10 密钥管理与公钥革命

10.1 密钥分发与密钥管理

在第 1 章到第 7 章中，我们看到了如何在通信双方持有共享密钥的前提下，使用**私钥密码学**确保双方通过不安全通道进行通信的机密性和完整性。然而，我们自第 1 章以来一直推迟回答的问题是：

> **通信双方最初是如何共享一个秘密密钥的?**

显然，密钥不能简单地通过公共通信通道发送，因为窃听的攻击者可能会在途中观察到它。因此，必须采用一些其他机制。

在某些情况下，双方可以访问一个**安全通道**，他们可以利用该通道可靠地共享一个秘密密钥。一个常见的例子是双方在某个时间点**位于同一地点**，此时他们可以共享密钥。或者，双方可以使用**受信任的信使服务**作为安全通道。我们强调，即使双方可以在某个时间点共享密钥（因此必定可以访问安全通道），私钥密码学仍然是有用的：在第一个例子中，双方在某个时间点拥有一个安全通道，但并非永久拥有；在第二个例子中，使用安全通道可能比通过不安全通道通信更慢、成本更高。

上述方法已被政府、外交和军事领域用于共享密钥。例如，20 世纪 60 年代连接莫斯科和华盛顿的“红线电话”就是用一次性密码本加密的，密钥由携带公文包文件的信使在两国之间交换 。类似的方法也可用于企业，例如在**新员工上岗第一天**建立其与中央数据库之间的共享密钥。（我们将在下一节重新讨论这个例子。）

然而，依赖安全通道来分发密钥在许多其他情况下并不可行。例如，考虑一个大型跨国公司，其中每对员工都需要安全通信的能力，并且他们的通信需要对其他员工保密。要求每对员工见面以便安全地共享密钥，至少是不方便的；对于在不同城市工作的员工来说，这甚至可能是不可能的。即使当前所有员工能够以某种方式彼此共享密钥，但对于在初次共享之后加入的新员工，再与他们共享密钥也是不切实际的。

假设这 N 名员工能够安全地相互共享密钥，另一个显著的缺点是每位员工将不得不管理和存储 **N - 1 个秘密密钥**（每位员工一个）。事实上，这可能大大低估了每位用户存储的密钥数量，因为员工可能还需要密钥来与数据库、服务器、打印机等远程资源安全通信。如此多的秘密密钥的扩散是一个重大的后勤问题。此外，所有这些密钥都必须**安全存储**。密钥越多，保护它们就越困难，密钥被攻击者窃取的几率就越高。计算机系统经常受到病毒、蠕虫和其他形式的恶意软件感染，这些软件可以窃取秘密密钥并悄悄地通过网络发送给攻击者。因此，将密钥存储在员工的个

人计算机上往往是不安全的。

诚然，秘密密钥可能被泄露，这始终是一个问题，无论各方持有多少密钥。然而，当只需要存储少量密钥时，有很好的解决方案来应对这种威胁。目前，一种典型的解决方案是将密钥存储在**安全硬件**上，例如**智能卡**。智能卡可以使用存储的秘密密钥执行加密计算，确保这些密钥永远不会进入用户的个人计算机。如果设计得当，智能卡比个人计算机更能抵抗攻击——例如，它通常不会被恶意软件感染——因此提供了一种很好的保护用户秘密密钥的方法。不幸的是，智能卡在内存方面通常非常有限，因此无法存储数百个（或数千个）密钥。

上述所有问题都可以**原则上**得到解决（即使在实践中不一定能），只要是在由明确的用户组成，且所有用户都愿意遵循相同的密钥分发和存储策略的**“封闭式”组织**中。然而，在**“开放式系统”**中，这些方法就会失效。在开放系统中，用户之间的交互是短暂的，无法安排面对面的会面，甚至在他们想要通信之前可能都不知道彼此的存在。在许多情况下，这种情况比我们最初想象的要普遍：例如，使用加密技术向您从未购买过商品的互联网商家发送信用卡信息，或者向您从未见过的人发送电子邮件。在这些情况下，仅仅依靠私钥密码学并不能解决问题，我们必须寻求其他解决方案。

总而言之，使用私钥密码学至少存在三个明显的问题：第一个是**密钥分发**问题；第二个是存储和管理大量秘密密钥的**密钥管理**问题；第三个是私钥密码学**不适用于开放系统**的问题。

10.2 部分解决方案：密钥分发中心 (KDC)

解决上述部分问题的一种方法是使用**密钥分发中心 (KDC)** 来建立共享密钥。再次考虑一个大型公司，其中每对员工都必须能够安全通信的情况。在这种设置中，我们可以利用所有员工都信任某个实体（例如系统管理员）的事实，至少在与工作相关的信息安全方面如此。这个**受信任的实体**就可以充当 KDC，帮助所有员工共享成对密钥。

当新员工加入时，KDC 可以在该员工上岗第一天（当面，在安全地点）与该员工共享一个密钥。同时，KDC 也可以在该员工与所有现有员工之间分发共享密钥。也就是说，当第 i 名员工加入时，KDC（除了在该新员工与 KDC 之间共享密钥外）可以生成 i - 1 个密钥 $k_1, \ldots, k_{i-1}$，将这些密钥交给新员工，然后将密钥 k_j 使用第 j 名现有员工与 KDC 共享的密钥加密后发送给该员工。此后，新员工就与所有其他员工（以及 KDC）共享一个密钥。

一种更好的方法是，利用 KDC **在线**地（“按需”）生成密钥，每当两名员工希望安全通信时就进行。与之前一样，KDC 将与每位员工共享一个（不同的）密钥，这可以在员工上岗第一天安全地完成。假设 KDC 与员工 Alice 共享密钥 k_A，与员工 Bob 共享密钥 k_B。在稍后的某个时间，当 Alice 希望与 Bob 安全通信时，她只需向 KDC 发送消息：“我，Alice，想与 Bob 通

话。”（如果需要，可以使用 Alice 与 KDC 共享的密钥对该消息进行认证。）然后，KDC 选择一个随机的**会话密钥** k，并将其发送给 Alice（用 k_A 加密）和 Bob（用 k_B 加密）。（该协议过于简单，不能在实践中使用；详见下文。）一旦 Alice 和 Bob 都恢复了这个会话密钥，他们就可以用它来安全通信。当他们的对话结束后，他们可以（并且应该）删除会话密钥，因为如果他们希望在稍后某个时间再次通信，可以随时再次联系 KDC。

考虑这种方法的优势：

1. 每位员工只需存储**一个长期秘密密钥**（即他们与 KDC 共享的密钥）。员工仍然需要管理和存储会话密钥，但这些是短期密钥，一旦通信会话结束就会被删除。 KDC 需要存储许多长期密钥。但是，KDC 可以放置在安全位置，并得到最高级别的网络攻击防护。
2. 当员工加入组织时，所需要做的就是在这名员工与 KDC 之间建立一个密钥。**不需要**其他员工更新他们持有的密钥集合。

因此，KDC 可以缓解我们已经看到的关于私钥密码学的两个问题：它们可以**简化密钥分发**（因为员工加入时只需共享一个新密钥，并且可以合理地假设 KDC 与该员工在上岗第一天之间存在安全通道），并且可以**降低密钥存储的复杂性**（因为每位员工只需存储一个密钥）。KDC 在使私钥密码学在所有员工都信任同一实体的**大型组织**中变得实用方面发挥了重要作用。

然而，依赖 KDC 也存在一些缺点：

1. 对 **KDC 的成功攻击**将导致整个系统被彻底攻破：攻击者可以泄露所有密钥，随后窃听所有网络流量。这使得 KDC 成为一个高价值的攻击目标。请注意，即使 KDC 受到良好的外部攻击保护，也始终存在内部人员（例如 IT 经理）攻击 KDC 的可能性。
2. KDC 是一个**单点故障**：如果 KDC 停机，安全通信将暂时无法进行。如果员工不断联系 KDC 并请求建立会话密钥，KDC 上的负载可能会非常高，从而增加其故障或响应缓慢的可能性。

解决第二个问题的简单方法是**复制 KDC**。这可行（并且在实践中也这样做），但也意味着系统现在有更多的攻击点。增加更多的 KDC 也使得添加新员工更加困难，因为必须安全地将更新传播到每个 KDC。

使用 KDC 进行密钥分发的协议

文献中有许多使用 KDC 进行安全密钥分发的协议。我们特别提到 **Needham–Schroeder 协议**，它是 **Kerberos** 的核心。Kerberos 是一种重要且广泛使用的服务，用于执行身份认证和支持安全通信。（Kerberos 在许多大学和公司中使用，是 Windows 和许多 UNIX 系统中支持安全网络身份认证和通信的默认机制。）

我们只强调该协议的一个特点。当 Alice 联系 KDC 并请求与 Bob 通信时，KDC 不会像我们前面描述的那样将加密的会话密钥发送给 Alice 和 Bob。相反，KDC **将用 Alice 的密钥加密的会话密钥以及用 Bob 的密钥加密的会话密钥一起发送给 Alice**。然后 Alice 将第二个密文转发给 Bob，如图 10.1 所示。第二个密文有时被称为**“票据”（ticket）**，可以被视为允许 Alice 与 Bob 通话的凭证（并允许 Bob 确信他正在与 Alice 通话）。事实上，正如我们之前没有强调的那样，基于 KDC 的方法也可以作为一种有用的**身份认证**机制。另请注意，Alice 和 Bob 不必都是用户；Alice 可以是用户，Bob 可以是服务器、远程磁盘或打印机等资源。

设计该协议的目的是**减轻 KDC 的负载**。在所描述的协议中，KDC 不需要启动与 Bob 的第二次连接，并且不需要担心当 Alice 启动协议时 Bob 是否在线。此外，如果 Alice 保留票据（和她自己的会话密钥副本），那么她可以通过简单地将票据重新发送给 Bob 来重新建立安全通信，**而无需 KDC 的参与**。（在实践中，票据会过期，最终需要续订。但会话可以在某个可接受的时间段内重新建立。）

最后，我们指出，在实践中，Alice 与 KDC 共享的密钥可能是一个简短、易记的**密码**。在这种情况下，会出现许多必须解决的额外安全问题。我们也一直隐含地假设攻击者是**被动窃听者**，而不是可能主动尝试干扰协议的攻击者。对于如何解决这些问题，我们建议感兴趣的读者参考本章末尾的参考文献。

10.3 密钥交换与迪菲–赫尔曼协议

KDC 和 Kerberos 等协议在实践中很常用。但这些解决密钥分发问题的方法仍然需要在某个时间点有一个**私密且经过认证的通道**来共享密钥。（特别地，我们假设在 KDC 与员工上岗第一天之间存在这样一个通道。）因此，它们仍然无法解决 Internet 等开放系统中的密钥分发问题，在这些系统中，希望通信的两个用户之间可能不存在任何私密通道。

为了在**从不通过私密通道通信**的情况下实现私密通信，需要一种截然不同的方法。1976 年，Whitfield Diffie 和 Martin Hellman 发表了一篇标题看似无害的论文**《密码学的新方向》（*New Directions in Cryptography*）。他们观察到，世界往往存在**不对称性**；特别是，有些操作可以**轻松执行**，但**难以逆转**。例如，挂锁可以无需钥匙上锁（即很容易），但之后没有钥匙则无法打开。更引人注目的是，打碎一个玻璃花瓶很容易，但将其重新组合起来却极其困难。从算法角度（对我们更有意义），将两个大素数相乘很容易，但从它们的乘积中找出它们却是困难的。（这正是我们在前几章中讨论的**因子分解问题**。）Diffie 和 Hellman 意识到，这种现象可以用于推导出**安全密钥交换的交互式协议**，允许双方通过在**公共通道**上的通信来共享一个秘密密钥，方法是让双方执行一些他们自己可以逆转但窃听者不能的操作。

安全密钥交换协议的存在是相当令人惊奇的。这意味着您和一位朋友可以通过在房间里喊话（并

执行一些本地计算）来商定一个秘密；即使其他人都听到了所有的对话内容，这个秘密对于他们来说仍然是未知的。事实上，直到 1976 年，人们普遍认为，如果不首先使用私密通道共享一些秘密信息，就无法实现安全通信。

Diffie 和 Hellman 的论文产生了巨大的影响。除了引入一种看待密码学的全新方式外，它还是将密码学从私人领域推向公共领域的第一步。我们引用他们论文的前两段：

> “我们今天正站在密码学革命的边缘。廉价数字硬件的发展使密码学摆脱了机械计算的设计限制，并将高等级加密设备的成本降低到可用于远程自动取款机和计算机终端等商业应用的程度。
>
> 反过来，此类应用产生了对新型加密系统的需求，这些系统最大限度地减少了对安全密钥分发通道的依赖……与此同时，信息论和计算机科学的理论发展有望提供**可证明安全**的加密系统，将这门古老的艺术转变为一门科学。”

Diffie 和 Hellman 并非言过其实，他们所说的革命很大程度上归功于他们的工作。

在本节中，我们将介绍 **迪菲–赫尔曼密钥交换协议**。我们将证明其对窃听攻击者是安全的，或者等效地，在假设通信双方通过公共但经过认证的通道通信（因此攻击者不能干扰他们的通信）的情况下，该协议是安全的。对窃听攻击者是安全的，这是一种相对较弱的安全保证，在实践中，密钥交换协议必须满足更强的安全要求，这超出了我们目前的范围。（此外，我们关注的是通信双方事先没有任何共享信息的场景，在这种情况下，无法阻止攻击者冒充其中一方。我们稍后会回到这一点。）

场景与安全定义

我们考虑一个场景，其中两方（传统上称为 Alice 和 Bob）运行一个概率协议 Π，以生成一个共享秘密密钥；Π 可以被视为 Alice 和 Bob 在协议中的指令集。Alice 和 Bob 首先持有安全参数 1^n；然后他们使用（独立的）随机比特运行 Π。在协议结束时，Alice 和 Bob 分别输出密钥 $k_A, k_B \in \{0,1\}^n$。基本的正确性要求是 $k_A = k_B$。由于我们只处理满足这一要求的协议，因此我们只讨论诚实执行 Π 所生成的密钥 $k = k_A = k_B$。

现在我们转向定义安全性。直观地说，如果 Alice 和 Bob 输出的密钥对于窃听攻击者来说是完全不可猜测的，那么密钥交换协议就是安全的。形式上，我们要求，对协议执行进行窃听的攻击者，**无法区分**该执行生成的密钥 k（现在由 Alice 和 Bob 共享）与长度为 n 的**均匀随机密钥**。这比仅仅要求攻击者无法准确计算出 k 要强得多，并且如果双方随后将 k 用于某些加密应用（例如作为私钥加密方案的密钥），这种更强的要求是必要的。

为了形式化上述内容，设 Π 是一个密钥交换协议，A 是一个攻击者，n 是安全参数。我们有以

下实验：

密钥交换实验 $\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)$：

1. 持有 1^n 的双方执行协议 Π。这会产生一个包含双方发送的所有消息的**通信记录** trans，以及双方输出的密钥 k。
2. 选择一个均匀比特 $b \in \{0,1\}$。如果 $b = 0$，设置 $\hat{k} := k$；如果 $b = 1$，则均匀随机选择 $\hat{k} \in \{0,1\}^n$。
3. 将 trans 和 $\hat{k}$ 提供给 A，A 输出一个比特 b'。
4. 当 $b' = b$ 时，实验的输出定义为 1，否则为 0。（如果 $\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1$，我们称 A 成功。）

A 获得 trans 捕获了 A 窃听协议的整个执行过程并因此看到双方交换的所有消息这一事实。在现实世界中，A 不会获得任何密钥；在实验中，向攻击者提供 $\hat{k}$ 仅仅是为了定义 A “攻破” Π 的含义。也就是说，如果攻击者能够正确判断 $\hat{k}$ 是对应于给定协议执行的**真实密钥**，还是独立于通信记录的**均匀随机密钥**，那么攻击者就成功“攻破”了 Π 的安全性。

正如预期的那样，我们说 Π 是安全的，如果攻击者成功的概率至多**略大于 1/2**。也就是说：

定义 10.1 一个密钥交换协议 Π 是在**窃听者存在下安全**的，如果对于所有概率多项式时间攻击者 A，存在一个可忽略函数 negl，使得：

$$\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathrm{negl}(n)$$

密钥交换协议的目的是生成一个共享密钥 k，双方随后将使用该密钥进行某些加密应用（例如，使用认证加密方案加密和认证他们的后续通信）。直观上，使用安全密钥交换协议生成的共享密钥应该“等同于”使用通过私密通道共享的密钥。这可以形式化证明；详见练习 10.1。

迪菲–赫尔曼密钥交换协议

现在我们描述 Diffie 和 Hellman 在其原始论文中提出的密钥交换协议（尽管他们当时不如我们现在形式化）。设 $\mathcal{G}$ 是一个概率多项式时间算法，它接受 1^n 作为输入，并（除了可能以可忽略的概率）输出一个循环群 G 的描述、其阶 q（其中 $|q| = n$）以及一个生成元 $g \in G$。（详见第 8.3.2 节。）迪菲–赫尔曼密钥交换协议的形式描述如构建 10.2 所示，图 10.2 对其进行了说明。

构建 10.2 迪菲–赫尔曼密钥交换协议

- **共同输入：** 安全参数 1^n

- **协议：**
  i. Alice 运行 $\mathcal{G}(1^n)$ 以获得 (G, q, g)。
  ii. Alice 选择一个均匀 $x \in \mathbb{Z}_q$，并计算 $h_A := g^x$。
  iii. Alice 将 (G, q, g, h_A) 发送给 Bob。
  iv. Bob 接收 (G, q, g, h_A)。他选择一个均匀 $y \in \mathbb{Z}_q$，并计算 $h_B := g^y$。Bob 将 h_B 发送给 Alice，并输出密钥 $k_B := h_A^y$。
  v. Alice 接收 h_B，并输出密钥 $k_A := h_B^x$。

在我们的描述中，我们假设 Alice 生成 (G, q, g) 参数并在她的第一条消息中将它们发送给 Bob。在实践中，这些参数是标准化的，并且在协议开始之前就已确定并为双方所知。在这种情况下，Alice 只需发送 h_A，Bob 也无需等待接收 Alice 的消息就可以计算并发送 h_B。

不难看出该协议是**正确**的：Bob 计算密钥 $k_B = h_A^y = (g^x)^y = g^{xy}$，而 Alice 计算密钥 $k_A = h_B^x = (g^y)^x = g^{xy}$，因此 $k_A = k_B$。（细心的读者会注意到共享密钥是一个群元素，而不是一个比特串。我们稍后会回到这一点。）

Diffie 和 Hellman 没有证明他们协议的安全性；事实上，当时还没有适当的安全概念（无论是定义框架还是形式化精确假设的想法）。让我们看看该协议需要什么样的假设才能是安全的。Diffie 和 Hellman 提出的第一个观察是，安全性的最低要求是**离散对数问题**相对于 $\mathcal{G}$ 是困难的。如果不是，那么获得通信记录（其中特别包括 h_A）的攻击者可以计算出其中一方的秘密值（即 x），然后利用该值轻松计算出共享密钥。因此，离散对数问题的困难性是协议安全性的必要条件。然而，它不是充分条件，因为可能存在其他方法可以在不显式计算 x 或 y 的情况下计算出密钥 $k_A = k_B$。**计算迪菲–赫尔曼假设**（CDH 假设）——它只保证在给定 g, g^x, g^y 的情况下，计算出密钥 g^{xy} 是困难的——也不足以保证协议的安全性。

定义 10.1 所要求的是，对于给定 g, g^x, g^y 的任何攻击者，共享密钥 g^{xy} 应该**与均匀随机密钥无法区分**。这正是第 8.3.2 节中引入的**判定迪菲–赫尔曼假设**（DDH 假设）。

正如我们将看到的，该协议的安全性证明几乎可以直接从 DDH 假设得出。这不应该令人惊讶，因为迪菲–赫尔曼假设是在 Diffie 和 Hellman 发表论文之后，作为一种**抽象**迪菲–赫尔曼协议（推测的）安全性的基础特性而引入的。鉴于此，质疑在此定义和证明安全性是否有意义是合理的。然而，通过本书的学习，希望您已确信答案是肯定的。精确地定义安全密钥交换迫使我们思考我们到底需要什么样的安全属性；指定一个精确的假设（即 DDH 假设）意味着我们可以独立于任何特定应用来研究该假设，并且——一旦我们确信其合理性——可以基于它构建其他协议；最后，证明安全性表明该假设确实足以使协议满足我们期望的安全要求。

在我们的安全性证明中，我们使用了一个修改后的定义 10.1 版本，其中要求共享密钥**与均匀群**

元素不可区分，而不是与均匀 n 比特串不可区分。在协议应用于实践之前，需要解决这种差异——毕竟，群元素通常不能直接用作加密密钥，而且均匀群元素的表示形式通常也不是均匀比特串——在证明之后，我们将简要讨论一种标准的处理方法。现在，我们让 $\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)$ 表示修改后的实验，其中如果 $b = 1$，攻击者获得的是从 G 中均匀选择的 $\hat{k}$，而不是均匀 n 比特串。

定理 10.3 如果 DDH 问题相对于 $\mathcal{G}$ 是困难的，那么迪菲–赫尔曼密钥交换协议 Π 在窃听者存在下是安全的（相对于修改后的实验 $\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}$）。

证明

设 A 是一个概率多项式时间攻击者。由于 $\Pr[b = 0] = \Pr[b = 1] = 1/2$，我们有

$$\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1\right] = \frac{1}{2} \cdot \Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1 \mid b = 0\right] + \frac{1}{2} \cdot \Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1 \mid b = 1\right]$$

在实验 $\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)$ 中，攻击者 A 接收 $(\mathrm{G}, q, g, h_A, h_B, \hat{k})$，其中 $(\mathrm{G}, q, g, h_A, h_B)$ 代表协议执行的通信记录，$\hat{k}$ 要么是双方计算出的实际密钥（如果 $b = 0$），要么是均匀群元素（如果 $b = 1$）。区分这两种情况**恰好等同于**解决 DDH 问题。也就是说

$$\begin{aligned}
\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1\right] &= \frac{1}{2} \cdot \Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1 \mid b = 0\right] + \frac{1}{2} \cdot \Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1 \mid b = 1\right] \\
&= \frac{1}{2} \cdot \Pr\left[A(G, g, q, g^x, g^y, g^{xy}) = 0\right] + \frac{1}{2} \cdot \Pr\left[A(G, q, g, g^x, g^y, g^z) = 1\right] \\
&= \frac{1}{2} \cdot \left(1 - \Pr\left[A(G, g, q, g^x, g^y, g^{xy}) = 1\right]\right) + \frac{1}{2} \cdot \Pr\left[A(G, q, g, g^x, g^y, g^z) = 1\right] \\
&= \frac{1}{2} + \frac{1}{2} \cdot \left(\Pr\left[A(G, q, g, g^x, g^y, g^z) = 1\right] - \Pr\left[A(G, g, q, g^x, g^y, g^{xy}) = 1\right]\right) \\
&\leq \frac{1}{2} + \frac{1}{2} \cdot \left|\Pr\left[A(G, q, g, g^x, g^y, g^z) = 1\right] - \Pr\left[A(G, g, q, g^x, g^y, g^{xy}) = 1\right]\right|
\end{aligned}$$

其中概率是在 $\mathcal{G}(1^n)$ 输出 (G, q, g) 以及均匀选择 $x, y, z \in \mathbb{Z}_q$ 的实验中计算的。（注意，当 z 在 $\mathbb{Z}_q$ 中均匀分布时，g^z 在 G 中均匀分布。）如果 DDH 假设相对于 $\mathcal{G}$ 是困难的，则恰好意味着存在一个可忽略函数 negl，使得

$$\left|\Pr\left[A(G, q, g, g^x, g^y, g^z) = 1\right] - \Pr\left[A(G, g, q, g^x, g^y, g^{xy}) = 1\right]\right| \leq \mathrm{negl}(n)$$

我们得出 $\Pr\left[\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \leq 1/2 + 1/2 \cdot \mathrm{negl}(n)$，完成了证明。

均匀群元素 vs. 均匀比特串

上一定理表明，Diffie–Hellman 协议输出的密钥与均匀群元素不可区分（对于多项式时间窃听者）。为了将该密钥用于后续的加密应用——并满足定义 10.1——双方输出的密钥应该与适当长度的**均匀比特串**不可区分。Diffie–Hellman 协议可以通过让双方对他们各自计算出的共享群元素 g^{xy} 应用适当的**密钥导出函数**（参见第 5.6.4 节）来达到这一目的。

主动攻击者

到目前为止，我们只考虑了窃听攻击者。虽然窃听攻击是最常见的攻击（因为它们最容易执行），但它们绝不是唯一的可能攻击。**主动攻击**（其中攻击者向其中一方或双方发送自己的消息）也是一个问题，在实践中使用的任何协议都必须能够抵御此类攻击。在考虑主动攻击时，区分**冒充攻击**（其中攻击者在与另一方交互时冒充其中一方）和**中间人攻击**（其中通信的双方都在执行协议，而攻击者正在拦截和修改从一方发送给另一方的消息）是很有用的。我们不会形式化定义针对任何一类攻击的安全性，因为此类定义非常复杂，并且除非双方事先共享一些信息，否则无法实现。（我们稍后会回到这一点。）然而，值得注意的是，**迪菲–赫尔曼协议完全容易受到中间人攻击**。事实上，中间人攻击者可以以这样一种方式行事：Alice 和 Bob 终止协议时持有不同的密钥 k_A 和 k_B，**而这两个密钥都为攻击者所知**，但 Alice 和 Bob 都没有检测到任何攻击发生。我们将该攻击的详细描述留作练习。

迪菲–赫尔曼密钥交换的实践应用

由于存在上述中间人攻击的不安全性，基本形式的迪菲–赫尔曼协议通常不在实践中使用。但这丝毫不减损其重要性。迪菲–赫尔曼协议是第一个证明**非对称技术**（和数论问题）可用于缓解密码学中密钥分发问题的方法。此外，迪菲–赫尔曼协议是目前广泛使用的**可抵御中间人攻击**的标准化密钥交换协议的核心。一个值得注意的例子是 **TLS**；详见第 12.8 节。

10.4 公钥革命

除了密钥交换之外，Diffie 和 Hellman 在他们的开创性工作中还引入了**公钥（或非对称）密码学**的概念。在公钥设置中（与我们在第 1 章至第 7 章中研究的私钥设置不同），希望安全通信的一方生成一对密钥：一个**公钥**，广泛传播；一个**私钥**，自己保密。（现在有两个不同的密钥，这是该方案非对称的原因。）生成这些密钥后，一方可以使用它们通过**公钥加密方案**来确保其接收到的消息的机密性，或通过**数字签名方案**来确保其发送的消息的完整性。（详见图 10.3。）我们在此提供这些原语的简要介绍，并将在第 11 章和第 12 章中详细讨论它们。

在**公钥加密方案**中，一方生成的公钥充当**加密密钥**；任何知道该公钥的人都可以使用它来加密消息并生成相应的密文。**私钥**充当**解密密钥**，由知道该私钥的一方使用，以从使用匹配公钥生成的任何密文中恢复原始消息。此外——令人惊奇的是竟然存在这样的东西！——即使攻击者知道加

密密钥（但不知道解密密钥），加密消息的机密性仍然得以保持。换句话说，对于试图解密使用该密钥加密的密文的攻击者来说，（公共的）加密密钥是毫无用处的。因此，为了实现秘密通信，接收者只需将其公钥发送给潜在发送者（无需担心观察到其公钥的窃听攻击者），或者在其网页上或公共目录中公布其公钥。公钥加密方案因此可以在**不依赖私密通道**进行密钥分发的情况下实现私密通信。

数字签名方案是消息认证码（MAC）的公钥模拟。在这里，私钥充当**“认证密钥”**（更通常称为**签名密钥**），允许知道该密钥的一方为其发送的消息生成**“认证标签”**（即**签名**）。公钥充当**验证密钥**，允许知道该密钥的任何一方验证发送者发布的签名。与 MAC 一样，数字签名方案可用于防止对消息的未被检测到的篡改。然而，任何知道发送者公钥的人都可以进行验证这一事实，具有深远的意义。具体来说，它使得将 Alice 签名的文档出示给第三方（例如法官），作为 Alice 确实签署了该文档的证据成为可能。这一属性被称为**不可否认性**，在电子商务中有着广泛的应用。例如，可以对合同进行数字签名、发送已签名的电子采购订单或付款承诺等。数字签名也用于安全分发公钥，作为**公钥基础设施**的一部分，我们将在第 12.7 节中对此进行更详细的讨论。

Diffie 和 Hellman 在他们的论文中提出了公钥密码学的概念，但没有给出任何候选构建方案。一年后，Ron Rivest、Adi Shamir 和 Len Adleman 提出了 **RSA 问题**，并提出了基于该问题困难性的第一个公钥加密和数字签名方案。他们的方案的变体是当今使用最广泛的加密原语之一。1985 年，Taher El Gamal 提出了一种加密方案，它本质上是对迪菲–赫尔曼密钥交换协议的略微修改，现在也得到了广泛应用。因此，尽管 Diffie 和 Hellman 没有成功构建一个（非交互式）公钥加密方案，但他们非常接近了。

最后，我们总结公钥密码学如何解决第 10.1 节中讨论的私钥设置的局限性：

1. **公钥加密**允许通过**公共（但经过认证的）通道**进行密钥分发。这可以简化密钥材料的分发和更新。
2. **公钥密码学**减少了用户存储许多秘密密钥的需求。再次考虑一个大型公司，其中每对员工都需要安全通信能力的环境。使用公钥密码学，每位员工只需存储一个私钥（自己的）以及所有其他员工的**公钥**就足够了。重要的是，后者的密钥**不需要秘密存储**；它们甚至可以存储在某个中央（公共）存储库中。
3. 最后，**公钥密码学**更适用于**开放环境**，在这些环境中，从未交互过的双方希望能够安全通信。举一个常见的例子，互联网商家可以在线发布其公钥；用户可以在需要加密信用卡信息时获取商家的公钥。

公钥加密的发明是密码学的一场革命。密码学及其应用在 20 世纪 70 年代末和 80 年代初之前一直属于情报和军事组织的领域，直到公钥技术的出现，密码学的应用才扩展到大众，这不是巧

合。