随机预言机是实用的：一种设计高效协议的模式

米希尔·贝拉雷*

菲利普·罗加韦†

摘要

我们认为，随机预言机模型——所有参与方都能访问一个公共随机预言机——在密码学理论和密码学实践之间架起了一座桥梁。在我们提出的模式中，首先设计并证明随机预言机模型中的协议 (P^R) 的正确性，然后用一个“适当选择”的函数 (h) 的计算来替代对预言机的访问，从而得到一个实用的协议 (P)。这种模式产生的协议比标准协议高效得多，同时保留了可证明安全性的许多优点。我们以加密、签名和零知识证明等问题为例说明了这些收益。

1 引言

密码学理论为密码学实践提供了一个可能非常宝贵的概念：可证明安全性。不幸的是，理论工作似乎常常只能以效率为代价来获得可证明安全性。这部分是由于以下原因。理论家将某些原语（例如，单向函数）视为“基本”的，并以低效的方式用它们构建更强大的原语（例如，伪随机函数）；但在实践中，强大的原语很容易获得，而所谓的基本原语似乎并不更容易实现。事实上，理论家们否认了自己证明此类原语系统存在的能力。

- 高性能计算与通信，IBM T.J. Watson 研究中心，美国纽约州约克镇高地，邮政信箱 704，邮编 10598。电子邮件：mihir@watson.ibm.com。
  † PS LAN 系统设计，IBM 个人软件产品部，美国德克萨斯州奥斯汀市伯内特路 11400 号，邮编 78758。电子邮件：rogaway@austin.ibm.com。

第 1 届会议 - 计算机与通信安全 '93 - 1993年11月 - 美国弗吉尼亚州 © 1993 ACM 0-89791-629-8/93/0011...$1.50

实践原语的性质不仅满足他们喜欢做的最强假设，甚至具有尚未定义或形式化的优势。

为了将可证明安全性的一些好处带给实践，将那些能捕捉实践原语真正似乎拥有的属性的对象纳入我们的模型是有意义的，并且即使关于它们的假设从理论角度来看非常强，也视这些对象为基本的。本文强调了一种此类方法的有效性和潜力。这个想法很简单：即为所有参与方——包括好的和坏的——提供对一个（公共的）随机预言机的访问；在这个模型中证明协议的正确性；然后用像哈希函数这样的对象替换随机预言机。我们强调，证明是在随机预言机模型中进行的，最后一步本质上是启发式的。本文的一个论点是，尽管如此，仍然保留了显著的安全保证益处。

这种模式的想法建立在 Goldreich、Goldwasser 和 Micali [20, 21] 以及 Fiat-Shamir [14] 的工作之上。它受到许多先前哈希函数“未经证明”的用途的指导。最后，它融合了我们社区许多成员分享并口头阐述的观点，应被视为民间智慧。有鉴于此，我们认为我们的贡献如下。首先，我们将使用随机预言机的隐含哲学提升为一个明确阐述的模式，我们坚持认为这给实践带来了显著的好处。其次，我们系统地将该模式应用于不同的密码学问题以获得高效的解决方案。第三，我们提供定义和证明，表明先前哈希函数的一些“未经证明”的用途可以在随机预言机模型中找到合理性。最后，我们建议构建我们认为适合实例化随机预言机的哈希函数。我们通过进一步详细描述该模式来进行。关于背景和相关工作的详细信息，请参见第 1.3 节。

1.1 随机预言机模式

上述理论家与实践者对原语看法的差异可以通过以下例子说明。理论家将单向函数视为基本对象，并用它们构建伪随机函数。但在实践中，如 Luby 和 Rackoff [30, 31] 所指出的，DES 提供了一个从 64 位到 64 位的伪随机函数。讽刺的是，如果需要一种实用的单向函数协议，很可能会从 DES 构造它——从而将“简单”原语归约到“复杂”原语。

如果试图设计高效的协议，更明智的做法是从一开始就对将要使用的原语做出强有力的、现实的假设。基于上述段落，一个 64 位字符串上的伪随机函数是一个极好的起点。正如我们下面描述的，采取更慷慨的假设似乎是合理的。

强大的原语。让我们看看第二个高效可计算的原始操作：由 MD5 算法 [35] 限制输入长度 (\le 400) 定义的映射 (h_2)。人们对这个函数有一些期望：难以找到一个 (x) 使得 (h_2(x) = x)；难以找到一个 (x) 使得 (h_2(x)) 的汉明权超过 120；(f_a(x) = h_2(xa)) 是（实际上）一个伪随机函数族；等等。这到底是什么对象？迄今为止，还没有令人满意的答案。也就是说，没有一个正式的定义能够涵盖这个函数似乎拥有的大部分优良属性——并且不清楚是否能找到一个。

该模式。我们对“像 (h_2) 这样的函数能实现什么？”的回答是说，它可以被视为一个随机函数，其含义是它可以在以下设计方法中扮演 (h) 的角色。假设有一个协议问题 (\Pi)（该问题“独立”于原语 (h)）。为了设计 (\Pi) 的一个好的协议 (P)：

（1）在计算模型中为 (\Pi) 找到一个正式定义，其中所有参与方（包括敌手）共享一个随机预言机 (R)。
（2）在此随机预言机模型中设计 (Π) 的高效协议 (P)。
（3）证明 (P) 满足 (\Pi) 的定义。
（4）将对 (R) 的预言机访问替换为 (h) 的计算。我们的论点是，当正确执行时，这种方法能产生安全高效的协议。事实上，根据这种模式构建的协议迄今为止在实践中被证明是“安全”的。但我们强调，所有可证明安全性的声明都是在随机预言机模型中的声明，并且用 (h) 实例化预言机只是一种我们根据经验信任其成功的启发式方法。

注意，(h) 不能真正像一个随机函数，因为它有一个简短的描述。在许多方面，(h) 与随机预言机非常不同。但这并没有改变该方法的成功。

我们强调，协议问题 (\Pi) 和协议 (P) 必须与我们打算使用的哈希函数“独立”。很容易构造非自然的问题或协议，其描述和目标明确依赖于 (h)，使得协议在随机预言机模型中安全，但当随机预言机用哈希函数实例化时会失败。“独立性”的概念不会在本文中形式化。

实例化。在本文主体中，我们假设一个从 ({0,1}^*) 到 ({0,1}^\infty) 的随机预言机 (R)。我们使用这样的预言机而不进一步解释，以提供描述给定协议所需的任何随机映射。

当用具体函数 (h) 实例化随机预言机时，必须首先注意确保 (h) 在设计上足够保守，以免屈服于密码分析攻击，其次确保 (h) 不暴露任何相关的“结构”（这些结构归因于它是从某些低级原语定义的）。第 6 节给出了两种类型陷阱的例子。正如该节所解释的，像 MD5 和 SHA 这样的标准哈希函数本身并不能很好地替代随机预言机；但人们不必看得太远。候选实例化包括输出被截断的哈希函数；输入长度受限的哈希函数；以及以某种非标准方式使用的哈希函数，例如 (h_3(x) = \mathrm{MD5}(xx))。参见第 6 节。

1.2 结果

本文的结果可以分为三种。首先是针对各种密码学问题的新颖高效解决方案。其次是对已知启发式方法的合理性证明。第三是一些我们在随机预言机模型中的调查引导我们证明的“理论”结果。在每种情况下，我们都提供了协议、定理以及适用于随机预言机设置的新定义。

高效加密。在标准设置中可能但不切实际的目标在随机预言机设置中变得可行。我们用一个例子说明：公钥加密。在下文中，(G: {0,1}^* \to {0,1}^\infty) 是一个随机生成器；(k) 是安全参数；(H: {0,1}^* \to {0,1}^k) 是一个随机哈希函数；(f) 是一个陷门置换，其逆为 (f^{-1})；(G(r) \oplus x) 表示 (x) 与 (G(r)) 输出的前 (|x|) 位的按位异或；“(|)” 表示连接。对于一个具体实现，(f) 可能是平方运算 [35] 或 RSA [38]。

我们建议在随机预言机模型中高效加密的两种方案：

（1）对来自 (f) 定义域的一个随机值 (r)，设置 (E^G(x) = f(r) \parallel G(r) \oplus x)。
（2）对来自 (f) 定义域的一个随机值 (r)，设置 (E^{G,H}(x) = f(r)\parallel G(r)\oplus x\parallel H(rx))。

这里 (x) 是要加密的消息，(f) 是接收者的公钥，(f^{-1}) 是他的私钥。关于背景、定义、结果的精确陈述以及与已知方案的效率比较，请参见第 3 节，但简而言之，那里论证了以下几点：第一个方案实现了 [24] 定义的多项式/语义安全性；第二个方案在 [36] 的意义上能抵抗选择密文攻击，并且在 [13] 的意义上也是不可延展的；两者都比之前针对相同目标的可证明安全方案 [24, 4, 34, 36, 11, 13] 效率显著提高。

已知启发式方法的合理性证明。通过转向随机预言机设置，各种著名的“技巧”找到了形式化的合理性证明。（这并不意味着现有的协议通常可以通过采用随机预言机模型来证明其合理性；恰恰相反，这似乎是个例外而非规则。）我们用以下一对例子来说明。

流行的签名方案，如 RSA，是以下情况的一个实例：对于一个陷门置换 (f) 和哈希函数 (H)，消息 (x) 的签名是 (f^{-1}(H(x)))。人们普遍认识到，哈希函数的任何自然属性都不能使这种方法成为一个安全的签名方案。然而，对于作为随机哈希函数的 (H)，我们证明了该方案能够抵抗自适应选择消息攻击。参见第 4 节。

一个消除交互式零知识证明中交互的启发式方法，归功于 M. Blum，(^{2}) 是让证明者基本上向自己询问验证者会问的问题，通过将这些查询计算为参与方之间已交换消息的哈希值。我们证明这种构造在随机预言机模型中是安全可证明的。提供这个证明需要为随机预言机模型中的零知识给出形式化定义。参见第 5 节。

理论结果。推广刚才描述的结果，我们证明任何具有交互式证明的语言都可以将其证明高效地转换为非交互式零知识证明。计算模型是所有参与方——包括作弊的证明者——只能对随机预言机进行多项式次数的查询。我们还证明，在随机预言机模型中，恒定轮次、信息论安全的函数评估是可能的。由于篇幅限制，这些结果的定义和证明被省略。

1.3 背景与相关工作

证明各方拥有随机预言机的模型中的协议正确性，然后用适当的密码学原语实例化该预言机的基本思想起源于 [20, 21]。[20] 建议并为此目的构建的密码学原语是伪随机函数。然而，为了使 PRF 保留其属性，指定其（并使其能够计算的）种子必须对敌手保持未知。因此，该模式的适用性仅限于敌手被拒绝访问随机预言机的协议。4 因此，在许多应用（特别是本文的应用）中，PRF 是不够的。但请注意，当设置允许通过 PRF 实例化预言机时，产生的协议通常可以在标准计算模型下基于标准复杂性理论假设被证明正确，这是我们建议的通过哈希函数实例化所无法实现的。

第一个明确采用公共随机预言机模型的工作——所有参与方，包括敌手，都可以访问该预言机——是 Fiat 和 Shamir [14] 的工作。作者使用此模型将身份认证方案转换为数字签名方案（在此转换过程中“完全”牺牲严谨性）。

上述 M. Blum 使交互式证明变为非交互式的想法可以被认为是 Fiat-Shamir 思想的扩展。Micali [32] 关于计算有界检查的一个令人兴奋的最新结果部分利用了同样的技术。

Impagliazzo 和 Rudich [27] 将单向函数建模为随机预言机。他们这样做是为了证明，给定一个黑盒单向函数，证明密钥交换协议的存在性就像分离 P 与 NP 一样困难。他们也使用随机预言机来获得积极结果；其中，他们形式化并证明了随机预言机模型中私钥密码系统的存在性。

与我们工作并行且独立，Leighton 和 Micali [28] 将哈希函数视为公共随机预言机，以证明一种新的高效签名方案的安全性。他们使用随机预言机模型来定义和证明精确的、非渐近的安全性。在另一篇论文 [29] 中，同样的作者使用被视为随机预言机的哈希函数给出了新的密钥交换方案。

由于本文主题广泛，特定目标的历史在描述该目标的章节中总结。

1.4 未来方向

在当前工作中仅有限地提出，并在 [28] 中充分体现的是，随机预言机模型有助于给出避免复杂性理论和渐近的精确定义和结果。将我们的结果在这种意义上精确化是可行且可取的。一个典型的定理会以敌手发出的预言机查询次数来表达敌手获得的优势。

我们不知道有什么复杂性理论假设能够很好地捕捉公共随机预言机的所有优良属性。是否有办法将 [20] 的伪随机函数族概念扩展到一个同样有用和引人注目的、不涉及隐藏随机性的概念？

2 预备知识

符号。 ({0,1}^*) 表示有限二进制字符串的空间，({0,1}^\infty) 表示无限二进制字符串的空间。除非另有说明，字符串都是有限的。我们用 (a|b) 或 (ab) 表示字符串 (a) 和 (b) 的连接。空字符串记为 (\Lambda)。多项式时间算法是指其运行时间是其第一个参数的多项式的算法。“PPT”代表“概率多项式时间”。函数 (\epsilon(k)) 是可忽略的，如果对于每个 (c) 都存在一个 (k_c)，使得对于每个 (k \geq k_c) 有 (\epsilon(k) \leq k^{-c})。如果函数不是可忽略的，则被称为非可忽略的。我们将使用符号“(k^{-\omega(1)})”来表示可忽略函数类或该

类中的一个特定匿名函数。为了指定概率实验和空间，我们使用源于 [26] 的符号。特别要回顾，如果 (A) 是输入为 (x, y, \ldots) 的概率算法，那么 (a \gets A(x, y, \ldots)) 表示通过运行 (A(x, y, \ldots)) 选择 (a) 的实验，而 ([A(x, y, \ldots)]) 表示所有能被 (A(x, y, \ldots)) 以正概率输出的元素的集合。

预言机。为方便起见，随机预言机 (R) 是从 ({0,1}^*) 到 ({0,1}^\infty) 的映射，通过对每个 (x) 均匀且独立地选择 (R(x)) 的每个比特来选择。当然，没有实际的协议使用无限长的输出，这只是省去了我们说明“足够长”是多长。我们用 (2^\infty) 表示所有随机预言机的集合。

字母 “(R)” 将表示“通用”随机预言机，而 (G: {0,1}^* \to {0,1}^\infty) 表示随机生成器，(H: {0,1}^* \to {0,1}^k) 表示随机哈希函数。每当提到多个预言机，所有这些预言机都是独立选择的。通过各种自然的编码，单个随机预言机 (R) 可以用来提供任意多个独立的随机预言机。

通常，提供给算法的预言机用上标表示。有时预言机是隐含的并从符号中省略。

陷门置换。遵循 [26]，一个陷门置换生成器是一个 PPT 算法 (\mathcal{G}_{*})，它在输入 (1^{k}) 时输出（编码的）三个算法的三元组 ((f, f^{-1}, d))。前两个是确定性的，最后一个是概率性的。我们要求 ([d(1^{k})]) 是 ({0, 1}^{k}) 的子集，并且 (f, f^{-1}) 是 ([d(1^{k})]) 上的置换，互为逆运算。我们要求存在一个多项式 (p)，使得 (f, f^{-1}) 和 (d) 在时间 (p(k)) 内可计算，并且对于所有非均匀多项式时间敌手 (M)，

$$\varepsilon(k) = \Pr[(f, f^{-1}, d) \leftarrow \mathcal{G}_*(1^k); \\ x \leftarrow d\left(1^k\right); y \leftarrow f(x) : M(f, d, y) = x]$$

是可忽略的。如前所述，对适当合数取模的平方运算 [42, 3]、其变体 [26] 或 RSA [38] 是陷门置换的好例子。如果对于所有 (k) 和所有 ((f, f^{-1}, d) \in [\mathcal{G}(1^k)])，都有 (d) 是 ({0, 1}^k) 上的均匀分布，则称陷门置换生成器 (\mathcal{G}_{*}) 是均匀的。

3 加密

我们依赖 [24, 33, 19, 18, 34, 13] 中的定义性工作。为简单起见，我们考虑非均匀（多项式时间）算法的敌手，可能是概率性的；扩展到均匀情况可以遵循 [18]。

加密。我们将公钥加密 [12] 的概念扩展到随机预言机模型。该方案由一个 PPT 生成器 (\mathcal{G}) 指定，它接收安全参数 (1^k) 并输出一对概率算法 ((E, D))，分别称为加密和解密算法，其运行时间以 (\mathcal{G}) 的时间复杂度为界。用户 (U) 运行 (\mathcal{G}) 得到 ((E, D))，并将前者公开，同时将后者保密。为了加密消息 (x)，任何人都可以计算 (y \gets E^R(x)) 并将其发送给 (U)；为了解密密文 (y)，用户 (U) 计算 (x \gets D^R(y))。我们要求对于所有 (x)，(D^R(E^R(x)) = x)，并假设为简单起见，如果 (y) 不是任何字符串 (x) 在 (E^R) 下的加密，则 (D^R(y) = 0)。

3.1 多项式安全性

背景。公钥加密的“基本”安全目标在 Goldwasser 和 Micali 的（等价的）多项式安全性和语义安全性概念 [24] 中找到了形式化。如果 (B*{f}) *表示 (f) 的一个硬核谓词（参见 [5, 43, 23]），那么可以通过设置 (E(x) = f(r*{1}) \parallel \ldots \parallel f(r*{|x|})) 来实现 [24] 意义上的安全性，其中每个 (r*{i}) 是从 (f) 的定义域中随机选择的，限制条件是 (B*{f}(r*{i}) =*

x{i})。这产生的加密长度为 (O(k \cdot |x|))，加密需要 (O(|x|)) 次 (f) 的求值，解密需要 (O(|x|)) 次 (f^{-1}) 的求值，这不实用。Blum 和 Goldwasser [4] 的一个更高效的构造产生的密文大小为 (O(|x| + k))，加密需要 (O(|x|)) 次模平方运算，解密需要 (O(1)) 次模幂运算加上 (O(|x|)) 次模平方，这仍然太昂贵。实践者通常将消息 (x) 嵌入到一个本应是随机的值 (r{x}) 中，然后设置 (E(x) = f(r{x}))。（例如，[39] 正是这样规定的。）通常使用的嵌入不能保证 (x) 和 (r{x}) 一样难以找到（更不用说 (x) 的所有属性都被隐藏）。

定义。我们将多项式安全性的概念 [24] 改编到随机预言机模型。（类似扩展的语义安全性概念仍然是等价的。）一个 CP-ad 敌手（选择明文敌手）(A) 是一对非均匀多项式时间算法 ((F, A_1))，每个都可以访问一个预言机。对于加密方案 (\mathcal{G}) 在随机预言机模型中安全，我们要求对于任何 CP-ad 敌手 (A = (F, A_1))，

$$\begin{aligned}&\Pr[R \leftarrow 2^{\infty}; (E, D) \leftarrow \mathcal{G}\left(1^k\right); (m_0, m_1) \leftarrow F^R(E);\\ &b \leftarrow \{0,1\}; \alpha \leftarrow E^R\left(m_b\right):\\ &A_1^R\left(E, m_0, m_1, \alpha\right) = b] \leq \tfrac{1}{2} + k^{-\omega(1)}.\end{aligned}$$

注意，用于加密和解密的预言机是提供给试图区分字符串 (m_0) 和 (m_1) 的加密的敌手的，因此，例如，从 (R) 导出的哈希 (H(x)) 肯定不能出现在字符串 (x) 的安全加密中。

通过 (E(x) = f(r)\parallel G(r)\oplus x) 加密。为了指定我们的加密方案，令 (\mathcal{G}{}) 是一个陷门置换生成器，并令 (G: {0,1}^{}\to {0,1}^{\infty}) 是一个随机生成器。在输入 (1^{k}) 时，我们的生成器 (\mathcal{G}) 运行 (\mathcal{G}{*}) 得到 ((f,f^{-1},d))。(E) 是以下算法：在输入 (x) 时，选取 (r\gets d(1^k)) 并输出 (E^{G}(x) = f(r)\parallel G(r)\oplus x,)，其中 (G(r)\oplus x) 表示 (x) 与 (G(r)) 的前 (|x|) 位的 XOR。当然，解密函数是 (D^{G}(y_{s}) = s\oplus G(f^{-1}(y)))。

定理。在附录 A 中，我们证明上述方案在随机预言机模型中是多项式安全的。

比较。我们实现了加密大小 (O(|x| + k))。除了可忽略成本的哈希，加密需要一次 (f) 的应用，解密需要一次 (f^{-1}) 的应用。将 (f) 设置为平方运算，这意味着加密需要一次模平方，解密需要一次模幂运算。这比上面讨论的 [4] 方案高效得多。

3.2 选择密文安全

背景。Naor 和 Yung [34] 提供了选择密文安全性的定义以及第一个可证明实现它的方案。Rackoff 和 Simon [36] 提出了一个更强的概念和相应的解决方案；De Santis 和 Persiano [11] 给出了另一个解决方案。后两个利用了知识证明，正如之前 [17, 6] 所建议的。所有已知的在标准假设下可证明安全的方案都依赖于非交互式零知识证明 [7, 16]，并且效率极低。Damgård [10] 提出了一个高效方案来实现 [34] 的定义，但该方案未被证明能达到 [34] 的定义，并且无法达到我们感兴趣的 [36] 的定义。与我们的方案密切相关的 Zheng 和 Seberry [44] 的方案将在后面讨论。

定义。我们将 [36] 的定义改编到随机预言机设置。一个 RS-敌手（"Rackoff-Simon 敌手"）(A) 是一对非均匀多项式时间算法 (A = (F, A_1))，每个都可以访问一个预言机 (R) 和一个 (D^R) 的黑盒实现。(F) 的任务是提出一对（等长的）消息 (m_0) 和 (m_1)，使得如果 (A_1) 获得其中一个的随机加密 (\alpha)，只要不允许 (A_1) 询问解密预言机 (\alpha)，(A_1) 就无法很好地猜测是哪一个。形式上，禁止 (A_1) 询问等于其最后一个参数的预言机查询。如果对于每个 RS-敌手 (A = (F, A_1))，

$$\begin{aligned}&\Pr[R \leftarrow 2^{\infty}; (E, D) \leftarrow \mathcal{G}\left(1^k\right); (m_0, m_1) \leftarrow F^{R, D^R}(E);\\ &b \leftarrow \{0,1\}; \alpha \leftarrow E^R\left(m_b\right):\\ &A_1^{R, D^R}(E, m_0, m_1, \alpha) = b] \leq \tfrac{1}{2} + k^{-\omega(1)}.\end{aligned}$$

则加密方案 (G) 是抗 RS-攻击安全的。

通过 (E(x) = f(r)\parallel G(r)\oplus x\parallel H(rx)) 加密。很容易看出，上一节的方案不能抵抗 RS-攻击。我们现在指定一个高效方案，它是安全的。令 (\mathcal{G}*{}) *是一个陷门置换生成器。令 (G: {0,1}^{}\to {0,1}^{\infty}) 是一个随机生成器，并令 (H: {0,1}^{*}\rightarrow {0,1}^{k}) 是一个随机哈希函数，独立地从随机预言机导出。我们方案的生成器 (\mathcal{G}) 运行 (\mathcal{G}*{*}) 得到 ((f,f^{-1},d))。(E) 是以下算法：在输入 (x) 时，选取 (r\gets d(1^k)) 并输出 (E^{G,H}(x)\gets) $f(r) \parallel x \oplus G(r) \parallel H(rx)$。要解密字符串 (y)，将其解析为 (a \parallel w \parallel b)，其中 (|a| = |b| = k)，并定义 (D^{G,H}(y)) 为 (w \oplus G(f^{-1}(a))) 如果 (H(f^{-1}(a) \parallel w \oplus G(f^{-1}(a))) = b)，否则为 0。

定理。在附录 A 中，我们证明上述方案能抵抗选择密文攻击。

比较。转换到随机预言机模型和我们的符号，Zheng 和 Seberry [44] 的方案是 (E^{*}(x) = f(r) \parallel (G(r) \oplus (xH(x))))。这个方案与我们的方案效率相同，并且我们相信它具有相同的安全属性。因此，随机预言机模型为 [44] 的构造提供了合理性证明。

3.3 不可延展性

背景。不可延展性的概念由 Dolev、Dwork 和 Naor [13] 引入。非正式地说，如果一个加密方案是不可延展的，那么你就不能通过目睹一个字符串 (x) 的加密来产生一个相关字符串 (x') 的加密。例如，给定 (x) 的加密，你不应该能够产生 (\overline{x}) 的加密。这个概念扩展了多项式安全性，特别是后者隐含于前者。[13] 给出了不可延展方案的构造。然而，这种构造完全不切实际，涉及巨大的公钥、多个签名的计算以及许多非交互式零知识证明。

定义。我们将 [13] 的定义改编到随机预言机设置。一个有趣的关系 (\rho*{E,\pi}^{R}: {0,1}^{}\times {0,1}^{}\to {0,1}) 必须满足对于每个 (x\in {0,1}^)，(i\in \mathbb{N})，(R\in 2^{\infty})，以及 (E,\pi \in {0,1}^)，有 (\rho*{E,\pi}^{R}(x,x) = \rho*{E,\pi}^{R}(x,0^{i}) = 0)；此外，(\rho) 必须可由一个多项式时间图灵机 (M^{R}(x,y,E,\pi)) 计算。一个 M-敌手（"延展性敌手"）(\mathcal{A}) 是一对 ((F,A)) 的非均匀概率多项式时间算法，每个都可以访问一个预言机 (R)。当 (F) 运行时，它输出一个算法 (\pi) 的描述，该算法也接受一个预言机，并且运行时间不超过 (F) 的复杂度。对于一个加密方案 (\mathcal{G}) 是不可延展的，我们要求对于每个有趣的关系 (\rho) 和每个 M-敌手 ((F,A))，存在一个（非均匀）多项式时间 (A{})，使得 (|\varepsilon (k) - \varepsilon_{}(k)|) 是可忽略的，其中

$$\begin{aligned}&\varepsilon(k) = \Pr[R \leftarrow 2^{\infty}; (E, D) \leftarrow G\left(1^k\right); \pi \leftarrow F^R(E);\\ &x \leftarrow \pi^R\left(1^k\right); \alpha \leftarrow E^R(x); \alpha' \leftarrow A^R(E, \pi, \alpha):\\ &\rho_{E,\pi}^R(x, D^R(\alpha')) = 1]\end{aligned}$$

$$\begin{aligned}&\varepsilon_*(k) = \Pr[R \leftarrow 2^{\infty}; (E, D) \leftarrow \mathcal{G}\left(1^k\right); \pi \leftarrow F^R(E);\\ &x \leftarrow \pi^R\left(1^k\right); \alpha'_* \leftarrow A_*^R(E, \pi):\\ &\rho_{E,\pi}^R(x, D^R(\alpha'_*)) = 1]\end{aligned}$$

参见 [13] 关于此定义背后直觉的解释，包括对关系 (\rho) 的限制。

通过 (E(x) = f(r)\parallel G(r)\oplus x\parallel H(rx)) 加密。加密方案与上一节相同。

定理。在附录 A 中，我们证明上述方案是不可延展的。

4 签名

定义。我们将 [26] 的定义扩展到随机预言机设置。一个数字签名方案是一个三元组 ((\mathcal{G}, \mathrm{Sign}, \mathrm{Verify}))，称为生成算法、签名算法和验证算法，都是多项式时间算法。前两个是概率性的，后两个可以访问随机预言机。在输入 (1^k) 时，生成器产生一对匹配的公钥和私钥 (PK, SK)。要对消息 (m) 签名，计算 (\sigma \gets \mathrm{Sign}^R(\mathrm{SK}, m))；要验证 ((m, \sigma))，计算 (\mathrm{Verify}^R(\mathrm{PK}, m, \sigma) \in {0, 1})。必须满足对于所有 (\sigma \in [\mathrm{Sign}^R(\mathrm{SK}, m)])，有 (\mathrm{Verify}^R(\mathrm{PK}, m, \sigma) = 1)。一个 (S)-ad 敌手（"签名敌手"）是一个（非均匀）多项式时间算法 (F)，可以访问 (R) 和一个签名预言机。(F) 的输出是一对 ((m, \sigma))，其中 (m) 未被询问过签名预言机。如果对于每个 (S)-ad 敌手 (F)，由下式定义的函数 (\varepsilon(k))

$$\Pr[R \leftarrow 2^{\infty}; (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathcal{G}\left(1^k\right); (m, \sigma) \leftarrow F^{R, \mathrm{Sign}^R(\mathrm{SK}, \cdot)}(\mathrm{PK}) : \mathrm{Verify}^R(\mathrm{PK}, m, \sigma) = 1]$$

是可忽略的，则该签名方案是安全的。如果其输出 ((m, \sigma)) 满足 (\mathsf{Verify}^R(\mathbf{PK}, m, \sigma) = 1)，我们说 (F) 是成功的。

协议。固定一个陷门置换生成器 (\mathcal{G}_{}*)。为简单起见，假设它是均匀的；下面说明如何为标准非均匀的进行修补。令 (H\colon {0,1}^{}\to) $\{0,1\}^k$ 像通常一样表示一个随机哈希函数。签名方案是 ((\mathcal{G},\mathrm{Sign}^H,\mathrm{Verify}^H))，其中 (\mathcal{G}) 在输入 (1^{k}) 时输出 (\mathrm{PK} = f) 和 (\mathrm{SK} = f^{-1})；(\mathrm{Sign}^H (f^{-1},m)) 是 (f^{-1}(H(m)))；而 (\mathrm{Verify}^H (f,m,\sigma)) 是 1 当且仅当 (f(\sigma) = H(m))。换句话说，就是借助哈希函数进行签名的“经典”方法。

均匀性：一个技术性问题。标准的陷门置换（基于平方或 RSA）不是均匀的，必须修补方案来处理它们。有许多修补方法。RSA 和 [26] 中定义的平方运算具有稠密定义域，并且成员资格可以高效测试。因此，为了对 (m) 签名，我们可以修改方案来计算 (H(1 \parallel m), H(2 \parallel m), \ldots)，直到找到一个属于定义域的成员 (y = H(i \parallel m))，然后返回 ((i, f^{-1}(y)))。验证以显而易见的方式定义。对于这些函数的另一个选择是应用 [2, 第 4.2 节] 的构造使它们均匀。[42, 3] 中定义的平方函数没有可高效测试的定义域，但仍然可以进行各种修补。事实上，甚至不需要函数是一个置换（参见 [35]）。

安全性。假设 (F) 是一个 (S)-敌手，以非可忽略的概率 (\lambda(k)) 成功。我们构造算法 (M(f, d, y))，它常常能非可忽略地计算 (f^{-1}(y))，如下所示。(M) 令 (\mathrm{PK} = f)。它为 (F) 抛硬币并开始运行 (F)。我们假设 (F) 恰好对 (H) 进行了 (n(k)) 次查询，所有查询都不同，并且如果 (F) 进行签名查询 (m)，那么它已经查询过 (H(m))；这很容易看出不失一般性。现在 (M) 随机选择 (t \in {1, \ldots, n(k)})。然后它按如下方式回复查询：

（1）令 (m_i) 表示 (F) 对 (H) 的第 (i) 次查询。如果 (i = t)，则 (M) 通过返回 (y) 来回答。否则，它选择 (r_i \gets {0, 1}^k) 并返回 (y_i = f(r_i))。
（2）假设 (F) 进行签名查询 (m)。如果 (m = m_t)，则 (M) 停止，承认失败。否则，(M) 用 (r_i) 回答，其中 (i \neq t) 满足 (m = m_i)。

令 ((m, \sigma)) 是 (F) 的输出。如果 (m \neq m_t)，则 (M) 停止承认失败。否则，它输出 (\sigma) 并停止。可以证明 (M(f, d, y)) 成功计算 (f^{-1}(y)) 的概率至少为

$$\left(1 - \frac{1}{n(k)}\right) \cdot \frac{\lambda(k)}{n(k)} - 2^{-k}$$

这仍然是非可忽略的。

5 零知识

我们提供随机预言机模型中零知识（ZK）证明的定义，然后展示如何在此模型中将 ZK 交互式证明变为非交互式的。该转换是高效的，因此我们得到的非交互式 ZK 证明的复杂度与交互式 ZK 证明相当。

5.1 定义

随机预言机模型中零知识的定义比简单地“相对化”标准定义要多一些。以下内容扩展了通常交互式设置 [25] 以及公共随机字符串模型 [6, 7] 中的表述。

设置。为简单起见，我们讨论 NP 语言 (L) 的证明。固定定义 (L) 的 NP 关系 (\rho)；(x) 在 (L) 中的成员资格的见证是指满足 (\rho(x, w) = 1) 的字符串 (w)。一个见证选择器是一个函数 (W)，它在任何输入 (x \in L) 时返回 (x) 在 (L) 中的成员资格的一个见证。

验证者是一个多项式时间函数 (V)，给定公共输入 (x)、到目前为止的对话 (\kappa \in {0,1}^*) 和一个（私有的）随机带 (r \in {0,1}^\infty)，返回 (V(x,\kappa,r))，这要么是给证明者的下一条消息，要么是表示他决定接受或拒绝的比特。证明者是一个 PPT (^5) 函数 (P)，给定公共输入 (x)、到目前为止的对话 (\kappa) 和辅助输入 (a)，返回给验证者的下一条消息 (P_{a}(x,\kappa))。（当 (x\in L) 时，辅助输入是这一事实的见证，否则是空字符串）。在随机预言机模型中，证明者和验证者也都接受这个预言机。

对于任何预言机 (R)，记 (\mathrm{conv}(V^R, P_a^R, x, r)) 为 (P_a^R) 和 (V^R) 之间当公共输入为 (x) 且 (V) 的随机带为 (r \in {0, 1}^\infty) 时的所有（对话记录）空间。记 (\mathrm{ACC}_V(\kappa, r) \in {0, 1}) 为验证者关于是否接受的判定。令

$$\begin{aligned}&\mathrm{ACC}(P_a, V, x) = \Pr[R \leftarrow 2^\infty; r \leftarrow \{0,1\}^\infty;\\ &\kappa \leftarrow \mathrm{conv}\big(V^R, P_a^R, x, r\big) : \mathrm{ACC}_V(\kappa, r) = 1]\end{aligned}$$

表示 (V) 在与 (P_{a}) 在公共输入 (x) 上的交互中接受的概率。在证明和协议中，我们经常滥用符号，只处理无限字符串 (r) 的任何相关前缀。

证明系统。如果 (\epsilon(n) \leq 1/2) 且满足以下两个条件，我们说 ((P, V)) 是 (L) 在随机预言机模型中的交互式证明，错误为 (\epsilon(n))。完备性条件要求如果 (x \in L)，那么对于 (x) 在 (L) 中的所有见证 (w)，有 (\mathbf{ACC}(V, P_w, x) = 1)。可靠性条件要求对于所有 PPT (\hat{P}) 和足够长的 (x)，有 (\mathbf{ACC}(\widehat{P}_\Lambda, V, x) \leq \epsilon(|x|))。

视图。为了定义零知识，首先更新验证者的视图以包含随机预言机；我们定义

$$\begin{aligned}&\mathrm{view}(V, P_a, x) = \{R \leftarrow 2^\infty; r \leftarrow \{0,1\}^\infty;\\ &\kappa \leftarrow \mathrm{conv}\big(V^R, P_a^R, x, r\big) : (\kappa, r; R)\}.\end{aligned}$$

模拟器。由于随机预言机是视图的一部分，它也必须成为模拟器输出的一部分；即，允许模拟器构建预言机的“模拟”。这类似于非交互式零知识 [6, 7]，其中允许模拟器构建并输出公共随机字符串的“模拟”。然而，随机预言机是一个无限对象，因此我们不能要求模拟器输出它。相反，我们允许模拟器规定一个小的（多项式大小的）预言机片段，其余的则“神奇地”

随机填充。形式上，模拟器是一个 PPT 算法，它在任何输入 (x) 上输出一个三元组 ((\kappa, r', T))，其中 (T = (x_1, y_1), \ldots, (x_t, y_t)) 是一个字符串对序列，且性质是 (x_1, \ldots, x_t) 是不同的。随机预言机补全操作 ROC 以 (T) 为输入，返回一个随机预言机 (R)，它服从约束：对于所有 (i = 1, \ldots, t)，(R(x*{i})) 以 (y*{i}) 为前缀。类似地定义随机字符串补全操作 RSC 很方便，它接受一个字符串 (r' \in {0, 1}^{*}) 并附加一个无限的随机比特序列。我们定义 (S(x)) 的补全为概率空间

$$
\begin{aligned}
&S^c(x) = \{(\kappa, r', T) \leftarrow S(x); R \leftarrow \mathrm{R\ O\ C}(T);\\
&r \leftarrow \mathrm{R\ S\ C}(r') : (\kappa, r; R)\}.
\end{aligned}
$$

区分器。一个区分器是一个多项式大小的预言机电路族 (D = {D_x}_{x \in L})。记 (D_x^R(\kappa, r)) 为电路 (D_x) 在给定预言机 (R) 和输入 (\kappa, r) 时的输出。然后定义

$$
\begin{aligned}
&\mathrm{d\,i\,f\,f}_D(S^c(x), \mathrm{r\ v\ i\ e\ w}(V, P_a, x)) =\\
&\left|\Pr\left[(\kappa, r; R) \leftarrow S^c(x) : D_x^R(\kappa, r) = 1\right] -\right.\\
&\left.\Pr[(\kappa, r; R) \leftarrow \mathrm{r\ v\ i\ e\ w}(V, P_a, x) : D_x^R(\kappa, r) = 1]\right|.
\end{aligned}
$$

零知识。如果对于每个区分器 (D)、每个见证选择器 (W)、每个常数 (d) 和所有足够长的 (x \in L)，有

$$
\mathrm{d\,i\,f\,f}_D\left(S^c(x), \mathrm{r\ v\ i\ e\ u}\left(V, P_{W(x)}, x\right)\right) < |x|^{-d}.
$$

我们说模拟器 (S) 是 (P) 在 (L) 上对于验证者 (\widehat{V}) 的模拟器。我们说 (P) 在随机预言机模型中定义了 (L) 上的（计算性）ZK 协议，如果对于每个验证者 (\widehat{V})，存在一个 (P) 在 (L) 上对于 (\widehat{V}) 的模拟器。统计 ZK 可以类似定义。如果 ((P, V)) 是 (L) 在错误为 (\epsilon) 的随机预言机模型中的证明系统，并且 (P) 定义了 (L) 上的 ZK 协议，则它是一个 ZK 证明。

多定理证明。在应用中，重要的是我们能够以零知识的方式证明多项式多个自适应选择的定理，就像公共随机字符串模型中的零知识一样。为简单起见，上面我们坚持了一个定理的情况；在最终论文中，我们将给出一般定义。

知识证明。在最终论文中，我们还将定义随机预言机模型中的知识证明，并展示如何构建高效的非交互式零知识知识证明。

5.2 协议

问题。令 ((P', V')) 是 (L \in \mathbf{NP}) 在标准（即不含随机预言机）模型中的 ZK 证明，实现错误概率 (1 / 2)。令 (k(n) = \omega (\log n)) 给定。我们想要一个随机预言机模型中的非交互式 ZK 证明 ((P,V))，它实现错误 (\epsilon (n) = 2^{-k(n)})，同时将计算时间和通信比特数最多增加 (O(k(n))) 倍。

简化假设。像大多数这样的 ZK 证明一样，假设 ((P', V')) 是三轮的：$P'_w \to V' : \alpha$ 然后是 $V' \to P'_w : b$ 然后是 $P'_w \to V' : \beta$。这里 (b) 是一个随机比特（现在我们认为 (V') 的随机带上第一个比特就是这个比特），(w) 是 (P') 的辅助输入。消息 (\alpha) 由一组信封组成，大小为 (n^{\Theta(1)})。这些信封的某个子集根据挑战 (b) 被打开，并且对于任何字符串 (\alpha)，恰好存在一个值 (b \in {0, 1})，使得存在一个 (\beta) 满足 (\mathsf{ACC}*{V'}(\alpha b\beta, b) = 1)。*零知识由算法 (S') 捕捉，给定 (x, b)，它输出 (\alpha b\beta) 使得 (\mathsf{ACC}*{V'}(\alpha b\beta, b) = 1)，并且对于任何见证选择器 (W)，以下集合是计算不可区分的：({b \gets {0, 1}; \alpha b\beta \gets S'(x, b): (\alpha b\beta, b)}*{x \in L}) 和 ({\alpha \gets P_W'(x, \Lambda); b \gets {0, 1}; \beta \gets P'(x, \alpha b): (\alpha b\beta, b)}_{x \in L})。

转换。令 (H \colon {0,1}^* \to {0,1}^{2k}) 是一个随机哈希函数。新的证明者 (P_w^H) 计算 (\alpha_1 \gets P_w'(x,\Lambda))；...；(\alpha*{2k} \gets P_w'(x,\Lambda))；将 (b_i') 设置为 (H(\alpha_1 \ldots \alpha*{2k})) 的第 (i) 比特；计算 (\beta_1 \gets P_w'(x,\alpha_1b_1'))；...；(\beta*{2k} \gets P_w'(x,\alpha*{2k}b*{2k}'))；并将 ((\alpha_1, \ldots, \alpha*{2k}, \beta_1, \ldots, \beta*{2k})) 发送给 (V^H)。(V^H) 将 (b_i) 设置为 (H(\alpha_1 \ldots \alpha*{2k})) 的第 (i) 比特，并且当且仅当对所有 (i) 有 (\mathsf{ACC}_{V'}(\alpha_i b_i \beta_i, b_i) = 1) 时接受。新协议是非交互式的且效率如所声称的那样是显而易见的。

((P,V)) 是一个错误为 (2^{-k(n)}) 的 ZK 证明系统。完备性是显然的。我们可以证明，如果 (\widehat{P}^H) 进行 (T(n)) 次预言机查询，那么 (\mathbf{ACC}(\widehat{P}*{\Lambda},V,x)\leq T(n)\cdot 2^{-2k(n)})，对于足够长的 (n)，这最多为 (2^{-k(n)})。对于 ZK，缺乏交互意味着我们只需要模拟诚实验证者 (V) 的视图，相应的模拟器 (S) 如下。给定 (x\in L)，算法 (S) 选择 (b*{1}\gets {0,1} ;\ldots ;b*{2k}\gets {0,1})。现在对于每个 (i = 1,\dots ,2k)，它令 (\alpha*{i}b*{i}\beta*{i}\gets S^{\prime}(x,b*{i}))。它设置 (T = (\alpha*{1}\dots \alpha*{2k},b*{1}\dots b*{2k})) 并输出 ((c,\Lambda ,T))。应用于 (T) 的随机预言机补全操作产生一个映射 (H:{0,1}^{*}\to {0,1}^{2k})，该映射是随机的，但受约束：(H(\alpha*{1}\dots \alpha*{2k}) = b*{1}\dots b_{k})。基于我们对 (S^{\prime}) 的假设，我们可以根据定义逐步检查，(S) 是 (P) 在 (L) 上对于 (V) 的模拟器。由于篇幅限制，我们省略了细节。

6 实例化

扩展第 1.1 节的讨论，这里我们提供进一步的指导，以用哈希函数等原语来实例化随机预言机。

首先也是最重要的，没有必要（也不可取）关注其随机预言机正在被实例化的目标协议的细节。唯一重要的是使用了多少个预言机以及它们的输入/输出长度要求是什么。我们的论点是，对于随机预言机的适当实例化应该适用于任何没有通过预期其预言机将采用的确切机制来故意阻碍我们的方法的协议。

在选择具体函数 (h) 来实例化预言机时，必须非常小心。让我们从一些不起作用的例子开始。

首先考虑映射 MD5。这个函数不适合替代随机预言机，因为 [41] 观察到，对于任何 (x)，存在一个 (y)，使得对于任何 (z)，MD5(xyz) 可以仅根据 (|x|)、MD5(x) 和 (z) 轻松计算。像这样的结构会在应用中显现；特别是，[41] 指出这意味着 MD5(ax) 不能用作字符串 (x) 在密钥 (a) 下的消息认证码。

试图通过避免像 MD5 这样“结构化”的操作来克服困难，人们可能更喜欢像其压缩函数这样的“较低级别”的原语，(\mu: {0,1}^{640} \to {0,1}^{128})。这也不适合替代随机预言机，因为 [8] 已经证明可以高效地在这个映射中找到碰撞。

尽管标准哈希函数结构太多，不能成为好的随机预言机（如上所述），但人们不必看得太远；自然的候选构造包括以下这些，或它们的组合：

（1）输出被截断或以某种方式折叠的哈希函数；例如，(h_1(x) =) (\mathrm{MD5}(x)) 的前 64 位。
（2）输入长度受到适当限制的哈希函数；例如，(h_2(x) = \mathrm{MD5}(x))，其中 (|x| \leq 400)。
（3）以某种非标准方式使用的哈希函数；例如，(h_3(x) = \mathbf{MD5}(xx))。

（4）密码哈希函数的“第一块压缩函数”，例如，(h_4 \colon {0,1}^{512} \to {0,1}^{128}) 是计算 MD5(x) 时 512 位 (x) 的压缩。

作为一个例子，假设人们确定了一个（纯粹启发式的）映射 (h' \colon {0,1}^{256} \to {0,1}^{64})，定义为 $h'(x) = h_4((xx) \oplus C)$ 的前 64 位，其中 (C) 是一个随机选择的 512 位常数。为了在给定应用中按需扩展定义域和值域，可以首先定义 (h''(x) = h'(x\langle 0\rangle) || h'(x\langle 1\rangle)||) $h'(x\langle 2\rangle)|| \ldots$，其中 (|x| = 224) 且 (\langle i\rangle) 是将 (i) 编码为 64 位。接下来，通过将每个输入 (x) 编码为 (x') 来扩展 (h'')，(x') 由 (x)、比特“1”和足够多的 0 组成，使得 (|x'|) 是 128 位的倍数。现在令 (x' = x_1' \ldots x_n')，其中 (|x_i'| = 128)，并定义 (h(x) = h''(x_0'\langle 0\rangle) \oplus h''(x_1'\langle 1\rangle) \oplus \dots \oplus h''(x_n'\langle n\rangle))，产生一个映射，实际上，取 (h: {0,1}^{*} \to {0,1}^{\infty})。当然，有许多其他同样简单的方法来实例化随机预言机；这只是一个例子。

7 结论

实践中使用的协议几乎总是通过一个反复的过程来设计的：假设一个具体协议，寻找成功的攻击，找到一个，然后试图修补它。这种方法效果不佳。通过坚持定义我们的目标并证明实现它们，现代密码学为实践提供的不仅仅是任何特定的结果集；它是一种超越了迭代设计过程以解决未明确定义任务的方法。尽管在使用我们的模式时，人们“只”能得到一个结果说“这个协议的安全程度取决于 (h) 对随机预言机的实例化”，但比起宣布一个协议是安全的仅仅因为迄今为止还没有人提出成功的攻击，人们已经取得了非常大的成就。

致谢

早期与 Bob Blakley 关于许可证服务器问题 [40] 的讨论有助于使我们的想法具体化。我们从 Oded Goldreich、Birgit Pfitzmann 和 Steven Rudich 那里得到了有益的建议和参考文献。最后，感谢 ACM 计划委员会的所有成员的所有评论。

参考文献

[1] D. BEAVER, S. MICALI AND P. ROGAWAY, "The round complexity of secure protocols," STOC 90.

[2] M. BELLARE AND S. MICALI, "How to sign given any trapdoor permutation," JACM Vol. 39, No. 1, 214-233, January 1992.

[3] L. BLUM, M. BLUM AND M. SHUB, “A simple unpredictable pseudo-random number generator,” SIAM Journal on Computing Vol. 15, No. 2, 364-383, May 1986.

[4] M. BLUM AND S. GOLDWASSER, "An efficient probabilistic public-key encryption scheme which hides all partial information," Crypto 84.

[5] M. BLUM AND S. MICALI, "How to generate cryptographically strong sequences of pseudo-random bits," SIAM Journal on Computing, Vol. 13, No. 4, 850-864, November 1984.

[6] M. BLUM, P. FELDMAN AND S. MICALI, "Noninteractive zero knowledge and its applications," STOC 88.

[7] M. BLUM, A. DE SANTIS, S. MICALI AND G. PERSIANO, "Non-interactive zero-knowledge proof systems," SIAM Journal on Computing, 20(4), 1084-1118 (December 1991).

[8] B. DEN BOER AND A. BOSSELAERS, "Collisions for the compression function of MD5," Eurocrypt 93.

[9] G. BRASSARD, D. CHAUM AND C. CRÉPEAU, “Minimum disclosure proofs of knowledge,” JCSS Vol. 37, No. 2, 156-189, October 1988.

[10] I. DAMGARD, "Towards practical public key cryptosystems secure against chosen ciphertext attacks," Crypto 91.

[11] A. DE SANTIS AND G. PERSIANO, “Zero-knowledge proofs of knowledge without interaction” FOCS 92.

[12] W. DIFFIE AND M. E. HELLMAN, “New directions in cryptography,” IEEE Trans. Info. Theory IT-22, 644-654 (November 1976).

[13] D. DOLEV, C. DWORK AND M. NAOR, “Non-malleable cryptography,” STOC 91.

[14] A. FIAT AND A. SHAMIR, "How to prove yourself: practical solutions to identification and signature problems," Crypto 86.

[15] U. FBIGE, A. FIAT AND A. SHAMIR, “Zero knowledge proofs of identity,” Journal of Cryptology, Vol. 1, pp. 77-94 (1987).

[16] U. FEIGE, D. LAPIDOT, AND A. SHAMIR, “Multiple non-interactive zero-knowledge proofs based on a single random string,” FOCS 90.

[17] Z. GALIL, S. HABER AND M. YUNG, "Symmetric public key cryptosystems," manuscript, July 1989.

[18] O. GOLDREICH, “A uniform complexity treatment of encryption and zero-knowledge,” Journal of Cryptology, Vol. 6, pp. 21-53 (1993).

[19] O. GOLDREICH, “Foundations of cryptography,” Class notes, Spring 1989, Technion University.

[20] O. GOLDREICH, S. GOLDWASSER AND S. MICALI, "How to construct random functions," Journal of the ACM, Vol. 33, No. 4, 210-217, (1986).

[21] O. GOLDREICH, S. GOLDWASSER AND S. MICALI, “On the cryptographic applications of random functions,” Crypto 84.

[22] O. GOLDREICH AND H. KRAWCZYK, "On the composition of zero knowledge proof systems," ICALP 90.

[23] O. GÖLDREICH AND L. LEVIN, “A hard predicate for all one-way functions,” STOC 89.

[24] S. GOLDWASSER AND S. MICALI, "Probabilistic encryption," J. of Computer and System Sciences 28, 270-299, April 1984.

[25] S. GOLDWASSER, S. MICALI AND C. RACKOFF, "The knowledge complexity of interactive proof systems," SIAM J. of Comp., Vol. 18, No. 1, pp. 186-208, February 1989.

[26] S. GOLDWASSER, S. MICALI AND R. RIVEST, “A digital signature scheme secure against adaptive chosen-message attacks,” SIAM Journal of Computing, 17(2):281-308, April 1988.

[27] R. IMPAGLIAZZO AND S. RUDICH, "Limits on the provable consequences of one-way permutations," STOC 89.

[28] T. LEIGHTON AND S. MICALI, "Provably fast and secure digital signature algorithms based on secure hash functions," Manuscript, March 1993.

[29] T. LEIGHTON AND S. MICALI, "New approaches to secret key exchange," Crypto 93.

[30] M. Luby AND C. RACKOFF, "How to construct pseudorandom permutations from pseudorandom functions," SIAM J. Computation, Vol. 17, No. 2, April 1988.

[31] M. Luby AND C. RACKOFF, “A study of password security,” manuscript.

[32] S. MICALI, "CS proofs," Manuscript.

[33] S. MICALI, C. RACKOFF AND B. SLOAN, “The notion of security for probabilistic cryptosystems,” SIAM J. of Computing, April 1988.

[34] M. NAOR AND M. YUNG, "Public-key cryptosystems provably secure against chosen ciphertext attacks," STOC 90.

[35] M. RABIN, "Digitalized signatures and public-key functions as intractable as factorization," MIT Laboratory for Computer Science TR-212, January 1979.

[36] C. RACKOFF AND D. SIMON, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," Crypto 91.

[37] R. RIVEST, "The MD5 message-digest algorithm," IETF Network Working Group, RFC 1321, April 1992

[38] R. RIVEST, A. SHAMIR, AND L. ADLEMAN, “A method for obtaining digital signatures and public key cryptosystems,” CACM 21 (1978).

[39] RSA DATA SECURITY, INC., "PKCS #1: RSA Encryption Standard," June 1991.

[40] P. ROGAWAY AND B. BLAKLEY, “An asymmetric authentication protocol,” IBM Technical Disclosure Bulletin (1993).

[41] G. Tsudik, "Message authentication with one-way hash functions," IEEE INFOCOM '92.

[42] H. WILLIAMS, “A modification of the RSA public key encryption procedure,” IEEE Transactions on Information Theory, Vol. IT-26, No. 6, November 1980.

[43] A. YAO, “Theory and applications of trapdoor functions,” FOCS 82.

[44] Y. ZHENG AND J. SEBERRY, “Practical approaches to attaining security against adaptively chosen ciphertext attacks,” Crypto 92.

A 加密方案的证明

我们提出一些加密方案的安全性证明。我们将假设（不失一般性）对于任何算法和该算法的任何预言机，向预言机提出的所有查询都是不同的。

$E(x) = f(r) \parallel G(r) \oplus x$ 方案是多项式安全的。证明采用反证法。令 (A = (F,A*{1})) *是一个击败协议的敌手；它常常获得优势 (\lambda (k))，其中 (\lambda) 是某个逆多项式。我们构造一个算法 (M(f,d,y))，当 ((f,f^{-1},d)\gets \mathcal{G}(1^{k});r\gets d(1^{k});y\gets f(r)) 时，常常能显著地计算 (f^{-1}(y))。算法 (M) 根据我们的方案基于 (f) 定义 (E)。它以一种自然的方式模拟预言机 (G)（通过自己抛硬币来回答查询）并采样 ((m_0,m_1)\gets F^G (E))。如果曾经询问 (G) 一个满足 (f(r) = y) 的 (r)，那么 (M) 输出 (r) 并停止。否则，(F(E)) 终止，并且 (M) 选择 (\alpha \gets y\parallel s)，其中 (s\gets {0,1}^{|m_0|})。然后 (M) 模拟 (A*{1}^{G}(E,m*{0},m*{1},\alpha))，观察 (A*{1}) 发出的预言机查询，看是否有任何预言机查询 (r) 满足 (f(r) = y)。如果有，(M) 输出 (r)。令 (\mathsf{A}k) 表示事件：(A*{1}) 询问查询 (r = f^{-1}(y))。在 (A*{1}) 没有询问 (G) 在 (r) 处的像的情况下，(A*{1}) 在区分 (m_0) 和 (m*{1}) 方面没有优势。所以

$$1/2 + \lambda(k) = \Pr[A\mathrm{succeeds}|\mathrm{A}_k] \cdot \Pr[\mathrm{A}_k] + \Pr\left[A\mathrm{succeeds} \mid \overline{A_k}\right] \cdot \Pr\left[\overline{A_k}\right]$$

最多为 (\operatorname*{Pr}[\mathsf{A}_k] + 1 / 2)。因此 (\operatorname{Pr}[\mathsf{A}_k] \geq \lambda(k)) 必须是非可忽略的，并且 (M) 常常以非可忽略的概率成功求逆 (f)。

$E(x) = f(r) \parallel G(r) \oplus x \parallel H(rx)$ 方案能抵抗选择密文攻击。令 (A = (F,A*{1})) *是一个 RS-敌手，它以概率 (1 / 2 + \lambda (k)) 成功，其中 (\lambda (k)) 是某个非可忽略函数。我们构造一个算法 (M(f,d,y))，它常常以非可忽略的概率计算 (f^{-1}(y))，其中 ((f,f^{-1},d)\gets \mathcal{G}*{*}(1^{k});r\gets d(1^{k});y\gets f(r))。算法 (M) 开始运行 (F(E))，其中 (E) 是根据我们的方案从 (f) 定义的。(F) 接受三个预言机，即 (G,H) 和 (D^{G,H})，它们的查询由 (M) 回答如下。如果对 (G) 的查询 (r) 满足 (f(r) = y)，那么 (M) 输出 (r) 并停止；否则它返回一个适当长度的随机字符串。如果对 (H) 的查询 (rx) 满足 (f(r) = y)，那么 (M) 输出 (r) 并停止；否则它返回一个适当长度的随机字符串。为了回答对 (D^{G,H}) 的查询 (a \parallel w \parallel b)，算法 (M) 查看是否已经询问过 (G) 某个查询 (r) 和 (H) 某个查询 (ru)，其中 (a = f(r)) 且 (w = G(r) \oplus u)，如果是，则返回 (u)；否则它返回无效。如果 (M) 完成了 (F(E)) 的运行，那么它获得一个输出 ((m_0, m_1))。现在 (M) 运行 (A_1(E, m_0, m_1, \alpha))，其中 (\alpha = y \parallel w \parallel b)，其中 (w \gets {0, 1}^{|m_0|}) 且 (b \gets {0, 1}^k)。再次，(M) 必须模拟对 (G)、(H) 和 (D^{G,H}) 的查询行为。这与之前 (M) 运行 (F) 时完全一样。

为了证明这个构造有效，首先考虑 (A) 与其预言机一起运行的“真实”环境。令 (\mathsf{A}k) *表示事件：对于某个 (a, w,) 和 (b)，有 (a \parallel w \parallel b \leftarrow F(E))，并且 (A) 对 (G(r)) 或 (H(ru)) 进行了某个预言机调用，其中 (f(r) = a)。令 (\mathsf{L}k) 表示事件：(A_1) 询问 (D^{G,H}) 某个查询 (a \parallel w \parallel b)，其中 (b = H(f^{-1}(a) \parallel w \oplus G(f^{-1}(a)))，但 (A_1) 从未询问其 (H)-预言机关于 (f^{-1}

(a) \parallel w \oplus G(f^{-1}(a))) 的像。令 (n(k)) 表示发出的预言机查询总数。很容易验证 (\operatorname*{Pr}\left[\mathsf{L}_k\right] \leq n(k)2^{-k})。也很容易看出

$$\Pr\left[A\mathrm{succeeds}|\overline{\mathrm{L}_k} \wedge \overline{\mathrm{A}_k}\right] = 1/2.$$

因此 (1/2 + \lambda(k) = \operatorname{Pr}[A \text{ succeeds}]) 的上界为

$$\begin{aligned}&\Pr[A\mathrm{succeeds}|L_k]\Pr[L_k]+\\&\Pr\left[A\mathrm{succeeds} \mid \overline{L_k} \wedge A_k\right]\Pr\left[\overline{L_k} \wedge A_k\right]+\\&\Pr\left[A\mathrm{succeeds} \mid \overline{L_k} \wedge \overline{A_k}\right]\Pr\left[\overline{L_k} \wedge \overline{A_k}\right]\end{aligned}$$

即最多为 (n(k)2^{-k} + \operatorname{Pr}\left[\mathsf{A}_k\right] + 1/2)。所以

$$\Pr\left[\mathrm{A}_k\right] \geq \lambda(k) - n(k)2^{-k}.$$

现在，回到 (M) 对 (A) 的模拟，注意 (M) 未能表现得像 (A) 的概率以 (\operatorname{Pr}\left[\mathsf{L}_k\right]) 为界，因此

$$\Pr[M\mathrm{inverts}f\mathrm{at}y] \geq \lambda(k) - n(k)2^{-k+1}$$

这仍然是非可忽略的。证明完成。

$E(x) = f(r) \parallel G(r) \oplus x \parallel H(rx)$ 方案是不可延展的。直观地说，在加密字符串 (\alpha^\prime) 中存在一个有效的标签 (H(r^{\prime}x^{\prime}))，而该字符串不是提供给敌手 (A) 的加密副本，这充当了一个“知识证明”，证明 (A) “知道”（能够恢复）(x^{\prime})。现在假设 (A) 看到加密了 (x = G(f^{-1}(a))\oplus w) 的 (\alpha = a\parallel w\parallel b)，设法想出了一个与 (x) 相关的字符串 (x^{\prime}) 的加密。当 (r) 没有被询问 (G) 时，敌手 (A) 不能将 (x) 与（已知值）(x^{\prime}) 相关联，因为她对 (G(r)) 的值一无所知。因此 (A) 必须相当频繁地询问 (G) 关于 (r) 的像。每当她这样做时，她实际上已经求逆了陷门置换。上面的论证可以形式化；我们现在勾勒如何做到这一点。

给定一个 M-敌手 (\mathcal{A} = (F, A)) 和一个由多项式时间机器 (M) 计算的有趣关系 (\rho)，定义多项式时间算法 (A_{*}(E, \pi)) 如下：

$$A_*(E,\pi)\mathrm{computes}x_* \leftarrow \pi(1^k); r_* \leftarrow d(1^k);$$

$$\alpha_* \leftarrow f(r_*)\|G(r_*) \oplus x_*\|H(r_*x_*); \alpha'_* \leftarrow$$

$A(f,\pi,\alpha_*)$。如果 (\alpha*{\prime} = \alpha{})，那么 (A_{}) 输出 0 的加密。否则，(A*{}) 输出 (\alpha{*}^{\prime})。

我们将证明 (|\varepsilon (k) - \varepsilon*{*}(k)|) 是可忽略的，其中这些量如不可延展性的定义所示。需要一些案例分析来证明这一主张。它基于考虑两个相关的实验，第一个定义 (\varepsilon (k))，第二个定义 (\varepsilon{}(k))。我们从描述实验 1 开始。这里 (G\gets 2^{\infty})；(H\gets 2^{\infty})；((f,f^{-1},d)\gets \mathcal{G}_{}(1^{k}))；然后 (E) 是我们的加密算法，由 (f) 指定，(D) 是对应的解密；(\pi \leftarrow F^{G,H}(E))；(x\gets \pi^{G,H}(1^{k}))；(r\gets d(1^{k}))；(a = f(r))；(w = G(r)\oplus x)；(b = H(rx))；(\alpha = a\parallel w\parallel b)；和 (\alpha^{\prime}\gets A(E,\pi ,\alpha))。记 (\alpha^{\prime} = a^{\prime}w^{\prime}b^{\prime})，(r^{\prime} = f^{-1}(a^{\prime}))，和 (x^{\prime} = w^{\prime}\oplus G(r^{\prime}))。我们对 (M^{G,H}(x,x^{\prime},E,\pi)) 的值感兴趣，其期望值，记作 (\mathbf{E}_1[\rho (x,x^{\prime})])，恰好是 (\varepsilon (k))。在进行实验 1 时，我们区分以下情况：

情况 1：(\alpha' = \alpha)。在这种情况下，根据我们有趣关系的定义，(\rho(x, x') = 0)。

情况 2：假设情况 1 不成立且 (\mathcal{A}) 没有对 (r'x') 进行 (H)-预言机查询：

情况 2a. (b' = H(r'x'))。这个事件以概率 (2^{-k}) 发生。
情况 2b. (b' \neq H(r'x'))。在这种情况下，加密是混乱的，解密是 0，并且根据我们有趣关系的定义，(\rho(x,x') = 0)。

情况 3：假设情况 1 和情况 2 都不成立。

情况 3a. 对于任何询问 (H) 的字符串 (r'x') 满足 (H(r'x') = b')，要么 (f(r') \neq a)，要么 (G(r') \oplus x' \neq w')。那么 (\rho(x, x') = 0)。
情况 3b. 这里 (\alpha') 是一个有效的加密，并且 (A) 可以提取 (r') 和 (x')。区分。令 (\lambda_1) 表示这种情况的概率。我们区分：

情况 3b(i). 当 (A) 没有对 (r) 进行 (G)-预言机调用时
情况 (\mathfrak{Z}b(i)') (M^{G,H}) 询问一个查询 (r)。令 (\epsilon(k)) 是一个可忽略的函数，作为这种情况概率的上界。令 (\lambda_2) 是这种情况的概率。
情况 $3b(i)''$ $M^{G,H}$ 没有询问查询 r
情况 3b(ii). 当 (A) 对 (r) 进行 (G)-预言机调用时。令 (\epsilon(k)) 是一个可忽略的函数，作为这种情况概率的上界。

我们可以将 (\mathbf{E}_1[\rho (x,x^{\prime})]) 的上界表示为

$$\begin{aligned}&\mathbf{E}_1[\rho(x,x')] \leq \Pr[\mathrm{Case\ 2a}] \cdot 2^{-k} + \\&\Pr[\mathrm{Case\ 3b}] \cdot \mathbf{E}[\rho(x,x') | \mathrm{Case\ 3b(i)}] + \\&\Pr[\mathrm{Case\ 3b(ii)}] \leq 2^{-k} + \lambda_1(\epsilon(k) + \lambda_2) + \epsilon(k).\end{aligned}$$

我们现在描述实验 2。它定义为 (G \gets 2^{\infty})；(H \gets 2^{\infty})；((f, f^{-1}, d) \gets \mathcal{G}*(1^{k}))；然后 (E) 是我们的加密算法，由 (f) 指定，(D) 是对应的解密；(\pi \gets F^{G,H}(E))；(x \gets \pi^{G,H}(1^{k}))；(x{} \gets \pi^{G,H}(1^{k}))；(r_{} \gets d(1^{k}))；(a*{} = f(r{}))；(w = G(r_{}) \oplus x*{}))；(b{} = H(r_{}x*{}))；(\alpha{} = a_{} | w*{} | b{})；(\alpha_{}' \gets A(E, \pi, \alpha*{}))。记 (\alpha{}' = a_{}'w*{}'b{}', r_{}' = f^{-1}(a'))，和 (x*{}' = w{}' \oplus G(r_{}')) 如果 (\alpha*{} = \alpha{}', 0) 否则。我们对 (M^{G,H}(x, x_{}', E, \pi)) 的值感兴趣，其期望值，记作 (\mathbf{E}2[\rho(x, x{}')]，恰好是 (\varepsilon_{}(k))。

在分析实验 2 时，我们进行与上述相同的案例分析。一个重要的观察是，在实验 1 和 2 中，(A) 的第三个参数的分布是相同的。因此，(\mathbf{Pr}1[\text{Case 3b}] = \mathbf{Pr}2[\text{Case 3b}])。同样，很容易看出 (\mathbf{E}1[\rho(x,x')|\text{Case 3b(i)}''] = \mathbf{E}2[\rho(x,x')|\text{Case 3b(i)}''])。然后可以将 (\mathbf{E}2[\rho(x,x*)|\text{Case 3b(i)}'']) 的下界表示为

$$\begin{aligned}&\mathbf{E}_2[\rho(x,x_*)] \geq \Pr[\mathrm{Case\ 3b\ (i)}''].\\&\mathbf{E}_2\left[\rho\left(x,x_*'\right) \mid \mathrm{Case\ 3b\ (i)}'\right] \geq \left(\lambda_1 - 2\epsilon(k)\right)\lambda_2.\end{aligned}$$

因此 (|\mathbf{E}[\rho(x, x')] - \mathbf{E}[\rho(x, x_*)]| \leq 4\epsilon(k) + 2^{-k})，证毕。

专业术语中英文对照表

- **Random Oracle (Model)**: 随机预言机（模型）
- **Provable Security**: 可证明安全性
- **Primitive**: 原语
- **One-way Function**: 单向函数
- **Pseudorandom Function (PRF)**: 伪随机函数

- **Efficient / Efficiency**: 高效 / 效率
- **Heuristic**: 启发式（方法）
- **Hash Function**: 哈希函数
- **Instantiation**: 实例化
- **Encryption**: 加密
- **Decryption**: 解密
- **Public-key Encryption**: 公钥加密
- **Semantic Security**: 语义安全性
- **Polynomial Security**: 多项式安全性
- **Chosen-plaintext Attack (CPA)**: 选择明文攻击
- **Chosen-ciphertext Attack (CCA)**: 选择密文攻击
- **Non-malleability**: 不可延展性
- **Trapdoor Permutation**: 陷门置换
- **Generator**: 生成器 / 生成算法
- **Signature**: 签名
- **Digital Signature Scheme**: 数字签名方案
- **Adaptive Chosen-Message Attack**: 自适应选择消息攻击
- **Zero-knowledge (ZK) Proof**: 零知识证明
- **Interactive Proof**: 交互式证明
- **Non-interactive Zero-knowledge (NIZK)**: 非交互式零知识
- **Verifier**: 验证者
- **Prover**: 证明者
- **Simulator**: 模拟器
- **Distinguisher**: 区分器
- **Soundness**: 可靠性
- **Completeness**: 完备性
- **Knowledge Complexity**: 知识复杂性
- **Proof of Knowledge**: 知识证明
- **Hard-core Predicate**: 硬核谓词
- **Negligible Function**: 可忽略函数
- **Non-negligible Function**: 非可忽略函数
- **Probabilistic Polynomial Time (PPT)**: 概率多项式时间
- **Uniform / Non-uniform**: 均匀的 / 非均匀的
- **Asymptotics**: 渐近（性）
- **Complexity-theoretic Assumption**: 复杂性理论假设
- **Message Authentication Code (MAC)**: 消息认证码
- **Collision (resistance)**: 碰撞（抵抗性）
- **Compression Function**: 压缩函数
- **Bitwise XOR**: 按位异或
- **Concatenation**: 连接
- **Security Parameter**: 安全参数
- **Adversary**: 敌手