第 4 章

协议

聪明人的后果是无法预见的。
– 克里斯托弗·斯特雷奇

如果可以证明它是安全的,那么它可能不是。
– 拉斯·克努森

4.1 简介

密码只是更普遍的概念（安全协议)的一个示例。
如果安全工程有一个核心主题,那可能就是对安全协议的研究。它们指定委托人用于建立信任关系的步骤。

它们是密码学和访问控制相交的地方;它们是我们用来将人类用户与远程机器连接起来、同步安全上下文以及调节支付等关键应用程序的工具。我们已经遇到了一些协议,包括质询-响应身份验证和 Kerberos。在本章中,我将深入细节,并给出许多协议如何失败的例子。

一个典型的安全系统由许多主体组成,例如人、公司、电话、计算机和读卡器,它们使用各种渠道进行通信,包括光纤、wi i、蜂窝网络、蓝牙、红外线,并通过在物理设备上传输数据比如银行卡和交通票。安全协议是管理这些通信的规则。

它们的设计使系统能够经受住恶意行为,例如人们在电话中说谎、敌对政府干扰无线电或伪造更改火车票上的数据。防范所有可能的攻击通常过于昂贵,因此协议设计会对威胁做出假设。例如,当我们通过在一台机器上输入密码让用户登录时,我们隐含地假设她可以将密码输入正确的机器。在工作场所硬连线终端的过去,这是合理的;现在人们通过 Internet 登录网站,它就不那么明显了。因此,评估协议涉及两个问题:首先,威胁模型是否现实?第二,协议是否处理它?

协议可能非常简单,例如通过阅读器刷卡进入建筑物。它们通常涉及交互,不一定是技术性的。例如,当我们在餐厅点一瓶好酒时,标准协议是酒水服务员给我们提供菜单(这样我们可以看到价格但客人看不到);他们带来了瓶子,所以我们可以检查标签、密封和温度;他们打开它让我们尝尝;然后上菜。这已经演变为提供一些隐私(我们的客人不知道价格)、一些完整性(我们可以确定我们得到了正确的瓶子,并且没有用廉价的 plonk 重新装满)和不可否认性(我们不能事后抱怨酒是o)。 Matt Blaze 在 [260] 中给出了来自机票检查、航空安全和投票的其他非技术协议示例。像这样的传统协议通常经过几十年或几个世纪的发展,以满足社会期望和技术威胁。

在技术方面,协议变得更加复杂,而且它们并不总是变得更好。随着汽车行业从金属钥匙转向带有按钮的电子钥匙,盗窃事件开始减少,因为新钥匙更难复制。

但是转向无钥匙进入已经看到汽车犯罪再次上升,因为坏人想出了如何制造中继设备,使钥匙看起来比实际更接近汽车。另一个被证明是棘手的安全升级是从磁条卡到智能卡的转变。欧洲在 2000 年代后期采取了这一举措,而美国在 2010 年代后期才迎头赶上。

针对欧洲发行的信用卡的欺诈行为实际上持续了数年;美国的磁条取款机使用了欧洲卡片的克隆版,因为这两个系统的保护机制并不完全吻合。还有一个协议故障,即使小偷不知道 PIN 码,也会让小偷在商店中使用偷来的芯片卡,银行花了几年时间才修复。

因此,我们需要系统地研究安全协议以及它们是如何失败的。

4.2 密码窃听风险

密码和 PIN 仍然是大部分计算机安全的基础,作为用于验证人机身份的主要机制。我们在上一章讨论了它们的可用性;现在让我们考虑一下在设计在一台机器和另一台机器之间运行的协议时我们必须阻止的技术攻击的种类。

远程键输入是一个很好的起点。早期的系统,例如用于打开您的车库或解锁直到 1990 年代中期制造的汽车的遥控器,只是广播一个序列号。杀死他们的攻击是“抓取器”,这是一种可以记录代码并稍后重播的设备。第一批 grabbers 似乎来自台湾,于 1995 年左右上市;小偷会潜伏在停车场或目标房屋外,记录用于锁车的信号,然后在车主离开后重播 1。

1有了车库门,情况就更糟了。常见的芯片是 Princeton PT2262,它使用 12 个三态引脚来编码312或531,441 地址代码。然而,实施者通常没有足够仔细地阅读数据表来理解三态输入,而是将它们视为二进制,得到212。其中许多只使用八个输入,因为其他四个在芯片的另一侧。由于芯片没有重试锁定逻辑,攻击者可以循环通过

第一个对策是使用单独的代码进行锁定和解锁。
但是小偷可以潜伏在你家外面,在你早上开车离开之前记录下解锁码,然后晚上回来自救。

其次,十六位密码太短。偶尔人们会发现他们可能错误地解锁了错误的汽车,或者甚至在车主不知道他有一辆汽车上设置了警报 [308]。到 20 世纪 90 年代中期,出现了可以一个接一个地尝试所有可能密码的设备。在大约215次尝试后,平均可以找到一个代码,并且每秒 10 个代码需要不到一个小时。

在范围内有一百辆车的停车场偷窃的小偷将在不到一分钟的时间内得到一辆有用的闪光灯的汽车作为奖励。

下一个对策是将密码的长度从 16 位增加到 32 位。制造商自豪地宣传“超过 40 亿个代码”。但这只能说明他们还没有真正理解问题所在。每辆车仍然只有一两个代码,抓取器仍然可以正常工作。

使用序列号作为密码还有一个弱点:很多人都可以访问它。就汽车而言,这可能意味着所有经销商员工 ，也可能是州机动车登记机构。一些防盗报警器还使用序列号作为主密码,这里更糟糕的是:当银行购买防盗报警器时,序列号可能会出现在订单、送货单和发票上。银行不喜欢派人出去买东西换现金。

简单的密码有时是合适的技术。例如,我们当地游泳池的月票只有一个条形码。我敢肯定我可以伪造一个还过得去的伪造品,但随着旋转门服务员逐渐了解“常客”,就没有必要再买更贵的东西了。然而,对于在线的事物,静态密码是危险的； Mirai 僵尸网络通过招募具有无法更改密码的 wi i 连接闭路电视摄像机而开始运作。对于人们想偷的东西,比如汽车,我们还需要更好的东西。这将我们带到了密码认证协议。

4.3 谁去那里？ – 简单的身份验证

一个简单的现代认证设备是一些多层停车场给订户提高门槛的令牌。令牌只有一个按钮;当你按下它时,它首先传输它的序列号,然后发送一个由相同序列号组成的验证块,然后是一个随机数,所有这些都使用设备唯一的密钥加密,然后发送到车库屏障（通常由434MHz 的无线电,但也使用红外线）。我们将把如何加密数据的讨论推迟到下一章,对用密钥 K 加密的消息 X简单地写成 {X}K。

然后访问令牌和停车场之间的协议可以是
写成:

吨！ G : T, {T,N}KT快速

组合并平均尝试27次后打开车库门。我在本书第二版中注意到这些问题十二年后,芯片还没有被撤回。它现在也用于家庭安全系统和玩具的遥控。

这是标准的协议表示法,所以我们慢慢来。

令牌 T 向车库 G 发送一条消息,该消息由其名称 T 后跟 T 的加密值和 N 连接而成,其中 N 代表“使用过一次的号码”或随机数。大括号内的所有内容都经过加密,加密将 T 和 N 绑定在一起并隐藏了它们的值。nonce 的目的是向收件人保证消息是最新的,也就是说,它不是旧消息的重播。验证很简单:车库读取T,得到对应的密钥KT,解密消息的其余部分,检查nonce N之前是否见过,最后明文包含T。

许多人感到困惑的原因之一是,在冒号的左侧,T 标识主体之一（代表订阅者的令牌）,而在右侧,它表示令牌的名称（即唯一设备编号）。

另一个是,一旦我们开始讨论对协议的攻击,我们可能会发现发给一个委托人的消息被另一个委托人截获并回放了。
所以你可能会想到 T !冒号左边的 G 暗示协议设计者的想法。

随机数可以是保证消息新鲜度的任何东西。它可以是随机数、计数器、从第三方收到的随机挑战,甚至是时间戳。它们之间存在细微差别 ,例如它们对各种重放攻击的抵抗程度,以及它们增加系统成本和复杂性的方式。在成本非常低的系统中,随机数和计数器占主导地位,因为仅在一个方向上进行通信更便宜,而且便宜的设备通常没有时钟。

此类设备中的密钥管理可以非常简单。在典型的车库代币产品中,每个代币的密钥只是其唯一的设备编号,在车库已知的全局主密钥 KM 下加密:

$$KT = \{T\}_{公里}$$

这称为密钥多样化或密钥派生。这是实现访问令牌的常用方法,也广泛用于智能卡。目标是通过钻入令牌并提取密钥来破坏令牌的人不能伪装成任何其他令牌;他所能做的就是复制一个特定订阅者的令牌。为了完全破解系统,并提取主密钥,使他能够伪装成系统的任何用户,攻击者必须破坏车库的中央服务器（这可能会在篡改中保护此密钥抗性智能卡或硬件安全模块）。

但仍有出错的余地。一个常见的故障模式是序列号 无论是唯一的设备编号还是协议计数器 不够长,以至于有人偶尔会发现他们的遥控器也适用于停车场中的另一辆车。这可以通过密码学来掩盖。

如果密钥是通过加密 16 位设备号或通过采用 16 位密钥并重复八次来派生的,那么拥有 128 位密钥也无济于事。在任何一种情况下,只有216个可能的密钥,即使

它们似乎是随机的 2。

与密码漏洞相比，协议漏洞通常会引发更多、更简单的攻击。一个例子来自预付费公用事业仪表的世界。英国超过 100 万户家庭以及发展中国家超过 4 亿户家庭拥有接受加密令牌的电表或煤气表：户主购买一个幻数并将其输入电表，然后电表会分配购买的电量。一种在南非广泛使用的早期计量器只检查了 nonce 是否与上次不同。因此，客户可以通过购买两张低价值的电力票，然后一张接一张地给他们的电表无限期地充电；给定两个有效代码 A 和 B，系列 ABABAB... 被视为有效 [93]。

所以是使用随机数还是计数器的问题并不像看起来那么简单。如果使用随机数，锁必须记住很多过去的密码。有代客泊车攻击，具有临时访问权限的人（例如代客泊车服务员）会记录一些访问代码并稍后重播以窃取您的汽车。此外，有人可能会租一辆车，记录足够多的解锁码，然后再回到租车点偷车。提供足够的非易失性内存来记住数千个旧代码可能会增加几美分的锁成本。

如果您选择计数器，问题是同步。钥匙可能用于一把以上的锁；它也可能被意外反复激活（我曾经把一个实验令牌带回家，但它被我的狗咬了）。

因此，您需要一种在计数器递增数百次甚至数千次后恢复的方法。一种常见产品使用 16 位计数器，并在解密的计数器值是最后一个有效代码时允许访问，增量不超过 16。为了应对令牌在其他地方被使用超过 16 次（或被家庭宠物啃咬）的情况，锁将在第二次按下时打开，前提是计数器值自输入有效代码以来已递增 17 到 32,767 次输入（计数器翻转，因此 0 是 65,535 的后继）。这在许多应用程序中都很好，但是如果小偷能够获得六个精心选择的访问代码（例如值 0、1、20,000、20,001、40,000 和 40,001），就可以完全破坏系统。在你的申请中，你会担心吗？

因此，即使设计一个简单的令牌认证机制也并不像看起来那么容易，如果您假设您的产品只会吸引低级对手，那么随着时间的推移，这种假设可能会失败。一个例子是配件控制。许多打印机公司在打印机中嵌入了身份验证机制，以确保使用原装碳粉盒。如果加载的是竞争对手的产品，打印机可能会悄悄地从 1200 dpi 降级到 300 dpi，或者干脆拒绝工作。从科学仪器到游戏机，各种其他行业都在参与进来。用于支持这一点的加密机制从 1990 年代开始就相当简陋，因为供应商认为任何在工业规模上规避它们的竞争对手都可能根据版权法被起诉甚至入狱。但随后一名法官发现，在 Lexmark 诉 SCC 案中，虽然供应商有权聘请他们能找到的最好的密码学家来锁定他们的客户，但竞争对手也有权

2我们将在第 5.3.1.2 节讨论生日定理时更详细地讨论这个问题
在概率论中。

聘请他们能找到的最好的密码分析员,让他们可以自由地从其他地方购买配件。这引发了一场严重的军备竞赛,我们将在后面的章节中不时遇到。在这里我只想说安全并不总是一件好事。安全机制用于支持许多商业模式,在这些模式中,它们通常会阻止设备所有者做她想做的事情,而不是保护她免受坏人的侵害。其效果可能违反公共政策;一个例子是手机锁定,这导致每年有数亿部手机最终进入垃圾填埋场,其中含有有毒重金属以及隐含的碳成本。

4.3.1 挑战与回应

自 1995 年以来,在欧洲销售的所有汽车都必须配备“启用加密功能的防盗器”,到 2010 年,大多数汽车也配备了遥控车门解锁功能,不过大多数汽车还配备了后备金属钥匙,因此您仍然可以进入汽车如果遥控钥匙电池没电了。发动机防盗器更难使用物理方法绕过,并使用两次通过的质询-响应协议来授权发动机启动。当车钥匙插入转向锁时,发动机控制器使用短程无线电向钥匙发送一个由随机 n 位数字组成的挑战。汽车钥匙通过加密质询来计算响应;这通常是由一个单独的 RFID 芯片完成的,该芯片由传入的无线电信号供电,因此即使电池没电也能继续工作。频率很低 (125kHz),因此汽车可以直接为转发器供电,而且交换机也相对不受嘈杂的 RF 环境的影响。

E代表发动机控制器,T代表车钥匙中的应答器,
K 代表转发器和发动机控制器之间共享的加密密钥,N 代表随机挑战,协议可能类似于:

电子！ T:新台
币！ E : T, {T,N}K

这在理论上是合理的,但安全机制的实施通常在人们尝试前两三次时就会失败。

2005 年至 2015 年期间,所有主要的远程钥匙输入和防盗系统都遭到破坏,无论是被安全研究人员、偷车贼或两者兼而有之。这些攻击涉及协议错误、对等密钥管理、弱密码和出口管制法律规定的短密钥的组合。

最先倒下的是TI的DST转发器芯片,至少有两家大型汽车制造商使用,也是SpeedPass通行费支付系统的基础。 Stephen Bono 及其同事在 2005 年发现,它使用了带有 40 位密钥的块密码,这可以通过仅从两个响应 [297] 的蛮力计算得出。这是我在 26.2.7.1 中讨论的美国密码出口管制的一个副作用。从 2010 年开始,福特、丰田和现代采用了后继产品 DST80。 DST80 在 2020 年被 Lennert Wouters 及其同事依次攻破,他们发现除了芯片的侧信道攻击外,密钥管理也存在严重的实现问题:现代密钥只有 24 位熵,而丰田密钥是源自

攻击者可以读取的设备序列号（特斯拉也容易受到攻击,但与老公司不同,它可以通过软件升级解决问题）[2048]。
其次是 Keeloq,它被用于车库门开启器以及一些汽车制造商； 2007 年,Eli Biham 和其他人发现,给定一个小时的令牌访问权限,他们可以收集足够的数据来恢复密钥 [243]。更糟糕的是,在某些类型的汽车中,还有一个协议错误,即密钥多样化使用异或:KT = T KM。所以你可以租一辆你想偷的那种车,然后找出任何其他这种车的钥匙。

同样在 2007 年,有人发布了 Philips Hitag 2 密码,它也有一个 48 位密钥。但是这个密码也很弱,并且由于受到各种密码分析者的攻击,提取密钥所需的时间从几天缩短到几小时再到几分钟。到 2016 年,攻击在笔记本电脑上进行了 8 次身份验证尝试和一分钟的计算;他们与来自所有法国和意大利制造商的汽车以及日产、三菱和雪佛兰 [748] 一起工作。

最后倒下的是大众汽车和其他公司使用的 Megamos Crypto 转发器。汽车锁匠工具从 2008 年开始出现在市场上,其中包括 Megamos 密码,由来自伯明翰和奈梅亨的研究人员进行逆向工程 Roel Verdult、Flavio Garcia 和 Bar ?s Ege 他们破解了它 [1952]。虽然它有一个 96 位密钥,但有效密钥长度仅为 49 位,与 Hitag 2 大致相同。大众汽车在伦敦高等法院获得禁令,禁止他们在 Usenix 2013 上展示他们的成果,声称他们的商业秘密遭到侵犯。研究人员拒绝了,认为锁匠工具供应商已经提取了秘密。经过两年的争论,此案在双方均未承认责任的情况下达成和解。进一步的研究提出了一些进一步的问题。还有一种协议攻击,因为对手可以一个接一个地重写 96 位密钥的每个 16 位字,并一次搜索 16 位密钥;这将攻击所需的时间从几天减少到几分钟 [1953]。

密钥管理普遍存在问题。许多大众汽车的实施并没有在汽车和应答器上分散密钥,而是一次为数百万辆汽车使用固定的全球主密钥。直到 2009 年,它使用一种称为 AUT64 的密码来生成设备密钥;此后,他们转向一种更强大的密码,称为 XTEA,但继续使用全局主密钥,直到 2016 年,大众-奥迪集团的 23 款车型都使用了全局主密钥 [748]3。

很容易发现汽车是否容易损坏:只需尝试购买一把备用钥匙即可。如果锁匠公司找到了复制钥匙的方法,您当地的车库会以几美元的价格向您出售备用钥匙。我们有一把我妻子 2005 年雷克萨斯的备用钥匙,是前车主买的。但是,当我们丢失了我的 2012 款梅赛德斯的一把钥匙时,我们不得不去找一家主要经销商,支付 200 多英镑,出示我的护照和行车记录簿,让机械师拍下底盘上的车辆识别号,然后寄出所有o到梅赛德斯并等待

3 在某些应用中,通用主密钥是不可避免的,例如在与心脏起搏器通信时 心脏病专家可能需要调整走进来的任何患者的起搏器,无论它最初安装在何处,也无论是否安装了心脏起搏器。网络已启动 因此供应商将相同的密钥放入其所有设备中。另一个例子是卫星电视机顶盒中的用户智能卡,我们将在后面讨论。但它们通常会导致一次中断随处运行 (BORA) 攻击。以一种便于盗窃的方式在汽车等贵重资产中安装通用万能钥匙,甚至没有使用适当的防篡改芯片来保护它们,这是一个严重的错误。

一周。我们在第 3 章中看到,设计密码系统的困难部分是在不使恢复机制本身成为漏洞或麻烦的情况下从妥协中恢复。完全一样适用于这里!

但最糟糕的情况还在后头:被动无钥匙进入系统 (PKES)。
Challenge-response 似乎非常好,以至于汽车供应商开始使用它,只需在仪表板上按下按钮即可启动汽车,而不是使用金属钥匙。
然后他们增加了无线电频率以扩大范围,因此一旦驾驶员坐在车内,它不仅可以用于短距离身份验证,还可以用作无钥匙进入机制。营销口号是,只要您将钥匙放在口袋或手提包中,就不必担心;汽车会在您走近时解锁,在您走开时锁定,并在您触摸控件时自动启动。有什么不喜欢的?

好吧,现在您不必按按钮来解锁您的汽车,小偷很容易使用放大或中继信号的设备。小偷用一个继电器潜入您的前门,而将另一个继电器留在您的汽车旁边。如果你把钥匙留在大厅的桌子上,车门就会打开,他就会离开。

即使汽车无法移动,他仍然可以偷走你的 stu 。在汽车盗窃案持续多年下降之后,2017 年的统计数据激增,英国的车辆被盗率增加了 56%,随后在 2018 年进一步增加了 9% [823]4。

外卖信息是,自 1990 年左右以来,尝试使用加密技术使汽车更难被盗的尝试取得了初步成功,因为防盗装置使汽车更难被盗,保险费也下降了。后来事与愿违,因为政治家和营销人员都从中作梗。政客们表示,如果允许人们使用他们无法破解的密码学,那将对执法部门造成灾难性后果,即使是为了阻止汽车盗窃。然后,防盗器供应商的营销人员想要专有算法来锁定汽车公司,而汽车公司自己的营销人员想要被动无钥匙进入,因为它看起来很酷。

我们能做什么?好吧,至少有两家汽车制造商在钥匙扣中安装了加速度计,因此除非钥匙在移动,否则它不会工作。我们的一位朋友带着孩子在室内时将钥匙留在汽车座椅上,然后被锁在门外。

当地警方建议我们使用老式金属方向盘锁;我们的居民协会建议将钥匙放在饼干罐中。至于我,我们买了这样的车,但发现无钥匙进入系统太不稳定了;当它根本无法工作时,我的妻子被困在超市停车场。

所以我们把那辆车拿回来,买了一辆带有合适的按钮遥控锁的二手车。现在有使用 NXP、Atmel 和 TI 的 AES 的芯片 其中 Atmel 是开源的,具有开放的协议栈。

然而,加密本身并不能修复中继攻击;正确的解决方法是基于具有固有测距的超宽带 (UWB) 的新无线电协议,它可以测量从钥匙扣到汽车的距离,精度为 10 厘米,最大范围为 150 米。要做到这一点相当复杂,Srdjan Capkun 及其同事 [1764] 描述了新的 802.15.4z 增强型脉冲无线电的设计;

4 公平地说,这不仅仅是由于中继攻击,因为大约一半的高价值盗窃似乎涉及将汽车盗窃套件连接到手套箱下方的车载诊断端口。
碰巧的是,车内 CAN 总线上使用的身份验证协议在许多方面也容易受到攻击 [891]。由于巨大的行业投资,更新这些协议将需要很多年。

第一款芯片于 2019 年上市，并将于 2020 年开始搭载在汽车中。此类芯片有可能同时取代蓝牙和 NFC 协议，但它们可能并不完全兼容；有一种开放式设计的低速率脉冲 (LRP) 模式，以及部分专有的高速脉冲 (HRP) 变体。

如果我建议一家汽车初创公司，LRP 将是我的起点。

锁并不是质询-响应协议的唯一应用。在 HTTP 摘要式身份验证中，Web 服务器通过向客户端或代理发送随机数来质询与其共享密码的客户端或代理。响应由随机数的哈希值、密码和请求的 URI [715] 组成。这提供了一种不容易受到密码窥探的机制。例如，它用于验证 SIP（IP 语音 (VOIP) 电话协议）中的客户端和服务器。这比以明文形式发送密码要好得多，但就像无钥匙进入一样，它也会受到中间人攻击（受益者是间谍）。

4.3.2 双因素认证

质询-响应最明显的用途可能是双因素身份验证。许多组织向员工发放密码生成器，让他们登录公司计算机系统，许多银行也为客户提供类似的设备。它们可能看起来像小型计算器（有些甚至可以这样工作），但它们的主要功能如下。当你想登录时，你会看到一个随机数，可能是七位数。您将其输入密码生成器，连同可能为四位的 PIN。设备使用与公司安全服务器共享的密钥对这 11 位数字进行加密，并显示结果的前 7 位数字。您输入这七位数字作为您的密码。该协议如图 4.1 所示。如果您有一个带有正确密钥的密码生成器，并且您输入了正确的 PIN，并且输入了正确的结果，那么您就可以进入了。

形式上，S 代表服务器，P 代表密码生成器，PIN 代表用户的个人识别号，U 代表用户，N 代表随机数：

!你：不！ P：N，PIN
P !你：{N,PIN}K你！ S：
{N,PIN}K

这些设备出现于 80 年代初，首先在电话公司流行，然后在 1990 年代流行于银行，供员工使用。有没有键盘的简化版本，只是通过加密计数器或时钟来生成新的访问代码。他们工作；美国国防部在 2007 年宣布，基于国防部通用访问卡的身份验证系统在前一年将网络入侵减少了 46% [320]。

就在这时，骗子开始大规模地对银行客户进行网络钓鱼，因此许多银行都采用了这项技术。我的一家银行给了我一个小型计算器，它为每次登录生成一个新代码，还允许我通过使用他们帐号的最后四位数字代替挑战来验证新收款人的身份。我的另一家银行使用芯片认证程序 (CAP)，一种

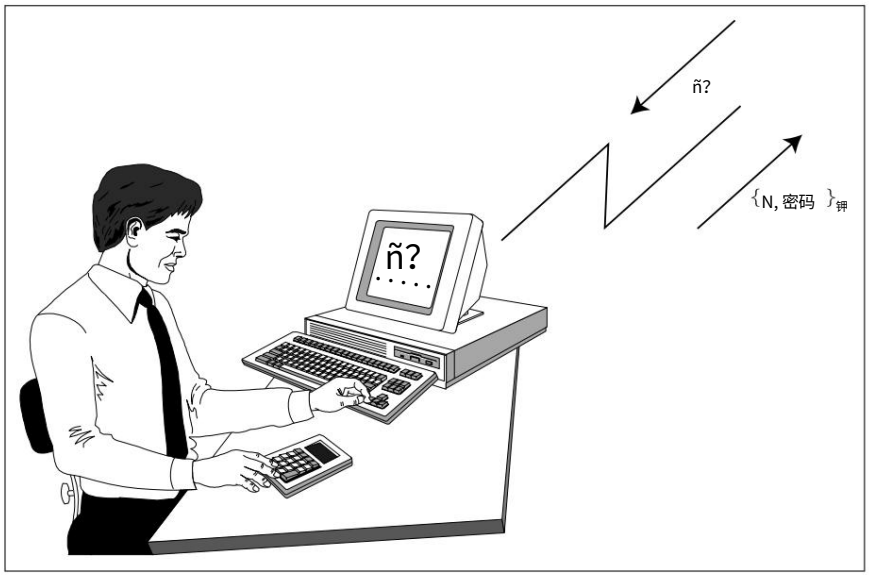

图 4.1：- 密码生成器的使用

计算器,我可以在其中插入我的银行卡来进行加密。

但这仍然不是万无一失的。在本书的第二版中,我提到“有人在持刀威胁下从你手中拿走你的银行卡,现在可以验证你是否告诉了他们正确的 PIN”,现在这种情况发生了。我还注意到,“一旦许多银行使用一次性密码,网络钓鱼者就会重写他们的脚本来进行实时中间人攻击”,这也变得很普遍。要了解此类攻击的工作原理,让我们看一个军事示例。

4.3.3 MIG-in-the-middle 攻击

质询-响应身份验证协议的首次使用可能是在军事领域,使用“敌友识别”(IFF) 系统。 1930 年代和 1940 年代战机速度不断提高,加上喷气发动机、雷达和火箭技术的发明,使得防空部队越来越难以区分自己的飞行器和敌人的飞行器。这导致飞行员误击落同事的风险,并推动了自动系统的开发以防止这种情况发生。这些在第二次世界大战中首次投入使用,使一架被雷达照亮的飞机能够广播一个识别号码,以表明友好的意图。

1952 年,该系统被用于向空中交通管制员识别民用飞机,由于担心一旦广泛使用会失去安全性,美国空军开始了一项将密码保护纳入系统的研究计划。如今,典型的防空系统通过其雷达信号发送随机挑战,友机可以通过正确的方式识别自己

回应。

设计一个好的 IFF 系统是很棘手的。下面的故事说明了其中一个问题,这是我从南非空军 (SAAF) 的一名军官那里听到的。本书第一版出版后,这个故事

是有争议的 正如我将在下面讨论的那样。尽管如此，自第二次世界大战以来，其他电子战系统也玩过类似的游戏。“MIG-in-middle”的故事从此成为民间传说的一部分，它很好地说明了如何在实时挑战响应协议。

80 年代后期，南非军队在纳米比亚北部和安哥拉南部打仗。他们的目标是让纳米比亚处于白人统治之下，并将附庸政府 (UNITA) 强加给安哥拉。由于南非国防军主要由少数白人应征入伍，因此限制伤亡非常重要，因此大多数南非士兵留在纳米比亚执行治安任务，而北部的战斗则由安盟部队完成。 SAAF 的作用是双重的：通过轰炸安哥拉的目标向安盟提供战术支持，并确保安哥拉人和他们的古巴盟友不会在纳米比亚回敬。

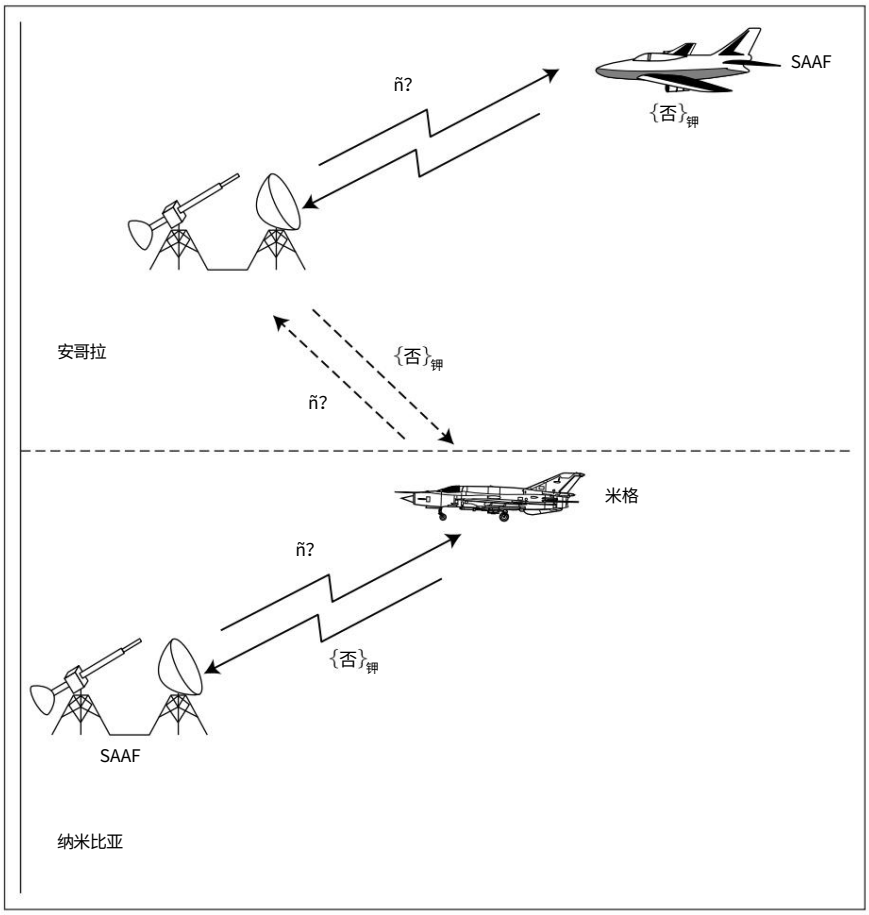

图 4.2：- MIG-in-the middle 攻击

突然，古巴人突破了南非的防空系统，对纳米比亚北部的一个南非营地进行了轰炸，炸死了多名白人应征者。这一证明他们失去制空权的证据帮助比勒陀利亚政府决定将纳米比亚移交给

叛乱分子 这本身就是几年后南非在多数人统治的道路上迈出的一大步。这次突袭也可能是苏联集团军进行的最后一次成功的军事行动。

几年后，一名 SAAF 军官告诉我古巴人是如何把它拉下来的。几架米格战斗机在南非防空带以北的安哥拉南部徘徊，直到一架 SAAF 黑斑羚轰炸机袭击了安哥拉的一个焦油井。然后米格机急转弯，公然飞过南非空军的防空系统，这给敌我识别带来了挑战。 MIG 将它们传送到安哥拉防空炮台，后者将它们传送到 SAAF 轰炸机上；响应被转发回 MIG，MIG 重新传输它们并被允许通过 如图 4.2 所示。据我的线人说，这震惊了比勒陀利亚的一般工作人员 。不仅被黑人对手打败，而且还被智取，这与他们一直以来所坚持的世界观不符。

这个故事在我的书的第一版中发表后，SA 通信安全局的一位前官员联系了我，他对这个故事的细节提出异议。他说，在安哥拉战争期间，他们的敌我识别设备还没有使用密码术，而且总是在敌方领土上空关闭 。

因此，他说，任何电子诡计都必须属于更原始的类型。
然而，其他人告诉我，“中间人”诡计在朝鲜、越南和各种中东冲突中发挥了重要作用。

无论如何，这个故事为我们提供了中间人攻击的另一个例证。对汽车的中继攻击是另一个例子。它也适用于密码计算器：网络钓鱼站点邀请标记登录并同时打开与他的银行的登录会话。银行发送质询；网络钓鱼者将此转发给标记，标记使用他的设备做出响应；网络钓鱼者将响应转发给银行，银行现在接受网络钓鱼者作为标记。

阻止中间人攻击比看起来更难，并且可能涉及多层防御。银行通常会寻找已知的机器、密码、第二个因素（例如来自 CAP 阅读器的身份验证代码）以及交易的风险评估。对于高风险交易，例如向帐户添加新收款人，我的两家银行都要求我计算收款人帐号的验证码。但由于可用性，他们只验证最后四位数字。如果支付需要两分钟并输入几十位数字，那么很多客户会弄错数字，放弃，然后要么打电话给呼叫中心，要么因为生气而去别处转账。此外，坏人可能能够利用任何回退机制，可能是通过欺骗客户拨打在客户和呼叫中心之间运行中间人攻击的电话号码。我将在有关银行业务和簿记的章节中进一步讨论所有这些。

我们将在从 Internet 安全协议到蓝牙的应用程序中一次又一次地遇到此类攻击。它们甚至适用于游戏。正如数学家约翰·康威曾经说过的那样，在邮政国际象棋中很容易与一位大师打成平局：只需同时对付两位大师，一位白棋，一位黑棋，并在他们之间接力走棋！

4.3.4 反射攻击

当两个委托人必须相互识别时，会出现更有趣的问题。假设设计用于防止高射炮手攻击友机的挑战-响应敌我识别系统也必须部署在战斗轰炸机中。现在假设空军只是在每架飞机上安装了一个空中炮手的挑战装置，并将其连接到火控雷达。

但是现在，当一架战斗机挑战一架敌方轰炸机时，轰炸机可能只是将挑战反射回战斗机的僚机，获得正确的响应，然后将其作为自己的响应发回：

F !乙:注意！ F0 : N
F0 ! B:{N}KB !女:{N}
K

有多种方法可以阻止这种情况，例如在交换中包含双方的名字。在上面的例子中，我们可能需要一个友好的轰炸机来回复挑战：

F !乙:乙

响应如下：

乙！ F : {B,N}K

因此，可以检测到来自僚机F0的反射响应 {F0 , N}。

这用于说明作为身份验证基础的信任假设的微妙之处。如果你发出一个挑战 N 并在 20 毫秒内收到一个响应 {N}K，那么 因为光可以在 20 毫秒内传播不到 3,730 英里 你知道在 2000 英里内有人拥有密钥 K . 但这就是你所知道的。如果您可以确定响应不是使用您自己的设备计算的，那么您现在就知道在 2000 英里内还有其他人使用密钥 K。如果您进一步假设密钥 K 的所有副本都安全地保存在可以信任可以正常运行的设备中，并且您看到 {B,N}K，您可能有理由推断呼号为 B 的飞机在2000 英里。仔细分析信任假设及其后果是安全协议设计的核心。

到目前为止，您可能认为我们了解 IFF 的所有协议设计方面。但是我们忽略了一个最重要的问题 也是早期 IFF 系统的设计者没有预料到的问题。由于雷达是无源的，回波很弱，而 IFF 是有源的，因此来自 IFF 发射器的信号通常比同一架飞机的雷达回波在更大的范围内可以听到。盟军以惨痛的方式学到了这一点。 1944 年 1 月，解密了

5不要忘记：您还必须检查入侵者是否只反映了您自己的信息
回击你。您必须能够记住或识别您自己的信息！

谜语信息显示,德国人正在通过询问他们的敌我识别系统,在正常雷达范围的两倍处策划英国和美国的轰炸机。因此,更多的现代系统会验证挑战和响应。例如,北约模式 XII 有一个 32 位加密质询,每个询问信号都会生成不同的有效质询,通常每秒 250 个。从理论上讲,不需要在敌方领土上切换 o ,但实际上,可以记录有效挑战的敌人可以将其作为攻击的一部分进行重放。继电器在模式 XII 中使用方向性和飞行时间变得困难。

其他 IFF 设计问题包括中立者造成的困难、密集操作环境中的错误率、如何处理设备故障、如何管理密钥以及如何应对多国联盟。我将在第 23 章回到 IFF。现在,伪挑战问题用来强调一个重要的观点:安全协议的正确性取决于对需求所做的假设。一种协议可以防止一种攻击(被你自己的一方击落),但增加了对更有可能发生的攻击(被另一方击落)的暴露可能无济于事。事实上,虚假挑战问题在第二次世界大战中变得如此严重,以至于一些专家主张完全放弃敌我识别,而不是冒着数百人编队中的一名轰炸机飞行员无视命令并在敌方领土上空开启敌我识别的风险.

4.4 处理消息

我们现在已经看到许多反映或欺骗用于验证参与者身份的信息的中间人攻击。但是,还有更复杂的攻击,攻击者不仅会冒充某人,还会操纵消息内容。

我们已经看到的一个例子是预付费计价器,它只记住它看到的最后一张票,所以它可以通过依次复制两张票 A 和 B 的代码来无限次充值:ABABAB ....另一个是不诚实的出租车司机在将计程器连接到出租车变速箱中的传感器的电缆中插入脉冲发生器。当传动轴转动时,传感器会发送脉冲,从而让计价器计算出出租车走了多远。海盗设备可以插入额外的脉冲,使出租车看起来走得更远。想要比法规允许的开得更快或更远的卡车司机可以使用类似的设备来丢弃一些脉冲,因此他似乎一直在开得更慢或根本没有开。我们将在 14.3 节的“监控系统”一章中讨论此类攻击。

与监控系统一样,控制系统通常需要针对消息操纵攻击进行强化。用于国际电话和数据跟踪的 Intelsat 卫星具有防止命令被接受两次的机制 否则攻击者可以重播控制跟踪并重复命令执行相同的操作,直到卫星耗尽燃料 [1526]。在后面的具体应用章节中,我们会看到很多涉及消息操纵的协议攻击的例子。

4.5 改变环境

协议失败的一个常见原因是环境发生变化,使得设计假设不再成立,安全协议无法应对新的威胁。

一个很好的例子来自提款机欺诈的世界。 1993 年,荷兰遭受了“幻影提款”的流行;媒体上有很多争议,银行声称他们的系统是安全的,而许多人写信给报纸声称被骗了。最终银行注意到,许多受害者曾在乌得勒支附近的某个加油站使用过他们的银行卡。这被放了下来,其中一名工作人员被逮捕了。原来他窃听了从读卡器到控制读卡器的 PC 的线路;他的水龙头记录了他们卡片上的磁条细节,同时他用眼球捕捉了他们的 PIN [54]。在 2000 年代中期转向“芯片和 PIN”智能卡之后,英国也发生了完全相同的欺诈行为;一个团伙窃听了大约 200 个加油站,从电线中收集卡数据,使用闭路电视摄像机观察 PIN,然后制作了数千张磁条克隆卡,这些卡在 ATM 仍然使用磁条技术的国家使用。在我们当地的加油站,200多位顾客突然发现自己的卡在泰国的ATM机上被刷过。

为什么系统设计得如此糟糕,为什么设计错误会通过重大技术变革持续存在十多年?好吧,当 IBM 和 VISA 等组织在 1980 年代早期制定管理磁条卡和 PIN 的标准时,工程师们做出了两个假设。首先是磁条的内容　卡号、版本号和有效期　不是秘密,而 PIN 是 [1301]。 （使用的类比是磁条是你的名字,PIN 是你的密码。）第二个假设是银行卡设备只能在可信赖的环境中操作,例如在物理坚固的自动柜员机中,或由银行职员操作在一个出纳站。因此，“显然”只需要在 PIN 从 PIN 键盘发送到服务器的过程中对 PIN 进行加密;磁条数据可以从读卡器以明文形式发送。

到 1993 年,这两种假设都发生了变化。20 世纪 80 年代后期,伪造卡的流行（主要发生在远东）促使银行在磁条上引入验证码。此外,银行卡行业的商业成功导致许多国家的银行将借记卡的使用从 ATM 扩展到各种商店的终端。这两种环境变化的结合破坏了原始系统架构背后的假设。人们不是将磁条不包含安全数据的卡放入受信任的机器中,而是将具有明确安全数据的卡放入不受信任的机器中。这些变化发生得如此缓慢,而且持续了如此长的时间,以至于业界没有看到问题的出现。

4.6 选择的协议攻击

热衷于推广身份证的政府已尝试将其用于许多其他交易;有些人想要一张卡用于身份证、银行业务甚至

交通票务。新加坡甚至试验了一张兼作军人身份证的银行卡。这引入了一些有趣的新风险:如果一名海军上尉在享用完丰盛的晚餐后试图从 ATM 机上提取现金,但忘记了他的 PIN,他是否会在星期一早上银行开门并给他密码时才能将船开出海?卡背?

一些公司正在推出可用于各种交易的多功能身份验证设备,让您不必随身携带数十张不同的卡和钥匙。对未来更现实的看法可能是人们的电话将用于大多数私营部门的身份验证功能。

但这也可能不像看起来那么简单。 “选择的协议攻击”背后的想法是,给定一个目标协议,你可以设计一个新的协议来攻击它,如果用户可以被诱骗重复使用相同的令牌或加密密钥。那么黑手党如何设计一个协议来攻击银行交易的认证呢?

这是一种方法。访问色情网站的人通常会被要求提供“年龄证明” ,这通常涉及向网站本身或年龄检查服务提供信用卡号码。如果智能手机被用来验证一切,色情网站自然会要求客户验证随机挑战作为年龄证明。如图 4.3 所示,色情网站可能会发起“中间黑手党”攻击。他们等到毫无戒心的客户访问他们的站点,然后从经销商处订购可转售的东西(例如金币) ,扮演硬币经销商客户的角色。

当硬币经销商向他们发送交易数据以进行身份验证时,他们会通过色情网站将其转发给等待的顾客。可怜的人同意了,黑手党得到了金币,当成千上万的人突然抱怨他们的卡在月底被巨额收费时,色情网站已经消失了连同黄金 [1032]。

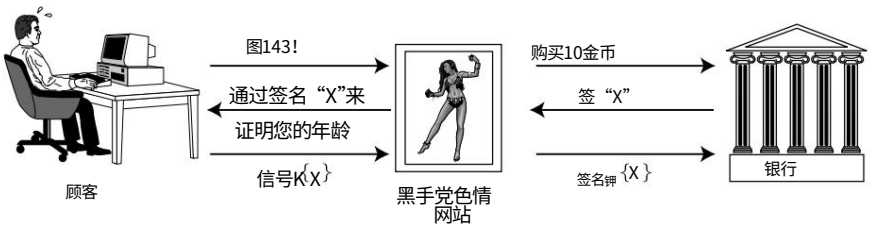

图 4.3：- 中间黑手党攻击

在 1990 年代,这种漏洞进入了国际标准:数字签名和身份验证标准可以以这种方式背靠背运行。事实证明,许多协议虽然本身是安全的,但如果可以诱使用户在其他应用程序中重复使用相同的密钥,则可能会被破坏 [1032]。这就是为什么,如果我们要使用我们的手机来验证所有内容,那么将银行应用程序和色情应用程序分开是非常重要的。这将是我们下一章“访问控制”的主题。

一般来说,在多个应用程序中使用加密密钥(或其他身份验证机制)是危险的,同时让其他人引导他们的

自己的应用程序安全性 你的可能是彻头彻尾的愚蠢。典型案例是银行依靠双因素身份验证将短信作为身份验证代码发送给客户。正如我在第 3.4.1 节中讨论的那样,坏人已经学会了通过 SIM 交换欺诈来攻击该系统 向电话公司假装他们是目标,声称丢失了手机,并获得了一张替换 SIM 卡。

4.7 管理加密密钥

到目前为止,我们讨论的安全协议示例主要是关于验证委托人姓名或应用程序数据(例如驾驶出租车计价器的脉冲)。还有一类非常重要的身份验证协议 用于管理加密密钥的协议。

4.7.1 复活的小鸭子

在物联网中,有时可以通过本地设置和首次使用信任或 TOFU 策略直接物理地管理密钥。

车辆提供了一个早期的例子。我在上面提到过,狡猾的出租车司机过去常常在从汽车变速箱传感器到出租车计价器的电缆中放置中断器,以增加额外的里程数。同样的问题反过来也发生在行驶记录仪上,卡车使用这种设备来监控司机的工作时间和速度。当行驶记录仪在 1990 年代后期实现数字化时,我们决定对来自传感器的脉冲序列进行加密。但是如何管理密钥呢?解决方案是,每当新的行驶记录仪在恢复出厂设置后启动时,它都会信任通过传感器电缆接收到的第一个加密密钥。我将在 14.3 节中进一步讨论这个问题。

第二个例子是 Homeplug AV,该标准用于加密国内电力线上的数据通信,并广泛用于 LAN 扩展器。在默认的“just-works”模式下,新的 Homeplug 设备信任它看到的第一个密钥;如果您的新 wi i 扩展器与邻居的 wi i 配对,您只需按下重置按钮并重试。还有一种“安全模式”,您可以在其中打开网络管理节点的浏览器并手动输入打印在设备包装上的加密密钥,但是当我们设计 Homeplug 协议时,我们意识到大多数人没有理由为此烦恼.

TOFU 方法也被称为“复活的小鸭”,经过 Frank Stajano 和我在行驶记录仪工作的背景下所做的分析。
这个想法是,当一只小鸭子孵化出来时,它会在它看到的第一个会动和嘎嘎叫的东西上留下印记,即使这是农夫 最终可能会被一只认为自己是木乃伊的鸭子到处跟着。如果这种错误的印记发生在电子设备上,你需要一种方法来杀死它并将它复活到一个新的状态 重置按钮就是这样做的 [1819]。

4.7.2 远程密钥管理

更常见和有趣的案例是远程设备中的密钥管理。基本技术是从 20 世纪 70 年代后期开发的,用于管理分布式计算机系统中的密钥,提款机是早期的应用。在本节中,我们将讨论诸如 Kerberos 之类的共享密钥协议,而在第 5 章中讨论了公钥密码学之后再讨论诸如 TLS 和 SSH 之类的公钥协议。

密钥分发协议背后的基本思想是,在两个委托人想要通信的地方,他们可以使用受信任的第三方来介绍他们。
习惯上给他们取人名,以免在过多的代数中迷失方向。因此,我们将两个通信委托人称为“Alice”和“Bob”,并将受信任的第三方称为“Sam”。 Alice、Bob 和 Sam 很可能是运行在不同设备上的程序。 （例如,在让汽车经销商将替换钥匙与汽车配对的协议中,Alice 可能是汽车,Bob 是钥匙,Sam 是汽车制造商。）

一个简单的身份验证协议可以按如下方式运行。

1. 爱丽丝先打电话给山姆,向他索要与鲍勃通信的密钥。
2. 山姆向爱丽丝发送一对证书作为回应。每个都包含一个密钥的副本,第一个加密因此只有 Alice 可以读取,第二个加密因此只有 Bob 可以读取。
3. 爱丽丝随后打电话给鲍勃,并出示第二张证书作为她的介绍。
   他们每个人都使用他们与 Sam 共享的密钥解密相应的证书,从而获得对新密钥的访问权。 Alice 现在可以使用密钥向 Bob 发送加密消息,并从他那里接收消息作为回报。

我们已经看到重放攻击是一个已知问题,因此为了 Bob 和 Alice 都可以检查证书是否最新,Sam 可能会在每个证书中包含一个时间戳。如果证书永不过期,则处理权限已被撤销的用户可能会出现严重问题。

使用我们的协议符号,我们可以将其描述为

一个！ S:A,废
话！ A : {A, B, KAB, T}KAS , {A, B, KAB, T}KBS A ! B :
{A, B, KAB, T}KBS , {M}KAB

扩展符号,爱丽丝打电话给山姆,说她想和鲍勃谈谈。
Sam 编写了一条消息,其中包含 Alice 的姓名、Bob 的姓名、供他们使用的会话密钥和时间戳。他用他与 Alice 共享的密钥加密所有这些,并用他与 Bob 共享的密钥加密它的另一个副本。他将两个密文都给了爱丽丝。
Alice 从加密给她的密文中检索会话密钥,并将为他加密的密文传递给 Bob。她现在向他发送她想发送的任何消息,并使用此会话密钥加密。

4.7.3 Needham-Schroeder 协议

很多事情都可能出错,这是一个著名的历史例子。许多现有的密钥分发协议都源自于 1978 年出现的 Needham-Schroeder 协议 [1426]。它与上面的有点相似,但使用随机数而不是时间戳。它运行如下:

消息 1 A ! S : A, B, NA消息 2 S ! A :
{NA,B,KAB, {KAB, A}KBS }KAS Message 3 A ! B : {KAB, A}KBS Message 4
B ! A : {NB}KAB Message 5 A ! B : {NB 1}KAB

在这里,爱丽丝主动告诉山姆:“我是爱丽丝,我想和鲍勃谈谈,我的随机数是NA。” Sam 向她提供了一个会话密钥,该密钥使用她与他共享的密钥进行了加密。该密文还包含她的随机数,因此她可以确认这不是重播。他还给了她一张证书,以将此密钥传达给 Bob。她将其传递给 Bob,Bob 然后进行挑战-响应以检查她是否在场并保持警觉。

这个协议有一个微妙的问题 Bob 必须假设他从 Sam(通过 Alice)那里收到的密钥KAB是最新的。这不一定是这样:爱丽丝可以在步骤 2 和步骤 3 之间等一年。在许多应用程序中,这可能并不重要;它甚至可以帮助 Alice 缓存密钥以防止可能的服务器故障。但是,如果对手 比如查理 拿到了爱丽丝的密钥,他就可以用它来设置与许多其他委托人的会话密钥。如果 Alice 被解雇了,Sam 最好有一份公司中每个人的名单,他向他们发放了与她通信的密钥,告诉他们不要再相信它了。换句话说,撤销是一个问题:Sam 可能必须保留他所做的一切的完整日志,并且这些日志的大小将永远增长,除非委托人的名字在未来的某个固定时间过期。

将近 40 年后,这个例子仍然存在争议。简单化的观点是李约瑟和施罗德只是弄错了。 Susan Pancho 和 Dieter Gollmann(对此我有些同情)认为,这是由不断变化的假设引起的协议失败 [780, 1491]。 1978 年是一个更友善、更温和的世界;当时的计算机安全关注的是将“坏人”拒之门外,而如今我们希望“敌人”成为我们系统的用户。 Needham-Schroeder 的论文假设所有委托人都拥有自己,并且所有攻击都来自外部 [1426]。在这些假设下,该协议仍然有效。

4.7.4 Kerberos

Needham-Schroeder 协议最重要的实用衍生产品是 Kerberos,这是一种分布式访问控制系统,起源于麻省理工学院,现在是标准网络身份验证工具之一 [1826]。它已成为 Windows 和 Linux 的基本身份验证机制的一部分,尤其是当机器通过局域网共享资源时。 Kerberos 不是单一的可信第三方,而是有两种:

用户登录,票证授予服务器为他们提供允许访问各种资源（如文件)的票证。这使得可扩展的访问管理成为可能。例如,在一所大学中,可能通过学院或宿舍管理学生,但按部门管理文件服务器;在公司中,人事人员可能会将用户注册到工资单系统,而部门管理员则管理服务器和打印机等资源。

首先,爱丽丝使用密码登录到认证服务器。她 PC 中的客户端软件从该服务器获取一张票,该票用她的密码加密并且包含会话密钥KAS。假设她获得了正确的密码,她现在控制了KAS并获得对由票据授予服务器 S 控制的资源 B 的访问权,将发生以下协议。它的结果是一个带有时间戳TS和生命周期 L的密钥KAB ,它将用于验证 Alice 使用该资源的后续 trac：

一个！ S:A,废
话！ A : {TS, L, KAB,B, {TS, L, KAB, A}KBS }KAS A ! B :
{TS, L, KAB, A}KBS , {A, TA}KAB B !答:{TA + 1}KAB

将其翻译成英文:Alice 向票证授予服务器请求访问 B。如果允许,则创建包含合适密钥KAB的票证{TS, L, KAB, A}KBS并提供给 Alice 使用。她还获得了一份她可读形式的密钥副本,即在KAS 下加密。她现在通过向资源发送时间戳TA来验证票证,资源通过发回递增 1 的时间戳来确认票证还活着（这表明它能够正确解密票证并提取密钥KAB）。

Needham-Schroeder 协议的撤销问题已通过引入时间戳而不是随机随机数得到修复。但是,就像在生活的大部分时间里一样,我们很少能免费获得安全感。现在有一个新的漏洞,即我们各个客户端和服务器上的时钟可能会不同步;作为更复杂攻击的一部分,它们甚至可能被故意去同步化。

更重要的是,Kerberos 是一个受信任的第三方 (TTP) 协议,因为 S 是受信任的:如果警察带着搜查令出现,他们可以让 Sam 交出密钥并读取跟踪记录。具有此功能的协议在 1990 年代的“加密战争”期间受到青睐,我将在第 26.2.7 节中讨论。不涉及或较少信任第三方的协议通常使用公钥密码术,我将在下一章中对此进行介绍。

与 Kerberos 非常相似的协议是 OAuth,这是一种允许安全委托的机制。例如,如果您使用 Google 登录 Doodle 并允许 Doo dle 更新您的 Google 日历,Doodle 的网站会将您重定向到 Google,让您登录（或依赖于先前登录的主 cookie)并征求您的同意让涂鸦写入您的日历。 Doodle 然后为您提供日历服务的访问令牌 [863]。我在第 3.4.9.3 节中提到,这会带来跨站点网络钓鱼风险。 OAuth 不是为用户身份验证而设计的,访问令牌也没有与客户端紧密绑定。它是一个复杂的框架,可以在其中构建委托机制,包括短期和长期访问令牌;详细信息与 cookie 和 Web 重定向如何操作和优化以使服务器处于状态有关

更少,因此它们可以很好地扩展现代网络服务。在上面的示例中,您希望能够撤销 Doodle 在 Google 的访问权限,因此在幕后 Doodle 仅获得短期访问令牌。由于这种复杂性,OpenID Connect 协议是 OAuth 的“配置文件”,它为唯一需要的服务是身份验证的情况绑定了详细信息。 OpenID Connect 是您使用 Google 或 Facebook 帐户登录报纸时使用的。

4.7.5 实用密钥管理

因此,我们可以使用像 Kerberos 这样的协议来设置和管理用户之间的工作密钥,因为每个用户都与充当密钥分发中心的服务器共享一个或多个长期密钥。但也可能有数万名员工的加密密码和大量设备的密钥。这是很多关键材料。如何管理?

密钥管理是一项复杂而艰巨的工作,并且经常出错,因为它是事后才想到的。您需要坐下来思考需要多少密钥、如何生成它们、它们需要保留多长时间以及它们最终将如何销毁。关注的问题要多得多 其中许多都在联邦信息处理标准中针对密钥管理 [1408] 进行了阐述。随着应用程序的发展,事情会出错;提供空间以支持明年的功能很重要。

支持从安全故障中恢复也很重要。然而,这两者都没有标准的方法。

我将在第 5 章中讨论的公钥密码术可以稍微简化密钥管理任务。在银行业,通常的答案是使用称为硬件安全模块的专用加密处理器,稍后我将对此进行详细描述。但是,这两者都带来了更多的复杂性,甚至更微妙的出错方式。

4.8 设计保证

我们在上面看到的那种微妙的困难,以及保护属性依赖于可能被误解的微妙假设的许多方式,促使研究人员将形式化方法应用于协议。这个练习的目的最初是决定一个协议是对还是错:它要么被证明是正确的,要么被展示为攻击。我们经常发现这个过程有助于澄清作为给定协议基础的假设。

有几种不同的方法来验证协议的正确性。最著名的逻辑之一是信念逻辑,或 BAN 逻辑,以其发明者 Burrows、Abadi 和 Needham [357] 的名字命名。它推断委托人在看到某些消息、时间戳等后可能会合理地相信什么。

其他研究人员应用了 CSP 等主流形式化方法和 Isabelle 等验证工具。

在使用正式方法证明正确的协议中发现缺陷的历史存在一些历史记录;我在第二版第 3 章中描述了一个示例,该示例是如何使用 BAN 逻辑来验证用于存储的银行卡的。

价值支付。它在德国仍被用作“Geldkarte”,但在其他地方它的使用已经消失（它是南非的 Net1、比利时的 Proton、法国的 Moneo 和一种名为 COPAC 的 VISA 产品）。因此,我决定从这个版本中删除血淋淋的细节;第二版在线免费,因此您可以下载并阅读详细信息。

形式化方法是在安全协议设计中查找错误的极好方法,因为它们迫使设计人员将所有内容都明确化,从而面对困难的设计选择,否则这些选择可能会被捏造。但它们也有其局限性。

我们经常在经过验证的协议中发现错误;他们只是不在我们验证的部分。例如,拉里·保尔森 (Larry Paulson) 在 1998 年使用他的伊莎贝尔定理证明器验证了 SSL/TLS 协议,从那以后,每年都会发现大约一个安全漏洞。这些并不是基本设计中的缺陷,而是利用了后来添加的附加功能,以及我们稍后将讨论的时序攻击等实施问题。在这种情况下,形式化方法没有失败;这只是告诉攻击者他们不需要费心寻找的地方。

由于这些原因,人们已经探索了确保身份验证协议设计的替代方法,包括协议健壮性的想法。正如结构化编程技术旨在确保软件的设计有条不紊并且不会遗漏任何重要内容一样,健壮的协议设计在很大程度上与明确性有关。稳健性原则包括协议的解释应仅取决于其内容,而不是其上下文;所以所有重要的事情（比如校长的名字）都应该在消息中明确说明。不可能以不止一种方式解释数据;所以消息格式需要明确什么是名字,什么是地址,什么是时间戳等等;字符串格式必须是明确的,并且应该不可能使用协议本身对处理它的软件发起攻击,例如通过缓冲区溢出。还有其他问题涉及计数器、时间戳和随机挑战提供的新鲜度,以及使用加密的方式。如果协议使用公钥加密或数字签名机制,则会出现更微妙的攻击和更进一步的稳健性问题,我们将在下一章开始解决这些问题。为了激发你的胃口,协议中的随机性通常有助于其他层的稳健性,因为它使得进行整个范围的攻击变得更加困难　从基于数学密码分析的攻击到利用功耗和时序等边信道的攻击,再到物理攻击。涉及微探针或激光的攻击。

4.9 总结

密码只是更普遍的概念（安全协议）的一个示例。
协议指定委托人用于在系统中建立信任关系的步骤,例如验证身份声明、证明凭证的所有权或建立对资源的声明。密码身份验证协议用于广泛的目的,从基本的实体身份验证到为分布式系统提供基础设施,允许将信任从存在的地方带到需要的地方。安全协议适用于所有领域

从远程车门锁到军事 IFF 系统到分布式计算机系统中的身份验证的各种系统。

协议出奇地难以正确。他们可能会遇到许多问题,包括中间人攻击、修改攻击、反射攻击和重放攻击。这些威胁可能与实施漏洞和糟糕的密码学相互作用。使用数学技术来验证协议的正确性会有所帮助,但它不会捕获所有错误。一些最有害的故障是由设计协议所针对的环境的逐渐变化引起的,因此它提供的保护不再相关。结果是,攻击仍然频繁发生在我们已经使用多年的协议上,有时甚至发生在我们认为我们有安全证明的协议上。失败会产生真正的后果,包括自汽车制造商开始采用被动无钥匙进入系统而没有停下来考虑中继攻击以来,全球汽车犯罪率上升。请不要设计自己的协议;获得专家的帮助,并确保您的设计被发表以供研究团体进行全面的同行评审。即使是专家也会弄错协议的第一个版本(我不止一次)。在协议实际部署之前修复错误要便宜得多,无论是在现金方面还是在声誉方面。

研究问题

在过去 30 年中,有几次有人认为方案已经“完成”,我们应该转向新的研究课题。新应用程序的出现一再证明他们是错误的,新的错误和攻击有待探索。正式方法在 20 世纪 90 年代初期蓬勃发展,然后是密钥管理协议;在 1990 年代中期,关于电子商务机制的提案如潮水般涌来,让我们忙得不可开交。自 2000 年以来,随着越来越多地使用安全机制来支持业务模型,协议研究的一部分已经具有经济意义;设计师的“敌人”通常是商业竞争者,甚至是客户。另一个应用了协议分析工具来查看应用程序编程接口 (API) 的安全性,我将在稍后返回该主题。

许多协议研究都是问题驱动的,但仍然存在深层次的问题。
例如,我们可以从形式化方法中得到多少?我们如何管理健壮的协议通常是那些完全指定和检查所有内容的原则与良好的规范不应过度约束实施者的系统工程原则之间的紧张关系?

进一步阅读

关于安全协议的研究论文相当广泛地散布在整个文献中。对于历史背景,您可以阅读原始的 Needham Schroeder 论文 [1426]、Burrows-Abadi-Needham 身份验证逻辑 [357]、关于协议稳健性的论文 [2, 112] 以及 Anderson 和

李约瑟[113]。除此之外,还有许多论文散布在各种会议中;您也可以从研究特定应用领域（例如支付)中使用的协议开始,我们将在第 2 部分中对此进行更详细的介绍。至于远程钥匙输入和汽车周围的其他安全问题,一个很好的起点是一份技术报告查理·米勒 (Charlie Miller) 和克里斯·瓦拉塞克 (Chris Valasek) 关于如何破解吉普切诺基 [1316]。