，首先想到的攻击就是分解模数N，给了N的因子攻击者可以计算得到$\varphi(N)$，从而也可以计算得到解密指数$d = e^{-1} \bmod \varphi(N)$，我们称这种分解模数的方法为针对 RSA 的暴力攻击。虽然分解算法已经稳步改进，但

是在正确使用 RSA 情况下，当前的技术水平仍远未对 RSA 的安全性构成威胁。大整数分解是计算数学之美，不过本文研究主题并不是大整数分解。为完整起见，我们顺便提一下，目前普通数字域筛法是效率最高的分解方法。分解n位整数需要时间为$\exp\big(((c+o(1))n^{1/3}\log^{2/3}n)\big)$其中$c<2$，对 RSA 进行攻击的方法花费时间超过这个范围就那么吸引人了，比如暴力搜索M方法，还有一些 RSA 发布不久后的旧方法。

我们的目的是研究在不直接分解 RSA 模数N情况下解密消息的攻击方法，值得注意的是，一些 RSA 模数的稀疏集，$N=pq$可以很容易地被分解，举个例子，如果$p-1$是乘积的质因子且小于B，那么在小于B^3时间内分解N。

如上所述，如果存在有效的因式分解算法，则 RSA 是不安全的，反之亦然。这是一个由来已久的公开问题：必须要有一个N的因子才能有效地计算e^{th}模N的根数？破解 RSA 和因式分解一样难吗？我们在下面提出了具体的开放性问题。

开放性问题 1 给定N和e满足$\gcd\big(e,\varphi(N)\big)=1$，定义$f_{e,N}$函数：$\mathbb{Z}_N^*\to\mathbb{Z}_N^*$，$f_{e,N}(x)=x^{1/e}\bmod N$。是否有多项式时间算法$A$来计算给定$N$的因子以及对于某个$e$求得$f_{e,N}(x)$的一个谕言？

$f(x)$的谕言用于评估在单位时间内任何输入x的函数，最近 Boneh 和 Venkatesan 提供的证据表明，在e比较小的情况下，上述问题的答案可能是否定的。换句话说，在e比较小的情况下，可能不存在从分解到破解 RSA 的多项式时间缩减。他们通过实验表明在某个模型中对小e的问题的肯定答案会产生一个有效的因式分解算法。我们注意到，对开放问题 1 的肯定回答会引起对 RSA 的“选择密文攻击”。因此，否定的回答可能才是大家喜闻乐见的。

接下来，我们证明公开私钥d和分解N是等价的。由此可见，对于知道d的任

何一方来说，隐藏N的因式分解是没有意义的。

事实 1 $\langle N, e\rangle$为 RSA 的公钥，给定私钥d可以有效地分解模数$N = pq$。相反地，给定N的因式分解，可以有效地算出私钥d。

证明 N的因式分解得到$\varphi(N)$，因为e已知的，那么可以算出d，反之亦然。我们现在证明给定d可以分解N。给定d，计算$k = de - 1$。根据d和e的定义我们知道k是$\varphi(N)$的倍数。由于$\varphi(N)$是偶数，$k = 2^t r$其中r为奇数且$t \geq 1$。对于每个$g \in Z_N^*$都有$g^k = 1$，因此$g^{k/2}$是单位模N的平方根。根据中国剩余定理，1 有四个平方根模$N = pq$。其中两个平方根是± 1，另外两个是$\pm x$，其中x满足$x = 1 \bmod p$和$x = -1 \bmod q$。用这最后两个平方根中的任意一个，通过计算$\gcd(x - 1, N)$来揭示N的因式分解。一个直截了当的论证表明，如果从Z_N^*中随机选择g，那么序列中的一个元素$g^{k/2}$，$g^{k/4}$，…，$g^{k/2^t} \bmod N$是统一平方根的概率至少为 1/2，从而揭示了N的分解，序列中的所有元素可以在时间$O(n^3)$内有效地计算，其中$n = \log_2 N$。

2 基本攻击

我们首先描述一些老的基本攻击，这些攻击说明了 RSA 的公然滥用情况。虽然存在许多这样的攻击，但我们仅举两个例子。

2.1 共模

为了避免为每个用户生成不同的模数$N = pq$，人们可能希望一劳永逸地固定使用一个N，所有用户都使用相同的N。可信的中央机构可以向用户i提供唯一的一对e_i，d_i，用户i从其中生成公钥$\langle N, e\rangle$和私钥$\langle N_i, e_i\rangle$。

乍一看，这似乎行得通：为 Alice 准备的密文$C = M^{e_a} \bmod N$无法由 Bob 解密，因为 Bob 不知道d_a。但是，这是不正确的，由此产生的系统是不安全的。事

实上，Bob 可以使用他自己的指数e_b，d_b来分解N。一旦N被分解，Bob 就可以从她的公钥e_a中计算出 Alice 的私钥d_a。Simmons 的这一观察结果表明，RSA 模不应被一个以上的实体使用。

2.2 盲化

设$\langle N, d\rangle$是 Bob 的私钥，而$\langle N, e\rangle$是他相应的公钥。假设攻击者 Marvin 想要 Bob 在消息$M \in Z_N^*$上签名。当然 Bob 不是傻瓜，他拒绝签署M。但是 Marvin 可以尝试以下方法：他随机选择一个$r \in Z_N^*$并设$M' = r^e M \bmod N$。然后他让 Bob 在随机消息M'上签名。Bob 可能愿意在看上去没什么问题的M上签名S'，但是回想一下$S' = (M')^d \bmod N$, Marvin 现在简单地计算$S = S'/r \bmod N$就得到 Bob 在初始M上的签名S。

$$S^e = (S')^e / r^e = (M')^{ed} / r^e \equiv M'/r^e = M \pmod N$$

这种称为盲化的技术使 Marvin 能够在他选择的消息上获得有效的签名，方法是让 Bob 在随机的“盲化”消息上签名。Bob 不知道他实际在签名的是什么消息。由于大多数签名方案在签名之前对消息M应用“单向散列”算法，因此此种攻击倒不是一个严重的问题。尽管我们将盲化描述为一种攻击，但它实际上是实现匿名数字现金所需的一个有用属性(可以用来购买商品的现金，但不会透露购买者的身份)。

3 低私钥指数

为了减少加密时间（或签名生成时间），人们可能希望使用小值d而不是随机d。由于模幂运算需要花费线性时间为$\log_2 d$，所以小d可以使性能提高至少 10 倍(对于 1024 位模数而言)。不幸的是，由 M. Wiener 发现的一种巧妙的攻击表明，一个小的d会导致密码系统完全被攻破。

定理 2（M. Wiener）令$N = pq$且$q < p < 2q$，$d < \frac{1}{3} N^{1/4}$，给定$\langle N, e\rangle$且满

足$ed = 1 \bmod \varphi(N)$，攻击者可以有效计算出d。

证明 证明基于使用连分数的逼近，由于$ed = 1 \bmod \varphi(N)$，那么存在一个k满足$ed - k\varphi(N) = 1$。所以，

$$\left|\frac{e}{\varphi(N)} - \frac{k}{d}\right| = \frac{1}{d\varphi(N)}$$

因此，$\frac{k}{d}$是$\frac{e}{\varphi(N)}$的逼近，尽管 Marvin 不知道$\varphi(N)$，但是他可能会使用N去近似$\varphi(N)$。因为$\varphi(N) = N - p - q + 1$（译者注：$\varphi(N) = (p-1)(q-1) = pq - p - q + 1 = N - p - q + 1$），$p + q - 1 < 3\sqrt{N}$（译者注：因为$q^2 < pq = N$，所以$q < \sqrt{N}$，所以$N - \varphi(N) = p + q - 1 < 2q + q - 1 < 3q < 3\sqrt{N}$），我们有$|N - \varphi(N)| < 3\sqrt{N}$。

使用N替换$\varphi(N)$，我们得到：

$$\left|\frac{e}{N} - \frac{k}{d}\right| = \left|\frac{ed - kN}{Nd}\right| = \left|\frac{ed - k\varphi(N) - kN + k\varphi(N)}{Nd}\right|$$

$$= \left|\frac{1 - k\big(N - k\varphi(N)\big)}{Nd}\right| \leq \left|\frac{3k\sqrt{N}}{Nd}\right| = \frac{3k}{d\sqrt{N}}$$

现在，$k\varphi(N) = ed - 1 < ed$，因为$e < \varphi(N)$，我们知道$k < d < \frac{1}{3}N^{1/4}$（译者注：可以得到$3k < 3d < N^{1/4}$和$dN^{1/4} > 3d^2$）。因此我们得到：

$$\left|\frac{e}{N} - \frac{k}{d}\right| \leq \frac{1}{dN^{1/4}} < \frac{1}{2d^2}$$（译者注：有的资料写成$\frac{1}{3d^2}$）

这是一个经典的逼近关系，分数$\frac{k}{d}$且$d < \varphi(N)$在$\log_2 N$约束内非常逼近$\frac{e}{N}$。实际上，所有类似$\frac{k}{d}$这样的分数都是$\frac{e}{N}$的连分数展开的收敛。因此我们首要做的便是计算$\frac{e}{N}$的连分数的$\log N$收敛，其中一个连分数就等于$\frac{k}{d}$。因为$ed - k\varphi(N) = 1$，我们有$\gcd(k, d) = 1$，因此$\frac{k}{d}$是一个最简分数。这是可以算出密钥d的线性时间算法。

由于通常N都是 1024 位，因此d必须至少 256 位长才能避免这种攻击。这对于诸如“智能卡”之类的低功耗设备来说是不幸的，因为小d就能节省大量能耗。

然而，并不是毫无办法。Wiener 提出了许多能够实现快速解密并且不易受其攻击影响的技术：

使用大e：假设不是减小e模$\varphi(N)$，而是使用$\langle N, e'\rangle$作为公钥，其中对于某些大t有$e' = e + t \cdot \varphi(N)$。 显然，$e'$可以代替$e$用于消息加密，当使用大的$e$值时，上述证明中的$k$不再小。一个简单的计算表明，如果$e' > N^{1.5}$，那么无论$d$多小，都无法实施上述攻击。然而，大的$e$值将导致加密时间的增加。

使用 CRT：另一种方法是使用中国剩余定理（CRT）。假设选择d使得$d_p = d \bmod (p-1)$和$d_q = d \bmod (q-1)$都很小，比如都是 128 位。则可以进行如下密文C的快速解密：首先计算$M_p = C^{d_p} \bmod p$和$M_q = C^{d_q} \bmod q$。 然后使用 CRT 计算满足$M = M_p \bmod p$和$M = M_q \bmod q$的唯一值$M \in \mathbb{Z}_N$. 得到的M满足等式$M = C_d \bmod N$。关键点在于虽然d_p和d_q很小，但是$d \bmod \varphi(N)$的值可以很大，大约在$\varphi(N)$的数量级上。因此，定理 2 的攻击不再适用。我们注意到，如果给定了$\langle N, e\rangle$，则存在一种攻击能够使攻击者能够在时间$\mathrm{O}\left(\min(\sqrt{d_p}, \sqrt{d_q})\right)$内对$N$进行因子分解。因此，$d_p$和$d_q$不能太小。

我们不知道这些方法中是否都安全。我们所知道的是，Wiener 攻击对它们无效。最近由 Boneh 和 Durfee 改进的定理 2 证明了只要$\mathrm{d} < N^{0.292}$，攻击者就可以从$\langle N, e\rangle$中有效地算出d。这些结果表明 Wiener 的界限并不固定。正确的界限可能是$\mathrm{d} < N^{0.5}$。截至撰写本文时，还是一个尚未解决的问题。

开放性问题 2 令$N = pq$，$d < N^{0.5}$，如果 Marvin 知道$\langle N, e\rangle$和$ed = 1 \bmod \varphi(N)$及$e < \varphi(N)$关系，他能有效算出d吗？

4 低公钥指数

为了减少加密或签名验证时间，通常会使用一个小的公钥指数e。e的最小可

能值为 3，但为防止某些攻击，建议使用$e = 2^{16} + 1 = 65537$。当使用值$2^{16} + 1$时，签名验证需要 17 次乘法，而使用随机的$e \leq \varphi(N)$时则需要大约 1000 次乘法。与上一节的攻击不同，当使用一个小e时，针对的攻击不只是攻破而已。

4.1 Coppersmith 定理

针对 RSA 低公钥指数最有力的攻击基于 Copper-smith 的一个定理，Coppersmith 定理有很多应用，这里我们只讨论其中的一些应用，证明使用 LLL 格基约化算法如下。

定理 3（Coppersmith） 令N为一个整数，$f \in \mathbb{Z}[x]$是d次的一元多项式，设$X = N^{\frac{1}{d}-\epsilon}$其中$\epsilon \geq 0$，在给定$\langle N, f \rangle$之后 Marvin 能够有效找到所有满足$f(x_0) = 0 \bmod N$的整数$|x_0| < X$，运行时间由在$O(w)$维数且$w = \min(1/\epsilon, log_2 N)$的格上运行 LLL 算法所需的时间决定。

该定理为有效地求f模N的所有小于$X = N^{1/d}$的根提供了一种算法，当X越小，算法的运行时间越短。这个定理的强大之处在于它能够找到多项式的小根。当模数为素数时，就目前而言，找不到比使用 Coppersmith 定理更好的求根算法了，没有理由不使用 Coppersmith 定理。

我们概述了 Coppersmith 定理证明背后的主要思想，我们采用由 Howgrave-Graham 提出的简化方法，给定一个多项式$f(x) = \sum a_i x^i \in \mathbb{Z}[x]$，定义$\|h\|^2 = \sum_i |a_i|^2$，证明依赖于下面的观察。

引理 4 令$h(x) \in \mathbb{Z}[x]$为d次多项式，X为正整数，假设$\|h(xX)\| < N/\sqrt{d}$，如果$|x_0| < X$满足$h(x_0) = 0 \bmod N$，那么$h(x_0) = 0$成立。

证明 从 Schwarz 不等式观察到：

$$|h(x_0)| = \left|\sum a_i x_0^i\right| = \left|\sum a_i X^i \left(\frac{x_0}{X}\right)^i\right| \leq \sum \left|a_i X^i \left(\frac{x_0}{X}\right)^i\right|$$

$$\leq \sum \left|a_i X^i\right| \leq \sqrt{d}\|h(xX)\| < N$$

因为$h(x_0) = 0 \bmod N$,我们得出结论$h(x_0) = 0$。

引理指出，如果h是一个低范数多项式，则$h \bmod N$的所有小根也是h在整数上的根。引理表明，要找到$f(x) \bmod N$的一个小根x_0，我们需要寻找另一个与f模N有相同根的低范数多项式$h \in \mathbb{Z}[x]$，这样就能容易找到h在整数上的根x_0。为此，我们可以寻找一个多项式$g \in \mathbb{Z}[x]$，使得$h = gf$具有低范数，即范数小于N。这相当于寻找具有低范数多项式$f, xf, x^2 f \ldots, x^r f$的整数线性组合。不过，大多数情况下，并不存在具有足够小的范数的非平凡线性组合。

Coppersmith 找到了解决这个问题的窍门：如果$f(x_0) = 0 \bmod N$成立，那么对于任意k则有$f(x_0)^k = 0 \bmod N^k$。更一般地，定义以下多项式：

$$g_{u,v}(x) = N^{m-v} x^u f(x)^v$$

对于一些预定义的m，则x_0是$g_{u,v}(x)$模N^m的一个根，其中$u \geq 0$和$0 \leq v \leq m$。要使用引理 4，我们必须找到多项式$g_{u,v}(x)$的一个整数线性组合$h(x)$，使得$h(xX)$的范数小于N^m (回想一下X是满足$X \leq N^{1/d}$的x_0上界)。由于范数（是N^m而不是N）的松弛上界，我们可以证明，对于足够大的m，总是存在一个线性组合$h(x)$满足所要求的界。一旦$h(x)$被找到，引理 4 就意味着它有x_0作为整数的根，因此,可以很容易地找到x_0。

如何有效地找到$h(x)$还有待证明，要做到这一点，我们必须说明一些关于$\mathbb{Z}^w$格的基本事实。 设$u_1, \ldots, u_w \in \mathbb{Z}^w$是线性独立的向量。由$\langle u_1, \ldots, u_w \rangle$构成的(满秩)格$L$是$u_1, \ldots, u_w$的所有整数线性组合的集合。$L$的行列式定义为$w \times w$方阵的行列式，它的行列式是向量$u_1, \ldots, u_w$。

在我们的例子中，我们把多项式$g_{u,v}(xX)$看作向量，并研究了它们所构成的格L。设$v = 0, \ldots, m$，$u = 0, \ldots, d-1$，则格的维数$w = d(m+1)$。例如，当f是

二次一元多项式且$m = 3$时，得到的格由以下矩阵的行构成：

$$\begin{array}{c} \\ g_{0,0}(xX) \\ g_{1,0}(xX) \\ g_{0,1}(xX) \\ g_{1,1}(xX) \\ g_{0,2}(xX) \\ g_{1,2}(xX) \\ g_{0,3}(xX) \\ g_{1,3}(xX) \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & x & x^2 & x^3 & x^4 & x^5 & x^6 & x^7 \end{array} \\ \begin{bmatrix} N^3 & & & & & & & \\ & XN^3 & & & & & & \\ * & * & X^2N^2 & & & & & \\ & * & * & X^3N^2 & & & & \\ * & * & * & * & X^4N & & & \\ & * & * & * & * & X^5N & & \\ * & * & * & * & * & * & X^6 & \\ & * & * & * & * & * & * & X^7 \end{bmatrix} \end{array}$$

*元对应于我们忽略其值的多项式的系数，所有空元为零。由于矩阵是三角形的，它的行列式是对角线上元素的乘积(如上所示)，我们的目标是在这个格中找到短向量。

Hermite 的一个经典结论表明：任意维数为w的格L包含一个非零向量$v \in L$，它的L_2范数满足$\|v\| \leq \gamma_w \, det(L)^{1/w}$，其中$\gamma_w$是只依赖于$w$的常数。Hermite 的界可以用来证明，对于足够大的m，我们的格包含需求小于N^m的范数向量。问题是我们能否有效地在L中构造长度小于 Hermite 界的短向量。LLL 算法是一种有效的算法，恰好可以做到。

事实 5（LLL） 设L是由$\langle u_1, \ldots, u_w \rangle$所构成的格。当$\langle u_1, \ldots, u_w \rangle$作为输入时，LLL 算法输出一个向量$v \in L$满足：

$$\|v\| \leq 2^{w/4} \, det(L)^{1/w}$$

LLL 算法的运行时间是输入长度的四分之一。

LLL 算法（以其发明者 L. Lovasz、A. Lenstra 和 H. Lenstra Jr 的名字命名）在计算数论和密码学中有许多应用。它在 1982 年的发现为整数上多项式的因式分解提供了一种有效的算法，更广泛地说，为数环上的多项式的因式分解提供了一种有效的算法。LLL 算法经常被用来攻击各种密码系统，例如，许多基于“背包问

题"的密码系统都是使用 LLL 算法破解的。

利用 LLL 算法，我们可以完成 Coppersmith 定理的证明。为了保证 LLL 算法产生的向量满足引理 4 的界，我们需要满足：

$$2^{w/4}\, det(L)^{1/w} < N^m/\sqrt{w}$$

其中$w = d(m+1)$是L的维数。常规计算表明，对于足够大的m，能满足约束条件。实际上，当$X = N^{\frac{1}{d}-\epsilon}$时，取$m = O(k/d)$和$k = \min(\frac{1}{\epsilon}, \log N)$就足够了。因此，运行时间主要由在维数为$O(k)$的格上运行 LLL 算法所决定。

一个自然而然的问题，Coppersmith 定理能否应用于二元和多元多项式。如果$f(x,y) \in \mathbb{Z}_N[x,y]$有根$(x_0, y_0)$且有适当的界$|x_0 y_0|$，Marvin 能有效地找到$(x_0, y_0)$吗？尽管相同的技术似乎适用于某些二元多项式，但目前还是一个有待证明的开放性问题。随着越来越多的结果依赖于 Coppersmith 定理的二元扩张，所以严密的算法将会非常有用。

开放性问题 3 找出 Coppersmith 定理可以推广到二元多项式的一般条件。

4.2 Hastad 广播攻击

作为 Coppersmith 定理第一个应用，我们对由 Hastad 提出的旧攻击进行了改进。假设 Bob 希望将加密消息M发送给多方$P_1, P_2, \ldots, P_k$。每一方都有自己的 RSA 密钥$\langle N_i, e_i \rangle$。我们假定M比所有N_i都小。Bob 为了发送M，天真地使用每个公共密钥对其进行加密，并将第i^{th}个密文发送给P_i。攻击者 Marvin 可以窃听 Bob 对外的连接，并收集传输的k个密文。

为了简单起见，假设所有公钥指数e_i为 3。一个简单的论证表明，当$k \geq 3$时，Marvin 可以计算出M。实际上，Marvin 得到C_1, C_2, C_3，其中：

$$C_1 = M^3 \bmod N_1, \qquad C_2 = M^3 \bmod N_2, \qquad C_3 = M^3 \bmod N_3$$

对于所有的$i \neq j$，我们可以假设$\gcd(N_i, N_j) = 1$，否则 Marvin 可以因式分解

一些N_i。因此，将中国剩余定理(CRT)应用于C_1, C_2, C_3，给出的$C' \in \mathbb{Z}_{N_1 N_2 N_3}$满足$C' = M^3 \bmod N_1 N_2 N_3$。由于$M$小于所有的$N_i$，我们有$M^3 < N_1 N_2 N_3$，那么$C' = M^3$在整数上成立，因此，Marvin 可以通过计算$C'$的实数立方根来得到$M$。更一般的情况是，如果所有的公钥指数都等于$e$，则只要$k \geq e$，Marvin 就可以计算出$M$。不过这种攻击只有使用较小的$e$值时才是可行的。

Hastad 提出了一种更强的攻击方法。为了抵御 Hastad 的攻击，考虑一下对上述攻击做一下天真防御。Bob 可能在加密之前"填充"消息，而不是广播加密的M。例如，如果M是m位长的，Bob 可以将$M_i = i2^m + M$发送给P_i。由于 Marvin 获得了不同消息的加密，他无法发起攻击。然而，Hastad 证明了这种线性填充是不安全的，事实上，他证明了在加密之前对消息应用任何固定多项式都不能阻止攻击。

假设对于每个参与者$P_1, P_2, \ldots, P_k$，Bob 有一个固定的公用多项式$f_i \in \mathbb{Z}_{N_i}[x]$。为了广播消息$M$，Bob 将$f_i(M)$的加密发送给$P_i$。Marvin 通过窃听知道了$C_i = f_i(M)^{e_i} \bmod N_i$，其中$i = 1, \ldots, k$。Hastad 表明，如果有足够的参与方，Marvin 可以从所有的密文中计算出明文M。下面的定理是 Hastad 原始结论的一个更强的版本。

定理 6 (Hastad) 设$N_1, \ldots, N_k$是成对的相对素数，集合$N_{min} = \min_i(N_i)$。设$g_i \in \mathbb{Z}_{N_i}[x]$是$k$个$d$次多项式。假设存在唯一的$M < N_{min}$满足：

$$g_i(M) = 0 \bmod N_i \quad (i = 1, \ldots, k)$$

假设$k > d$，给定$\langle N_i, g_i \rangle_{i=1}^k$，我们可以有效地找到的$M$。

证明 令$\overline{N} = N_1 \cdots N_2$，我们假定所有的$g_i$都是一元的。(实际上，对于某些$i$，$g_i$的首项系数在$\mathbb{Z}_{N_i}^*$中是不可逆的，那么$N_i$的因式分解就会显现出来。通过将每个$g_i$乘以$x$的适当幂，假定它们都有$d$次。构造多项式：

$$g(x) = \sum_{i=1}^{k} T_i g_i(x)\text{，其中 } T_i = \begin{matrix} 1 \bmod N_j \text{ 若} i = j \\ 0 \bmod N_j \text{ 若} i = j \end{matrix}$$

其中T_i是整数，被称为中国剩余系数。那么$g(x)$一定是一元的，因为它首项模了所有的N_i，且次数为d。此外，我们还知道$g(M) = 0 \bmod \overline{N}$。定理 6 现在便可由定理 3 推导而来，因为$M < N_{min} < \overline{N}^{1/k} < \overline{N}^{1/d}$。

该定理表明，如果提供了足够多的方程，可以有效地求解以相对素数复合模的一元方程组。令$g_i = f_i^{e_i} - C_i$，我们可以知道，当参与方数至少为d时，Marvin 可以从给定的密文中计算出M，其中d是$e_i \deg(f_i)$在所有$i = 1, \ldots, k$上的最大值。特别地，如果所有的e_i都等于e，并且 Bob 发送线性相关的消息，那么 Marvin 只要$k > e$就可以算出明文。

Hastad 的原始定理比上述定理更弱。与d次多项式不同，Hastad 定理需要$d(d+1)/2$次多项式。Hastad 定理的证明类似于上一节中提到的 Coppersmith 定理证明。由于 Hastad 定理没有在格中使用g的幂，从而得到了一个较弱的界。

总结这一节，我们注意到，要正确地防御上述广播攻击，必须使用随机填充方法，而不是使用固定填充方法。

4.3 Franklin-Reiter 相关消息攻击

当 Bob 用相同的模数发送与 Alice 相关的加密消息时，Franklin 和 Reiter 发现了一种聪明的攻击。$\langle N, e \rangle$是爱丽丝的公钥，假设$M_1, M_2 \in Z_N^*$是两个不同的消息，对于某些已知的多项式$f \in Z_N[x]$，M_1, M_2满足$M_1 = f(M_2) \bmod -N$。为了将M_1和M_2发送给 Alice，Bob 可能会天真地对消息进行加密，并传输得到的密文C_1, C_2。我们通过证明可以知道，在给定C_1, C_2的情况下，Marvin 可以很容易地计算出M_1, M_2。虽然攻击对任意小e都有效，但为了简化证明，我们给出了$e = 3$的引理。

引理 7(FR) 令$e=3$，$\langle N,e\rangle$为 RSA 公钥。设$M_1 \neq M_2 \in Z_N^*$对于$b \neq 0$的线性多项式$f=ax+b \in Z_N[x]$满足$M_1 = f(M_2) \bmod N$。然后，给定$\langle N,e,C_1,C_2,f\rangle$，Marvin 可以在$\log N$的平方时间内计算出$M_1, M_2$。

证明 为了保证这部分证明的一般性，我们使用任意e来表示它（而不是限制为$e=3$）。由于$C_1 = M_1^e \bmod N$，我们知道M_2是多项式$g_1(x) = f(x)^e - C_1 \in Z_N[x]$的根。同样，$M_2$也是$g_2(x) = x^e - C_2 \in Z_N[x]$的根。线性因子$x - M_2$是两个多项式的除法。因此，Marvin 可以使用欧几里德算法来计算g_1和g_2的最大公约数(Greatest Common Divisor, GCD)。如果 GCD 是线性的，则可以找到M_2。GCD 可以在e和$\log N$的平方时间内算出。

我们证明了当$e=3$时，GCD 一定是线性的。多项式$x^3 - C_2$因子将p和q都模成一个线性因子和一个不可约二次因子(因为$\gcd(e,\varphi(N))=1$，所以$x^3 - C_2$在Z_N中只有一个根)。因为g_2不能整除g_1，所以 GCD 一定是线性的。对于$e>3$情况，GCD 几乎总是线性的。然而，对于一些罕见的M_1, M_2和f，有可能得到一个非线性的 GCD，在这种情况下攻击会失败。

对于$e>3$情况，攻击所需时间是e的平方时间。因此，只有在使用小的公钥指数e时才能应用这种攻击。对于大型电子计算机来说，计算 GCD 的工作令人望而却步。一个有趣的问题(尽管可能很难)，为任意的e设计这样的攻击，尤其是能否在$\log e$的多项式时间中找到上述g_1和g_2的 GCD?

4.4 Coppersmith 短填充攻击

Franklin-Reiter 的攻击可能看起来有点人为。毕竟，为什么 Bob 要给 Alice 发送相关消息的加密呢？Coppersmith 加强了攻击，并证明了一个关于填充攻击的重要的结论。

随机填充算法可以通过将一些随机位附加到其中一个端来填充明文M，但是以下攻击指出了这种简单填充的危险。假设 Bob 向 Alice 发送了正确填充的M加密。攻击者 Marvin 拦截密文并阻止其到达目的地。Bob 注意到 Alice 没有回复他的消息，并决定将M重新发送给 Alice。他随机填充M并传输生成的密文。Marvin 现在有两个密文，对应于使用两种不同随机填充对同一消息的两次加密。以下定理表明，虽然他不知道使用的填充算法，但 Marvin 仍能够算出明文。

定理 8 设$\langle N, e\rangle$为 RSA 公钥，其中N的长度为是n位。令集合$m = [n/e^2]$。设$M \in Z_N^*$是长度最长为$n - m$位的消息。定义$M_1 = 2^m\ M + r_1$和$M_2 = 2^m\ M + r_2$，其中r_1和r_2是分别为$0 \le r_1, r_2 \ge 2^m$的整数。如果 Marvin 知道了$\langle N, e\rangle$和M_1, M_2的加密C_1, C_2 (但不知道r_1或r_2)，则他可以有效地计算出 M。

证明 定义$g_1(x, y) = x^e - C_1$和$g_2(x, y) = x^e - C_2$，我们知道当$y = r_2 - r_1$时，这些多项式有相同的根M_1。换句话说，$\triangle = r_2 - r_1$是结式$h(y) = \mathrm{res}_x(g_1, g_2) \in \mathbb{Z}_N[y]$的根。$h$的次数最多是$e^2$。此外有$|\triangle| < 2^m < N^{1/e^2}$，因此$\triangle$是$h$模$N$的一个小根，而 Marvin 可以利用 Coppersmith 定理（定理 3）有效地求出这个根。一旦$\triangle$已知，便可以使用上一节的 Franklin-Reiter 攻击算出M_2，从而得到M。

当$e = 3$时，只要填充长度小于消息长度的$1/9^{th}$，就可以进行攻击。这是一个重要的结论。注意，对于建议值$e = 65537$，对于标准的模数大小来说，这种攻击是无用的。

4.5 部分密钥泄露攻击

设$\langle N, d\rangle$为 RSA 私钥，假设 Marvin 通过某种方式知道了d的一部分，比如说四分之一。他能得到d剩下的部分吗？当相应的公钥指数很小时，答案是肯定的，令人惊讶吧。最近，Boneh，Durfee 和 Frankel 证明了只要$e < \sqrt{N}$，就有可能从

它的一小部分位算出d的所有部分。可见结论说明了保护整个RSA私钥的重要性。

定理 9 (BDF) 设$\langle N, d\rangle$为 RSA 私钥，其中N长度为n位.。给定d的$\lceil d/4\rceil$最小有效位，Marvin 可以在$e\ \log_2 e$的线性时间算出d。

证明依赖于另一个完美精妙的 Coppersmith 定理。

定理 10 (Coppersmith) 设$N = PQ$是一个n位 RSA 模。然后，给定p的$n/4$最小有效位或p的$n/4$最有效位，可以有效地将N分解。

定理 9 很容易从定理 10 推理出来，事实上，根据e和d的定义，存在一个整数k，使得:

$$ed - k(N - p - q + 1) = 1$$

由于$d < \sqrt{N}$，我们必有$0 < k \leq e$。对方程模$2^{n/4}$进行约化，设$q = N/p$，得到:

$$(ed)p - kp(N - p + 1) + kN = p(\mathrm{mod}\ 2^{n/4})$$

由于 Marvin 知道了d的$n/4$最小有效位，他知道$ed \bmod 2^{n/4}$的值，因此，他得到了一个关于k和p的方程。对于k的每一个e的可能值，Marvin 求解p的二次方程，并能得到了$p \bmod 2^{n/4}$的一些候选值。对于这些候选值，他运用定理 10 尝试去分解N。可以证明$p \bmod 2^{n/4}$的候选值的总数最多为$e\ \log_2 e$，因此，在最多$e\ \log_2 e$次尝试之后，N将被分解。

定理 9 被称为部分密钥泄露攻击，对于更大的e值，只要$e < \sqrt{N}$，也存在类似的攻击，不过，要实现此种攻击的技术有点复杂。有趣的是，基于离散日志的密码系统，如 ELGamal 公钥系统，似乎不容易受到部分密钥泄漏攻击的影响。事实上，如果给出$g^x \bmod p$和x的常数部分，则没有已知的多项式时间算法来计算x的其余部分。

为了总结这一节，我们将证明当加密指数e很小时，RSA 系统会泄漏相应私

钥d一半的最高有效位。要了解这一点，再考虑一个方程$ed - k(N - p - q + 1) = 1$，其中$k$是$0 < k \leq e$的整数。给定$k$，Marvin 可以很容易地计算出：

$$\hat{d} = \lfloor (kN + 1)/e \rfloor$$

之后：

$$\left|\hat{d} - d\right| \leq \frac{k(p+q)}{e} \leq \frac{3k\sqrt{N}}{e} < 3\sqrt{N}$$

因此，$\hat{d}$是d的很好的近似值。该界表明，对于大多数d，$\hat{d}$中一半的最高有效位与d相同。由于k只有e个可能的值，因此 Marvin 可以构造一个大小e的小集合，使得集合中的一个元素等于d的一半最高有效位的。$e = 3$的情况特别有趣，在这种情况下，可以知道$k = 2$，系统完全泄漏了d的一半最高有效位。

5 执行攻击

我们将注意力转向另一类完全不同的攻击。这些攻击不是攻击 RSA 函数的底层结构，而是专注于 RSA 的实现。

5.1 时序攻击

想一下存储 RSA 私钥的智能卡，由于卡是防篡改的，攻击者 Marvin 可能无法审阅其内容并使其泄露出密钥。然而，Kocher 的一个巧妙攻击表明，通过精确测量智能卡执行 RSA 解密（或签名）所需的时间，可以快速发现私有解密指数d。

我们将解释如何使用“重复平方算法”对一个简单的 RSA 实现进行攻击。设$d = d_n d_{n-1} \ldots d_0$是$d$的二进制表示（即$d = \sum_{i=0}^{n} 2^i d_i$，其中$d_i \in \{0,1\}$）。基于$C = \prod_{i=0}^{n} M^{2^i d_i} \bmod N$的观察基础，我们可以知道用重复平方算法来计算$C = M^d \bmod N$最多使用$2n$次模乘，算法是如下工作的：

令z等于M，C等于 1，对于$i = 0, \ldots, n$，执行以下步骤：

（1）如果$d_i = 1$，令C等于$C \cdot z \bmod N$，

（2）令z等于$z^2 \bmod N$。

最后，C有值为$M^d \bmod N$。

当$i = 0, \ldots, n$时，变量z遍历$M^{2^i} \bmod N$值的集合，变量C在集合中“收集”适当幂以获得$M^d \bmod N$。

为了发起攻击，Marvin 要求智能卡在大量随机消息$M_1, \ldots, M_k \in \mathbb{Z}_N^*$上生成签名，并测量每个签名生成所需的时间$T_i$。

攻击从最低有效位开始一次一个地算出d的比特位。我们知道d是奇数，因此$d_0 = 1$。考虑第二次迭代。最初$z = M^2 \bmod N$且$C = M$。如果$d_1 = 1$，则智能卡会计算乘积$C \cdot z = M \cdot M_i^2 \bmod N$，否则，它是不会计算的。设$t_i$是智能卡计算$M_i \cdot M_i^2 \bmod N$所花费的时间。由于计算$M_i \cdot M_i^2 \bmod N$的时间取决于$M_i$的值，因此$t_i$彼此不同（简单模约化算法需要不同的时间，取决于所减少的值）。一旦 Marvin 获得智能卡的物理规格，之后他便会测量得到t_i（在发起攻击之前）。

Kocher 观察到当$d_1 = 1$时，两个集合$\{t_i\}$和$\{T_i\}$是相关的。例如，如果，对于某些i，t_i比预期的要大得多，那么T_i也可能大于预期。另一方面，如果$d_1 = 0$，则两个集合$\{t_i\}$和$\{T_i\}$表现为独立的随机变量。通过测量相关性，Marvin 可以确定d_1是 0 还是 1。继续使用这个方法，他可以很快得到d_2，d_3。注意，当使用低公钥指数e时，上一节的部分密钥泄露攻击表明，使用 Kocher 的时序攻击，只需要知道d的四分之一的位就行。

有两种方法可以抵御攻击。最简单的是添加适当的延迟，以使模幂运算总是要花费一定的时间。第二种方法是由 Rivest 提出的基于盲化的方法。在解密 M 之前，智能卡选择一个随机的$r \in \mathbb{Z}_N^*$并计算$M' = M \cdot r^e \bmod N$，然后将d应用于M'上并获得$C' = (M')^d \bmod N$，最后，令$C = C'/r \ mod \ N$。通过这种方法，将d

应用于 Marvin 不知道的随机消息M'上，这样的话，Marvin 就不能发起攻击了。

Kocher 最近在这些线路上发现了另一种叫做功率密码分析的攻击。 Kocher 表明，通过在签名生成过程中精确测量智能卡的功耗，Marvin 通常可以轻松发现密钥。事实证明，在多精度乘法期间，卡的功耗高于正常值。通过测量高消耗周期的长度，Marvin 可以很容易地确定在给定的迭代中卡是执行一次还是两次乘法，从而暴露出d的比特位。

Kocher 最近发现了另一种类似的攻击，称为能量分析攻击。Kocher 指出通过精确测量智能卡在签名生成过程中的功耗，Marvin 通常可以很容易地得到秘密密钥。结果表明，在多精度乘法过程中卡的功耗会高于正常值，通过测量高消耗周期的长度，Marvin 可以很容易地确定在给定的迭代中卡是否执行一次或两次乘法，从而得到d的比特位。

5.2 随机故障

RSA 的解密和签名的实现经常使用中国剩余定理来加速$M^d \bmod N$的计算，签名者 Bob 为了替换模N的工作，先计算签名模p和q的结果，然后利用中国剩余定理将结果结合起来。更准确地说，Bob 首先计算：

$$C_p = M^{d_p} \bmod p \text{和} C_q = M^{d_q} \bmod q$$

其中$d_p = d \bmod (p-1)$和$d_q = d \bmod (q-1)$。然后，他得到签名C通过令：

$$C = T_1 C_p + T_2 C_q \pmod N$$

其中：

$$T_1 = \left\{ \begin{matrix} 1 \bmod p \\ 0 \bmod q \end{matrix} \right\} \text{和} T_2 = \left\{ \begin{matrix} 1 \bmod p \\ 0 \bmod q \end{matrix} \right\}$$

与d_p和d_q两个指数相比，CRT 最后一步的运行时间可以忽略不计。注意p和q是N的一半长，然后由于乘法的简单实现需要平方时间，所以模p的乘法速度是模N的 4 倍，而且，d_p是d的一半长，计算$M^{d_p} \bmod p$的速度是计算$M^d \bmod N$的

8 倍，因此，整个签名时间减少了四倍，许多实现都使用这种方法来提高性能。

Boneh，DeMillo 和 Lipton 观察到使用 CRT 方法有内在的危险。假设在生成签名时，Bob 的计算机上的一个小故障导致它在一条指令中错误计算。例如，在将寄存器中的值从一个位置复制到另一个位置时，其中一个比特位被翻转了。(故障可能是由环境电磁干扰引起的，也可能是由于罕见的硬件缺陷造成的，比如早期版本的奔腾芯片。) Marvin 得到了无效的签名给定之后可以很容易地对 Bob 的模数N进行分解。

正如 A. K. Lenstra 所说，我们发现了一个新的攻击。假设在 Bob 生成签名时发生单个错误，那么，C_p或C_q中将有一个被错误地计算。如果说C_p是正确的，那么$\hat{C}_q$就会不正确，得到的签名为$\hat{C} = T_1 C_p + T_2 \hat{C}_q$。一旦 Marvin 获取到了$\hat{C}$，通过计算$\hat{C}^e \neq M \bmod N$，他就知道这是一个错误的签名。然而注意到：

$$\hat{C}^e = M \bmod p \text{ 当 } \hat{C}^e \neq M \bmod q$$

因此，$\gcd(N, \hat{C}^e - M)$便是N的一个非平凡因子。

要使攻击奏效，Marvin 必须对M有充分的了解。也就是说，我们假设 Bob 不使用任何随机填充方法，签名前的随机填充可以防御此种攻击，对于 Bob 来说，一个更简单的防御方法是在将生成的签名发送给全世界之前检查生成的签名。当使用 CRT 加速方法时，检查是特别重要的。随机故障攻击对许多密码系统都是有害的，许多系统，包括 RSA 的非 CRT 实现，都可以使用随机故障进行攻击。不过，这些结论更多的是理论性的。

5.3 Bleichenbacher 对 PKCS 1 的攻击

设N是n位 RSA 模，M是m位消息，其中$m < n$。在应用 RSA 加密之前，一般会通过向其添加随机位，将消息M填充到n位。公钥加密标准 1（Public Key

Cryptography Standard 1, PKCS 1）的旧版标准就是使用的这种方法。填充后，消息如下所示：

02	随机位	00	M

生成的消息长度为n位，并直接使用 RSA 加密。包含“02”的初始块长度为 16 位，从上图可看出已在消息中添加了随机填充。

当运行在 Bob 的机器上应用程序(例如，Web 浏览器)接收到消息时，会对其进行解密，检查初始块，并去掉随机填充。但是，一些应用程序会检查“02”初始块，如果不存在，就会返回一条错误消息，说明“无效的密文”。Bleichenbacher 表示这个错误消息可能导致灾难性的后果：攻击者 Marvin 可以使用错误消息解密他所选择的密文。

假设 Marvin 截获了一个针对 Bob 的密文C，并希望对其进行解密。为了发动攻击，Marvin 随机挑选了一个$r \in \mathbb{Z}_N^*$，计算$C' = rC \bmod N$，并将C'发送到 Bob 的机器。运行在 Bob 的机器上的应用程序接收C'并尝试解密它。它要么用错误消息进行响应，要么根本不响应(如果C'的格式正确的话)。因此，Marvin 知道C'解密过程中 16 位最高有效位是否等于 02。实际上，Marvin 有这样的谕言，对于他选择的任何r，他都可以测试$rC \bmod N$解密的 16 位最高有效位是否等于 02。Bleichenbacher 证明了这样的谕言足以解密C。

6 结论

对 RSA 系统进行了 20 年的研究以来，产生了一些有见地的攻击，但还没有发现破坏性的攻击。到目前为止发现的攻击主要说明了在实现 RSA 时需要避免的陷阱，目前看来，可以信任正确的 RSA 密码系统实施来提供数字世界中的安全性。我们将针对 RSA 的攻击分为四类：(1)利用系统公然误用的基本攻击；(2)低私钥指数攻击，此种攻击非常严重，绝不能使用低私钥指数；(3)低公钥指数攻

击；(4)对 RSA 系统执行时的攻击。这些持续不断的攻击说明，我们对基本数学结构的研究还是不够的。另外，Desmedt 和 Odlyzko、Joye 和 Quisquater 以及 DeJonge 和 Chaum 还提出了一些额外的攻击。在整篇论文中，我们观察到通过在加密或签名之前正确填充消息可以防御许多攻击。

致谢

我要感谢 Susan Landau 鼓励我撰写调查问卷，感谢 Tony Knapp 帮忙编辑手稿。我还要感谢在早些时候 Mihir Bellare、Igor Shparlinski 和 R. Venkatesan 对草案发表的评论。