第20话

高级密码学工程

给我一块立足之地,我将推动世界。

– 阿基米德

认为他的问题可以用密码学解决的人,不理解他的问题也不理解密码学

– 由 Roger Needham 和 Butler Lampson 相互归因

20.1 简介

密码学通常用于构建更复杂的设计可以依赖的可信组件。这样的设计来自三个相当不同的背景。第一个是我们在第 9 章中描述的政府系统世界,其理念是使用数据二极管和多级安全加密设备等机制最小化可信计算基础。第二个是第 12 章中描述的银行业世界,其中智能卡用作身份验证令牌,而 HSM 用于保护 PIN 和密钥。第三个是 1980 年代和 90 年代的密码学研究世界,人们梦想用数学解决社会问题:创建匿名通信,使受压迫群体可以逃避国家监视,导致抗审查出版、无法追踪的数字现金和电子选举那是不可能操纵的。在所有这些情况下,事实证明现实生活比我们预期的要混乱一些。

我们使用更复杂的加密组件作为平台。但工程不仅仅是减少攻击面,或简化我们的故障树分析。在大多数情况下,政策、责任和其他复杂因素之间存在着重要的相互作用。

在本章中,我将讨论密码工程的六个示例

– 全盘加密、Signal 协议、Tor、硬件安全模块、飞地和区块链。第一个是设置场景的简单示例;其他五个以更复杂的方式使用加密来支持广泛的应用程序,包括最后三个的支付。除了 HSM 之外,其他所有东西都被网络罪犯使用。

硬盘加密自 1980 年代就已问世,是最简单的安全产品之一,至少在概念上如此。通过在机器使用时加密硬盘上的数据,可以确保窃贼只能窃取硬件,而不能窃取数据。

Signal 是一种用于手机之间安全消息传递的协议。它可能是复杂性的下一个层次,是关于使人们能够在设备受到威胁时尽可能安全地管理社交网络。 Signal 通过 enclave 进行私人联系人发现。

Tor 通过提供匿名性将这一点提升到一个新的水平,当您不希望有人通过观察您的 trac 知道您正在与谁交谈或您正在访问哪些网站时。

自 1980 年代以来,HSM 一直为支付服务提供信任平台。
但是在它们上运行的加密应用程序可能会受到对其应用程序编程接口的攻击,这些接口与支付应用程序紧密相关,很难修复。

飞地是 CPU 供应商提供通用加密平台的尝试:我们自 2004 年以来就拥有 Arm 的 TrustZone,自 2015 年以来就拥有英特尔的 SGX。
他们开始在支付应用程序中取代 HSM,并且还支持 Signal 中的私人联系人发现。但他们一直受到从侧信道攻击到类中断的问题的困扰。例如,如果可以从 SGX 芯片中提取主密钥,就可以破坏整个生态系统。

最后,对于一种完全不同的可信计算机,我们看看比特币。
这是一个自 2009 年以来的项目,旨在创建一种基于共享分类账的数字货币,该分类账是使用加密机制从相互不信任的各方合作中出现的。许多利益相关者远非值得信赖,并且在技术堆栈的多个级别上都有主导者。然而,由于密码学和经济激励的结合,一台受信任的计算机以某种方式出现了,并且尽管成功攻击可能会损失巨额资金,但它仍在继续运行。

将银行家和歹徒的可信平台集中在一章中可能会有用,这样我们就可以对它们进行对比。出现了一些惊人的事实。例如,顶级科技公司生产可信计算机的最佳尝试产生了有缺陷的产品,而歹徒似乎创造了一些有用的东西 至少目前是这样。

20.2 全盘加密

全盘加密 (FDE) 背后的想法很简单。您在将数据写入磁盘时加密数据,并在再次读取数据时进行解密。密钥取决于初始身份验证步骤,例如密码,当

机器休眠或关闭 。因此,如果医生将笔记本电脑遗忘在火车上,丢失的只是硬件;医疗记录不是。 FDE 已成为许多行业的监管要求。在欧洲,隐私监管机构普遍认为丢失具有 FDE 的机器的严重程度不足以引起罚款或需要强制通知数据主体。许多手机和笔记本电脑都带有 FDE;有些默认情况下启用 (Android),而有些则只需单击一下 (Mac)。

但是,在表面下刮一点,质量差异很大。从 20 世纪 80 年代早期的硬盘开始,软件 FDE 产品就可以使用,但会带来性能损失,而硬件产品成本更高且受出口管制。工程并不简单,因为您需要一个平台来运行初始身份验证步骤。早期产品提供额外的加密卷,但不保护主机操作系统,可能会被恶意软件攻破。初始身份验证在其他方面很棘手。如果您从用户密码中导出磁盘密钥,那么小偷就可以在线尝试无数次,正如我们在 3.4.4.1 中讨论的那样,并猜测普通用户设置的任何内容。硬件 TPM 芯片可以限制密码猜测,从 2007 年开始,它可以用于带有 Bitlocker 的 Windows。将 FDE 集成到一个平台中,使供应商能够设计连贯的机制,以可信地启动操作系统的真实副本,设置和管理恢复密钥,并处理与软件升级、交换空间、设备维修、备份和备份的相当复杂的交互。恢复用户数据,并在设备售出时恢复出厂设置。

第三方产品 erings 开始提供一些额外的功能:例如,TrueCrypt 提供了一个隐写文件系统,在该系统中,除非用户知道正确的密码,否则磁盘卷的存在将保持隐藏 [114]1。

出售给犯罪分子的加密电话 EncroChat 有一个完整的隐藏分区,其中包含加密聊天和 VOIP 应用程序;我将在第 25.4.1 节中更详细地讨论此类产品。然而,大多数人现在使用他们的电话或笔记本电脑供应商提供的 FDE 工具,因为正确的集成涉及相当多的平台。自 2010 年以来,我们拥有专为 FDE 设计的特殊操作模式 XTS-AES;它用扇区号对每个加盐的块进行加密,并具有使磁盘块适合块密码的机制。当在支持 AES 的 CPU 上运行时,Microsoft 的 BitLocker 和 Apple 的 FileVault 等产品的开销仅为百分之几。

然而攻击仍在继续。 2008 年,普林斯顿大学的 Alex Halderman 及其同事提出了冷启动攻击,它击败了当时市场上的主要 FDE 产品,并且仍然会给许多机器带来问题 [854]。正如我在第 18.3 节中所述,您冻结了存储临时加密密钥的计算机 DRAM,然后使用轻量级操作系统重新启动设备并获取内存映像,从中可以读取密钥。 2015 年,我们发现大多数 Android 都不安全:大多数 OEM 的出厂重置功能设计得很糟糕,包括 FDE 密钥在内的凭据可以从二手设备中恢复 [1757]。而且大多数安卓手机都没有

1该产品突然停产,其匿名开发者因不明漏洞建议用户迁移至其他产品;有些人怀疑这是一个“保证金丝雀”,一种预先计划好的警告消息,开发人员通过定期证明他们不受胁迫来抑制其传输,但是一旦他们收到传票或保证书就会发出警告[61]。

一旦它们不再销售就打补丁。而在 2019 年,Carlo Meijer 和 Bernard van Gastel 发现占据 60% 市场份额的三个第三方 FDE 产品不安全,开源软件加密会更好,如果其中一个 Bitlocker 自行关闭这些硬件产品似乎存在;由于他们的工作,它不再这样做 [1285]。然后是附带损害。现在许多敏感数据不保存在硬盘上,而是保存在 Amazon S3 存储桶中,审计人员通常要求对这些存储桶进行加密;但由于 S3 存储桶的故障模式不是亚马逊数据中心的窃贼,而是访问控制的疏忽,因此不清楚 S3 存储桶加密是否实现了除勾选框合规性之外的任何其他功能。

最后,必须考虑滥用性,其中至少有两种重要类型。首先,FDE 代码的广泛可用性是导致最近勒索软件攻击浪潮的两个因素之一,一个团伙侵入你的系统,安装 FDE,让它运行,直到你加密了足够多的备份,使恢复变得痛苦,然后索要钥匙的赎金。 （另一个组成部分是加密货币,我将在本章后面讨论。）其次,许多人认为 FDE 是防止妥协的神奇保险,如果笔记本电脑启用了 FDE（或应该）,即使发现者可能已经看到密码,或者能够轻易猜到它。

因此,即使是最简单的加密产品也与合规性有很大的纠缠,其背后的内容比您乍看之下所想的要复杂得多,通常会造成一些性能损失,并且容易受到有能力的对手的攻击 即使在相关攻击发生多年之后已经发表。

20.3 信号

随着智能手机在世界范围内普及,人们从 SMS 转向 WhatsApp、Telegram 和 Signal 等消息应用程序,因为它们更便宜、更灵活,可以让您创建家人和朋友群组。很快他们也开始支持语音和视频通话,并提供端到端加密。

以前可以使用 PGP 等程序加密电子邮件,但它相当繁琐（正如我们在 3.2.1 节中讨论的那样)并且仍然是一种小众活动。新平台的到来意味着消息加密可以普及,作为应用程序的默认设置;斯诺登的泄密有助于激发公众的需求。

Signal 是一款免费的消息传递应用程序,最初由一个名叫 Moxie Marlinspike 的人开发。它为消息传递的端到端加密设定了标准,其机制已被包括 WhatsApp 在内的竞争产品所采用。手机信息可能非常敏感,从情人的约会到商业交易,再到外交峰会上的政治阴谋,应有尽有;然而,手机经常丢失或被盗,或因屏幕损坏而送修。因此,手机中的密钥材料经常会遭到泄露,而在应用程序中仅拥有一个长期存在的私钥是不够的。因此,Signal 协议提供了前向保密的特性,即今天的关键妥协不会暴露任何未来的流量,以及后向保密的特性,这意味着它也不会暴露以前的流量。这些是现在

形式化为妥协后的安全性 [451]。

该协议包含三个主要组件:扩展三重死海曼 (X3DH) 协议,用于在 Alice、Bob 和服务器之间设置密钥;一个棘轮协议,一旦建立了一个秘密密钥就派生出消息密钥;以及在您的地址簿中查找其他人的 Signal 密钥的机制。

我们不能使用普通的 Die-Hellman 在 Alice 和 Bob 之间建立新密钥,因为他们可能不同时在线。因此,在 X3DH 协议 [1227] 中,每个用户 U 向服务器发布一个身份密钥 IKU和一个预密钥 SKU ,以及后者的签名,后者可以使用前者进行验证。这些算法是椭圆曲线 Die-Hellman 和椭圆曲线 DSA。当 Alice 想给 Bob 发送消息时,她从服务器获取 Bob 的密钥 IKB和 SKB ,生成一个临时的 Die-Hellman 密钥EKA,并以所有可行的方式将它们与 Bob 的密钥组合： DH(IKA,SPKB), DH (EKA,IKB)和DH(EKA,SPKB)。这些被散列在一起以提供新的密钥KAB。然后,Alice 向 Bob 发送一条包含她的密钥 IKA和EKA的初始消息、她使用了 Bob 的哪些预密钥的注释以及使用KAB加密的密文,以便他可以检查他是否也得到了它。 Bob 可以选择上传一个一次性的临时密钥,Alice 会将其与 EKA 结合并散列到组合中。

给定一个初始的 Die-Hellman 密钥KAB， Alice 和 Bob 然后使用双棘轮算法为各个文本和呼叫派生消息密钥。其目的是在他们的一部手机遭到破坏时恢复安全性。它使用两种机制:密钥派生函数 (KDF) 或单向哈希函数来更新存储的密钥,以及进一步的 Die-Hellman 密钥交换。 Alice 和 Bob 每个都维护单独的 KDF 链用于发送和接收,每个链都有一个共享密钥和一个 Die-Hellman 密钥。每条消息都携带一个新的 Die Hellman 密钥部分,该部分与相关链的密钥相结合,而共享密钥通过 KDF 传递。由于需要处理无序消息 [1512],因此实际细节稍微复杂一些。目标是对手必须连续破坏 Alice 的电话或 Bob 的电话才能访问他们之间的通信。

真正棘手的部分是初始身份验证步骤。如果 Charlie 可以接管服务器并向 Alice 发送他自己的 IK 而不是 Bob 的,则所有赌注都是 o 。这是一些情报机构对消息传递应用程序发起的攻击。

Apple 的 iMessage 等系统不仅会向您的交易对手发送单个身份密钥 KI,还会向您的交易对手发送一整套设备密钥 一个用于您的每台 MacBook、iPhone 和其他 Apple 设备。 GCHQ 的伊恩·利维 (Ian Levy) 和克里斯平·罗宾逊 (Crispin Robinson) 提议使用英国的调查权力法案等法律来强制提供商向他们获得授权的任何用户的密钥环添加额外的执法密钥 [1153]。这导致了美国、英国和其他地方的政策争论,我将在第 26.2.8 节中返回。保持这种监视的秘密将取决于手机应用程序软件对用户保持不透明;否则双棘轮算法将阻止 Alice 和 Bob 的私人对话被 Charlie 作为无声电话会议伙伴或“幽灵用户”加入。 Signal 试图通过开源来阻止这种情况。

结果是,如果 Charlie 想在假装是 Bob 的同时与 Alice 交换 Signal 消息,他必须要么破坏 Bob 的电话,要么窃取

鲍勃的电话号码。这些选项与他想从 Bob 的银行帐户中窃取钱的选项非常相似。它们包括入侵和窃取电话;使用 SS7 漏洞窃取 Bob 的短信;以及 SIM 交换攻击以接管 Bob 的电话号码。对个人来说,最容易发动的攻击可能是 SIM 交换,我们在 12.7.4 节中讨论过。 Signal 现在提供一个额外的 PIN,您在恢复之前使用不同手机的电话号码的服务时需要输入该 PIN。民族国家拥有复杂的黑客工具,并拥有 SS7 访问权限 因此,如果 FSB 在你的威胁模型中,最好使用他们不知道号码的电话,并且不要随身携带,同时开机他们确实知道手机是你的,或者他们可能会关联这些痕迹 正如我在第 2.2.1.10 节中描述的那样。

正如我们将在 26.2.2 节中讨论的那样,信号情报的大部分好处来自于元数据,来自于知道谁在什么时候给谁打电话(或者谁在什么时候和谁一起旅行)。因此,对于举报人来说,游戏取决于有多少人和你一样成为嫌疑人 匿名集。如果你是一位高级公务员,想向报纸泄露一项非法政策,并且你是知道这个故事的十个人之一,那么你可能是这十个人中唯一一个曾经使用过 Signal 的人。

但是,如果您是数百名低级别嫌疑人中的一员(假设您是工会组织者或非政府组织的工作人员),并且可能在一长串主题收集目标中,那么您可能想要阻止当地警方通过系统地记录您的联系模式,Signal 在这里确实可以提供帮助。它提供了私人联系人发现的有趣创新。

以前帮助普通人使用端到端加密的尝试,例如电子邮件加密程序 PGP,在专业领域之外从来没有得到太多关注,因为密钥管理太麻烦了。消息应用程序通过要求访问你的地址簿来解决可用性问题,在它们的服务器上查找你的所有联系人以查看还有谁是用户,然后标记他们以便你知道你可以向他们发送消息。然而,给服务公司一份你的通讯录副本已经是一种隐私妥协,如果你还让他们保留你的社交图谱、个人资料名称、位置、群组成员资格以及谁在向谁发送消息的明文记录,那么调查人员就可以得到所有这是通过传票。 Signal 的原始版本通过比较人们通讯录中电话号码的哈希值来发现谁在使用它;然而,Christof Hagen 及其同事在 25 天内使用 100 个帐户扫描了美国所有 5.05 亿个电话号码,发现了 250 万个 Signal 用户 [848]。 Signal 现已实现私人联系人发现;我将在稍后讨论 SGX 及其使用的机制的第 20.6 节中讨论它。然而,当您在手机上设置 Signal 帐户时,即使是私人联系人发现也会让您地址簿中也在使用 Signal 的每个人立即明白这一事实(他们会说 ‘嘿,弗雷德要泄露一些东西’ 所以一个小心的泄密者会用现金购买一次性手机。)

系统中一个关键但不太明显的部分是消息服务器。这必须存储尚未传送的加密消息,但还有多少消息会被保留以及保留多长时间2? Signal 保留群组成员的记录,但现在有一项匿名群组消息的提议,这将使

2关于密钥更改时如何处理未传递的消息存在争论,并且 WhatsApp 实施因将交付优先于失败关闭而受到批评。

组成员彼此认识但 Signal 的服务器不认识 [409]。同样,技术只能做这么多;如果你的团队中有一个成员不忠诚,他们就会背叛其他人。然而,作为公众可用的领先通信安全工具,Signal 已经获得了真正的关注。 2016 年大选后美国的使用率显着上升,2020 年欧盟委员会（欧洲公务员制度)在包含数千条外交电报的服务器遭到破坏后命令其工作人员切换到 Signal [399]。

2020 年 7 月发生了一件令人不安的事情,当时 Signal 更新强制用户选择 PIN,以期将每个用户的联系数据加密保存在飞地中,以便在用户获得新手机时可以恢复,因此除了共享电话号码之外,还可以通过其他方式建立 Signal 联系。

这引发了一场抗议风暴,因为用户认为 Signal 也会保留消息内容;其他用户认为 PIN 无法提供足够的保护,或者不想向 Signal 提供他们用于银行业务的 PIN,或者根本不喜欢任何集中数据的想法。人们开始质疑依赖安全通信应用程序是否明智,该应用程序的主要维护者是使用假名的人,可以随心所欲地挟持数百万用户,其支持部分来自政府,部分来自亿万富翁3。公共利益关键基础设施的治理应该是什么样的?

Signal 声称不保留任何 trac 记录,但如果来自 NSA 的 FISA 授权令迫使他们这样做并撒谎怎么办?这给我们带来了一个更棘手的问题,即如何使通信匿名。

20.4 托尔

洋葱路由器 (Tor) 是人们用来在线实现严格匿名的主要系统,2020 年有大约 200 万并发用户。它于 1998 年在美国海军研究实验室开始使用,被称为洋葱路由,因为其中的消息是像洋葱层一样嵌套 [1590]。如果 Alice 想在 Eve 或其他任何人无法识别她的情况下访问 Eve 的网站,她将建立一个到 Bob 操作的 Tor 中继的 TLS 连接,Bob 将建立一个到 Carol 操作的 Tor 中继的 TLS 连接,而 Carol 又是一个TLS 连接到由 David 操作的 Tor 中继　从他的“出口节点”Alice 现在可以建立到 Eve 网站的连接 [1360]。这个想法是将路由与身份分开　任何想要将 Alice 链接到 Eve 的人都必须颠覆 Bob、Carol 和 Dave,或者监控进出 Bob 和 David 系统的流量。

灵感来自于 1981 年 David Chaum 的想法,混合或匿名邮件转发器 [410]。它接受加密的消息,剥离加密,然后将它们重新邮寄到它在里面找到的地址。 1990 年代,许多人对这些进行了试验,发现还需要三样东西才能使其正常工作。首先,您需要不止一种组合;对手可以通过胁迫操作员或简单地关联进出流量来妥协单一混音。其次,您需要针对您想要保护的流量进行设计,无论是电子邮件、网络还是消息。第三,也是最困难的一点,你需要规模。

3Brian Acton,WhatsApp 的创始人之一。

海军在 2003 年向全世界开放了 Tor,因为你只能在人群中保持匿名。如果 Tor 仅限于美国情报人员使用,那么任何使用它的人都将成为攻击目标。它现在由 Tor Project 维护,这是一家维护 Tor 浏览器的美国非营利组织,它已成为默认的 Tor 客户端。这不仅处理电路设置和加密,还管理 cookie、javascript 和其他对隐私有害的浏览器功能。类似的功能也内置于其他一些浏览器中,例如 Brave。还有用于 Tor 中继的软件,由具有高带宽连接的志愿者运行； 2020年约6000个活跃中继服务约200万用户。当您打开支持 Tor 的浏览器时,它会通过找到三个 Tor 中继来打开电路,通过这些中继连接到外部世界。

面对各种威胁和滥用,Tor 的加密和软件设计已经发展了 20 多年,现在它被用作许多应用程序中的一个组件。它被用来在伊朗和巴基斯坦等国家击败审查制度,因此您可以连接到 Facebook 并阅读美国和欧洲的报纸。美国国务院支持它,Facebook 是最大的 Tor 目的地。

它还可用于连接到地下黑市,您可以在那里购买毒品和恶意软件。它可以用来泄露机密文件。它可用于访问儿童性虐待网站。警察也用它来访问这些网站,所以操作员不知道他们是警察。

主要漏洞从第一天起就为人所知,并记录在 1998 年的论文中,该论文向世界介绍了洋葱路由,比 Tor 本身出现早了六年 [1590]。但他们经常被粗心的用户所忽视。首先,如果 Eve 的网站不使用加密,或者如果她以出口节点可以进行中间人攻击的方式使用它,则恶意出口节点可以监视 trac。 2007 年 9 月,有人设置了五个 Tor 出口节点,监控通过它们的流量,并发布了有趣的研究 [1359]。这包括使馆使用的许多网络邮件帐户的登录名和密码,包括来自伊朗、印度、日本和俄罗斯的使团 4。然而 Tor 文档清楚地表明可以读取出口跟踪,因此更谨慎的外交官会使用支持 TLS 加密的邮件服务,就像 Gmail 那时已经做的那样。

第二个问题是网页用于跟踪用户的许多技巧。
这是 2008 年推出 Tor 浏览器的主要原因,它限制了 cookie 和其他指纹识别机制的跟踪能力。但是许多应用程序让用户明确表明自己的身份,或者在没有意识到的情况下泄露信息。在 11.2.3 节中,我讨论了 AOL 的所谓匿名搜索历史如何识别用户:一些本地搜索（告诉你住在哪里)和一些特殊兴趣搜索（揭示你的爱好)就足够了。

第三,诸如 Tor 之类的低延迟、高带宽系统有一些固有的 trac 分析[1363]。像 NSA 这样的全球对手在 Internet 的许多点上窃听 trac,只需要窃听少量交换点就可以获得足够好的样本来重建电路 [1365]。在实践中,这比看起来更难5。 Tor 从一开始就明确表示

4这让我们深入了解密码选择:乌兹别克斯坦的密码名列前茅,例如 s1e7u0l7c ,而突尼斯只使用了 突尼斯 和印度大使馆 1234 。

5 情报界在 Tor 泄露的 GCHQ 幻灯片上称赞了 Tor

它不能防止 trac 确认攻击,在这种攻击中,对手控制进入和退出继电器并关联 trac 的时间、音量或其他特征以识别特定电路。事实上,在 2014 年,人们发现有人（大概是情报机构)一直在这样做,自愿将中继中继到修改协议标头的系统中,以使其更容易 [561]。 Tor 中继现在有针对此类调整的对策,但 trac 确认仍然是一个威胁。

第四,由于 Tor 通过大约 6,000 个中继池进行连接,防火墙可以简单地阻止它们的 IP 地址。一些公司和一些国家（尤其是中国)正在这样做。为了规避这种封锁,志愿者提供了 Tor 网桥 未在公共目录中列出的 Tor 入口节点。

中国和其他审查员也试图找到并阻止这些游戏,并描述 Tor trac 的特征,因此玩了各种游戏。中国似乎更喜欢绕过其国家防火墙的人改用 VPN;这些不仅更具可扩展性,而且在危机时刻（例如在 2020 年冠状病毒爆发的早期阶段)更容易完全关闭。

执法机构曾多次设法找到并关闭 Tor 洋葱服务,这些网站只能通过 Tor 网络访问;他们有一个“.onion”地址,而不是一个普通的 URL,它本质上是一个加密密钥。最著名的此类服务是丝绸之路,这是一个人们买卖毒品的地下市场;它的操作员因操作安全性差而被捕（他用来宣布新服务的电子邮件地址可以追溯到他）。其他洋葱服务的服务器遭到黑客攻击,或供应链被追踪。他们中的许多人使用加密货币,我们稍后将对此进行描述,并且还可以通过各种方式进行追踪。 Tor 用户的浏览器也曾受到零日和沙箱逃逸等技术的攻击。即使没有技术故障,匿名本质上也很难;真实世界的交易（实际上是真实世界的网络流量)可能非常脏,因此经常会得出意想不到的推论。

与 FDE 一样,Tor 与合规性有很大的关系,帮助各种参与者逃避监视并规避好的和坏的法律。引擎盖下的工程比看起来复杂得多。它肯定会带来性能损失 网站可能需要一秒钟的时间来加载,而不是几百毫秒。尽管 Tor 系统本身很稳健,但它具有内在的局限性,这些局限性在直觉上并不明显,这使得建立在其上的匿名系统使用起来很危险。匿名系统需要谨慎的操作安全性以及正确的软件。

治理方面很有趣。 Tor 由 Tor Project 维护,这是一家成立于 2006 年的美国非营利组织,旨在使 2002 年开始的志愿者项目正式化。尽管它有很多志愿者,但多年来,越来越多的核心永久员工获得了各种来源的资助,来自EFF 到美国国务院。它的核心仍然是一个以人权为动力的国际社会。 Ben Collier 的一项人种学研究将其描述为由三个重叠的群体组成:一群工程师将 Tor 视为一个结构,并相信政治问题可以通过做 engi 来解决

Ed Snowden,说“Tor 很臭！”

学习;一群积极分子将其视为一场斗争,并致力于特定的政治价值观,例如反种族主义;而第三组人主要维护 Tor 中继,通常在政治上是不可知论者,并将他们所做的视为提供基础设施 “隐私即服务”[458]。大规模安全需要基础设施,而要在很大程度上通过志愿者努力提供这一点,就需要能够在不同利益相关者的议程之间进行转换并协商价值而不仅仅是合同的领导者。

20.5 HSM

在银行业务和簿记一章中,我们描述了银行如何使用 HSM 来执行职责分离政策:银行的任何一个人都不应该能够获得客户的卡详细信息和 PIN。 HSM 还用于保护许多网站的 SSL/TLS 密钥;您不希望重要的实时密钥位于开发人员的笔记本电脑上,或者云提供商可以通过内存转储轻松提取。在加密货币行业,HSM 用于保护可以签署大量资产的密钥。在防篡改章节中,我们描述了用于使 HSM 防篡改的机制。但这还不够。您还必须确保当您在更受信任的组件(例如 HSM)和不太受信任的组件之间拆分计算时,攻击者无法利用该拆分。

每当一台受信任的计算机与不太受信任的计算机对话时,您必须预料到不太受信任的设备会撒谎和欺骗,并通过使用意想不到的命令组合来探测边界,以欺骗更受信任的设备。我们如何系统地分析这一点?

银行 HSM 有很多东西要教。 1988 年,Longley 和 Rigby 在为安全模块供应商 Eracom [1184] 工作时确定了分离密钥类型的重要性。 1993 年,我们报告了一个由添加到安全模块的自定义事务引起的安全漏洞 [107]。然而,我们在 2000 年取得了成功,当时 Mike Bond、Jolyon Clulow 和我观察到 HSM API 变得非常复杂,数百种不同的交易涉及加密操作的复杂组合以支持数十种支付协议变体,并开始系统地思考是否可能有一系列 HSM 交易会破坏它 [71]。我们问:“你怎么能确定不存在会泄露明文密钥的 17 笔交易链?”在我们花了一些时间盯着手册看之后,我们开始发现许多此类漏洞。

20.5.1 xor-to-null-key 攻击

HSM 由银行服务器或现场 ATM 发送给它们的交易驱动。 HSM 包含许多保存在篡改响应内存中的主密钥。大多数密钥都存储在设备外部,使用一个或多个主密钥进行加密。在用于管理它们的数据库中管理 ATM 和其他终端的密钥很方便;现在,许多 HSM 位于 Azure 和 Amazon 云中,它们为多个租户提供服务。

加密的工作密钥有一个类型系统,可以按功能对它们进行分类。例如,在用于安全模块的 PCI 标准中,PIN 派生密钥 用于从帐号派生 PIN 的主密钥,如第 12.4.1 节所述 在一对特定的主 DES 密钥下加密存储,以便将其标记为不可导出的工作密钥。 ATM 的终端主密钥属于同一类型,您会从 12.4.1 节回忆起 ATM 安全策略是双重控制,因此银行为两个 ATM 保管人生成单独的密钥,比如分行经理和分行会计师,他们在设备调试时或在服务访问后在键盘上输入它们。因此,HSM 具有生成密钥组件并在连接的安全打印机上打印出来的事务。它还将其加密值返回给调用程序。还有另一个交易结合了两个组件来生成终端主密钥:给定两个加密密钥,它会解密它们,异或一起,并返回结果 以标记为不可导出的方式加密工作键。

攻击是将一个密钥与其自身结合,产生一个已知密钥 全零的密钥 标记为不可导出的工作密钥。因为有进一步的交易,它会用任何其他的加密任何不可导出的工作密钥,你现在在家里干了。您可以通过使用全零密钥对其进行加密来提取皇冠上的珠宝 PIN 派生密钥。您现在可以解密 PIN 派生密钥并计算出任何客户帐户的 PIN。

HSM 已被击败。

上述攻击多年未被发现。该文档没有详细说明设备中各种类型的密钥应该做什么;不可导出的工作密钥只是被描述为“在主密钥 14 和 15 下加密提供的密钥”,并且交易将一个这样的密钥加密在另一个密钥下的含义并不是很明显。事实上,HSM 只是从早期的、更简单的设计演变而来,因为 1980 年代引入了 ATM 网络,银行要求提供更多功能,以便它们可以使异构网络相互通信。

因此 Mike Bond 建立了设备中使用的密钥类型的正式模型,并立即发现了另一个缺陷。您可以为 HSM 提供一个帐号,假装它是一个 MAC 密钥,然后使用 PIN 验证密钥对其进行加密 这也会直接为您提供客户 PIN。使困惑?

最初每个人都是 现代 API 太复杂了,以至于在不经意的检查中无法发现错误。无论如何,完整的细节在 [100]。最新的 HSM 具有强类型,可以更轻松地对键进行正式推理。

20.5.2 使用向后兼容性和时间的攻击 内存权衡

我们与 HSM 供应商 nCipher 合作,他们向我们提供了他们竞争对手产品的样本,因此我们可以破解它们不仅是为了帮助他们的营销,而且是为了让他们能够将客户密钥材料迁移到他们自己的产品中。当时的首要目标是 IBM 产品 4758 [951]。这是唯一通过 FIPS 140-1 4 级认证的设备;实际上,美国政府曾表示它牢不可破。事实证明它容易受到向后利用的攻击

相容性 [279]。

由于 DES 在 1980 年代变得容易受到密钥搜索的攻击,银行开始迁移到双密钥三重 DES:每个块都用左密钥加密,用右密钥解密,然后再用左密钥加密。

这个聪明的想法提供了向后兼容性:如果将左密钥设置为与右密钥相等,则加密将恢复为单 DES。 4758 分别存储左密钥和右密钥,并对它们进行不同的加密,赋予它们不同的类型 但未能将三重 DES 密钥的两半绑定在一起。您可以将单个 DES 密钥的“左半部分”加上另一个的“右半部分”,将它们放在一起形成一个真正的三重 DES 密钥,然后使用它来导出其他密钥。

因此,要破解 4758,您所要做的就是进行单一 DES 密钥搜索。这在现在并不算太难,但在 2002 年仍然是一项相当大的工作。幸运的是还有另一个漏洞 时间记忆权衡攻击。那一代 HSM 具有密钥的“检查值” 每个密钥的单向哈希值,通过加密一串零来计算。假设您需要一个特定类型的 DES 密钥。您预先计算了一个包含（比方说） 240 个键及其散列的表。您让 HSM 生成所需类型的密钥并输出哈希值,直到您看到表中已有的哈希值。这需要大约216个哈希,这需要一个小时左右 [447]。向后兼容性和时间记忆权衡攻击是针对 HSM 平台本身而非 PCI PIN 管理应用程序的 API 攻击的示例。

20.5.3 差异协议攻击

4758 个错误得到修复,最新型号的 ATM 提供用于自动注册的公钥机制。但是遗留的密钥管理和 PIN 管理机制仍然存在于应用程序层,因为很难改变拥有数百家供应商和数千家银行的分布式系统的架构。还有更多的事情要做。下一波针对 HSM API 的攻击是由 Jolyon Clulow 在 2003 年发起的;他们对应用程序逻辑进行主动操作以泄露信息。许多 HSM 支持为特定应用程序量身定制的交易;最大的细分市场是支持卡支付,不过也有用于预付费电表、认证机构甚至核指挥与控制的 HSM。

Clulow 的第一次攻击利用了错误消息 [449]。我在第 12.4.2 节中描述了刚刚将客户的加密 PIN 写入银行卡的银行是如何受到攻击的,因为客户可以将帐号更改为另一个帐号并使用他们的 PIN 盗用该帐户。为了阻止此类攻击,Visa 引入了一种可选的 PIN 块格式,该格式在加密之前将 PIN 与帐号进行异或运算。但是,如果错误的帐号与 PIN 块一起发送,HSM 会对其进行解密,对帐号进行异或运算,当结果不是十进制数时,它会返回一条错误消息。因此,通过使用各种错误的帐号向 HSM 发送几十笔交易,您就可以计算出 PIN6。现在有专

6现在有四种不同的 PIN 块格式用于 PIN 传输,其中三种也包括 PAN;还有另一种格式,即 PIN 验证值 (PVV),它是 PIN 和 PAN 的单向加密,由银行发送到交换机,例如

VISA 和 Mastercard 如果他们希望交换机在他们自己的时候进行替代 PIN 验证

关于 PIN 转换的 HSM 的 PCI 规则 [977]。复杂性引发了新的攻击,需要更复杂的方法来修补它们。

Mike Bond 和 Piotr Zielinski 随后发现了另一类攻击。
回想一下 IBM（以及大多数行业)用于生成 PIN 的方法,如第 12 章图 12.3 中所示。主帐号使用 PIN 验证密钥加密,提供 16 位十六进制数字。前四位转换为十进制,虽然大多数银行通过取十六进制数字模 10 来执行此操作,但并非所有银行都这样做。 HSM 供应商通过十进制表对操作进行参数化,默认值为 0123456789012345,它只是将十六进制输出以 10 为模减少。这是一个大错误。

如果我们将十进制表设置为全零（即 0000000000000000）,那么 HSM 将返回“0000”的 PIN,尽管是加密形式。然后我们使用表 1000000000000000 重复调用。如果加密结果发生变化,我们知道 DES 输出的前四位数字中包含 0。给出几十个查询,就可以推断出 PIN。比较重复但略有修改的相同协议运行的攻击,我们称为差异协议分析。唯一真正的解决方案是向 HSM 供应商支付额外费用,购买一台硬编码了您自己银行的十进制表的机器。当您想将银行迁移到云端并共享由 Amazon 或 Azure7 维护的 HSM 时,这可能会导致更多问题。

在哲学层面上,这说明了设计一个健壮的安全多方计算的困难 一种使用来自一方的秘密信息的计算,但也有一些可以被敌对方操纵的输入[99]。即使在这种极其简单的情况下,它也非常困难,以至于您最终不得不放弃 IBM 的 PIN 生成方法,或者至少很难确定其参数,以至于您最好不要首先调整它们。

在实际层面上,它说明了 API 随着时间的推移而失败的主要原因之一。
它们变得越来越复杂,以满足越来越多客户的需求,直到突然出现攻击。

20.5.4 EMV攻击

您可能认为,在 2000 年代初发布第一波 API 攻击之后,HSM 设计者在添加新事务时会更加谨慎。然而,正如安全研究人员和 HSM 供应商发现并修复漏洞一样,银行业强制要求新的漏洞。

例如,EMVCo 订购的 HSM 功能支持智能卡和银行 HSM 之间的安全消息传递,在所有符合 EMV 的 HSM [22] 中引入了可利用的漏洞。目标是让银行能够命令其发行的任何 EMV 卡在下次进行在线交易时更改某些参数,例如密钥。因此,EMVCo 定义了一个事务安全消息传递,服务器可以通过它命令 HSM 进行加密

系统已关闭。
7 一个供应商规定一张表必须至少有八个不同的值,并且没有值出现超过四次。但这不起作用：0123456789012345,然后是 1123456789012345,依此类推。

一条文本消息,后跟一个用于与银行智能卡共享的类型的密钥。
加密可以采用CBC或ECB方式,文本消息可以是可变长度的。攻击是选择消息长度,使目标密钥只有一个字节穿过加密块的边界。然后可以通过发送一系列长一个字节的消息来确定该字节,并且额外的字节循环遍历所有 256 个可能的值,直到找到关键字节。

20.5.5 破解 CA 和云中的 HSM

最近一次 HSM 破解是在 2019 年,由 Jean-Baptiste B´edrune 和 Gabriel Campana 在 Gemalto HSM 上破解,其应用程序支持公钥密码学的 PKCS#11 标准,因此它可以在证书颁发机构中使用并作为TLS 加速器。 (众所周知,该标准晦涩难懂且难以实施。)他们获得了 HSM 的软件开发套件,其中包含该设备的模拟器,并对其进行模糊测试,直到发现多个漏洞。他们设法修补了身份验证功能,以便他们可以以管理员身份登录 HSM 并安装读取密钥的工具 [203]。这只是复杂的密码学被粗心的软件工程致命破坏的众多例子之一。

20.5.6 管理 HSM 风险

曾几何时,有人发现对市场上每个安全模块的至少一个版本的攻击。正如安全工程中经常出现的那样,根本原因是特征炎。人们使 API 变得更加复杂,直到它们崩溃。

银行仍然必须使用 HSM 来遵守 PCI 规则,但其中的加密密钥并不仅仅受到篡改响应外壳的保护。配置管理必须严格,供应商软件补丁必须及时应用,就像在其他系统中一样。但是,尽管任何规模的大多数银行都有了解软件安全和补丁生命周期的人员,但他们不太可能拥有真正的 HSM 专业知识。

专业公司提供 HSM 管理系统,我们必须看看这些系统最终是否会被大型云服务提供商纳入。云 HSM 的管理仍在进行中,Microsoft Cloud Key Vault 等产品允许密钥在 HSM 和提供类似功能的 enclave 之间来回移动。当然,如果 PIN 管理应用程序存在固有 API 漏洞,那么这些漏洞将与它是在传统的本地 HSM、云数据中心的 HSM 还是 enclave 上运行无关。

事实上,Microsoft 产品的一个卖点是“消除对硬件安全模块内部知识的需求”[1309]。

有了这个警告,是时候看看飞地了。

20.6 飞地

飞地就像 HSM,因为它们旨在提供一个平台,您可以在该平台上在您不完全信任的人操作的机器上安全地进行一些计算。早期的尝试涉及数字版权管理 (DRM) 机制,该机制混淆了代码以使其难以受到干扰8,随后是 2000 年代初期的“可信计算”计划。这提出了一种架构,其中 CPU 将执行加密代码,密钥存储在单独的可信平台模块 (TPM) 芯片中。正如我在第 6.3.2 节中所述,Arm 在 2004 年正式推出了 TrustZone。

TrustZone 通常在现代 Android 手机核心的片上系统 (SoC) 中实现,尽管其信任边界通常是整个主板;飞地数据可以在总线上和 DRAM 芯片中以明文形式提供。主要应用是移动电话,其供应商需要一种机制来保护基带免受用户篡改（出于监管原因)并使电话本身能够被锁定（以便补贴电话的移动网络运营商可以将它们与合同联系起来）。在这两种情况下,硬件攻击都不是真正的问题。

是否可以使用诸如 TrustZone 之类的飞地机制来加强电话银行系统以抵御我们在 12.7.4 节中讨论的那种攻击?为此,有人试图将其推向市场,但即使是编写银行应用程序的公司也不愿意采用它。直到 2015 年,它还是一个封闭的系统,只有 OEM 签名的代码才能在 TrustZone 中运行。

因此,想要“更安全”的身份验证组件的银行应用程序开发人员必须让三星为三星手机、华为为他们的产品等签名。更重要的是,代码会根据产品使用的 SoC 而有所不同。现在,要让应用程序在足够多的 Android 版本上稳健运行,而不必处理在不同 SoC 产品上运行的多个自定义版本的 TrustZone,已经非常困难了。评估供应商关于封闭平台的安全声明也很困难。有关详细信息,请参阅 Sandro Pinto 和 Nuno Santos [1529]。

2015 年,英特尔推出了 SGX,我在第 6.3.1 节中讨论了其访问控制方面。 SGX 飞地瞄准了一个更雄心勃勃的用例,即云计算。在 AWS、Azure 和谷歌等服务上运行系统变得更便宜:虚拟化让资源得到有效共享,因此数据中心、系统管理员等的成本可以分摊给成千上万的客户。但这引发了许多问题。您如何确定敏感数据不会泄露给云服务的其他租户,例如通过管理程序软件的技术漏洞?此类产品每年都会修复数十个漏洞 [479]。你有什么保护措施来防止一个民族国家使用授权来访问你的数据 实际上是对管理程序的合法利用?

云服务提供商本身也渴望有一种技术机制,可以让他们免去处理此类授权的麻烦。由于这些担忧,SGX 的安全边界就是芯片本身的边界。代码和数据在离开芯片时被加密,并在被导入缓存时被解密。 CPU 的硬件同时保护机密性和完整性。

8有关介绍,请参阅第二版中的“版权和 DRM”一章
这本书,在线免费提供。

关键加密机制是软件证明,它使 CPU 能够向软件所有者证明它正在运行,无需在可信赖的硬件上进行修改。 SGX 飞地作为应用程序运行,在第 3 环,CPU 机器将它们的代码和数据与下面的所有内容隔离开来,包括操作系统和管理程序 9。 enclave 初始化、地址转换、页面驱逐、异常处理等的全部细节极其复杂;有关解释和分析,请参见 Victor Costan 和 Srini Devadas [479]。他们提出的一个担忧是,除了内存加密外,SGX 是用微码实现的,可以更新;因此整个系统是可变的。还有多种侧通道攻击,特别是自从 Meltdown 和 Spectre 引入了侧通道攻击的瞬态执行系列之后,我在第 19.4.5 节中对此进行了讨论。有些已被修补,但真正的丑闻可能是英特尔已表示不会将修复 Membuster 攻击作为政策问题 10。

在这里,我关心的是用于支持 enclave 和证明其上运行的软件的密码学,以及它作为其他加密或应用程序的加密支持平台的适用性。

由于高端 CPU 使用的硅工艺不支持非易失性存储器,因此第一个问题是提供唯一且持久的芯片密钥。每个芯片都有熔丝,工厂在其中烧入一个密封秘密和一个配置秘密,英特尔不知道前者,但后者知道。这用于生成主派生密钥 (MDK),后者又可以跨电源循环可靠地生成密钥材料。配置密封密钥是永久性的,因此当计算机更改所有者时,英特尔不需要知道。这些密钥使 CPU 能够向英特尔证明其真实性,英特尔为其提供证明密钥 英特尔增强型隐私 ID (EPID) 中的成员私钥,这是一种旨在保护签名者匿名的组签名方案。

这些操作在特权启动飞地 (LE) 中完成。最初所有 SGX 代码都必须由 Intel 签名,但最近的版本允许第三方签名代码。每个 enclave 作者现在都是一个 CA 并证明每个 enclave,它有一个公钥、一个产品 ID 和一个版本号(秘密迁移只允许到更高的版本号以支持补丁但不能回滚)。

相同的棘轮适用于 CPU 微代码的更新。

一个问题是一个芯片的 MDK 的妥协 在任何地方的任何 CPU 中 破坏了同一组中每个 CPU 的认证安全。对于 AMD 的 SGX 等价物,这发生在 2019 年,当时微码中的一个错误使得这样的密钥能够被提取 [337]。英特尔以同样的方式易受攻击:鉴于 MDK 的明确价值,您可以在 SGX 的保护机制之外创建一个 SGX 飞地。如果发现此类中断,英特尔将不得不将同一 EPID 组中的所有 CPU 列入黑名单。我们不知道这些群体有多大,因为所有证明都是由英特尔不透明地完成的,用户必须简单地相信结果。

9 Trusted Computing Group 的早期提案要求 enclave 下的整个软件堆栈都经过证明和可信,这与不受信任的管理程序不兼容。

10SGX不防御cache时序攻击,所以在写enclave代码的时候,不能使用data-dependent jumps。更一般地说,它不能防止依赖性能计数器的软件侧信道攻击,但也不能为开发人员提供足够的信息来模拟可能的泄漏。

现在有一些 SGX 系统在做真正的工作。我在本章前面提到的一个例子是消息传递应用程序 Signal，它使用 enclave 来发现私人联系人。它的开发人员在 Signal 博客 [1226] 上发布了源代码以及对开发它的困难的广泛讨论。

目标是使 Signal 客户端能够确定其地址簿中的联系人是否也是 Signal 用户，而无需向 Signal 服务透露他们的地址簿。你怎么能在没有任何洞察力的情况下建立一个庞大的社交图谱呢？这个想法是客户可以联系飞地，验证它运行的是正确的软件，然后发送他们的联系人以查看谁也是用户。然而，在 SGX 飞地 (128Mb) 的内存限制内执行此操作需要仔细组织用户电话号码倒置文件的哈希表。

要防止内存访问模式导致信息泄漏，您还需要做很多事情：由于可能会通过此类模式观察到分支，因此代码的关键部分不得包含分支。简而言之，阻塞侧信道很像组织加密代码以在恒定时间内运行：繁琐、临时、手动且容易出错。 SGX 也很慢：虽然内存加密本身增加的开销很小，但上下文切换是一个杀手。检查与其他人的联系方式真的很慢，因此必须为多个加入者分批处理该过程才能接受。

SGX 应用程序的另一个示例是 Microsoft 的 Cloud Key Vault，它使 Azure 租户能够将密钥、密码和令牌等秘密与其代码分开存储 [1309]。有一个应用程序可以帮助您创建和管理 TLS 证书；秘密和密钥也可以存储在顶端的云 HSM 中，而如果您不必在代码中内联存储数据库密码，则常规应用程序可以更安全、更易于管理。

简而言之，编写好的 SGX 代码很难。工具链受到限制，并且排除了防病毒之类的东西。如果你很聪明，你可以编写可信的恶意软件。您甚至可以编写将在一个 SGX 飞地中运行的恶意软件，并对同一台机器上其他飞地中的代码进行定时攻击，使用 SGX 机制来隐藏自身以免受检测 [1689]。

即使您完全信任英特尔；即使您认为 NSA 不会使用 FISA 授权书来强制英特尔证明处于调试模式的 enclave；即使您不担心 MDK 妥协或利用侧通道 仍然存在应用层暴露的风险，就像 HSM 一样。

如果您编写的 enclave 代码可以被不太受信任的代码用作 oracle，那么您就有麻烦了。

英特尔（和 Arm）正在谈论他们的 enclave 技术的后续版本。与此同时，英特尔将加密货币开发人员指向他们的管理引擎 (ME)，这是 CPU 芯片组中的一个单独的微控制器，用于启动 CPU 并包含固件 TPM 以进行安全启动。如果报告机器被盗，它可以通过擦除密钥来阻塞 CPU。它的代码是专有的，基于 Minix，并由英特尔签名。它支持另一个具有 Java 可信执行环境的飞地，开发人员可以在其中进行加密；例如，在支付终端中，您可以设计一条从 ME 到密码键盘的硬件可信路径 [1698]。这使得加密代码能够免受 CPU 上的恶意软件的侵害，但也带来了自身的问题，例如涉及物理访问的攻击。

ME 也有一系列的漏洞和漏洞利用。这是考虑

EFF 将其作为后门,并且至少有一家供应商已将机器提供给政府,并在启动后将其关闭 。

20.7 区块链

前面关于密码学的用途和限制、密码学如何用于支持匿名性以及加密应用程序如何在堆栈的各个级别遭受缺陷的部分,让我们开始讨论加密货币和智能合约。在 2016-7 年间,加密货币是“热门”事物,在大数据和物联网以及人工智能和量子之后在炒作周期中占据一席之地。对许多人来说,“加密货币”一词现在指的是比特币而不是密码。

2008 年,有人使用中本聪的化名悄悄发布了比特币,并附有白皮书和实现 [1375]。这种匿名数字现金系统最初在密码朋克邮件列表上的爱好者和活动家之间传播,但在两年内它就走红了。

2011 年 2 月,一位名叫罗斯·乌布利希 (Ross Ulbricht) 的年轻自由主义者创立了丝绸之路,这是一个不受政府控制的在线市场。买家和卖家在 Tor 洋葱服务上相遇,可以使用比特币支付商品和服务费用。他们可以互相评价,就像在 eBay 上一样,并且有一个托管服务,这样买家就可以存入比特币,以便在货物交付时释放。丝绸之路迅速成为管制药物邮购供应的市场,在 FBI 于 2013 年 10 月逮捕 Ulbricht 之前,价值超过 10 亿美元的交易通过它进行 [421]。其他地下市场也采用了比特币。 Silk Road交易期间,价格从1美元左右涨到100多美元,不断上涨的价格吸引了投资者11。进一步的交易需求来自希望将资金转移出外汇管制国家的人们,这导致人们将比特币视为一种在危机时期可以购买的资产,就像黄金一样,从而产生了投资需求。到 2017 年,我们出现了泡沫 比特币的价格急剧上涨,突破了千美元大关,并在 2017 年 12 月达到了近 2 万美元的峰值。

比特币催生了多个模仿者 其中大部分是骗局,但也有一些真正的创新。助推器声称,加密货币将带来新一轮的创新和自动化浪潮,因为机器可以在没有人类或银行阻碍的情况下相互协商智能合约。在撰写本文时(2020 年),热情的高峰已经过去,但加密货币已成为投资者的新资产类别,并给金融监管机构和执法部门带来多重问题。

总而言之,比特币是一种迷人的密码学和经济学结构,它导致了一种支付系统的出现,它也是一种可信的计算机,来自数百万试图挖掘比特币的机器的分布式努力。除了编写软件的人之外,没有任何可信方,也没有参与者的预设身份。这些机制提供了一种在分布式系统中达成共识的新方法,与我们在

11当 Ulbricht 被捕时,比特币价格从 145.70 美元跌至 109.76 美元,但与其他人一样毒品市场开始运转,很快就恢复了。

第 7.3.1 节。这是将加密货币作为高级密码工程示例的原因之一;另一个是建立在它们之上的智能合约和其他第二层协议,尽管到目前为止它们对业务影响不大（数字交易所的总资本可能只有 10 亿美元左右）,但它们具有技术意义。

以下是基本机制的简要总结。

1. 比特币区块链是一个包含一系列交易的附加文件
   动作。

2. 用户以地址的形式出现在区块链上 假名是公钥的哈希值。

3. 大多数交易将货币从一个地址转移到另一个地址,方法是从之前的交易中获取未花费的交易输出（UTXO)并将其转移到一个或多个地址。这样的交易必须由UTXO地址对应的私钥签名。

4. 要进行支付,您签署交易并通过点对点网络将其广播给其他用户。其他用户可以自由选择一组请求的交易,检查它们是否有效,并将它们挖掘到区块链的新块中。

5. 每个交易块都由矿工通过块内容的 SHA256 散列和随机盐进行验证。矿工尝试不同的盐,直到哈希输出有足够多的前导零使其成为一个足够难的谜题。这样的哈希构成工作量证明,找到它们是一个随机过程,因此很难预测哪个矿工会找到下一个。区块链由哈希链和它们验证的块组成。拼图的难度会自动调整,以便大约每十分钟开采一个新区块。

6. 矿工每开采一个区块都会获得区块奖励;在撰写本文时,这是 12.5 个比特币,或超过 100,00012 美元。

7. 矿工还获得交易费,这是每笔交易的输入超过输出的金额。用户出价交易手续费以获得交易优先权;它们通常是几十美分,但在拥堵时可能会涨到几十美元。

8. 如果开采了两个相互竞争的下一个区块,则冲突由矿工开采最长链的规则解决。因此,直到大约六个其他区块被开采 对于经典比特币来说大约一个小时 交易才真正被认为是最终的。即便如此,大多数矿工可以通过构建一条延伸到更远的链 即所谓的链重组 来改写历史。

12 在 2020 年初,如果不考虑设备成本,可以以每千瓦时 5 美分的价格购买电力的矿工预计开采的比特币价值约为硬币在市场上的一半。然而,奖励会不时减半以限制比特币的总供应量,并且奖励将在 2020 年年中降至 6.25 个比特币。因此,投资矿机的人是在赌比特币价格会上涨,监管机构不会有效抑制需求。

9. 如果冲突没有解决,那么你可能会得到一个分叉 系统会产生两个不兼容的后继者。比特币在 2017 年因关于区块长度的政策纠纷而分裂为比特币和比特币现金,而在分叉之前拥有比特币的用户最终在两者中都拥有比特币。但有些分叉是有意为之的,除此之外,企业家们已经开始了数千个比特币克隆 其中大部分是骗局。

10.交易还可以包含脚本,使支付可编程。

有关详细说明,可参考三个标准。前两个是一组普林斯顿计算机科学家的技术阐述:2015 年由 Joe Bonneau、Andrew Miller、Jeremy Clark、Arvind Narayanan、Joshua Kroll 和 Ed Felten 发表的 18 页知识系统化论文 [293],同时在 308页面上有一本 2016 年的书,作者是 Arvind Narayanan、Joe Bonneau、Ed Felten、Andrew Miller 和 Steven Goldfeder [1383]。第三篇是 Rainer B¨ohme、Nicolas Christin、Benjamin Edelman 和 Tyler Moore 于 2015 年发表在《经济展望杂志》上的一篇论文 [274]。在撰写本文时,这些已经过时了,因此在下文中我将专注于此后的发展。我假设您知道详细信息,或者可以查找,或者不太在意。

要了解加密货币可能出现的问题,我们不仅要看密码数学,还要看更多的东西。一个常见的模式是,优雅的密码学思想被劣质的软件工程、缺乏系统思考和几乎完全不关心用户所辜负。

20.7.1 钱包

一开始,所有的比特币用户都是对等的:完整的客户端软件会挖掘比特币并让你花费你挖掘的硬币。但事情很快就开始专门为矿工定制钻机,为普通用户提供轻型客户端,它们不挖矿或存储整个区块链,但让买卖过程更易于管理。没有账户的内在概念,因为你通过知道将解锁一个或多个 UTXO 的私钥来拥有比特币。

钱包最初存储一个或多个私钥并提供一个界面,以便用户可以看到这些私钥可以花费的 UTXO(“我的比特币”)。钱包安全迅速成为一个大问题。从用户选择的密码短语生成私钥的所谓“大脑钱包”被攻击者对区块链上可见的公钥进行详尽搜索而破解;具有可猜测密码的大脑钱包通常会在 24 小时内清空 [1947]。

将签名密钥保存在硬盘上并受密码保护的软件钱包是一种改进,但容易受到恶意软件和其他攻击。

认真的运营商使用硬件钱包,它们本质上是小型 HSM,可以保持离线(所谓的冷钱包)。即便如此,众所周知拥有价值数百万美元比特币的人被武装劫匪劫持在家中并被迫转移比特币的情况也并非鲜为人知。如果你拥有比特币钱包的唯一物理保管权,那么你就像几个世纪前人们将储蓄存入金币时一样容易受到攻击。到 2013 年,我们看到了托管钱包的出现,其中交易所或其他在线服务提供商为您提供一切服务。这并不能真正解决抢劫问题,因为

强盗只会强迫你登录并付钱给他。但是托管钱包导致了广泛的其他欺诈和滥用行为,我将在下面进行描述。

20.7.2 矿工

随着比特币的普及和价值的增长,越来越多的人加入进来开采它们。
采矿设备使用 FPGA 出现,然后是 ASIC,它们在通用机器上的运行速度比软件快得多,以至于在几年内就被取代了。矿工在电力自然便宜的地方运营,例如冰岛和魁北克,但他们大多在俄罗斯或中国等可以与当地官员进行交易的地方。 2019 年加密货币挖矿的总能耗约为 75TWh,二氧化碳排放量超过 35Mt 与新西兰的碳足迹相当。截至 2020 年,每笔比特币交易消耗超过半兆瓦时并排放超过四分之一吨的二氧化碳。

矿工们将自己组织成少数平均收入的矿池。这些池的控制是不透明的。容量可以租用,有时用于在所谓的 51% 攻击中攻击加密货币。

区块链的全部意义在于通过创建防篡改、公开、仅可追加的交易日志来防止双重支出;但如果大多数矿工串通一气,那么他们就可以重写历史并多次花费硬币。早期,人们认为这样的攻击会立即对货币的可信度造成致命打击,但事实证明,事实要复杂得多。例如,2019 年 1 月,攻击者使用这种技术从 Ethereum Classic 窃取了超过 100 万美元,这是一种市值超过 5 亿美元的加密货币,链重组了数十个区块 [1428]。然而,其市场价值并未受到重大影响 。如果他们偷了大部分,价格就会暴跌,他们的战利品就会一文不值。 2020 年 8 月又发生了两次攻击,其中一次攻击者花费 192,000 美元购买了窃取 560 万美元所需的算力 [1519]。所以我们在推理区块链时需要仔细考虑博弈论和密码学;简单化的论点并不总是符合现实。

20.7.3 智能合约

比特币的脚本语言很简单,但后来的加密货币系统以太坊有一个图灵完备的虚拟机,其字节码通常是从一种叫做 Solidity 的语言编译而来的。以太坊已成为市值第二大的加密货币,因为它展现了可以自动执行复杂交易的智能合约的前景。在泡沫期间,许多初创公司谈到使用智能合约来激活物联网,并创建新的服务,例如分布式存储,人们可能会向其他人付费以使用他们的备用硬盘空间进行备份。这种分布式自治组织的想法在泡沫期间得到了大力提倡。这与“再去中心化”运动有关,该运动旨在使网络世界远离在 2000 年代主导网络世界的大型服务公司;虽然我们有很好的工具来分散静态只读内容的分发,但我们缺乏分散交易的好方法 [509]。截至 2020 年,主要应用似乎围绕交易,其中分布式交易所

（DEX)使人们能够在没有人为干预的情况下将一种加密货币换成另一种加密货币。（它们仍然只占总交易量的一小部分。）

这导致了有趣的新故障模式。尽管原始比特币区块链的共识机制被认为是激励兼容的,但当区块链上的交易代表矿工可以通过操纵共识提取额外价值时,情况并非如此。现在出现了套利机器人,它们通过抢先交易（预期和利用)来利用 DEX 的低效性。机器人抬高交易费用,在以太坊中称为gas;已经有数以亿计的此类优先 gas 拍卖,交易者争先恐后地为他们的交易争取优先权 [508]。理论上,如果机器人能够筹集到足够的资金,它们可能会接管市场的治理并掠夺它 [869];他们已经通过利用智能合约中的漏洞赚取了巨额利润 [1507]。

修复错误可能很昂贵。 2016 年,一个名为 DAO 的投资基金作为以太坊区块链上的智能合约成立,并从 10,000 多名投资者那里吸引了超过 1.5 亿美元的资金。攻击者利用合约中的一个缺陷来窃取资金 13,经过一番讨论后,以太坊软件被更改为将被盗资金转移到恢复账户。这导致了区块链的硬分叉,原始加密货币的持有者同时使用修改后的货币和“以太坊经典”,因为未修改的版本广为人知。

丹麦的一项研究说明了在实际应用环境中使用智能合约的更多问题。曾有人提议用它们来支付父母必须花时间照顾生病的孩子的费用,其中有复杂的法律规定,而文员经常会忽略这些规定,从而导致上诉。这个想法是将案件文件的哈希值放在以太坊区块链上,以便父母和上诉委员会都可以跟踪它们,希望自动化执行决定会减少官僚主义的拖延。但是内部人员、黑客和错误呢?地方政府往往会经常遭到黑客攻击,并最终支付勒索软件。当法律发生变化或发现错误时,谁来更新合约?区块链在设计上是不可变的,因此无法修补。但真正的交易破坏者是地方政府担心失去对流程的控制。两个进一步的问题包括这样一个事实,即人们经常不得不改变规则来完成研究 ,并且程序员更有可能使用不熟悉的语言（如 Solidity)而不是熟悉的语言（如 Python 或什至 Cobol)编写错误 - 一种众所周知的语言新语言的问题,我在 7.3.1.2 节中讨论过。

20.7.4 O链支付机制

标准的比特币交易可能需要六个区块或一个小时才能成为最终交易,在拥堵时甚至更长。这对于在线支付赎金或购买毒品来说可能已经足够快了,但与 EMV 相比,它并不令人印象深刻。更重要的是,比特币每秒约 5 笔交易的吞吐量是

13另一种观点是,如果合约接受代码的输出,那么
缺陷在于用户对代码功能的理解,在这种情况下,没有人偷走任何东西!

比不上Visa的50,000。

人们正在尝试使用侧链（第 2 层协议的一个示例)来解决此问题;此类协议在外部进行交易,但受限于第 1 层协议,例如比特币或以太坊。爱丽丝和鲍勃通过在第 1 层区块链上锁定硬币来打开通道,现在可以在他们之间进行快速交易。

关键思想是他们使用由两个条件传输组成的散列时间锁定合约 (HTLC) 相互提交一些加密货币。在这样的转账中,鲍勃向爱丽丝发送 h(R),其中 R 是一个随机数,爱丽丝在区块链的脚本语言中做出承诺,即“如果你在时间 t 之前给我看 R,我会给你这枚硬币。” Bob 做出了类似的承诺。

这为他们打开了一个通道来快速交易已签名的交易,直到他们决定结算并关闭通道。

需要更多的工程才能将其变成一个有效的支付系统。你需要一个争议解决机制,以防 Alice 和 Bob 不同意他们每个人应该从收益中拿走多少。然后你为爱丽丝建立机制,通过鲍勃向查理支付费用,并建立路由算法,这样你就可以把钱给任何人。从理论上讲,这可以是点对点的,但在实践中,此类系统似乎将自己组织成枢纽,具有始终开放的渠道,如银行网络。协议安全涉及确保诚实的用户即使在其他人串通的情况下也不会损失金钱。成本包括中间节点需要有足够的流动性来转发交易,以及所有活跃玩家都需要在线 其影响范围从热钱包的盗窃风险,到矿工在 Bob 广播 R 时领先的风险,网络故障后大规模崩溃的风险[831]。 2020 年领先的此类系统是闪电网络,它可以在几秒钟内完成支付,使拥有正确手机应用程序的人能够像微信支付一样通过二维码进行支付,现在每天处理 1000 笔交易。这里的限制似乎是流动性:虽然闪电链本身是无信任的,但它们会占用节点的容量,接收者必须决定是否接受它们。因此,恶意用户可以设置数百笔付款,让它们停留数小时,然后免费取消。由于 Lightning 的总资本似乎只有几百万美元,这可能会使它有些脆弱。监管机构也很有可能打击转发节点。

20.7.5 交易所、加密犯罪和监管

开采所有自己的硬币很不方便,到 2010 年,企业家们已经建立了可以用比特币换取传统货币的交易所。大多数都破产了,通常是因为它们被黑了,或者因为内部人员偷了钱并声称被黑了。 2011 年的领导者是日本的 Mt Gox,它在 2011 年的一次黑客攻击中幸存下来,但在 2014 年破产,声称它被黑客攻击损失了 4.6 亿美元。法庭案件仍在继续;当时的新闻报道称,内部控制和软件开发流程混乱 [1280]。

这还不是全部。 Mt Gox 的创新之一是在 2013 年期间成为托管交易所。不是将客户的比特币保存在单独的钱包中,在客户输入正确的密码后,交易所可能会或可能不会临时访问私钥， Mt Gox 开始将所有比特币存放在自己的钱包中,向客户展示一个名义账户

当他们访问其网站时平衡。它实现了我们在 18 世纪金融业中看到的从金商到银行的转变：客户不再在金库中拥有一袋特定的金币，而现在只对银行的全部资产提出索赔。受害者讲述了在他们的钱包被托管后，他们如何开始看到未经授权的传出交易。

Mt Gox 倒闭后的分析显示，其中许多交易甚至没有出现在区块链上。从 2013 年中开始，当你从他们那里购买比特币时，他们所做的只是向你展示一个网页，说你有一个比特币的余额。 （这就是今天有多少交易所在运作。）

比特币世界充满了骗局，看起来大多数加密犯罪的受害者都是被破产、被黑或声称被黑的交易所骗走了。即使在交易所存在的前三年，2010-13 年，40 家交易所中有 18 家倒闭了 [1339]。

比特币分析公司 Chainalysis 的一份报告得出结论，2018 年交易所因黑客损失了约 10 亿美元，其中大部分盗窃是由两个犯罪团伙实施的；其中一个与朝鲜有关。除此之外，买卖毒品和其他非法商品的地下市场的营业额为 6 亿美元，大约是 2017 年价值的两倍 [400]。

还有市场操纵。 John Grin 和 Amin Shams 提供的证据表明，在 2017 年的繁荣期间，比特币的价格受到涉及 Tether 的内幕交易的支撑，Tether 是一种与美元挂钩的数字货币 [822]，这增加了人们对许多加密货币的市场价格可能经常受到影响的前景的看法。非法操纵的结果。随后的研究证实了这一点，表明大部分现货交易是由不受监管的外汇交易产生的 [1615]。

除了市场操纵，迄今为止最大的单一加密货币骗局似乎是一个名为 PlusToken 的庞氏骗局，在组织者于 2019 年被捕之前，该骗局从中国公民那里净赚了约 30 亿美元 [864]。但比特币也影响了许多其他犯罪类型。勒索软件从 2001 年到 2015 年的每年约 2-300 万美元增加到每年 8 美元，因为比特币突然使赎金变得容易收集 [91]；这种犯罪类型正在稳步增长，尽管也通过礼品卡收取赎金 [1190]。到 2018 年，为网络犯罪分子提供服务的防弹托管站点正在转向加密货币，因为其他支付机制变得更加困难 [1452]。那一年，世界上最大的暗网儿童色情网站 Welcome to Video 在通过区块链上的比特币流量追踪其运营者后被关闭，因此加密货币的假名性质有其局限性 [551]。

总体而言，诈骗和其他滥用直接加起来约占加密货币交易量的 3%；除了可见的加密货币交易外，还有许多场外交易经纪人，其中约 100 家已被确认参与洗钱活动 [401]。定期交流也让执法部门的日子不好过。犯罪团伙可能会通过一个渠道将收益转化为比特币，在第二个国家将其转换成另一种不同的硬币，然后将其发送到第三个国家，在那里他们通过银行转账将其取出。

然而，尽管比特币使用假名，但区块链包含所有交易的永久记录。正如我们在许多上下文中所讨论的那样 从我们关于推理控制的章节到本章中关于 Tor 的部分 匿名是很难的。真实世界的交易和数据具有上下文并允许

要作出的推论。比特币用户已经尝试了各种技巧来使交易更加匿名,例如将付款分成许多较小的部分,将它们混合起来,然后重新组合它们 所谓的“混音器”或“混合器”。但是,如果你这样做,你的比特币就会因洗钱企图而受到污染;总共可能有 10% 的比特币至少被盗过一次,或者通过洗钱服务。 （有关分析,请参见 [116]。）

例如,俄亥俄州的一名男子在 2020 年因操作这种洗钱 3 亿美元的混音器而被起诉 [553]。还有一些加密货币使用进一步的加密技术提供更多隐私,特别是 Zcash 和 Mon ero。目前,Monero 提供最强的隐私性,其设计使得硬币可以使用软件进行挖掘;超过 4% 的硬币被运行在其他人机器上的恶意软件开采 [1529]。

各国政府一直在试图通过金融监管来反击。美国财政部的金融犯罪执法网络 (FinCEN) 在全球范围内推动反洗钱 (AML) 和了解你的客户 (KYC) 法规,这些法规已纳入当地法律,例如通过欧盟的第 5 条反洗钱指令。一些政府走得更远。例如,德国的监管机构 BaFin 使用现有的金融法规来坚持要求所有交易所获得许可； localbitcoins.com 是一种点对点交易所,它使个人能够相互买卖加密货币以换取现金,但没有申请,它在那里被封锁了。但在撰写本文时,最大的推动来自 2019 年的 FinCEN 咨询,该咨询要求加密货币 ex 更改以实施“旅行规则”,即任何处理超过 10,000 美元交易的人都必须识别发送者和接收者并提交可疑活动报告如果相关。交流直到2020年6月才能提出解决方案;至少有一个人因交易金额超过 10,000 美元而被罚款 [688]。

进一步的监管也在欧洲的议程上。 Mt Gox 主要有日本客户,而大多数中国人似乎使用 Binance,英国和美国的许多人使用 Coinbase。当一个英国或美国用户向另一个用户发送比特币时,交易很可能永远不会接近区块链:如果他们都是 Coinbase 客户,那么 Coinbase 可以简单地调整其比特币钱包网页上显示的余额。这立即引发了一个问题,即为什么交易所不像任何其他货币服务业务那样受到监管。在欧盟,电子货币指令似乎适用,但英国和德国的监管机构仅针对客户在交易所拥有的传统货币余额执行该指令;交易所认为,由于交易需求远小于投资需求,虚拟货币应被视为资产而非支付机制。但既然如此,为什么监管机构不要求交易所按照与股票经纪人相同的规则运作,让客户的比特币不能用于交易,而只是卖回市场,收益被发送到银行账户以前买过吗?

在我和同事对交易所业务和追踪被盗比特币机制的分析中,我们还建议应用支付服务指令,这将为交易所客户提供与银行相媲美的消费者保护 [116]。例如,值得注意的是,虽然银行对如何阻止 SIM 卡交换攻击表现出很大兴趣

对于他们客户的手机,大多数加密货币交易所根本没有兴趣 尽管交换凭证是 SIM 交换团伙的主要目标之一 [1449]。加密货币世界中的消费者保护是一项未完成的事业,欧洲和其他地方的监管机构正在努力解决这个问题。

20.7.6 许可区块链

围绕加密货币和区块链的炒作激起了商业兴趣,大约从 2015 年开始,从达沃斯回来的首席执行官们告诉他们的 IT 部门他们需要区块链。然后,首席信息官们不得不探索是否可以创建区块链来完成有用的工作,同时避免比特币的环境浪费、非法内容和非法行为者。这导致了 Hyper ledger 和企业以太坊联盟等倡议的产生,企业支持者开发了各种区块链工具和标准。许多涉及基于拜占庭容错而不是工作证明的许可区块链结构,并且仍然可以支持智能合约。其中一些使用 SGX 作为其共识机制的一部分,例如英特尔自己的经过时间证明 (PoET) 提案。还有许多其他建议的共识机制;有关调查,请参见 Bano 等人 [165]。

作为一个应用示例,摩根大通从 2015 年开始开发一个系统,该系统将使参与银行能够在区块链上输入抵押贷款,因此其脚本语言将允许交易员创建任意复杂的期货和期权。他们探索了许多设计权衡,例如在对抗性设置中的低延迟和安全性之间,以及如何扩展交易隐私以保持业务逻辑的私密性以及个体参与者的姓名 [1421]。一个结论是,对于绝大多数应用程序,您不需要区块链;一个前向安全的密封日志就可以了。

在区块链可能有用的地方,你不能使用公共区块链。最重要的是,区块链应用程序必须与遗留系统对话,并且必须不再可能造成应用程序安全错误或可用性危害。搞砸的事情已经够多了:例如,阿根廷在区块链上发布了其官方公报 (Boletin Ocial),并规定其具有法律效力,于是有人对其进行了黑客攻击,发布了有关冠状病毒的假新闻 [499]。这种现实世界的经验似乎正在抑制泡沫最初的繁荣。

也许最具争议的项目是 Libra,Facebook 提议创建一个价值与一篮子货币挂钩的支付系统。

这本应由金融、科技和其他公司组成的财团运营,但遭到中央银行的强烈反对,导致 Visa、MasterCard 和 PayPal 等主要金融机构退出。

20.8 失败的加密梦想

许多人提出了基于区块链的电子投票系统,因为它们被认为是不可变的,你可以使用加密技术在它们之上构建功能。这些提议是在对电子选举中密码学的可能使用进行了三十多年的研究之后提出的,以提供一个系统

同时匿名且可证明准确。事实上,在 2017-8 年的比特币繁荣期间,一个常见的学生项目提案是“通过将选举放在区块链上来解决世界和平”。

声称使用区块链的选举系统现已在俄罗斯和美国部署,但效果并不理想。 2018 年,莫斯科市三个选区的系统使用以太坊区块链进行计票,但在选举前修复两个加密漏洞时,计票与区块链之间的联系被打破 区块链在选举后不久就消失了 [782].同样在 2018 年,西弗吉尼亚州成为美国第一个允许部分选民使用手机应用程序投票的州。麻省理工学院的 Michael Specter、James Koppel 和 Danny Weitzner 对它进行了逆向工程,并发现了一些漏洞,这些漏洞会让攻击者暴露或改变选票,尽管该应用程序使用了与攻击无关的区块链 [1810]。

根据研究人员的说法,攻击者可以创建一个受污染的文件记录,使可靠的审计变得不可能 尽管区块链的卖点包括透明度和问责制。

区块链可以解决选举问题的想法让经验丰富的安全工程师感到绝望。你无法用这项技术修复选举,因为它无法解决选举被盗的问题。执政党不断改变规则并颠覆堆栈中各个级别的技术,从选民登记到竞选资金和广告规则,再到媒体审查、选民恐吓和可操纵的投票计划。

我们将在 25.5 节中更详细地讨论这个问题。

20.9 小结

从 1980 年代开始,许多人尝试将密码学用作系统安全某些方面的可信平台。正如我们在第 12 章中所描述的,商业密码学最初的杀手级应用是保护 ATM 中的 PIN,然后是更普遍的卡支付。许多密码学研究人员(包括我)开始希望我们能够解决其他经济和社会问题密码学问题。匿名通信将停止审查;匿名数字现金将保护我们的隐私;数字投票将使选举更难被操纵;阈值签名将帮助我们建立强大的内部控制系统;电子拍卖将打击腐败。这一时期的 Crypto 和 Eurocrypt 会议上的研究论文充满了这样的想法。一代人之后,随着人们对全球化技术的影响产生怀疑,可能是时候进行评估了。

我们的案例研究讲授技术要点、经济要点和政策要点。

技术要点是密码系统并不神奇;他们有错误,必须像其他任何东西一样进行修补。即使是最简单的应用程序,如 FDE,也会随着产品的成熟而变得复杂,并且在实现质量上千差万别。 HSM 是密码系统的另一个例子,它获得了越来越多的功能,直到这些功能破坏了它们,现在需要其他组件来阻止有针对性的攻击。 SGX 运行在如此复杂的处理器上,以至于

它很容易受到多种侧信道攻击,英特尔甚至不认为其中一些属于其威胁模型:如果一个有能力的有动机的对手可以在与您相同的机器上运行他们的代码,那么您基本上就完蛋了。区块链也是如此,它开发了最复杂的生态系统。当应用程序创造必要的激励时,即使是理性矿工没有动力重写历史的基本假设也开始失败。

同样,加密货币可以继续获取功能直到它们被破坏,而智能合约可以帮助这个过程。

经济点是我们看到部署的高级加密机制都伴随着巨大的成本。 HSM 的成本高于服务器。 SGX 有内存限制和上下文切换的实际性能开销。比特币矿工排放的二氧化碳与新西兰一样多。智能合约也许能够做一些聪明的事情,但实际上与其他软件相比,在规模和范围上都非常有限。成本是否值得,有一个精细的计算;随着维护成本的增加和系统陷入技术债务,这种计算可能会随着时间的推移变得更加不利。

政策要点是高级加密机制都与责任纠缠在一起。如果成功,作为其核心目标的一部分,他们似乎会获得满足某些监管的愿望或避免监管的愿望。因此,部署或维护它们的决定可能涉及微妙的外部因素。

硬件安全模块在卡支付系统中是强制性的,因为卡计划规则最终基于银行不对欺诈承担责任的愿望。 SGX 被视为一种向云计算服务客户保证的方式,他们可以保护他们最有价值的资产免受流氓系统管理员和情报机构的侵害。比特币及其众多克隆产品已成为规避从证券和支付法到反洗钱法规等一切事物的机制。真正的系统是出于战略原因而建立的,这往往意味着为其创造者创造或巩固权力 无论是市场权力还是政治权力。

至于加密货币,到目前为止,它们的波动性极大、容量有限、交易成本不可预测、缺乏治理且透明度有限。他们中的大多数人使用的工作量证明机制会导致合理的人可能认为不可接受的二氧化碳排放,并且他们在实践中的使用与各种犯罪活动纠缠在一起。虽然法律应捍卫私营公司和个人创造优惠券和航空里程等价值代币的权利,但一旦这些代币开始被用作货币,并且出现像银行一样运作的机构,立法者将它们视为银行是合理的。

立法者考虑碳税,或要求使用区块链的组织对其产生的二氧化碳进行核算也是合理的。

如果我们必须总结四十年来试图运用数学的魔力解决现实世界问题的经验,那很可能是 TANSTAAFL:天下没有免费的午餐。

研究问题

围绕分散化存在跨越密码学和系统安全边界的深层问题。去中心化协议倾向于 fos silise;我们仍在使用 1990 年代初期的电子邮件、DNS 和 BGP 机制,因为很难改变任何东西。端到端加密无法在 SMTP 电子邮件之上分层,尽管 PGP 做出了努力,但需要等待像 Signal 这样的新平台可以通过法令强制实施。

比特币提供了另一个例子。最初的密码朋克理想是一个完全去中心化的支付系统,提供一种交换手段和一种价值存储,而无需政府或其他主导参与者（如银行)的参与。但矿机的生产已成为垄断,由比特大陆控制,而 ASIC 全部来自台积电。绝大多数比特币用户依靠托管交易所来持有他们的加密货币,而这些交易所进行了大部分交易　去中心化交易所只占其中的 0.01%。托管交易所实际上变成了不受监管的银行。

在 Signal、Tor 和比特币等系统中,真正的共识不是加密的,而是社交的;这是开发商的共识。在 Tor 中,这是一个社区,而在加密货币的世界中,存在相互竞争的开发团队以盈利为目的。安全经济学可能比密码学更重要,我们已经看到智能合约如何创建可能破坏底层共识层的应用程序层激励。

一般来说,智能合约的可靠性如何？ API 安全问题的计算机科学方法一直是尝试采用形式化方法工具来证明接口是安全的。关于这方面的文献越来越多,甚至还有一系列的研讨会,但这些方法仍然只能处理相当简单的 API。智能合约遇到了类似的问题,由于难以更改它们以修复错误或响应不断变化的环境而变得复杂。毫不奇怪,许多用于设置 DEX 的智能合约都具有硬编码的管理密钥,可以在需要时启用人工干预。这只是谨慎的工程,但质疑这种“无需信任”的交流的意识形态理由。

进一步阅读

要加快使用 Tor 的速度,一个好的起点是 Tor 项目的文档页面。有关比特币如何工作的更多详细信息,请阅读普林斯顿书籍 [1383] 或 JEP 论文 [274],而有关追踪被盗比特币和加密货币监管的更详细观点,请参阅 [116]。有关集中化与隐私之间相互作用的讨论,请参阅 Carmela Troncoso 及其同事 [1910]。可以在 [1917] 中找到关于 2015 年消息应用程序状态的调查（Signal 从以前的消息传递和 VOIP 应用程序中汇集到一起的时间）。