第 3 章　可证明安全性理论与方法

启发式设计方法存在以下问题：新的分析技术的提出时间是不确定的，在任何时候都有可能提出新的分析技术；这种做法使人们很难确信协议的安全性，反反复复地修补更增加了人们对安全性的担心，也增大了实现代价或成本。

那么有什么解决办法呢？可证明安全性理论与方法就是为解决上述问题而提出的一种解决方案（当然并非是唯一解决方案）。

实际上，可证明安全性理论与方法是在一定的敌手模型下证明了安全方案或协议能够达到特定的安全目标，因此，合适的安全目标定义、适当的敌手模型是我们讨论可证明安全性的前提条件。

可证明安全性理论与方法的应用价值是显而易见的，一方面，可以把主要精力集中在“极微本原”的研究上，这是一种古老的、基础性的、带有艺术色彩的研究工作；另一方面，如果你相信极微本原的安全性，不必进一步分析协议即可相信其安全性。

必须说明的是，可证明安全性理论与方法也有局限性。首先必须注意模型规划，即注意所建模型都涵盖了哪些攻击，显然一些基于物理手段的攻击都不包含在内，但这并不意味着可证明安全性的方案就一定不能抵抗这类攻击，而是说未证明可以抵抗这类攻击；其次即使应用具有可证明安全性的方案，也可能有多种方式破坏安全性：有时证明了安全性，但问题可能是错误的，也可能应用了错误的模型或者协议被错误操作，甚至软件本身可能有错误。

另一需要注意的问题是基础假设的选取。可证明安全性是以某一假设为基础的，因此一旦该假设靠不住，安全性证明也就没有意义（当然不一定意味着可构造对方案的攻击实例）；选取基础假设的原则就是“越弱越好”，通常称弱假设为标准假设。基础假设的强弱是比较不同安全方案的重要尺度之一。

上述表述较为抽象，下面以 RSA 为例加以说明。

给定某个基于 RSA 的协议 P，如果设计者或分析者给出了从 RSA 单向函数到 P 安全性的归约，那么 P 具有以下转换性质：对于任何声称破译 P 的敌手（程序）A，以 A 为“转换算法”的输入，必然导致一个协议 Q，Q 可被证明破译 RSA。结论是：只要你不相信 RSA 是可破译的，那么上述的 Q 就不存在，因而 P 是安全的。

3.1　基本概念与计算假设

3.1.1　基本概念

对可证明安全性的精确形式化有多种形式，一般是在计算复杂性理论框架下讨论，如主要考虑“概率多项式时间（PPT）”敌手 A 和转换算法，以及“可忽略”的成功概率。这是一种“渐近”观点，有着广泛的适用范围。详细内容可参阅文献[1]。概率多项式时间算法实质上

是指有随机输入、并在其输入长度的多项式时间内一定有输出结果的算法。下面先给出可忽略函数的精确描述。

定义 3.1(可忽略函数) 设 $\mu(n):N\to R$,称函数 $\mu(n)$是可忽略的,如果对任意多项式 $p(\cdot)$,对于足够大的 n,满足 $\mu(n)<1/p(n)$。

定义 3.2(计算不可区分) 设$\{X_n\}$和$\{Y_n\}$是两个概率空间,称它们是(多项式时间)计算不可区分的,表示为$\{X_n\}\overset{c}{\approx}\{Y_n\}$或$\{X_n\}\approx\{Y_n\}$,如果对任意多项式 $p(\cdot)$,任意 PPT 算法 D 及所有辅助输入 $z\in\{0,1\}^{poly(n)}$,满足

$$|\Pr[D(X_n,1^n,z)=1]-\Pr[D(Y_n,1^n,z)=1]|<1/p(n)$$

定义 3.2 是说,不存在明显可区分概率空间$\{X_n\}$和$\{Y_n\}$的 PPT 算法,即在多项式时间内,任何 PPT 区分算法的成功概率总是可忽略的。

本质上,可证明安全性理论与方法的主要研究途径是规划安全方案或协议的形式化安全模型,不同的安全方案或协议会导致不同的安全模型,而这些安全模型大多都基于一些很基本的密码学概念。因此,对一些最基本的密码学概念(如加密、签名及其安全性定义等)给予精确的形式化定义,是可证明安全性理论与方法的基础组成部分,有助于消除自然语言的语义二义性。

定义 3.3(数字签名方案) 一个数字签名方案由以下 3 个算法组成。

(1) 密钥生成算法 K。对于输入 1^k,K 产生一对匹配值(k_p,k_s),分别称为公钥和私钥,K 可以是概率算法。k 称为安全参数,密钥等因素的规模都依赖于 k。

(2) 签名算法 Σ。给定消息 m 和(k_p,k_s),Σ 产生签名 σ,Σ 可以是概率算法。

(3) 验证算法 V。给定签名 σ、消息 m 和公钥 k_p,V 测试 σ 是否是 m 的对应公钥 k_p 的合法签名,通常情况下 V 是确定性算法,输出 1(表示签名有效)或 0(表示签名无效)。

对于任一数字签名方案(K,Σ,V),敌手 A 的模型如下。

A 的目标有以下 3 个:揭示签名者私钥(完全破译);构造成功率高的伪签名算法(通用伪造);提供一个新的消息-签名对(存在性伪造)。

存在性伪造一般并不危及安全,因为输出消息很可能无意义,但这样的方案本身不能确保签名方的身份,如不能用来确认伪随机元素(如密钥),也不能用来支持非否认。

A 的两类攻击:未知消息攻击和已知消息攻击。后一种情况中最强的攻击是"适应性选择消息攻击",即 A 可以向签名方询问对任何消息的签名(当然不能询问欲伪造消息的签名,这是一类自明的约定,后不注),因而可能根据以前的答复适应性地修改随后的询问。

定义 3.4(数字签名方案的抗适应性选择消息攻击安全性) 对任一数字签名方案(K,Σ,V),如果敌手 A 的攻击成功概率

$$\mathrm{Succ}_A=\Pr[(k_p,k_s)\leftarrow K(1^k),(m,\sigma)\leftarrow A^{\Sigma_{k_s}}(k_p):V(k_p,m,\sigma)=1]$$

是可忽略的,则称该方案能够抵抗适应性选择消息攻击。这里 A 可以获得签名 $\mathrm{Oracle}\Sigma_{k_s}$(实际上是一个"黑盒"),这模拟了以上所说的"适应性选择消息询问"。

定义 3.5(公钥加密方案) 一个公钥加密方案由以下 3 个算法组成。

(1) 密钥生成算法 K。对于输入 1^k,K 产生一对匹配值(k_p,k_s),分别称为公钥和私

钥，K 可以是概率算法。

(2) 加密算法 E。给定消息 m 和公钥 k_p，E 产生 m 对应的密文 c。E 可以是概率算法，这时记为 $E(k_p,m;r)$，r 表示随机输入。

(3) 解密算法 D。给定密文 c 和私钥 k_s，D 产生 c 对应的明文 m，D 一般是确定性算法。

一般而言，公钥加密方案的安全目标是单向性(One Wayness，OW)，即在不知道私钥的情况下，敌手 A 在概率空间 $M\times\Omega$ 上成功地对 E 求逆的概率是可忽略的(这里 M 是消息空间，Ω 是公钥加密方案的随机掷硬币空间)，亦即概率

$$\mathrm{Succ}_A = \Pr[(k_p,k_s)\leftarrow K(1^k):A(k_p,E(k_p,m;r))=m]$$

是可忽略的。

然而，许多应用要求更强的安全性。

定义 3.6(多项式安全/密文不可区分) 对任一公钥加密方案(K,E,D)，如果满足

$$\begin{aligned}\mathrm{Adv}_A = {} & 2\times\Pr[(k_p,k_s)\leftarrow K(1^k),(m_0,m_1,s)\leftarrow A_1(k_p),c\\ & = E(k_p,m_b;r):A_2(m_0,m_1,s,c)=b]-1\end{aligned}$$

是可忽略的，则称该方案是多项式安全的或密文不可区分的，这里敌手 $A=(A_1,A_2)$ 是一个 2 阶段攻击者(都是 PPT 算法)，概率取于(b,r)之上。

定义 3.6 形式化了以下性质：敌手了解明文某些信息(可任选一对消息，其中一个被加密)，但它不能从密文得到除明文长度之外的任何信息。

另一个更新的安全概念是“非延伸性”(Non-Malleability)，即敌手得到一个密文，敌手不能或以可忽略概率生成一个新的密文使得两个明文意义相关(如 DES 的互补明文结构)。该概念强于抗选择明文攻击的多项式安全性，但和抗选择密文攻击的多项式安全性是等价的。

敌手的几种攻击类型(相当于敌手拥有的 Oracle 数量及性质)如下。

(1) CPA(选择明文攻击)，该攻击在公钥加密方案中显然是平凡的。

(2) PCA(明文校验攻击)，敌手获得明文校验 Oracle，用以回答关于任一输入对(m,c)是否是对应明密文对的询问。

(3) CCA(选择密文攻击)，除获得加密 Oracle 外，敌手还获得解密 Oracle，即对于任何询问的密文(除了应答密文)，Oracle 都给予相应的明文作为回答。这是最强的攻击(根据是否适应性选择密文，还可以细分为 CCA1 和 CCA2)。

对应以上攻击条件的相应安全性定义，均可用类似于定义 3.6 的方法给出，区别仅在于敌手获得的 Oracle 数量和性质不一样。对称密码方案的安全性可类似定义。

3.1.2 计算假设

许多安全概念并不能在无条件的情况下得到保证，因此，安全性一般依赖于以下计算假设：单向函数的存在性、或者单向置换的存在性、或者是陷门单向函数(置换)的存在性。单向函数是一个满足以下条件的函数 f：任何人容易计算函数值，但是给定 $y=f(x)$，恢复 x(或 y 的任何原像)在计算上是不可行的。单向置换是一个双射的单向函数。对于加密处理来说，希望只有接收者才可以求逆，于是陷门单向置换是一个特殊的单向置换，其秘密信

息(即陷门)有助于对函数进行求逆。

给定计算假设“在没有陷门信息的情况下,计算函数的逆是不可行的”,希望不需要额外的假设即可得到安全性。形式上证明这一事实的唯一方法是证明“攻击安全方案或协议的敌手可以用于构造一个算法,该算法能够求解基础计算假设”。

在计算假设之间存在一个偏序关系:如果问题 P 比问题 P' 更困难(P' 归约到 P),那么问题 P 的困难性假设就比问题 P' 的困难性假设弱。所需要的假设越弱,安全方案或协议的安全性就越高。

目前主要使用以下两类计算假设。

(1) 整数分解与 RSA 问题。

(2) 离散对数与 Diffie-Hellman 问题。第一个使用的群是 $\boldsymbol{Z}_p^*$ 的循环子群,现在也常常使用基于椭圆曲线的循环子群。

1. 整数分解与 RSA 问题

最著名的困难问题是整数分解问题,即把两个素数 p 和 q 相乘得到 $n=p\cdot q$ 是容易的,而把合数 n 分解为素因子 p 和 q 则不是一件容易的事情。目前,最有效的整数分解算法是数域筛法(NFS)。数域筛法是超多项式的、亚指数算法,其复杂度是

$$O(\exp((1.923+o(1))(\ln n)^{1/3}(\ln\ln n)^{2/3}))$$

该方法在 1999 年 8 月创造了一个记录,它把一个 155 位的整数(512b)分解为两个 78 位的素数之积。所分解的数称为 RSA-155,来自于“RSA 挑战列表”,用于衡量 RSA 密码体制的安全性,该体制在 SSL 握手协议的软件和硬件实现中广泛使用。

整数乘法只是提供了一个单向函数,没有任何可能来对其求逆。现在不知道如何使得整数分解更容易一些。但是,有一些代数结构是基于整数 n 的分解,其中某些计算在不知道 n 的分解的情况下是困难的,在知道 n 的分解的情况下很容易。如对有限环 $\boldsymbol{Z}_n$,如果 $n=p\cdot q$,则同构于 $\boldsymbol{Z}_p\times\boldsymbol{Z}_q$。

例如,对任何元素 x,计算其 e 次幂是容易的。但是,要计算 e 次根,看起来需要知道满足 $ed\equiv 1 \bmod \varphi(n)$ 的整数 d。这里 $\varphi(n)$ 是 Euler 函数,对于 $n=p\cdot q$ 这种特殊情形,$\varphi(n)=(p-1)(q-1)$。因此,$ed-1$ 是 $\varphi(n)$ 的倍数,故等价于 n 的分解。

1978 年,Rivest、Shamir 和 Adleman 定义了著名的 RSA 问题,并设计了首个公钥密码体制——RSA。

RSA 问题:设 $n=p\cdot q$ 是两个相同规模的大素数的乘积,e 是与 $\varphi(n)$ 互素的整数。

对给定的 $y\in\boldsymbol{Z}_n^*$,计算 y 的模 e 次根 x,即满足 $x^e=y \bmod n$ 的 $x\in\boldsymbol{Z}_n^*$。

设 $n=\prod p_i^{v_i}$,则 Euler 函数容易用下式计算,即

$$\varphi(n)=n\times\prod\left(1-\frac{1}{p_i}\right)$$

因此,利用 n 的分解(陷门),RSA 问题很容易求解。但是没有人知道是否必须利用 n 的分解来求解 RSA 问题,更不知道如何在不知道 n 的分解的情况下来求解 RSA 问题。

RSA 假设:对任何两个足够大的素数的乘积 $n=p\cdot q$,RSA 问题是难解的(可能与分解

n 一样困难)。

2. 离散对数与 Diffie-Hellman 问题

设($\mathbf{G}$,·)表示一个阶为 q 的循环群,其中 q 是素数。令 g 表示 $\mathbf{G}$ 的一个生成元,即 $\mathbf{G}=<g>$。与离散对数相关的困难问题,以简洁的形式描述如下:

离散对数(DL)问题:给定 $y\in\mathbf{G}$,计算 $x\in\boldsymbol{Z}_q^*$,使得 $y=g^x$,记为 $x=\log_g y$。

计算 Diffie-Hellman(CDH)问题:对于任意的整数 $a,b\in\boldsymbol{Z}_q^*$,给定$<g,g^a,g^b>$,计算 g^{ab}。

判定性 Diffie-Hellman(DDH)问题:对于任意的整数 $a,b,c\in\boldsymbol{Z}_q^*$,给定$<g,g^a,g^b,g^c>$,判定是否有 $c\equiv ab \bmod q$ 成立。

上述问题显然是以从强到弱的顺序进行排列的,即 DL≥CDH≥DDH,其中 $A\geqslant B$ 表示问题 A 至少与问题 B 一样困难。然而,在实际中,没有人知道如何求解其中的任何一个问题,除非可以破解 DL 问题本身。而且,这些问题都是随机自归约的,即任何实例都可以归约到一个均匀分布的实例。例如,对于一个给定的元素 y,想要计算它对于基底 g 的离散对数 x。可以选择一个随机的 $t\in\boldsymbol{Z}_q$,计算 $z=ty$。那么 z 就是群中均匀分布的元素,而根据离散对数 $\alpha=\log_g z$ 就可以计算出 $x=\alpha/t$。因此,它们只有平均复杂情形,如果能够在多项式时间内求解不可忽略的部分实例,那么就可以在期望多项式时间内求解任何实例。

目前,求解离散对数问题最有效的算法依赖于其基础群。对于一般性的群(没有特定的代数性质可以应用),算法复杂度是 q 的平方根。但是,对于 $\boldsymbol{Z}_p^*$ 的子群,存在一个更好的方法。最好的算法是基于数域上的筛法,正如因子分解问题一样。一般数域筛法的复杂度是超多项式且亚指数的:

$$\mathcal{O}(\exp((1.923+o(1))(\ln p)^{1/3}(\ln\ln p)^{2/3}))$$

在 2001 年 4 月,该算法创造了一个最新纪录,它对 120 位的素数 p 计算出了 $\boldsymbol{Z}_p^*$ 中的离散对数。因此,只要一般攻击不能应用,而且其生成元的阶至少为 160b,512b 的素数仍然是足够安全的。

对于签名方案只要求群的离散对数问题困难即可,而对于加密方案需要陷门,因而要求群中的某些 DH 问题也是难以求解的。

3.2 随机预言模型方法论

20 世纪 80 年代初,Goldwasser、Micali 和 Rivest 等人首先比较系统地阐述了可证明安全性这一思想,并给出了具有可证明安全性的加密和签名方案[2,3]。但不幸的是,这些方案的可证明安全性是以严重牺牲效率为代价的,因此这些方案虽然在理论上具有重要意义,却完全不实用,这种情况严重制约了这一方向的发展。直到 20 世纪 90 年代中期出现了“面向实际的可证明安全性(Practice-Oriented Provable-Security)”概念,特别是 Bellare 和 Rogaway 提出了著名的随机预言(Random Oracle,RO)模型方法论[4],才使得情况大为改观:过去仅作为纯粹理论研究的可证明安全性理论与方法,迅速在实际应用领域取得重大

进展，一大批快捷、有效的安全方案相继提出；同时还产生了另一个重要概念“具体安全性”(Concrete Security or Exact Security)，其意义在于，人们不再仅仅满足于知道安全性的渐近度量，而是可以确切地了解较准确的安全度量。面向实际的可证明安全性理论与方法取得了巨大的成功，已为学术界广泛接受；但 Canetti 和 Goldreich 等对此持有异议[5]，并坚持仍在标准模型(Standard Model)中考虑安全性。

Canetti 和 Goldreich 认为，密码方案在 RO 模型中的安全性和通过“Hash 函数实现”的安全性之间无必然的因果关系。具体说来，存在这样的实际签名方案和加密方案，它们在 RO 模型中是安全的，但任何具体实现都是不安全的。这实际上是提出了一个反例。不过 Goldreich 也认为，应该明确 RO 模型方法论并不能作为实际方案安全的绝对证据，但该方法论仍是有意义的，如可以作为一种基本测试，任何实际方案通过这种安全测试是必要的，RO 模型方法论至少可以排除很多不安全设计，虽然并非完备的(即有些不安全设计可能排除不了)。Canetti 则进一步指出，RO 模型方法论虽然有以上缺点，但它可用于设计简单有效的协议，可以抵抗许多未知攻击；更重要的是，其基本思想可以用来设计某些安全的理想系统。

Pointcheval 等人则认为[6]，目前还没有人能提出令人信服的关于 RO 模型实际合法性的反例。文献[5]中的反例仅仅是一种理论上的反例，是针对实际目的的“明显错误设计”；RO 模型已经被广泛接受，并被认为是度量实际安全级别的一种很好的手段；即使并未提供一个正规的安全性证明(像标准模型那样)，但在其“安全性论断”(Hash 函数没有弱点)下，RO 模型中的证明确保了整个方案的安全性。更正规些，RO 模型可视为对敌手能力的某种限制，敌手的攻击是不考虑任何特殊 Hash 函数实例的一般攻击，而且如果假定存在某些防串扰设备(如 Smart Cards)，则 RO 模型等价于标准模型，这时只要求伪随机函数存在[1]。最重要的是，仅就实现效率这一点，RO 模型中的可证明安全性方案就远远优于那些提供标准安全性证明的安全方案，即仅此一点就可以从实际应用中排除当前所有“在标准模型中具有可证明安全性”的方案。事实上一些有代表性的有效标准解决方案，如文献[2]、[3]中的方案，过于复杂且代价昂贵，归约的复杂性使得难以确定实际安全参数，其有效性也只是相对过去的标准方案而言。

但可以肯定的是，迄今为止，RO 模型方法论是可证明安全性理论与方法最成功的实际应用，其现状是：几乎所有国际安全标准体系都要求提供至少在 RO 模型中可证明的安全性设计，而当前可证明安全性的方案也大都基于 RO 模型。

3.2.1 RO 模型介绍

文献[4]中提出以下观点：假定各方共同拥有一个公开的 RO，就在密码理论和应用之间架起了一座“桥梁”。具体办法是，设计一个协议 P 时，首先在 RO 模型(可看成一个理想模拟环境)中证明 P^R 的正确性，然后在实际方案中用“适当选择”的函数 h 取代该 Oracle(潜在论断是理想模拟环境和现实环境在敌手看来是多项式时间计算不可区分的)。一般来说，这样设计出来的协议可以和当前协议的实现效率相当。

必须指出，这并非是严格意义上的可证明安全性，因为安全性证明仅在 RO 模型中成

立，随后的“取代”过程本质上是一种推测：RO 模型中的安全特性可以在标准模型中得以保持。

假设提出一个协议问题 Π（这个问题和 h 函数“独立”），要设计一个安全协议 P 解决该问题，可按以下步骤执行。

(1) 建立 Π 在 RO 模型中的形式定义，RO 模型中各方（包括敌手）共享随机 Oracle R。

(2) 在 RO 模型中设计一个解决问题 Π 的有效协议 P。

(3) 证明 P 满足 Π 的定义。

(4) 实际应用中用函数 h 取代 R。

严格来讲，h 不可能真的像随机函数：首先其描述较短；其次，所谓的随机 Oracle 即 Hash 函数对每一个新的询问产生一随机值作为回答（如果问相同的询问 2 次，回答仍相同），这也是和随机函数的一个微小区别。但这并未改变上述方法论的成功，因为只要求在敌手看来像随机函数。此外，h 函数“独立”于 Π 也是至关重要的（否则可能不安全，可构造反例）。

一般来说，函数 h 至少要满足以下基本要求：设计上足够保守，能够抵抗各种已知攻击；不会暴露某些相关数学“结构”。文献[4]指出，选择 h 并不需要太麻烦，一个适当选择（但并不需过分苛求）的 Hash 函数就是以上 h 函数的一个很好选择。尽管 SHA-1 本身不是一个好的选择，但只需截短其输出或用某种非标准方式使用，如 $h(x)=\text{SHA-1}(xx)$。

RO 方法论也易于推广到基于对称密码本原的协议/方案研究，如 CBC-MAC，虽然没有 Hash 函数，但把一个恰当选择的分组密码（如 AES）视为随机函数。

3.2.2 归约论断和具体安全性

归约论断是可证明安全性理论的最基本工具或推理方法，简单地说就是把一个复杂的协议安全性问题归结为某一个或几个难题（如大整数分解或离散对数等）。在 RO 模型中的归约论断一般表现为：首先形式化定义方案的安全性，假设 PPT 敌手能够以不可忽略概率破坏协议安全性（如伪造签名）；然后模仿者 S（就是设计者或分析者）为敌手提供一个与实际环境不可区分的模拟环境（RO 模型），回答敌手的所有 Oracle 询问（模拟敌手能得到的所有攻击条件）；最后利用敌手的攻击结果（如一个存在性伪造签名）设法解决基础难题。如果把 RO 模型换成现实模型，就得到标准安全性证明。

RO 归约论断的一个显著优点是能够提供具体安全性结果。具体地说，就是试图显式地得到安全性的数量特征，这一过程称为“具体安全性处理”（Concrete or Exact Treatment of Security），与前面提到的“渐近”观点有明显区别。其处理结果一般表述为以下形式（举例）：“如果 DES（本原）可以抵抗这样条件的攻击，即敌手至多获得 2^{36} 个明密文对，那么该协议可以抵抗一个能执行 t 步操作的敌手发动的攻击，t 值如下……。”这样，协议设计者就能够确切地知道具体获得了多少安全保证，不必再笼统地说协议是否安全。

例 3.1 文献[8]中研究了 CBC-MAC 的安全特征，结论是：对任意一个运行时间至多为 t、至多见过 q 个正确 MAC 值的敌手，成功模仿一个新消息的 MAC 值的概率至多为 $\varepsilon+(3q^2n^2+l)/2^l$，这里 l 是基础密码的分组长度，n 是明文消息总数，ε 是检测到密码偏离随机

行为的概率(在$\mathcal{O}(nql)$时间内)。

具体安全性处理的一个重要目标就是,在把一个基础极微本原转化成相应协议时,尽可能多地保持极微本原的强度。这表现为要求"紧"的归约方法,因为一个"松"的归约意味着要求采用更长的安全参数,从而降低了效率。

3.2.3 RO 模型下安全的公钥加密和数字签名方案

1. 公钥加密方案

3.1 节中的公钥加密方案的概念可直接推广到 RO 模型中,从而得到 RO 模型中的公钥加密方案的定义。公钥加密方案可通过 PPT 生成器 g 规定:以安全参数 1^k 为输入,输出一对概率算法(E,D),分别称为加密算法和解密算法,D 保密,运行时间以 g 的运行时间为界。加密过程为:$y \leftarrow E^R(x)$;解密过程为:$x \leftarrow D^R(y)$。

像定义 3.6 一样,称 g 在 RO 模型中是 CPA 多项式安全的,如果对任意的选择明文敌手(F,A_1),满足

$$\Pr[R \leftarrow 2^{\infty};(E,D) \leftarrow g(1^k);(m_0,m_1) \leftarrow F^R(E);b \leftarrow \{0,1\}$$
$$\alpha \leftarrow E^R(m_b):A_1^R(E,m_0,m_1,\alpha) = b] \leqslant 1/2 + \mu(n)$$

这里,R 表示一般的 Oracle,是从$\{0,1\}^*$到$\{0,1\}^{\infty}$的函数,2^{∞}表示所有 Oracle 的集合,"∞"并非真的无限,只是避免提问"足够长是多长"这类问题,$\mu(n)$是可忽略函数。

CCA 安全性:这里的敌手 A 称为 RS 敌手,即有非一致多项式算法 $A=(F,A_1)$,各自获得一个 Oracle R 及一个解密 Oracle 的黑盒实现 D^R;F 的任务就是提出一对明文 m_0,m_1,A_1 被随机给予其中一个的密文 α,则只要不允许向解密 Oracle D^R 询问 α(因为禁止提出和最终论断等价的询问),A_1 就不可能以不可忽略优势猜中是哪一个明文。

称 g 在 RO 模型中抗 CCA 攻击是安全的,如果对任意 RS 敌手(F,A_1),满足

$$\Pr[R \leftarrow 2^{\infty};(E,D) \leftarrow g(1^k);(m_0,m_1) \leftarrow F^{R,D^R}(E);b \leftarrow \{0,1\}$$
$$\alpha \leftarrow E^R(m_b):A_1^{R,D^R}(E,m_0,m_1,\alpha) = b] \leqslant 1/2 + \mu(n)$$

Bellare 等在文献[4]中提出了一个在 RO 模型中抗 CCA 攻击是安全的方案,由于归约并不"紧",应用意义并不大,但其设计思想很能体现 RO 方法论的特点。Bellare 等把该思想作了进一步改进,在 1994 年提出了著名的公钥加密方案 OAEP[9],可证明该方案抗 CCA2 攻击是安全的,目前已成为新一代 RSA 加密标准。基本组成如下:核心组件是一个 Padding 函数,即 $\mathrm{OAEP}^{G,H}(x,r)=x \oplus G(r)||r \oplus H(x \oplus G(r))$,这里 x 是被加密消息,r 是随机输入;加密算法为 $E^{G,H}(x)=f(\mathrm{OAEP}^{G,H}(x,r))$,这里 f 是陷门置换(如 RSA 函数)。基本设想是构造一个具有良好随机性的"遮掩函数"隐蔽明文的统计特性。

2. 数字签名方案

Bellare 等在文献[4]中也给出了一种具有可证明安全性的签名方案,该签名方案要求陷门置换 f 具有"均匀分布"特点,而标准 RSA 置换不具有这个性质,因此基于 RSA 无法设计该类方案。文献[10]中提出了一个基于 RSA 的签名方案,该签名方案引入了概率机制,有更好的安全界。该方案不仅可证明其安全性,而且相应的归约是很"紧"的,一个敌手伪造

签名的能力和对 RSA 求逆的能力相当，总之安全性和分解整数的困难性紧密相关。稍作改进，也可以具有消息恢复功能。

目前其他可证明安全性的签名方案大都是基于识别协议的签名方案。例如，Fiat 和 Shamir 曾应用 RO 假设构造了一个安全性和分解因子一样困难的签名方案[7]，并证明了其与识别协议的等价性。文献[11]对基于 Fiat-Shamir 识别协议的签名方案[7]作了具体安全性分析，通过交换应答和承诺的顺序改进设计了一类新的 Fiat-Shamir 类型签名方案(E-swap 签名方案)，具有更好的具体安全性。

其他一些进展可参见文献[12]～[14]等。文献[14]基于计算 Diffie-Hellman(CDH)假设，对一个源于 Schnorr 签名的改进方案 EDL 给出了归约很“紧”的安全性证明，使得该签名方案得到了业界的广泛重视。另外值得特别说明的是，文献[12]对于完善 RO 模型方法论具有重大贡献，即提出了以 Folklore 引理为代表的一般性安全论断，主要适用于许多基于识别协议的签名方案，特别是证明了迄今为止唯一一个 ElGamal 变形签名方案 MEG 的安全性。Folklore 引理的基本思想是 Oracle 重放(Replay)攻击，即在 RO 模型中实施归约化证明时，重放多项式不同(但有一定联系)的随机 Oracles(这相当于为敌手提供多个模拟环境)，如果敌手能以不可忽略的概率伪造多个签名，就可以求解该方案的基础困难问题，如离散对数问题。该方法的缺点是所得到的归约不够“紧”。

这里主要介绍 Bellare 和 Rogaway 于 1996 年提出的 RSA-FDH 签名方案[10]，该方案的基本思想是结合 RSA 假设并基于经典的全域杂凑(Full-Domain Hash)签名方法论。该方案由以下 3 个算法组成。

(1) 密钥生成算法：输入 1^k，该算法随机选择两个 $k/2$b 的素数 p 和 q，计算 $n=p\cdot q$；随机选择 $e\in \boldsymbol{Z}^*_{\varphi(n)}$ 并计算 d 使得 $ed\equiv 1 \bmod \varphi(n)$。

用户的公钥为 (e,n)，私钥为 (d,n)。设 $H:\{0,1\}^*\rightarrow \boldsymbol{Z}^*_n$ 是一个抗碰撞的 Hash 函数。

(2) 签名算法：给定消息 m 和私钥 (d,n)，该算法计算并输出签名 $\sigma=H(m)^d \bmod n$。

(3) 验证算法：给定签名 σ、消息 m 和公钥 (e,n)，该算法检验 $\sigma^e\equiv H(m) \bmod n$，如果成立，输出 1(签名有效)，否则输出 0(签名无效)。

为了证明 RSA-FDH 的安全性，下面给出 RSA 问题的定量描述。

对于所有的 $k\in \mathbf{N}$，如果一个求逆算法 $\mathcal{I}$ 可以在 $t(k)$ 时间内以 $\varepsilon(k)$ 的成功概率破解 RSA 问题，则称算法 $\mathcal{I}$ 可 (t,ε) 破解 RSA 问题。

如果任何求逆算法都不能 (t,ε) 破解 RSA 问题，则称 RSA 是 (t,ε) 安全的。

定理 3.1 假设 RSA 问题是 (t',ε') 安全的，则 RSA-FDH 签名方案是 (t,ε) 安全的，其中

$$t = t' - (q_h + q_{\text{sig}} + 1)\cdot \mathcal{O}(k^3)$$

$$\varepsilon = \frac{1}{\left(1-\dfrac{1}{q_{\text{sig}}+1}\right)^{q_{\text{sig}}+1}}\cdot q_{\text{sig}}\cdot \varepsilon'$$

式中，q_{sig} 表示签名询问的次数；q_h 表示随机预言询问的次数。当 q_{sig} 较大时，有

$$\varepsilon \approx \exp(1)\cdot q_{\text{sig}}\cdot \varepsilon'$$

证明 假设伪造者$\mathcal{F}$经过q_{sig}次签名询问和q_h次随机预言询问，可以(t,ε)攻破RSA-FDH签名方案。下面构造一个求逆算法$\mathcal{I}$可以(t',ε')破解RSA困难问题。

假设给定了实例y，挑战者要找到满足$x^e \equiv y \bmod n$的x。注意到，在随机预言模型方法论中，伪造者$\mathcal{F}$不能自己计算Hash值，他只能通过Oracle询问来得到消息的Hash值。

算法$\mathcal{I}$设置用户的公钥为(n,e)，以此来运行算法$\mathcal{F}$。攻击者$\mathcal{F}$需要访问签名Oracle和随机Oracle，$\mathcal{I}$需要自己回答这些Oracle。为了简单起见，当$\mathcal{F}$询问一个消息的签名时，假定其已经对该消息进行了相应的Hash询问。如果没有，$\mathcal{I}$自己进行Hash询问。$\mathcal{I}$使用一个计数器i，初始化为0。

算法$\mathcal{I}$按照以下方式来回答伪造者$\mathcal{F}$对消息m_i的Oracle询问：

当$\mathcal{F}$对消息m进行Hash Oracle询问时，$\mathcal{I}$把计数器i增加1，令$m_i=m$，并随机选择$r_i \in \boldsymbol{Z}_n^*$。$\mathcal{I}$依概率$p$返回$h_i=r_i^e \bmod n$，依概率$1-p$返回$h_i=yr_i^e \bmod n$，这里$p$是一个固定的概率，其值将在后面确定。

当$\mathcal{F}$对消息m进行签名询问时，它已经对m进行了Hash询问，于是m等于某个m_i。如果$h_i=r_i^e \bmod n$，则$\mathcal{I}$返回r_i作为签名。由于$h(m_i)=h_i=r_i^e \bmod n$，显然r_i就是m_i的有效签名。否则，算法中止，求逆算法失败。

最终，算法$\mathcal{F}$中止并输出一个伪造(m,σ)。假设$\mathcal{F}$之前已经对m进行了Hash询问。如果没有，$\mathcal{I}$自己进行Hash询问。无论何种情况，存在某个i使得$m=m_i$。那么，如果$h_i=yr_i^e \bmod n$，则有$\sigma=h_i^d \bmod n=y^d r_i \bmod n$，于是$\mathcal{I}$输出

$$x = y^d \bmod n = \sigma / r_i \bmod n$$

作为y的逆。由于$y \equiv x^e \bmod n$，x即为所求。否则，算法中止，$\mathcal{I}$失败。

算法$\mathcal{I}$能够回答所有的签名询问的概率至少为$p^{q_{sig}}$，然后$\mathcal{I}$输出y的逆的概率为$1-p$。所以，$\mathcal{I}$至少以概率$\alpha(p)=p^{q_{sig}} \cdot (1-p)$输出$y$的逆。容易看出，当$p$取$p_{max}=1-1/(q_{sig}+1)$时，$\alpha(p)$取最大值，并且

$$\alpha(p_{max}) = \frac{1}{q_{sig}}\left(1-\frac{1}{q_{sig}+1}\right)^{q_{sig}+1}$$

从而，有

$$\varepsilon(k) = \frac{1}{\left(1-\frac{1}{q_{sig}+1}\right)^{q_{sig}+1}} \cdot q_{sig} \cdot \varepsilon'(k)$$

而当q_{sig}较大时，有$\varepsilon \approx \exp(1) \cdot q_{sig} \cdot \varepsilon'$。

算法$\mathcal{I}$的运行时间等于算法$\mathcal{F}$的运行时间加上计算h_i所需要的时间，于是得到时间t的公式。

注意到FDH类型的签名方案是确定性的签名方案，每一个消息都有唯一的签名，所以在其安全性证明中，一个模拟器要么能够产生消息的签名，要么就不能产生其签名，这在一定程度上限制了安全性归约。这一发现推动了概率签名方法的设计思想，如PSS签名方法和PFDH签名方法，其中每一个消息都有多个签名。