经典译丛

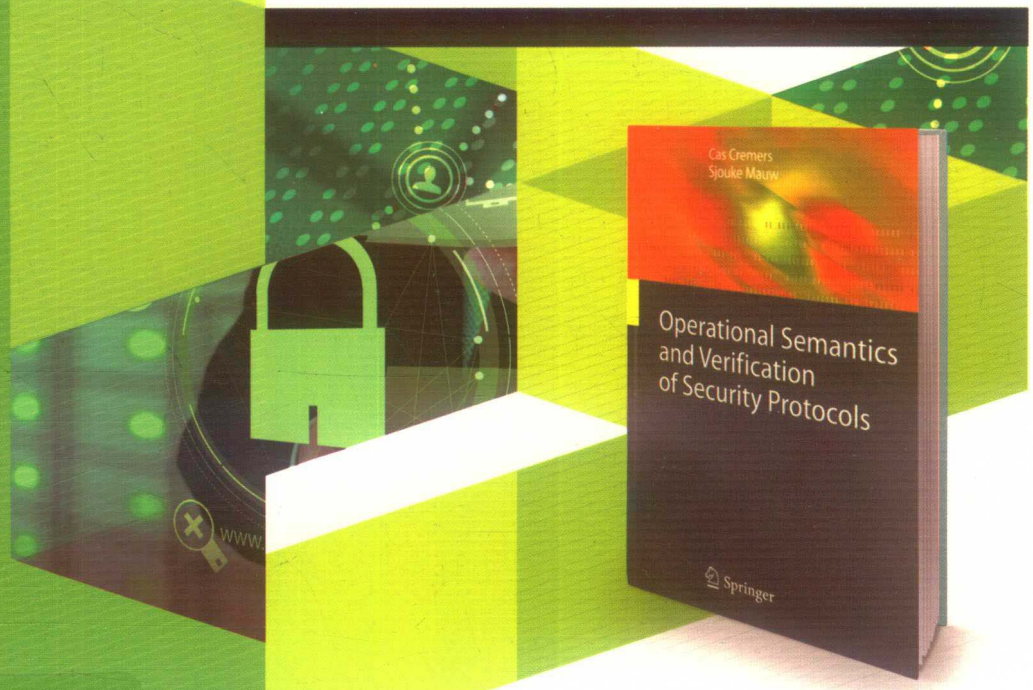
网络空间安全

安全协议操作语义与验证

Operational Semantics and Verification of Security Protocols

【瑞士】 Cas Cremers
【卢森堡】 Sjouke Mauw 著

吴汉炜 译 卿斯汉 审校



Springer

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

经典译丛·网络空间安全

安全协议操作语义与验证

Operational Semantics and Verification of Security Protocols

[瑞　士] Cas Cremers
[卢森堡] Sjouke Mauw　著

吴汉炜　译
卿斯汉　审校

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

安全协议作为信息安全的重要基础之一，其安全属性能否达到设计者的初始目标成为一个重要研究内容，关系到依赖于协议的上层应用系统的安全性。本书的内容主要涵盖两部分：用形式化的语义定义协议的执行规格和安全属性，精确表示安全协议的安全属性；综合运用各种形式化方法设计一个高效的验证算法，在可接受的时间内验证安全属性。本书还探讨了多协议安全分析，比较分析了各种验证理论和发展趋势。

本书可作为高等院校信息安全、计算机和通信等专业的教学参考书，也可供从事相关专业的教学、科研和工程技术人员参考。

版权贸易合同登记号 图字：01-2017-5381

图书在版编目(CIP)数据

安全协议操作语义与验证 /（瑞士）卡斯·克雷默斯(Cas Cremers)，（卢森堡）肖克·毛弗(Sjouke Mauw)著；吴汉炜译. —北京：电子工业出版社，2018.11
（经典译丛. 网络空间安全）
书名原文：Operational Semantics and Verification of Security Protocols
ISBN 978-7-121-35195-2

I. ①安… II. ①卡… ②肖… ③吴… III. ①计算机网络－网络安全－通信协议－操作语义－验证
IV. ①TP393.08 ②TP301.2

中国版本图书馆 CIP 数据核字(2018)第 230371 号

策划编辑：杨 博
责任编辑：谭丽莎
印 刷：北京天宇星印刷厂
装 订：北京天宇星印刷厂
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编：100036
开 本：787×1092 1/16 印张：9.25 字数：203 千字
版 次：2018 年 11 月第 1 版
印 次：2018 年 11 月第 1 次印刷
定 价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：yangbo2@phei.com. cn。

译 者 序

伴随着计算机技术和通信技术的发展，人们的生活方式也发生了极大的改变。移动终端的普及、大数据分析和各种人工智能应用悄悄地改变了人们的生活方式，信息安全问题也日益突出，上关国家安全，下系个人隐私保护。各种应用都依赖于底层的通信协议，为了确保协议的安全性，设计者往往利用密码要素增加协议的安全性。大量协议还被应用在关键性系统上，它们的安全分析和证明，乃至如何设计一个安全的协议，仍然是一件很困难的事情。

正如本书作者所述，安全工程师用高强度的密码算法设计出一个协议后，并不能保证通信协议的安全性。传统的安全分析依赖于设计者的经验和手工分析，这种做法在实践中已经被证明很难完全检测出系统的各种隐藏漏洞。协议运行环境的改变、安全假设的改变，都可能导致新的攻击。例如，伴随着移动计算的普及，一个设备上运行着多个不同的应用，每个应用对应着一个协议会话，多协议的并发运行将增加新的安全隐患。

本书涵盖了从安全协议验证的基础理论到代码的实现，不仅为协议构建了牢固的数学形式化基础，进而精确地定义了协议的运行规范和安全属性，还设计了高效的验证算法。作者不仅熟悉各种形式化理论，更难能可贵的是开发了多个开源的安全协议验证系统：基于标准模型的 Scyther、提供机器证明的 Scyther-Proof、高级的可交互式 Tamarin。这些系统非常有利于协议的分析和学习，信息安全工程设计人员可以很方便地用验证系统检测协议的安全性能。

本书有一个配套的网页，提供了协议验证需要的工具、教学资料和参考文献。读者在阅读过程中可以下载相关工具并对协议实例进行验证分析，将有利于理解协议的运行和有关安全机制。验证工具提供了源码，非常方便高级研究者深入了解验证算法的细节。因此，本书非常适合作为信息安全和相关专业的教材，也可作为相关工程研究人员的参考书。

本书的翻译得到了国家自然科学基金(编号：61170282、61661019)和海南省自然科学基金(编号：614231)的资助和支持。特别感谢中国科学院软件研究所首席研究员、北京大学软件与微电子学院卿斯汉教授一直以来给予的关怀和帮助。全书由吴汉炜翻译，卿斯汉审校。由于时间仓促，不当之处在所难免，恳请读者批评指正。联系 E-mail：wuhanwei@hainu.edu.cn。

译 者

于海南大学东坡湖畔

前　言[1]

大约十年前，Sjouke Mauw 和 Erik de Vink 创建了荷兰艾恩德霍芬科技大学的计算机安全研究小组。凭借在形式化理论领域的深厚功底，他们开始关注安全协议的形式化建模与安全属性分析。他们发现，如果首先构建一个简洁的模型，然后用易于理解的操作语义概念描绘该模型，将非常有助于理解安全协议固有的复杂性。在研究开始后不久，Cas Cremers 加入了研究小组，并把这个极具挑战性的领域作为他的博士研究课题。

这些年来，我们不仅在研究中深入探讨协议模型，还把它作为研究生教学中学习安全协议的理论基础。其间我们开发了 Scyther 验证系统。该工具运行效率很高，能帮助我们快速证明协议的正确与否，并将验证理论应用到真实协议检测中。

本书的写作目的有三个。首先，简略介绍我们在安全协议理论研究上的成果。其次，本书可作为协议验证的基础教材。最后，当阐述完理论基础后，读者能将验证理论应用于实际分析。我们希望读者在阅读完本书后，能理解协议分析的内部机理和各种不同验证理论。

本书的出版得益于许多人的支持和帮助，没有他们的付出本书将无法面世。首先我们要特别向 Erik de Vink 表达我们的敬意。书中的许多概念、技术方案均由他提出，通过与他的热烈探讨，我们从中获益匪浅。

我们还要真挚地感谢 Jos Baeten 和 David Basin，他们给了我们在这个领域连续几年研究和著书的机会。同时，还要感谢我们的编辑 Ronan Nugent，在我们的写作过程中他给予了我们耐心的支持。

本书涉及的理论和陈述在这些年来经历了许多变化。我们要特别感谢为本书提出各种建设性意见的人士，他们是：David Basin、Ton van Deursen、Hugo Jonker、Barbara Kordy、Simon Meier、Matthijs Melissen、Marko Horvat、Saša Radomirović、Benedikt Schmidt 和 Christoph Sprenger。

最后，我们各自的家人在我们的写作过程中给我们提供了精神上的支持。没有他们，本书不可能面世。

Cas Cremers 于苏黎世

Sjouke Mauw 于卢森堡

① 中译本的一些图示、参考文献、符号及正斜体形式等沿用了英文原著的表示，特此说明。

目　录

第1章 背景介绍

摘要 在简略地叙述完密码学和安全的历史背景后，我们说明了学习安全协议的动机。本章提供了本书其他章节的内容概要和它们之间的关系。

1.1 历史背景

本书不是一本讲述密码学的专著。

密码学，或者称之为有关“隐写”的艺术，可以回溯到公元前600年，那时希伯来学者已经开始使用密码术。为了给单词girl编码，他们对单词的每个字母单独编码，通过对字母表重新倒序排列：a被换为z，b被换为y，以此类推，这样，单词girl将被编码为trio，反之亦然。只要没有人发现这样的编解码方案，这样的加密就是安全的。

公元前400年左右，据说斯巴达人使用了一种叫作Scytale的设备加密信息，这个设备可以看作世界上第一个用来加密的设备。实际上，Scytale就是一个具有特定直径的圆棍。只有发送者和接收者才知道正确的直径。发送者把长条纸(根据传说，是一条腰带)缠绕在木棍上，然后按照从左到右的顺序，把机密信息写在纸上，如图1.1所示。

如果我们要发送秘密消息consoles，假设现在木棍的直径大小刚好够环绕木棍写下两个字符，我们在木棍的前方从左到右写下c o n s，如图1.1所示。接着，把木棍翻转过去在另外一面写下剩余的字符 o l e s。现在，如果我们把长条纸从木棍解下，阅读所有字符，可以发现加密后的消息是coolness，如图1.2所示。

如果接收者需要对密文解密，可以把密文缠绕在相同粗细的木棍上。如果木棍的直径不对，如环绕木棍一次可以写三个字符，则消息会被解码为clsonsoe，接收者就无法得到正确的明文消息。在这里，木棍的直径扮演了密钥的角色。即使敌人知道加密的具体方案，但加密和解密密文需要一个特定的密钥，这个密钥仅由消息的发送方和接收方共享。

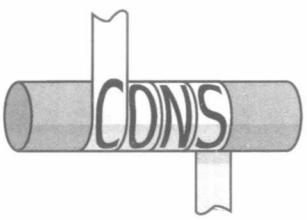


图1.1 Scytale

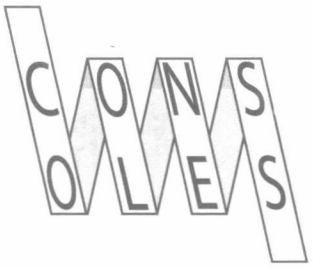


图 1.2 展开后的 Scytale

回顾历史，加解密算法的演变非常重要。古典加解密方案多取决于加解密算法的机密性。

这种设计理论也被称为基于算法机密性的安全设计，即使在今天仍被使用。当然，还可以设计出更高安全强度的加解密方案。19 世纪，著名的密码学家奥古斯特·柯卡霍夫(Auguste Kerckhoffs)指出，系统的安全性不应依赖于加解密算法的机密性，仅取决于密钥的保密性。这就是著名的 Kerckhoffs 原理。20 世纪，克劳德·香农(Claude Shannon)明确地阐述了类似的概念，指出“敌手总能了解系统”，我们称之为香农准则。

在第二次世界大战之前和期间，密码学被广泛地使用，其中也包括德国军队。尽管有许多不同的密码机，但其中最有名的莫过于 Enigma(谜)密码机，这是一种有多个转轮的密码设备。1932 年，波兰研究者最早尝试对 Enigma 密码机密文进行破译。在他们的成果基础上，英国布莱奇利庄园的一个研究组(其中包括著名的计算机科学家 Alan Turing)在战争后期终于能在每日收集的敌人密文基础上破译密文。能取得这样的成果并不是由于他们已经对德国人的 Enigma 密码机构造了如指掌，而是他们设计出了专用的解密机器(花费了几年的时间)，可以从密文信息中恢复出密钥。

1948 年，当香农发表了有关信息论的重要论文后[145]，密码学领域开始出现了很多出版著作，其中大部分集中于如何构建新的密码学方案。这些方案不再仅限于新算法的发明，研究者们还开始用优雅的数学方案构建密码系统：现在密码系统的安全强度取决于某些数学难题。为了证明一个密码方案的安全性，设计者可以证明如果攻击者要破解系统，他必须能解决一个数学难题，而这个数学难题是公认难以求解的。

1976 年，Diffie 和 Hellman 发表了他们的重要论文，介绍了一种非对称密码学的思想[70]。我们可以给出一种非正式的比喻来说明他们的方案：在 Diffie-Hellman 密码系统中，每个人都有自己的一个特定的扣锁，以及打开这个扣锁的钥匙。现在假设 Alice 希望能安全地收到加密消息，只有她才能阅读这些消息。那么，她创建了一个扣锁和一把对应的钥匙。她不会公开她的钥匙，相反，她把自己的钥匙秘密地收藏好，然后把多个扣锁的备份公开地发送给所有人。如果她的兄弟 Bob 想发送一条机密消息给她，可以找来一个盒子，把机密消息放进去，然后用 Alice 公开的扣锁把盒子锁上。现在除 Alice 外，其他人都无法阅读这个消息。这个精妙的方案解决了一直困扰着传统对称密码学的一个难题，即消息的接收者和发送者要有相同的共享密钥。

这个突破性的成果刺激了后续一系列非对称密码系统的研究，其中的很多系统演变为国际标准。今天，非对称密码系统和对称密码系统被广泛地使用，尤其是在因特网上大量使用，如无线通信、智能卡应用、手机通信，以及其他各种应用。

简略地对密码学历史回顾后，也许有人会认为安全的加密算法就是通信安全的“圣杯”了。他们认为，一旦找到某个完美的加密算法，所有的通信就是安全的，然后就可以高枕无忧了。遗憾的是，这不是实情。单纯的密码学不足以保证通信的安全。这就是为什么本书不是有关密码学专著的原因。

1.2 基于黑盒的安全协议分析

想象一下，如果你有一个结实的自行车链锁(见图 1.3)，但是却用了错误的方式来锁住你的自行车，那么小偷仍然能偷走你的自行车。与此类似，计算机系统的安全性取决于安全部件的交互方式。密码学加密系统就好比自行车的链锁，在构造安全的计算机系统时是一个非常有用的机制，但是你仍然可能会错误地使用密码方案。就安全本身而言，密码学基本机制不足以提供完全的保证。

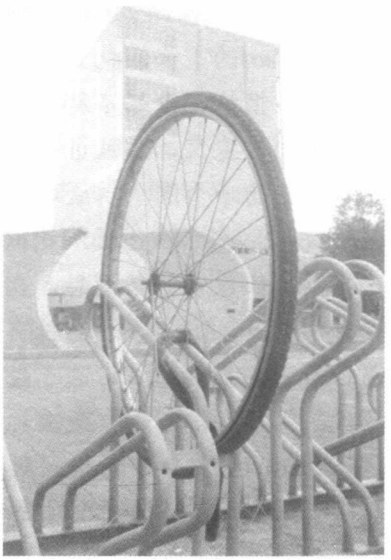


图 1.3 自行车链锁

安全协议是确保通信安全性的手段，这种通信总带有特定的安全目标。一般在安全协议里总会使用某些加密机制。安全协议构成了今天的通信系统的基础，如安全因特网通信，手机系统网络，信用卡、ATM 取款机与银行之间的各种通信。在这些应用中，最关键的环节是保证恶意方无法妨碍协议的预期行为，或者不能获知他未被授权的信息。一个安全协议宣称自己是安全的，这是远远不够的。实际上，我们希望能强有力地保证它的安全性。

为了创建协议的安全保证，我们将寻求数学的支持。我们将创建关于协议和网络的一个数学模型，并假设这个网络处于敌手的完全控制之下。这样的模型允许我们证明敌人无法妨碍协议执行或无法获得任何机密信息。由于这些模型在使用简单密码方案时就已经很复杂了，如果想在它的基础上推理整个协议的安全性，则必须对某些密码细节予以抽象。终于，对协议的安全目标的演绎需要导致了 1983 年 Dolev 和 Yao[76] 对加密过程的理想化抽象，并且包含两个主要特性。首先，密码系统被假设为完美的：一条加密的消息只能被拥有正确密钥的人解密(没有其他方法可以破解该密码方案)。其次，消息被看作抽象的项：要么敌手能得到完整的消息内容(因为他有正确的密钥)，要么他什么也得不到。我们可以对这样的基于抽象黑盒的模型进行分析，并且意识到模型总是把所有加密抽象为具有特定属性的函数。对于加密细节和属性，我们并不具体建模，而是假设某人已经发明了一个完美的密码学方案，可以直接将其用于构建安全协议。

在这两个密码学假设之后，Dolev 和 Yao 就计算机网络建立了第三个抽象概念。整个网络被假设在敌手的完全控制之下。他可以任意删除消息，检查消息的内容，插入他自己的消息，重定向消息或简单地重发消息。这三个属性合称为 Dolev-Yao 模型：加密是完美的、消息是抽象项、网络被敌手完全控制。

给定一个安全协议，我们可以在 Dolev-Yao 的假设模型中，利用数学手段推导出协议的安全属性。最终，Dolev 和 Yao 的工作演化为安全协议研究的一个方向，大体上被称为黑盒安全协议分析。然而，把这三个基本特征用精确的数学模型表示后，即使有清晰的安全假设和清晰的安全属性定义，实践证明这样做仍然是有风险的。

接着，我们给出了这个研究领域的一个例子，展现了协议安全的微妙性。该实例是设计于 1978 年的 NSPK(Needham-Schroeder Public-Key)协议[124]，大约在 Dolev 和 Yao 提出抽象模型 5 年之前。原始协议包含三条消息，在两个通信方之间传送。协议的目的是为所有参与实体提供身份认证。该设计出现后的 20 年，NSPK 协议一直被认为是正确的设计；现在，我们认为它的正确与否取决于使用它的具体环境。在很多强大的分析方法中，这样的微妙攻击并没有被发现，主要原因在于敌手假设被改变了。

1989 年，Burrows、Abadi 和 Needham 发表了一篇突破性的、关于身份认证逻辑(即著名的 BAN 逻辑)推导的论文[39]，该逻辑也依赖于 Dolev-Yao 模型的黑盒假设。按照论文中的逻辑推导，他们成功地证明了多个协议满足身份认证的安全目标。[①] 这些协议里就有 NSPK 协议。该协议已经被正式证明是正确的。该协议在以后演化为 Kerberos[27]协议。大约在 NSPK 协议发布 20 年后，Gavin Lowe 于 1996 年声称在协议中找到一个攻击。结果显示，Lowe 的攻击需要一个强大的敌手，要比 Dolev 和 Yao 模型的原始版本中的敌手更加强大。1980 年前后，网络上的用户总是被看作诚实的使用者：攻击者只能来源于外部。但是在 20 世纪 90 年代，对网络的看法发生了改变：许多大型网络被用户使用，这些用户不一定是可信任的用户。Lowe 的攻击要求敌手

① 该论文的主要贡献是，BAN 逻辑明确了某些 Dolev-Yao 假设，同时对认证的概念给出了一个可接受的数学定义。

是一个内部用户，或者他能拉拢一个内部用户。这样，关于敌手的模型被改变了，现在敌手被假设为他能控制一部分系统中的合法用户。

在同一篇论文中，Lowe 还介绍了在协议中查寻攻击的自动化程序。一个协议的高阶描绘可以用 Casper 程序处理，Casper 程序根据进程代数为协议的行为和敌手可能的操作建立模型。类似地，协议的安全属性被转化为第二个进程集合，这是一个理想化的系统行为模型，只有安全属性被满足时才出现。Casper 系统使用的模型检测工具，其原理基于进程代数理论，可以检查实际的协议模型是否拥有和理想系统一样的行为集合。如果这些行为是相同的，则敌手无法影响协议的正确执行。按照这样的方法，Lowe 能自动化地查找 NSPK 协议里存在的安全攻击，然后使用相同的方法证明在改进后的版本中不存在类似的攻击。修正后的版本就是著名的 NSL(Needham-Schroeder-Lowe)协议。

在 Lowe 的重要成果之后，涌现了大量安全协议形式化理论和工具。许多安全协议形式化理论仅仅专注于协议的描述，却没有可用的工具对这些描述建立形式化语义分析。另外，大多数工具仅有明确的形式化说明，缺少模型的形式化定义和实际要被检测的安全属性的形式化定义。这样，就很难解释分析后的结果。

以上背景导致了本书要阐述的一系列问题，我们将在下一节中说明。

1.3 目的与方法

本书的目的是提供一套理论框架，用于形式化分析和安全抽象协议的验证。特别是，我们致力于提供一套形式化语义和安全属性直观上的形式化定义。当然，还有高效率的工具支持。

首先，我们用操作语义构建了清晰的理论基础，允许我们为黑盒安全协议形式化地建模，还定义了协议所有可能的行为。接着，我们提供了已知的和新的安全属性的形式化定义。根据协议行为的形式化定义和给定的安全属性定义，允许我们校验某个属性是否符合预期目标。利用本书中介绍的一种自动化方法，我们可以检验或反证安全属性是否满足。使用这种基本方法构建的程序工具，我们可以得到安全协议间交互的结果。更进一步地，通过多方协议族的演化例子，我们说明了安全协议形式化规范的应用。

本书的面世得益于附录中列出的有关研究成果[18~20, 53, 55, 56, 58~64, 113]，书中的素材还来源于一系列课程教学资料，这些教学资料曾用于苏黎世联邦理工学院、艾恩德霍芬科技大学和卢森堡大学的有关课程中。

1.4 概要

我们将简要说明本书的组织结构，以及各个章节的内容概要。本书包括 5 个主要

的章节，每章结束后有一些练习。第 2 章给出的简要的常用数学概念符号，将在这 5 个主要章节中使用。

1.4.1 协议分析模型

第 3 章和第 4 章定义了协议分析模型。

在第 3 章中，我们给出了安全协议和它们的行为模型。借助操作语义，我们在模型中明确地制定了协议的执行。结果显示，一个基于角色的安全协议模型是否可判定取决于并发协议的数量。对于协议的安全分析，模型设立了几个清晰的假设，例如，敌手知识如何从协议描述中派生。在协议模型里，安全属性总是被建模为局部安全断言事件。

在第 4 章中，第 3 章中的模型被进一步扩展，增加了几种安全属性的定义，包括机密性和目前已知的几种认证属性。我们描绘了强身份认证的概念，称之为单射同步一致性。然后，我们给出了身份认证属性的层次关系。接着，我们用形式化定义人工地证明了 NSL 协议的安全属性。

1.4.2 模型的应用

第 5 章、第 6 章和第 7 章可以独立阅读。它们建立在第 3 章和第 4 章所定义的模型之上。每章分别突出了模型的不同应用。第 5 章通过提供工具支持强调了模型的算法特征，第 6 章描绘了使用该理论分析现存的协议，第 7 章强调了模型在协议构建中的使用。

第 5 章的标题为“验证”，介绍了一套用于检验安全属性或发现攻击的运算法则，利用该法则还能得到一个协议完整的描绘。这个运算法则在原型工具 Scyther 中实现。该工具的性能代表了目前安全协议分析的业界水准。我们运用该工具分析了大量协议。然后，我们给出了一个语法规范来建立单射同步一致性。

该原型工具还在第 6 章中使用，自动分析多协议的并行执行。这种情况在嵌入式系统中经常发生，如智能卡协议或手机应用。安全实践结果揭露了几种新的攻击，表明即使单独的两个协议是安全的，但是它们的组合未必是安全的。

第 7 章的标题为“基于 NSL 扩展的多方认证”，验证了一个具体的模型应用并描述了对应工具。在该章中，NSL 协议被扩展为身份认证协议族。我们给出了一个证明，说明 NSL 协议的扩展版本能满足既定安全属性。在扩展版本的基础上，可以设计一个高效率的多方同步协议。

在第 8 章，我们回顾了前面的内容，并且对相应工作做了展望。特别地，我们列出了一些参考资料，讨论了协议分析工作的深入内容，并给出了进阶阅读的建议。

第2章 预备知识

摘要 本章叙述了后续章节中将使用的数学概念和符号。

2.1 集合与关系

给定一个集合 T，将 $\mathcal{P}(T)$ 记为集合 T 的幂集，即所有 T 的子集的集合。T^*表示所有 T 的元素构成的有限序列的集合。这样的序列是一个包含 n 个元素的序列，其中 $t_0,t_1,\cdots,t_{(n-1)} \in T$，可以记为 $\left[t_0,t_1,\cdots,t_{(n-1)}\right]$，在不至于引起混淆的情况下，也可以简单地记为 $t_0,t_1,\cdots,t_{(n-1)}$。空序列记为 []。对于序列 t 和 t'，它们的联系关系记为 $t\cdot t'$。$t=t_0,t_1,\cdots,t_{(n-1)}$ 长度为 n，记为 $|t|$。其中，我们把 t_i 表示为序列的第 $(i+1)$ 个元素，这样 t_0 表示序列 t 的第一个元素。记号 $e <_t e'$ 表示 $\exists i,j : i<j \wedge t_i = e \wedge t_j = e'$，即在序列中有前后关系的两个元素。记号 $set(t)$ 表示由序列 t 中的元素生成的集合，也就是说，集合 $set(t)=\{t_i \mid 0 \leqslant i \lhd |t|\}$。沿用上面的记号，元素 $e \in t$ 表示 $e \in set(t)$。

一个元组写为 (x,y)。如果想把元组的两个部件中的某个取出，可以用操作运算子 π 萃取。更严格地说，对于所有的 x 和 y 有 $\pi_1((x,y))=x$ 和 $\pi_2((x,y))=y$。

设 f 为函数，记号 $dom(f)$ 表示函数 f 的定义域，$ran(f)$ 表示函数 f 的取值范围（即取值的上域）。记号 $f:A \to B$ 定义了一个完全函数，对于 A 中的每一个元素，根据映射关系，B 中都有一个对应的元素。$f:A \nrightarrow B$ 表示为一个部分函数，即 A 中的部分元素对应于 B 中的元素。我们说函数 f 是单射的，记为 $injective(f)$，当且仅当对于所有的 $x,x' \in dom(f)$，有 $f(x)=f(x') \Rightarrow x=x'$。

二元关系 R：$T\times T$ 为 $T\times T$ 的一个子集，满足下列特性（对于任何 $x,y,z \in T$）：

自反性（reflexivity）	$R(x,x)$
非自反性（irreflexivity）	$\neg R(x,x)$
对称性（symmetry）	$R(x,y) \Rightarrow R(y,x)$
非对称性（asymmetry）	$R(x,y) \Rightarrow \neg R(y,x)$
传递性（transitivity）	$R(x,y) \wedge R(y,z) \Rightarrow R(x,z)$
多选性（trichotomicity）	either $R(x,y), R(y,x)$, or $x=y$

令 P 为自反性、对称性或传递性的三种特性之一。令 R：$T\times T$ 为一个二元关系。定义 R 的闭包 P-closure 为最小的 R 的超集，且该超集满足 P 特性。定义 R 的传递闭包为 R^+。

一个二元关系<：$T\times T$是一个严格的偏序，仅当它满足非自反性、非对称性和传递性时。一个严格的全序是在一个偏序的基础上增加了多选性。

例2.1 关系(Relations) 令T为集合$\{a,b,c,d\}$，R表示关系$\{(a,b),(b,c),(c,d)\}$，则R满足非自反性和非对称性。R的自反闭包为$\{(a,b),(b,c),(c,d),(a,a),(b,b),(c,c),(d,d)\}$。$R$的传递闭包$R^+$为$\{(a,b),(b,c),(c,d),(a,c),(a,d),(b,d)\}$。传递闭包(具有传递性)$R^+$还满足多选性、非自反性和非对称性，因此，它是一个严格的全序。

2.2 巴科斯范式

由字符串构成的集合可以通过巴科斯(Backus-Naur Form，BNF)范式来定义[43]。一个BNF范式由一组派生规则组成。这样一条派生规则的左边部分称为一个符号，它表示依据派生规则定义的某个集合。用派生规则定义的*setname*的形式为：

$$setname ::= alt_1 \mid alt_2 \mid \cdots \mid alt_n$$

派生规则的右边部分为一系列可选项，每个可选项用垂直线分隔开。可选项为各种生成集合元素的方法，*setname*标识在派生规则的左边。每一个可选项本身可以是一个符号、一个集合或一个字符串，以及它们的组合体。当我们把某一个选项记为[*exp*]时，表示它是可选的，如果记为[*exp*]*则表示有零或零个以上的项目出现。

例2.2 范式(Grammars) 令*FuncName*表示函数名称的集合，*FuncName*= {*f*,*g*,*h*}，令*Const*表示常量集合，*Const* = {*c*,*d*,*e*}。令符号“(”，“)”、“,”和“+”为字符串。下面的BNF范式定义了*Term*，每一项由一些应用函数构成。

$$FuncApp ::= FuncName(Const\ [,\ Const]^*)$$

$$Term ::= FuncApp \mid Term + Term$$

第一条规则定义了一个应用函数，要求该函数有至少一个参数，参数间用逗号分隔。第二条规则说明每一项都来源于应用函数；另外，项由其他项递归定义而来，如右边所示，每个项间用+号分隔。

这样，*Term*的一些元素可以表示为$f(c)$，$f(d,d,d,c)$，$h(c)+f(d)$，以及$g(c,c)+h(d,c)+h(d,c)+f(e,e,c,e)$。下面的字符串则不是项的元素，如：$c$，$c+c$，$f()$，以及$f(c)+(f(d)+f(e))$。

2.3 符号变迁系统

一个符号变迁系统(Labelled Transition System，LTS)是一个四元组$(S,L,\rightarrow,s_0)$，这里：

(i)S是一个状态集合；