安全协议的形式化描述和分析*)

宋 震[1] 张 艳[2] 李舟军[1,3] 陈火旺[1]

(国防科技大学计算机学院 长沙410073)[1] (四川大学数学学院 成都610064)[2]

(武汉大学软件工程国家重点实验室 武汉430072)[3]

Formal Description and Analysis of Security Protocols

SONG Zhen[1] ZHANG Yan[1,2] LI Zhou-Jun[1,3] CHEN Huo-Wang[1]

(Computer School, National University of Defence Technology, Changsha 410073)[1]

(Mathematics School, Sichuan University, Chengdu 610064)[2]

(State Key Laboratory for Software Engineering, Wuhan University, Wuhan 430072)[3]

Abstract Security protocols use cryptography system to complete the tasks of principal identity authentication and seccion key distribution. The correctness of security protocols is of vital importance to ensure the security of the Internet application. Formal methods have been proved to be a valid approach to analyze and verify security protocols. This paper briefly introduces the three main styles in the field of security protocol analysis and their representative work. After that, it points out the future deveopment direction.

Keywords Seciruty protocols, Security, Formal anaylsis, Belief logic, Model checking

1. 引言

随着计算机网络的发展，特别是 Internet 的高速发展，网络逐渐渗透到人类生活的各个领域。然而，在普及和发展的同时，它也对网络环境下的信息安全保护提出了巨大的挑战。特别是在电子商务、网上银行等网络应用出现后，网络安全问题更成为影响计算机深层次应用的一大因素。如何在一个无法确定的操作环境下，保证计算机间传送信息的安全性，从而确保通信双方主体之间的“信任”以及通信数据的秘密和完整，已成为许多网上应用的核心问题之一。

计算机通信的核心问题是协议。所谓协议是指两个或两个以上的参与者为完成某项特定的任务而采取的一系列步骤。在进行通信时，通信协议的协议数据单元将传递控制信息(包括通信双方对通信过程一些选项与参数的设置信息等)与数据信息(即通信数据)。通信协议不仅要能够完成自身的通信工作，同时它还必须具备足够的安全性，即保证秘密通信数据的安全与完整。安全协议即是利用密码系统中完成主体身份认证和会话密钥分发等任务的具有安全性功能的协议。因此，安全协议是安全通信的基础，协议本身的正确性和安全性至关重要，直接关系到整个系统的安全。

但是，目前的实际情况却表明：安全协议的正确性或者说安全性却不容乐观。即使我们不考虑与实现相关的协议缺陷和密码系统被成功破译的可能性，安全协议也会存在一些隐藏很深的漏洞和直观上难以发现的攻击序列。目前已经发表甚至投入实用的安全协议不少，但许多协议在发表甚至实用几年或十几年后，却被发现存在安全漏洞。Needham-Schroeder 协议即是一个典型的例子，在该协议发表近3年后，Denning 和 Sacco 发现该协议存在安全缺陷[17]：当主动攻击者用一个已经破译的旧会话密钥冒充新的会话密钥时，攻击成功；另外，在其它一些情况下该协议也可能失败。然而，该协议被研究过相当一段时间且已实际应用于网络通信中，但在此期间却一直没有发现上述漏洞。

使用具有安全漏洞的安全协议将无法保证网络通信的安全。造成安全协议失败的主要原因包括：缺乏安全协议设计的形式化准则，缺乏对安全性的精确定义以及严格的验证手段等。其中，对安全协议的分析和验证是保证所设计的协议满足安全性质及应用环境要求的重要技术手段。

在采用形式化方法进行安全协议的分析与验证之前，对安全协议的分析一般采用了攻击检验方法。它是根据已知的各种攻击策略来构造对安全协议的攻击，如果发现有效的攻击，那么证明协议将是不安全的。实际上，由于安全协议的复杂性，构造对安全协议的攻击是十分困难的，这一过程很难进行，很难有效地对安全协议进行分析验证。而形式化分析方法则首先将安全协议进行形式化建模，然后借助人工推导或计算机辅助(或自动)分析，判断安全协议的形式模型是否满足所要求的安全性质。

形式化分析方法并非从攻击的角度，而是从协议模型执行的状态转换角度来分析协议系统的状态空间，通过分析其是否满足相应的安全性质，检验协议的安全性。形式化分析方法能够全面、深刻地检测到安全协议中细微的漏洞，有助于发现对安全协议的新的攻击方法，从而便于对协议进行改进和完善，进而可以得到安全协议设计的一般性准则。

在采用形式化分析方法对安全协议进行分析和验证时，一般都采用了如下的“完善加密假设”来处理安全协议所采用的密码系统：

(1)不考虑密码系统被破解的可能(形式化分析方法只研

*)本课题得到国家自然科学基金(90104026)、国家863计划(2002AA144040)和高等学校重点实验室访问学者基金的资助。宋 震 博士研究生，研究方向：信息安全。张 艳 博士研究生，研究方向：信息安全。李舟军 博士，教授，博士生导师。研究方向：计算机软件与理论。陈火旺 中国工程院院士，教授，博士生导师，研究方向：计算机软件与理论。

究协议本身的结构所带来的攻击，对由于密码系统的脆弱性所带来的攻击不予考虑）；

（2）只有在拥有一个密文所对应的解密密钥时，解密过程才能成功进行，并得到对应的明文，即：若$\{\{m\}_K\}_{K'^{-1}}=m$，则$K'=K^{-1}$。其中，$\{m\}_K$表示加密过程，即使用密钥K对明文m进行加密，$\{c\}_{K'^{-1}}$表示解密过程，即使用密钥K'对密文c进行解密，K^{-1}表示加密密钥K所对应的解密密钥。

（3）无密文冲突，即：对于任意明文m，m'及密钥K，K'，如果$\{m\}_K=\{m'\}_{K'}$，那么$m=m'$且$K=K'$。

上述的“完善加密假设”使得整个安全协议的形式化分析独立于具体的密码系统。在“完善加密假设”下具有安全漏洞的密码协议，在实际密码系统中一定存在安全漏洞。

安全协议的形式化分析一般研究安全协议的保密性（Secrecy）与认证性（Authentication）。保密性是指特定的信息（可能包含很多内容，比如通信的内容、密钥等）不会被泄露；而认证性是指在协议运行时，协议的某一参与者可以确认另一参与者的真正身份与其所声称的身份是否一致的。

根据对安全协议进行形式化建模的不同途径[1,2]，目前用于安全协议分析的形式化方法大致可以分为如下三类：基于信念逻辑（beilief logic）的方法，基于状态机（state machine）的方法和基于进程代数（Process Algebra）的方法。下面我们分别对这三种研究方法（流派）逐一加以介绍。

2. 基于信念逻辑的方法

基于信念逻辑的方法从每一个参与协议运行的主体的角度，将其关于协议运行状态的信念进行建模，每一个参与协议运行的主体都将维持一个信念库，根据协议运行过程中该主体所得到的消息并结合已有的知识来维护信念库。在协议运行结束后，如果主体的信念库中含有特定的目标信念（例如，参与协议运行的主体相信某密钥确为二者之间所共享的定义良好的密钥等），那么我们就认为协议是安全的。

从信念逻辑的设计目的来看，基于信念逻辑的方法关心的是协议的认证性，而并不关心安全协议的保密性。

M. Burrows，M. Abadi，R. Needham于1989年提出的基于信念（belief）关系的BAN逻辑[3]是采用信念逻辑方法进行形式化分析和验证的典型代表。

2.1 BAN逻辑简介

BAN逻辑刻画一些直观的命题，比如“A说过X”、“A相信X”、“K是可靠的密钥”和“S是X的仲裁机构”等等，并包含了5个基本的推理规则（BAN逻辑并不对攻击者进行建模，入侵者实际上被隐式地建模于推导规则中），这些规则分别表示了在通信过程中信任关系的建立和传递过程。在协议的运行过程中，根据一些关于协议使用环境的初始假设，BNA逻辑可使用推理规则推导出一些关于主体信念的公式，比如，“通信双方A与B均相信密钥K是A、B之间可靠的密钥”。

在对一个具体的协议进行分析，验证其安全性，BAN逻辑要经过以下几个步骤：

（i）协议理想化：协议的理想化是指由原始协议导出理想化协议，即将协议中的信息转换成相应的BAN逻辑语言表示；

（ii）确定初始假设：确定协议运行的初始信念假设与状态假设；

（iii）逻辑推理：对假设运用推理规则，可依此得到协议参与者所拥有的信念库；

（iv）得出结论：如果通过逻辑推理过程可以证明某些表示信念关系的公式，我们认为BAN逻辑证明该协议是安全的。例如，某两方安全协议的运行双方A与B，如果能够得到如下的信念，即A相信K是A与B之间可靠的密钥；同时，B也相信同样的结论，那么，我们认为该安全协议完成了A与B的密钥协商，它是安全的。

2.2 BAN逻辑的缺点

BAN逻辑推出后，取得了很大的成功，它第一次为密码协议提供了一整套形式化分析方法，成功地找到了密码协议的许多缺陷及攻击。

但BAN逻辑也存在很多的不足之处，大体上来说，对BAN逻辑的大多数批评都认为BAN逻辑的形式化程度还不够，其中主要的批评包括：

（1）语义：BNA逻辑在发表之初并没有一个清晰的语义定义，其描述仍然大量使用了非形式化的描述方法，对BAN逻辑证明的结论没有可靠性定理来保证其正确性；

（2）理想化步骤：虽然BAN逻辑给出了一些协议理想化工作的一般准则（例如，略去明文，一些信息被替换，例如将产生的意在用于进行通信的会话密钥替换为一个BAN逻辑公式，它表示该密钥为A与B之间可靠的密钥），但理想化步骤没有可以严格遵循的形式化规则，是一个非形式化的过程。

（3）初始状态假设：在BAN过程中，初始状态的假设难以确定（而这种假设对分析结果的正确得出至关重要），BAN逻辑中没有形式化的规则来确定假设，无法确认和自动验证假设的正确性和有效性。

2.3 BAN逻辑的发展

为了克服BAN逻辑的局限性，在对BAN逻辑进行大量分析研究的基础上，许多研究人员提出了一些对BAN逻辑的必要改进与扩展（可称为BAN类逻辑）。例如：Gong、Needham、Yahalom提出了GNY逻辑[4]，M. Abadi和M. R. Tuttle提出了AT[5]，Mao和Colin Boyd提出了MB逻辑[13]（特别地，Mao提出了解决协议理想化问题的基于安全协议上下文的理想化规则[12]），P. F. Syverson和P. C. van. Oorschot提出了SvO逻辑[6]，S. H. Brackin提出了一个扩展的GNY逻辑，即BGNY逻辑，以及其改进BGNY2逻辑。

下面分别对其中的代表性工作GNY逻辑、AT逻辑及SvO逻辑进行简要的介绍：

（1）GNY逻辑　GNY逻辑对BAN逻辑中一些不合理的假设进行了修正。例如，GNY逻辑不再假设被加密的信息中包含充分的冗余以识别消息内容，而是采用了“可辨识性（recognizability）”的概念，“主体P可辨识消息X”的含义为消息X符合主体P此刻对所接收消息的期待；同时，GNY逻辑中也增加了关于可辨识性的规则。GNY逻辑还对BAN逻辑中的一些概念进行了细分，例如，GNY逻辑区分了“用户A拥有X”与“用户A相信X”，从而彻底将消息与信念区别开来。

GNY逻辑与BAN逻辑相比更为细致和全面，但它的推导规则急速膨胀到了50个，这妨碍了它的应用与推广。另外，在语义基础、协议理想化步骤方面，GNY逻辑都具有与BAN逻辑相同的缺点。

（2）AT逻辑　AT逻辑参考了GNY逻辑的一些成果，对BAN逻辑进行了一些修订。AT逻辑的主要成果在于为BAN类逻辑建立一个清晰的语义定义。同时，AT逻辑证明了在其定义的语义框架内，逻辑推理具有可靠性，即语义错误

的结论不能通过逻辑推理获得。

(3)SvO 逻辑　SvO 逻辑实际是从 VO 逻辑发展而来，它综合了 GNY 逻辑、AT 逻辑(当然还包括 BAN 逻辑)的优点，具备了清晰的语义(采用了与 AT 逻辑类似的方法，但稍有不同)，修正了 BAN 逻辑的一些缺点，同时在形式上比 GNY 逻辑更为简化。

SvO 逻辑还增加了对密钥协商协议的处理(从 VO 逻辑继承而来)，可以直接用来分析类似于 Diffie-Hellman 等密钥交换协议。

3. 基于状态机的方法

基于状态机的方法将每一个参与协议运行的主体都表示成一个状态机，整个安全协议建模成为状态机的元组。随着协议的运行，根据主体所得到的消息，参与协议运行的主体的状态机发生状态转换，当整个系统的状态机模型出现一条不安全的踪迹或进入某一不安全的状态时，则说明安全协议是不安全的。

大多数采用状态机方法的工作都在某种程度上借鉴了 Dolev 和 Yao 以及 Dolev，Even 和 Karp 的工作。

在 Dolev-Yao 模型[7]中，入侵者完全控制了网络，他能够读取、修改、添加和删除消息；他还能够执行任何可以由合法用户执行的动作，如加密/解密等，但它假设(至少在最初时刻)入侵者并不知道任何属于用户的保密信息，如加密密钥等。入侵者有效地把被攻击的系统当作一台产生单词(消息)的状态机。单词遵循一些重写规则，例如基于对称加密特性的重写规则。入侵者的任务是发现秘密的单词。因此，协议的安全问题被转换为基于项重写系统的搜索问题。Dolev 和 Yao 给出了一个算法来判断协议在他们的模型中是否安全。

然而，Dolev-Yao 模型的描述能力十分有限，只考虑了保密性，并且协议参与者不能保存从一个状态到下一个状态的状态信息。在其后的许多检验方法中，都只把 Dolev-Yao 模型作为基础，扩充了合适的协议建模技术来描述协议参与者的行为。

Woo 和 Lam[8,9]提出了认证协议的一个直观的模型。在此模型中，协议的每一个主体都建模成一个状态机模型。协议分析者很容易把协议转化为基于该模型的刻画。通过研究，他们认为保密性(Secrecy)与对应性(Correspondence，对应性也可认为是认证性的另一种定义)是认证协议最根本的性质。保密性描述为一些消息(一般是密钥)攻击者无法得到。对应性的一般形式为，如果主体 A 完成了与 B 的协议，则主体 B 必定参与与 A 的协议。他们还提出一套关于对应性的推理规则，但并没有给出自动推理工具来支持协议分析。

4. 基于进程代数的方法

进程代数是一类并发计算框架的总称，它包括 CCS(通信系统演算，Calculus of Communication System)、CSP(通信顺序进程，Communicating Sequential Processes)及 π-演算等。进程代数采用了代数方法研究并发分布式系统的行为，是研究并发系统的重要途径之一。20世纪80年代以来，进程代数因其概念简单，可用的数学理论与工具丰富，在并发系统的规范、分析、设计和验证等方面获得了广泛的研究与应用。

使用进程代数对安全协议分析和验证时，协议的每一个主体都被建模为一个单独的进程(在一些研究方法中，攻击者也被建模成为一个单独的进程)，这些子进程并发运行，并使用进程之间的共享通道进行同步通信，这样得到的并发系统将作为安全协议的基本模型。

利用进程代数的方法对安全协议进行分析和验证时，借鉴并使用了进程代数基本的验证理论与方法。因此，基于进程代数安全协议的验证技术也可分为模型检测技术与互模拟验证技术。

4.1　模型检测技术

模型检测技术是一种验证有限状态系统的自动化分析技术，它对协议的自动验证和协议的工程化设计具有重要意义。

使用模型检测技术进行验证时，对象系统将使用某种建模语言被建模成(有限)状态转换图(称为实现，即 Implementation)。同时，该对象系统需要满足的性质将使用某种规范语言进行描述(一般采用某种时态逻辑约束，称为规范，即 Specification)，模型检测技术将使用搜索算法来确定系统实现的状态转换图是否满足规范描述。模型检测技术需要搜索整个状态空间，因此，在状态急剧增大时，模型检测技术的效率也将下降。

使用模型检测技术验证安全协议方面的代表性工作包括 G. Lowe[10,14~16]等的研究。

G. Lowe 将安全协议的参与者看作是并发的 CSP 进程，攻击者也被建模为具有多种攻击操作能力(例如窃听、冒充、重放等)的进程。并发的参与者进程与攻击者进程共同组成了安全协议的系统实现描述。

G. Lowe 同样也将协议的安全性质(规范)描述为 CSP 进程。例如：保密性可以描述为踪迹中不泄露有关保密信息的进程；而认证性的定义则采用了与 Woo 和 Lam 所提出的对应性类似的"一致性"定义，它被描述为如下的 CSP 进程：当在其踪迹中出现表示主体 B 作为响应者完成与主体 A 的一次协议运行的事件时，在此事件之前必定出现表示 A(作为协议发起者)完成与 B 的一次协议运行的事件。

G. Lowe 使用了基于 CSP 的通用模型检测工具 FDR(故障偏差精炼检测器，Failures Divergences Refinement Checker)。FDR 接受两个 CSP 进程，其中一个为安全协议的 CSP 描述(作为实现)，另外一个则为描述相应安全性质的 CSP 进程(作为规范)。FDR 将通过枚举系统实现的所有行为(踪迹)，检查其是否包含在规范描述的行为当中(CSP 中称之为系统实现为规范的精炼)。当系统实现的行为符合规范时，认为安全协议满足安全性质；当检查失败时，FDR 将返回一个违反规范描述的行为(即导致攻击成功的系统踪迹)。

G. Lowe 还设计了 Casper 程序[16]，它从安全协议的抽象描述(类似于消息序列的形式)半自动地产生 CSP 描述，因此大大简化了建模和分析过程。

由于模型检测技术需要搜索整个状态空间，因此，在状态急剧增大时，模型检测技术的效率也将下降(称为"状态爆炸问题")。在使用模型检测技术对安全协议进行形式化验证时也将面临这一问题。一般情况下，对这一问题通常采用了对安全协议模型进行限制的方法进行处理。这种限制是两方面的，即：

第一，对参与协议运行的主体个数进行限制。实际的网络环境下可能允许每个主体同时运行多个协议实体，但允许无穷多个协议实例的并发运行显然会带来无穷的协议模型状态空间。使用模型检测技术进行安全协议验证时一般都只允许有限个协议实例的运行。特别是关于保密性，G. Lowe 曾经证明[15]：如果构造一个运行协议的小系统模型(协议各方参与

者都只有一个，这样的小系统通常都是一个有限状态转换系统），同时结合一个通常意义下的入侵者模型（即入侵者可窃听及中途拦截系统中传送的任何消息、可在系统中插入新的消息、即使不知道加密部分的内容也可重放他所看到的任何消息（其中可改变明文部分）、可运用他所知道的知识产生新的消息，最后他还具有一般的正常用户的能力），如果在这样的协议模型中没有对协议的攻击导致某种强安全性破坏，那么在任意系统上一定没有攻击导致安全性的破坏。

第二，对攻击者可生成的消息数目进行限制。如果攻击者可生成无穷多的消息（事实如此），协议的状态空间也将是无穷的。使用模型检测技术对安全协议进行验证时，一般将对攻击者可生成的消息进行限制，例如，限制进行加密时所使用的密钥必须为原子密钥（而并非由其他的元素通过加密或其他的函数生成），要求攻击者可生成的消息与协议正常参与者可接受消息具有相同的结构。

为了进一步提高模型检测的效率，安全协议的模型检测技术还借鉴了在进程代数研究领域内非常重要的符号化方法与 Partial Order 方法，以便压缩状态空间。这方面的主要研究工作包括 M. Boreale 与 E. Clarke 及 R. Alur 等。限于篇幅，本文不再对此进行详述。

4.2 互模拟验证技术

互模拟（bisimulation）是进程代数领域最重要的核心概念之一，基于互模拟概念的行为等价语义理论的成功也是进程代数能够广泛应用于并发系统建模和程序验证的重要原因之一。采用互模拟验证技术进行安全协议的形式化验证的主要思想是：如果一个安全协议进程（实现）与对应的特定理想化进程（规范）是测试等价的（即在各种环境中，协议实现与规范在外部的观察者看来是没有区别的），那么安全协议进程将与该特定理想化系统一样具有相同的安全性质。

采用互模拟验证技术的工作包括有 Spi-演算[11]等。Spi-演算是 M. Abadi 和 A. D. Gordon[11]在 π-演算的基础上加以扩充得到的，用以描述和分析安全协议的模型。作为并发移动计算的基本模型，π-演算中引入了诸如通道（channel）、限制（restrction）和作用域（scope）等概念，这些概念与某些密码性质具有较强的对应性（例如，通道具有作用域，作用域之外的进程不能对该通道进行存取，与此类似，安全协议中使用某一密钥进行通信的数据相当于在受限通道中进行传输，该通道的作用域为共享此密钥的通信方；π-演算允许通过传递通道名扩大通道的作用域，而安全协议允许通过密钥传递建立新的保密通信信道，等等），特别适合于对安全协议进行描述。但 π-演算并没有为数据加/解密的描述提供相应的原语（基于公钥的安全协议无法通过 π-演算建模而必须引入描述数据加/解密原语），Abadi 和 Gordon 在 Spi-演算中增加了支持密码学的原语以便描述安全协议，同时引入测试等价关系作为进程的行为等价标准，并将安全协议的安全性质（认证性、保密性）均用测试等价来加以刻画。例如，保密性定义为：仅当在安全协议进程使用不相同的保密信息进行通信，而外界观察到的协议进程行为完全相同（测试等价）时，安全协议满足保密性；认证性的定义稍有不同，它也是以测试等价关系来定义的：理想化的协议规范与安全协议实现的形式类似，但在协议规范中，消息接收子进程已经"魔术般地"预知了发送者子进程将要发送的消息，当协议规范与协议实现测试等价时，安全协议满足认证性。

Spi-演算的缺点是目前还缺乏相应的工具支持，而采用互模拟验证技术验证协议实现与规范测试等价的过程比较复杂，导致其很难得到很好的应用。

但 Spi-演算的语法描述简洁，对安全协议刻画能力很强，许多研究人员都在采用 Spi-演算的语法定义的基础，另行定义了合适的形式语义，并采用了包括模型检测技术在内的许多其他方法进行安全协议的形式化验证，这方面的工作可见 M. Boreale 等人的工作。

5. 现状和发展趋势

综上所述，在最近十多年时间里，安全协议的形式化分析已取得了相当大的进展，但该领域尚有诸多问题亟待解决：

5.1 理论方面

1. 采用模型检测技术的进程代数方法通常使用小系统（即协议的参与者每一方都只有一个主体）进行分析验证，但小系统下的安全性是否与一般系统（实际网络环境下，网络协议可能并发运行，每个主体可能同时运行多个协议实例）一致是关系到该方法是否确实有效的重要保证。关于保密性，G. Lowe 已经证明了在安全协议满足一些假设的前提下，对小系统进行某种强保密性的证明即可保证一般系统的保密性。但对于认证性，是否存在类似的定理，还是一个没有回答的问题。

2. 安全协议的形式化分析研究采用了"完善加密假设"，从而将密码的因素排除在形式化分析研究之外。但在实际环境中，"完善加密假设"并不是完全成立的。例如，"不存在密文冲突"可能在实际的密码系统中只是以很大的概率成立。如何将密码的因素结合在形式化分析框架中，还是一个没有解决的问题。

5.2 实践方面

1. 目前的安全协议验证技术还只能对简单的协议进行分析和验证，对于实用的、大型化的协议无法进行验证。一方面这是由于原有的协议描述不精确，没有形式化的协议描述手段，协议所涉及的实际特性与使用环境及假设比较复杂；另一方面安全协议验证技术还存在一些弱点，比如，模型检测技术在大型化协议验证中需要处理"状态爆炸问题"。

2. 目前安全协议验证技术的研究还局限在保密性与认证性两类安全性质上，对于其它安全性质的研究还比较少。例如对电子商务协议的研究，除一般的保密性与认证性外，电子商务协议还应满足不可否认性、原子性、匿名性等，而对于这些性质的研究工作还不多。

3. 造成协议失败的重要原因之一是因为协议的设计者对安全需求的定义研究得不够透彻，缺乏一般性的安全协议设计准则。但这方面的工作比较缺乏，除早期一些研究工作外，还没有特别重要的成果出现。

上述这些问题都要求进一步推进安全协议的形式化研究，进一步完善和发展形式化验证方法与技术，促进安全协议的形式化验证技术走向实用化。

参 考 文 献

1 Meadows C A. Formal Verification of Cryptographic Protocols: A Surver

2 Meadows C. Open Issues in Formal Methods for Cryptographic Protocol Analysis

3 Abadi M, Burrows M, Needham R. A Logic of Authentication. In: Proc. of the Royal Society, Series A, 426, 1871, Dec. 1989. 233～271. Also appeared as SRC Research Report 39 and, in a shortened form, in ACM Transactons on Computer Systems, 1990, 8: 18～36

（下转第36页）

态$(a,0,1,1')$与$(b,0,1,1')$在C'中用一个-Data连接，但不可能对它们的所有分解状态都用-Data来连接，仅仅有状态$(a,1,1')$与$(b,1,1')$用-Data连接，因为在P_1中仅有状态1到状态1的转移为-Data。

经过以上的构造，完整的协议转换器可以取得，如图4所示。由此例可知，在构造前对实体P_1与Q_0的状态进行归并，达到简化的目的，使构造过程中的状态空间降到了$1*2*2=4$（如图3中(c)），使计算复杂度得到了改善。

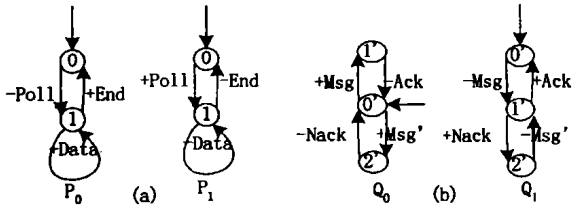

图1 (a)简单的Poll-End协议(P)
(b)简单Ack-Nack协议(Q)

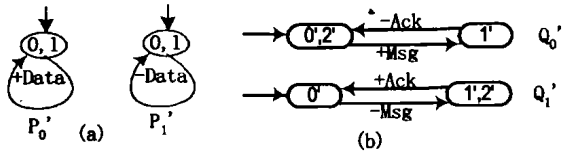

图2 (a)简化的Poll-End协议(P)
(b)简化Ack-Nac协议(Q)

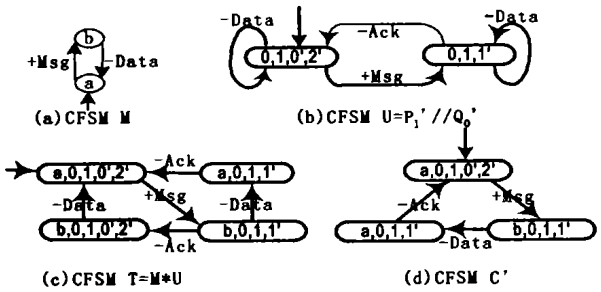

图3 Okumura转换过程

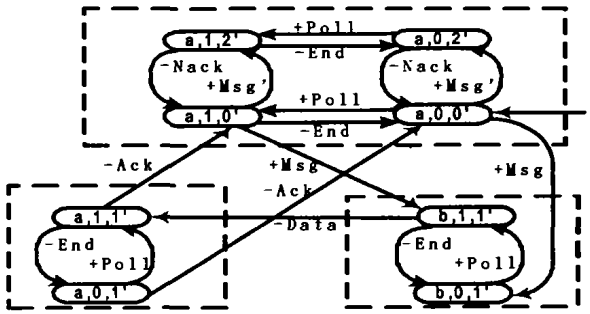

图4 经过分解C'得到转换器C

结束语 在两个异构网络之间进行通信，协议转换是一种有效的解决方法。在本文中，给出一种新的构造转换器方法。此方法在构造转换器之前对初始协议的CFSM模型进行了简化，协议实体中的状态、消息数量得到减少，使转换器构造过程中所用的状态空间得到降低，从而改善了构造过程中的计算复杂度。

参考文献

1 Okumura K. A formal protocol conversion method. In: Proc. of the ACM SIGCOMM Conf. on Communications Architecture & Protocols, 1986. 30～38

2 Saleh K, Jaragh M. Synthesis of protocol converters: annotated bibliography. Computer Standards &Interface 1998, 19: 105～117

3 Hallal H, Negulescu R, Petrenko A. Design of divergence-free protocol converters using supervisory control techniques. In: The 7th IEEE Intl. Conf. on Electronics, Circuits and Systems, 2000, 2(1): 705～708

4 Jaragh M, Saleh K. Synthesis of communications protocol converters using the timed Petri net model. Journal of Systems and Software, 1999, 47(1): 53～69

5 Huang C M, Lai H Y, Huang D T. A reduced incremental ECFSM-based protocol verification. In: The 17th Annual Intl. Conf. on Computer Software and Applications, 1993. 166～172

6 Calvert K L, Lam S S. Formal methods for protocol conversions. IEEE Journal on Selected Areas in Communications, 1990, 8(1): 127～148

（上接第27页）

4 Gong L, Needham R, Yahalom R. Reasoning about belief in cryptographic protocols. In: IEEE Computer Society Symposium in Security and Privacy. IEEE Computer Society Press, May 1990. 234～248

5 Abadi M, Tuttle M R. A Semantics for a Logic of Authencation (Extended Abstract). In: Proc. of the 10th Annual ACM Symposium on Principles of Distributed Computing, Aug. 1991. 201～216

6 Syverson P F, van Oorschot P C. On Unifying Some Cryptographic Protocol Logics. In: Proc. of the 1994 IEEE Computer Security Symposium on Security and Privacy, IEEE Computer Society Press, 1994. 14～28

7 Dolev D, Even S, Karp R. On the Security of Public Key Protocols. IEEE Transactions on Information Theory, 1983, 29(2): 198～208

8 Woo T W, C, Lam S S. A Semantic Model for Authentication Protocols. In: Proc. IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, May 1993. 178～194

9 Woo T Y C, Lam S S. Verifying Authentication Protocols: Methodology and Example. In: Proc. Intl. Conf. on Network Protocols, Oct. 1993

10 Ryan P, Schneider S. Modelling and analysis of Security protocols. Addison-Wesley Press, 2001

11 Abadi M, Gordon A D. A Calculus for Cryptographic Protocols: The Spi Calculus. Information and Computation: [SRC Research Report 149(January 1998)]. in preliminary form as Technical Report 414, University of Cambridge Computer Laboratory, Jan. 1997

12 Mao W. An Augmentation of BAN-Like Logics

13 Mao W, Boyd C. Towards Formal Analysis of Security Protocols. In: Proc. of the Computer Security Foundations Workshop VI, IEEE Computer Society Press, P. 147～158

14 Lowe G. Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR. In: Proc. of TACAS (Tools and algorithms for the Construction and Analysis of Sysmstems), Spring Verlag, 1996, 1055: 147～166

15 Lowe G. Towards a Completeness Result for Model Checking of Security Protocols. June 1999

16 Lowe G. Casper: A Compiler for the Analysis of Security Protocols. In: Proc. of 10th IEEE Computer Security Foundations Workshop, 1997. Also in Journal of Computer Security, 1998, 6: 53～84

17 Denning D E, Sacco G M. Timestamps in key distribution protocols. Communications of the ACM, 1981, 24(8): 533～536