安全协议

实施安全性自动化分析与验证

孟 博　王德军　著

科学出版社

安全协议实施安全性自动化分析与验证

孟 博 王德军 著

科学出版社

北京

内 容 简 介

本书系统介绍安全协议实施安全性自动化分析与验证的基本理论和关键技术及最新成果。主要内容包括安全协议实施安全性分析与验证的国内外发展现状、一阶定理证明器 ProVerif 及应用、自动化安全协议证明器 CryptoVerif 及应用、基于计算模型自动化抽取安全协议 Blanchet 演算实施模型、安全协议 Blanchet 演算实施自动化抽取工具 Swift2CV、基于消息构造的安全协议实施安全性分析方法、安全协议实施安全性分析工具 SPISA、面向多个混合安全协议轨迹的安全协议实施安全性分析方法、安全协议实施安全性分析工具 NTISA、典型安全协议实施安全性分析等。

本书可供从事安全协议、密码学、计算机、软件工程、通信、数学等专业的科技人员、硕士和博士研究生参考，也可供高等院校相关专业的师生参考。

图书在版编目（CIP）数据

安全协议实施安全性自动化分析与验证 / 孟博，王德军著. —北京:科学出版社，2019.11

ISBN 978-7-03-062506-9

Ⅰ.① 安… Ⅱ.① 孟… ②王… Ⅲ. ①计算机网络-安全技术-通信协议 Ⅳ.① TP393.08

中国版本图书馆 CIP 数据核字（2019）第 220811 号

责任编辑：闫 陶 / 责任校对：高 嵘
责任印制：徐晓晨 / 封面设计：苏 波

科 学 出 版 社出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

北京凌奇印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

开本：B5（720×1000）
2019 年 11 月第 一 版 印张：16 1/2
2020 年 7 月第二次印刷 字数：325 000

定价：98.00 元

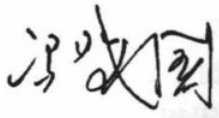
（如有印装质量问题，我社负责调换）

序

网络空间安全已成为国家安全的重要组成部分，解决网络空间安全问题离不开技术的创新和发展，而安全协议是解决网络空间安全问题的重要技术之一，是保障网络空间安全的关键。安全协议的研究包括很多方面的内容，如安全协议设计理论与方法、安全协议安全性分析与验证方法、安全协议实施（安全协议代码）安全性分析与验证方法等。近年来，安全协议实施安全性研究已成为安全协议研究领域的研究热点，其研究对促进安全协议的发展和保障国家网络空间安全具有重要而深远的意义。

本书作者分别以三个假设为前提，即能够获取安全协议客户端实施及安全协议服务器端实施、仅能够获取安全协议客户端实施、不能获取安全协议客户端实施和安全协议服务器端实施，对安全协议实施安全性进行了系统研究，取得了一系列重要成果。特别是在安全协议实施安全性的自动化分析模型、方法及相应的软件工具等方面取得的成果得到了国内外同行的好评。他们发表了一批高质量的学术论文，培养了多名优秀研究生。本书是他们长期研究成果的总结。本书的出版必将为传播安全协议实施安全性分析与验证的基础知识、交流安全协议实施安全性分析与验证的理论和技术、扩大安全协议实施安全性分析与验证系统的应用做出重要贡献。

我为他们取得的研究成果和学术著作的出版感到由衷的高兴，并向他们表示衷心祝贺！



2019 年 11 月 16 日于北京

前　言

安全协议作为网络空间安全的重要组成部分，是保障网络空间安全的关键和灵魂。从安全协议设计、安全协议抽象规范安全性的分析与验证、到安全协议实施（安全协议代码），人们主要集中在对安全协议抽象规范的安全性分析和验证方面，实用性较差。近几年来，人们对安全协议的最终表现形式，安全协议实施越来越感兴趣。因为无论任何安全协议，要想发挥作用，必须进行安全协议实施，故对其安全性进行分析对保障网络空间安全具有重要意义。安全协议实施不仅比安全协议抽象规范复杂，而且在安全协议实施过程中，因程序员的专业素质参差不齐，无法保证不引入逻辑错误或者是代码错误，进而可能会造成安全协议实施与其抽象规范不一致。很多实践表明，即使对形式化方法已证明安全的安全协议，在实施过程中，也可能因为人为的失误而引入了新的安全问题。由此可见，仅在安全协议抽象模型层面对其进行安全性分析研究是远远不够的，必须对安全协议实施的安全性进行研究，以得到具有很强实用性的安全协议实施，保障网络空间安全。

安全协议实施由安全协议客户端实施和安全协议服务器端实施组成。目前，安全协议实施安全性分析研究工作主要基于以下三个前提进行：能够获取安全协议客户端实施及安全协议服务器端实施；仅能够获取安全协议客户端实施；不能获取安全协议客户端实施和安全协议服务器端实施。故本书以此三个前提为基础，系统全面地介绍安全协议实施安全性自动化分析与验证的基本理论和关键技术及最新成果。

本书共分五部分 14 章。第一部分包括第 1 章，介绍安全协议实施安全性分析与验证的国内外发展现状与最新成果。第二部分包括第 2～5 章，重点对安全协议规范形式化分析与验证相关技术和应用做介绍，包含 Applied PI 演算与其 BNF 范式、一阶定理证明器 ProVerif 及应用、概率进程演算 Blanchet 演算与其 BNF 范式、自动化安全协议证明器 CryptoVerif 及应用。第三部分包括第 6～8 章，重点介绍以能够获取安全协议客户端实施及安全协议服务器端实施为前提，分析安全协议实施安全性的相关方法与应用成果，主要包括基于计算模型自动化抽取安全协议 Blanchet 演算实施模型、安全协议抽象规范模型生成工具 Swift2CV、OpenID Connect 协议与 Oauth2.0 协议和 TLS1.2 协议 Swift 实施安全性分析。第四部分包括第 9～11 章，重点介绍以仅能够获取安全协议客户端实施为前提，分析安全协

议实施安全性的相关方法与应用成果，主要包括基于消息构造的安全协议实施安全性分析方法、安全协议实施安全性分析工具 SPISA、RSAAuth 认证系统与腾讯 QQ 邮件认证系统安全性分析。第五部分包括第 12～14 章，重点介绍以不能获取安全协议客户端实施和安全协议服务器端实施，分析安全协议实施安全性的相关方法与应用成果，主要包括面向多个混合安全协议轨迹的安全协议实施安全性分析方法、安全协议实施安全性分析工具 NTISA、统一身份认证平台登录协议 CAS-SSO 和 CAS-OAUTH 实施安全性。

本书由中南民族大学孟博和王德军共同撰写，其中第 1～5 章由王德军撰写，其余章节由孟博撰写。全书最后由孟博统稿、王德军校稿。研究生张金丽、鲁金钿与何旭东对此书亦有贡献。

本书的研究工作得到了湖北省自然科学基金（No.2018ADC150，No.2014CFB249）、中南民族大学中央高校基本科研业务费专项资金（No.CZZ19003, No.CZZ18003）项目资助，同时也得到了科学出版社的大力支持，在此表示衷心的感谢。

由于水平有限，对一些问题的理解和表述或有肤浅之处，诚请读者批评指正。

孟　博　王德军

于 2019 年 6 月 3 日

目　录

第1章　安全协议实施安全性分析与验证现状

1.1　引　　言

安全协议是网络空间安全的核心。安全协议实施[1-4]是安全协议的最终表现形式，其安全性分析也越来越受到人们的关注。从安全协议设计，到安全协议抽象规范安全性的分析与验证，再到安全协议实施（安全协议代码）[5]，人们主要关注安全协议抽象规范的安全性分析和验证[5-7]。近几年来，人们对安全协议的最终表现形式，安全协议实施越来越感兴趣。因为无论任何安全协议，要想发挥作用，必须进行安全协议实施，所以对其安全性进行分析，对保障网络空间安全而言具有重要意义。安全协议实施不仅比其抽象规范复杂，而且在安全协议实施过程中，因程序员的专业素质参差不齐，无法避免逻辑错误或代码错误，进而可能会造成安全协议实施与其抽象规范不一致。此外，即使对采用形式化方法[8, 9]已证明安全的安全协议，在实施过程中，也可能因为人为的失误而引入新的问题，变得不再安全。由此可见，仅在安全协议抽象模型层面对其进行安全性分析研究是远远不够的，必须对安全协议实施的安全性进行研究，以得到具有很强实用性的安全协议实施[1, 2, 10]，保障网络空间安全。

1.2　能够获取安全协议客户端实施和安全协议服务器端实施

基于能够获取安全协议客户端实施及安全协议服务器端实施的假设，主要研究方法分为两种：程序验证、模型抽取。

1.2.1　程序验证

程序验证方法对已有安全协议实施的安全性进行分析，主要分别基于逻辑、

类型系统、类型系统与逻辑证明，对其安全性进行分析。这种方法既不能证明分析过程的正确性，又过于依赖在安全协议实施中添加大量的代码注释与断言。

基于符号模型，2005 年，Goubault-Larreecq 等[11]通过求解子集验证安全协议 C 语言实施安全漏洞。首先，使用指针分析技术分别对攻击者收集和伪造的消息进行分析，进而建立运行时数据与代表性消息的抽象逻辑间的关联关系。然后，通过控制流图，利用信任断言得到 Horn 语句表示的抽象语义，最后用 H1 求解器验证其保密性。2009 年，Rjens[12]以安全协议的 Java 实施为研究对象，提出一个建立安全协议软件实施和安全协议规范模型之间映射关系的方法，此方法首先基于控制流图，通过对开源的安全套接层（secure sockets layer，SSL）协议 Java 实施进行分析，进而得到安全协议 Java 实施的抽象模型，最后应用定理证明器高阶逻辑（higher-order logic，HOL）分析其保密性。2009 年，Chaki 等[13]提出基于特定域协议和符号攻击模型，对迭代抽象改进的软件模型检测方法进行拓展，将拓展后的软件模型检测与标准协议相结合，分析安全协议 C 语言实施的认证性和保密性的方法，基于此方法，开发了 ASPIER 工具，通过实验并结合控制流图，ASPIER 应用断言抽象技术验证了开放式安全套接层（open secure sockets layer，OpenSSL）协议 C 语言实施的认证性和保密性。2010 年，Bhargavan 等[14]首先用 F#语言开发了一个密码原语库，进而提出一个验证安全协议 F#实施安全性的方法，并使用 F#语言实现，最后用 F7 语言证明这种方法的正确性。在这种方法中使用公式记录安全协议 F#实施的不变量。基于此方法，应用 ProVerif 可以对典型密码原语和安全协议 F#实施的安全性进行分析和验证。

基于计算模型，2010 年，Backes 等[15]使用 F#语言的安全类型检查器和 λ 演算来验证安全协议 F#实施的安全性，并用通用计算合理性证明框架（a general framework for computational soundness proofs，CoSP），证明了 F#语言类型检查器的计算合理性，这种方法使用理想的符号模型和精化的并发不动点（refined concurrent fixpoin，RCF）演算对安全协议 F#实施进行建模，并结合 F7 语言，实现安全协议 F#语言实施的自动化验证。2011 年，Bengtson 等[16]应用 F#实现了一个检测安全协议实施安全属性的类型检测器，进而使用具有精确类型的 λ 演算来建模一阶逻辑中的前件后件，从而得到密码学原语的形式化模型，生成可满足性模理论（statifiablity modulo theories，SMT）求解器的验证条件，验证安全协议 F#实施的认证性和保密性。同年，Dupressoir 等[17]在无限会话条件下，首先应用初始状态（ghost state）检测当两个不同条件的数组映射到同一个数组上时的碰撞；其次基于符号项的约束对密码学应用程序接口（application programming interface，

API）进行建模；随后对 C 程序中的攻击者进行建模；进而把强类型函数式编程语言 F#和 F7 中的密码结构的不变量重新映射到弱类型的低级命令式语言中。2011 年，Aizatulini 等[18]基于计算模型验证了安全协议 C 语言实施的安全属性，首先符号化执行应用程序，从安全协议 C 语言实施中抽取进程计算模型，然后把抽取的进程计算模型转化成基于 CryptoVerif 的安全协议模型，从而验证了安全协议 C 语言实施的安全性。2011 年，Swamy 等[19]首先提出 F*语言。它是一种新的用于安全分布式编程的类型语言，F*语言包含 F7 语言和 Fine 语言。它能提供任意递归，同时可以保持逻辑的一致性，使用加密凭证和逻辑证明支持精化属性（refinement properties，RP）的证明。然后实现了一个基于 Fine（F*）语言的编译器原型，该编译器把 Fine（F*）语言编译成.NET 字节码，同时产生具有开销较小的可验证二进制字段，实现高效的字节码验证，这种方法分析并验证了安全协议 F*实施的认证性与保密性。2015 年，Swamy 等[20]对 F*语言进行扩展，并结合类型推测、SMT 求解实现扩展后的 F*语言核心功能的半自动化验证，并开发出 F*系统，此外还实现了基于 F*语言证明纯代码片段（pure fragment，PF）的可靠性，并对新的类型选择器的选择进行了分析。F*系统可以产生 F#代码和 OCaml 代码。2016 年，Swamy 等[21]又对 F*语言在高阶逻辑和按值调用（Call- by-value，C-b-v）进行了拓展，能通过添加更少的注释，验证安全协议 F*实施的更多安全属性。

1.2.2　模型抽取

模型抽取即从安全协议实施中抽取安全协议抽象规范，并且证明抽取方法的正确性，然后用协议抽象规范安全性分析工具来分析其安全性。

安全协议源语言（如 Java 语言、C 语言、F 类语言等）实施 SP[S]转换为安全协议目标语言（如 Blanchet 演算、应用 PI 演算、LySa 演算等）实施 SP[T]；然后利用自动化安全协议分析或验证工具分析安全协议目标语言实施 SP[T]，即源语言是编程语言，而目标语言是形式化语言，两者之间是互模拟的关系。如果安全协议目标语言实施 SP[T]对于根据任意攻击者 Adv[S]构造的任意攻击者 Adv[T]是安全的，那么，安全协议源语言实施 SP[S]对于任意攻击者 Adv[S]也是安全的。安全协议实施抽取模型如图 1.1 所示。

这种方法被认为非常有效和合理，适用于分析协议实施这种较小规模的代码。在程序验证部分中，也有部分研究工作用到模型抽取的方法。其主要研究见表 1.1。

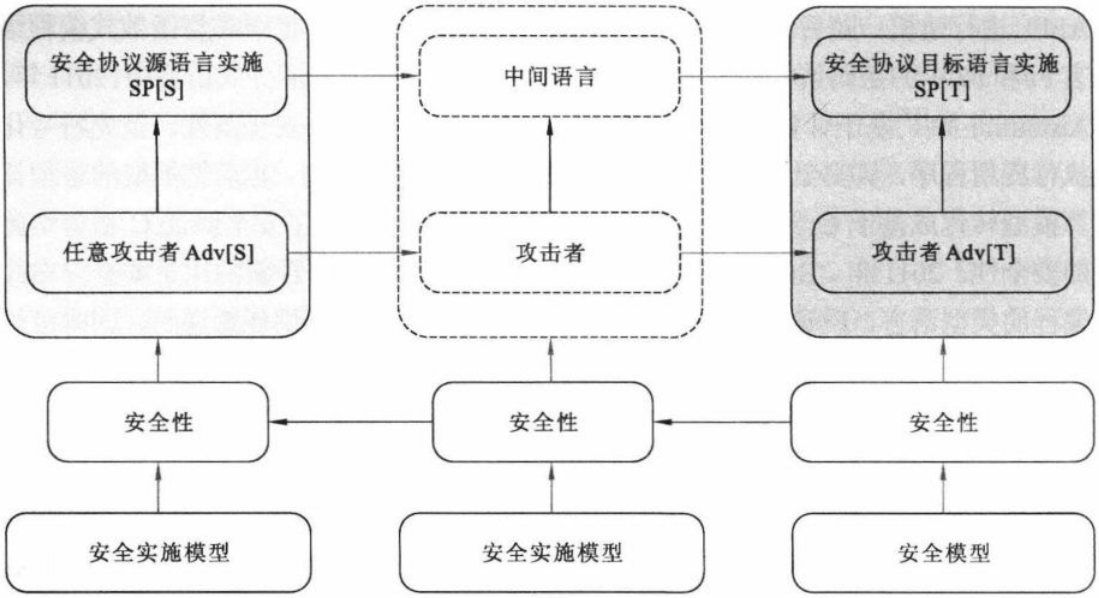


图 1.1　安全协议实施抽取模型

表 1.1　模型抽取方法

文献	程序语言	形式化语言	形式化模型	安全属性	正确性	开发的软件工具
[11]	C	Horn 子句	符号模型	保密性	证明	
[12]	Java	一般逻辑	符号模型	FOL 属性	未证明	
[13]	C	ASPIER 协议语言	符号模型	保密性、认证性	证明	ASPIER
[15]	F#	RCF	计算模型、符号模型	迹属性	证明	H1 求解器
[22]	F#	Applied PI 演算	符号模型	保密性、认证性	证明	FS2PV
[23]	Java	LySa 演算	符号模型	保密性	未证明	Elygah
[24]	C	Applied PI 演算	符号模型	保密性、认证性	证明	
[25-26]	F#	Blanchet 演算	计算模型	保密性、认证性	未证明	FS2CV
[27]	C	Blanchet 演算	计算模型	保密性、认证性	证明	
[28]	Java	Blanchet 演算	计算模型	认证性	未证明	SubJAVA2CV
[29]	Javascript	应用 PI 演算、Blanchet 演算	符号模型 计算模型	认证性、保密性	未证明	RefTLS
[30]	Swift	Blanchet 演算	计算模型	认证性、保密性	未证明	SubSwift2CV

基于符号模型，2008 年，Bhargavan 等[22]提出一个验证安全协议 F#实施安全性的架构，支持密码原语的具体实施和符号化实施，其中具体实施用于互操作性测试，符号化实施用于调试和形式化验证。该方法运用 FS2PV 工具把安全协议 F#实施转换为 ProVerif 的输入语言：Applied PI 演算，进而用 ProVerif 工具验证安

全协议 F#实施的保密性和认证性。2008 年，Nicholas[23]开发了 Elygah 系统，该系统首先把安全协议的 Java 实施转化成 Lysa 演算进程，然后得到安全协议 Java 实施的形式化模型，进而分析其认证性。但是这种方法没有证明抽取方法的正确性。2012 年，Aizatulin 等[24]首先应用符号执行技术，符号化执行安全协议 C 语言实施，获得安全协议 C 语言实施发送/接收消息的符号化描述，然后得到其形式化抽象模型，随后使用 ProVerif 验证其保密性和认证性。

基于计算模型，2008 年，Bhargavan 等[25-26]开发了一个标记语言（markup language，ML）子集到 CryptoVerif 的编译器原型 FS2CV，该编译器能够把安全协议 F#实施中的密码原语、数据库、信道语句转换成 CryptoVerif 形式化模型，进而用 CryptoVerif 验证其 F#实施的保密性和认证性，并且采用手工方式证明抽取方法的正确性。此方法对 TLS 协议（transport layer security，TLS）协议 F#实施进行分析，证明了其认证性和保密性。2012 年，Aizatulin 等[27] 从安全协议 C 语言实施中抽取其形式化模型，然后用 CryptoVerif 验证其认证性与保密性：首先通过符号化执行从安全协议的 C 语言实施中抽取安全协议 C 语言实施抽象模型；然后把抽取的模型转换成 CryptoVerif 语法描述，使用 CryptoVerif 验证安全协议 C 语言实施的认证性与保密性。但是目前只能支持顺序执行协议，不支持分支语句。2015 年，Li 等[28]首先定义 SubJAVA 与 Blanchet 演算间的语法映射关系，然后基于模型抽取技术，开发模型抽取工具 SubJAVA2CV，该工具对安全协议的 Java 实施首先进行词法分析、语法分析和生成抽象语法树，之后化简抽象语法树，抽取出安全模型，将其转换为 Blanchet 演算的抽象语法树，生成 Blanchet 演算代码，最后使用 SubJAVA2CV 抽取一个认证协议 Java 实施的安全模型，将其转换为 Blanchet 演算代码，应用自动化分析工具 CryptoVerif 分析安全属性，证明协议实施的认证性。2017 年，Bhargavan 等[29]首先分别基于符号模型和计算模型，应用 ProVerif 和 CryptoVerif 分析了 TLS1.3 安全协议抽象规范的安全性，然后给出了 TLS1.0-1.3 安全协议实施-RefTLS。2018 年，孟博等[30]首先对已有的安全协议 Swift 语言实施进行分析，进而确定与安全协议 Swift 实施紧密相关的 Swift 语言子集 SubSwift，然后根据操作语义，建立从 SubSwift 语言到 Blanchet 演算的映射模型，提出从安全协议 SubSwift 语言实施中抽取安全协议 Blanchet 演算实施的方法，并开发安全协议 Blanchet 演算实施生成工具 SubSwift2CV，同时对 OpenID Connect 协议、Oauth 2.0 协议及 TLS 协议的安全性进行分析，结果表明 OpenID Connect 协议、Oauth 2.0 协议和 TLS 协议的 SubSwift 语言实施与安全协议 Blanchet 演算实施的安全性分析结果分别是“客户端能够认证 OpenID 供应商”和“客户端无法认证授权服务器”。在 SubSwift 客户端与服务器通信过程中能够保证预置密钥保密性，且客户端能认证服务器端。

1.3 仅能够获取安全协议客户端实施

基于仅能够获取安全协议客户端实施的假设，分析安全协议实施安全性。其主要思路首先获取包含安全协议客户端实施的网络安全应用程序，获取安全协议客户端实施，进而分析其安全性，或根据提取的安全协议实施抽取、还原出安全协议抽象规范，使用安全协议分析工具分析其安全性。其依据是安全协议客户端实施包含安全协议客户端的具体功能，如消息参数的产生、消息的加密、解密处理及对安全协议服务器端响应消息的解析等，通过对其进行分析，得到安全协议服务器端的模型，分析安全协议实施的安全性。

1.3.1 网络轨迹

网络轨迹方法属于协议逆向技术中的重要分支。协议逆向技术不依赖于任何协议描述的情况，通过对协议实体的网络输入、输出，系统行为和指令执行对流程进行监控和分析，提取协议文法、语法和语义[31]。网络轨迹方法主要在能获得网络负载的情形下，对网络负载进行字段提取、语义分析及状态机提取，进而分析安全协议实施安全性。其依据为：每个抓取到的报文样本都是协议实体规范的一个具体实例，所以相同类型的不同样本之间具有一定相似性[32]，通过对得到的报文进行聚类和统计分析来确定消息边界和消息格式，使用关键字提取或者是统计分析提取消息字段、推测协议语法语义及协议状态机等信息，由此可分析安全协议实施安全性。主要分析方法是：①利用第三方软件或者工具，通过守候捕获和隐蔽截获的方法截取安全协议网络数据包；②对截获的安全协议网络数据包应用聚类、数据挖掘等技术进行分析、处理，获取相应安全协议的规范、协议字段格式、语法和语义及协议状态机等信息；③对安全协议的实施安全性进行分析，其主要原理如图 1.2 所示。

根据图 1.2，首先在安全协议通信实体进行通信时，使用第三方网络数据包截获工具的方式守候截获或隐蔽嗅探的方式获取协议通信 N 网络轨迹或网络负载；再对获取的网络数据报文进行分析、处理，根据分析结果提取安全协议的消息字段、语义语法及协议状态机等信息，由此得到安全协议实施安全性分析结果。

2013 年，Bai 等[33] 基于安全协议实施密码学算法正确、加密/解密密钥安全及网络 DNS 基础架构安全这三个假设，开发了 AuthScan 系统。其主要思路：首先初始化安全协议客户端 JavaScript 实施，创建测试用例，该测试用例包含 HTTP 数据轨迹和分析器提供的初始化数据，进而推断安全协议的抽象规范初始化模型。