

安全协议的形式化分析技术与方法

薛 锐 冯登国

(中国科学院软件研究所信息安全国家重点实验室 北京 100080)

摘 要 对于安全协议的形式化分析方法从技术特点上做了分类和分析. 对于安全协议分析技术的发展历史、目前的状况以及将来的趋势作了总体的介绍和总结. 根据作者的体会, 从纵向和横向两个角度进行了总结. 纵向方面主要是从用于分析安全协议的形式化方法的出现和发展的历史角度加以总结. 横向方面主要从所应用的技术手段、技术特点入手, 进行总结分析. 说明了目前协议形式化分析发展的主要方向. 对于目前国际流行的方法和模型进行了例解.

关键词 安全协议; 形式化分析; 安全目标; Dolev-Yao 模型; 密码学可靠性

中图法分类号 TP309

The Approaches and Technologies for Formal Verification of Security Protocols

XUE Rui FENG Deng-Guo

(State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100080)

Abstract The formal methods for verifying security protocols are classified and analyzed from technological points of view. Authors survey the developing history of analyzing approaches and techniques on security protocols; describe the current status, as well as point out the tendency of them. This work comes from authors' personal research interesting, and it processes in two different lines: The first line follows the trace of emergence and developments of formal methods in verification of security protocols. The other line is to analyze the features when concrete systems are used during verification. The most popular methods and systems are briefly introduced by examples.

Keywords secure protocol; formal analysis; security goal; Dolev-Yao model; cryptographic reliability

1 安全协议形式化分析的发展历程

安全协议是通过一系列步骤定义的分布式算法. 这些步骤确切规范了两方或多方主体为达到某个安全目标要采取的动作^[1].

安全协议的分析与普通协议的分析一个不同之处在于: 普通的通信协议并不假设通信系统之外有别的主体. 因此, 分析普通的通信协议主要着眼于协

议的活性、保险性以及公平性, 这里的公平性并非电子商务协议中要求的公平性, 而是协议激活的公平性. 而安全协议则要针对一个额外的敌手, 做出尽可能的防范.

安全协议是在不可靠或者敌意的通信环境下, 保障一定安全特性的(网络)协议. 在这种环境中, 敌手对于协议可以有各种各样的攻击手段, 包括窃听、篡改、截断协议的通信, 甚至加入任何可能的消息, 把协议的消息转向到其它接收者等等. 因此, 对于协

议的攻击隐晦而多样. 这些因素决定了协议安全性分析的复杂性.

在形式化方法出现前, 几乎没有一般可以遵循的安全协议分析的过程. 即使在形式化方法出现之后, 许多协议也是依然通过观察的方式, 直观地说明它们的安全性. 所谓形式化方法是指用数学方法描述和推理基于计算机的系统, 直观的说, 就是规范语言+形式推理, 在技术上通过精确的数学手段和强大的分析工具得到支持. 其表现形式通常有逻辑、离散数学、状态机等等. 规范语言包括语法、语义以及满足关系等几部分. 规范语言可以分为四类: 抽象模型规范法、代数规范法、状态迁移规范法和公理规范法. 后面我们将看到, 几乎所有类型的方法都被用来规范(描述)以及分析安全协议.

1.1 安全协议形式化方法的发展

一般公认, Needham 和 Schroeder^[2] 首次提出了使用形式化的方法对协议进行分析. 而第一个用形式化方法对协议进行分析的文章, 当属 Dolev 和 Yao 在 1981 年的结果^[3]. 他们用算法的手段分析了两类特殊的协议的安全性. 随后的 Dolev, Even 和 Karp^[4] 的分析方法与之一脉相承. 都是用多项式时间的算法对于某些特定类型协议的安全性进行判定. 不幸的是, 这些方法只能适用于特殊的协议. 原因是, 对于所验证的协议稍加拓宽, 所得到的协议就是不可判定的^[5]. 一般说来, 协议的安全性问题是不可判定的. 这就意味着, 没有一个通用的算法, 判定所有协议的安全性. 这一点我们在后面还有涉及.

Dolev-Yao 结果的意义不仅仅是开启了用形式化方法分析安全协议的先河, 而且为随后进行的工作奠定了基础. 他们定义了协议并发运行环境的形式模型、密码算法无关的验证以及敌手的攻击行为等等, 形成了后来者尊崇的 Dolev-Yao 模型(详见第 1.4.1 节).

人们几乎是同时开始了一般安全协议分析工具的开发, 以便在 Dolev-Yao 模型的基础上对安全协议进行自动分析. 最早用于安全分析的是系统 Interrogator. 它由 Millen 开发的^[6]. 作为一个软件工具, 它的原理是企图通过遍历整个状态空间, 找到协议安全的漏洞. 基本思想被后来的许多系统所继承.

Kemmerer^[7,8] 通过传统形式规范语言描述安全协议. 与 Millen 类似的是, 他将协议规范为状态机, 但是他的系统中附着了一个证明器, 可以把一些性质描述为系统的不变量, 证明他们是否被系统保持不变.

在安全协议的分析领域, 如果把 Dolev-Yao 的工作视为形式化方法的标志性工作的话, Burrows, Abadi 和 Needham 的工作^[9] 就应该是形式化方法的代表作和里程碑. 他们利用知识和信念逻辑, 描述和推理认证协议. 这个逻辑系统被称为 BAN 逻辑(取作者们名字的第一个字母). 这是一个全新的方法, 用一集公式表示协议主体的信念, 或者知识, 用一集推理规则从原有公式得到新的信念公式. 这个逻辑可以指出一些协议的漏洞^[9], 因此, 它的出现大大激发了应用形式化的手段对于安全协议方向的兴趣. 沿着这个方法, 许多逻辑被构造了出来, 其中大多是 BAN 逻辑的变种, 目的是弥补 BAN 逻辑的不足. 如 GNY 逻辑、AT 逻辑以及 SVO 逻辑等等. 大多是基于模态逻辑. 模态逻辑有一个特点, 就是在多数情况下是可判定的^[10]. 然而, 逻辑推理的方法有着自己的不足. 主要表现为逻辑的抽象性较高, 这种抽象性往往会掩盖(或丢掉)协议执行的状态信息, 因而难以完全反映协议运行的全貌, 有着自己的局限性.

BAN 逻辑之后, 许多形式化方法逐渐被应用于对安全协议的分析上. 这有两方面的原因: (1) 受 BAN 逻辑的影响和启发, 安全协议的研究者寻找安全协议适当的验证工具, 包括量身定制的工具和对已有方法的改制; (2) 许多成熟的形式化方法寻找有意义的应用领域, 包括 CSP 等等. 因此, 上个世纪 90 年代起, 安全协议的形式化分析研究出现了空前的繁荣景象. 其中值得指出的是 Lowe 的工作^[11]. 在这个工作中, Lowe 成功地应用模型检测的工具 FDR 发现 Needham-Schroeder 协议的一个漏洞. 这个结果对于应用模型检测的方法分析安全协议有着典型意义. FDR 是应用进程代数-通信序列进程(CSP)原理开发的模型检测工具, 其目的是检测一个通信系统中进程并发运行的性质. 对于安全协议的成功应用, 激发了模型检测技术领域的热情. 不仅如此, 这个漏洞的发现使得人们更加清醒地认识到安全协议分析具有的复杂性, 因而系统化的发展协议分析方法显得愈发重要.

在这两类方法之后, 较为新颖的方法就是类型检测(type checking)方法^[12,13]. 这个方法利用 Spi-演算系统^[14], 给每个通道和消息赋以类型, 将协议的安全性目标的达到归结为协议执行过程中类型的保持. 类型的破坏将导致安全性的丧失.

还有一类基于定理证明的方法: 这种方法的主要目的是证明协议的正确性. 对于不正确的协议往往缺乏提供有用参考的能力. 最具代表性的有

Paulson 的归纳证明法^[15]和 Thayer Fábrega, Herzog 和 Guttman 等人提出的串空间模型(strand space). 定理证明的方法可以通过定理证明器协助完成证明过程. 串空间是用图的形式表达协议的执行过程. 协议的一个丛就是协议的一个并发运行. 协议的安全性通过所有丛保持的性质来刻画. 利用串空间原理, Song 发展了一种自动检测工具 Athena^[16].

1.2 基于计算复杂性的形式化方法

在基于 Dolev-Yao 模型的安全协议形式化分析方法蓬勃发展时期, 另外一种基于计算复杂性的安全证明的方法也悄然兴起. 先驱者是 Bellare 和 Rogaway^[17]. 主要思想与密码算法的可证安全的方法类似: 将一个协议 S 的安全性有效地归约到另一个问题 P 的安全性上. 使得, 如果 S 的安全性被(有效)破坏, 那么就可以有办法破坏 P 的安全性(这里的有效性, 通常是指用多项式时间的算法能够做到). 如果 P 的安全性难以攻破, 那么就可以保证 S 的安全性没有问题. 通常把这种方法称为可证明安全. 也是目前大多数人认为的最为可靠的安全性保障手段(尽管有人对这一点颇具微词^[18]).

Shoup 发展了模拟模型(simulation model). 这里模拟的概念来自于零知识证明的密码学成果. 这个模型与 Bellare-Rogaway 模型类似, 并且更为抽象. 主要的新颖之处在于, Shoup 使用了两个系统: 理想(ideal)系统和实际(real)系统. 主要安全保障在于: 敌手在同实际系统交互时能够得到的结果与敌手同理想系统交互所得到的结果一样多. 这样, 从理想模型的(显然的)安全性就可以保证实际系统的安全性.

上述两种方法的一个缺点是无法重用. 对于每个新的协议都必须从头至尾地重新证明. Bellare 等人 1998 年^[19]利用模拟的思想又设计了模式(modular)方法. 2001 年 Canetti 和 Krawczyk^[20]进一步用不可区分的思想改进了这个模型. 从而产生了一系列的结果, 在当前安全协议分析中有着重要的影响.

1.3 形式化方法的进一步发展

前面所述的两个不同研究团体使用了根本不同的方法. 普遍的认识是: 基于计算复杂性的方法是(密码学)可靠的. 而(大多数)建立于 Dolev-Yao 模型之上的形式化方法没有真正建立起密码学的可靠性. 因此, 将这两种方法统一到一个框架之中, 建立形式化方法的可靠性弥补其不足之处就成为目前倍受关注的研究内容. 近年来, 人们的注意力集中到扩充 Dolev-Yao 模型上以及证明形式化方法的密码

学可靠性的研究上.

1.3.1 形式化方法的密码学可靠性

Abadi 和 Rogaway^[21]定义了加密表达式的简单语言, 证明了如果两个表达式在逻辑公式下等价, 则它们在计算的解释下, 根据计算不可区分的标准概念是等价的. Micciancio 和 Warinschi^[22]进一步证明了, 如果使用充分强的加密方案, 任何两个表达式计算等价当且仅当它们可以在逻辑下等价. Gligor 和 Horvitz^[23]精确刻画了这种等价成立对于加密方案的要求. Micciancio 和 Warinschi^[24]进一步给出具有主动敌手的安全协议安全性证明的方法. 是第一个将具有主动敌手的、简洁的逻辑转换成标准计算情形下的结果.

值得注意的是 Impagliazzo 和 Kapron^[25]的结果. 他们给出了两个逻辑系统推理关于标准密码安全定义可靠的密码构造. 文章的一个新颖之处是, 某些证明利用了模型论中非标准分析手段.

Backes, Pfizmann 和 Waidner^[26]等人给出了一个密码学可靠的所谓的“密码库”(crypto-library): 证明了由 Dolev-Yao 模型的要素构成的抽象库, 在任意主动敌手存在的情况下, 任意协议环境下对于任意安全性质都可以密码实现. 这个库包括公钥加密、数字签名、时鲜值、列表操作和应用数据. 主要来源于任意密码安全的公钥加密, 签名系统通过附加的诸如标签和随机化的操作增强其功能. 主体的行为环境是交互式的, 其中建立了主体的模型、交互模型以及敌手模型. 安全性的刻画也是利用模拟的思想, 计算不可区分的机制实现.

1.3.2 形式化方法的密码学扩充

对于多数从事形式化方法安全协议分析的研究者来说, 克服密码学可靠性这个障碍的手段自然是扩充 Dolev-Yao 模型, 在原有的系统中增加密码学要素, 以增加系统表达能力(比如串空间模型中的研究^[27]).

Mitchell^[28]等人则将概率函数直接引入系统之中. 他们基于通信系统演算(CCS)语言, 将进程中不确定性替换为随机性, 从而能够将所谓的渐近协议等价刻画为观察等价(observational equivalence). 而观察等价在 CCS 中是一类标准的等价, 可以通过互模拟等价来界定. 他们通过定义概率互模拟的形式建立了基于观察等价的等式证明系统的可靠性.

1.4 敌手的模型

正如我们前面所述, 参加协议运行的主体可以分为两类: 一类是诚实主体. 在协议运行中按照协议

的规范进行,如果收到不合规范的消息,则认为与自己无关.另一类主体就是所谓的敌手,或者称为入侵者.对于多数协议来说,可以认为只有一个敌手.他的能力的强弱,直接反映到我们使用的敌手模型之中.参照文献[29],我们将敌手模型分为3种.

1.4.1 Dolev-Yao 模型

最为常见的、形式化方法中最流行的敌手模型是 Dolev-Yao 模型^[3].在协议运行的开始或者运行过程中,敌手可以窃听、消去以及任意安排公开通信通道上的消息.他也可以从观察到的消息产生新的消息,并将他们加入到信道之中.敌手能够将非加密消息分成若干个新的消息或者将若干个已知的消息合并生成一个新的消息.敌手可以用自己已知的密钥对任意的消息加密.敌手还可以解密一个收到的密文,前提是敌手知道相应的密钥.敌手可以根据需要,截断信道上传输的任何消息,注入自己产生的新消息,发送该消息至目标主体或者任何其它主体.

形式化分析通常的做法是:把协议视为状态迁移系统,分析所有可能的轨迹(trace);判定安全性质是否在所有的轨迹上被保持.在众多的以轨迹为主的模型中,应用最广的有 CSP^[11,30~33]、高阶逻辑^[15]、多集重写系统^[34,35]以及串空间^[36].

1.4.2 互模拟等价模型

完善保密下的互模拟等价

Spi-演算^[14]是 Pi-演算^[37]关于密码操作的一个扩充.作为描述通信进程并发过程的语言、Pi-演算的扩充,Spi-演算系统可以将协议的诚实主体描述为一个进程.这个进程可以重复执行任意多个,也就意味着可以有任意多个会话并发运行.敌手可以观察和参加任意的通信.这个模型也依赖于完善密码系统假设.协议的安全属性被描述为两个系统的观察等价.一个系统是任意一个进程 A 与实际协议的交互,另一个是 A 与理想进程的交互.如果两个进程在进程演算的意义下是观察等价的,则这个协议就是安全的.利用这个模型可以分析比 Dolev-Yao 模型更为广泛的安全性.

概率意义下的互模拟等价

最近的发展显示,开发基于密码学的更为实际的形式模型的分析技术是目前研究的重点.其目标就是用概率模型替代对密码学的“黑盒子”式的抽象.这类模型的代表有:概率多项式时间的进程演算^[38,39]以及基于密码学传统的模拟模型^[40~42].然而,迄今为止,这类的方法还没有实现自动化分析过程.

2 安全协议形式化分析技术分类

在上一节的历史回顾中,我们已经能够大致地体会到形式化方法的发展脉络.本节具体阐述形式化方法的技术类型和技术特点.我们首先根据技术特点把形式化方法加以分类,而后简单阐述相应类型的技术概要.

形式化方法在计算机科学中的兴起、应用与发展由来已久.形式化方法以其技术特点来分有两大类:(1)定理证明;(2)模型检测.这两类方法都被应用于安全协议的分析.原有的模型通过改造,也就形成了相应的安全协议分析方法.不仅如此,人们认识到安全协议本身的特点,新设计了一些方法专门用于分析安全协议,还延伸出一类混合技术,即定理证明与模型检测相结合的方法.

根据敌手的模型类型,我们将分析安全协议的形式化方法分为两类:(1)Dolev-Yao 模型下的形式化方法;(2)互模拟等价模型.

在 Dolev-Yao 方法下,我们又根据不同方法的技术特点,把它们划分为定理证明方法、模型检测方法以及混合方法.概率模型下又有概率互模拟方法与黑盒子互模拟方法.

为了理解每一类型的技术特点,我们在每个类型之中选择一种影响较大的方法进行技术分析.为此,我们罗列出它们的典型代表(包括计算意义下的分析方法).

(1)Dolev-Yao 模型.

定理证明

逻辑系统: BAN 逻辑^[9].

其它: Paulson^[15], 串空间^[36].

模型检测: CSP 方法^[11].

混合系统: NRL 分析器^[43].

(2)互模拟等价模型——黑盒子互模拟等价模型; Spi-演算.

(3)计算(密码学方法)方法.

概率互模拟模型: BPW 方法^[26].

LMMS 概率进程演算方法^[38].

随机问答器(random Oracle)方法.

Canetti 的模式(modular)方法.

可复合性^[44]可证明安全的方法.

我们将对前两项中的典型模型进行技术特点的分析,希望能够通过分析,掌握基本的分析思路 and 手段,为进一步的研究工作提供必要的基础.在进行具

体技术分析之前,我们先对定理证明、模型检测以及互模拟的基本特征加以介绍.

2.1 定理证明方法

定理证明方法简单说就是数学方法.这种方法考虑协议的所有行为,并且验证这些行为满足一集正确条件.一般可以用这种方法证明协议的正确性,难以用于发现协议的缺陷.而且,基于定理证明的方法在自动化方面无法与模型检测方法比拟.

定理证明有以下特点:

(1)用一集代数或者逻辑公式定义系统的行为,构成系统的行为集.用一集公理和系统的行为集作为推理的基础公式集.

(2)所期望的系统行为和性质被描述成为一组公式,称为定理.

(3)从基础公式集出发,进行定理证明过程,以达到所期望的结果.

定理证明的过程中有些部分是可以自动化的.这样的自动证明系统称为定理证明器.定理证明器与模型检测系统不同,通常需要人的帮助.常见的定理证明器有: Isabelle; HOL; Paradox; ACL2; PVS 等等.

定理证明由公理、假设以及推理规则组成.这系统通常有两个刻画.一个是可靠性或者相容性(soundness);另一个是完备性(completeness).可靠性是指系统证明出的每个定理都是语义正确的.而完备性指所有语义正确的定理都可以通过这个系统推理出来.完备性是一个非常强的性质,通常的证明系统无法保证是完备的.但是可靠性是每个系统都必须具备的,否则就会产生矛盾结果.

2.2 模型检测方法

模型检测考虑的是协议的有限多种行为,检测它们满足一些正确条件.它更适合于去发现协议的攻击,而并非去证明协议的正确性.

模型检测有以下特点:

(1)关于协议操作行为的有限状态系统被刻画为有限状态迁移系统.这个系统的状态取决于与环境的交互,满足一定条件就迁移到另一个状态.这些条件被标记到迁移的边上.这个系统称为标记迁移系统(Labelled Transitive System, LTS).

(2)在每个状态上,有某些性质被满足,这些性质被描述为一个(古典逻辑或者时态等逻辑)公式.

(3)与定理证明一样,系统要满足的性质也被刻画为逻辑公式.

(4)用自动的手段检测上述的性质是否在系统的每个轨迹(trace)上都被满足.这里的轨迹是指系

统的一个可能迁移路径.

形式上说,假如一个系统为 S ,期望的系统性质表达为逻辑公式 φ ,那么模型检测就是验证是否 S 满足 φ .通常把 S 满足 φ 表示为 $S \models \varphi$.

模型检测的方法完全是自动化的,这是该方法的优点.但是,因为协议的行为是潜在无限的,而模型检测的方法只能够处理有限状态的系统,故而决定了这个方法的不完善性,即在实际应用过程中,一定要对检测的范围加以限制.同时,这个方法的另一个问题在于,尽管人们已经探索了 20 余年,对于状态爆炸问题仍然没有得到解决办法.就是说,随着系统变元的增加,状态数量成指数增加.因而对于复杂系统,用模型检测的方法去分析,需要进一步地研究和探索.实际上,常用的协议是不可判定的,这就决定了没有一个算法系统能够描述协议所有可能的行为.

总之,这个方法注定是一个不完全的方法,其显著的优点在于能直观地发现协议的漏洞.然而如果没有发现协议存在漏洞,将不保证协议的正确性.

2.3 互模拟等价

理论计算机科学中互模拟是状态迁移系统间的等价关系.互模拟等价的系统间有相同的行为方式.直观上,如果两个系统的迁移动作是相互对应的,则它们是互模拟的.在这种意义下,互模拟系统是观察者不可区分的.

假如 (S, \rightarrow) 是一个迁移系统,并且每个边上有一个标记 $\alpha \in \Delta$, 构成标记迁移图 (S, Δ, \rightarrow) . 假设 (S, Δ, \rightarrow) 和 $(S', \Delta, \rightarrow)$ 是两个标记迁移图. R 是 $S \times S'$ 上的一个二元关系,如果它满足下面的条件,则称 R 是 $S \times S'$ 上的一个互模拟关系.

$$\forall (p, q) \in R, \forall \alpha \in \Delta,$$

如果 $\exists p' \in S$ 使得 $p \xrightarrow{\alpha} p'$, 一定存在 $q' \in S'$, 使得 $q \xrightarrow{\alpha} q'$,

反之,如果 $\exists q'' \in S'$ 使得 $q \xrightarrow{\alpha} q''$, 一定存在 $p'' \in S$, 使得 $p \xrightarrow{\alpha} p''$.

互模拟关系分为弱互模拟和强互模拟等等.关于互模拟与各种等价的关系,请参考文献[45].

3 技术分类解析

本节主要讨论典型的形式化分析方法的主要技术特点.根据上节的分类,我们对于每一类中的典型情形给出一个说明.

3.1 基于逻辑推理的方法和模型

我们通过 BAN 逻辑的分析过程,说明基于逻

辑方法的安全协议分析过程. 正如我们前面所述, 这个方法属于定理证明范畴, 也就遵循定理证明的一般规律.

BAN 逻辑由 Burrows, Abadi 和 Needham 三个人在 1989 年提出^[9], 并习惯上用他们名字的首写字母命名. 这是最早的用于认证和密钥交换的形式化方法. BAN 逻辑的出现激发了对于这个领域的广泛兴趣. BAN 逻辑中的证明构造简洁, 容易完成. 但是, 要应用 BAN 逻辑分析一个协议, 从具体协议到逻辑表达式的抽象过程是比较困难的一步.

3.1.1 BAN 逻辑的构成

BAN 逻辑的语法中区分了三种密码要素: 主体、密钥和时鲜值. 协议的每个消息表达为该逻辑的一个公式. 假设 P, Q 代表参加协议的主体, K 代表密钥, X, Y 是公式. 逻辑的基本公式形式及其含意如图 1 所示.

$P \models X$	P 相信 X ; P 相信 X 真.
$P \triangleleft X$	P 看见 X ; P 曾经收到过包含 X 的消息并且读到了 X .
$P \vdash X$	P 曾经说过 X ; P 曾经发送过包含 X 的消息.
$P \models X$	P 可以裁定 X ; 信任 P 对于 X 的真值的判定.
$\#(X)$	X 是新鲜的; X 在当前协议运行前没有被发送过.
$P \xleftrightarrow{K} Q$	P 和 Q 分享一个好的密钥 K . 意思是, 密钥 K 对于 P, Q 以及他们信任的主体来说, 仍然具有保密性.
$\xrightarrow{K} P$	P 具有密钥 K . 相应的私钥是 K^{-1} , 这个私钥只有 P 以及他信任的人知道.
$P \xleftrightarrow{X} Q$	P 和 Q 分享秘密 X . 这个秘密只有 P, Q 以及他们信任的人知道.
$\langle X \rangle_K$	用 K 加密 X 后的消息.
$\langle X \rangle_Y$	X 与公式 Y 的结合.

图 1 BAN 逻辑的基本公式及其语义

BAN 的推导规则直观地反映了逻辑公式构造的语义. 我们把“逻辑公式 X_1, X_2, \dots, X_n 成立则 Y 成立”记为

$$\frac{X_1, X_2, \dots, X_n}{Y}$$

下面是一些特定的推演规则^[9].

消息意义规则:

$$\frac{P \models P \xleftrightarrow{K} Q, P \triangleleft \langle X \rangle_K}{P \models Q \vdash X}$$

公钥意义规则:

$$\frac{P \xrightarrow{K} Q, P \triangleleft \langle X \rangle_{K^{-1}}}{P \models Q \vdash X}$$

新鲜性规则:

$$\frac{P \models \#(X)}{P \models \#(X, Y)}$$

时鲜值验证规则:

$$\frac{P \models \#(X), P \models Q \vdash X}{P \models Q \models X}$$

裁定规则:

$$\frac{P \models Q \models X, P \models Q \models X}{P \models X}$$

3.1.2 示例分析

应用 BAN 逻辑进行协议验证的过程如下:

1. 首先将安全协议进行理想化, 一个理想化的协议是一集逻辑公式, 这些逻辑公式表达传送的消息的意图.

2. 加入协议的初始假设(公式).

3. 利用推演规则, 从上述的公式中推导目标公式. 这些所谓目标公式, 实际是协议的安全目标属性.

我们用一个简单的例子, 简要说明这个过程.

Needham-Schroeder 协议:

$$M_1 \quad A \rightarrow S: A, B, N_a,$$

$$M_2 \quad S \rightarrow A: \{N_a, B, K_{ab}, \langle K_{ab}, A \rangle_{K_{bs}}\}_{K_{as}},$$

$$M_3 \quad A \rightarrow B: \langle K_{ab}, A \rangle_{K_{bs}},$$

$$M_4 \quad B \rightarrow A: \{N_b\}_{K_{ab}},$$

$$M_5 \quad A \rightarrow B: \{N_b - 1\}_{K_{ab}}.$$

理想化后的协议:

$$M_2 \quad S \rightarrow A: \{N_a, A \xleftrightarrow{K_{ab}} B, \#(A \xleftrightarrow{K_{ab}} B), \langle A \xleftrightarrow{K_{ab}} B \rangle_{K_{bs}}\}_{K_{as}},$$

$$M_3 \quad A \rightarrow B: \langle A \xleftrightarrow{K_{ab}} B \rangle_{K_{bs}},$$

$$M_4 \quad B \rightarrow A: \{N_b, A \xleftrightarrow{K_{ab}} B\}_{K_{ab}} \text{ 来自 } B,$$

$$M_5 \quad A \rightarrow B: \{N_b, A \xleftrightarrow{K_{ab}} B\}_{K_{ab}} \text{ 来自 } A.$$

初始假设:

$$A \models A \xleftrightarrow{K_{as}} S, \quad B \models B \xleftrightarrow{K_{bs}} S,$$

$$S \models A \xleftrightarrow{K_{as}} S, \quad S \models B \xleftrightarrow{K_{bs}} S,$$

$$S \models A \xleftrightarrow{K_{ab}} B, \quad A \models (S \models A \xleftrightarrow{K_{as}} B),$$

$$B \models (S \models A \xleftrightarrow{K_{as}} B), \quad A \models (S \models \#(A \xleftrightarrow{K_{as}} B)),$$

$$A \models \#(N_a), \quad B \models \#(N_b),$$

$$S \models \#(A \xleftrightarrow{K_{ab}} B), \quad B \models \#(A \xleftrightarrow{K_{ab}} B).$$

目标公式:

$$A \models A \xleftrightarrow{K_{ab}} B, \quad B \models A \xleftrightarrow{K_{ab}} B,$$

$$A \models B \models A \xleftrightarrow{K_{ab}} B, \quad B \models A \models A \xleftrightarrow{K_{ab}} B.$$

利用这些公式和规则, 推导目标公式(略).

应该指出的是, 在假设部分的最后一个公式

$B \models \#(A \xleftrightarrow{K_{ab}} B)$ 是不合理的, 而我们推理最后一个目标时需要这个假设, 说明协议不能完全达到目标, 存在漏洞.

3.1.3 常见的用于安全协议分析的逻辑系统

由于 BAN 逻辑发现了一些协议存在漏洞,使得人们对于这种方式的验证过程有了极大的信心,从而激发了人们对于逻辑系统的研究.一方面,为了对 BAN 逻辑的局限性加以补足,新的逻辑系统被设计出来.另一方面针对不同的密码属性,研究、发展了新的逻辑系统.但是这些逻辑系统的技术原理大致是相同的.

我们把用于安全协议分析的逻辑系统罗列如下,供读者参考.

BAN 逻辑:基于信念逻辑,专门用于安全协议分析的逻辑系统.

GNY 逻辑^[46]:BAN 的扩充.

AT 逻辑^[47]:首次给出了 BAN 类逻辑的语义模型.

SVO 逻辑^[48]:综合了以前逻辑系统的优点,设计了一个简洁清晰的逻辑系统.

Kailar 逻辑^[49]:第一个用于电子商务协议验证的逻辑.主要验证协议的可追究性这个安全属性.

CS 逻辑^[50]:一种将时间与逻辑结合起来的系统,可以验证时间相关的协议.

KG 逻辑^[51]:一种与 BAN 逻辑类似的逻辑系统.

非单调逻辑^[52]:注意到知识推演的非单调性,Rubin 设计了一个用于安全协议分析的关于主体知识推演的非单调逻辑,并发现了一些协议中从未发现的漏洞.但是这个方向的工作没有进一步开展下去.

其它逻辑:BGNY^[53]; Moser^[54]; CKT5^[55]; KW 逻辑^[56].

3.2 归纳证明方法

归纳证明方法是数学中最常用的方法之一.应用归纳法证明方法证明协议的安全性可以不必一一考虑协议无限多种可能的行为,从而建立协议正确性的结论.这方面最有效的工作是 Paulson 的工作^[15].他利用高阶逻辑描述公式,使用定理证明器 Isabelle 证明协议的正确性.这个方法在协议的正确性方面通常较之模型检测的方法更有说服力.

数学中的归纳法一般是针对某个与自然数 n 相关的性质 $P(n)$.要证明 $P(n)$ 对于任何自然数都成立,需要说明 $P(0)$ 的正确性,并且在假设 $P(n)$ 正确的情况下,证明 $P(n+1)$ 的正确性.对于安全协议来说,归纳法要说明的是协议的安全属性 P 在协议运行环境下的正确性.这就需要证明在任何给定的观察事件集合的任何扩展下, P 都是安全的.归纳定义要罗列出一个主体或者系统的所有可能的动作,对应的归纳规则使得我们可以推理这些动作的任意有

限序列.

3.2.1 归纳证明方法的要素

安全协议分析的一个特点就是考虑敌手的能力.在归纳分析中,Paulson 定义了三种运算:parts, analz 和 synth.简单说来,parts(M)是消息集合 M 的所有消息的构成部分的集合. analz(M)是集合 M 中消息用(敌手)可得到的密钥解密能够得到的消息的集合.而 synth 刻画的是消息的伪造.后两种运算实际上是敌手能力的刻画.这些运算都是归纳定义的,在协议推理中起着主要作用.

(1)归纳法证明中,消息的要素有

主体名: A, B, \dots ;

时鲜值: N_a, N_b, \dots ;

密钥: K_a, K_b, K_{ab}, \dots ;

复合消息: $\{X_1, X_2\}$;

Hash 消息:Hash X ;

加密消息:Crypt KX .

该方法中的敌手模型仍然采用 Dolev-Yao 模型.三个运算的代数性质,也是 Dolev-Yao 模型下的性质.

(2)敌手的能力通过三个归纳定义的运算刻画.

对于任意消息集合 H , parts H 是包含 H 并对下列规则封闭的最小集合:

$$\frac{X \in H}{X \in \text{parts } H}, \quad \frac{\text{Crypt } KX \in \text{parts } H}{X \in \text{parts } H},$$

$$\frac{\{X, Y\} \in \text{parts } H}{X \in \text{parts } H}, \quad \frac{\{X, Y\} \in \text{parts } H}{Y \in \text{parts } H}.$$

对于任意消息集合 H , analz H 是包含 H 并对下列规则封闭的最小集合:

$$\frac{X \in H}{X \in \text{analz } H},$$

$$\frac{\text{Crypt } KX \in \text{analz } H \quad K^{-1} \in \text{analz } H}{X \in \text{analz } H},$$

$$\frac{\{X, Y\} \in \text{analz } H}{X \in \text{analz } H}, \quad \frac{\{X, Y\} \in \text{analz } H}{Y \in \text{analz } H}.$$

对于任意消息集合 H , synth H 是包含 H 、主体名、猜测数并对下列规则封闭的最小集合:

$$\text{Agent } A \in \text{synth } H, \quad \text{Number } N \in \text{synth } H,$$

$$\frac{X \in H}{X \in \text{synth } H}, \quad \frac{X \in \text{synth } H}{\text{Hash } X \in \text{synth } H},$$

$$\frac{X \in \text{synth } H \quad Y \in \text{synth } H}{\{X, Y\} \in \text{synth } H},$$

$$\frac{X \in \text{synth } H \quad K \in H}{\text{Crypt } KX \in \text{synth } H}.$$

(3)事件、轨迹以及主体的知识.

一个事件是协议中的一个通信动作,而轨迹是事件序列.在归纳证明方法中,事件有两种形式:一

种是 $Say\ A\ B\ X$, 表示主体 A 发送消息 X 给主体 B . 另一个可能的事件是 $Note\ A\ X$, 表示主体 A 内部存储 X .

轨迹可以根据协议定义的事件以及敌手的事件扩充. 首先 $[\]$ 表示空事件. 如果 evs 是一个事件序列, 即一个轨迹, 而 ev 是一个事件, 在符合协议要求的情况下, $ev \# evs$ 是一个轨迹 (注意这里沿用 Paulson^[15] 中的习惯, 将后一个事件连接在前面事件的前面, 而不是习惯上的后面).

归纳证明法中的主体有三类: 友好主体 $Friend\ i$ (i 是自然数, 表示第 i 个出现的友好主体)、服务器 S 以及敌手 Spy . 他们的初始知识集合由下列规则定义:

$initState\ S \stackrel{def}{=} \text{所有长期密钥},$

$initState(Friend\ i) \stackrel{def}{=} \{Key(shrK(Friend\ i))\},$

$initState\ Spy \stackrel{def}{=} \{Key(shrK(A)) \mid A \in bad\}.$

其中的 Key 表示所有密钥的集合, $shrK(A)$ 是与 A 分享的密钥. bad 是在协议运行过程中被敌手腐蚀的主体集合. 这里最为关注的是函数 $spies$. $spies$ 通过轨迹的扩充而归纳定义. 设 evs 是一个轨迹, $spies$ 归纳定义如下:

$spies[\] \stackrel{def}{=} initState\ Spy$

$spies((Say\ A\ B\ X) \# evs) \stackrel{def}{=} \{X\} \cup spies\ evs$

$spies((Note\ A\ X) \# evs) \stackrel{def}{=} \begin{cases} \{X\} \cup spies\ evs, & A \in bad \\ spies\ evs, & \text{否则} \end{cases}$

3.2.2 示例分析

我们通过对 Otway-Rees 协议进行归纳证明法的分析得出如下结果.

(1) Otway-Rees 协议的一个变种为

$A \rightarrow B: N_a, A, B, \{N_a, A, B\}_{K_a},$

$B \rightarrow S: N_a, A, B, \{N_a, A, B\}_{K_a}, N_b, \{N_a, A, B\}_{K_b},$

$S \rightarrow B: N_a, \{N_a, K_{ab}\}_{K_a}, \{N_b, K_{ab}\}_{K_b},$

$B \rightarrow A: N_a, \{N_a, K_{ab}\}_{K_a}.$

(2) 协议的刻画. 在归纳证明法中, 协议的动作用于扩充事件. 这可以逐步说明如下:

(a) 如果 evs 是一个轨迹, N_a 是一个新鲜的时鲜值, B 是不同于 A, S 的主体, 则 evs 可以扩充事件: $Say\ A\ B\ \{N_a, A, B, \{N_a, A, B\}_{K_a}\};$

(b) 对于轨迹 evs , 如果其中含有形如 $Say\ A' B\ \{N_a, A, B, X\}$ 的事件, N_b 是时鲜值, $B \neq S$, 则轨迹 evs 可以扩充事件: $Say\ B\ S\ \{N_a, A, B, X, N_b, \{N_a, A, B\}_{K_b}\};$

(c) 如果轨迹 evs 含有形如

$Say\ B' S\ \{N_a, A, B, \{N_a, A, B\}_{K_a}, N_b, \{N_a, A, B\}_{K_b}\}$

的事件, K_{ab} 是新的密钥, $B \neq S$, 则轨迹 evs 可以扩充事件: $Say\ S\ B\ \{N_a, \{N_a, K_{ab}\}_{K_a}, \{N_a, K_{ab}\}_{K_b}\};$

(d) 如果轨迹 evs 含有两个事件

$Say\ B\ S\ \{N_a, A, B, X', N_b, \{N_a, A, B\}_{K_b}\},$

$Say\ S' B\ \{N_a, X, \{N_b, K\}_{K_b}\}$

并且 $A \neq B$, 则 evs 可以扩充事件: $Say\ B\ A\ \{N_a, X\}.$

(e) 有一个蕴含的一步, 就是在最后, 主体 A 检测自己的时鲜值以确定会话的结束.

(3) 对于任何协议都有效的扩充规则是一个协议的描述通常有 3 个附加的规则:

(a) 空序列 $[\]$ 是一个轨迹.

(b) 如果 evs 是一个轨迹, $X \in synth(analz(H))$, 并且 $B \neq Spy$, 则 evs 可以扩充事件

$Say\ Spy\ B\ X,$

其中 H 含有过去轨迹中的所有消息.

(c) 如果 evs 是一个轨迹, S 在一个包含时鲜值 N_a 和 N_b 的运行中分发了 K , 则 evs 可以扩充事件

$Notes\ Spy\ \{N_a, N_b, K\}.$

(4) 证明保密性定理.

如果一个密钥 K 可以从一集密钥和一集消息得到, 那么, 如果 K 不是这集密钥 \mathcal{K} 中的一个, 就一定能够由这些消息单独得到. 形式化的描述是:

$K \in analz(\mathcal{K} \cup spies\ evs) \Leftrightarrow$

$K \in \mathcal{K} \vee K \in analz(spies\ evs),$

这里的 \mathcal{K} 是一集密钥, 而且不必是轨迹中的密钥.

证明这个定理是复杂的, 从原理上说, 只要应用归纳法, 对于 evs 的不同扩充, 进行归纳证明即可. 但是实现起来需要一系列的过程. 在 Paulson^[15] 中建议了以下的步骤 (详细略):

1. 应用归纳法.

2. 对于转发某个消息一部分的步骤中, 利用对应结论中以 $analz$ 表达的转发引理 (我们这里不作介绍).

3. 利用重写规则符号计算 $analz$ 以简化所有的情形, 包括抽取主体名、时鲜值、复合消息以及解密消息等等.

3.2.3 同类方法

一个非常类似于 Paulson 的归纳证明法的模型是 Woo-Lam 模型^[57]. Woo-Lam 定义了协议执行的分布式系统, 并且定义了扩充的规则, 这些规则在 Woo-Lam 系统中是迁移的条件. 该文中也定义了安全属性的表示方法.

另一个同类的模型^[58] 是利用抽象状态机 (ASM) 构造的安全协议的语义模型^①. 该模型利用了抽象状态机的分布式表达机制, 将认证和保密性

① 这个方法既不同于有限状态机的方法, 也不同于抽象机的方法, 三者模型各有千秋.

用一阶逻辑表达,利用分布式扩充规则,证明所期望的性质是协议运行的不变量。

3.2.4 其它类型的证明技术

串空间^[36]是近年来出现的一种表达协议的简洁模型。这个模型的主要优点是利用了图论的方式,定义了协议丛的概念以刻画协议分布式运行的状态。协议的安全属性是通过证明它们在所有丛上成立,而建立协议的正确性。正是这个模型的简洁而富有表达能力,它的出现引起了广泛的兴趣。这个模型与其它模型的关系^[35]也得到了探讨。基于串空间的模型检测系统^[16]也设计了出来。关于串空间的介绍国内已经有多篇文章,我们这里不做介绍。

其它的基于定理证明方法的系统有 Dolev-Yao 方法^[3]、Kemmerer^[7]的方法、Cevesato 等人^[34]的利用线性逻辑与多集重写系统构置的系统、Denker 等人的基于 Maude 系统的利用重写逻辑构建的规范系统、Merritt^[59]的代数方法、Toussaint^[60]的方法等等。

3.3 基于模型检测的方法

在第 2.2 节中,我们已经对于模型检测的基本思想和构成作了简单的介绍。在本节我们将通过基于 CSP 的模型检测方法,利用 FDR 检测器说明模型检测的一般技术方法。

Lowe^[11]发展了利用 FDR 进行安全协议检测的方法。这个方法发现了 Needham-Schroeder 协议的一个前所未见的攻击。全面介绍这个方法的书籍有 Ryan 和 Schneider 写的“Modelling and Analysis of Security Protocols”^[61]。

3.3.1 CSP 介绍

CSP(Communicating Sequential Process)是一个描述传值并行系统的概念,是研究并发系统问题的良好工具。Hoare 作为发明人有一部很好的著作^[30]介绍 CSP。对于利用 CSP 以及利用 FDR 检测安全协议方面的论述,请参考文献^[61]。

(1)CSP 中进程的表达式

CSP 是描述交互的概念,可以用于表达非常广泛的一类系统。它们的共同特性是,都由相互影响的不同部分构成。一个进程是一序列的动作,每个或者一系列(通信)动作是一个事件。CSP 的进程通过通信与其它进程或者环境进行交互。因此,CSP 主要刻画进程的方式。可见的动作有若干种,用算子(见图 2)表示。所有的动作集合是 Σ ,而不可见的动作用一个符号 τ 表示。

$Stop$	没有任何动作的进程
$\tau x;A \rightarrow P$	事件前节选择
$P \square Q$	在两个进程中选择
$P \sqcap Q$	非确定性选择
$P \parallel Q$	锁步并行
$P \parallel_x Q$	界面并行
$\parallel S$	一般的层叠
$P[R]$	进程关系的重命名
$P = F(P)$	递归定义
$a \rightarrow P$	事件前节
$c \tau x;A \rightarrow P$	输入前节选择
$\square S$	一般选择
$\sqcap S$	一般非确定性选择
$P_x \parallel_y Q$	同步并行
$Skip$	成功地终止
$P \setminus X$	事件隐藏
$P;Q$	序列复合
$\mu P.F(P)$	递归进程

图 2 CSP 的算子

下面我们通过例子对于图 2 所示的算子进行简短的说明,不尽之处,请参看相应的参考书。

最简单的进程是 $Stop$,它的意思是不做任何动作。

加前缀。假如 a 是一个通信, P 是一个进程。则 $a \rightarrow P$ 是一个进程,表示通信按照 a 执行,然后依照 P 行动。

设 to 和 fro 是 Σ 中的两个动作,则 $Alt = to \rightarrow fro \rightarrow Alt$ 是一个进程,表示循环依次的执行 to 和 fro 。这是一个递归进程,也可以表示为 $\mu P.to \rightarrow fro \rightarrow P$ 。

在前节构造中,可以提供一种选择机制:如果 $A \subseteq \Sigma$ 是一集可见动作, $\tau x;A \rightarrow P(x)$ 表示进程;如果选择 $x \in A$,则按照 $P(x)$ 动作。这里的 x 是 $P(x)$ 中的变量。

如果 c 是一个信道的名字,每个信道名有一个类型 L 。用 $?$ 表示输入动作, $!$ 表示输出动作。那么 $c \tau x \cdot x+1$ 表示动作:输入 L 类的一个数据 x ,而后输出数据 $x+1$ 。

\square 是一个选择符号,其作用是在两个进程中选择一个进程,并且执行选定的进程。另一个记号 \sqcap 是非确定性选择。这两个符号都是表示选择, $P \square Q$ 表示每次选定 P, Q 中的一个执行。而 $P \sqcap Q$ 表示每次选定的进程不可预测,在同样的条件下,选择的可能不同。

进程的并行有 3 种: $P \parallel Q, P \parallel_x Q$ 和 $P_x \parallel_y Q$ 。但是与安全协议的规范有关的只有前两种:

$P \parallel Q$ 锁步并行:意思是 P 和 Q 每次执行的动作同步。例如

$$(\exists x:A \rightarrow P(x)) \parallel (\exists x:B \rightarrow Q(x)) = \\ \exists x:A \cap B \rightarrow (P(x) \parallel Q(x)),$$

又如:

$$(c \downarrow x \rightarrow P) \parallel (c \downarrow y \rightarrow Q(y)) = c \downarrow x \rightarrow (P \parallel Q(x)).$$

$P \parallel_X Q$ 界面并行: 迫使 P 和 Q 在 X 中的所有事件上同步, 而在 X 之外的动作上自由执行. 如果 $P = \exists x:A \rightarrow P'(x)$, $Q = \exists x:B \rightarrow Q'(x)$, 则

$$P \parallel_X Q = \exists x:X \cap A \cap B \rightarrow (P'(x) \parallel Q'(x)),$$

$$\square \exists x:A \setminus X \rightarrow (P'(x) \parallel_X Q),$$

$$\square \exists x:B \setminus X \rightarrow (P \parallel_X Q'(x)).$$

(2) CSP 中进程的轨迹模型(trace model)

进程的轨迹模型是描述进程行为的重要手段.

某个进程的一个有限轨迹实际上是这个进程通信至某个时刻为止发生的可见事件的序列. 由此, CSP 的进程 P 的轨迹模型是它的轨迹集合. 例如

$$\text{traces}(\text{Stop}) = \{\langle \rangle\}. \text{ 其中 } \langle \rangle \text{ 是空轨迹.}$$

$\text{traces}(\mu P.a \rightarrow P \square b \rightarrow \text{Skip}) = \{\langle a \rangle^n, \langle a \rangle^n \langle b \rangle, \langle a \rangle^n \langle b, \sqrt{\} \rangle\}$. 其中 st 是轨迹 s 和 t 的联结. s^n 是 n 个 s 的联结. 事件 $\sqrt{\}$ 是一个特殊事件, 表示一个轨迹的终止.

进程的轨迹模型是非空的对于前节封闭的轨迹集合. 有一些规则可以直接计算进程的轨迹模型, 对于每种构造, 有一种对应的规则. 图 3 的各款表明的是各个算子的意义.

$$\begin{aligned} \text{traces}(\text{Stop}) &= \{\langle \rangle\} \\ \text{traces}(a \rightarrow P) &= \{\langle \rangle\} \cup \{\langle a \rangle s \mid s \in \text{traces}(P)\} \\ \text{traces}(\exists x:A \rightarrow P) &= \{\langle \rangle\} \cup \{\langle a \rangle s \mid a \in A \wedge s \in \text{traces}(P[a/x])\} \\ \text{traces}(c \downarrow A \rightarrow P) &= \{\langle \rangle\} \cup \{\langle c.a \rangle s \mid a \in A \wedge s \in \text{traces}(P[a/x])\} \\ \text{traces}(P \square Q) &= \text{traces}(P) \cup \text{traces}(Q) \\ \text{traces}(\square S) &= \cup \{\text{traces}(P) \mid P \in S\} \\ \text{traces}(P \sqcap Q) &= \text{traces}(P) \cup \text{traces}(Q) \\ \text{traces}(\sqcap S) &= \cup \{\text{traces}(P) \mid P \in S\} \\ \text{traces}(P \parallel Q) &= \text{traces}(P) \cap \text{traces}(Q) \\ \text{traces}(P \parallel_X Q) &= \{s \in (X \cup Y)^* \sqrt{\} \mid s \uparrow X \in \text{traces}(P) \wedge \\ &\quad s \uparrow Y \in \text{traces}(Q)\} \\ \text{traces}(P \parallel_X Q) &= \cup \{s \parallel_X t \mid s \in \text{traces}(P) \wedge t \in \text{traces}(Q)\} \\ \text{traces}(\parallel_{i \in I} P_i) &= \cup_{F \subseteq \text{finite } I} \text{traces}(\parallel_{i \in F} P_i) \\ \text{traces}(P \setminus X) &= \{s \setminus X \mid s \in \text{traces}(P)\} \\ \text{traces}(P \parallel R) &= \{t \mid \exists s \in \text{traces}(P) \bullet sR * t\} \\ \text{traces}(\text{Skip}) &= \{\langle \rangle, \langle \sqrt{\} \rangle\} \\ \text{traces}(P; Q) &= \text{traces}(P) \cap \Sigma * \cup \{st \mid s \langle \sqrt{\} \rangle\} \\ \text{traces}(P) &= \cup \{\text{traces}(F^n(\text{Stop})) \mid n \in \mathbb{N}\}, \text{ 其中 } P = F(P). \end{aligned}$$

图 3 简单进程的轨迹模型——算子的意义

这里对几个符号作简要解释: $X \sqrt{\}$ 是 $X \cup \{\sqrt{\}\}$ 的简写. $s \uparrow Z$ 的意思是 s 限制到 Z , 是 s 中的所有非 Z 中的成员被去掉后得到的轨迹.

轨迹模型的作用很大. 它可以刻画进程的行为

和性质, 在安全协议中, 认证的性质就是轨迹性质. 下面的例子取自文献[61].

例 1. 性质: 信道 *right* 的输出总是信道 *left* 输入的前节, 并且没有其它行为. 这是缓冲器的一个刻画.

$$tr = tr \uparrow \{\text{right}, \text{left}\} \wedge tr \downarrow \text{right} \leq tr \downarrow \text{left},$$

其中 $tr \downarrow a$ 是序列 tr 中沿信道 a 通信的序列.

例 2. 性质: 事件 *commit* 之前一定有事件 *start* 接下来 *running* 的发生. 而这两个事件一定先于 *commit* 发生.

$$tr = tr' \langle \text{commit} \rangle \rightarrow \exists tr_1, tr_2 \bullet tr' = tr_1 tr_2 \wedge$$

$$\langle \text{start}, \text{running} \rangle \leq tr_2 \uparrow \{\text{start}, \text{running}\} \wedge$$

$$tr_2 \uparrow \{\text{commit}\} = \langle \rangle.$$

例 3. 性质: 事件 *error* 永不发生:

$$tr \uparrow \{\text{error}\} = \langle \rangle.$$

如果进程 P 的所有轨迹均满足逻辑性质 $S(tr)$, 则记为 $\forall tr \in \text{traces}(P) \bullet tr \text{ sat } S(tr)$, 简记为 $P \text{ sat } S(tr)$. 一般在逻辑中, 我们用 $P \models S(tr)$ 表达. 它有如下的性质:

对于任何进程 P , $P \text{ sat } \text{true}(tr)$,

$$P \text{ sat } S(tr) \wedge P \text{ sat } T(tr) \Rightarrow (P \text{ sat } S(tr) \wedge T(tr)),$$

$$P \text{ sat } S(tr) \wedge (S(tr) \Rightarrow T(tr)) \Rightarrow P \text{ sat } T(tr).$$

一个进程是否满足一个逻辑性质, 可以通过定义来验证. 但是一般说来, 这种方法比较复杂. 可以通过复合推演规则来将一个进程的可满足性化为它的子进程的满足性问题. 常用的一些规则如下:

Rule sat.stop

$$\frac{}{\text{Stop sat } tr = \langle \rangle}.$$

Rule sat.prefix

$$\frac{P \text{ sat } S(tr)}{a \rightarrow P \text{ sat } tr = \langle \rangle \vee () \text{ sat } \langle a \rangle tr' \wedge S(tr')}.$$

Rule sat.extchoice

$$\frac{\forall i \bullet P(i) \text{ sat } S(tr)}{\square_i P(i) \text{ sat } S(tr)}.$$

Rule sat.parallel

$$\frac{P \text{ sat } S(tr) \quad Q \text{ sat } T(tr)}{P \parallel Q \text{ sat } S \wedge T(tr)}.$$

Rule sat.interleave

$$\frac{P \text{ sat } S(tr) \quad Q \text{ sat } T(tr)}{P \parallel \parallel Q \text{ sat } S(tr \uparrow \sigma(P)) \wedge T(tr \uparrow \sigma(Q))},$$

$$[\sigma(P) \cap \sigma(Q) = \emptyset].$$

3.3.2 描述和检测安全协议

在 CSP 中刻画和检测协议主要是描述参与协议的主体的进程,描述敌手的行为,将他们的行为进程在 CSP 中并行起来,构成一个系统进程;进一步,在轨迹模型中描述安全属性,验证是否被系统进程所满足.我们仍然用 Needham-Schroeder 协议说明这一点.

Needham-Schroeder 协议在 3.1.2 节中已经给出,我们这里利用的是简化的 Needham-Schroeder 协议:

$$\begin{aligned} M_1 \quad A \rightarrow B: \{A, N_a\}_{K_b}, \\ M_2 \quad B \rightarrow A: \{N_a, N_b\}_{K_a}, \\ M_3 \quad A \rightarrow B: \{N_b\}. \end{aligned}$$

(1) 利用 CSP 描述简化的 Needham-Schroeder 协议

在上述协议中有两类参与协议的主体:一类是协议的初始者,他们寻求与另一类主体——响应者进行一个新的会话.我们记初始者全体为 *Initiator*,响应者全体的集合为 *Responder*,公钥的集合为 *Key*,时鲜值的集合为 *Nonce*.

根据 3.3.1 节的介绍,将下面的协议消息:

$$M_1 \quad A \rightarrow B: \{A, N_a\}_{K_b}$$

表达为 CSP 的事件:

$$comm.Msg1.A.B.Encrypt.K_b.N_a.A.$$

对应于协议中三步的通信事件的集合为

$$MSG1 \stackrel{\text{def}}{=} \{Msg1.a.b.Encrypt.k.n_a.a' \mid a, a' \in Initiator, \\ b \in Responder, k \in Key, n_a \in Nonce\},$$

$$MSG2 \stackrel{\text{def}}{=} \{Msg2.b.a.Encrypt.k.n_a.n_b \mid a \in Initiator, \\ b \in Responder, k \in Key, n_a, n_b \in Nonce\},$$

$$MSG3 \stackrel{\text{def}}{=} \{Msg3.a.b.Encrypt.k.n_b \mid a \in Initiator, \\ b \in Responder, k \in Key, n_b \in Nonce\},$$

$$MSG \stackrel{\text{def}}{=} MSG1 \cup MSG2 \cup MSG3.$$

对于标准的通信信道使用 *comm*,为了描述敌手的伪造和截断行为,分别使用 *fake* 和 *intercept* 信道.

Channel *comm, fake, intercept; MSG*.

还有两个外部信道: *user.a.b* 和 *session.a.b*.前者表示一个来自 *a* 的请求与响应者 *b* 建立一个会话.后者表示会话的结果.为了表达认证的关系,增加表达主体的状态的信道:

$I_{\text{running}.a.b}$ 表示初始者 *a* 认为自己与响应者 *b* 参加了协议的运行.

$R_{\text{running}.a.b}$ 表示响应者 *b* 认为自己与初始者 *a*

进行了协议运行.

$I_{\text{commit}.a.b}$ 表示初始者 *a* 对会话的承诺.

$R_{\text{commit}.a.b}$ 表示响应者 *b* 对会话的承诺.

对于初始者 *a* 用时鲜值 n_a 与响应者 *b* 进行的会话进程为

$$\begin{aligned} INITIATOR(a, n_a) \stackrel{\text{def}}{=} & user.a \bar{a} \rightarrow I_{\text{running}.a.b} \rightarrow \\ & comm !Msg1.a.b.Encrypt.key(b).n_a.a \rightarrow \\ & comm.Msg2.b.a.Encrypt.key(a).n'_a.n_b \rightarrow \\ & \text{if } n_a = n'_a \\ & \text{then } comm !Msg3.a.b.Encrypt.key(b).n_b \rightarrow \\ & \quad I_{\text{commit}.a.b} \rightarrow session.a.b \rightarrow Skip \\ & \text{else } Stop. \end{aligned}$$

如果考虑敌手的行为,敌手可以作为合法的初始者,也可以截断和伪造消息,那么一般的初始者的进程通过对于上述合法主体的行为重命名的方式得到进程 *INITIATOR1*:

$INITIATOR(A, N_A)$

$$\left[comm.Msg1/comm.Msg1, comm.Msg1/intercept.Msg1, \right. \\ \left. comm.Msg2/comm.Msg2, comm.Msg2/fake.Msg2, \right. \\ \left. comm.Msg3/comm.Msg3, comm.Msg3/intercept.Msg3 \right]$$

响应者的进程 *RESPONDER1* 可以类似地描述.我们要描述敌手的其它行为.根据 Dolev-Yao 模型,敌手可以观察道系统中的所有消息,可能截断它们.如果消息是由敌手知道的密钥加密而成的密文,则敌手就可以知道新的时鲜值;否则敌手就记住这个消息.敌手可以增加伪造的消息等等.这些可以通过敌手的一般进程表示,如图 4.

$$\begin{aligned} I(m1s, m2s, m3s, ns) \stackrel{\text{def}}{=} & comm.Msg1 \bar{a}.b.Encrypt.k.n.a' \rightarrow \\ & \text{if } k = K_i \text{ then } I(m1s, m2s, m3s, ns \cup \{n\}) \\ & \text{else } I(m1s \cup \{Encrypt.k.n.a'\}, m2s, m3s, ns) \\ \square & intercept.Msg1 \bar{a}.b.Encrypt.k.n.a' \rightarrow \\ & \text{if } k = K_i \text{ then } I(m1s, m2s, m3s, ns \cup \{n\}) \\ & \text{else } I(m1s \cup \{Encrypt.k.n.a'\}, m2s, m3s, ns) \\ \square & comm.Msg2 \bar{b}.a.Encrypt.k.n.n' \rightarrow \\ & \text{if } k = K_i \text{ then } I(m1s, m2s, m3s, ns \cup \{n, n'\}) \\ & \text{else } I(m1s, m2s \cup \{Encrypt.k.n.n'\}, m3s, ns) \\ \square & intercept.Msg2 \bar{b}.a.Encrypt.k.n.n' \rightarrow \\ & \text{if } k = K_i \text{ then } I(m1s, m2s, m3s, ns \cup \{n, n'\}) \\ & \text{else } I(m1s, m2s, m3s \cup \{Encrypt.k.n, ns\}) \\ \square & comm.Msg3 \bar{a}.b.Encrypt.k.n \rightarrow \\ & \text{if } k = K_i \text{ then } I(m1s, m2s, m3s, ns \cup \{n\}) \\ & \text{else } I(m1s, m2s, m3s \cup \{Encrypt.k.n\}, ns) \\ \square & intercept.Msg3 \bar{a}.b.Encrypt.k.n \rightarrow \\ & \text{if } k = K_i \text{ then } I(m1s, m2s, m3s, ns \cup \{n\}) \\ & \text{else } I(m1s, m2s, m3s \cup \{Encrypt.k.n\}, ns) \\ \square & fake.Msg1 \bar{a}.b \bar{m}; m1s \rightarrow I(m1s, m2s, m3s, ns) \\ \square & fake.Msg2 \bar{a}.b \bar{m}; m2s \rightarrow I(m1s, m2s, m3s, ns) \\ \square & fake.Msg3 \bar{a}.b \bar{m}; m3s \rightarrow I(m1s, m2s, m3s, ns) \\ \square & fake.Msg1 \bar{a}.b !Encrypt \bar{a} \bar{n}; ns \bar{a}' \rightarrow I(m1s, m2s, m3s, ns) \\ \square & fake.Msg2 \bar{a}.b !Encrypt \bar{a} \bar{n}; ns \bar{a}' \rightarrow I(m1s, m2s, m3s, ns) \\ \square & fake.Msg3 \bar{a}.b !Encrypt \bar{a} \bar{n}; ns \rightarrow I(m1s, m2s, m3s, ns) \end{aligned}$$

图 4 敌手进程的一般形式

如果敌手开始知道时鲜值 N_i , 则

$$INTRUDER \stackrel{\text{def}}{=} I(\{\}, \{\}, \{\}, \{N_i\}).$$

现在可以定义具有敌手的系统进程为

$$SYSTEM \stackrel{\text{def}}{=} INITIATOR1 \parallel RESPONDER1 \parallel INTRUDER.$$

(2) 认证属性的描述.

认证有两方面, 一种是初始者对于响应者的认证, 另一方面是响应者对于初始者的认证. 初始者对于响应者的认证要求描述: 初始者 A 承诺一个与 B 的会话, 只有当 B 的确参与了会话. 根据我们的记号, 有

$$AR_0 \stackrel{\text{def}}{=} R_{\text{running}}.A.B \rightarrow I_{\text{commit}}.A.B \rightarrow AR_0,$$

$$A_1 \stackrel{\text{def}}{=} \{R_{\text{running}}.A.B, I_{\text{commit}}.A.B\},$$

$$AR \stackrel{\text{def}}{=} AR_0 \parallel RUN(\Sigma \setminus A_1).$$

AR 表达事件 $I_{\text{running}}.A.B$ 只应该发生在事件 $R_{\text{running}}.A.B$ 之后. 与 $RUN(\Sigma \setminus A_1)$ 的层叠允许其它事件可以任意出现.

同样, 响应者对于初始者的认证要求描述: 响应者 B 承诺一个与 A 的会话, 只有当 A 的确参与了会话. 根据我们的记号, 有

$$AI_0 \stackrel{\text{def}}{=} I_{\text{running}}.A.B \rightarrow R_{\text{commit}}.A.B \rightarrow AI_0,$$

$$A_2 \stackrel{\text{def}}{=} \{I_{\text{running}}.A.B, R_{\text{commit}}.A.B\},$$

$$AI \stackrel{\text{def}}{=} AI_0 \parallel RUN(\Sigma \setminus A_2)$$

有了这些准备, 我们可以利用 FDR 进行验证, 则有下面的关系成立.

$$SYSTEM \text{ sat } AR \wedge AI.$$

3.4 模型检测与定理证明的混合方法

美国海军研究实验室(NRL)开发了一个专用的软件工具, 称为分析器^[43]. 这是最早的专用于安全协议分析的工具之一, 是用 Prolog 语言实现的系统.

这个分析器是混合系统, 同时具有模型检测和定理证明器的特征. 搜索的起点是不安全状态, 如果这个状态能够由初始状态达到, 那么这就是一个攻击. 这个系统利用项重写系统进行推理, 可以证明一个无限状态类是不可达的. 这就可能证明一些非安全状态是(从初始态)不可达的. 因此, 这个工具具有模型检测和定理证明两种方法的特征.

协议的规范由下面几个因素构成:

系统状态. 包括敌手和协议主体知道的知识以及已经发生的事件序列.

协议规则. 说明诚实主体的行为, 每一步协议后敌手会得到的知识. 敌手可能用已知的密钥加密或解密, 联结已知消息.

重写规则. 用来定义密码性质. 比如加密与解密

运算是互逆运算.

这个分析器被用于分析大量的协议, 找到过新的漏洞, 是一个代表性的系统.

3.5 互模拟等价模型下的系统——Spi-运算

Spi-演算^[14]是 Pi-演算^[62]的一种应用密码原语进行的扩充, 是专门设计用来描述安全协议的. 这些安全协议依赖于密码学和通信渠道, 因此密码运算以及通过信道的通信是 Spi-演算的要素.

在 Pi-演算中, 信道的描述是简单而强大的, 信道可以产生或者被传递. 而 Pi-演算的辖域规则确保了协议的环境无法存取一个没有显然给出的信道. Pi-演算是一种理想的安全通信协议的演算.

正因如此, Abadi 和 Gordon 扩充 Pi-演算, 使得密码学的一些要素能够得到合理的表达. 我们在这节主要介绍这个技术方法, 领略如何使用进程代数中一个有力的武器——互模拟(bisimulation)进行安全协议性质的验证.

3.5.1 Spi-演算

我们给出 Spi-演算的语法和非形式的语义, 介绍操作语义的主要概念, 定义测试等价的概念.

语法.

假设有两个无限集合: 一个是名字集合, 另一个是变元集合. 用 c, m, n, p, q 表示名字, w, x, y, z 表示变元, 密钥 k, K 也是名字.

项的集合通过图 5 所示的语法过程定义.

$L, M, N ::=$	项
n	名字
(M, N)	对偶
0	零
$\text{suc}(M)$	后继
$\{M\}_N$	共享密钥加密
x	变元

图 5 Spi-演算中项的定义

直观上, $\{M\}_N$ 表示用密钥 N 加密项 M 而得到的项. 在标准的 Pi-演算中, 名字是仅有的项, 这里增加了对偶、0 以及 $\text{suc}(M)$, 也区分了变元与名字.

进程的集合的语法定义见图 6.

$P, Q, R ::=$	进程
$M(N).P$	输出
$M(x).P$	输入
$P Q$	复合
$(\nu n)P$	限制
$!P$	重复
$[M \text{ is } N]P$	匹配
0	零
$\text{let}(x, y) = M \text{ in } P$	对偶分裂
$\text{case } M \text{ of } 0: P \text{ suc}(x): Q$	整数情形
$\text{case } L \text{ of } \{x\}_N \text{ in } P$	解密

图 6 Spi-演算中进程的定义

约束变元和名字的范围为:在 $M(x).P$ 中, x 的范围为 P . 在 $(\nu n)P$ 中 n 的范围为 P . 在 $let(x, y) = M \text{ in } P$ 中, x 和 y 的范围是 P . 在 $case M \text{ of } 0: P \text{ suc}(x): Q$ 中, x 的范围是 Q . 在 $case L \text{ of } \{x\}_N \text{ in } P$ 中, x 的范围是 P .

一般把 $\bar{M}\langle N \rangle.0$ 记为 $\bar{M}\langle N \rangle$. 如果把 P 中自由变元 x 替换为 M 后得到的进程记为 $P[M/x]$. 我们介绍上述图标中每个进程的非形式语义:

输出进程 $\bar{M}\langle N \rangle.P$ 准备在 M 输出 N , 然后行使 P . 这个输出只有当有一个在 M 输入值时, 才成为可能. 输入进程 $M(x).Q$ 准备在 M 输入值, 然后行使 $Q[N/x]$, 其中 N 是收到的输入.

复合 $P|Q$ 为 P 和 Q 的并行.

限制 $(\nu n)P$ 是产生新的、可能在 P 中出现的私名 n , 然后行为 P .

重复 $!P$ 的行为是无限多个 P 的副本并行运行.

匹配 $[M \text{ is } N]P$ 在 $M=N$ 的条件下行为如 P , 其它情况下则宕掉, 即没有任何动作.

零进程 0 没有任何动作.

对偶分裂进程 $let(x, y) = M \text{ in } P$ 在 M 是一个对偶 (N, L) 时的行为 $P[N/x][L/y]$, 否则宕掉.

一个整数情形的进程 $case M \text{ of } 0: P \text{ suc}(x): Q$ 当 M 是 0 时行为 P , 当 M 为 $suc(N)$ 时行为 $Q[N/x]$, 其它情形则宕掉.

一个解密进程 $case L \text{ of } \{x\}_N \text{ in } P$ 试图用密钥 N 解密 L . 如果 L 具有形式 $\{M\}_N$, 则进程行为 $P[M/x]$; 否则宕掉.

操作语义.

假设 P 是一个进程, 则 $(x)P$ 表示一个抽象. $(x)P$ 就像进程 $p(x).P$ 减去名字 p . 一个具体化 (concretion) 是一个表达式 $(\nu m_1, \nu m_2, \dots, \nu m_k)\langle M \rangle P$, 其中 M 是一个项, P 是一个进程, 名字 m_1, m_2, \dots, m_k 在 M 和 P 中约束. 在 Pi-演算中动作有三种: 一种是不可见动作, 记为 τ , 一种是 m , 另一种是 \bar{m} , 其中 m 是一个名字. 承诺关系: $P \xrightarrow{\alpha} A$ 定义如下:

$P \xrightarrow{\tau} Q$ 意思是在一个不可见步子之后 P 变成 Q .

$P \xrightarrow{m} (x)Q$ 意思是 P 已经准备用一步接收 m 上的一个值 x , 而后变成 Q .

$P \xrightarrow{\bar{m}} (\nu m_1, \nu m_2, \dots, \nu m_k)\langle M \rangle Q$ 意思是 P 已经准备用一步产生新的名字 m_1, m_2, \dots, m_k , 在 m 上发送 M , 而后变成 Q .

测试等价.

称两个闭(没有自由变量的)进程 P 和 Q 是测试等价的, 记为 $P \simeq Q$, 如果下面的条件成立:

对于每个闭进程 R 以及所有可见动作 m , 如果对于某个 P' 和 A , 有

$$P|R \xrightarrow{\tau^*} P' \xrightarrow{m} A,$$

则一定存在某个 Q' 和 B , 有

$$Q|R \xrightarrow{\tau^*} Q' \xrightarrow{m} B,$$

反之亦然.

许多性质, 包括安全属性(如秘密性)都是通过测试等价定义的, 我们在后面的例子中将给出具体的描述实例.

3.5.2 互模拟等价

为了计算测试等价, Spi-演算的作者定义了一种互模拟等价, 被作者称为框架互模拟 (framed bisimulation). 这个互模拟是通过框架和理论定义的. 空间和理论的内容将两个进程 P 和 Q 联系起来. 框架和理论表示了进程环境的知识.

一个框架是一有限名字集. 直观上, 一个框架是进程 P 和 Q 的环境可能出现的所有名字的集合. 变元 fr 在框架上取值.

一个理论是项对的有限集合. 直观上, 一个理论包括一个对 (M, N) , 表明环境无法区分来自 P 的数据 M 和来自 Q 的数据 N . 变元 th 在理论上取值.

下面通过一集规则递归定义断言 $(fr, th) \vdash M \leftrightarrow N$. 直观上这个断言意思是环境不能区分来自 P 的 M 和来自 Q 的 N , 并且环境具有(能够构造) M 和 N 分别与 P 和 Q 交互.

$$\frac{n \in fr}{(fr, th) \vdash n \leftrightarrow n}, \quad \frac{(M, N) \in th}{(fr, th) \vdash M \leftrightarrow N},$$

$$\frac{}{(fr, th) \vdash x \leftrightarrow x},$$

$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash N \leftrightarrow N'}{(fr, th) \vdash (M, N) \leftrightarrow (M', N')},$$

$$\frac{}{(fr, th) \vdash 0 \leftrightarrow 0},$$

$$\frac{(fr, th) \vdash M \leftrightarrow M'}{(fr, th) \vdash suc(M) \leftrightarrow suc(M')},$$

$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash N \leftrightarrow N'}{(fr, th) \vdash \{M\}_N \leftrightarrow \{M'\}_{N'}}.$$

为了定义模拟关系, 首先定义一个概念: 说 (fr, th) 是好的, 记为 $(fr, th) \vdash ok$, 如果下面两个条件满足:

①一旦 $(M, N) \in th$,

则 M 是闭的, 并且存在项 M_1 和 M_2 使得 $M =$

$\{M_1\}_{M_2}$ 以及不存在 N_2 , 使得 $(fr, th) \vdash M_2 \leftrightarrow N_2$;

N 是闭的, 并且存在项 N_1 和 N_2 , 使得 $N = \{N_1\}_{N_2}$ 以及不存在 M_2 , 使得 $(fr, th) \vdash M_2 \leftrightarrow N_2$.

② 对于 $(M, N) \in th$ 以及 $(M', N') \in th$, 如果 $M = M'$, 则 $N = N'$.

一个框架性的进程关系 \mathcal{R} 是一个由四元对 (fr, th, P, Q) 构成的集合. 通常如果 $(fr, th, P, Q) \in \mathcal{R}$, 则记为 $(fr, th) \vdash PRQ$. 一个框架性模拟是一个框架性关系 \mathcal{S} , 满足如果 $(fr, th) \vdash PSQ$, 下面 3 个条件成立:

① 如果 $P \xrightarrow{\tau} P'$ 则存在一个进程 Q' , 使得 $Q \xrightarrow{\tau} Q'$, 并且 $(fr, th) \vdash P'SQ'$.

② 如果 $P \xrightarrow{c} (x)P'$ 以及 $c \in fr$, 则存在一个抽象 $(x)Q'$ 使得 $Q \xrightarrow{c} (x)Q'$ 以及对于所有集合与 $fn(Q) \cup fn(\pi_2(th)) \cup fr$ 不交的集合 $\{n_1, n_2, \dots, n_k\}$, 所有闭的 M, N , 如果 $(fr \cup \{n_1, n_2, \dots, n_k\}, th) \vdash M \leftrightarrow N$, 则 $(fr \cup \{n_1, n_2, \dots, n_k\}, th) \vdash P'[M/x]SQ'[N/x]$. 其中 $fn(P)$ 是 P 中所有自由变元的集合, 而 π_2 是对于第 2 个坐标的投射函数.

③ 如果 $P \xrightarrow{c} (\nu \vec{m}) \langle M \rangle P'$, 集合 \vec{m} 与 $fn(P) \cup fn(\pi_1(th)) \cup fr$ 不交, 则存在一个具体化 $(\nu \vec{n}) \langle N \rangle Q'$, 使得 $Q \xrightarrow{c} (\nu \vec{n}) \langle N \rangle Q'$, 并且集合 \vec{n} 与 $fn(Q) \cup fn(\pi_2(th)) \cup fr$ 不交. 存在一个框架理论 (fr', th') , 满足 $(fr, th) \leq (fr', th')$, $(fr', th') \vdash M \leftrightarrow N$ 和 $(fr', th') \vdash P'SQ'$.

一个框架性的互模拟关系是一个框架性的模拟关系 \mathcal{S} , 而且 \mathcal{S}^{-1} 也是框架性模拟关系. 最后定义所有框架性互模拟关系的并为互模拟关系 \sim_f .

定理 1 (可靠性定理). 对于任何闭进程 P 和 Q , 任意名字 $n \notin fn(P) \cup fn(Q)$, 如果 $(fn(P) \cup fn(Q) \setminus \{n\}, \emptyset) \vdash P \sim_f Q$, 则 $P \simeq Q$.

这个定理建立了测试等价与互模拟之间的关系, 尽管这个关系有些遗憾 (即完备性不成立). 然而正如作者指出的那样, 这个关系的成立, 使得我们可以用互模拟的手法验证测试等价, 对于我们的目的通常足够了.

3.5.3 协议的描述和互模拟等价验证举例

我们的目的是通过例子说明这个方法的原理, 因此我们选择最为简单的例子希望能够使得读者理解这个方法的整个应用过程.

例 4. 用 Spi-演算描述下面的协议, 并描述认证和保密性质^[14].

Message1 $B \rightarrow A: N_b$,

Message2 $A \rightarrow B: \{N_a, N_b, M\}_{K_{ab}}$.

在 Message1 中, B 用新鲜值 N_b 挑战 A . Message2 中 A 通过 K_{ab} 加密 M 后传输给 B . N_b 的出现证明 Message 2 的新鲜性. 而 N_a 则保证密文的差异性. 这个协议可以表示如下:

$$A(N_a, M) \stackrel{\text{def}}{=} c_a(x).c_b \langle \{N_a, x, M\}_{K_{ab}} \rangle,$$

$$B(N_b, F) \stackrel{\text{def}}{=} \bar{c}_a \langle N_b \rangle | c_b(x). \text{case } x \text{ of } \{y\}_{K_{ab}} \text{ in}$$

$$\text{let } (y_1, y_2, y_3) = y \text{ in } [y_2 \text{ is } N_b] F(y_3),$$

$$Sys(M_1, M_2, \dots, M_m, F) \stackrel{\text{def}}{=} (\nu K_{ab}) \left(\left(\prod_{k \in 1..m} (\nu N_a) A(N_a, M_k) \right) | (\nu N_b) B(N_b, F) \right).$$

信道 c_a 和 c_b 分别表示 A 和 B 的输入信道. 参数 F 是一个抽象, 使得 $F(M)$ 是 B 成功接收到消息 M 后的行为. 这里要求公式中的约束变元 (包括 K_{ab}) 在 F 中不自由出现. 进程 $Sys(M_1, M_2, \dots, M_m, F)$ 描述的是 A 发送消息 M_1, M_2, \dots, M_m 给 B , 而 B 应用 F 到所接收的消息上的一个完整的系统. 其中的重复表示 B 将可以进行任意多次的传输.

认证属性.

直观上, 协议具有认证性质: 如果 B 接收一个消息, 那么它是 A 所发送的. 而且 B 不能接收重放消息. 换句话说, 这个性质是: 如果 $Sys(M_1, M_2, \dots, M_m, F)$ 促使 $F(L_1), F(L_2), \dots, F(L_l)$ 运行, 那么, $\{L_1, L_2, \dots, L_l\}$ 是 $\{M_1, M_2, \dots, M_m\}$ 的子集.

命题 1 (认证属性). 对于任意数 m 以及不同的名字 p_1, p_2, \dots, p_m , 有一个闭进程 Q , 使得 $fn(Q) \subseteq \{p_1, p_2, \dots, p_m, c_a, c_b\}$, 满足对于所有闭项 M_1, M_2, \dots, M_m 和所有闭抽象 F , 如果 $p_1, p_2, \dots, p_m \notin fn(Sys(M_1, M_2, \dots, M_m, F))$, 则

$$Sys(M_1, M_2, \dots, M_m, F) \simeq (\nu p_k^{k \in 1..m}) \left(Q \left| \prod_{k \in 1..m} p_k(x).F(M_k) \right. \right).$$

保密性.

这个协议还有保密性质: 如果没有关于接收到的消息的消息, 则外部的观察者无法鉴别这些消息. 但是可以知道有多少消息被传输了.

命题 2 (保密性). 对于任意多对闭项 $(M_1, M'_1), \dots, (M_m, M'_m)$ 以及任意的抽象 F , 如果对于每个 $i \in 1..m$, 有 $F(M_i) \simeq F(M'_i)$, 则 $Sys(M_1, M_2, \dots, M_m, F) \simeq Sys(M'_1, M'_2, \dots, M'_m, F)$.

通过这种检测等价的表示, 秘密性和认证性的验证就成为测试等价的计算问题. 利用互模拟关系是计算测试等价的一个方便的手法.

例 5. 利用互模拟方法证明进程

$$!(\nu K)c\langle\{M\}_k\rangle$$

和进程

$$(\nu K)!(\nu K')c\langle\{K',M\}_M\rangle$$

是测试等价的。

前一个进程重复产生密钥并发送用这些密钥解密的消息 M 。后一个进程产生一个固定的密钥,然后重复地产生名字,并不停地发送用这个密钥加密名字和 M 的密文。

对于这个例子,设 S 是满足 $(fr, th) \vdash PSQ$ 的最小关系,这里

$$fr = \{c, n\};$$

th 是任意有限集对 $(\{M\}_k, \{k'_i, M\}_k)$, 每个对中的名字 k_i 和 k'_i 互不相同,但是具有相同的 M 和名字 k , 且 $k-i, k \notin fr$;

P 是 $!(\nu K)c\langle\{M\}_k\rangle$ 与任意多个 $(\nu K)c\langle\{M\}_k\rangle$ 和任意多个 0 的平行复合;

当 th 是空集时, Q 是 $(\nu K)!(\nu K')c\langle\{K', M\}_k\rangle$ 与任意多个 0 的复合; 如果 th 含有对 $(\{M\}_{k_i}, \{k'_i, M\}_k)$, 则 Q 是 $!(\nu K')c\langle\{K', M\}_k\rangle$ 与任意多个 $(\nu K')c\langle\{K', M\}_k\rangle$ 和任意多个 0 的复合;

可以验证 S 是一个互模拟关系。

3.6 概率互模拟和模式复合方法

本小节对于概率互模拟的方法以及模式复合的方法作简单的介绍。

3.6.1 概率互模拟模型——BPW 密码库方法

基于人们对于建立形式化安全协议分析方法和密码学的协议分析方法之间联系的强烈愿望, BPW^[26] 三个作者利用了与众不同的方法。他们不是企图构造密码学可靠的形式化方法的模型,而是结合两种方法的优势。他们规范了一个抽象的密码库 (crypto-library), 可以进行复合运算, 定义密码实现, 并且证明这个抽象在任意交互环境下对于主动攻击是可靠的。密码库目前包括公钥加密、签名、时鲜值、列表以及应用数据等。其证明过程是一个新颖的、概率的不完全互模拟与密码归约和静态信息流分析的组合。

他们的工作第一次证明了, 由 Dolev-Yao 模型的原语组成的抽象库在任意攻击的情形下, 在任何安全协议中, 对于所有安全属性, 通过密码学的手段的安全实现过程。这些属性包括秘密性、认证属性以及其它的完整属性等。这些原语在现实中由密码学安全的公钥加密算法和签名系统, 通过附加的运算——加标签和随机性进行增强。这些附加的运算并不影响效率。因而对于协议的设计来说, 运用抽象

的方法并没有效率方面的问题。

为了刻画主动攻击, 他们使用了一个交互的环境。他们的 Dolev-Yao 模型不仅仅是一个代数, 而且是一个状态性的系统, 根据事先的密码运算和网络动作, 具有可能的诚实用户和敌手的动作。

这个方法可以形象地比喻为拼图法。假设我们的理想模型已经设计完毕。这个理想的系统是在 Dolev-Yao 的模型假设下完成的, 可以容易地验证其安全性。为了在现实中实现这个系统 (协议), 就必须对于理想的加密、时鲜值、签名等等方案用现实的、被证明为密码安全的系统来代替。就像要换掉一个拼图中的一些块。这就要对于实际的算法或系统进行一个评估。这种评估宜用: “实际密码库与抽象的库是‘同样好’”这个概念。这个概念是通过模拟而给出的。就是说, 对于实际用户, 可能发生的干扰、截断或篡改消息、调度错乱等等一切都可以在理想 (抽象) 的模型中出现。他们证明的复合和保持性定理保证了所需要的结果。因此, 如果用抽象的密码库设计一个协议, 然后插入实际密码库的系统, 得到的协议在模拟意义下与抽象的协议同样好。而且, 如果对于抽象的协议证明了某些安全性质, 对于实际的协议也是成立的。

3.6.2 安全协议分析的 PPT-演算

斯坦福大学的 Mitchell 等人一直致力于用概率多项式时间的进程运算进行安全协议的分析^[28]。这种方法的特点是, 其结构与基于计算复杂性的密码学机制非常协调, 或者说这个系统构成的思路是沿用计算密码学的思想设计而成的。作为一个长期的项目, 其目标是开发一种与标准密码学假设相吻合的安全协议分析的形式, 提供表达概率多项式时间协议步骤的语言, 发展一种关于等价的可复合的描述手段以及建立关于等价的推理方面的逻辑基础。

他们所选择的进程运算是 CCS^[45] 的一个变种。其中在消息和布尔检测中使用有界的重复运算和概率多项式时间的表达式。为了达到安全与不确定性的协调, 消息的次序是通过概率的方式确定的, 并非以“不确定”的方式进行。

他们证明了可以在概率多项式时间内计算这样的进程。定义了一个渐近的协议等价形式, 使得安全性质可以通过观察等价来表达。他们还发展了一种概率互模拟, 用于建立基于观察等价的等式, 证明系统的可靠性。

4 最近的结果以及发展趋势

密码分析的方法是随着应用的发展而需要不断发展的. 在这个通信日益发达, 人们的生活因而相互依赖的星球上, 在我们的数据越来越多地依赖于计算机网络进行传输的时代, 一些新的密码手段被发明和应用, 安全协议的应用范围不断扩大并且它们的复杂性不断加剧. 新的漏洞和新的攻击手段和变化层出不穷. 我们的安全协议的分析技术也在与时俱进.

(1) 实用性、极大的适应性与复杂性

随着网络应用的扩展, 安全协议的应用也更加广泛. 安全协议分析的技术现状远远无法满足对于实用安全协议的分析要求. 同时, 不同的平台和系统的应用增加了网络的复杂性, 也使得安全协议越来越复杂. 一个协议中可以包括多种的加密解密的算法, 也可以由不同目的的协议共同构成一个大的具有一个总目标的协议. 这就要求协议的设计、实现和验证几个方面都要有更为鲁棒的、能够极大地适应这种复杂性的技术方法. 目前的形式化方法大多针对一个或者两个目标而设计, 难以适应日益庞大的应用协议的要求.

(2) 对于密码原语的适应性

过去认为, 我们在安全协议中只使用保守的、最为安全的算法. 但是在实际应用中, 实现安全协议的算法具有我们没有考虑的性质. 如加密解密的可交换性、同态性质、无界性等等. 这些性质都有可能对于协议的安全造成事先没有验证的漏洞和攻击.

另外, 现代密码学利用各种手段证明一些系统的安全性, 是否可以在形式化系统中继承这些安全性质, 以保证或简化验证过程和技术的可靠性, 是一个值得探讨的新问题.

(3) 应对新的安全威胁

对于早期安全协议的分析, 大多都针对清晰的攻击手段, 如伪装、截断、腐蚀参与者等等. 现在必须考虑随意性的攻击手段, 如, 造成拒绝服务的攻击. 另外, 在电子商务协议中, 参与者的利益有时是由时间保证的. 这时, 协议不仅要保证安全性, 还要考虑系统的活性和公正性. 对于这类协议的验证, 敌手的模型刻画和安全属性描述都是不同于 Dolev-Yao 模型的技术.

(4) 密码学可靠性

以前的形式化分析方法以 Dolev-Yao 模型为

基础. 它们将密码算法抽象为黑盒子, 进行模型检测或者证明分析. 这个方法越来越受到质疑. 比如, 对于加密模式的选择和已知明文的分析曾发现了新的问题^[63]. 人们已经在扩展形式化方法, 以囊括更为广泛的密码算法. 逻辑方面有文献^[64, 65]; 模型检测方面有文献^[66]. 在将理论密码学提供的安全性证明与形式化方法集合的方法中, Abadi 和 Rogaway^[67]的工作是先驱性工作, 还有文献^[39]、BPW^[26]等等. 人们希望得到一种既可以进行形式化分析, 又具有密码学可靠性的方法. 这个方面是目前形式化方法研究的热点^[22, 24, 25].

(5) 可复合性

可复合性问题是目前安全协议分析研究的另一个热点问题. 主要原因是, 协议的应用范围不断扩大, 它们所承载的任务越来越庞大, 因而协议的构成也越来越复杂. 这就给设计和验证工作带来了极大的问题. 目前看来, 现有的协议验证工具和技术难以适应大型协议的评估工作. 人们期望通过利用小型的、可以验证安全的协议, 构建大型的协议, 以满足实际需要. 从另外一个方面, 网络环境中越来越多的任务要求同时运行多种多类的协议, 这就要求每个独立的协议运行时, 不威胁其它协议的安全性. 由此, 协议的可复合安全性就成为一个自然需要解决的问题.

大多数协议分析者都注意到复合性的问题, 其中以 BPW^[26]的方法, Mitchell^[28]的方法和 Canetti^[44]的方法为代表.

BPW 的方法是利用安全归约的思想, 将一个协议的某些系统(子协议)的实现与抽象的理想模块进行归约比较, 通过保持性定理保证可复合性. 这一点像 Canetti 方法中模拟器的作用. 而 Mitchell 的方法是利用进程运算中的 context 概念作为进程运行的外部环境, 通过进程在任何环境下, 对于任何其它进程的不可区分性表达协议复合的安全性. 这一点与 Canetti 模式方法中 \mathcal{Z} 环境模拟器使用有着异曲同工之妙.

Canetti^[44]给出的关于安全协议的框架可以用一般的方法描述多方协议的复合. 这个框架对于不同的协议用相同的方法处理安全的概念. 其中协议与相关的理想协议进行比较, 如果实际协议可以在任何环境对于任何攻击者都有与理想协议同样的“行为”时, 就认为这是协议的一个安全的实现. 这里的协议运行环境利用一个环境机器 \mathcal{Z} 模拟. 同时, 这个环境机器还担任区分器的作用. 如果没有哪个环

境机器能够区分具有实际攻击者的实际协议的运行和一个具有实际攻击者模拟器的理想版本的运行,那么,这个协议就安全地实现了一个理想的功能(属性). Canetti 等人证明了这个安全的刻画具有一定的可复合性,并在此基础上开展了许多的工作^[68~70].

但是由于可复合性是一个复杂的问题,如果进一步考虑到并发运行的复杂性,安全协议的可复合性问题远远没有得到刻画和解决,有大量的工作有待完成.

(6)最近的结果

我们在这里指出几个值得注目的结果,这些结果有理由被认为对于安全协议形式化分析的发展将起到一定的作用.

在文献^[25]中 Impagliazzo 和 Kapron 提出了两个逻辑系统以推理密码学的构造. 它们关于标准密码学的安全定义是可靠的. 并且给出例子说明任何利用这些系统去证明基本密码构造的行为的正确性. 在证明系统的可靠性时,论文利用了算术的非标准分析的技术手段. 这两个关于安全的逻辑是现代密码学的方法和语言的简化形式.

要提出一个关于安全的形式系统,使其具有密码学可靠性,并且能够具有强有力的表达能力,而且足够简单有用,是一个极大的挑战. 其困难包括:安全定义的形式描述;安全定义中概率的使用;随机选择和随机分布的推理;量化敌手的计算能力以及归纳法的直观而危险的错误使用等等.

该文的结果表明,形成一个蕴含地处理这些难点的逻辑系统是可能的. 其中糅合了密码学、蕴含复杂性以及计算复杂性的思想;用项代数表达可行性函数,量化隐式定义敌手能力的函数,通过增加计数量词使得简单的概率推理成为可能;限制了归纳的范围;通过消去渐近的和显然的资源界,使得结果的证明极大地无量词化,由等式和不等式构成;逻辑的可靠性是标准意义上的,并且使用算术的非标准模型得以证明. 这个逻辑是一般的框架,可以用于构建简单的有限目的的逻辑系统. 作者给出了一个简单的关于计算不可区分推理的可靠的逻辑.

在文献^[22,24]中 Micciancio 和 Warinschi 仔细地研究了 Abadi 和 Rogaway^[67]的逻辑,得出原逻辑的不完备性,并给出完备的条件. 在此基础上,作者进一步给出了在主动敌手存在的情形下证明安全协议安全性质的一般方法. 这个方法允许使用简单的逻辑表达并证明安全性质. 作者证明了这个方法

的可靠性,即逻辑命题可以自然地在计算的环境中解释:如果这个命题在 Dolev-Yao 模型的敌手存在的情形下协议抽象运行时成立,那么它的计算解释在标准计算模型中,在概率多项式时间的敌手存在的情形下也成立.

主要结果是给出了两个模型:一个是具体的模型,其中协议利用任何加密方案(满足选择密文攻击不可区分的安全概念)实现,在具有多项式时间的敌手的环境中运行. 另一个是抽象模型,协议以符号的形式运行. 证明过程显示了 Dolev-Yao 模型的敌手与概率多项式时间的敌手之间的对应关系. 从而提供了一般的设计和评估密码意义下安全协议的方法.

群组协议是由一组主体为了得到某个目标而采取的一系列步骤. 这类协议中最为典型的的就是密钥协商协议. Pereira 和 Quisquater 在文献^[71]中分析了一族群组密钥协商协议. 作者构造了一个简单的分析模型,发现了 Cliques 群组密钥协商协议的一些漏洞. 而且,作者最近用串空间模型构造了一般的群组协议的模型^[72],有着重要的意义. 这是对于群组协议分析成功的为数不多的案例之一.

参 考 文 献

- 1 Menezes A., van Oorschot P., Vanstone S.. Handbook of Applied Cryptography. New York: CRC Press, 1996
- 2 Needham R. M., Schroeder M. D.. Using encryption for authentication in large networks of computers. Communications of the ACM, 1978, 21(12): 993~999
- 3 Dolev D., Yao A. C.. On the security of public key protocols. In: Proceedings of the IEEE 22nd Annual Symposium on Foundations of Computer Science, Nashville, TN, 1981, 350~357
- 4 Dolev D., Even S., Karp R.. On the security of ping-pong protocols. Information and Control, 1982, 55(1): 57~68
- 5 Even S., Goldreich O.. On the security of multi-party ping-pong protocols. In: Proceedings of the 24th IEEE Symposium on the Foundations of Computer Science, Los Alamitos, 1983, 34~39
- 6 Millen J. K., Clark S. C., Freedman S. B.. The Interrogator: Protocol security analysis. IEEE Transactions on Software Engineering, 1987, SE-13(2): 274~288
- 7 Kemmerer R. A.. Analyzing encryption protocols using formal verification techniques. IEEE Journal of Selected Areas in Communications, 1989, 7(4): 448~457
- 8 Kemmerer R. A.. Analyzing encryption protocols using formal verification techniques. In: Carl Pomerance ed.. Advances in Cryptology—CRYPTO'87. Lecture Notes in Computer Science 293. Springer-Verlag, 1988, 289~305

- 9 Burrows M. , Abadi M. , Needham R. . A logic of authentication. *ACM Transactions in Computer Systems*, 1990, 8(1): 18~36
- 10 Vardi M. Y. . Why is modal logic so robustly decidable? In: Immerman N. , Kolaitis Ph. G. eds. . *Descriptive Complexity and Finite Models*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. New York: AMS, 1997, 149~184
- 11 Lowe G. . Breaking and fixing the Needham-Schroeder public key protocol using FDR. In: *Proceedings of TACAS. Lecture Notes in Computer Science 1055*. Passau, Germany: Springer-Verlag, 1996, 147~166
- 12 Abadi M. . Secrecy by typing in security protocols. *Journal of the ACM*, 1999, 46(5): 749~786
- 13 Gordon A. , Jeffrey A. . Authenticity by typing in security protocols. In: *Proceedings of the 14th IEEE Computer Security Foundations Workshop*, Nova Scotia, Canada, 2001, 145 ~ 159
- 14 Abadi M. , Gordon A. D. . A calculus for cryptographic protocols: The Spi calculus. *Information and Computation*, 1999, 148(1): 1~70
- 15 Paulson L. C. . The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 1998, 6(1): 85~128
- 16 Song Dawn Xiaodong, Berezin Sergey, Perrig Adrian. Athena: A novel approach to efficient automatic security protocol analysis. *Journal of Computer Security*, 2001, 9(1/2): 47~74
- 17 Bellare M. , Rogaway P. . Entity authentication and key distribution. In: *Proceedings of CRYPTO'93*, California, USA, 1994, 232~249
- 18 Koblitz N. , Menezes A. . Another look at "provable security". University of Waterloo, Waterloo, Canada: Technical Report CORR 2004-20, 2004
- 19 Bellare M. , Canetti R. , Krawczyk H. . A modular approach to the design and analysis of authentication and key exchange protocols. In: *Proceedings of the 30th Annual Symposium on the Theory of Computing*, ACM, Dallas, TX, 1998, 419~428
- 20 Canetti R. , Krawczyk H. . Analysis of key-exchange protocols and their use for building secure channels. In: Pfitzmann ed. *Proceedings of Eurocrypt'01. Lecture Notes in Computer Science 2045*. Springer-Verlag, 2001, 453~474
- 21 Abadi M. , Rogaway P. . Reconciling two views of cryptography: The computational soundness of formal encryption. In: *Proceedings of the 1st IFIP International Conference on Theoretical Computer Science. Lecture Notes in Computer Science 1872*. Springer, 2000, 3~22
- 22 Micciancio D. , Warinschi B. . Completeness theorems for the Abadi-Rogaway logic of encrypted expressions. *Journal of Computer Security*, 2004, 12(1): 99~129
- 23 Gligor V. , Horvitz D. O. . Weak key authenticity and the computational completeness of formal encryption. In: Boneh D. ed. . *Proceedings of CRYPTO 2003. Lecture Notes in Computer Science 2729*. Springer-Verlag, 2003, 530~547
- 24 Micciancio D. , Warinschi B. . Soundness of formal encryption in the presence of active adversaries. In: *Proceedings of the Theory of Cryptography Conference (TCC)*, Cambridge, Massachusetts, 2004, 133~151
- 25 Impagliazzo R. , Kapron B. . Logics for reasoning about cryptographic constructions. In: *Proceedings of STOC'03*, San Diego, California, 2003, 372~383
- 26 Backes M. , Pfitzmann B. , Waidner M. . A universally composable cryptographic library. In: *Proceedings of the 10th ACM Conference on Computer and Communications Security*, Washington, DC, 2003, 220~230
- 27 Maneki A. P. . Honest functions and their application to the analysis of cryptographic protocols. In: *Proceedings of the 12th IEEE Computer Security Foundations Workshop (CSFW 1999)*, Mordano, Italy, 1999, 83~89
- 28 Mitchell J. , Ramanathan A. , Scedrov A. , Teague V. . A probabilistic polynomial-time calculus for analysis of cryptographic protocols (Preliminary report). In: *Proceedings of the 17th Annual Conference on the Mathematical Foundations of Programming Semantics*, Arhus, Denmark, 2001, 45
- 29 Comon H. , Shmatikov V. . Is it possible to decide whether a cryptographic protocol is secure or not? *Journal of Telecommunications and Information Technology, Special Issue on Cryptographic Protocol Verification (Goubault-Larrecq J. ed.)*, 2002, 4(1): 5~15
- 30 Hoare C. A. R. . *Communicating Sequential Processes*. London: Prentice-Hall International, 1985
- 31 Roscoe A. W. . Model-checking CSP. In: Roscoe A. W. ed. . *A Classical Mind: Essays in Honor of C. A. R. Hoare*. Prentice-Hall, 1994
- 32 Roscoe A. W. . Modelling and verifying key-exchange protocols using CSP and FDR. In: *Proceedings of the 8th IEEE Computer Security Foundations Workshop*, County Kerry, Ireland, 1995, 98~107
- 33 Schneier B. . *Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd Edition*. New York: John Wiley & Sons, 1996
- 34 Cervesato I. , Durgin N. , Lincoln P. , Mitchell J. , Scedrov A. . A metanotation for protocol analysis. In: *Proceedings of the 12th IEEE Computer Security Foundations Workshop*, Mordano, Italy, 1999, 55~69
- 35 Cervesato I. , Durgin N. , Lincoln P. D. , Mitchell J. C. , Scedrov A. . Relating strands and multiset rewriting for security protocol analysis. In: *Proceedings of the 13th IEEE Computer Security Foundations Workshop*, Cambridge, England, 2000, 35~51
- 36 Fabrega F. J. T. , Hertzog J. , Guttman J. . Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 1999, 7(2/3): 191~230
- 37 Milner R. . Functions as processes. *Mathematical Structures in Computer Science*, 1992, 2(2): 119~141

- 38 Lincoln P. D., Mitchell J. C., Mitchell M., Scedrov A.. A probabilistic poly-time framework for protocol analysis. In: Proceedings of Computer and Communications Security—CCS'98, San Francisco, California, USA, 1998, 112~121
- 39 Lincoln P., Mitchell J., Mitchell M., Scedrov A.. Probabilistic polynomial-time equivalence and security analysis. In: Wing J., Woodcock J., Davies J. eds.. Proceedings of FM'99—Formal Methods. Lecture Notes in Computer Science 1709. Springer-Verlag, 1999, 776~793
- 40 Pfitzmann B., Waidner M.. Composition and integrity preservation of secure reactive systems. In: Proceedings of the 7th ACM Conference on Computer and Communications Security, Athens, Greece, 2000, 245~254
- 41 Pfitzmann B., Waidner M.. A model for asynchronous reactive systems and its application to secure message transmission. In: Proceedings of the 22nd IEEE Symposium on Security & Privacy, Oakland, California, 2001, 184~200
- 42 Canetti R.. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 2000, 3(1): 143~202
- 43 Meadows C.. The NRL protocol analyzer: An overview. *Journal of Logic Programming*, 1996, 26(2): 113~131
- 44 Canetti R.. Universally composable security: A new paradigm for cryptographic protocols. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS), 2001, 136~145
- 45 Milner R.. A calculus of communication systems. *Lecture Notes in Computer Science* 92. New York: Springer-Verlag, 1980
- 46 Gong L., Needham R., Yahalom R.. Reasoning about belief in cryptographic protocols. In: Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Los Alamitos, California, 1990, 234~248
- 47 Abadi M., Tuttle M.R.. A Semantics for a logic of authentication. In: Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing, Montreal, Quebec, Canada, 1991, 201~216
- 48 Syverson P. F., van Oorschot P. C.. On unifying some cryptographic protocol logics. In: Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Washington, DC, 1994, 14~28
- 49 Kailar R.. Accountability in electronic commerce protocols. *IEEE Transactions on Software Engineering*, 1996, 22(5): 313~328
- 50 Coffey T., Saidha P.. Logic for verifying public key cryptographic protocols. *IEE Journal of Computers and Digital Techniques*, 1997, 144(1): 28~32
- 51 Kailar R., Gligor V. D., Gong L.. On the security effectiveness of cryptographic protocols. In: Proceedings of the 4th International Working Conference on Dependable Computing for Critical Applications, San Diego, California, 1994, 139~157
- 52 Rubin A. D., Honeyman P.. Nonmonotonic cryptographic protocols. In: Proceedings of IEEE Computer Security Foundations Workshop VII, New Hampshire, 1994, 100~116
- 53 Brackin S. H.. A HOL extension of GNY for automatically analyzing cryptographic protocols. In: Proceedings of Computer Security Foundations Workshop, 1996
- 54 Moser L.. A logic of knowledge and belief for reasoning about computer security. In: Proceedings of the Computer Security Foundations Workshop II, Washington, DC, 1989, 57~63
- 55 Bieber P.. A logic of communication in a hostile environment. In: Proceedings of the Computer Security Foundations Workshop III, Los Alamitos, California, 1990, 14~22
- 56 Kessler V., Wedel G.. AUTLOG: An advanced logic of authentication. In: Proceedings of the Computer Security Foundations Workshop VII, New Hampshire, 1994, 90~99
- 57 Woo T., Lam S.. A semantic model for authentication protocols. In: Proceedings of the 1993 Symposium on Research in Security and Privacy, Oakland, California, 1993, 178~194
- 58 Xue R., Feng D.. A new semantic model for authentication protocols in ASMs. *Journal of Computer Science and Technology*, 2004, 19(4): 555~563
- 59 Merritt M. J.. Cryptographic protocols [Ph. D. dissertation]. Georgia Institute of Technology, Atlanta, USA, 1983
- 60 Toussaint M.. Verification of cryptographic protocols [Ph. D. dissertation]. Universite de Liege, Belgium, 1991
- 61 Ryan P., Schneider S.. Modelling and Analysis of Security Protocols. Harlow, England: Addison-Wesley, 2001
- 62 Milner R., Parrow J., Walker D.. A calculus of mobile processes, Part I and II. *Information and Computation*, 1992, 100(1): 1~40, 100(1): 41~77
- 63 Stubblebine S., Gligor V.. On message integrity in cryptographic protocols. In: Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, 1992, 85~104
- 64 van Oorschot P.. Extending cryptographic logics of belief to key agreement protocols (extended abstract). In: Proceedings of the 1st ACM Conference on Computer and Communications Security, ACM, Fairfax, Virginia, 1993, 232~243
- 65 Syverson P., van Oorschot P.. On unifying some cryptographic protocol logics. In: Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Washington, DC, 1994, 14~28
- 66 Pavlovic D., Smith D. R.. Guarded transitions in evolving specifications. In: Hélène Kirchner, Christophe Ringeissen eds.. Proceedings of the 9th International Conference on Algebraic Methodology and Software Technology Source. *Lecture Notes In Computer Science* 2422. London, UK: Springer-Verlag, 2002, 411~425
- 67 Abadi M., Rogaway P.. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 2002, 5(2): 103~127
- 68 Canetti R., Fischlin M.. Universally composable commitments. In: Joe Kilian ed.. *Advances in Cryptology—Proceed-*

- ings of CRYPTO 2001. Lecture Notes in Computer Science 2139. Springer-Verlag, 2001, 19~40
- 69 Damgård I., Nielsen J. B.. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: Moti Yung eds., Advances in Cryptology—Proceedings of CRYPTO 2002. Lecture Notes in Computer Science 2442. Springer-Verlag, 2002, 581~596
- 70 Canetti R., Lindell Y., Ostrovsky R., Sahai A.. Universally composable two-party and multi-party secure computation. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, Montréal, Québec, Canada, 2002, 494~503
- 71 Pereira O., Quisquater J.. A security analysis of the cliques protocols suites. In: Proceedings of the 14th IEEE Computer Security Foundations Workshop, Cape Breton, Nova Scotia, Canada, 2001, 73~81
- 72 Pereira O., Quisquater J.. Generic insecurity of cliques-type authenticated group key agreement protocols. In: Proceedings of the 17th IEEE Computer Security Foundations Workshop, Pacific Grove, CA, USA, 2004, 16~29



XUE Rui, Ph. D., associate professor. His research interests include cryptography and network security, especially in designing and analysis of security protocols, application of formal methods for verification of cryptographic construction.

FENG Deng-Guo, professor and Ph. D. supervisor. He mainly engaged in the research and development of information and network security.

Background

This work is supported by the National Natural Science Foundation of China (grant No. 60373048).

This project is managed to make up the incompleteness and unreliability in the current formal models for analysis of cryptographic protocols, and to complement hardness in operation and intricacy of provable security. To find and design a more advanced formal model for analysis of protocols in which ingredients and concepts of computational complexity should be borrowed so as to bridge the analysis approach of provable security and formal one. It will constitute of three

phases. The first phase is to investigate current formal models and find out a succinct and powerful enough model with flexible extensibility so that it is used as the potential candidate. The second phase lies at explorations how to relax the requirement during the implementation of real protocols in the ideal model. The last phase will tend to combine the precedent results to form a securely, easy to operate model for analysis of cryptographic protocols. The results would be of helpful to protocol designers and motivated further development.