两种安全协议形式化理论的研究

赵华伟

(山东大学网络信息安全研究所，济南，250100)

摘 要

在本文中，我们探讨了有关两种安全协议形式化理论的若干问题，其中，工作重点集中在对三方认证密钥交换协议的形式化分析上。

网络技术具有快速实现信息共享的特点，这大大提高了人们通信的效率，给人们的生活、学习和工作带来了巨大的改观。但与此同时，人们对网络本身具有的公开性和匿名性所带来的日益严重的信息安全问题也深感不安。

为了解决网络的信息安全问题，人们设计了许多用于开放网络的安全协议来解决各种安全应用问题。所谓安全协议，就是两个或两个以上的参与者采取一系列步骤来完成某项特定的安全任务。它包含三层含义：1. 协议需要至少两个参与者。2. 参与者之间执行的是消息处理和消息交换交替进行的一系列步骤。3. 通过协议执行必须能够完成某种安全任务。

由于每个安全协议都是为了某种安全应用而精心设计的，协议中的各条消息间有着微妙的相互制约关系，因此若采用人工方式对协议进行安全性分析，往往不能发现协议存在的问题，所以必须借助于形式化的方法来完成。形式化方法是一种用于描述系统性质的数学方法，它主要用于发现一个系统中的歧义性、不一致性与不完备性。该系统可以大到一个企业级的软硬件系统，也可以小到有若干条消息组成的协议。对安全协议使用形式化的方法，可挖掘出协议消息所表示的内在含义，对协议的正确性进行验证。通过验证，不仅可证明协议是否符合预期的安全目标，而且对不符合安全目标的协议可分析其缺陷之所在，进而为协议的设计提供有力的支持。

目前在安全协议形式化的研究领域有两种截然不同的理论：符号理论与计算理论：

在符号理论的形式化中，协议各要素被抽象为赋予了特定含义的符号，然后通过对这些符号的形式化分析来考察协议的安全性。虽然符号理论为协议的验证工作提供了强大的证明手段，但是这些证明手段都是在一定的假设下进行的，比如均假设协议所采用的算法是理想化的，敌手在进行密码分析时得不到任何有用的信息。由于这些假设不仅高估了一些常见算法的强度，同时低估了敌手的攻击能力，因此符号理论不具有计算可靠性。

在计算理论中，协议各要素被看成位串，协议所采用的算法由函数表示。其安全性的证明通常是通过构造一个“矛盾归约”的证明得到，这里的矛盾指的是计算复杂性领域中人们普遍相信的困难问题出现了一个有效解。计算理论通过分析敌手的攻击能力来考察协议的安全性，更具有实际意义。但是计算理论下的分析方法难以考察协议各要素的完备性，因此不适于分析协议的逻辑正确性。

在本文中，我们主要以认证密钥交换协议为载体，深入考察了符号理论与计算理论各自的优缺点，然后提出了若干有关安全协议形式化的新思路，主要的工作有以下几个方面：

第一，提出了 CS 逻辑及其改进方案的不足之处，对 CS 逻辑重新进行了扩展。CS 逻辑在公理方面存在一些不足，比如违背了公钥的公开性原则，知识与信念的转化是通过非形式化的手段得到的等等，而已有改进方案中有关时间单调性与消息新鲜性的公理不能正确反映现实。我们通过对逻辑公理的改进与扩展实现了：1. 更准确地描述了有关知识和信念与时间有关的单调性概念。2. 对发送和接收有关密文的知识和信念进行了描述，优化了原有的推理过程。3.改进后的推理公理描述了知识与信念之间的转化关系，更客观地反映了人类的认知能力。4. 可分析协议要素在一定时间段内的安全需求，为深入了解协议消息的性质以及设计更优质的安全协议提供了强有力的帮助。

第二，针对 BAN 类逻辑的不足，构造了一个新的形式化分析工具—MBL 逻辑。该逻辑的特点在于：1. 克服了 BAN 类逻辑缺乏有效证明机制的不足，具有严格的证明体系，可证明在给出的语义模型下推理规则的正确性，说明了逻辑系统的合理性。2. 对协议进行简单的处理便可进行安全性分析，不需进行理想化处理，降低了分析者分析协议的难度。3. 可分析协议消息的保密性，使得主体不必过分依靠可信中心，主要根据自己的判断来确定秘密是与哪个主体共享的，从而能弥补 BAN 类逻辑不能发现出于敌手欺骗可信中心所造成攻击的不足。此

外，我们基于 MBL 逻辑并采用 Prolog 语言实现了一个协议的自动化分析工具，可以大大提高分析协议安全性的效率。

第三，发现了敌手针对认证密钥交换协议的一种新型攻击：敌手利用针对协议的某种看似“无用”的攻击过程，能够完成另一种严重的攻击，即敌手能够延续对协议的攻击效果从而实现另一种攻击。由于目前所普遍使用的 Dolev-Yao 敌手模型无法具体涵盖该类型的攻击，所以我们对 Dolev-Yao 敌手模型进行了补充。

第四，分析了设计认证密钥交换协议所需注意的事项，指出设计安全的协议不仅要保证具备必要的协议要素，还要保证采用正确的密码算法，由此引出了在计算方法下讨论协议安全性的初衷。然后通过两个攻击游戏模型化了敌手攻击认证协议的行为，进而通过主体在游戏中的成功概率给出了敌手在实施有关密码分析的攻击时认证密钥交换协议认证性的定义。该定义充分考虑了敌手实施主动攻击时协议的安全性，具有实际意义。

第五，结合符号理论中的模态逻辑法与计算理论下的矛盾归约法，提出了一种安全协议的调和分析法。该方法与符号理论下的方法相比，克服了它们总假设密码体制是完善的，并且仅考虑敌手的被动攻击的不足，可有效地分析在面对敌手实施有关密码分析的主动攻击时协议的安全性；与计算理论下的方法相比，该方法能考察协议各要素的完备性，能有效发现协议在面对重放攻击、平行穿插攻击等非密码分析的攻击时是否存在漏洞。最后我们给出了采用该方法分析协议的一个实例。

下一步的工作包括：建立一种设计安全协议的逻辑方法，然后利用 prolog 语言建立该逻辑方法的自动化方案，并实现由协议目标来自动定位协议的缺陷；研究在各种攻击模型下对称钥体制的安全性，以便提出更好的用于安全协议的对称钥加密方案，以及更全面地分析加密方案的安全性。

关键词：安全协议，符号理论，计算理论，保密性，自动化分析

Research on Two Formal Theories of Security Protocols

Zhao Huawei
(Institute of Network Information & Security, Shandong University, Jinan, 250100)

ABSTRACT

In this dissertation, we discuss some problems related to the two theories of the formal methods in the field of security protocols, and our work is mainly on the formal analysis about three-part authenticated key-exchange protocols.

With the advance of the Internet, the efficiency of communications is greatly improved, which bulkily changes appearances of people's methods of living, studying, and working. But at the same time, the problem of information security caused by anonymity and publicity of INTRENET arises people's misgivings.

In order to solve the problem of information security, people already designed a lot of security protocols for secure applications. Security protocol are process having two or above participants to complete a kind of security task. It has three points: firstly, protocols have two participants at least. Secondly, participants must deal with messages and exchange messages by turns. Thirdly, protocols must achieve some kind of security task.

Elaborately designed messages of protocol have delicate interactive relationship, so if we analyze protocols using non-formal methods, defects are hardly to be found commonly. In order to solve the problem, we need to recur to the method of formal analysis. Formal analysis is a mathematic method used to describe system properties and its purpose is to find ambiguity, congruity, and imperfection of a system. The system may be so big as a enterprise system including software and hardware, or so small as protocols including some messages. Through formal analysis methods, we can disclose messages' inherent meanings, and verify protocols' rightness. If a protocol isn't secure, formal analysis method can find the reason and advance the opinion about mending the protocol.

Now, there are two different theories in the fields of security protocols' formal analysis: symbol theory and computational theory:

In the symbol theory, all kinds of methods look each protocol message as symbol having special meaning, and analyze these symbols to verify protocols' security. Symbol theory can support powerful approaches for protocols' verification, though these approaches need some assumptions, for example: all cryptographies are supposed perfect, and adversary cannot achieve any useful information in the period of cryptographic analysis. But the assumption doesn't correspond to actual needs, because it overrates the cryptography's security, and underestimates adversary's ability.

In the computational theory, message in a protocol is strings of bit, an encryption function is just an algorithm, and an adversary is essentially a Turing machine. Good protocols are those in which adversaries cannot do "something bad" too often and efficiently enough. Usually a security proof is related to a induction, that is: if an adversary could successfully attack a protocol easily, hen he could successfully solve a hard problem easily, but in fact the problem couldn't be solved, then through the contradiction, we can believe the protocol is secure. Computational theory is difficult to check whether a protocol has necessary messages, and doesn't adapt to analyze protocols' logistic rightness.

In this dissertation, through analyzing authenticated key-exchange protocols, we study the flaws and virtues of symbol method and computational method, and purpose some new ideas about security protocols' formal methods. Our primary work includes:

Firstly, We find flaws of both CS logic and its improved schemes, and improve the logic again. CS logic has some limitations in its axioms, such as violating public-key's exoteric character, lacking formal methods to transform between belief and knowledge. In addition, in its improved scheme, some axioms about time monotony and freshness are wrong. In the paper, we improve CS logic again, and achieve the following goals: 1, To depict time monotony conception of belief and knowledge more accurately.2, To depict knowledge and belief about sending and

receiving messages, and optimize former inference process.3, To depict the relationship between Knowledge and belief in the modified inference axioms, which images human cognitive powers objectively.4, To be able to point out messages' secure requirement in a period of time, and help to understand messages' properties and to design security protocols better.

Secondly, Facing the flaws of BAN-like logicsss, we build a new formal analytic tools-MBL logic, the logic has the following characters: 1,MBL logic has strict proof system, and can prove the rightness of inference rules, on the contrary BAN-like logicsss lacks effective proof mechanism.2, The logic doesn't need idealization process, so can depress the difficulty of analyzing protocols.3, The logic can analyze the secrecy of protocols' messages, which could help analyzer judge who is sharing a secret with him by himself, and reduce the dependence on trusted center. Moreover, based on MBL logic and using Prolog language, we realize an automatic analyzing tools for protocols, which could improve efficiency of analyzing protocols.

Thirdly, in the process of building MBL logic, we find a new attack ability of adversary to authentication key-exchange protocols: he could use an "useless" attack to actualize a redoubtable attack, that is adversary firstly carry out an attack which is believed harmless firstly, but he could continue the attack effect to realize another attack which could cause severe harm. Through studying the adversary ability, we find the DOLEV-YAO adversary model couldn't depict this ability, so we add the new ability on the he DOLEV-YAO adversary model.

Fourthly, we analyze how to design a secure authenticated key-exchange protocols, and point out a secure protocol not only rely on necessary messages, but also rely on sound cryptography, which gives the initial ideas of discussing the security under computation theory. Then we build two games to model adversary's ability to attack protocols' authentication, and through principle's success probability, we give the definition of authentication in the field of computation theory.

Fifthly, reconciling theorem proof method of symbol theory and inducing-to-contradiction method of computation theory, we design a new analytic method. In contrast to methods of symbol theory, the new method could analyze

adversary's ability to attack cipher text, and in contrast to methods computation theory, the method could analyze whether protocols having necessary messages, and verify their security. Finally, we give an analyzing example using the new method.

The next work include: building a logic to design security protocols, then use Prolog language to realize automatic designing scheme, through which we could locate protocols' flaws automatically; studying symmetrical-key cryptography's security in kinds of attack model, In order to advance some better schemes to design protocols, and analyze protocols' security all-around.

Key words: security protocols, symbol theory, computation theory, secrecy, automatic analysis

符号说明和缩略词

A、B	主体
S^I	可信中心
E	特指敌手
K_A、K_A^{-1}	A 的公钥、私钥
$\{M\}_K$	用密钥 K 对 M 加密的密文
Na	A 的 *nonce*
k	密码体制的安全参数
$\oplus M$	公开消息
$\otimes M$	保密消息
$\odot M$	M 具有完整性
ϕ,φ	表达式
$A\ bels(\phi)$	A 相信 ϕ 成立
$A\ says(M)$	A 刚刚说过 M
$A\ said(M)$	A 曾说过 M

原 创 性 声 明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对论文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明，本声明的法律责任由本人承担。

论文作者签名：赵华伟　　日　期：2016年3月15

关于学位论文使用授权的声明

本人完全了解山东大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅；本人授权山东大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

（保密论文在解密后遵守此规定）

论文作者签名：赵华伟　　导师签名：　　日　期：2016年3月15

第一章 绪论

当今社会，随着计算机技术和互联网技术的日益发展和广泛应用，人们越来越强烈地感受到无地域性和实时性的网络数字流正在全方位地改变着信息的传播方式，使之变得更加快捷和高效。

但人们在享受计算机网络技术所带来的梦幻般快捷的通信时，也越来越深刻地感受到随之而来的一个严重威胁—信息的不安全：由于互联网本身具有公开性和匿名性的特点，使得网上传递的信息可被随意窃听和篡改，人们不仅不能产生面对面通信时所获得的信任，而且对于所接收信息的真实性和保密性也感到怀疑。由于计算机技术和互联网技术已经渗入到国家的社会生活、政治、军事和经济的方方面面，能否很好地解决这些信息安全问题，不仅关系到人们的日常生活，而且还涉及国家安全问题，因此各国政府和科研机构已将其作为高额投入的焦点。此外，信息安全技术涉及密码学、计算机科学、信息论等各个学科，是一门综合性研究领域，吸引着国内外大量学者的兴趣，因此也是科学研究领域的一个重要方向。

1.1、安全协议的一些基本概念

信息安全领域的一项重要内容就是研究如何保证网络上传输数据的机密性、完整性、身份或数据来源的鉴别、不可否认性、可追究性、匿名性等功能。这些目标的实现，通常是运用密码算法，设计一个在网络上运行的两方或多方安全协议[1]来完成的。随着人们对网络安全需求的日益增强，安全协议作为一种有效的安全保护手段也受到了越来越多的关注，针对它的研究已经成为信息安全领域的一项重要分支。

首先我们给出有关安全协议的一些基本概念。

1.1.1、网络协议及会话

网络协议可以看作是通过消息来驱动的：设 N 个主体 $P_1,.....P_n$ 组成一个集合，他们之间通过点对点的连接来传递消息。所谓网络协议[2]是这些交互过程的

集合，执行这些过程的主体同步运行，并按照一定的规范处理接收到的消息、产生新消息。协议可由外部请求触发一方主体开始，然后依靠消息的收发执行下去，主体所能执行的动作有：接收并处理消息、产生并发出新消息、等待下一条消息的到达、拒绝执行协议、结束协议。

每个主体可同时运行多个协议的副本。所谓会话，指在主体中运行的某个协议的副本。会话由主体标示符、通信对方标示符和一个会话标示符来唯一确定，它们在实际中也指明了存在于通信对方中的对应会话。

1.1.2、安全协议及分类

安全协议是一类特殊的网络协议。简单而言，安全协议是指那些利用密码学算法实现某些安全目标或者安全性质的网络协议。

根据安全协议的应用目的，一般可划分为以下四种[3]：

1. 密钥交换协议：为了实现主体间的安全通信，这类协议通过一系列消息的交互，在两方或多方主体之间共享了一个秘密的临时会话密钥。协议中的密码算法可采用对称钥算法或公钥算法。常见的密钥交换协议有：Diffle-Hellman 协议[4,5]、Blom 协议[6]、Kerboros 协议[1]、端-端协议[4,5]、MIT 协议[7]和 Girault 协议[8]等等。

2. 认证协议：这类协议通过消息的交互来实现通信主体身份的认证或数据源的认证，来防止敌手的冒名或篡改数据源等攻击。在这类协议中，比较著名的有 Feige、Fiat 与 Shamir 在 1986 年提出的基于零知识的身份识别协议[9]、Schnorr 协议[10]、Okamoto 协议等[11]。

3. 认证密钥交换协议：在协商秘密临时会话密钥的同时，实现主体认证或者数据源认证。Yahalom 协议[12,13]，Needham-Schroeder 协议[14]等均是这类协议的代表。

4. 电子商务协议：是随着电子商务的广泛应用而产生的一种协议。协议的运行不仅要保证数据的保密性，防止假冒和篡改攻击，而且公平性、不可否认性也是需要考虑的重要性质。常见的电子商务协议有 SET 协议、iKPI 协议。

1.1.3、安全协议的服务

作为一种在公开网络上为信息提供安全保障的手段，安全协议可提供以下几种安全服务：

机密性：机密性也称为保密性，通常在不同的应用领域有着不同的解释。一个严格的保密性描述需要敌手得不到合法用户活动的任何有用信息，即既不能推断加密消息的内容，也不能做任何的流量分析。但对于大多数应用而言，这个描述过于严格。大多数情况下，只要防止攻击者能得到通过可信节点的被保密的消息就足够了。即敌手能够通过某种方式知道 *A* 向用户 *B* 发送了一个消息，甚至清楚地知道这个消息持续的时间，但无法知道消息的内容。

认证性：是最重要的安全服务之一，许多的安全应用都依赖于该服务。它是用来获取对谁或对什么事情信任的一种方法，包括数据源认证和身份认证两方面。

数据源认证：接收者能确定消息声称的源地址是消息真正发出的地址。例如：*B* 收到一条自称是发自 *A* 的消息，则该消息就一定是用户 *A* 发出的。但是在该认证通过时，消息可能已被许多系统重放，且这些系统并没有通过认证。数据源认证与完整性有一些联系，必须确认消息在离开它的信源后没有被篡改。然而，数据源认证不能免遭消息的复制、重排或者丢失。

身份认证：通过消息的交互来确认对话者的身份。它可以是单向的或是双向的。所谓单向身份认证是指通信双方中只有一方向另一方进行认证；所谓双向认证是指通信双方相互认证。它与数据源认证较为明显的一点区别在于它包含“实时性”的含义，即身份的认证局限于当前协议的执行阶段。而数据源认证没有该要求。

完整性：是指数据在传输或者存储的过程中没有损坏（修改、丢失、重排序或替换）。在具体应用的时候，我们是通过针对数据的任何篡改都会被检测出来来保证数据的完整性。在某种意义上，我们认为完整性是协议正确运行的前提，即每一步正常协议动作的执行都隐含着消息的完整性。

不可否认性：主要应用于电子商务协议。对于大多数安全目标，我们总假设合法用户是诚实可信的，也就是他们将遵循协议规范。但在电子商务协议中，则是另一种情况。因为利益的驱使，合法主体可能会欺骗通信的对方以便获取利益。

而不可否认性就是讨论如何保护通信的一方不被欺骗。采取的方法是要为用户提供证据，来证明协议中某些步骤确实已经发生过。

公平性：电子商务中可能会出现这种情况：*A* 签名后，*B* 却在即将签名前拒绝继续执行协议，这样 *B* 就获得了比 *A* 多的优势。因此我们需要考虑如何避免参加协议的任何一个用户通过半途中止协议来获取比另一方更多的利益，这就是公平性。一般采用引入可信的第三方并/或增加承诺机制来实现。

1.2、密码学的相关内容

密码学技术是信息安全的关键技术，也是为安全协议提供安全服务的基础。

1.2.1、常用的密码体制

密码学对信息的安全保护主要分为两个方面：保密性和认证性。保密性是防止敌手破译系统中的机密信息。认证性有两个目的：一是验证信息的发送者不是伪造的；二是验证在传输或存储中的信息没有被篡改、重放或伪造。在密码学中为了实现这两个方面的安全，通常采用以下两种密码体制：

一. 加密体制

加密体制从原理上可分为两大类：对称钥体制和公钥体制[15]。两种加密体制的保密性仅取决于相关密钥的保密性，与密码算法的保密性无关。

- **对称钥体制**

对称钥体制的加密密钥和解密密钥是相同的，或者两者能很容易地相互推出。在这种体制中，需要考虑的主要问题是如何产生满足保密要求的密钥，以及如何将密钥安全可靠地分配给通信双方。

对称钥体制下对明文消息的加密有两种情况：一种是明文消息按字符逐位进行加密，称为流密码[16-18]；另一种方式是将明文分组，逐组加密，称为分组密码[19-27]。对称钥体制不仅能用于数据加密，还能用于消息认证。

- **公钥密码体制**

公钥体制中每一用户有一对密钥：解密密钥和加密密钥。两种密钥不同，其中加密密钥公开，称为公钥，解密密钥需要保密，称为私钥，且从加密密钥难以

推导出解密密钥[5,28-31]。

公钥体制的特点是将加密和解密功能分开，因而能实现多个用户加密的消息只能由一个用户解密，或者一个用户加密的消息可被多个用户解读。前者用于消息的秘密传输，后者实现对用户身份的认证。

两种加密体制相比，对称钥体制加解密的效率高，适于对大量数据进行加解密操作，但是如何在密文传输前，为公开网络上的通信双方分配对称钥是个难题。公钥体制的公钥具有公开性，易于分配密钥，但其加解密的效率低。因此在安全协议中，我们一般将两种体制有机地结合，来实现秘密消息的传输。

二．散列函数

散列函数是一种能将任意长度的输入数据映射成某种固定长度输出数据的函数，且输入数据任何一比特或几比特的改变都会造成输出数据在输出空间中的均匀变化。理论上讲，好的散列函数的存在依赖于单向函数的存在[1,32,33]，且具有破坏消息的某种数学结构的功能，使得从散列值不能反向推导出消息本身。

散列函数的这种单向性使之广泛应用于数字签名、消息完整性检测、消息源认证等方面。比如，A 打算传递消息 M 给 B，且要保证 M 的完整性，他可以这样做：首先用与 B 协商好的散列函数 $H(\ )$ 获取 M 的散列值 $H(M)$，然后用与 B 共享的秘密 K 进行加密得到 $\{H(M)\}_K$，最后将 M 与 $\{H(M)\}_K$ 发送给 B。当 B 接收到消息 M' 后，计算出其散列值 $H(M')$，然后对其用 K 加密得到 $\{H(M')\}_K$，最后比较 $\{H(M')\}_K$ 与 $\{H(M)\}_K$ 是否相等，就可以判断 M 在传输中是否被篡改。

1.2.2、常用的密码要素

由于网络的公开性，所传输的消息能被任意复制，被存储一段时间然后重放，因此接收消息的主体很难判断消息的新鲜性。安全协议为了解决该问题，经常借助于以下两种密码要素：

一．随机数

随机数在安全协议中有着重要的应用：第一，在安全协议的相互认证中，使用一次性随机数作为 *nonce* 可防止消息重放攻击；第二，一次性随机数能构造双方的确认值；第三，随机数能作为密钥产生的种子。

随机数在应用时，需要满足两种性质：随机性和不可预测性。

随机性：以下两个准则常用来保障数列的随机性：

1）均匀分布 数列中每个数出现的频率应相等或近似相等。

2）独立性 数列中任意一数都不能由其他数推出。

数列的均匀分布能通过检测得出[34]，但是否满足独立性则无法检测出。由于有很多方法能检测出数列不满足独立性，因此通常检测数列独立性的方法是在足够多次检测都不能证明不满足独立性，就可较有把握地确信该数列满足独立性。

不可预测性：

除了随机性，在应用中，随机数还要满足数列中以后的数不可预测。对于真随机数而言，数列中每个数都独立于其他数，对于伪随机数而言，就需特别防止敌手从数列左边的数预测出后面的数。

在安全协议中所需的随机数都是借助于安全的密码算法产生的。但由于算法是确定的，因此产生的数列不是真随机的。然而，若算法设计的好，产生的数列就能通过各种随机性检验，这种数就是伪随机数。

二．时间戳

在安全协议设计中，时间戳是一个很重要的概念，它不仅能保证信息的新鲜性，而且可作为非否认机制的基本要求。例如，在一个简单的发起者数字签名机制中，可能会发生由于发起者故意公开密钥然后鼓动撤销密钥从而导致自己过去的签名无效。为了预防这种攻击，就需要时间戳的参与。

时间戳的产生有好几种方式，比如主体产生时间戳、可信时间戳服务器（TS）产生时间戳、根据链接协议和分布式协议来产生时间戳等等。但是考虑到时间戳产生的效率问题和用户的计算能力问题，我们一般采用由可信时间戳服务器来产生时间戳的方式。我们可假设将可信时间戳服务器（TS）放在可信中心 S 处，由可信中心来管理，这样我们可认为时间戳是由 S 产生的。如果一个主体 A 要产生一个自己的时间戳，那么过程是：A 先计算 $z=h(x)$ 和 $y=Sig_A(z)$，然后将 (z,y) 发送给 S。S 将级联日期 D，并对 (z,y,D) 进行签名。这种方法只能证明 A 在某一时间之前签了消息。如果 A 想说明他在某一时间之后签了消息，他可以插入某一公开知道的信息 pub，即可将过程改为：A 将 (z,y,pub) 发送给 S，S 级联日期 D，并对 (z,pub,y,D) 进行签名。只要使用的散列函数有足够好的抗碰撞性，我们就能解决签名过程中的明文问题、传输过程中的带宽和存储容量问题、文件

的完整性等问题。如果还要验证者的信任问题，即怎样使验证者相信用户没有和可信中心 S 串通起来欺骗，我们还可以采用链式协议技术，具体方法读者可参考有关资料[15, 35, 36]。

但为了使时戳生效，需要一种可靠的方法来保证网络中时间的一致性，这是个难题。

1.3、针对安全协议的攻击

安全协议是由主体间需传递的消息组成，每条消息都经过精心构造，消息之间相互作用与制约，以确保达到安全目标。但由于网络环境的复杂性，协议的设计者往往对协议运行环境的安全需求估计不足，或者采用了不当的技术，这就会导致协议存在漏洞，并引发敌手的攻击。下面我们介绍一下敌手常用的攻击策略。

1.3.1、消息重放攻击

消息重放攻击是最常见的一类攻击。由于敌手能在网络中监视安全协议的执行，所以他可将监听到的协议消息记录下来，然后将消息进行有意义的组合后在某一个特定时刻进行重放，若协议消息中没有区分协议步骤，或不能检测消息的新鲜性，那么合法的主体就可能上当受骗，将重放消息作为一个真实消息予以接收。重放攻击是敌手实施攻击的常用手法，很多攻击都可以看作消息重放攻击的一种。Paul F.Syerson 根据消息的来源与去向对消息重放攻击进行了细化[37]。

根据消息的来源将重放攻击分为本轮协议内的攻击和本协议轮次外的攻击。前者是指对一轮协议内的消息进行重放，后者是指对一个协议不同轮次的消息进行重放。对于协议的轮外攻击，根据协议的不同轮次的执行时间是否重叠又可分为交叉攻击和典型重放攻击。前者指不同轮次的协议执行时间有重叠，而后者是无重叠的。

根据消息的去向，将重放攻击分为偏转攻击和直接攻击。前者是指改变了消息的去向，使消息被非意定的主体接收，又可分为两种情况，一种是将消息返回给了发方，称为反射攻击，这种攻击与协议设计的结构有关。另一种是将消息发给了协议合法通信双方之外的任一方，称为第三方攻击。后者是指消息的确发给了意定的接收方，但是被延迟了。

下面我们举一个重放攻击的例子，它一般被称作中间人攻击，但本质上是一

种本协议轮次外的交叉重放攻击。

Needham-Schroeder 公钥认证协议

1. $A \rightarrow B : \{Na, A\}_{K_B}$
2. $B \rightarrow A : \{Na, Nb\}_{K_A}$
3. $A \rightarrow B : \{Nb\}_{K_B}$

Lowe 发现的对 Needham-Schroeder 公钥认证协议的交叉重放攻击[38]：

$$A \xrightarrow{(1)} E : \{Na, A\}_{K_E} \quad , \quad E \xrightarrow{(2)} B : \{Na, A\}_{K_B}$$

$$A \xleftarrow{(4)} E : \{Na, Nb\}_{K_A} \quad , \quad E \xleftarrow{(3)} B : \{Na, Nb\}_{K_A}$$

$$A \xrightarrow{(5)} E : \{Nb\}_{K_E} \quad , \quad E \xrightarrow{(6)} B : \{Nb\}_{K_B}$$

敌手 E 为了欺骗 B，使 B 认为正与 A 通信，E 建立了一次与 A 运行的正常协议，然后将 B 产生的消息 $\{Na, Nb\}_{K_A}$ 重放给 A。在此次攻击中，E 利用了 A 的解密预言服务获得了 B 的一个秘密 Nb，从而达到了欺骗 B 的目的。由于重放是在不同的协议轮次之间运行的，所以是一种交叉重放攻击。

下面我们观察一种类型缺陷攻击，它本质上是一种轮次内的反射攻击。

Neuman-Stubblebine 协议[39]：

1. $A \rightarrow B : A, Na$
2. $B \rightarrow S : B, \{A, Na, T_B\}_{K_{BS}}, Nb$
3. $S \rightarrow A : \{B, Na, K_{AB}, T_B\}_{K_{AS}}, \{A, K_{AB}, T_B\}_{K_{BS}}, Nb$
4. $A \rightarrow B : \{A, K_{AB}, T_B\}_{K_{BS}}, \{Nb\}_{K_{AB}}$

对该协议的攻击[40,41]：

1. $E("A") \rightarrow B : A, Na$
2. $B \rightarrow E("S") : B, \{A, Na, T_B\}_{K_{BS}}, Nb$
3. 跳过
4. $E("A") \rightarrow B : \{A, Na, T_B\}_{K_{BS}}, \{Nb\}_{Na}$

该攻击针对 Neuman-Stubblebine 协议结构上的缺陷：B 不能辨别消息 2 和消息 4 的差别。敌手利用了该缺陷，将 B 发送的消息又反射给 B，造成攻击。

1.3.2、平行会话攻击

平时会话攻击中，在敌手 E 的特意安排下，不同轮次的协议并发运行，而 E 能依靠这种并发运行，从一个运行中得到对另外某个运行的攻击能力。

这种平行会话攻击与重放攻击中的交叉重放攻击类似，都是利用了不同轮次协议的运行，但是平行会话攻击中具有攻击效果的消息是敌手自主产生的，而并非重放消息。

下面我们举一个平行会话攻击的实例[42,43]。

Woo-lam 协议：

前提：A 与可信中心 S 共享对称密钥 K_{AS}，B 和 S 共享对称密钥 K_{BS}。

目标：A 向 B 进行认证。

1. $A \rightarrow B : A$
2. $B \rightarrow A : Nb$
3. $A \rightarrow B : \{Nb\}_{K_{AS}}$
4. $B \rightarrow S : \{A, \{Nb\}_{K_{AS}}\}_{K_{BS}}$
5. $S \rightarrow B : \{Nb\}_{K_{BS}}$

对 Woo-Lam 的平行会话攻击：

注：敌手 E 也是一个合法主体，它与 S 共享会话密钥 K_{ES}。

1. $E("A") \rightarrow B : A$
1'. $E \rightarrow B : M$
2. $B \rightarrow E("A") : Nb$
2'. $B \rightarrow M : Nb'$
3. $E("A") \rightarrow B : \{Nb\}_{K_{ES}}$
3'. $E \rightarrow B : \{Nb\}_{K_{ES}}$
4. $B \rightarrow S : \{A, \{Nb\}_{K_{ES}}\}_{K_{BS}}$
4'. $B \rightarrow S : \{E, \{Nb\}_{K_{AS}}\}_{K_{BS}}$
5. $S \rightarrow B : \{"乱码"\}_{K_{BS}}$
5'. $S \rightarrow B : \{Nb\}_{K_{BS}}$
6. B拒绝和E的运行
6'. B接收“与A的运行”，但实际上是在与E运行

在 Woo-Lam 协议的攻击中，敌手 E 将协议中本该发给 B 的消息 $\{Nb\}_{K_{AS}}$ 改为

$\{Nb\}_{K_{ES}}$，而将另一轮协议中，本该发给 B 的消息 $\{Nb'\}_{K_{ES}}$ 也改为 $\{Nb\}_{K_{ES}}$，从而使 B 认为“与 E 的运行”发生了错误，而“与 A 的运行”是正确的，但事实上，“与 A 的运行”恰恰是 E 参与的。

在此次攻击中，E 发给 B 的消息 $\{N_B\}_{K_{AS}}$ 是 E 自己生成的，这是与交叉重放攻击的最大区别。

1.3.3、密码学缺陷攻击

若协议所采用的密码体制没有提供恰当的安全保护，利用此缺陷，敌手可进行有效攻击，最常见的有以下两种：

- 针对误用安全服务的攻击

很多安全协议都采用保密性来实现协议的认证[12,13,44]。该方法的思路是：认证方需要向被认证方发送一条加密的消息 $\{M\}_K$，被认证方为了从加密消息中提取 M，就必须执行解密操作，而正是这种解密能力向认证方证明了拥有密钥 K 的事实，从而证明了自己的身份。

但这种认证方法存在着极大的安全隐患。例如 1.3.1 节描述的 Lowe 对 Needham-Schroeder 协议的攻击，敌手利用主体的解密预言服务就获得了某种解密能力，从而可冒充一方向另一方认证成功。另外，很多加密算法都具有良好的数据结构，敌手据此就可发起有效的攻击。比如文献[45,46]中提出使用 CBC 加密算法来实现协议的认证性，但文献[47]指出，敌手通过精心修改 CBC 的密文，就能造成攻击。

- 对弱加密体制的攻击

很多的安全协议重点描述协议的运行步骤，但对所采用密码体制的说明却很简单，比如仅仅规定 $\{M\}_K$ 表示使用密钥 K 对消息 M 的加密，至于采用何种具体算法，则由协议的使用者自己决定。这就为密文的保密性带来了风险，因为若协议使用者对加密算法不甚了解，采用了弱强度的加密体制，敌手就很容易获得密文。

大部分教科书中的加密体制都是不安全的，甚至于缺少最基本的“语义安全性”[47]。比如 Lipton 就曾经发现了当采用 RSA 体制的一种变形方案时，智力扑克协议存在的缺陷[48]，因为这种 RSA 不具备语义安全性,不能抵抗敌手的被动攻

击。而文献[47]更指出了一种对语义安全的 ElGamal 体制的攻击，因为该体制无法抵抗敌手的主动攻击。

1.4、本文工作的背景及意义

安全协议以密码算法为基础，通过对消息实施各种密码变换来实现特定的安全服务。安全协议所面临的一个主要问题是如何判断协议的安全性。由于协议的运行环境中存在一个强大的敌手，所以即使协议构架在优质的密码算法之上也可能存在缺陷，然而由于协议的消息间有着巧妙的制约关系，采用人工分析方法往往不能发现缺陷和可能受到的威胁，因此必须通过形式化的手段来完成。

纵观形式化方法的发展，上世纪 80 年代末提出的 BAN 逻辑[49]是其开始繁荣的起点。BAN 逻辑以 Dolev-Yao 模型[60]为基础，是一个简单而实用的形式化方法。但由于它依赖于过多的系统假设，不考虑出现不可信的合法主体的情况，使得它无法分析 Needham-Schroeder 公钥协议[13]中的缺陷。其后，为了弥补 BAN 逻辑的这个缺陷，人们提出了各种各样的形式化方法，比如 GNY 逻辑[51]、AT 逻辑[52]、VO 逻辑[53]、SVO 逻辑[54,55]，都从不同的角度完善了 BAN 逻辑。另有一些研究人员跳出 BAN 逻辑的范畴，提出了新的形式化方法，比如，串空间理论[56]、CSP 理论[57]等等，使得人们可以摆脱更多的系统假设，更有效地分析协议的安全性。

目前，安全协议形式化方法呈现一片百家争鸣的局面，而其背后隐藏的是信息化社会对安全问题解决方案的强烈需求。而正是这种需求促使该理论快速地发展完善，并从理论研究阶段向实际应用阶段迅速转化。

虽然现在我们拥有了很多形式化方法的理论，但是各种理论在有其优势的同时，也存在着很多的不足。这就需要我们不断的研发出更完善、易于实用的新方法，同时对已有方法进行改进、整合并加以完善，以发挥出它们新的作用。后者是本文工作的重点所在。

1.5、研究工作概要及论文安排

本文将对安全协议的形式化分析、设计及其自动化方法进行研究。具体的研究内容如下：

第二章：

从符号理论与计算理论两方面对安全协议的形式化分析作了详细讨论：

- 介绍了符号理论下有关形式化分析的传统分类方法，并对符号理论所固有的缺陷作了讨论。
- 介绍了计算理论下具有代表性的形式化分析方法，并对计算理论存在的缺陷作了讨论。
- 对形式化分析技术的发展前景作了展望。

第三章

- 对 CS 逻辑进行了简要介绍。
- 讨论了原有 CS 逻辑及其后来改进方案的若干缺陷。
- 对 CS 逻辑进行了修正及其扩展。

第四章

- 讨论了 BAN 类逻辑存在的若干不足。
- 发现了敌手的一种新的攻击能力，然后对 Dolev-Yao 敌手模型作了补充。
- 基于推理逻辑，提出了一种新的形式化分析工具—MBL 逻辑，并对该逻辑进行了详细介绍。
- 给出了使用 MBL 逻辑的一个分析实例。
- 讨论了 MBL 逻辑的可扩展性。

第五章

- 详细介绍了目前已有的有关形式化分析的自动化技术。
- 实现了基于 MBL 逻辑的一种自动化分析技术，并给出了一个实例分析的部分结果。

第六章

以计算理论为主，讨论了设计安全协议所需注意的因素。

- 介绍了一些加密体制的安全性概念。并讨论了安全协议在采用公钥体制和对称钥体制时各适合采用何种安全级别的加密算法。
- 讨论了设计安全协议时所必需的消息元素，以及它们所需的安全保护。
- 根据我们所研究的成果，对 Yahalom 协议与 Needham-Schroeder 公钥认证协议进行了改进。

第七章

主要讨论了在计算理论下如何分析协议的认证性。

- 提出了两个攻击游戏，并通过主体在游戏中的成功概率来刻画协议的认证性。
- 分析并指出对于采用单向函数设计的协议，为了抵抗敌手通过密码分析破解密文，在公钥体制下采用 IND-CCA2 安全的加密算法为最佳；而在对称钥体制下采用 IND-CPA 安全的、基于伪随机函数的概率加密方案即可。
- 在计算理论下给出了关于改进的 Needham-Schroeder 对称钥认证密钥交换协议认证性的分析。

第八章

鉴于符号理论与计算理论各有优缺点，调和符号理论下的模态逻辑法与计算理论下的矛盾归约法，提出了一种新的分析方法。

- 给出了协议通信过程形式化描述，定义了协议的完全性与正确性概念。
- 对 MBL 逻辑的若干推理规则进行修改或扩展，提出了调和分析法。
- 给出了一个使用调和方法分析安全协议的实例。

第九章

- 全文总结，指出了进一步研究的方向。

第二章 安全协议的形式化分析方法

安全协议的形式化分析方法存在两种截然不同的理论：符号理论和计算理论。符号理论不考虑具体的算法，只将协议消息表示为简单明确的符号，然后利用适当的形式化工具利用这些符号进行分析；计算理论以计算复杂性为基础，通过衡量敌手攻击协议成功的概率与计算代价来判断协议的安全性。下面我们以这两种理论为主线，介绍各种安全协议的形式化分析方法。

2.1、符号理论

符号理论是安全协议形式化方法中普遍采用的一种理论，从上世纪八十年代至今，很多形式化方法都出自符号理论。这种理论中的形式化分析法有一个共同点[59-64]，即在形式化过程中均以符号来表示协议，比如主体用符号 A、B 等来表示；*nonce* 用 *Na* 或者 *Nb* 来表示；密文由 $\{M\}_K$ 来表示，其中 M 代表明文，K 代表加密密钥。这种表示方法基于 Dolev-Yao 理论，该理论将认证协议本身与认证协议所采用的密码体制分开，在假定密码体制是“完善”的基础上讨论认证协议的正确性、安全性、冗余性等问题，这样可保障研究人员专心地研究安全协议的内在安全性质。

由于这种理论非常简单直观，方便协议设计者和使用者的理解，因此广泛应用于安全协议形式化分析的各种方法中。到目前为止，研究人员已经发展出多种基于符号理论的形式化分析法，对此进行综述的文章也很多[65-69]，其中文献[65]的分析最详细全面，我们将采用该文提出的分类法。该分类法将符号理论下的形式化分析法具体划分为：基于推理结构的方法、基于攻击结构的方法和基于证明结构的方法。

2.1.1、基于推理结构的方法

基于推理结构方法的主要特征是有一个完整的逻辑系统，该系统由推理规则、公理、语义模型、计算模型等部分组成。该方法将知识集合和信念集合表示为消息某些性质的函数，并通过推理规则或公理的形式表达出来。在进行协议的

安全性分析时，通过推理规则或公理跟踪主体的知识集合和信念集合的变化，通过发现两个集合的异常情况来判断协议的安全性。其中，推理规则可分为两大类：基于知识的推理[70]、基于信念的推理[70,71]。

在基于推理结构的方法中，最著名的莫过于 1989 年由 Burrows、Abadi 和 Needham 提出的基于信念的 BAN 逻辑。BAN 逻辑是安全协议形式化分析的一个里程碑，正是因为它的出现引起了人们对这一领域的广泛兴趣，从此形式化分析才逐渐成为信息安全的一个热点。

BAN 逻辑进行形式化分析的步骤为：

(1) 对协议进行理想化处理，该过程挖掘出协议消息的内在含义，并用形式化的方法表达出这些含义。

(2) 对系统的初始状态进行描述，建立初始假设集合。

(3) 给出协议的最终目标。

(4) 通过推理规则或公理对理想化处理后的协议消息和前提假设进行逻辑推理，并得出最终各主体的信念或知识。

(5) 对各主体的最终信念或知识进行考察，从而判断协议是否符合最终目标，或者存在安全隐患。

BAN 逻辑的特点是“简单实用”，能够揭发安全协议的设计缺陷，比如曾成功地发现了 CCITT X.509 标准[73]推荐草案中的安全漏洞。但是 BAN 逻辑过于简单，只讨论诚实合法主体的认证问题，且抽象级别太高，没有明确的语义模型来证明推理规则的正确性。

BAN 逻辑之后，许多类似逻辑相继提出，旨在弥补其缺陷。比如 GNY 逻辑，它是对 BAN 逻辑的扩充，有超过 40 条的推理规则。与 BAN 逻辑相比，增加了“拥有密钥”的表达式，能区分主体自己的消息和外来的消息，能分析某些应用单向函数的协议等等。AT 逻辑有一个详细的计算模型，首次给出了模型论语义，并抛弃了 BAN 逻辑中语义和实现细节的混合部分。此外，在 AT 逻辑中只有两条基本推理规则，其逻辑公理都是从这两条规则中延伸而出的。SVO 逻辑集中了 BAN 逻辑，GNY 逻辑和 AT 逻辑的优点，并加以改进，是以上逻辑的集大成者。它吸收了 AT 逻辑中语义模型和计算模型的概念，并对其中的一些概念进行了重新的定义，从而取消了 AT 逻辑中的一些限制。

以上提到的逻辑都是从 BAN 逻辑发展而来，它们有一个统一的名字：BAN

类逻辑。除此之外，还有其他的一些逻辑，比如 Kailar 逻辑[74]，主要用于电子商务协议，能验证协议的可追究性；CS 逻辑，一种将时间与逻辑相结合的系统，可验证与时间相关的协议。这些基于推理结构的方法有如下一些弱点,比如语义说明还不够明确；某些公理存在缺陷；理想化方法依赖于协议分析者对协议的理解能力，较难正确实现；只能发现已知的攻击等等。

2.1.2、基于攻击结构的方法

该方法的思路是考察协议的状态，将协议中的状态转换以路径的形式来表示，然后从协议的初始状态出发，对合法主体和敌手的所有可能路径进行穷尽搜索，以期找到协议可能存在的错误。由于涉及到的状态变化太多，用人工的方式进行分析不现实，因此该类方法一般需要借助于自动工具来完成，如模型检测工具 FDR[75],NRL 分析器[76,77]和 Interrogator[78]。

在这类方法中，很多最初目的并不是分析安全协议的，但是随着安全协议形式化分析研究的日益升温，这些方法逐渐应用到安全协议的分析中。

比如 Model Checking 方法[79]，最早用于分析和模拟硬件工作的过程，但如今在安全协议的形式化分析中占有了一席之地。Model Checking 在对协议进行建模时，不仅需要考虑主体行为序列，还要考虑敌手的行为序列。而模型的状态包括协议中参加的主体、每个主体的状态和秘密信息的集合。模型中的路径就是模型状态和行为的交替有限序列。通过深度优先搜索我们不仅能检查不符合安全说明的状态能否到达，而且在协议不安全时还能给出敌手的攻击路径。

通信顺序进程 CSP(Communicating Sequential Processes)是 1985 年由 Hoare 为解决并发现象而提出的一种进程代数[80-82]。最初是一个专门为描述并发系统中通信主体的交互行为而设计的工具。近年来，CSP 方法也逐渐应用到安全协议的分析与设计上。1996 年英国学者 Gavin Lowe 就首先启用 CSP 的模型检测工具 FDR，分析并发现了 Needham-Schroeder 公钥协议的一个近 17 年来未知的并行会话攻击缺陷[75]。

在 CSP 中，一个协议系统是由该协议所能运行的事件来建模的。所有可能的事件用 Σ 来表示。进程是系统的一个组成部分，它描述了系统可能发生的行为序列，比如 Stop 表示没有任何事情发生；$c!v \rightarrow P$ 表示在信道 c 上输出 v，然后

象 P 一样行动。进程的语义是通过可能发生的事件序列的集合来表示的，而这里的事件序列称为路径。

CSP 中的保密性被描述为：除了意定的接收者，任何主体都不能从消息中计算出任何消息。而 CSP 的认证性采用的是消息趋向法(message- oriented)[57]：若 T 出现意味着 P 必先出现，则说明 T 认证 P。在进行安全协议的分析时，CSP 首先对协议进行建模，这包括用 CSP 语言描述主体进程和敌手进程；然后用 CSP 语言描述协议的安全性质，包括认证性和保密性；最后验证协议的安全性质是否满足。

由于用 CSP 方法来描述协议十分复杂，若采用人工的方式既浪费时间又容易出错，因此一般使用称为 Casper 的编译器[83]来自动生成协议的 CSP 描述。用户首先制作一份描述协议的脚本作为 Casper 的输入，然后使用 Casper 将其编译成 CSP 代码。FDR 作为一种商业用的模型检验机，可用来得到 CSP 代码中所描述的并发反映结果。

基于攻击结构的方法能有效发现协议中的错误，并且能采用自动化技术来加快安全协议的分析。但该领域也存在着一些突出的问题：只能分析有限主体数目的协议；难以解决状态空间爆炸的问题。

2.1.3、基于证明结构的方法

如前所述，推理结构法缺少明确的语义说明，有时很难明确信念逻辑究竟证明了什么。而攻击构造法在分析复杂的协议时，很容易造成状态空间的爆炸，使得分析协议时所需的时间空间超出了我们能提供的资源。

为了克服上述困难，Paulson 提出了一个用协议消息和事件的攻击构造法标注的结构性证明法，称为 Paulson 归纳法[84]。在进行分析时，它首先将协议归纳定义为所有可能事件路径的集合，并借鉴信念逻辑法来形式化地表现主体接收消息的内在含义，然后通过状态探测法对所有可达的状态进行穷举搜索，以证明协议的安全性。由于 Isablelle 自动定理证明机[85]支持归纳性定义的集合，因此证明过程可部分由 Isablelle 自动实现。

1997 年提出的串空间"strand space"也是一种著名的证明结构方法。该方法将合法主体或敌手行为的序列表示成串，串空间是所有主体的串集合。从作为串空

间的一部分，用通过收发消息而相连的多个串来表示。串空间机制包括一个描述因果结构的偏序和一个类递归证明方法，并使用插图法来描述和推理协议的正确性，可使分析者描绘出安全协议的执行画面、受到的攻击、正确性定理以及证明中的关键步骤等。随着串空间理论的发展，1999 年 Song 通过扩展串空间模型，结合定理证明和模型检测技术，开发了一个自动化证明工具 Athena[86]，它能利用 Strand 空间模型筛选出那些能够实现其设计规范的协议。Athena 将只是顺序不同的状态视为一个状态，然后进行并行推理，从而可避免状态爆炸问题。2000 年，Guttman 等提出了认证测试法[87]，该方法在分析协议时比单纯应用串空间模型更为简捷直观，而且能用于协议的设计。

2.1.4、符号理论的缺陷

随着安全协议形式化研究的深入发展，人们逐渐发现采用符号理论来表述和分析安全协议有诸多不便。比如，若在设计协议时，一条加密消息表示为 $\{M\}_K$，那么协议使用者在执行协议时，究竟采用哪种加密算法来实现密文呢？在符号理论下，有时看到的情况是加密和签名可“相互抵消”，但是既使是教科书式版本，也很少有公钥算法会具备此特点。在更多的情况下，由于符号理论中对密码算法的描述过于抽象，协议的使用者很容易采用确定的算法来实现加密操作，而这样就会有利于敌手进行统计分析。

符号理论的缺点不仅在此，由于符号理论将复杂的密码算法抽象为简单符号，完全屏蔽了算法的具体实现细节，因此在安全协议的分析中存在着如下弱点。

(1) 假设密码体制是完善的，可提供理想化的密码服务

对于密文 $\{M\}_K$，在符号理论下认为其采用的加密体制是完善的，这种观点揭示了这样一种直觉：对于一条加密信息，只有知道密钥的人，才能得到相应的明文，其他人得不到关于明文的任何有用信息，这就是所谓的“all-or-nothing”安全性，这个概念不考虑敌手的主动攻击，也不考虑敌手能否通过极低的概率或者极高计算代价获得一些有关明文的有用信息。

因此采用符号理论的形式化分析经常被警告既使分析结果是安全的，也不能保证协议的安全性，因为一个加密方案在形式化模型中的表达与它的实现之间有很大差距。在实际中，敌手并非总是一无所知，比如他能通过网络监听发现是否

有数据流的产生，从而推断出某个特殊事件的发生；他也能通过相关环境推断出明文空间的范围，比如银行的账户密码通常是6位的数字组成，通过这些信息，敌手采用特殊的手段，通过穷举的方式就能破译密码。

(2) 在符号理论中考虑的敌手的攻击主要是被动攻击

在符号理论中，不考虑敌手运用自己掌握的数据操纵或篡改密文，也不考虑敌手攻击前后要求拥有密钥的用户提供加密或解密服务。

但在现实中，我们永远都不能期望敌手是善意的和被动的，也不能低估敌手的攻击能力。强大的敌手能运用任何手段来攻击协议，下面我们举个敌手主动攻击的例子：

当用户采用具有良好代数结构的加密算法来加密消息时，敌手可能就会针对密文采用可展性攻击。所谓可展性攻击[88]，就是敌手以一种有意义的方式来修改密文，从而使解密出的明文与原明文有某种相关性。例如，在一次投标过程中，所有投票者根据协议要求都需要加密自己的价格，投标者 A 将自己的报价 100 万通过加密发给了招标单位。但是如果投标协议所采用的加密算法具有可展性，那么 A 的商业对手 E 就能在网络中拦截这个密文，并通过可展性攻击修改它，使得招标单位解密出的明文价格是原来价格的一半。那么当招标单位解密出 A 的报价时，这个报价就会变成 50 万，这样一来，A 的竞争力就大打折扣了。在这个过程中，E 并不知道 A 的报价是多少，也就是说密文仍是保密的，但他仍可以通过篡改密文来造成攻击。

这个攻击说明，在协议面对具有主动攻击的敌手时，既使采用了具有良好保密性的加密算法也是不安全的。

在其他一些主动攻击中，敌手可作为一个合法的主体参与同目标用户的交互，发送密文给目标用户，得到相应的明文。这种交互可以看作用户为敌手提供的解密预言服务，敌手能通过这种服务来进行解密训练。如果协议的使用者在执行协议时使用的加密算法不具有高强度的安全性[47]，那么敌手通过这种解密训练课程就能以很大的概率解密出目标密文。

从上面的分析可以看出，由于符号理论将密码算法过于抽象，导致了与其实现的脱节，使得在进行形式化分析时忽略了对密码算法安全性的关注。因此现在普遍的看法认为采用符号理论判定为安全的协议不能作为协议安全的有力证据。

2.2、计算理论

计算理论[89-95]与符号理论不同，它侧重于密码分析，通过评估敌手攻击协议的成功概率和计算代价来判断协议的安全性。通常的做法是采用“矛盾归约”的方法[96,97]，即对一个想要证明安全性的问题 P，构造一个有效的多项式时间的归约变换，将对问题 P 的有效攻击归约成一个计算复杂性理论中人们普遍相信困难问题的重大突破。正是由于人们广泛相信这个重大突破是不可能的，所以导致了与所谓攻击的存在性相矛盾，从而说明 P 是安全的。也就是说，证明是通过矛盾归约给出的。

在计算理论下，安全性的形式化证明包括以下三步：

1） 形式模型化协议参与者和敌手的行为：该模型化通常是以敌手和攻击目标之间进行攻击游戏的形式给出的。

2） 安全性目标的形式化定义：这里定义敌手在攻击游戏中的成功，通常是以不可忽略的概率和可承受的时间复杂度公式的形式给出的。

3） 形式化证明一个多项式时间的归约，把对给定目标的所谓攻击归约到计算复杂性理论中的一个认为不可能的突破。

下面我们给出计算理论下具有代表性的一个例子。

2.2.1、模式方法

1998 年，Bellare,Canetti 和 Krawczyk 提出了采用模式方法[98]来分析安全协议。这种方法考虑了两个系统：理想系统和实际系统。其安全理论在于：敌手在与实际系统交互时所得的结果，与敌手在与理想系统交互时得到的结果一样多。这样从理想模型的安全性就能保证实际系统的安全性。

- 敌手模型

在协议的运行中，敌手控制着所有主体的执行和网络的通信，所有协议中发出的消息都由敌手进行传递。他能决定哪一个主体将要运行，可制定主体接收哪条消息和外部请求，并能获得协议中所有输出的消息。

在认证模式下，敌手被要求诚实地传递数据，即诚实地将数据从发方传递到收方。但是敌手可改变消息传递的次序，延迟消息的传输，还可收买主体以获得

其内部信息。

在非认证模式下，敌手的行为与认证模式下相似，区别在于敌手不再诚实地传递数据，他可将数据进行任意地处理后传递给收方并能在信道中插入虚假消息。

- 认证器：

模式方法的核心是构造一个通用编译器 C，它可将一个理想认证模型下的协议 π 转化为协议 $\pi'=C(\pi)$，π' 可完成与 π 相同的功能，但它运行在实际的环境中，面临非认证模式下的敌手。这样的编译器称为认证器。

认证器的定义涉及协议的认证和非认证两种模式的形式化定义，并且还需定义协议在这两种模式中运行的一个等价关系。

- 分析方法：

在模式方法中，证明协议的安全性是通过认证器来实现的。首先设计一个以非认证方式来发送数据的认证器 C。然后对于一个在认证模式下安全的协议 π，通过认证器 C 得到一个在非认证模式下的协议 $\pi'=C(\pi)$，若能证明敌手在认证模式下攻击协议 π 得到的信息与在非认证模式下攻击 π' 得到的信息是计算不可区分或者统计不可区分的，那么可认为在非认证模式下的协议 π' 是安全的，同时认为该认证器是安全的。

引入认证器的好处是，由于它能被重用，因此若另一个认证协议是由同一个认证器转化而来，我们就不需要重复证明认证器的安全性了，这就简化了协议的证明。此外，认证器作为一个可重用的模块，还能简化安全协议的设计。

2.2.2、计算理论的缺陷

计算理论主要分析的是敌手对协议在密码分析上的攻击能力，若敌手能以很大的概率和较少的计算代价伪造协议消息成功，那么就认为协议是不安全的。但这考虑的只是协议设计时所采用密码体制的安全性，却忽略了对协议消息设计合理性的考察。也就是说，在计算理论下证明是安全的协议可抵抗敌手有关密码分析的攻击，但能否抵抗敌手的非密码分析的攻击，比如重放攻击，却是未知数。而这种非密码分析的攻击，考验的是协议在逻辑上的安全性，而这正是符号理论所考虑的。

2.3、安全协议形式化方法的新发展

在上面的论述中，我们展示了安全协议形式化分析中的符号理论和计算理论。这两种理论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，而大多数符号理论并没有真正建立起密码学的可靠性。但是双方各有优缺点，近年来，将这两种理论统一到一个框架中，建立形式化方法的密码学可靠性并弥补其不足成为人们研究的重点。

Abadi 和 Rowaway 作了开创先河的工作，他们定义了加密表达式的简单语言，证明了若两个表达式模逻辑公式等价，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[99]。Micciancio 和 Warinschi 进一步证明了，若使用充分强的加密方案，任何两个表达式等价当且仅当它们在逻辑下等价[100]。V.Gligor 和 D.Ohorvitz 精确刻画了这种等价成立对于加密方案的要求[101]。Micciancio 和 Warinshi 进一步给出了具有主动攻击敌手的密码协议安全性证明方法[102]。

下面我们主要介绍一下 Abadi 和 Rowaway 为统一这两种理论而做的工作。

2.3.1、Abadi 和 Rowaway 的可调和性思想

Abadi 和 Rowaway 提出的可调和性思想主要讨论的是形式化加密的计算正确性。

- **符号理论的等价概念**

形式化加密的等价概念是指：若两个命题对于某一主体在计算上是等价的，只要该主体对于这两个命题产生相同的表达式。而当敌手对一个密文和一个同等长度的随机串产生相同的表达式时，我们就认为该加密方案是安全的。

用符号□表示敌手不能解密的密文。我们定义一个“*pattern*”来作为协议表达式的扩展：

$P,Q :=$	*pattern*
K	*key*(*for* $K \in Keys$)
i	*bit*(*for* $i \in Bool$)
(P,Q)	*pair*
$\{P\}_K$	*encryption*(*for* $K \in Keys$)
□	*undecryptable*

引入"*pattern*"是为了描述表达式中不能被敌手解密的部分。

给定一个函数 p、一个密钥集 T 和一个表达式 M，我们通过函数 p，可将消息 M 归约到一个 *pattern*。*pattern* 有几种情况：*key*、*bit*、*pair*、*encryption*、*undercryptable*。直观上，若一个敌手含有 T 中的密钥，那么该敌手能看到 M 的 *pattern*。

$$p(K,T)=K$$

$$p(i,T)=i$$

$$p((M,N),T)=(p(M,T),p(N,T))$$

$$p(\{M\}_K,T)=\begin{cases}\{p(M,T)\}_K & \text{if } K\in T\\ \square & \text{otherwise}\end{cases}$$

更进一步，在没有额外的密钥集 T 时，可仅利用表达式所拥有的密钥信息来定义一个 *pattern*：$pattern(M)=p(M,\{K\in Keys\mid M\mid -K\})$。

直观上，敌手用在消息 M 中得到的密钥能看到该 *pattern*。

我们说若两个表达式的 *pattern* 是相等的，那么这两个表达式是等价的。

有这样一种情况，虽然 M 与 N 两个表达式不等价，但将 N 中的某些符号替换以后，就会与 M 等价了，比如两个表达式：$(\{0\}_K,K)$和$(\{0\}_{K'},K')$不等价，但将 K' 换成 K 后，两者就等价了。因此我们引入等价关系的一个替代品$\cong$：

$M\cong N$，当且仅当存在一个σ，使得 $M\equiv N\sigma$。

（σ 在这里可以看作 N 中符号的一个替换关系。）

● 计算的不可区分性概念

(1) 不可区分性：

称字串分布的集合 η 为一个系集。对于每一个 η 有一个系集 $D=\{D_\eta\}$。称 $x\xleftarrow{R}D_\eta$ 为 x 在系集 D_η 上的取样。

则称两个系集是计算不可区分的，若对于每一个概率多项式时间的敌手 E，有：$\varepsilon(\eta)=\Pr[x\xleftarrow{R}D_\eta:E(\eta,x)=1]-\Pr[x\xleftarrow{R}D_\eta':E(\eta,x)=1]$ 是可忽略的。记为：$D\approx D'$。

(2) 加密方案的三个特征：

循环隐藏：给定两个密文，若我们不能判断出它们的潜在明文是相等的，则称该加密体制是循环隐藏的[96]。

密钥隐藏：给定用不同密钥加密的信息，若我们不能判断出哪些信息是在同一个密钥下加密的，则该加密方案为密钥隐藏的。

信息长度隐藏：若密文不能揭示明文的长度，我们称为信息长度隐藏。

我们采用三个位来分别表示加密方案的三个特征，则可共有 8 种不同的加密方案。我们称具有循环隐藏、密钥隐藏和信息长度隐藏的加密方案为 type-0 方案。

下面我们给出 type-0 加密方案的计算表示。

令 $\prod=(K,\varepsilon,D)$ 是一个加密方案，η 是安全参数，如果对于任何概率多项式时间的敌手 E，有

$$Adv^{0}_{\prod[\eta]}(E)\overset{def}{=}\Pr[k,k'\xleftarrow{R}K(\eta):E^{\varepsilon_k(\cdot),\varepsilon_{k'}(\cdot)}(\eta)=1]-\Pr[k\xleftarrow{R}K(\eta):E^{\varepsilon_k(0),\varepsilon_k(0)}(\eta)=1]$$

且 $Adv^{0}_{\prod[\eta]}(E)$ 是可忽略不计的，则为 type-0 方案；

可以看出，加密方案的定义是通过计算不可区分性来定义的。

- 表达式等价的计算不可区分性

通过表达式向系集的转化，我们可将等价与计算不可区分性相联系。分两步进行处理，第一：给定一个加密方案，将一个系集与一个表达式 M 联系起来；第二：我们说明在合适的假设下，等价表达式能够得到计算不可区分性。

(1) 将一个表达式与一个系集联系起来

令 $\prod=(K,\varepsilon,D)$ 是一个加密方案，$\eta\in parameter$ 是一个安全参数。我们在每一个形式化表达式 $M\in Exp$ 上关联在字串 $[\![M]\!]_{\prod[\eta]}$ 上的一个分布。这种关联可通过算法 *Convert*(*M*)来执行：

第一，将每一个 M 中的 K 用密钥产生器 $K(\eta)$ 映射到一个串 $\tau(K)$。

第二，将形式化的 0，1 串映射到标准的字串。

第三，通过串联 M 的映射和 N 的映射，来得到形式化对(*M*,*N*)的映射。

第四，通过计算 $\varepsilon_{\tau(K)}(x)$ 来得到形式化表达式 $\{M\}_K$ 的映射。其中 x 是 M 的映射。

第五，为了避免混淆，所有的字串都标注上类型。

(2) 等价关系意味着不可区分性

如果表达式 M 和 N 有关系：$M\cong N$，那么我们通过算法 *Convert*(*M*)能将 M 和 N 转化为对应的系集 $[\![M]\!]$ 和 $[\![N]\!]$，则两个系集是不可区分的。具体的证明可见文献[99]。

2.4 本章小结

在本章中，我们对安全协议形式化分析领域的发展动态作了全面而客观的评

述，以符号理论与计算理论为两条主线，介绍了各种形式化方法的发展及其分类。对于每一种观点，我们都选择具有代表性的分析方法进行了介绍，并讨论了其优缺点。此外，我们还介绍了该领域研究的新热点。

第三章 对CS逻辑的改进与扩展

3.1、CS逻辑的介绍

CS逻辑由Coffey和Saidha于1997年提出，是一种将时间与模态逻辑相结合的逻辑[58]，可用来分析有关公钥协议的安全性。

3.1.1、基本介绍

CS逻辑采用了一种将知识和信念相结合的技术，能够分析协议执行过程中知识与信念的变化，对于评估协议的保密性与信任关系十分有用。在CS逻辑中有一个信念操作符和两个知识操作符。其中信念操作符用来处理主体的信任关系；一个知识操作符用来处理协议中有关陈述或事实的知识，另一个知识操作符用来处理有关消息对象的知识。

CS逻辑中的推理规则是有关自然演绎的标准推理规则。CS逻辑中的公理反映的是公钥协议的基本属性，例如，表述了主体在掌握密钥后的加解密能力。这些公理反映了逻辑中的潜在假设：

1) 通信环境充满敌意，会受到攻击；数据传输系统是可靠的，在没有敌手干预的情况下，不会发生数据丢失或者传输错误。

2) 公钥体系是理想的，即在没有对应密钥的情况下，加密和解密是不可逆转的；密码系统是碰撞自由的，即两个不同明文不可能产生一样的密文。

3) 系统中的公钥在失效前是有效的，对应的私钥是保密的。

4) 若一条数据被加密或者解密，那么执行加解密的主体一定知道该数据。

3.1.2、CS逻辑的符号

惯例上常用Σ和Ψ表示任意主体，ENT表示所有主体；k_Σ和k_Σ^{-1}分别表示主体Σ的公钥和私钥；$e(x,k_\Sigma)$表示公钥的加密运算；$d(x,k_\Sigma^{-1})$表示私钥的运算；$K_{\Sigma,}\phi$表示主体Σ在t时知道表达式ϕ；$L_{\Sigma,}x$表示主体Σ在t时知道并能有意识

地重新产生消息 x； $B_{\Sigma,t}\phi$ 意味着主体 Σ 在 t 时相信表达式 ϕ； $S(\Sigma,t,x)$ 意味着在 t 时主体 Σ 发送了消息 x； $R(\Sigma,t,x)$ 意味着在 t 时主体 Σ 接收了消息 x。

3.1.3、CS 逻辑的推理系统

CS 逻辑的推理系统由一组公理和推理规则组成。公理是一些命题，它们描述了有关逻辑的或目标系统的事实。推理规则用来从已知命题推出新命题，它由一些前提和可从这些前提中推理出的结论组成。

一. CS 逻辑的推理规则

整个 CS 逻辑只有两种基本推理规则：(1) *R1* 为 *MP* 规则(*modus ponens*)：由 p 和 $p \rightarrow q$ 能推导出 q；(2)*R2* 为 *Nec* 规则(*necessitation*)：由|- p 能推导出 $B_{\Sigma,t}p$ 和 $K_{\Sigma,t}p$ 来[52;103]。

从这两个基本推理规则又可得出五个推理规则[104]：

R3: *from* $(p \wedge q)$ *infer* p

R4: *from* p *and* q *infer* $(p \wedge q)$

R5: *from* p *infer* $(p \vee q)$

R6: *from* $\neg(\neg p)$ *infer* p

R7:*from* (*from* p *infer* q) *infer* $(p \rightarrow q)$

二. CS 逻辑的公理

公理是应用到一个系统中的为真的命题，它有两种类型：逻辑公理和非逻辑公理。

逻辑公理是可应用到任何系统的一般命题。非逻辑公理是对特定系统隐藏特性的形式化描述，在这里是对公钥体制协议特性的描述。

(1) 有关的逻辑公理：

A1:*MP* 公理

(*a*) $\exists t,\exists p,\exists q(K_{\Sigma,t}p \wedge K_{\Sigma,t}(p \rightarrow q) \rightarrow K_{\Sigma,t}q)$

(*b*) $\exists t,\exists p,\exists q(B_{\Sigma,t}p \wedge B_{\Sigma,t}(p \rightarrow q) \rightarrow B_{\Sigma,t}q)$

公理 *A1* 中的(*a*)和(*b*)是 *MP* 规则对于知识和信念的应用。

A2:知识公理

$\exists t,\exists p(K_{\Sigma,t}p \rightarrow p)$

公理 *A2* 刻画了知识的特征，即知道即为真，这个特征是区分知识与信念的关键。

A3：单调性公理

$(a):\exists t,\exists x,\exists i,i\in\{ENT\},(L_{i,t}x \rightarrow \forall t' \geq t \quad L_{i,t'}x)$

$(b):\exists t,\exists x,\exists i,i\in\{ENT\},(K_{i,t}x \rightarrow \forall t' \geq t \quad K_{i,t'}x)$

$(c):\exists t,\exists x,\exists i,i\in\{ENT\},(B_{i,t}x \rightarrow \forall t' \geq t \quad B_{i,t'}x)$

公理 *A3* 描述了知识和信念的单调性，即一旦获得，便不会丢失。

A4:子消息公理

$\exists t,\exists x,\exists y(\exists i,i\in\{ENT\},L_{i,t}y \wedge C(y,x) \rightarrow \exists j,j\in\{ENT\} \quad L_{i,t}x)$

公理 *A4* 描述了若一条消息是其他消息的子消息，则肯定有人知道这条消息。

(2) 有关的非逻辑公理：

A5：发送消息公理

$\exists t,\exists x(S(\sum,t,x) \rightarrow L_{\Sigma,t}x \wedge \exists i,i\in ENT\setminus\{\sum\},\exists t' > t,R(i,t',x))$

公理 *A5* 描述了若∑在 t 时刻发出了一条消息，则∑在 t 时刻知道该消息，且在 t 时刻后，除∑外，肯定有某一主体收到了该消息。

A6：接收消息公理

$\exists t,\exists x(R(\sum,t,x) \rightarrow L_{\Sigma,t}x \wedge \exists i,i\in ENT\setminus\{\sum\},\exists t' < t,S(i,t',x))$

公理 *A6* 描述了若∑在 t 时刻收到一条消息，则∑在 t 时刻知道该消息，且在 t 时刻前，除∑外，肯定有某一主体发送了该消息。

A7：加解密能力公理

$(a)\ \exists t,\exists x,\exists i,i\in\{ENT\},(L_{i,t}x \wedge L_{i,t}k_{\Sigma} \rightarrow L_{i,t}(e(x,k_{\Sigma})))$

$(b)\ \exists t,\exists x,\exists i,i\in\{ENT\},(L_{i,t}x \wedge L_{i,t}k_{\Sigma}^{-1} \rightarrow L_{i,t}(d(x,k_{\Sigma}^{-1})))$

公理 *A7* 中的(*a*)和(*b*)反映了主体的加解密能力。即当他有公钥或者私钥时，则能进行相应的加解密运算。

A8：加解密公理

(*a*) $\exists t,\exists x,\exists i,i\in\{ENT\},(\neg L_{i,t}k_{\Sigma}\wedge\forall t',t'<t\ \ \neg L_{i,t'}(e(x,k_{\Sigma}))$
$\wedge\neg(\exists y(R(i,t,y)\wedge C(y,e(x,k_{\Sigma}))))\rightarrow\neg L_{i,t}(e(x,k_{\Sigma})))$

(*b*) $\exists t,\exists x,\exists i,i\in\{ENT\},(\neg L_{i,t}k_{\Sigma}^{-1}\wedge\forall t',t'<t\ \ \neg L_{i,t'}(d(x,k_{\Sigma}^{-1}))$
$\wedge\neg(\exists y(R(i,t,y)\wedge C(y,d(x,k_{\Sigma}^{-1}))))\rightarrow\neg L_{i,t}(d(x,k_{\Sigma}^{-1})))$

公理 *A8* 中的(*a*)和(*b*)指出没有关于正确密钥的知识就不能进行加解密运算。

A9：私钥保密公理

$\forall t(\forall i,i\in\{ENT\},L_{i,t}k_{i}^{-1}\wedge\forall j,j\in ENT\setminus\{i\}\neg L_{j,t}k_{i}^{-1})$

公理 *A9* 表达了私钥的私有性。

A10：解密消息公理

$\exists t,\exists x(\exists i,i\in\{ENT\},L_{i,t}d(x,k_{\Sigma}^{-1})\rightarrow L_{\Sigma,t}x$

公理 *A10* 表达了私钥的所用者应该知道所有用该私钥解密的消息。

3.2、CS 逻辑及已有扩展的缺陷

3.2.1、CS 逻辑的缺陷

1）描述缺陷

公理 *A8* 描述的是主体加解密的能力，其目的是让一方主体 *A* 能判断另一方主体 *B* 能否进行加解密操作，但该公理在逻辑推理中并无实际价值，因为在并行运行协议的环境中，*B* 能同时运行若干协议，*A* 无法判断 *B* 接收的所有消息中是否已包含加密后的密文或者解密后的明文。

2）公理缺陷

第一，公钥具有公开性，任何主体可获得任何公钥，进而产生 $e(x,k_{\Sigma})$，而公理 *A8*(*a*)违背了公钥的公开性。第二，公理 *A7*、*A8*、*A10* 表达有关加解密的含义时有重复，而且将公理 *A8*(*b*)理解为有关解密的公理时并无实际意义。第三，未对信念与时间相关的不可知性进行描述。第四，未对与发送和接收密文有关的知识和信念进行描述，这使得 CS 逻辑对于一些知识和信念的推理十分冗长。比如因为没有签名消息的推理公理，对于 $d(x,k_{\Sigma}^{-1})$ 这样一条消息，要经过若干步推理才能得出主体∑曾经发出过该消息。第五，公理中缺少知识与信念间的转化，

在推理时某些结论只能通过非形式化的手段的得到。

3）其他缺陷

没有考虑敌手 E 的知识和信念。在形式化分析中，考察敌手能够从协议中得到什么样的知识与信念，能更清晰全面地分析协议的安全性。

3.2.2、已有扩展的缺陷

2003 年范红等人对 CS 协议进行了一些改进[105]，这些改进包括：时间单调性规则的改进、包含规则的改进、发送和接收规则的改进、消息获取规则的改进等等，使其能分析 *Timed-Release* 公钥协议[106]的正确性。但是改进后的逻辑公理仍有若干不合理的地方：

1）发送规则和密钥相结合的推理公理 *I5*(*b2*)：

$S(\Sigma,t,e(x,K_i)) \rightarrow B_{i,t'>t}\exists j \in ENT \setminus \{\Sigma,i\} \neg K(j,t',x)$

该公理只有在消息 x 是新鲜的情况下才成立，而逻辑中并没有给出 x 是新鲜的声明。

2）时间单调性规则中的时间段的拆分规则 *I2*(*c2*)：

$\forall tn \le t \le ts \quad K(B)_{i,t}x \rightarrow \forall tn \le t' \le tm \quad K(B)_{i,t'}.x \wedge \forall tm \le t'' \le ts \quad K(B)_{i,t''}.x$

该规则是有歧义的。因为若 tn 为协议开始时刻，ts 为协议结束时刻，该逻辑的描述为：主体 i 在协议运行的 t 时知道消息 x，那么主体 i 在协议中的任意时刻都知道 x。这违反了知识与信念的单调性。

3.3、我们对 CS 逻辑的扩展

这一节，我们在 CS 逻辑的基础上不仅对某些公理进行了改进和扩展，使其更加准确地反映基于公钥的逻辑的事实，而且增加了一些逻辑符号和推理公理，使其还能反映基于对称钥的逻辑的事实。

3.3.1、增加的符号

为了清楚表达扩展后的 CS 逻辑，我们增加了几个符号：

$x = x_1 \| x_2 \| ..... \| x_n$ 表示 x 由消息 $x_1, x_2, ..., x_n$ 串联组成。

$\otimes x$ 表示在一轮协议中消息 x 是保密的。

$\otimes_t x$ 表示消息 x 在时段 $[t_0,t]$ 中是保密的，其中 t_0 表示协议运行的起始时间。

$\oplus x$ 表示 x 是公开的。

*Pub(x)*表示消息 x 中的公开部分。

*Fresh(x)*表示消息 x 是新鲜的，即 x 在本轮协议运行前没有出现过。

E 表示敌手，他能控制整个网络中消息的传播。可认为传播中的消息均被 E 截获，再由 E 转发给接收者。

Σ 和 Ψ 表示任意合法主体，*ENT* 表示包括敌手 E 在内的所有主体。

还需要对协议中消息的传播标注几个时间点。标记规则为：$ti(m_i)ti+1$，表示发送消息时间若为 ti，则接收者收到消息的时间为 $ti+1$。此外，敌手 E 作为一个特殊主体在 $ti<t\le ti+1$ 时间内能够得到消息 m_i。对于一个合法主体而言，m_{i+1} 消息的发送时间等于消息 m_i 的接收时间。

此外，我们在协议中隐含用正确的密钥进行解密操作，而将 $d(x,k_\Sigma^{-1})$ 明确定义为签名数据，即 $d(x,k_\Sigma^{-1})=(x,f_{k_\Sigma^{-1}}(x))$，其中 x 是消息，$f_{k_\Sigma^{-1}}(x)$ 是签名值。我们可认为签名数据中包含签名值和被签名的明文消息两部分，这对于大部分的签名运算(签密除外)是合理的。

3.3.2、对已有逻辑公理的改进和扩展

- 对 *A1* 的扩展：

e): $K(B)_{\Sigma,t}(p)\wedge K(B)_{\Sigma,t}(q)\rightarrow K(B)_{\Sigma,t}(p\wedge q)$

f): $B_{\Sigma,t}\phi\rightarrow t'|t'>t,B_{\Sigma,t'}\phi$

g): $\neg L(B)_{\Sigma,t}x\rightarrow\forall t'<t\ \neg L(B)_{\Sigma,t'}x$

f)指出信念的单调性；*g*)指出若主体 Σ 在 t 时不相信或者不知道某消息，则在 t 时以前均不相信或不知道该消息。

- 对 *A5* 的修改和扩展：

a): 发送消息

$\exists t,\exists x(S(\Sigma,t,x)\rightarrow L_{\Sigma,t}x\wedge L_{E,t+1}x$

敌手 E 控制着整个网络，发出的消息 E 均可得到。但若 E 将该消息删除，则接收者收不到该消息。所以主体 Σ 发出消息 x 后，所能确定的是 Σ 和 E 可拥有消息 x。

b):发送签名消息

$$\exists t,\exists x(S(\Sigma,t,d(x,k_{\Sigma}^{-1}) \rightarrow L_{\Sigma,t}x \wedge L_{E,t+1}x$$

同理 *a*)。

c):发送公钥加密的消息

$$\exists t,\exists x\ \ K_{\Sigma,t}(S(\Sigma,t,e(x,k_B) \wedge fresh(x)) \rightarrow B_{\Sigma,t}(\forall i,i \in \{ENT/\Sigma,B\}\ \ \neg L_{i,t+1}x))$$

当 Σ 使用公钥 k_B 加密消息 x，且 x 是新鲜的时，在 t+1 时刻，Σ 相信除了 Σ 和 B 外，没有其他主体(包括 E)能从密文中有意识地重新产生 x。该公理之所以描述了从知识到信念的转化，是因为该公理的成立还隐含着相应私钥没有泄漏的假设。

d)：发送对称钥加密的消息

$$\exists t,\exists x\ \ K_{\Sigma,t}((S(\Sigma,t,e(x,k_{\Sigma B}) \wedge fresh(x) \wedge \otimes_t k_{\Sigma B}) \rightarrow B_{\Sigma,t}(\forall i,i \in \{ENT/\Sigma,B\}\ \ \neg L_{i,t+1}x))$$

当 $k_{\Sigma B}$ 是 Σ 和 B 的保密共享对称密钥，如果主体 Σ 在 t 时刻发送了加密消息 $e(x,k_{\Sigma B})$,且 x 是新鲜的，那么在 t+1 时刻，Σ 相信除了 Σ 和 B 外，没有其他主体(包括 E))能从密文中有意识的重新产生 x。

- 对 *A6* 的修改和扩展：

a):接收消息

$$\exists t,\exists x(R(\Sigma,t,x) \rightarrow L_{\Sigma,t}x \wedge L_{Ewe,t}x \wedge \exists i,i \in \{ENT/\Sigma\}\exists t',t'<tS(i,t',x))$$

当主体 Σ 在 t 时收到一个消息 x，则他与敌手 E 在 t 时均拥有 x，且在 t 时以前一定有一个主体 i(包括敌手 E)发送了该消息。

注："t 时以前" 没有时间下界，包括本轮协议之前产生的消息。以下同。

b):接收用私钥签名的消息

$$\exists t,\exists x(R(\Sigma,t,d(x,k_B^{-1})) \rightarrow L_{\Sigma,t}x \wedge L_{E,t}x,\exists t'|t'<t,S(B,t',x))$$

当主体 Σ 在 t 时收到一个签名消息 $d(x,k_B^{-1})$，则他则他与敌手 E 在 t 时均拥有 x，且在 t 时以前主体 B 曾发送了该消息。

c):接收用公钥加密的消息

$\exists t,\exists x,\exists i \mid i \in \{ENT/\sum\}, R(\sum,t,e(x,k_i) \rightarrow \exists t' \mid t' < t, S(i,t',e(x,k_i))$

即当∑在 t 时收到其他主体的公钥加密的消息，则在 t 时以前一定有某个主体 i(包括敌手 E)发送了该消息。

$\exists t,\exists x(R(\sum,t,e(x,k_\sum)) \rightarrow L_{\sum,t}x \wedge \exists t' \mid 0 < t' < t, \exists i \mid i \in \{ENT/\sum\} S(i,t',e(x,k_i))$

即当∑在 t 时收到一个自己公钥加密的消息，则∑拥有该消息，且在 t 时以前一定有一个主体 i(包括敌手 E)发送了该消息。

d)：接收用对称密钥加密的消息

$\exists t,\exists x(R(\sum,t,e(x,k_{\sum B})) \wedge \otimes_i k_{\sum B} \rightarrow L_{\sum,t}x \wedge B_{\sum,t}(\exists t' \mid t' < t, S(B,t',x))$

即 $k_{\sum B}$ 是∑的对称密钥且没有被泄漏，若主体∑在 t 时接收到该加密消息，则∑相信除了∑和 B 外，没有任何主体能有意识地产生 x，且在时间 t 前的某一时刻 t'，主体 B 发送了该消息。

- **对公理 *A10* 的重新解释**

公理 *A10* 在文献[58]中解释为私钥 $k_\sum^{-1}$ 的拥有者∑知道用 $k_\sum^{-1}$ 解密密文。而我们将表达式 $d(x,k_\sum^{-1})$ 定义为签名数据，因此该公理被重新解释为：签名者知道被签名的明文。

此外，由于公理 *A8* 用来推理加密消息和签名消息的获取，而该公理有前述的缺陷，所以将其摒弃。我们扩展后的逻辑公理完全可推理与加密消息和签名消息有关的知识和信念。

3.3.3、新添加的公理

- **新鲜性公理**

消息 $x = x_1 \| x_2 \| ..... \| x_n$，且 $1 \le i \le n$，则

$fresh(x_i) \rightarrow fresh(x)$

$K_{\sum,t}(fresh(x)) \rightarrow \forall t' \mid t' > t, K_{\sum,t'}(fresh(x))$

即若在 t 时刻，∑知道 x 是新鲜的，则在本轮协议的 $t' > t$ 时刻，∑知道 x 是新鲜的。

- 保密性公理

令 $x = x_1 \| x_2 \| ..... \| x_n$，且 $1 \le i \le n$

在公钥加密的情况下：

$K_{\Sigma,t}(R(\Sigma,t,e(x,k_\Sigma)) \wedge x_i \notin pub_i(x) \wedge fresh(x)) \rightarrow B_{\Sigma,t}(\otimes x_i)$

即当∑收到一个公钥加密的消息 x，若∑知道 x 是新鲜的，且 x_i 不属于 x 中公开的部分，则∑相信 x_i 是保密的。（因为∑还要假设对方没有泄漏 x_i）。

在 k 是对称密钥的情况下：

$K_{\Sigma,t}(R(\Sigma,t,\{x\}_k) \wedge fresh(x)) \wedge B_{\Sigma,t}(x_i \notin pub_i(x) \wedge \otimes_i k)) \rightarrow B_{\Sigma,t}(\otimes x_i)$

即当∑收到一个对称密钥 k 加密的消息 x，知道 x 是新鲜的，若∑相信 k 是保密的且 x 为被公开，则∑相信 x_i 是保密的。

- 拥有对称密钥的判定公理

若 $t'>t$，则有：

$K_{B,t'}(S(B,t,\{M\}_{k_{ab}})) \wedge K_{B,t'}(fresh(M)) \wedge B_{B,t'}(L_{A,t'}(M))$

$\wedge B_{B,t'}(\otimes_i k_{ab}) \rightarrow B_{B,t'}(L_{A,t'}(k_{ab}))$

即，若 B 在 t 时发送了消息 $\{M\}_{k_{ab}}$，且在 $t'>t$ 时 B 知道 k_{ab} 是保密的、M 是新鲜的，且相信 A 拥有 M 时，B 认为 A 拥有对称钥 k_{ab}。

3.4、实例分析

3.4.1、一种对称密钥交换协议的分析实例

$1) A \rightarrow S : A, B, Na$	(M_1)
$2) S \rightarrow A : \{Na, B, k_{ab}, \{k_{ab}, A\}_{k_{bs}}\}_{k_{as}}$	$\{M_2\}_{k_{as}}$
$3) A \rightarrow B : \{k_{ab}, A\}_{k_{bs}}$	$\{M_3\}_{k_{bs}}$
$4) B \rightarrow A : \{Nb\}_{k_{ab}}$	$\{M_4\}_{k_{ab}}$
$5) A \rightarrow B : \{Nb-1\}_{k_{ab}}$	$\{M_5\}_{k_{ab}}$

为了下面分析时书写方便，将5个消息分别用M_1.....M_5来代替。

3.4.2、协议分析

扩展逻辑对协议的分析包括以下几步：

1)首先对协议中消息的流动进行时间标注。

2)提出该协议的假设和前提。

3)形式化说明协议将达到的目标。

4)运用CS扩展逻辑中的规则和公理、前提和假设开始推理，验证协议是否达成其最终目标。

第一步：时间标注如图1：

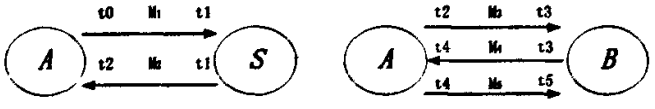

图3.1 协议消息的时序

将协议中合法主体收发消息的时间顺序如图3.1排序。

第二步：前提和假设条件：

1） 有关密钥的假设

$a1$：$\otimes k_{as}$，$\otimes k_{bs}$，$L_{A,t0}(k_{as}), L_{B,t0}(k_{bs}), L_{S,t0}(k_{as}, k_{bs})$

即：协议主体A、B和可信中心S在协议运行开始时知道自己的对称密钥，且这些密钥是保密的。

2） 有关随机数的假设

$a2$：$K_{A,t0}(fresh(Na)), K_{B,t3}(fresh(Nb))$

即：两个随机数都是本轮产生的，且主体A和B知道S产生新鲜的k_{ab}。

3） 其他假设：

$a3$：$L_{A,t0}(\oplus A, \oplus B, \oplus Na)$，$L_{B,t0}(\oplus A, \oplus B)$

即：标识符A、B和一次性随机数Na都是公开的。

第三步：协议的目标：

(i) $B_{A,t_{end}}(\otimes_{t_{end}} k_{ab} \wedge L_{B,t_{end}} k_{ab})$

(ii) $B_{B,t_{end}}(\otimes_{t_{end}} k_{ab} \wedge L_{A,t_{end}} k_{ab})$

即在协议结束时，A 相信对称密钥 k_{ab} 是保密的，且相信 B 拥有 k_{ab}；B 相信 k_{ab} 是保密的，且相信 A 拥有 k_{ab}。这样 A 和 B 就认证了对方，并可通过 k_{ab} 发送加密消息。

第四步：逻辑推理

证明：

1) 若在第二步主体 A 收到了消息，则有下面的结论：

$K_{A,t2}(R(A,t2,e(M_2,k_{as})))$，$K_{A,t2}(fresh(Na)) \rightarrow K_{A,t2}(fresh(M_2))$（新鲜性公理），$K_{A,t2}(k_{ab} \notin pub_{t2}(M_2))$。那么根据保密性公理，有下面的结论成立：

$B_{A,t2}(\otimes_{t2} k_{ab})$ [exp1]

2) 若在第三步主体 B 收到了消息，则下面的结论成立：

$K_{B,t3}(R(B,t3,e(M_3,k_{bs})))$，$K_{B,t3}(k_{ab} \notin pub_{t3}(M_3))$，但由于不能判断消息 M_3 是否新鲜，所以此时无法利用保密性公理来得出结论：

$B_{B,t3}(\otimes_{t3} k_{ab})$ [exp2]

而从下面的推理可知，没有[exp2]，我们得不出协议目标。现在假设[exp2]成立，以便我们可以继续分析协议。

3) 在第四步，主体 B 发送了消息 $\{M_4\}_{k_{ab}}$，则有 $K_{B,t3}(S(B,t3,\{Nb\}_{k_{ab}}))$，加上前提假设 $a2$，根据公理 $A5\text{-}c$，则下面的结论成立：

$B_{B,t3}(\forall i \mid i \in \{ENT/B,A\}, \neg L_{i,t4}(Nb))$ [exp3]

此外，若主体 A 在第四步得到了消息 $\{M_4\}_{k_{ab}}$，则 $K_{A,t4}(R(A,t4,\{M_4\}_{k_{ab}}))$ 为真。但从前提假设可以看出，A 并不知道 Nb 是否是新鲜的，不能确认 Nb 是否被重放过，且 A 不知道在 $[t2,t4]$ 时间区间 k_{ab} 是否被泄漏，所以根据保密性定理，推不出结论：$B_{A,t4}(\otimes_{t4} Nb)$

4) 在第五步，若主体 B 接收到了消息 $\{M_5\}_{k_{ab}}$，则有 $K_{B,t5}(R(B,t5,\{M_5\}_{k_{ab}}))$；根据 2)中的假设[exp2]：$B_{B,t3}(\otimes_{t3} k_{ab})$ 成立；根据前提假设 $a2$ 和新鲜性公理有：

$K_{B,t3}(fresh(Nb)) \rightarrow K_{B,t5}(fresh(Nb))$ [exp4]

假设 B 相信在[$t1,t5$]时间区间 S 没有泄露 k_{ab}，在[$t3,t5$]时间区间 A 没有泄露 k_{ab},且在[$t4,t5$]时间区间 A 没有泄露 Nb，则有 $B_{B,t5}(\otimes_{t5} k_{ab})$ 和 $B_{B,t5}(Nb \notin pub_{t5}(M_4))$。那么根据保密性公理，下面的结论成立：

$B_{B,t5}(\otimes_{t5} Nb)$ [exp5]，

由 $K_{B,t5}(R(B,t5,\{Nb-1\}_{k_{ab}}))$ 和公理 *A6-d* 可知：在 $t5$ 时，B 相信 A 在 $t5$ 前的某一时刻发送了消息 Nb，结合前提假设 *a2* 有：$B_{B,t5}(L_{A,t4} Nb)$ [exp6]

最后假设 B 相信在[$t1,t5$]时间区间 S 没有泄露 k_{ab}，在[$t3,t5$]时间区间 A 没有泄露 k_{ab}，由[exp4]、[exp6]、$K_{B,t5}(S(B,t3,\{Nb\}_{k_{ab}}))$，根据拥有对称密钥的判定定理，下面的结论成立：

$B_{B,t5}(L_{A,t4} k_{ab})$ [exp7]

通过利用扩展 *CS* 逻辑对该对称密钥交换协议实例的分析，得出该协议只能达到部分认证性，即不能让 A 确认 B；而在 B 确认 A 时，还需要这样一些假设：1）$\{M_3\}_{k_{ab}}$ 应该是新鲜的，[exp2]才成立。2) S 在[$t1,t5$]时间区间没有泄露 k_{ab}，且 A 在[$t3,t5$]时间区间没有泄露 k_{ab}，这样才能保证 $K_{B,t5}(\otimes_{t5} k_{ab})$ 和结论[exp7]成立。通过这两个假设，根据公理 *A1-e* 我们能得到协议目标 *ii*。

对于中间结论[exp5]：$B_{B,t5}(\otimes_{t5} Nb)$，我们看出它和协议目标没有关系，因此我们为[exp5]所做的假设“在[$t4,t5$]时间区间内 Nb 是保密的”是多余的，其实协议的保密性是通过临时会话密钥 k_{ab} 的保密性与 Nb 的完整性来实现的。能分析出这一点对于进一步改进该协议，采用正确的安全服务来保护消息有指导意义。此外，这个结论为下面的想法提供了理论依据：既然 Nb 不需要保密，我们是否可以用具有数据完整性服务的单项函数来代替对称密钥加密来实现认证协议？进一步的研究工作在后面的章节进行说明。

3.5、本章小结

在本章中，我们对 CS 逻辑进行了深入研究，发现了原有 CS 逻辑及其改进

方案中的一些缺陷，并对 CS 逻辑在符号和公理方面进行了扩展。该扩展的优势在于：1.更准确地描述了有关知识和信念的单调性概念。2.对发送和接收有关密文的知识和信念进行了描述，增加了在公钥体制和对称钥体制下保密性的推理公理，可以优化原有的推理过程。3.推理公理反映了知识与信念之间的转化关系，更客观地表达了人类的推理能力。4.从例子中有关 *Nb* 所需安全保护的分析能看出，该逻辑能分析出消息在某一个时间段内是否采用了正确安全服务，这对分析者了解协议消息的性质以及设计者设计更优质安全协议提供了强有力的帮助。5.刻画了敌手的知识，这样能够更好地分析协议的安全性。

第四章 MBL 逻辑

近年来，虽然出现了多种符号理论下的安全协议形式化分析法，但是 BAN 类逻辑仍然是研究安全协议的重要方法之一。BAN 类逻辑是一类基于信念的模态逻辑，它采用逻辑推理的方法，针对协议的前提假设与协议中的各个消息元素进行推理，而主体的信念随着推理的进行不断地发展变换，最后通过主体的最终信念我们可判断协议能否完成预期的目标。由于 BAN 类逻辑的推理公理反映了有关密码算法的基本逻辑关系，简单易懂，便于协议分析者掌握，所以受到了广泛的关注。但是 BAN 类逻辑也存在若干缺陷，比如理想化方法不规范，采用的推理规则对可信中心的依赖太强等等。我们在对 BAN 类逻辑中的各种逻辑进行深入研究后，总结了这类逻辑存在的若干不足，在对它们进行更正的同时，提出了一种新的推理逻辑—MBL 逻辑。

4.1、BAN 类逻辑的缺陷

缺陷 1 理想化方法不规范

BAN 逻辑在分析安全协议时，首先要对协议进行理想化处理，通过理想化，不仅能将协议的一般说明转化为可被逻辑系统所理解的形式，而且还能表达协议消息的内在含义。但是 BAN 逻辑的这种理想化方法缺少规范化手段，需要分析者对协议充分理解并依靠经验来完成，这样就会产生安全隐患。例如下面的协议：

1) $A \rightarrow B: A, Na$
2) $B \rightarrow S: A, Na, B, Nb$
3) $S \rightarrow B: \{K_{AB}, Nb\}_{K_{BS}}, \{K_{AB}, Na\}_{K_{AS}}$
4) $B \rightarrow A: \{K_{AB}, Na\}_{K_{AS}}, \{Na\}_{K_{AB}}, Nb'$
5) $A \rightarrow B: \{Nb'\}_{K_{AB}}$

在对协议第三步理想化时，按照文献[1]对 Needham-Schroeder 对称钥协议的理想化过程，由于 S 在步骤 2 知道要为 A 和 B 分配临时会话密钥，所以理想化过程应写为 $S \rightarrow B: \{A \xleftrightarrow{K_{AB}} B, \#(A \xleftrightarrow{K_{AB}} B), Nb\}_{K_{BS}}, \{A \xleftrightarrow{K_{AB}} B, Na\}_{K_{AS}}$，当 B 收到消息 $\{K_{AB}, Nb\}_{K_{BS}}$ 时，通过理想化步骤中的 $\{A \xleftrightarrow{K_{AB}} B, \#(A \xleftrightarrow{K_{AB}} B), Nb\}_{K_{BS}}$ 就可认为 K_{AB} 是 A、B 间的良好临时会话密钥。采用 BAN 逻辑对理想化后的协议进

行分析，就会得出该协议是安全的结论。

但该理想化过程是依靠经验完成的，所以并不正确，因为 B 通过所接收的消息并不能判断出在与谁共享 K_{AB}。事实上，上面的协议不能抵抗敌手通过欺骗可信中心 S 发起的冒充攻击，攻击步骤如下：

1) $E(A) \rightarrow B: A, Na$
2) $B \rightarrow S: E(A), Na, B, Nb$
3) $S \rightarrow B: \{K_{AB}, Nb\}_{K_{BS}}, \{K_{AB}, Na\}_{K_{AS}}$
4) $B \rightarrow E(A): \{K_{AB}, Na\}_{K_{AS}}, \{Na\}_{K_{AB}}, Nb'$
5) $E(A) \rightarrow B: \{Nb'\}_{K_{AB}}$

第 1 步，攻击者 E 冒充 A 给 B 发出通信请求；第 2 步 E 将 B 发出消息中的主体标示符"A"改为"E"来欺骗 S。第 4 步，E 截获 B 发给 A 的消息 $\{K_{AB}, Na\}_{K_{AS}}$，解出其中的会话密钥 K_{AB}；第 5 步，E 将 $\{Nb'\}_{K_{AB}}$ 发给 B，使得 B 误认为 A 拥有 K_{AB}。这样 E 就可以冒充 A 来和 B 进行通信了。

缺陷 2：安全服务的误用

SVO 逻辑是 BAN 类逻辑的佼佼者，它吸取了 BAN 逻辑、GNY 逻辑、AT 逻辑的优点，公认是一种比较成熟的模态逻辑。但是它在分析协议的认证性时采用的却是错误的安全服务：利用保密服务来进行有关认证的推理。比如在 SVO 逻辑中的源相关推理规则中，使用保密服务来确认消息源，即：K_{AB} 为主体 A、B 间的良好会话密钥，当 A 看到 $\{M\}_{K_{AB}}$ 时，则认为 B 曾说过 M。但事实上这是一种安全服务的误用，由于密文也会被敌手篡改，保密服务往往不能提供消息源的确认。例如，Wenbo Mao 曾描述了一种对 Needham-Schroeder 对称钥认证协议的攻击[9]。在该攻击中，敌手对采用 AES-CBC 算法的密文进行有意义的修改，使得接收方仍认为修改后的消息是来自发送方的完好消息。该攻击说明，若密文没有受到消息完整性保护，解密者就不能确信解密数据的真实性。由此看出，保密服务无法提供有效的消息源认证。

缺陷 3 不讨论保密性

BAN 类逻辑不讨论保密性，本质上是一类分析认证性的逻辑。由于逻辑中没有关于保密性的推理规则或公理，主体对于所分配的临时会话密钥的认可是通过认证来自可信中心的消息而实现的，也就是说，临时会话密钥的良好性（包括

保密性）完全依赖于对可信中心的信赖。而事实上，这种信赖已超出了合理范围，因为虽然可信中心不说谎，但敌手通过欺骗能诱导可信中心发出错误的消息，若此时主体相信可信中心，那么可能就会受到敌手的攻击。例如缺陷 1 中的例子：根据仲裁公理，B 通过信任可信中心 S 来相信 S 所分配的临时会话密钥 K_{AB} 是与 A 秘密共享的，但敌手 E 却通过欺骗 S，使 S 发出了错误消息，最后造成 B 与 E 共享 K_{AB} 的结果，此时临时会话密钥就失去了对敌手的保密性。

缺陷 4 缺少证明系统合理性的严格机制

BAN 类逻辑中的 BAN 逻辑与 GNY 逻辑都没有提供独立且明确的语义基础，造成逻辑系统缺乏合理性的依据，因此受到了很多质疑。鉴于此，AT 逻辑与 SVO 逻辑均给出了语义模型，为逻辑系统的合理性打下了基础。但在 AT 逻辑与 SVO 逻辑中，推理逻辑都以公理的形式给出，而公理无需证明，因此对于逻辑公理与语义模型间的正确联系仍无法判断。

4.2、MBL 逻辑

我们通过研究发现具有认证性的对称钥交换协议在非 PKI 的安全域中有着广泛的应用，因此我们希望在 BAN 类逻辑的基础上为这种协议建立一种系统的、切合实际的形式化分析工具，来弥补 BAN 类逻辑的不足。

具有认证性的对称钥交换协议通常分为两大类：一类由可信中心来产生会话密钥，一类由一方主体来产生会话密钥。因为由可信中心产生会话密钥的协议中，协议的发起者与协议的响应者为协议的对等参与方，双方处于对等的地位，各自不能通过协议谋求任何特权，这更具有实际意义，所以我们针对这类认证协议建立一种基于模态逻辑的形式化分析工具：MBL 逻辑。

4.2.1、MBL 逻辑对安全协议设计的要求

如果安全协议认证性的实现采用了错误的安全保护，那么既使这种安全保护是优质的，该协议也可能不安全。例如 Wenbo Mao 在文献[47]中指出：使用保密服务来设计认证协议就是安全保护的误用，通过篡改密文就会导致该认证协议的不安全；认证协议所需的安全保护其实是消息源认证服务和消息完整性服务，这两种安全服务可通过带密钥的单向函数来实现。因此，我们建议采用带密钥的单

向函数来实现协议的认证性。

此外，最小程度地使用保密服务是我们所追求的，这能限制可能有利于密码分析的信息泄露的数量。而采用单向函数来完成认证能达到这一要求，因为在设计安全协议时，可仅对需要保密的消息使用保密服务，从而使得保密服务的使用最小化。

4.2.2、MBL 逻辑的符号

MBL 常用到的符号：A、B、P 表示协议中的主体，S 表示可信中心，E 表示敌手，M 表示消息集合，m 表示消息元素；$\odot M$ 表示 M 具有消息完整性；M_B 表示 B 曾发出过 M；$(\odot M)_B$ 表示 B 曾发出过 M，且 M 具有消息完整性；ϕ,φ 代表公式；$A\ bels(\phi)$ 表示 A 相信 ϕ；$\otimes_{A\leftrightarrow B}M$ 表示消息是在 A、B 间保密的；$good1(K,A,B)$ 为 A、B 间的保密密钥；$good2(K,A,B)$ 为 A、B 间新鲜的保密密钥；$good(K,A,B)$ 表示 $good1(K,A,B)$ 或者 $good2(K,A,B)$；${}_{A\leftrightarrow B}K$ 表示为 A 和 B 产生的密钥；$\{M\}_K$ 表示用密钥 K 对 M 的加密；$[M]_K$ 表示用带密钥 K 的单向函数对 M 进行的单向变换，其中 $[M]_K=(M,prf_K(M))$，$prf_K(\ )$ 在对称钥体制中表示一个带密钥的伪随机函数（例如可用 CBC-MAC 来实现），可用该密钥验证消息完整性和识别消息源；$A\ verifies[M]_K$ 表示 A 用 K 对 $[M]_K=(M,prf_K(M))$ 验证通过（我们认为若对 $[M]_K$ 的验证通过，则即使消息 M 中不含有关它的源的信息，验证者也能根据正在使用的验证密钥识别正确的源）；其他用到的符号在语义模型中给出。

4.2.3、MBL 逻辑的敌手模型

在符号理论下，广泛采用的敌手模型是 1983 年由 Dolev 和 Yao 提出的[50]，他们认为敌手可以控制整个通信网络，且具有如下能力：

(1) 可以窃听所有经过网络的消息；

(2) 可以阻止和截获所有经过网络的消息；

(3) 可以存储所获得或自身创造的消息；

(4) 可以根据存储的消息伪造消息，并发送该消息；

(5) 可以作为合法的主体参与协议的运行。

在对具体的安全协议进行攻击时，敌手除了能窃听、阻止、截获所有经过网络的消息外，还具备以下的知识和能力：

(1)熟悉加密、解密、散列等密码运算，拥有自己的加密密钥和解密密钥；

(2)熟悉参与协议的主体标示符及其公钥；

(3)具有密码分析的知识和能力；

(4)具有进行各种攻击，例如重放攻击的知识和能力。

Dolev-Yao 模型指出了一个重要的原则：永远不要低估敌手的知识和能力。应当根据具体的协议与应用环境，建立正确的敌手模型。

在 MBL 逻辑中，我们认为 Dolev-Yao 敌手模型中对敌手知识和能力的描述尚不足以表达现实中敌手对安全通信产生的危害。譬如：针对某些安全协议，Dolev-Yao 敌手模型中的敌手实施攻击的效果并不太明显，甚至可看作是一种无意义的攻击，但在实际中敌手却能延续其攻击效果，并产生有巨大破坏力的另一种攻击。

比如有一个协议：

1. $A \rightarrow B: A, Na$
2. $B \rightarrow S: \{A, B, Na, Nb\}_{K_{AS}}$
3. $S \rightarrow A: \{A, B, K_{AB}, Na\}_{K_{AS}}, \{A, B, K_{AB}, Nb\}_{K_{BS}}$
4. $A \rightarrow B: \{A, B, K_{AB}, Nb\}_{K_{BS}}$

当敌手 E 阻塞掉 A 时，可以冒充 A 和 B 完成一次协议的运行：

1. $E("A") \rightarrow B: A, Na$
2. $B \rightarrow S: \{A, B, Na, Nb\}_{K_{BS}}$
3. $S \rightarrow E("A"): \{A, B, K_{AB}, Na\}_{K_{AS}}, \{A, B, K_{AB}, Nb\}_{K_{BS}}$
4. $E("A") \rightarrow B: \{A, B, K_{AB}, Nb\}_{K_{BS}}$

虽然 E 能冒充 A，但由于 E 没有掌握密钥 K_{AS} 与 K_{BS}，因此不能得到临时会话密钥 K_{AB}，从直观上，人们会认为敌手 E 的这种攻击无意义。但是我们发现在实际中，敌手可利用这种看似无意义的攻击来发动有效的拒绝服务攻击：

在上面的攻击中可以看到，敌手通过欺骗已经让 B 得到一个临时会话密钥

K_{AB}，并且相信 A 处于活动状态。据此，敌手在下面的环境下就能够发起拒绝服务攻击。

假设在一个网络环境中，B 为一台服务器，A 为若干客户机中的一台。当客户机发出连接请求时，将通过协议与 B 协商会话密钥，然后再进行消息的安全传输。若 A 与 B 完成协议，B 会认为 A 接着要发送数据，则会分配一定的 CPU 时间，等待 A 准备并发送数据。由于上述冒充攻击的存在，敌手可以宕掉并冒充 A 向 B 发出连接请求时，协议也同样能成功完成，因此 B 会分配 CPU 时间并等待接收数据。倘若敌手冒充大量的合法主体向 B 并行发起连接请求时，B 会迅速耗尽 CPU 时间，从而无法完成合法用户提出的连接请求，造成拒绝服务攻击。

因此，在 MBL 逻辑的敌手模型中，敌手除了具有 Dolev-Yao 敌手模型中所列出的能力外，我们再加入一条：敌手有能力延续以前的攻击效果，进而造成其它的攻击。

4.2.4、MBL 逻辑的语义模型

在我们的语义模型中，对于每一个主体 P 都有一个本地状态 S_p，它是一个多维的向量族 $(B_p, Send_p, Receive_p, H_p)$，其中 B_p 是主体 P 的信念集合，$Send_p$ 是 P 曾经说过消息的集合，包括自己产生的消息和转发的消息，$Receive_p$ 是 P 曾经收到消息的集合，H_p 是 P 所拥有消息的集合。这里的消息包括原子消息和若干原子消息组成的集合。我们认为任何主体都可区分自己产生的消息和他人产生的消息。

全局状态含有包括所有主体局部状态的向量族 $(st_1, st_2, ..., st_n)$，还包括一个全局的公开消息集合 $public()$——所有公开的消息与消息集合的集合。若 ST 是全局状态，那么 $B_p(ST), Send_p(ST), Receive_p(ST), H_p(ST)$ 分别是主体 P 的局部状态中各相应的集合。*MBL* 逻辑中还隐含着一个秘密消息集合，集合中消息的保密性是通过公开集合 $public()$来描述的。所有的集合都是单调、递增和闭包的。

定义一轮协议 r 是一个由整数时间索引的全局变量的有限集合，协议中的 t 时记为 (r,t)。定义 V 为原始命题集合，定义 π 为一映射，将每一个常量命题 $v \in V$ 映射为点集 $\pi(v)$，即命题 v 为真的点。公式 Ψ 在点 (r,t) 为真记为：$(r,t) \models \Psi, \models \Psi$

意味着Ψ全真。

为了描述公式的语义，我们首先给出基本的逻辑关系和一些基本公理。

1) 基本逻辑关系

$(r,t)\models v \ \ iff \ \ if \ v\in V\ ,\ then\ \ (r,t)\in\pi(v)$

$(r,t)\models(\phi\varphi) \ \ iff \ (r,t)\models\phi\wedge\varphi$

$(r,t)\models\phi\wedge\varphi \ \ iff \ (r,t)\models\phi\wedge(r,t)\models\varphi$

$(r,t)\models(\phi\rightarrow\varphi) \ \ iff \ (r,t)\models\phi\rightarrow(r,t)\models\varphi$

$(r,t)\models(\phi\rightarrow\varphi)\wedge(r,t)\models\phi\Rightarrow(r,t)\models\varphi$

$(r,t)\not\models\varphi\wedge(r,t)\models(\phi\rightarrow\varphi)\ \Rightarrow(r,t)\not\models\phi$

2）基本公理

公理 1：主体的发送集合与接收集合都属于主体的拥有集合。

$\models Send_p\subset H_p$；$\models Receive_p\subset H_p$

公理 2：若主体拥有ϕ，则主体相信他拥有ϕ。

$\phi\in H_p\supset(\phi\in H_p)\in B_p$

公理 3：若M和M'均为消息，则$M\rightarrow M'$和$f(M)\rightarrow M'$意味着M'为M的元素或子集合。（f为M的一个映射）

$M\rightarrow M'\ or\ f(M)\rightarrow M' \quad iff\ M'\subset/\in M$

公理 4：信任关系

$\phi\in B_p\wedge\varphi\in B_p\supset\phi\wedge\varphi\in B_p$

公理 5：当主体发送或接收一个复合消息，则发送或接收其子消息

$(m_1,m_2)\in Send_r\supset m_1\in Send_r\wedge m_2\in Send_r$

$(m_1,m_2)\in Receive_r\supset m_1\in Receive_r\wedge m_2\in Receive_r$

公理 6：完整性

$\odot(m_1,m_2)\supset\odot m_1\wedge\odot m_2$

3）逻辑公式的语义表示如下

语义 1：看到

$(r,t)\models A\ \ sees(M)\ \ iff\ \ (r,t)\models M\in H_A\ \ or\ \ (r,t)\models k\in H_A\wedge\{M\}_k\in H_A$

语义2：诉说

$(r,t)\models A\ \ says(x)\ \ iff\ \ (\exists 0<t'\le t)\ \ (r,t')\models x\in Send_A$

$(r,t)\models A\ \ said(x)\ \ iff\ \ (\exists t'\le t)\ \ (r,t')\models x\in Send_A$

$(r,t)\models M_B\ \ iff\ \ (r,t)\models B\ \ said(M)$

语义3：新鲜性

$(r,t)\models fresh(x)\ \ iff\ \forall P,\ \forall t'<0,\ (r,t')\not\models P\ \ said(x)$

语义4：公开性

$(r,t)\models M\in public()\ \ iff\ \forall t,\forall P,(r,t)\models P\ \ sees(M)$

语义5：完整性 （消息在 t 时刻是完整的，当且仅当存在一个主体在 t 时可验证单向函数通过）

$(r,t)\models\odot M\ \ iff\ \exists P,(r,t)\models P\ \ verifies[M]_K$

语义6：被传递消息的保密性

（1）对于一个消息集合 M

$(r,t)\models\otimes_{A\leftrightarrow B}M\ \ iff\ \ (r,t)\models\exists K,good(K,A,B)\wedge\odot\{M\}_K$

（2）对于一个原子消息 m

$(r,t)\models\otimes_{A\leftrightarrow B}m\ \ iff\ \exists M,\ (r,t)\models\otimes_{A\leftrightarrow B}M\wedge m\in M\wedge m\notin public()$

语义7：属于

$(r,t)\models m\in M\ \ iff\ \exists P,\ (r,t)\models m\in H_P\wedge(M=m_1..m...)\in H_P$ or

$K\in H_P\wedge(M=\{m_1..m...\}_K)\in H_P$

语义8：保密会话密钥

$(r,t)\models good1(K,A,B)\ \ iff\ \ (r,t)\models\otimes_{A\leftrightarrow B}K$

$(r,t)\models good2(K,A,B)\ \ iff\ \ (r,t)\models fresh(K)\wedge\otimes_{A\leftrightarrow B}K$

语义9：可信中心管辖密钥

$(r,t)\models S\ \ controls(K)\ \ iff\ \exists P\ \ if$

$$(r,t)\models P\ \ bels(S\ \ says(M,A,B))\wedge P\ \ bels(K\in M)$$

$$then\ \ (r,t)\models P\ \ bels(fresh(K))\wedge P\ \ bels({}_{A\leftrightarrow B}K)$$

语义 10：验证单向函数

$(r,t)\models A\ \ verifies[M]_K\ \ iff\ \ (r,t)\models(M\in public()\wedge\odot M)\in B_A\ \ \wedge$

$if\ \ (r,t)\models(good(K,A,B))\in B_A\wedge[M]_K\in Receive_A\ \ then$

$(B\ \ said[M]_K\wedge K\in H_B)\in B_A$

语义 11：具有消息来源的完整性

$(r,t)\models(\odot M)_A\ \ iff\ \exists t'\le t,(r,t')\models M\in Send_A\wedge\odot M$

4.2.5、推理规则

规则 1：可识别消息来源的完整性规则：

$A\ \ bels(good(K,A,B))\wedge A\ \ verifies[M]_K\supset$

$A\ \ bels(\odot M)_B\wedge A\ \ bels(M\in public())\wedge A\ \ bels(K\in H_B)$

即，当 A 相信 K 为 A、B 间的良好会话密钥，且 A 验证单向函数成功后，A 相信 M 是来自 B 的具有完整性的公开消息，且相信 B 拥有 K。

推论：$A\ \ bels(\odot M)_B\supset A\ \ bels(\odot M)\wedge A\ \ bels(M_B)$

规则 2：消息提取规则

(1) $A\ \ bels(M)\wedge A\ \ bels(m\in M)\supset A\ \ bels(m)$

(2) $A\ \ bels(\odot M)_B\wedge A\ \ bels(m\in M)\supset A\ \ bels(\odot m)_B$

(3) $A\ \ says(M)\wedge m\in M\supset A\ \ says(m)$

规则 3：消息合取规则

$A\ \ bels(\odot m_i)_B\wedge A\ \ bels(\odot m_j)_B\supset A\ \ bels(\odot(m_i,m_j))_B$

规则 4：新鲜性规则

$A\ \ bels(fresh(m)\wedge A\ \ bels(m\in M))\supset A\ \ bels(fresh(M))$

规则 5：新鲜性验证规则

$$A\ bels(\odot M)_B \wedge A\ bels(fresh(M)) \supset A\ bels(B\ says(M))$$

规则 6：良好临时会话密钥判定规则(S 为可信中心)

$$A\ bels(good1(K,A,S)) \wedge A\ bels(S\ says(M,A,B)) \wedge A\ bels(K \in M)$$

$$\wedge A\ bels(S\ controls(K)) \supset A\ bels(good2(K,A,B))$$

即，当 A 相信 K 在 S、A 间秘密共享，且相信 S 刚刚发布了包含有 K 和会话双方标示符的消息，则 A 相信 K 的确是为 A 和 B 颁发的新鲜的临时会话密钥(此时 B 也许还没有得到 K)。

规则 7：保密性规则

$$A\ bels(good(K,A,B)) \wedge A\ bels \ominus \{M\}_K \wedge A\ bels(m \notin public()) \wedge A\ bels(m \in \{M\}_K) \supset$$

$$A\ bels(\otimes_{A \leftrightarrow B} m)$$

规则 8：拥有即相信规则

$$M \in H_A \supset A\ bels(M \in H_A)$$

4.2.6、推理规则的正确性证明

对 BAN 类逻辑一个有争议的焦点就是它们缺乏有效的正确性证明机制。在 MBL 逻辑中，所有推理规则在其语义模型下都是正确的，下面我们给出具体证明。

1. 可识别消息来源的完整性规则：

$$A\ bels(good(K,A,B)) \wedge A\ verifies[M]_K \supset$$

$$A\ bels(\odot M)_B \wedge A\ bels(M \in public()) \wedge A\ bels(K \in H_B)$$

证：

左边 $\equiv (r,t)\models(good(K,A,B))\in B_A \wedge (r,t)\models(M \in public() \wedge \odot M)\in B_A \wedge$

$if \quad (r,t)\models(good(K,A,B))\in B_A \wedge [M]_K \in Receive_A$

$then \quad (r,t)\models(B \quad said[M]_K \wedge K \in H_B)\in B_A$

语义10

$\Rightarrow (r,t)\models(M \in public() \wedge \odot M)\in B_A \wedge ([M]_K \in Send_B)\in B_A \wedge (K \in H_B)\in B_A$

公理3

$\Rightarrow (r,t)\models(M \in public() \wedge \odot M)\in B_A \wedge (M \in Send_B)\in B_A \wedge (K \in H_B)\in B_A$

公理4

$\Rightarrow A \ \ bels(\odot M)_B \wedge A \ \ bels(M \in public()) \wedge A \ \ bels(K \in H_B) \equiv$ 右边

（注：由于假设 A 能够识别自己的消息和外来消息，所以 A 可以判断 $[M]_K$ 是接收的消息）

推论：$A \ \ bels(\odot M)_B \supset A \ \ bels(\odot M) \wedge A \ \ bels(M_B)$

证：

左边 $\equiv (r,t)\models(\exists t' \le t,(r,t')\models(M \in Send_B \wedge \odot M))\in B_A$

基本逻辑关系

$\Rightarrow \quad (r,t)\models(\exists t' \le t,(r,t')\models(M \in Send_B))\in B_A \wedge$

$(r,t)\models(\exists t' \le t,(r,t')\models \odot M)\in B_A$

语义2

$\Rightarrow A \ bels(M_B) \wedge A \ bels(\odot M) \equiv$ 右边

2．消息提取规则

（1）$A \ \ bels(M) \wedge A \ \ bels(m \in M) \supset A \ \ bels(m)$

证：

左边 $\equiv (r,t)\models M \in B_A \wedge (M \to m)\in B_A$

基本逻辑关系，公理3

$\Rightarrow \quad (r,t)\models m \in B_A \equiv$ 右边

（2）$A \ \ bels(\odot M)_B \wedge A \ \ bels(m \in M) \supset A \ \ bels(\odot m)_B$

证：

左边 $\equiv (r,t)\models(\exists t' \le t,(r,t')\models(M \in Send_B \wedge \odot M))\in B_A \wedge (r,t)\models(m \in M)\in B_A$

基本逻辑关系

$\Rightarrow \quad (r,t)\models(\exists t' \le t,(r,t')\models(M \in Send_B \wedge \odot M \wedge m \in M)\in B_A$

公理5，6

$\Rightarrow \ (r,t)\models(\exists t' \le t,(r,t')\models(m \in Send_B \wedge \odot m))\in B_A \equiv$ 右边

(3) $A \ \ says(M) \wedge m \in M \supset A \ \ says(m)$

证：

左边 $\equiv (\exists 0 < t' < t) \ (r,t')\models M \in Send_A \wedge \ \ m \in M$

公理5

$\Rightarrow (\exists 0 < t' < t) \ (r,t')\models m \in Send_A$

语义2

$\Rightarrow A \ says(m) \equiv$ 右边

(4) $A\ said(M) \supset A\ said(m)$

证：同(3)

3．消息合取规则

$A\ bels(\odot m_i)_B \wedge A\ bels(\odot m_j)_B \supset A\ bels(\odot(m_i,m_j))_B$

证：

$$左边 \equiv (r,t) \models (\exists t' \le t,(r,t') \models (m_i \in Send_B \wedge \odot m_i)) \in B_A \wedge$$
$$(r,t) \models (\exists t' \le t,(r,t') \models (m_j \in Send_B \wedge \odot m_j)) \in B_A$$

基本逻辑关系

$$\Rightarrow \quad (r,t) \models (\exists t' \le t,(r,t') \models (m_i,m_j \in Send_B \wedge \odot m_i,m_j)) \in B_A$$

基本逻辑关系

$$\Rightarrow \quad A\ bels(\odot(m_i,m_j))_B \equiv 右边$$

4．新鲜性规则

$A\ bels(fresh(m) \wedge A\ bels(m \in M)) \supset A\ bels(fresh(M))$

证：

$$左边 \equiv (r,t) \models (\forall P,\forall t' < 0,(r,t') \not\models (P\ said(m))) \in B_A \wedge (r,t) \models (m \in M) \in B_A$$

基本逻辑关系

$$\Rightarrow \quad (r,t) \models (\forall P,\forall t' < 0,(r,t') \not\models (P\ said(M))) \in B_A$$

语义3

$$\Rightarrow A\ bels(fresh(M)) \equiv 右边$$

5．新鲜性验证规则

$A\ bels(\odot M)_B \wedge A\ bels(fresh(M)) \supset A\ bels(B\ says(M))$

证：左边 $\equiv (r,t) \models (\exists t' \le t,(r,t') \models (M \in Send_B \wedge \odot M)) \in B_A \wedge$

$$(r,t) \models (\forall P,\forall t' < 0,(r,t') \not\models (P\ said(M))) \in B_A$$

$$\supset (r,t) \models (\exists 0 < t' \le t,(r,t') \models (M \in Send_B \wedge \odot M)) \in B_A$$

$$\supset A\ bels(B\ says(M)) \equiv 右边$$

规则 6：良好临时会话密钥判定规则(S 为可信中心)

$$A\ bels(good1(K,A,S)) \wedge A\ bels(S\ says(M,A,B)) \wedge A\ bels(K \in M)$$

$$\wedge\ A\ bels(S\ controls(K)) \supset A\ bels(good2(K,A,B))$$

证：

左边 $\equiv A\ bels(good1(K,A,S)) \wedge A\ bels(S\ says(M,A,B)) \wedge A\ bels(K \in M) \wedge$
$iff\ \exists P\ if\ (r,t) \models P\ bels(S\ says(M,A,B)) \wedge P\ bels(K \in M)$
$then\ (r,t) \models P\ bels(fresh(K)) \wedge P\ bels(_{A \leftrightarrow B} K)$

语义9

$\Rightarrow A\ bels(good1(K,A,S)) \wedge A\ bels(fresh(K)) \wedge A\ bels(_{A \leftrightarrow B} K)$

基本逻辑关系, 语义8

$\Rightarrow A\ bels(\otimes_{A \leftrightarrow B} K) \wedge A\ bels(fresh(K))$

语义8

$\Rightarrow A\ bels(good2(K,A,B))$

7. 保密性规则

$A\ bels(good(K,A,B)) \wedge A\ bels \odot \{M\}_K \wedge A\ bels(m \notin public()) \wedge A\ bels(m \in \{M\}_K) \supset$
$A\ bels(\otimes_{A \leftrightarrow B} m)$

证：

左边 $\equiv (r,t) \models (good(K,A,B) \wedge \odot \{M\}_K) \in B_A \wedge (m \notin public()) \in B_A \wedge$
$(K \in H_A \wedge \{M\}_K \in H_A \wedge m \in M) \in B_A$
$then\ (r,t) \models P\ bels(fresh(K)) \wedge P\ bels(_{A \leftrightarrow B} K)$

语义6, 7

$\Rightarrow (r,t) \models (\otimes_{A \leftrightarrow B} M) \in B_A \wedge (m \notin public()) \in B_A \wedge (m \in M) \in B_A$

基本逻辑关系

$\Rightarrow (r,t) \models (\otimes_{A \leftrightarrow B} m) \in B_A \equiv$右边

8. 拥有即相信规则

$M \in H_A \supset A\ bels(M \in H_A)$

证：

左边 $\equiv (r,t) \models M \in H_A$

公理2

$\Rightarrow (r,t) \models (M \in H_A) \in B_A \equiv$右边

4.3、分析举例

分析一种改进的 Needham-Schroeder 对称钥认证交换协议，其中只对含有密码运算结果的消息进行分析，不需进行理想化处理。

1. $A \rightarrow B : Na, A$
2. $B \rightarrow S : Na, Nb, A, B$
3. $S \rightarrow B : [\{K\}_{K_{BS}}, Nb, A, B]_{K_{BS}}, [\{K\}_{K_{AS}}, Na, A, B]_{K_{AS}}$
4. $B \rightarrow A : [\{K\}_{K_{AS}}, Na, A, B]_{K_{AS}}$
5. $A \rightarrow B : [Na']_K$
6. $B \rightarrow A : [Na'-1]_K$

前提假设

$A\ bels(good1(K_{AS},A,S))$； $A\ bels(fresh(Na))$； $A\ bels(S\ controls(K))$；

$B\ bels(good1(K_{BS},B,S))$； $B\ bels(fresh(Nb))$； $B\ bels(S\ controls(K))$；

$A\ bels(\{A,B,Na,Na'\}\subset public())$； $A\ bels(\{A,B,Na,Na',K_{AS}\}\subset H_A)$；

$B\ bels(\{A,B,Nb\}\subset public())$；$B\ bels(\{A,B,Nb,K_{AS}\}\subset H_B)$

$S\ bels(good1(K_{AS},A,S))$， $S\ bels(good1(K_{BS},B,S))$；

目标：

$A\ bels(good2(K,A,B))$； $A\ bels(K\in H_B)$；

$B\ bels(good2(K,A,B))$； $B\ bels(K\in H_A)$

形式化分析：

协议第三步，当 B 收到消息 $[\{K\}_{K_{BS}},Nb,A,B]_{K_{BS}}$ 后，用 K_{BS} 对消息 $[\{K\}_{K_{BS}},Nb,A,B]_{K_{BS}}$ 验证通过后，运用可识别消息来源的完整性规则：

$$B\ bels(good(K_{BS},B,S))\wedge B\ verifies([\{K\}_{K_{BS}},Nb,A,B]_{K_{BS}})\supset$$

$$B\ bels(\odot(\{K\}_{K_{BS}},Nb,A,B))_S\wedge B\ bels((\{K\}_{K_{BS}},Nb,A,B)\in public())\quad [exp1]$$

由公理 1 和基本逻辑关系得：

$$([\{K\}_{K_{BS}},Nb,A,B]_{K_{BS}})\in Receive_B$$

$$\supset(\{K\}_{K_{BS}},Nb,A,B)\in H_B$$

$$\supset\{K\}_{K_{BS}}\in H_B \qquad [exp2]$$

对[*exp2*]运用拥有即相信规则可得：

$$B\ bels((\{K\}_{K_{BS}},Nb,A,B)\in H_B)，\ B\ bels(\{K\}_{K_{BS}}\in H_B) \qquad [exp3]$$

对[*exp3*]运用公理 4 和属于语义，可得：

$$B\ bels((\{K\}_{K_{BS}},Nb,A,B)\in H_B)\wedge B\ bels(\{K\}_{K_{BS}}\in H_B)$$

$$\supset\ B\ bels(\{K\}_{K_{BS}}\in(\{K\}_{K_{BS}},Nb,A,B)) \qquad [exp4]$$

对[*exp1*]和[*exp4*]运用消息提取规则可得：

$$B\ \ bels(\odot\{K\}_{K_{BS}})_S \supset B\ \ bels(\odot\{K\}_{K_{BS}}) \qquad [exp5]$$

结合前提假设$K_{BS}\in H_B$和[*exp2*]，由属于语义可得：

$$K_{BS}\in H_B \wedge \{K\}_{K_{BS}}\in H_B \supset K\in\{K\}_{K_{BS}} \qquad [exp6]$$

由[*exp6*]，根据拥有即相信规则和语义 7 可得：

$$B\ bels(K_{BS}\in H_B)\wedge B\ bels(\{K\}_{K_{BS}}\in H_B)\supset B\ bels(K_{BS}\in H_B\wedge\{K\}_{K_{BS}}\in H_B)$$

$$\supset B\ \ bels(K\in\{K\}_{K_{BS}}) \qquad [exp7]$$

此时，B并没有将K加入$public()$集合，所以有：

$$B\ \ bels(K\notin public()) \qquad [exp8]$$

由前提假设$B\ \ bels(good2(K_{BS},B,S))$、[*exp5*]、[*exp7*]、[*exp8*]，根据保密性规则可得：

$$B\ bels(good(K_{BS},B,S))\wedge B\ bels(\odot\{K\}_{K_{BS}})\wedge B\ bels(K\in\{K\}_{K_{BS}})\wedge B\ bels(K\notin public())$$

$$\supset B\ \ bels(\otimes_{B\leftrightarrow S}K) \qquad [exp9]$$

由前提假设$Nb\in H_B$，和[*exp2*]，根据拥有即相信规则和语义 7 可得：

$$B\ \ bels(Nb\in H_B)\wedge B\ \ bels((\{K\}_{K_{BS}},Nb,A,B)\in H_B)\supset$$

$$B\ \ bels(N_B\in(\{K\}_{K_{BS}},Nb,A,B)) \qquad [exp10]$$

由前提假设$B\ \ bels(fresh(Nb))$和[*exp10*]，根据新鲜性规则可得：

$$B\ \ bels(fresh(\{K\}_{K_{BS}},Nb,A,B)) \qquad [exp11]$$

由[*exp1*]和[*exp11*]，根据新鲜性验证规则可得：

$$B\ \ bels(\odot(\{K\}_{K_{BS}},Nb,A,B))_S\wedge B\ \ bels(fresh(\{K\}_{K_{BS}},Nb,A,B))\supset$$

$$B\ \ bels(S\ \ says(\{K\}_{K_{BS}},Nb,A,B)) \qquad [exp12]$$

由前提条件：$B\ \ bels(S\ \ controls(K))$和[*exp7*]、[*exp9*]、[*exp12*]，根据良好临时会话密钥判定规则可得：

$$B\ bels(good2(K,A,B)) \qquad [exp13]$$

在协议的第五步，当 B 结合[exp13]验证 $[N_A']_K$ 通过后，运用可识别消息来源的完整性规则：

$$B\ bels(good2(K,A,B)) \wedge B\ verifies([Na']_K) \supset B\ bels(K \in H_A) \qquad [exp14]$$

通过以上推理可得出结论：$B\ bels(good2(K,A,B))$、$B\ bels(K \in H_A)$．同理我们还可以得出结论：$A\ bels(good2(K,A,B))$、$A\ bels(K \in H_B)$。

由于协议结束时，主体的信念符合协议的预定目标，所以该协议是安全的。

4.4、MBL 逻辑的可扩展性

文献[47]已经指出，使用带密钥的单向函数来保护消息的完整性与消息源的可识别性才是设计具有认证性的安全协议的规范方法。由于 MBL 逻辑的一个主要特点就是可对被单向函数保护的消息进行逻辑推理，所以对其进行适当扩展，MBL 逻辑就能广泛适用于分析采用规范设计方法的安全协议。

例如，针对采用公钥体制的协议，我们可在 MBL 逻辑中添加有关公私密钥对的语义：

$$(r,t) \models good(K_A,K_A^{-1},A)\ \ iff\ \ \forall P,\exists Q,(r,t) \models K_A \in H_P \wedge if\ \ K_A^{-1} \in H_Q, then\ \ Q=A$$

即，若 K_A 与 K_A^{-1} 为 A 的良好公私密钥对，当且仅当 K_A 是公开的，K_A^{-1} 为 A 所私有。

$(r,t) \models good(K_A,A)\ \ iff\ \ (r,t) \models good(K_A,K_A^{-1},A)$ 即 K_A 是 A 的良好公钥，当且仅当 A 拥有良好的密钥对 K_A 与 K_A^{-1}。

有了密钥对的语义说明，我们就可对 $[M]_K$ 的含义在公钥体制下进行扩展，将 $[M]_K=(M,prf_K(M))$ 中的 $prf_K(M)$ 看作是一个数字签名算法，是由与 K 相匹配的签名私钥 K^{-1} 所作的签名结果，通过 K 可验证该签名的有效性，从而辨别数据的完整性并识别消息源。这样对可识别消息来源的完整性规则稍加以下修改，我们就能分析基于公钥的认证协议和基于 Diffle-Hellman 方案的认证密钥交换协议了：

对可识别消息来源的完整性规则的修改：

$$A\ bels(good(K_B,B)) \wedge A\ verifies[M]_{K_B} \supset$$

$$A\ bels(\odot M)_B \wedge A\ bels(M \in public()) \wedge A\ bels(K_B^{-1} \in H_B)$$

另外，我们还可以添加有关保密性的推理规则：

$$A\ bels(good(K_A,K_A^{-1},A))\wedge A\ bels\odot(M')_B\wedge A\ bels(fresh(M'))\wedge$$
$$A\ bels(\{M\}_{K_A}\in M')\wedge A\ bels(m\notin public())\wedge A\ bels(m\in\{M\}_K)\supset A\ bels(\otimes_{A\leftrightarrow B}m)$$

即当A认为自己具有良好的密钥对，且被K_A加密的M属于一条来自于B的新鲜的、具有消息完整性的消息M'，此时，倘若m属于$\{M\}_{K_A}$且A认为m没有公开过，则A认为m为A与B之间的保密消息。

添加这条保密性推理规则后，MBL 逻辑就能分析基于公钥的一般形式的密钥交换协议了。

4.5、本章小结

在本章中，我们首先总结了 BAN 类逻辑存在的一些不足之处，而后介绍了我们的研究成果——MBL 逻辑。MBL 逻辑是针对具有可信中心参与的认证密钥交换协议进行分析的形式化工具，具有下列一些特点：

1．对协议设计提出了建议

MBL 逻辑指出传递的协议消息应具有完整性保护，这可通过带密钥的单向函数来实现，另外为了限制可能有利于密码分析的信息泄露的数量，应最小程度的使用保密服务。这两点为设计安全的协议指出了方向。此外，MBL 的逻辑规则在分析协议时，可对消息性质进行考察，从而加深对协议设计的理解，比如在[*exp14*]的分析中，我们可以看出$[N_A']_K$中的N_A'并不需要是一个新鲜数就可以达到协议目标。

2．丰富了 Dolev-Yao 敌手模型

我们通过对密钥交换协议的深入研究，发现了一种新的攻击方法，而传统的 Dolev-Yao 敌手模型并不能充分地涵盖这种攻击，因此我们在 Dolev-Yao 敌手模型中添加了一条新的攻击能力：敌手有能力延续以前的攻击效果，进而造成其它的攻击。

3．具有严格的证明机制

对逻辑中的各公式进行了详细的语义说明，并将公理与推理规则区分开来。此外利用语义说明和公理对推理规则作了严格的正确性证明，克服了 BAN 类逻辑中缺乏正确性证明机制的缺陷。

4. 更严格地考察协议的安全性

在BAN类逻辑的推理中，均假设主体在临时会话密钥方面对可信中心绝对信任，因此不能发现敌手通过误导可信中心而造成的攻击协议。而在MBL逻辑中，由于含有保密性推理规则，因此可分析消息的保密性，并能确定在与谁共享保密消息，这样主体就不需要对可信中心绝对信任，可通过自己的推理来判断临时会话密钥，因此就能防止敌手通过欺骗可信中心而造成的攻击。

5. 具有可扩展性

MBL逻辑与其他BAN类逻辑一样具有良好的可扩展性。MBL逻辑的一个主要特点是可分析被单向函数保护的消息，因此对采用规范设计方法的安全协议有良好的适应性。通过在MBL逻辑中添加有关公钥的语义说明以及推理规则，MBL逻辑不仅能分析公钥认证密钥交换协议，而且还能分析基于Diffle-Hellman协议的密钥交换协议。

第五章 MBL 逻辑的自动化分析

5.1、有关自动化的已有工作

随着安全需求的不断提高，安全协议的结构日趋复杂，倘若形式化分析采用人工方式进行，则很容易因人为失误而产生错误。为了解决这个问题，也为了提高形式化分析的效率，人们开发出了许多安全协议自动化分析工具。

在 1997 年，出现了一种应用于安全认证协议和密钥分配协议的高级语言—通用认证协议说明语言 CAPSL(Common Authentication Protocol Specification language)[107]，它能充分表达认证协议的抽象特征来支持分析协议的缺陷。由于各种形式化工具的研究者能按照 CAPSL 的语法和语义将协议转换成自己工具所需的内部描述，因此 CAPSL 可作为不同形式化分析工具的输入。在 2000 年，Denker 与 Millen 介绍了关于 CAPSL 的集成环境[108]，在该环境中 CAPSL 被转换为一种中间语言 CIL，该语言有两种作用：一是定义 CAPSL 的语义，二是作为 CAPSL 协议的描述与各种分析工具所使用语言的接口。CIL 语言通过重写规则可表达协议的状态，然后通过与之匹配的连接器，将 CIL 作为不同安全协议形式化自动化分析工具的输入，如 Prolog 状态搜索分析工具，NRL 协议分析器[61,77]、FDR 等等。

Casper 安全协议分析编译器(a Complier for the Analysis of Security Protocols)与 CAPSL 类似，同样是作为协议分析工具的基础输入语言。Casper 编译器的主要作用是为协议生成 CSP 描述，进而可让分析者使用模型检测工具 FDR 对描述的协议系统进行攻击检测。FDR 是通过明确列举并以询问方式搜索系统的状态空间，只能处理有限状态系统，考虑到 FDR 对系统的这种特殊要求，Casper 编译器不仅可定义协议的操作，而且可定义要检测的系统。在系统的定义中，Casper 提供了一个关于定义哪些主体参与系统，它们所扮演的角色，以及它们所使用的数据项的机制，这是 CAPSL 所不具备的。Casper 通过改进也可生成为其它协议分析工具所用的输入。

上面所介绍的两种自动化方法都是针对基于攻击结构性的形式化分析法而设计的，可提高该分析法的准确度和效率。对于基于推理逻辑的分析方法而言，

由于在协议分析中要对初始假设、协议消息以及中间推理结果不断的应用推理规则和公理，以判断协议是否能实现最终目标，若采用人工分析，效率不高而且极易出错，因此也需要实现分析的自动化。基于推理逻辑的自动化分析方法已有很长的研究历史，在 1992 年，Campbell 与 Safavi-Naini 提出了基于 BAN 逻辑的自动化分析法，开创了推理逻辑方法自动化的先河[109]。而后，相继出现了一些关于 BAN 逻辑的其他自动化分析方法[110,111]。由于 GNY 逻辑推理规则的复杂性，Mathuria 等人改进了 GNY 逻辑，这种改进既保留了原有的逻辑，又可保证自动化分析不会陷入死循环之中，基于改进的 GNY 逻辑，他们开发了一种自动化分析工具[112]。Brackin 在 1996 年介绍了一种认证协议自动分析器—AAPA(Automatic Authentication Protocol Analyzer)[113,114]。该分析器采用 GNY 逻辑的扩展逻辑来生成其证明结果。在对协议进行自动分析时，采用简单接口描述语言 ISL（Interface Specification Language）形式化地描述协议及其关键性质，然后通过归纳协议步骤自动化地证明协议是否具有它所说明的所有性质，或者找出协议的错误。AAPA 能捕捉协议常见的大部分错误，比如识别由不合理的初始假设条件造成的错误，识别协议运行时是否具有必要的信任关系等等。在 1999 年，Brackin 又提出了 AAPA 的第二版本—AAPA2[115]，与 AAPA 相比，AAPA2 能够发现新鲜性、类型、捆绑、并行会话和其他的攻击所造成的安全隐患。

5.2、基于 MBL 逻辑的自动化分析法

MBL 逻辑是针对模态逻辑分析法的一种形式化分析工具。这种分析工具的特点就是由初始假设和消息出发，通过逻辑中给出的推理规则和公理进行推理，来判断给定协议是否能达到预期目标。而我们要做的工作就是实现这种推理过程的自动化。

作为目前两种最重要的人工智能程序设计语言之一，Prolog 语言[116]是一种以逻辑推理为基础的逻辑型设计语言，这是一种描述性的语言，强调的是描述对象之间的逻辑关系，在求解问题时，只需程序员描述带解问题中的对象及它们之间的一些已知事实和规则，就能得出答案。因此该语言很适合实现基于信念逻辑的协议自动化分析。目前，Prolog 有多个版本：Amzi Prolog、B Prolog、Turbo Prolog、SWI Prolog 等等，我们所采用的是 SWI Prolog 作为 MBL 逻辑自动推理

实现的工具。

5.2.1、MBL 逻辑符号到 Prolog 语言的转化

1. 符号的表示

在 MBL 逻辑中，很多的常量都是用大写字母表示，而 Prolog 语言规定大写字母开头的字符是变量。为了将两种表达方式统一起来，我们将 MBL 逻辑中的各常量，如实体名称、随机数、密钥等在 Prolog 语言中表示为用单引号"括住的字符（在 Prolog 语言中，用单引号括住得任何字符表示为常量）。比如主体标示符 A 在 Prolog 语言中标示为'A'。

此外，我们定义了各符号在 Prolog 语言中的表达形式：

MBL 逻辑符号	Prolog 表达
$\{M\}_K$	*encrypt(M,K)*
$[M]_K$	*onewayfunction(M,K)*
$M \in H_A$	*holds(A,M)*
$\odot M$	*integrate(M)*
$N_A \in public()$	$pub(N_A)$
$K \notin public()$	*notpub(K)*
K_{AS}	*KAS*

2. 对逻辑规则的改动

在我们进行人工形式化分析时，是人为取出对达到协议目标有用的中间结论进行下一步的分析，因此推理过程显得简单。但在进行自动化分析时，程序不仅会产生有用的中间结论，而且会产生大量无用的中间结论，比如 4.3 节的分析例子中，在对[*exp1*]使用消息提取规则时，除了产生结论 $B\ bels(\odot\{K\}_{K_{BS}})_S$，还会产生结论 $B\ bels(\odot(N_B,A,B))_S$，结合前提假设 $B\ bels(fresh(N_B))_S$，就会产生结论 $B\ bels(fresh(N_B,A,B))_S$，而该结论对于达到协议目标是无意义的。为了避免这种无用中间结论的大量产生，我们去掉新鲜性规则，并将新鲜性验证规则改为：

$$A\ bels(\odot M)_B \wedge A\ bels(fresh(m) \wedge A\ bels(m \in M)) \supset A\ bels(B\ says(M))$$

为了方便自动化处理，我们规定协议中消息的排列顺序为：$[\{K\}_{K_{BS}},A,B,N_B]_{K_{BS}}$，即消息排列依次为：加密数据项、主体标示符、新鲜数。

5.2.2、协议信息在 Prolog 语言的存储

通过对 MBL 逻辑的研究，我们发现在对协议进行分析时，所出现的信息不外乎这几项：初始假设、协议消息、中间结论和推理规则。在 Prolog 语言中，我们采用谓词 *fact*/3 对这些信息进行统一存储。*fact*/3 的具体表示为：

fact(*Index*, *Statement*,*reason*(*PremiseList*,*Rule*))

其中 *Index* 表示 *fact*/3 实例的索引号；*Statement* 表示协议的前提假设、协议步骤中的消息和推理的中间结论；*Rule* 用于指出 *Statement* 是一个前提假设、协议步骤、还是推理出的中间结论。下面是 *fact*/3 的一些具体例子：

fact(1,*bels*('*A*',*good*1('*KAS*',['*A*','*S*'])),*reason*([],'*assumptions*'))，表示存储的是一个前提假设。

fact(2,*bels*('*A*',*integrate*(['*NA*',*encrypt*(['*K*'],'*KAS*'),'*A*','*B*']))

,*reason*([5,20],'*R*1')),表示根据命题 5 和 20，由规则 *R*1 可推导出结论：*A* *bels*(N_A,⊙{*K*}$_{K_{AS}}$,*A*,*B*)。

对于协议目标，我们采用谓词 *goal*/2 来表达：*goal*(*Index*,*Statement*)。

除了上述要存储的信息之外，我们还需得到当前阶段 *fact*/3 的最大索引号，以便为推理过程中新推导出的中间结论分配新的索引号，并将该中间结论存储起来。 我们通过文献[112]，定义了两个谓词 *getMaxFactIndex*/1 和 *getMaxInList*/2 来完成此项工作：

```
getMaxInList([Item],Max) :- Max is Item.

getMaxInList([Head|Tail],Max) :-
getMaxInList(Tail,TempMax),
Max is max(Head,TempMax).

getMaxFactIndex(MaxIndex) :-
bagof(Index ,X^Y^fact(Index,X,Y),IndexList),
getMaxInList(IndexList,MaxIndex).
```

5.2.3、推理的实现

所有 MBL 逻辑的推理规则都由 *rules*/0 谓词来表示，每个推理规则至少有一个 *rules*/0 实例，对于能推理出多个结论的推理规则，则需要由多个 *rules*/0 实例

来表达。下面的实例是“拥有即相信规则”的表达：

```
rules :-
fact(PremiseIndex5,holds(A,M),_),
Conclusion=bels(A,holds(A,M)),
not(fact(_,Conclusion,_)),
getMaxFactIndex(MaxIndex),NewIndex is MaxIndex+1,
asserta(fact(NewIndex,Conclusion,reason([PremiseIndex5],'R4'))),
asserta(addedFacts).
```

Prolog 在执行此规则时，首先检查在 Prolog 数据库中命题 *holds(A,M)* 是否为真，若为真，则产生结论 *bels(A,holds(A,M))*，接着检查在数据库中是否已有该结论，若没有，则就获取当前 *fact/3* 的最大索引号，然后将该索引号分配给该结论，并将该结论写入数据库中。

值得一提的是，在 4.3 节对协议进行人工分析时，对于[*exp*2]，我们是通过“验证单向函数的语义”直接得到结论 $M \in Receive_A$，结论[*exp*4]也是通过“属于语义”得到的，因此在 Prolog 中，我们要将这两个语义转化为推理规则：

属于规则：

```
rules :-
fact(PremiseIndex6,holds(A,M),_),
fact(PremiseIndex7,holds(A,X),_),
member(X,M),
Conclusion=bels(A,belongs(X,M)),
not(fact(_,Conclusion,_)),
getMaxFactIndex(MaxIndex),NewIndex is MaxIndex+1,
PremiseIndices1=[PremiseIndex6,PremiseIndex7],
asserta(fact(NewIndex,Conclusion,reason(PremiseIndices1,'R5'))),
asserta(addedFacts).
```

单向函数规则：

```
rules :-
fact(PremiseIndex3,holds(A,onefunction(M,K)),_),
Conclusion=holds(A,M),
not(fact(_,Conclusion,_)),
getMaxFactIndex(MaxIndex),NewIndex is MaxIndex+1,
asserta(fact(NewIndex,Conclusion,reason([PremiseIndex3],'R2'))),
asserta(addedFacts).
```

在推理结论[*exp*8]时，对于*K*是否属于*public*()集合，我们只能通过叙述来表

达，无法进行严格的形式化描述，但这并不意味着我们是通过非形式化的方法得出的结论，因为在自动化分析中，我们可以给出该问题的解决办法：

```
rules :-
fact(PremiseIndex4,bels(A,pub([X|_])),_),
Conclusion=bels(A,pub(X)),
not(fact(_,Conclusion,_)),
getMaxFactIndex(MaxIndex),NewIndex is MaxIndex+1,
asserta(fact(NewIndex,Conclusion,reason([PremiseIndex4],'R3'))),
asserta(addedFacts).

rules :-
fact(PremiseIndex4,bels(A,pub([_|Rest])),_),
length(Rest,LengthofRest),
(
  LengthofRest>1,
  Conclusion=bels(A,pub(Rest));

  LengthofRest=:=1,
  getHead(Rest,Head),
  Conclusion=bels(A,pub(Head))
),
not(fact(_,Conclusion,_)),
getMaxFactIndex(MaxIndex),NewIndex is MaxIndex+1,
asserta(fact(NewIndex,Conclusion,reason([PremiseIndex4],'R3'))),
asserta(addedFacts).
```

其中谓词*getHead*/2的定义为:*getHead*([*X*|_],*Head*) :- *Head* = *X*.

然后将保密性规则描述为：

```
rules :-
fact(PremiseIndex10,bels(A,good1(K,[A,B])),_),
fact(PremiseIndex11,bels(A,integrate(M)),_),
fact(PremiseIndex12,bels(A,belongs(X,encrypt(M,K))),_),
not(fact(_,bels(A,pub(X)),_)),
Conclusion=bels(A,good1(X,[A,B])),
not(fact(_,Conclusion,_)),
getMaxFactIndex(MaxIndex),NewIndex is MaxIndex+1,
PremiseIndices3=[PremiseIndex10,PremiseIndex11,PremiseIndex12],
asserta(fact(NewIndex,Conclusion,reason(PremiseIndices3,'R7'))),
asserta(addedFacts).
```

为了在推理过程中能将所有的规则循环套用一次，我们定义了谓词*oneCycle*：

```
oneCycle :-
rules,
fail.
```

其中 *fail*/0 谓词在 Prolog 语言中可导致回溯的发生，从而能使推理遍历所有的推理规则。

为了实现前向推导，我们使用了谓词 *forward*/1：

```
forward(Cycle) :- Cycle>0,done.
forward(Cycle) :- not(oneCycle),NextCycle is Cycle+1,forward(NextCycle).
```

为了检查当前推理中有无新结论添加到数据库中，我们定义了谓词*done*/0：

```
done :-not(retract(addedFacts));(retractall(addedFacts),fail)。
```

最后我们定义了谓词 *analyze*/0，用来触发协议的自动分析：

```
analyze :- fact(_,_,_),asserta(addedFacts),forward(1)。
```

5.2.4、实例分析

我们将 MBL 逻辑的推理规则用 Prolog 语言表示，并存储为 mblogic.pl 语言，在分析协议时，我们将协议的初始信息、假设和目标分别用 *fact*/3 和 *goals*/2 表示，存储为一个协议描述文件：

```
%==========Needham-Schroeder Description.
%==========Messages.
fact(1,holds('B',onefunction(['NB',encrypt(['K'],'KBS','A','B'],'KBS')),reason([],'setp')
).
fact(2,holds('A',onefunction(['NA',encrypt(['K'],'KAS','A','B'],'KBS')),reason([],'setp')
).
fact(3,holds('B',onefunction(['NA1'],'K')),reason([],'setp')).
fact(4,holds('A',onefunction(['NA1-1'],'K')),reason([],'setp')).

%==========Assumptions.
fact(5,bels('A',good1('KAS',['A','S'])),reason([],'assumptions')).
fact(6,bels('A',fresh('NA')),reason([],'assumptions')).
fact(7,bels('A',controls('S',['K'])),reason([],'assumptions')).
fact(8,bels('B',good1('KBS',['B','S'])),reason([],'assumptions')).
fact(9,bels('B',fresh('NB')),reason([],'assumptions')).
fact(10,bels('B',controls('S',['K'])),reason([],'assumptions')).
fact(11,bels('A',pub('A','B','NA','NA-1')),reason([],'assumptions')).
fact(12,holds('A',['A','B','NA','NA1','KAS'],reason([],'assumptions')).
fact(13,bels('B',pub('A','B','NB')),reason([],'assumptions')).
fact(14,holds('B',['A','B','NB','KBS'],reason([],'assumptions')).
fact(15,bels('S',good1('KAS',['A','S'])),reason([],'assumptions')).
fact(16,bels('S',good1('KBS',['B','S'])),reason([],'assumptions')).

%==========More Assumptions.
fact(17,verifies('B',onefunction(['NB',encrypt(['K'],'KBS','A','B'],'KBS')),reason([],'as
```

sumptions')).
fact(18,verifies('B',onefunction(['NA1'],['K'])),reason([],'assumptions')).
fact(19,verifies('A',onefunction(['NA',encrypt(['K'],'KAS'),'A','B'],'KAS')),reason([],'as sumptions')).
fact(20,verifies('A',onefunction(['NA1-1'],['K'])),reason([],'assumptions')).

%==========Goals.
goals(1,bels('A',good2('K',['A','B'])).
goals(2,bels('A',holds('B',['K'])).
goals(3,bels('B',good2('K',['A','B'])).
goals(4,bels('B',holds('A',['K'])).

在 *More Assumptions* 部分，我们假设单向函数可以被验证通过，这是因为在符号理论中，默认密码体制是理想化的，不做该假设则无法进行协议的分析。

由于篇幅关系，我们仅列出目标 1 的推理过程：

[1] *proof for bels('A',good2('K',['A','B'])):*

22. *bels('A', integrate(['NA', encrypt(['K], 'KAS'), 'A', 'B']))* {5,20,R1}
26. *bels('A',from(['NA', encrypt(['K'], 'KAS'), 'A', 'B'], 'S'))* {5,20,R1}
31. *holds('A', ['NA', encrypt(['K'], 'KAS'), 'A', 'B'])* {2,R2}
40. *holds('A', [encrypt(['K'], 'KAS'), 'A', 'B'])* {31,R3}
42. *holds('A', ['B', 'NA', 'NA1', 'KAS'])* {12,R3}
47. *bels('A', integrate([encrypt(['K'], 'KAS'), 'A', 'B']))* {22,R3}
86. *bels('A', says('S', ['NA', encrypt(['K'], 'KAS'), 'A', 'B']))* {6,22,26,R9}
91. *holds('A', encrypt(['K'], 'KAS'))* {40,R3}
93. *holds('A', ['NA', 'NA1', 'KAS'])* {42,R3}
99. *bels('A', says('S', [encrypt(['K'], 'KAS'), 'A', 'B']))* {86,R3}
100. *bels('A', integrate(encrypt(['K'], 'KAS')))* {47,R3}
112. *bels('A', holds('A', encrypt(['K'], 'KAS')))* {91,R4}
134. *holds('A', ['NA1', 'KAS'])* {93,R3}
162. *holds('A', 'KAS')* {134,R3}
169. *bels('A', holds('A', 'KAS'))* {162,R4}
179. *bels('A', belongs(['K'], encrypt(['K'], 'KAS')))* {112,169,r6}
180. *bels('A', good1(['K'], ['A', 'S']))* {5,100,179,R7}
181. *bels('A', good2(['K'], ['A', 'B']))* {180,99,179,7,R10}

由上面的推理步骤可以看出协议能达到目标 1。

推理步骤看似比我们人工推理的中间结果要多的原因是，在人工推理的过程中，我们省去了一些很直观的中间结论，而在自动化的推理过程中，这些结论却要一步步地推理得出，比如：162. *holds*('*A*', '*KAS*')在人工推理中直接可在前提假设 12 中得出，但在自动推理中，却要经过 42、93、134 三个中间结果才能得出。

5.3、本章小结

本章对安全协议的自动化分析进行了详细地研究，主要工作有：1. 详细介绍了目前安全协议自动化分析领域的各种已有成果。2. 基于 MBL 逻辑设计了一个安全协议自动化分析工具，该工具中实现了 MBL 推理规则的 Prolog 表达，其中为优化中间结果对某些规则进行了改进。3. 举例说明了如何使用该工具对协议进行自动化分析。

第六章 设计安全的认证密钥交换协议

在第四章我们已经提到，实现协议的认证性所需要的安全服务是消息源认证服务与消息完整性服务，此外，为了实现临时会话密钥的交换，还需最小程度地使用保密服务。这些建议的提出实质上是考虑了敌手对协议所实施的另一种攻击：研究协议所采用的密码算法，以期找出其中存在的缺陷来发动攻击。为了有效抵制这种攻击，我们就需要对协议采用算法的安全性进行考察，分析敌手攻击算法的成功概率与计算代价，这就涉及到了计算理论的研究。

首先我们以计算理论为主，讨论如何设计安全的认证密钥交换协议。

6.1、加密体制的安全性

在计算理论下讨论协议的安全性，就不可避免地要讨论加密体制的安全性。

在符号理论下对安全协议进行分析时，一般不讨论加密体制的安全性，或者仅将所要采用的加密算法想象为教科书中出现的算法并指出在一定的假设下该算法是安全的。例如，符号理论下加密体制的安全性基于以下两方面假设：

1）“all-or-nothing”的安全性

已知一种加密算法和一条被该加密算法输出的密文，敌手的任务是恢复出由加密算法中的安全参数所决定的整条明文；或者已知一种加密算法，并给出明密文对，敌手需要恢复出对应的密钥。而“all-or-nothing”的含义是：敌手或完全得到想要的秘密，或什么也得不到。“nothing”意味着在攻击前后，敌手没有得到关于秘密的任何信息。

2）被动攻击

默认情况下，仅考虑加密体制在敌手被动攻击下的强度：敌手不能运用所掌握的数据操纵或修改密文，也不能要求拥有密钥的用户为敌手提供加解密服务。

其实，符号理论的这两点假设大大低估了敌手的实际攻击能力。首先，敌手对于加密信息往往并非一无所知，一个典型的例子就是：在各大银行中，用户账号密码的明文取值均为0~9之间，且明文长度为6位，这就给敌手穷举破解密码提供了条件。再者，敌手对于密码体制的攻击更可能的是主动攻击，同时还具有

被动窃听的能力：他们能用某种不确定的方式修改密文或者计算明文，然后将所得结果发给一个防备不强的用户以得到预言服务，并通过这种预言服务来获得想要的秘密。

不难看出，符号理论对敌手的能力过于低估，以至于造成其加密体制安全性的概念不够强。为了提出更强的安全性概念，我们就需要考虑在面对一个具有主动攻击的足够聪明的敌手时加密体制安全性，以便可采用具有更强安全性的加密体制来保护秘密消息。为了建立更严格的安全性概念，人们提出一些攻击游戏来模型化并获得各种攻击情形。这些游戏是在敌手和随机预言机之间进行的。游戏的规则允许敌手获得由随机预言机提供的密码学帮助，这些帮助能看作是为敌手提供了一种“密码训练课程”。对于一种攻击游戏的形式化模型，若既使给敌手足够的“密码分析训练课程”，他也不能获得满意的成功，就认为这个密码体制是安全的。

下面我们通过模型化三种主动敌手的攻击行为来刻画三种不同安全程度的加密体制[47]。

6.1.1、不可区分的选择明文攻击

我们通过敌手与随机预言机间进行的游戏来形式化地描述这种攻击：

- 不可区分的选择明文攻击(IND-CPA)：

设：敌手 E 和一个随机预言机 O 商定了一个安全参数为 k 的目标密码体制 ε，它的明文空间是 M，密文空间为 C；O 固定了 ε 的一个加密密钥 K。

1）敌手 E 选择两条不同的消息 $m_0, m_1 \in M$，将它们发送给 O。

2）若这两条消息不同长，O 就把短的消息扩充，使其与另一个一样长。

O 投掷一个公平硬币 $b \in_U \{0,1\}$，然后执行下列加密操作：

$$c^* = \begin{cases} \varepsilon_K(m_0) & \text{若 } b=0 \\ \varepsilon_K(m_1) & \text{若 } b=1 \end{cases}$$

O 向敌手 E 发送 $c^* \in C$；

3）收到 c^* 后，敌手 E 必须回答 0 或 1，作为他对 O 投掷硬币结果的猜测。

令 Adv 表示敌手 E 区分 $\varepsilon_K(m_0)$ 和 $\varepsilon_K(m_1)$ 的优势，则有等式：

$$Prob[0 \leftarrow E(c^*) \mid c^* = \varepsilon_K(m_0)] = \frac{1}{2} \pm Adv \qquad [6.1]$$

- 抗不可区分的选择明文攻击的安全性：

一个安全参数为 k 的密码体制被称为对于 IND-CPA 是安全的，若任何多项式有界的敌手进行上述的攻击游戏后，式[6.1]中给出的优势 Adv 都是关于 k 的可忽略量。

6.1.2、不可区分的选择密文攻击

在 IND-CPA 中，我们假设敌手可访问加密机，但不能访问解密机，这种假设是不切合实际的。在密码系统的应用中，我们要求用户一直保持警惕而不提供解密预言服务是很难的，因为用户也许很天真，也许面对非常狡猾的敌手，这些都有可能使用户无意中给敌手提供了解密预言服务。

因此我们需要强化安全性的概念。考虑另一种攻击模型，称为不可区分选择密文攻击。在该模型中，我们降低敌手攻击密码体制的难度：除了在 CPA 游戏中可获得加密帮助外，我们进一步允许敌手获得解密模式下有条件的帮助。这个有条件的帮助是指敌手可在有限时间内访问解密机。

- 不可区分的选择密文攻击(IND-CCA)

前提假设：与不可区分的选择明文攻击一样，敌手 E 首先和 O 商定一个目标密码体制，且 O 选定了一个加密密钥。此外，由于敌手 E 访问解密机的时间有限，所以需要事先准备好一些密文。

1）敌手 E 向 O 发送一条准备好的密文消息 $c \in C$。

2）O 解密 c，返回解密结果给敌手 E。

3）敌手 E 选择另一条准备好的密文消息，并重复过程 1 和 2，以便反复进行解密训练课程。

4）一旦敌手 E 对于解密训练课程感到满意，他就要求 O 进行 IND-CPA 的游戏。在该游戏中 E 选择的明文消息 m_0 和 m_1 称为适应性选择明文，它们是 E 在进行整个解密训练课程后所产生的结果。

5）当选择明文游戏结束时，E 应该回答 0 或者 1，作为 IND-CPA 中对 O 投掷硬币的猜测。

若 E 进行整个密文分析训练课程的历史表示为 Hist-CCA，那么敌手的优势

为：

$$Prob[1 \leftarrow E(c^*, m_0, m_1, Hist-CCA) \mid c^* = \varepsilon_K(m_1)] = \frac{1}{2} \pm Adv \qquad [6.2]$$

- 抗不可区分的选择密文攻击的安全性：

对不可区分选择密文攻击的安全性描述：一个安全参数为 k 的密码体制被称为对于 IND-CCA 是安全的，若任何多项式有界的敌手在进行上述攻击游戏之后，式[6.2]给出的优势是关于 k 的可忽略量。

6.1.3、不可区分适应性选择密文攻击

IND-CCA 并不是很强的攻击，因为在 IND-CCA 中，我们假设敌手只能在有限的时间里获得解密服务，因此敌手需要在访问解密机之前事先准备好一些密文，而这些密文的选择与敌手在进行解密训练过程中得到的知识没有任何关系。

为了描述更安全性的密码体制，我们给予敌手更强的攻击能力。在这里，我们除去了对敌手访问解密机的时间限制，这样一来，敌手就能在解密训练的过程中根据 O 返回的上一次的解密信息来适应性的构造下一个密文，并再提交 O 进行解密。由于敌手是在适应性地构造密文，因此这种攻击称为不可区分适应性选择密文攻击(IND-CCA2)。

- 不可区分适应性选择密文攻击(IND-CCA2)：

前提假设：与不可区分的选择明文攻击一样，敌手首先和 O 商定一个目标密码体制，且 O 选定了一个加密密钥。

1）敌手向 O 发送了一条准备好的密文消息 $c \in C$。

2）O 解密 c，返回解密结果给敌手。

3）敌手根据解密结果适应性地构造下个密文 c'，然后重复步骤 1、2 多项式次，进行适应性的解密训练。

4）一旦敌手对于上述解密训练课程感到满意，他就要求 O 进行 IND-CPA 的游戏。在该游戏中 E 选择两条明文消息 m_0 和 m_1 发给 O，并接收 O 返回的密文 c^*。

5）收到 c^* 后，敌手仍有时间构造密文 c''，提交 O 进行解密，此时 c'' 是敌手综合前 4 步所得的信息适应性地构造的。这一步可重复多项式次，直到敌手满意为止。

6）最后敌手回答 0 或者 1，作为不可区分选择明文攻击中对 O 投掷硬币的猜测。

若敌手进行密文分析训练课程的历史表示为 Hist-CCA2，则敌手的优势为：

$$Prob[1 \leftarrow Malice(c^*, m_0, m_1, Hist-CCA2) \mid c^* = \varepsilon_K(m_1)] = \frac{1}{2} \pm Adv \qquad [6.3]$$

- 抗不可区分适应性选择密文攻击的安全性：

对不可区分适应性选择密文攻击的安全性描述：一个安全参数为 k 的密码体制被称为对于 IND-CCA2 是安全的，若任何多项式有界的敌手在进行上述的攻击游戏之后，式[6.3]给出的优势是关于 k 的可忽略量。

6.1.4、加密体制的选取

由于在公钥体制中公钥具有公开性，所以任何人都能完全控制加密算法来得到加密服务。也就是说，IND-CPA 攻击永远对公钥体制有效，所以一个公钥体制必须抵抗 IND-CPA 攻击，否则就不算一个有效的加密系统。

一般而言，大多数的公钥密码体制所基于的数学问题都有一些很好的代数结构性质，比如闭包、结合律和同态性。一个主动敌手能运用这些良好的代数性质精心地构造一条密文，提交解密者请求解密。若敌手能得到解密服务，则他可能通过对解密信息实施巧妙的计算而获得有关明文的消息（如对 GM 体制的攻击[47]），或者能破解整个密码体制（如对 BG 密码体制的攻击[117]）。在现实中，由于安全协议的“自动询问-应答”机制以及用户未必知道攻击的存在而无防备地提供解密，敌手很容易获得解密服务，因此公钥体制特别容易受到 IND-CCA 攻击与 IND-CCA2 攻击。但 IND-CCA 攻击模型中对敌手访问解密机有限制，不易恰当地描绘出敌手的攻击能力，因此应主要考虑 IND-CCA2 攻击。鉴于此，在公钥体制下设计安全协议时，我们需采用抗 IND-CCA2 安全的加密算法。目前在公钥体制下比较著名的、对 IND-CCA2 攻击是“可证明安全”的加密体制有 RSA-OAEP 方案[118]和 Cramer-Shoup 公钥加密方案[119]等。

所谓“可证明安全”，是指首先确定密码系统的安全目标，并将该安全目标与一个著名的计算复杂度理论中的难题相联系，然后根据敌手的实际能力构造一个敌手模型，最后给出一个有效的归约变换，指出若敌手模型中的敌手能有效地破坏该安全目标，那么我们就能找到对困难问题的一个有效解。换句话说“可证

明安全”也可称为“归约安全”。

可以看出，“可证明安全”是以若干困难问题为基础的，在公钥体制下我们通常以整数分解或者离散对数等困难问题为基础来讨论“可证明安全”的加密方案。但相对于公钥体制而言，分组密码体制在很多方面还不很成熟，由于很难找到相关的困难问题，目前其安全性还完全建立在经验主义的基础之上。它们之所以被认为是安全的，是因为目前还没有针对性的有效攻击。而作为在各种安全系统中起着重要作用的密码算法，这种状况显然是大家不愿意看到的。可证明安全理论在分组密码中的研究始于 Luby 和 Rackoff 对 DES 的工作[120]，他们形式化地描述了分组密码的密码特性，依据不同的攻击，定义了伪随机置换和超伪随机置换的概念。如果分组密码对选择明文/消息是安全的，则称它是伪随机的。如果分组密码对选择明文（消息）/ 密文是安全的，则称它是超伪随机的。Luby 和 Rackoff 的主要结果是：如果轮函数是伪随机函数，则 3 轮 Feistel 结构是伪随机的，4 轮 Feistel 是超伪随机的。

由于目前普遍采用伪随机性函数来构造安全协议，因此我们也采用伪随机函数构造基于对称钥方案的认证密钥交换协议，所采用的方案是，假设存在分组算法所构造的具有伪随机性的函数，则采用概率加密的办法来实施加密。

在此，我们还要特别指出具有伪随机性的函数与其它重要的密码学本原（Cryptographic primitives,是指安全方案或协议的最基本组成构件或模块,例如某个基础密码算法或数学难题等）如：单向函数、伪随机数产生器以及数字签名方案在理论上是等价的，因为其中任何一个密码学本原的存在都意味着其他密码学本原的存在[121]。

6.2、一个协议及其攻击

上面讨论了加密体制的安全性，但仅采用安全的加密体制能否保证协议的安全呢？下面我们来看一个协议及其攻击。

设 A 的公钥为 K_A，B 的公钥为 K_B，E 的公钥为 K_E，Na 与 Nb 为一次性随机数，安全参数为 k，假设所采用的加密算法为抗 IND-CCA2 安全的。

1. $A \rightarrow B:\{Na, A\}_{K_B}$
2. $B \rightarrow A:\{Na, Nb\}_{K_A}$
3. $A \rightarrow B:\{Nb\}_{K_B}$

由于该协议中的加密体制采用抗 IND-CCA2 安全的算法，所以敌手进行关于 k 的多项式次的 IND-CCA2 攻击练习后，也不会找到明密文间的有效联系，因此该协议中的密文应该说是安全的。但是该协议并不安全，在 1.3.1 节我们已经指出了针对该协议的攻击。敌手攻击成功的原因在于第 2 步消息不清晰，由于缺少认证因素，使得 A 并不知道该消息属于哪个会话。这说明仅采用可靠的密码体制并不能保证所设计的协议安全。

6.3、安全因素

我们通过研究发现，一个安全的认证密钥交换协议应满足两个性质：认证性、保密性。下面我们基于这两个方面来考虑如何设计安全的认证密钥交换协议。

6.3.1、认证性

在设计安全的认证密钥交换协议时，保证协议的认证性是极其重要的。认证性不仅保证协议的双方在协议运行结束后相互确认对方的身份，更重要的是它还可有效地防止敌手对协议进行如 6.2 节所描述的冒名攻击。

为了保证协议的认证性，我们研究发现下面的消息元素是必需的：

- 通信双方的主体标示符

在协议的重要消息中加入通信双方或者意定通信方的主体标示符能使消息的含义清晰化，从而抵抗敌手的交互穿插组合攻击。比如，标示符与会话密钥的适当结合能让主体确认该会话密钥是属于哪两个主体的，从而确保不给敌手提供未授权的解密服务。

- 随机数

在安全协议的相互认证中，可以使用一次性随机数作为 *nonce*，该 *nonce* 在每次会话中都有不同的值，将它与其他消息相结合，可在一定程度上防止敌手的重放攻击。

- 确认值

确认值是一方主体(假设为 A)将通信方(假设为 B)的一次性随机数与一个 B 能验证的 A 的秘密相结合而产生的消息，它的作用是让 A 能够向 B 表明：按照

协议步骤，A 已经得到了 B 所发出的消息，通过该确认值，B 就能认证 A 的存在性。在传统的认证协议中，确认值是通过加密的方式构造的，比如 NSSK 协议：

1) $A \rightarrow S : A, B, Na$

2) $S \rightarrow A : \{Na, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$

3) $A \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$

4) $B \rightarrow A : \{Nb\}_{K_{AB}}$

5) $A \rightarrow B : \{Nb-1\}_{K_{AB}}$

在该协议中，A 向 B 提交了一个确认值 $\{Nb-1\}_{K_{AB}}$，B 通过解密出 $Nb-1$ 可确认 A 拥有解密 $\{Nb\}_{K_{AB}}$ 的能力，从而确认了 A 拥有临时会话密钥 K_{AB}。

6.3.2、正确的密码学保护

我们在 6.1 节描述了密码体制的安全性，在 6.3.1 节描述了协议为保证认证性所必备的一些消息元素，下面我们讨论各种消息元素所需的正确安全服务，以便对协议进行有效保护。

对于临时会话密钥而言，它需要在通信的双方或者三方之间秘密共享，所以应采用安全的加密算法来提供保密服务。在公钥体制下，由于已发现对密码体制的 IND-CCA2 攻击，所以在不能保证协议主体不为敌手提供解密服务的情况下，我们应采用抗 IND-CCA2 安全的加密体制如 Cramer-Shoup 公钥加密方案来保证临时会话密钥的安全性。在对称钥体制下，由于还没有发现针对分组密码算法的有效的 IND-CCA 与 IND-CCA2 攻击，所以可利用基于分组算法的伪随机函数实施概率加密来保证临时会话密钥的安全性（可抗 IND-CPA，安全性分析见第八章）。

虽然找到了合适的加密算法，但加密算法的使用应符合密码学中的一条原则，即最小程度的使用保密服务，这能限制可能有利于密码分析的信息泄露的数量。但通过对传统安全协议的观察，我们发现很多协议中的公开信息如主体标示符、一次性随机数均被加密来提供认证性，例如 6.3.1 节中的 NSSK 协议通过对一次性随机数 Nb 加密来构造确认值，实现 A 向 B 的认证。但在第三章我们采用 CS 逻辑对该协议的分析中已经得出：Nb 的保密性与认证性无关，认证性其实是

通过临时会话密钥 K_{AB} 的保密性与 Nb 的完整性来实现的。由于带密钥的单向函数可利用保密的 K_{AB} 来确保 Nb 的完整性，因此使用这种单向函数作用于一个随机数来构造确认值才是正确的选择。

6.4、对 Yahalom 协议的改进

Yahalom 是一种经典的认证密钥交换协议，参与协议的主体有通信双方 A、B 和认证服务器 S，其目的是在通信双方之间分配临时会话密钥，并进行通信双方的确认。

原始的 Yahalom 协议如下：

1. $A \rightarrow B: A, Na$
2. $B \rightarrow S: B, \{A, Na, Nb\}_{K_{BS}}$
3. $S \rightarrow A: \{B, K_{AB}, Na, Nb\}_{K_{AS}}, \{A, K_{AB}\}_{K_{BS}}$
4. $A \rightarrow B: \{A, K_{AB}\}_{K_{BS}}, \{Nb\}_{K_{AB}}$

该协议存在缺陷。使用 BAN 逻辑对该协议进行形式化分析后，会发现如果 A 在第 4 步选择一个旧密钥重放给 B，则 B 发现不了这个问题[49]。因此 BAN 逻辑的设计者们对 Yahalom 协议进行了如下改进：

1. $A \rightarrow B: A, Na$
2. $B \rightarrow S: B, Nb, \{A, Na\}_{K_{BS}}$
3. $S \rightarrow A: Nb, \{B, K_{AB}, Na\}_{K_{AS}}, \{A, K_{AB}, Nb\}_{K_{BS}}$
4. $A \rightarrow B: \{A, K_{AB}, Nb\}_{K_{BS}}, \{Nb\}_{K_{AB}}$

协议改进后，第 4 条消息中加入了一个 *nonce*：Nb，这样 A 就不能重放以前的消息来欺骗 B 了。但该协议仍存在缺陷：由于 B 在收到临时会话密钥后没有向 A 发出确认值，所以 A 无法对 B 进行认证，利用这个缺陷，敌手能通过冒充 B 来欺骗 A[122]。下面我们按照 6.3 节的要求，对改进的 Yahalom 协议再作修改：

1. $A \rightarrow B: A, Na$
2. $B \rightarrow S: B, Nb, [A, Na]_{K_{BS}}$
3. $S \rightarrow A: Nb, [A, B, \{K_{AB}\}_{K_{AS}}, Na]_{K_{AS}}, [A, B, \{K_{AB}\}_{K_{BS}}, Nb]_{K_{BS}}$
4. $A \rightarrow B: [A, B, \{K_{AB}\}_{K_{BS}}, Nb]_{K_{BS}}, [Nb]_{K_{AB}}$
5. $B \rightarrow A: [Na]_{K_{AB}}$

在该协议中我们对消息采用了两种保护措施：一是对协议中不需要保密的消息如 *nonce*、主体标示符、密文本身采用了具有消息源识别的消息完整性保护；二是对需要保密的消息：临时会话密钥，采用了加密方式进行保护。此外，在协议最后我们添加了一个确认值$[Na]_{K_{AB}}$，用于主体 B 在收到临时会话密钥后对 A 确认。另外，通过观察改进的 Yahalom 协议会发现，协议消息中仅出现了临时会话密钥的密文，并没有出现其明文，这说明协议本身不会给任何人提供解密服务，这大大增加了敌手攻击加密消息的难度。在后面的章节，我们将分析该协议的安全性。

6.5、对 Needham-Schreoder 公钥认证协议的改进

在文献[47]中，Mao 已对 Needham-Schroeder 公钥认证协议使用单向函数进行了改进。但是我们发现改进后的协议仍不合理。

改进后的 Needham-Schroeder 公钥认证协议：

1. $A \rightarrow B$: $[\{Na\}_{K_B}, A]_{K_A}$
2. $B \rightarrow A$: $[\{Na, Nb\}_{K_A}]_{K_B}$
3. $A \rightarrow B$: $[\{Nb\}_{K_B}]_{K_A}$

我们研究发现，该改进方案仍没摆脱利用主体能解密特定消息能力来向对方进行认证的思路。由于带密钥的单向函数本身就具有认证性，而且一次性随机数属于公开消息，因此可直接利用单向函数来设计认证协议。我们的改进方案如下：

1. $A \rightarrow B$: Na, A, B
2. $B \rightarrow A$: $[Na, Nb, A, B]_{K_B}$
3. $A \rightarrow B$: $[Nb, A, B]_{K_A}$

在该方案中，A、B 双方利用私钥构造签名消息的能力就能向对方进行认证，从而避免了效率低下的公钥加密算法。

6.6、本章小结

本章的主要工作是讨论如何设计一个安全的认证密钥交换协议。主要工作有以下几个方面：1. 从三种敌手模型入手，介绍了有关加密体制的三种实用的安

全性概念，并讨论了在公钥体制与对称钥体制下安全协议分别适合采用何种安全级别的加密算法。2．通过一个安全协议的攻击指出，在设计协议时仅采用安全的密码体制是不够的，然后从认证性与保密性入手，讨论了如何设计安全的认证密钥交换协议：一方面，为了保证协议的认证性，协议必需含有某些特殊的消息元素；另一方面，对于不同的消息元素，应采用不同的密码体制进行保护。3．按照上述的研究结果，给出了 Yahalom 协议与 Needham-Schroeder 协议的改进方案。

第七章 计算理论下认证密钥交换协议的安全性

在本章，我们将在计算理论下讨论认证密钥交换协议的安全性。

7.1、基本概念

7.1.1、消息驱动的认证密钥交换协议

认证密钥交换协议的执行可看成是通过消息来驱动的：对于协议中进行点对点通信的两个主体，首先由一个外部请求触发一方主体来执行协议的一个副本，然后该主体与其通信方依赖消息的交互来运行。在运行中，当主体收到一条消息时，则按照协议规定对其进行处理，处理完毕后，可根据协议规定执行下列步骤之一：接收并处理消息、产生并发出新消息、等待下一条消息的到达、拒绝执行协议、结束协议。在协议结束时，通信双方协商出一个临时秘密会话密钥，同时完成协议双方的认证。

7.1.2、匹配对话

每个主体执行的一次协议称为一个会话。每个主体的会话中，所有输入消息的总体可用形式$(P_i,P_j,s,role)$来代表，其中P_i为会话主体，P_j为主体的通信方，s为会话标示符（唯一标示主体的一次会话），$role \in \{initiator,responder\}$，用来指明会话主体为发起者还是响应者。若主体$P_i$的一个会话的输入消息与$P_j$的一个会话的输入消息有这样的形式：$(P_i,P_j,s,initiator)$和$(P_j,P_i,s,responder)$，则称这两个会话为匹配对话。由于每对匹配对话的标示符均唯一，我们可用该标示符来代表该匹配对话。

7.2、认证密钥交换协议的安全性

因为认证密钥交换协议涉及到会话双方的相互认证以及临时会话密钥的秘密交换，所以我们应从以下两点考虑协议的安全性：一是协议的认证性；二是秘

密消息的保密性。

7.2.1 分析协议的认证性

为了从计算角度分析协议的认证性，我们设计了两个攻击游戏，并通过主体在游戏中的成功概率来考察协议的认证性。为此，我们扩展了主体的能力：主体需要在协议结束时输出一个结果，作为对敌手行为的猜测。

针对一个安全参数为 k 的认证密钥交换协议，敌手首先任意选择两个主体 A 和 B，并将其之间的一对会话作为测试对话（假设 A 为发起者，B 为响应者），然后对其他主体的任意会话与主体 A 和 B 的除测试对话外的任意会话实施其所能发动的各种攻击，如窃听、重放、篡改、收买合法方等等，攻击次数可以是关于安全参数的多项式次，这些攻击可看作敌手的攻击练习。当敌手对所作的攻击练习感到满意时，对选择的测试对话进行如下两个游戏，其中要求 A 和 B 的任何秘密均没受到攻击（即 A、B 没有被收买）。

游戏 1：针对测试对话，敌手首先选取一个随机位 b，若 b=0，则敌手发出一个外部请求，触发 A 与 B 建立匹配对话；若 b=1，则敌手根据在攻击练习中获得的知识，设法冒充 A 与 B 建立对话。在对话终止后，B 输出 b'，作为对 b 的猜测。

游戏 2：针对测试对话，当 A 发出消息欲与 B 建立匹配对话时，敌手首先选取一个随机位 b，若 b=0，则敌手将 A 的消息传递给 B，触发 B 与 A 建立匹配对话；若 b=1，则敌手根据在攻击练习中获得的知识，设法冒充 B 与 A 建立对话。在对话终止后，A 输出 b'，作为对 b 的猜测。

定义 7.1：当满足下列条件时，认证密钥交换协议满足认证性：

敌手在进行游戏 1 和 2 时，主体 A 和 B 分别能以 1/2+一个关于安全参数 k 的不可忽略的概率正确猜测出 b 来。

说明：在两个游戏中，敌手分别攻击协议的发起者和应答者。若在游戏中主体 A 和 B 均能以 1/2+一个关于安全参数 k 的不可忽略的概率正确猜测出 b 来，说明两主体均能辨别出敌手是否在进行冒充攻击，则协议具有认证性。

下面我们根据这两个游戏来考察改进后的 Needham-Schreoder 认证密钥交换协议的认证性。

● 改进 Needham-Schroeder 认证密钥交换协议的描述

A 和 B 分别与可信中心 S 共享一个长期密钥 K_{AS} 和 K_{BS}，长度均为 k。$f1()$、$f2()$：$\{0,1\}^k \to \{0,1\}^k$ 为两个不带密钥的单向函数，且 $K_{BS1}=f1(K_{BS})$、$K_{AS1}=f1(K_{AS})$、$K_{BS2}=f2(K_{BS})$、$K_{AS2}=f2(K_{AS})$。Na 为 A 的一次性随机数，Nb、Nb' 为 B 的一次性随机数，长度均是 k；$\{x\}_K$ 表示安全的对称加密方案 E 对 x 实施加密的结果。$[x]_K=(x, prf_K(x))$，其中 $x \in \{0,1\}^*$，$prf_K:\{0,1\}^* \to \{0,1\}^k$ 是由密钥 K 控制的伪随机函数[123]。

1) $A \to B: A, Na$
2) $B \to S: [A, Na, B, Nb]_{K_{BS2}}$
3) $S \to B: [\{K_{AB}\}_{K_{BS1}}, Nb, A, B]_{K_{BS2}}, [\{K_{AB}\}_{K_{AS1}}, Na, A, B]_{K_{AS2}}$
4) $B \to A: [\{K_{AB}\}_{K_{AS1}}, Na, A, B]_{K_{AS1}}, [Na]_{K_{AB}}, Nb'$
5) $A \to B: [Nb']_{K_{AB}}$

● 分析认证性

从协议中能看出，会话密钥 K_{AB} 的密文与通信双方的标示符和一次性随机数通过单向函数绑定在一起，具有消息完整性与消息源认证性，主体能清晰地分辨出 K_{AB} 是属于哪两个主体的哪次匹配会话，所以敌手无法实施针对协议的穿插组合攻击（相关的形式化证明在第八章给出）。但敌手若能伪造 $[Nb']_{K_{AB}}$（$[Na]_{K_{AB}}$），则可冒充 A（B）与 B（A）成功完成协议，实施攻击。

为了论述方便，我们仅分析敌手冒充 B 的情况，冒充 A 的情况与此类似。

若敌手可冒充 B，则存在两种可能，第一：敌手通过 Na 直接构造出 $[Na]_{K_{AB}}$；第二：敌手通过 Na 和 $\{K_{AB}\}_{K_{BS1}}$（或者 $\{K_{AB}\}_{K_{AS1}}$）计算出 $[Na]_{K_{AB}}$。（因为 $[X]_K=(X, prf_K(X))$，所以敌手从消息中可直接得到密文 $\{K_{AB}\}_{K_{BS1}}$ 和 $\{K_{AB}\}_{K_{AS1}}$）。

对于这两种情况，我们分别进行讨论。

第一. 若敌手能通过 Na 直接计算出 $[Na]_{K_{AB}}$，那么我们可构造一个多项式时间的区分器来区分真随机函数和伪随机函数。

我们考虑两个实验。

在第一个实验中采用真随机函数来计算 $[Na]_{K_{AB}}$。由于 $[Na]_{K_{AB}}$ 的密文部分在长度为 k 的空间随机分布，所以敌手能伪造它的概率为 2^{-k}，当安全参数 k 足够大时，该概率可忽略不记。

在第二个实验中采用伪随机函数来计算$[Na]_{K_{AB}}$。假设这种情况下，敌手成功的概率为 k 的不可忽略量，则我们可构造一个多项式时间的区分器 Y，来区分真随机函数和伪随机函数。

设 Y 拥有一个随机预言机 O，O 按以下方式工作：当 Y 访问 O 时，O 选择一个随机比特 b（b 不被 Y 所知），若 b=0，则 O 输出函数 g 为一个真随机函数；否则 O 输出 g 为伪随机函数。为了猜测 g 为真随机函数还是伪随机函数，Y 构造一个虚拟认证密钥交换协议 π 来模拟 Needham-Schroeder 协议的运行，π 中有 n 个主体和一个可信中心 S。对于 n 个主体，Y 用密钥产生器在密钥空间中随机产生 n 个密钥，作为这些主体与 S 的长期密钥。Y 随机选取两个主体 A 和 B，并将他们之间的一对会话的标示符记为 s^*，作为测试对话。

敌手需要建立的会话都由 Y 代表主体来运行。当敌手要建立除 s^* 外的所有对话来实施攻击练习时，Y 都按照协议来运行，并用函数 g 来计算确认值。当敌手要建立测试对话 s^* 时，Y 停止敌手的运行，并随机输出 b' 作为对 O 选择比特 b 的猜测。（即在敌手的攻击练习阶段不能攻击对话 s^*）

当敌手要破坏一次协议执行的认证性时，若会话的主体是除 A、B 外的任意两主体，则 Y 将两主体的长期密钥交给敌手，则敌手可解密出临时会话密钥，并能正确计算出确认值；若会话的一方是 A 或 B，或者两方是 A 和 B 但不是会话 s^*，则 Y 将临时会话密钥解密后交给敌手，则敌手也可正确计算出确认值（让敌手作足够的选择消息的攻击练习）。这个过程进行 k 的多项式次，直到敌手满意为止。

最后，Y 建立测试对话 s^*，访问 O 来获得函数 g，并让敌手来构造 B 的确认值$[Na]_{Kab}$。

若测试对话成功完成（因为 Y 控制着测试会话的运行，所以 Y 知道该测试会话的双方能否最终达到接受状态），则说明敌手构造确认值成功，Y 就输出 1；若协议失败，则说明敌手构造确认值失败，Y 就输出 0。

这样我们看到 Y 区分真随机函数与伪随机函数的概率与敌手成功构造确认值的优势相同，都是关于 k 的不可忽略量。这与我们认为不存在多项式时间的区分器这样做相矛盾。

第二．若敌手通过 Na 和$\{K_{AB}\}_{K_{BS}}$（或者$\{K_{AB}\}_{K_{AS}}$）计算出$[Na]_{K_{AB}}$来冒充 B，这说明敌手能够得到临时会话密钥 Kab，这就涉及到协议采用算法的保密性问题。为了讨论的方便，我们这里先假设协议采用的算法是安全的，即敌手只能以

关于 k 的可忽略的概率通过 Na 和 $\{K_{AB}\}_{K_{BS1}}$ (或者 $\{K_{AB}\}_{K_{AS1}}$)计算出 $[Na]_{K_{AB}}$ 来。(关于保密性的讨论见 7.2.2 节)

通过上述两种情况的分析与假设，我们认为敌手构造出 $[Na]_{K_{AB}}$ 的可能性是关于 k 的可忽略量。

此时，我们可以构造游戏 2：

当 A 接到一个外部请求与 B 建立测试对话 s^* 时，敌手首先选择一个随机位 b，当 b=0 时，敌手激活 B 同 A 进行会话；当 b=1 时，敌手冒充 B 同 A 进行会话。由于敌手不能有效计算出 $[Na]_{K_{AB}}$，所以当 b=1 时 A 以 1/2+一个关于安全参数 k 的不可忽略的概率验证确认值失败。因此 A 就能通过验证确认值成功或失败以 1/2+一个关于安全参数 k 的不可忽略的概率猜测出 b 来。

敌手冒充 A 的情况与此同理。

7.2.2 保密性的讨论

本节我们讨论有关加密消息的保密性问题。

- 公钥体制下的保密性问题

若认证密钥交换协议是在公钥体制下实现的，则采用 IND-CCA2 安全的加密体制是保护临时会话密钥的最佳方案。

我们之所以采用 IND-CCA2 安全的加密算法是因为很多协议消息不具有消息源认证性，利用这一点，敌手会获得合法用户的未授权的解密服务。比如 6.2 节所描述的攻击：由于消息 $\{Na,Nb\}_{K_A}$ 不具有消息源认证性，所以敌手通过重放就得到了解密消息 Nb。结合敌手在第一条消息中得到的 Na，则敌手通过重放攻击，在协议中就解密出了消息 $\{Na,Nb\}_{K_A}$ 的明文 Na,Nb。

在协议消息受到单向函数的保护时，消息受到消息源识别的消息完整性保护，合法主体就不会给敌手提供未授权的解密服务，在这种情形下能否采用安全性弱一些的的加密算法呢？比如我们对 6.2 节的协议采用如下的改进方式：

1. $A \rightarrow B:[\{Na\}_{K_B},A]_{K_A}$
2. $B \rightarrow A:[\{Na,Nb\}_{K_A}]_{K_B}$
3. $A \rightarrow B:[\{Nb\}_{K_B}]_{K_A}$

由于该协议中，消息均具有消息源识别性，所以敌手不能发起 6.2 节所描述的重放攻击，在这种情况下，我们是否能采用 IND-CPA 安全的加密体制呢？

我们通过对各种公钥加密算法的研究发现，有些 IND-CPA 安全的加密体制在这种情形下仍然不安全。虽然敌手不能获取未授权的解密服务，但是敌手作为一个合法的主体，仍能与别的主体执行协议，正大光明地获取对方主体的解密服务，而在有些 IND-CPA 安全的加密体制下，只要敌手获得了解密服务，就可以在概率多项式时间内攻破整个密码体制：在 BG 公钥体制下，其加密方案实质是 IND-CPA 安全的 Rabin 加密，但为敌手提供的解密服务能使敌手在概率多项式时间内分解模 N。

因此，在无法对各种公钥算法进行深入研究的情况下，安全协议在采用公钥加密算法时，选择已证明的 IND-CCA2 安全的加密算法为最佳。此时我们可认为敌手通过密码分析只能以关于安全参数的可忽略的概率从密文得到已知明文。

- 对称钥体制下的保密性问题

基于对称钥体制的安全协议在采用加密算法时，一般采用概率加密算法，其安全性考虑在于伪随机函数的可证明安全性。其具体实施的步骤是首先由分组密码构造一个具有足够伪随机性的函数 f，然后通过下述形式进行加密：$\{M\}_K \overset{def}{=} (r, f_K(r) \oplus M)$，其中 K 为伪随机函数的密钥，r 为一个随机量。

由于伪随机函数具有可证明的 IND-CPA 安全性，$f_K(r)$ 的输出与同等长度的真随机数产生器的输出在多项式时间内无法区分，因此 $f_K(r)$ 的输出能将消息 M 很好地随机化，可以证明概率加密方案也是 IND-CPA 安全的（具体证明见第八章）。

在协议采用单向函数来设计时，消息具有消息源识别性，敌手无法获得其他主体的解密预言服务，所以无法针对别的主体间的临时会话密钥发起选择密文攻击。并且由于敌手作为一个合法主体与另一方执行协议从而获得解密服务时，已与对方共享相应的临时会话密钥，所以也不能获得任何有用信息，由此我们认为，在协议消息具有消息源识别的情况下，对称钥体制下的安全协议采用 IND-CPA 安全的加密体制即可，此时我们也可以认为敌手通过密码分析只能以关于安全参数的可忽略的概率从密文得到明文。

7.3、本章小结

本章讨论了在计算理论下如何考察认证密钥交换协议的安全性。主要工作有

以下两个方面：1. 首次提出了通过两个攻击游戏来刻画协议认证性的想法，按照主体在游戏中的成功概率给出了协议认证性的定义，并通过一个实例介绍了如何在计算理论下分析协议的认证性。2. 对公钥体制下与对称钥体制下的协议应采用何种安全级别的加密方案进行了讨论，在协议的形式化描述与协议的具体实现之间搭起了一座桥梁，并为安全地实现协议指出了方向。

第八章 符号理论和计算理论相结合的分析法

在第七章，我们讨论了计算理论下认证密钥交换协议的安全性，即在敌手能进行密码分析练习的情况下，怎样考察协议的认证性和保密性。这种定义基于敌手对密码体制的攻击能力，能有效地分析协议采用算法的安全性。但它也有不足之处：无法分析协议的消息是否完备，无法考察协议的逻辑可靠性，这将导致不能有效分析出敌手所造成的非密码分析的攻击。比如在对改进的Needham-Schroeder认证密钥交换协议的安全性分析中，对于协议是否能遭受穿插组合攻击只能通过非形式化的方法给出。

从第三、四章与第七章可以看出，在安全协议的形式化分析中，符号理论与计算理论各有优缺点。基于扬长避短、优势互补的考虑，综合这两种理论，我们提出了一种调和的分析方法，这种方法的核心思想是将符号理论中的模态逻辑法和计算理论中的矛盾归约法相结合，建立一种既能分析协议逻辑可靠性，又能分析计算可靠性的形式化分析工具。

8.1、通信的形式化

协议的抽象由关于下列输入值的一个概率多项式时间函数 P 来描述.

1^k：安全参数。

A，B：表示主体。

S：为可信中心。

$a1,a2$：主体之间，或者主体与 S 之间共享的长期对称密钥。

r：主体的随机输入，可将 r 看作主体生成的随机数。

X，Y，Z：均表示主体在协议运行中的所有输入的集合。

协议中的敌手 E 控制着整个通信网络，作为一个主动攻击者，对于任一组给定的主体 A、B、S，他不仅能观察主体间的交互信息，而且能在主体间进行任意次的会话，他能说服一个主体开始一次协议的运行，就像是在和另一个合法主体运行该协议。

敌手 E 的攻击行为符合 4.2.3 节提出的敌手模型，值得注意的是，在 E 进行密码分析训练时，我们认为他能够贿赂一些合法的主体来得到他们的秘密，但我

们假设 E 对于要攻击的目标得不到任何相关的秘密。

主体 A、B、S 能作为敌手 E 的黑盒形式的预言机。这意味着 E 能通过向主体提供输入值来进行提问，而主体可在 E 的输入中加入自己的秘密输入 $\alpha 1, \alpha 2, r$ 进行运算，然后发出一条消息。该消息或者是一条需要发往对方的消息，或者是一个判决：{*Accept*,*Reject*}：当协议最终完成时，输出 *Accept*，当主体发现协议错误时，输出 *Reject*。若该消息是发往对方的，那么 E 能得到该消息并继续攻击。将 $\tilde{X},\tilde{Y},\tilde{Z}$ 记为 E 进行主动攻击时，各主体进行运算的所有输入值，此时，一次协议的运行可记为：$P(A(\tilde{X}),B(\tilde{Y}),S(\tilde{Z}))$。

我们假设总存在一种特别友好的“良性敌手”，他的行动只限于选择一组主体 A，B，S 作为他的一组随机预言机，然后如实地将一方主体产生的消息传给正确的另一方主体。因此，可将良性敌手看作是连接主体间的一根导线，此时一次协议的执行可记为：$P(A(X),B(Y),S(Z))$。

8.2、完全性与正确性

对于一个有可信中心 S 参与的认证协议，设 L 是 $\{0,1\}^*$ 上的一种语言，$(X=x_1,x_2,...x_l)\in L$，$(Y=y_1,y_2,...y_l)\in L$ 和 $(Z=z_1,z_2,...z_l)\in L$ 分别是 A、B 和 S 的输入，通信的各方通过一个被敌手 E 控制的通信信道交换信息。协议 P 可在关于 $|X\|Y\|Z|$ 的一个多项式时间内完成（$X\|Y\|Z$ 为消息的级联，$|X\|Y\|Z|$ 为级联消息的长度），且输出的类型为：$P(A(X),B(Y),S(Z))\in\{Accept,Reject\}$。

定义 8.1：设 L 是 $\{0,1\}^*$ 上的一种语言，P 称为 L 上的一个有效安全协议，若：

$Prob[P(A(X),B(Y),S(Z))=Accept\,|\,X,Y,Z\in L]=1$，且

$Prob[P(A(\tilde{X}),B(\tilde{Y}),S(\tilde{Z}))=Accept\,|\,\tilde{X},\tilde{Y},\tilde{Z}\in L]\le\delta$

其中 δ 是常数，满足 $\delta\in[0,\frac{1}{2})$.

第一个公式刻画了协议的完全性概念：若敌手是良性的，协议将以概率 1 成功。第二个公式刻画了正确性概念：若敌手进行主动攻击，最多以 δ 的概率攻击成功。

定义 8.2：对于一个有效的认证协议 P，若 δ 可以小到忽略不计，那么 P 是

一个安全的安全协议。

8.3、对 MBL 逻辑的修改和扩展

1. 添加的符号

在调和方法中所用到的符号含义与 MBL 逻辑基本相同，除此之外，增加了符号$\supset_k$，表示以关于 k 的概率推导出；$rand(m)$，表示 m 由随机数产生器产生。

2. 对语义的修改及扩展

(1) 在 MBL 语义中加入：

语义：随机数

$(r,t)\models rand(m)$ *iff* $(r,t)\models \exists 0<t'\le t, m\in_R \{0,1\}^k$

$\in_R$ 表示由随机数产生器随机产生，k 为安全参数。

(2) 修改语义 9：可信中心管辖密钥语义

$(r,t)\models S$ *controls*(K) *iff if* $(r,t)\models S$ *says*(K) *then* $(r,t)\models rand(K)$

3. 推理规则的修改与扩展

规则 2：可识别消息来源的完整性规则：

$$A\ bels(good(K,\tilde{A},B))\wedge A\ verifies[M]_K \supset_k$$

$$A\ bels(\odot M)_B \wedge A\ bels(M\in public())\wedge A\ bels(K\in H_B)$$

规则 7：保密性规则

$$A\ bels(good(K,A,B))\wedge A\ bels\odot\{M\}_K \wedge A\ bels(m\notin public())\wedge A\ bels(m\in M)\supset_k$$

$$A\ bels(\otimes_{A\leftrightarrow B} m)$$

规则 9：随机数规则

$$A\ bels(rand(M))\supset_k A\ bels(fresh(M))$$

证：因为 A 相信 M 是在 $\{0,1\}^k$ 上均匀随机产生的，那么在 t 时 A 必然相信以前任一主体说过 M 的概率为 2^{-k}，所以 A 能以 $1-2^{-k}$ 的概率认为 M 是新鲜的。

8.4、改进 Yahalom 协议的安全性分析

8.4.1、协议及其安全目标的描述

- 改进的 Yahalom 认证密钥交换协议

A、B 为通信的双方，S 为可信中心。Na 为 A 的随机数，Nb 为 B 的随机数，它们的大小都是 k。A 和 B 分别与 S 共享一个 $2k$ 比特的密钥：K_{AS1} K_{AS2} 和 K_{BS1} K_{BS2}。协议会话密钥 K 的大小为 $\sigma(k)$，其中 $\sigma(k)$ 为 k 的一个多项式。

1. $A \rightarrow B: A, Na$
2. $B \rightarrow S: B, Nb, [A, Na]_{K_{BS1}}$
3. $S \rightarrow A: Nb, [A, B, \{K\}_{K_{AS2}}, Na]_{K_{AS1}}, [A, B, \{K\}_{K_{BS2}}, Nb]_{K_{BS1}}$
4. $A \rightarrow B: [A, B, \{K\}_{K_{BS2}}, Nb]_{K_{BS1}}, [Nb]_K$
5. $B \rightarrow A: [Na]_K$

- 算法描述

以 K_{AS1} K_{AS2} 为例：f 为抗选择消息攻击安全的伪随机函数，用于消息认证，K_{AS1} 是长度为 k 的密钥：$f_{K_{AS1}}: \{0,1\}^* \rightarrow \{0,1\}^k$，且 $[x]_{K_{AS1}}$ 表示 $(x, f_{K_{AS1}}(x))$。f' 为另一抗选择消息攻击安全的伪随机函数，用于消息加密，K_{AS2} 为其长度为 k 的密钥：$f'_{K_{AS2}}: \{0,1\}^k \rightarrow \{0,1\}^{\sigma(k)}$。使用 K_{AS2} 对 $K \in \{0,1\}^{\sigma(k)}$ 进行的概率加密可定义为：$\{K\}_{K_{AS2}} \overset{def}{=} (r, f'_{K_{AS2}}(r) \oplus K)$，其中 $r \in \{0,1\}^k$ 是一个随机量。

- 消息的前提假设和初始信仰

$A\ bels(good1(K_{AS1}K_{AS2}, A, S))$；$A\ bels(rand(Na))$；$A\ bels(S\ controls(K))$；

$A\ bels(\{A, B, Na\} \subset public())$；$A\ bels(\{A, B, Na, K_{AS1}K_{AS2}\} \in H_A)$

$B\ bels(good1(K_{BS1}K_{BS2}, B, S))$；$B\ bels(rand(Nb))$；$B\ bels(S\ controls(K))$；

$B\ bels(\{A, B, Nb\} \subset public())$；$B\ bels(\{A, B, Nb, K_{BS1}K_{BS2}\} \subset H_B)$

$S\ bels(good1(K_{AS1}K_{AS2}, A, S))$，$S\ bels(good1(K_{BS1}K_{BS2}, B, S))$；

- 协议目标

一．完全性：若敌手是良性的，则应有：

$A\ bels(good2(K,A,B))$； $A\ bels(K \in H_B)$；

$B\ bels(good2(K,A,B))$； $B\ bels(K \in H_A)$

二．正确性：若敌手是概率多项式时间的，则攻击成功的概率可忽略。

8.4.2、形式化分析

一．完全性分析

协议第二步，当 B 收到 $[A,Na]_{K_{BS1}}$ 后，用 K_{BS1} 对该消息验证通过后，运用修改后的可识别消息来源的完整性规则：

$$B\ bels(good(K_{BS1},B,S)) \wedge B\ verifies([A,Na]_{K_{BS1}}) \supset_k$$

$$B\ bels(\odot(A,Na))_S \wedge B\ bels((A,Na) \subset public()) \qquad [exp1]$$

协议第三步，当 A 收到消息 $[A,B,\{K\}_{K_{AS2}},Na]_{K_{AS1}}$ 后，用 K_{AS1} 对消息 $[A,B,\{K\}_{K_{AS2}},Na]_{K_{AS1}}$ 验证通过后，运用修改后的可识别消息来源的完整性规则：

$$A\ bels(good(K_{AS1},A,S)) \wedge A\ verifies([A,B,\{K\}_{K_{AS2}},Na]_{K_{AS1}}) \supset_k$$

$$A\ bels(\odot(A,B,\{K\}_{K_{AS2}},Na))_S \wedge A\ bels((A,B,\{K\}_{K_{AS2}},Na) \subset public()) \qquad [exp2]$$

由基本逻辑关系得：

$$(A,B,\{K\}_{K_{AS2}},Na) \in Receive_A$$

$$\supset\ (A,B,\{K\}_{K_{AS2}},Na) \in H_A \supset \{K\}_{K_{AS2}} \in H_A \qquad [exp3]$$

由拥有即相信规则可得：

$$A\ bels((A,B,\{K\}_{K_{AS2}},Na) \in H_A),\ A\ bels(\{K\}_{K_{AS2}} \in H_A) \qquad [exp4]$$

对[exp4]运用公理 4 和属于语义，可得：

$$A\ bels((A,B,\{K\}_{K_{AS2}},Na) \in H_A) \wedge A\ bels(\{K\}_{K_{AS2}} \in H_A)$$

$$\supset\ A\ bels(\{K\}_{K_{AS2}} \in (A,B,\{K\}_{K_{AS2}},Na)) \qquad [exp5]$$

对[exp2]和[exp5]运用消息提取规则可得：

$$A\ bels(\odot\{K\}_{K_{AS2}})_S \supset A\ bels(\odot\{K\}_{K_{AS2}}) \qquad [exp6]$$

结合前提假设 $K_{AS2} \in H_A$ 和[exp3]，由语义 7 可得：

$K_{AS2} \in H_A \wedge \{K\}_{K_{AS2}} \in H_A \supset K \in \{K\}_{K_{AS2}}$ [exp7]

由[exp7]，根据拥有即相信规则和基本逻辑关系可得：

$A\ bels(K_{AS2} \in H_A) \wedge A\ bels(\{K\}_{K_{AS2}} \in H_A) \supset A\ bels(K_{AS2} \in H_A \wedge \{K\}_{K_{AS2}} \in H_A)$

$\supset A\ bels(K \in \{K\}_{K_{AS2}})$ [exp8]

此时，A 并没有将 K 加入 $public()$ 集合，所以有：

$A\ bels(K \notin public())$ [exp9]

由前提假设 $A\ bels(good1(K_{AS2},B,S))$、[exp6]、[exp8]、[exp9]，根据保密性规则可得：

$A\ bels(good1(K_{AS2},B,S)) \wedge A\ bels(\odot\{K\}_{K_{AS2}}) \wedge A\ bels(K \in \{K\}_{K_{AS2}}) \wedge$

$A\ bels(K \notin public()) \supset_k A\ bels(\otimes_{A \leftrightarrow S} K)$ [exp10]

由前提假设 $Na \in H_A$，和[exp3]，根据拥有即相信规则可得

$A\ bels(Na \in H_A) \wedge A\ bels((A,B,\{K\}_{K_{AS2}},Na) \in H_A) \supset$

$A\ bels(Na \in (A,B,\{K\}_{K_{AS2}},Na))$ [exp11]

由前提假设 $A\ bels(rand(Na))$ 根据随机数规则可得：

$A\ bels(rand(Na)) \supset_k A\ bels(fresh(Na))$ [exp12]

由[exp11]和[exp12]，根据新鲜性规则可得：

$A\ bels(Na \in (A,B,\{K\}_{K_{AS2}},Na)) \wedge A\ bels(fresh(Na))$

$\supset A\ bels(fresh(A,B,\{K\}_{K_{AS2}},Na))$ [exp13]

由[exp2]和[exp13]，根据新鲜性验证规则可得：

$A\ bels(\odot(A,B,\{K\}_{K_{AS2}},Na))_S \wedge A\ bels(fresh(A,B,\{K\}_{K_{AS2}},Na)) \supset$

$A\ bels(S\ says(A,B,\{K\}_{K_{AS2}},Na))$ [exp14]

由前提条件：$A\ bels(S\ controls(K))$ 和[exp8]、[exp10]、[exp14]，根据良好临时会话密钥规则可得：

$\supset A\ bels(good2(K,A,B))$ [exp15]

在协议的第五步，当 A 结合[exp15]验证 $[Na]_K$ 通过后，运用可识别消息来源

的完整性规则：

$$A\ bels(good2(K,A,B)) \wedge A\ verifies([Na]_k) \supset_k A\ bels(K \in H_B) \qquad [exp16]$$

在敌手不进行有关密码分析的攻击，且密码体制是完善的情况下，结论[*exp2*]、[*exp10*]、[*exp12*]、[*exp16*]都能以概率1推导得出，因此可以得到协议目标 $A\ bels(good2(K,A,B))$ 与 $A\ bels(K \in H_B)$。

同理，在对 B 进行分析时，我们还可以得到协议 $B\ bels(good2(K,A,B))$ 与 $B\ bels(K \in H_A)$。

此外，在形式化分析中我们可以看出，在[*exp1*]中得到的结论在分析中并没有作用，这说明我们在第一条消息中不需要对明文 A 与 Na 进行单项函数的保护。这表明了形式化分析可以作为协议设计的有利补充。

二．正确性分析

从完全性分析中，[*exp2*]、[*exp10*]、[*exp12*]、[*exp16*]是以概率推导的。这说明存在 E 能攻击这四处消息来欺骗主体 B 的可能。下面分析 E 欺骗成功的概率。

对于[*exp2*]，倘若敌手能针对消息 $(A,B,\{K\}_{K_{AS2}},Na)$ 构造出 $[A,B,\{K\}_{K_{AS2}},Na]_{K_{AS1}}$ 来，则敌手可以冒充可信中心 S。下面我们分析敌手冒充 S 成功的概率：

$[A,B,\{K\}_{K_{AS2}},Na]_{K_{AS1}} = (A,B,\{K\}_{K_{AS2}},Na,f_{K_{AS1}}(A,B,\{K\}_{K_{AS2}},Na))$，其中 $f_{K_{AS1}}:\{0,1\}^* \to \{0,1\}^k$ 为带密钥的伪随机函数，是可证明选择消息攻击安全的。所以在概率多项式时间内，伪随机函数 f 的输出与一个同等长度的真随机数的输出无法相区分，因此可证明敌手通过选择消息攻击，针对 $(A,B,\{K\}_{K_{AS2}},Na)$ 能成功伪造 $f_{K_{AS1}}(A,B,\{K\}_{K_{AS2}},Na)$ 的概率为 2^{-k}，当 k 足够大时，该概率是可忽略量的。具体的证明方法与7.2.1节认证性的证明方法相同。（由于伪随机函数等价于单向函数[133]，无法从密文推导出其明文，所以对于伪随机函数不存在选择密文攻击）

考虑[*exp10*]中密文的安全性，其中密文的结构为：$\{M\}_{K_{AS2}} \overset{def}{=} (r, f'_{K_{AS2}}(r) \oplus M)$。

由于 $f'_K:\{0,1\}^k \to \{0,1\}^{\sigma(k)}$ 为带密钥的伪随机函数，是可证明选择消息攻击安全的，所以我们能证明基于 $f'_K()$ 的概率加密方案为IND-CPA安全的，证明如下。

假设当$f'_K()$为伪随机函数时，加密方案是IND-CPA不安全的。

我们构造一个随机预言机O，O按以下方式工作：O在初始化时秘密选择一个随机比特$b\in\{0,1\}$，若b=0，则O每次被访问时所输出的结果$g\in\{0,1\}^{\sigma(k)}$均为真随机数；若b=1，则O在每次被访问时所属出的g通过以下方式构造：通过伪随机数产生器生成一个r与一个K,长度均为k,然后产生伪随机函数值$g=f'_K(r)$并输出。

接着我们构造一个概率多项式时间的区分器T，它按以下方式工作：首先初始化随机预言机O并访问O获取g，然后与敌手进行IND-CPA的游戏：敌手一开始选取一些明文提交O进行解密练习，该过程可以是k的多项式次，当敌手对该练习过程满意后，向T提交两条明文$m_0,m_1\in\{0,1\}^{\sigma(k)}$，$T$选取一个随机比特$bt\in\{0,1\}$,若$bt$=1，则$T$计算$c^*=g\oplus m_0$返回给敌手，否则计算并返回$c^*=g\oplus m_1$。然后敌手通过$c^*,m_0,m_1$来猜测$bt$的值。

下面我们分析敌手猜测bt的概率：

当O的随机比特b=0时，O每次输出的g均为真随机数，那么在T的计算：$c^*=g\oplus m_{bt}$中，g肯定会将明文m_{bt}随机化，所以在敌手进行IND-CPA游戏时，敌手能正确猜测出bt的概率为1/2+关于k的一个可忽略量。

当O的随机比特b=1时，O每次输出的g均为伪随机函数，由于我们假设在伪随机函数的情况下，加密方案是IND-CPA不安全的，那么敌手在进行IND-CPA游戏时，能以1/2+关于k的一个不可忽略量成功。

这样，我们可以看出根据敌手进行IND-CPA游戏的结果，T能在关于k的多项式的时间内区分出真随机函数产生器与伪随机函数产生器，这与我们认为现实中不存在这样的区分器相矛盾。

因此，可得出结论：当$f'_K()$为伪随机函数时，该概率加密方案是可证明IND-CPA安全的，敌手通过密文得到明文的概率与明文的长度有关，为$2^{-\sigma(k)}$。

注：在仅考虑协议本身的安全强度时（就是说，忽略了敌手对密钥的物理攻击，且协议中的主体作为一种预言机，仅按照协议消息的格式传递信息），通过观察我们发现在协议中主体不会给敌手提供任何形式的解密服务，因此可认为协议中的加密消息不会受到选择密文攻击，换句话说，在仅考虑协议本身强度的情况下，所采用的概率加密算法可保证加密消息的保密性。

在[*exp12*] 中$Na\in\{0,1\}^k$是A在每次协议运行时所产生的一次性随机数，在

实现时，我们可用伪随机数发生器来产生 Na，由于目前公认不存在一个多项式时间的区分器可以将真随机数与伪随机数区分开来，我们可认为主体在两次协议中产生相同 Na 的概率为 2^{-k}。则敌手可以发起重放攻击的概率为 2^{-k}。

在[*exp16*]中，若敌手能通过 Na 来伪造 $[Na]_K$，则可冒充 B 来欺骗 A，但是由于在实际中产生 $[Na]_K$ 的伪随机函数与真随机函数是多项式不可区分的，所以敌手可伪造 $[Na]_K$ 的概率为 2^{-k}，具体证明方法与7.2.1节认证性的证明方法相同。

由此可以看出，敌手通过密码分析攻击协议加密消息的成功概率是 $2^{-\sigma(k)}$ 数量级的，攻击确认值成功的概率是 2^{-k} 数量级的，在安全参数 k 足够大时，这两个概率是可忽略不计的。

根据以上我们对协议的完全性与正确性的分析，根据定义8.2我们可以判断，改进后的 Yahalom 协议是一个安全的安全协议。

8.5、本章小结

本文的研究成果第一次将符号理论和计算理论有机地相结合，从逻辑可靠性与计算可靠性两方面对安全协议进行了分析，摆脱了对安全协议分析的片面性：1）与较为流行的符号理论下的分析方法如 Strand 空间、CSP 法相比，调和方法纠正了认为密码体制的安全性是“all-or-nothing”的看法，弥补了它们不能考察敌手密码分析能力的缺陷。2）与 BAN 类逻辑相比，调和方法具有详细的计算模型和语义模型，保证了推理规则的正确性，并使得分析者容易理解调和方法的适用范围，掌握推理规则的使用；增加了保密性规则，使得主体可推理消息的保密性，从而避免了过分信赖可信中心。3）与计算理论中的模式(modular)方法相比，调和方法可分析协议消息是否完备，发现协议在面对重放攻击、平行穿插攻击等非密码分析的攻击时是否存在漏洞。4）我们通过一个实例分析，不仅给出了 Yahalom 协议逻辑可靠性的证明，并且详细分析和证明了该协议所采用的密码体制在敌手攻击下的安全性，给具体实现该协议提供了有力的理论支持。

第九章 总 结

网络技术是一把双刃剑，它给人们提供快捷通信的同时，也给信息的安全造成了严重的威胁。随着人们对信息安全问题的日益重视，安全协议的应用也越来越广泛，因此有关安全协议的研究已经成为信息安全领域的重要课题。

本文以符号理论与计算理论为两条主线，对两种理论下的形式化分析方法作了深入的研究，不仅对原有的分析方法进行了改进，而且提出了一些有关分析与设计安全协议的新思路。本文的主要工作及创新点如下：

第一. 对安全协议形式化分析领域的发展动态作了全面而客观的评述。从目前有关安全协议形式化分析的两种截然不同的理论入手，介绍了形式化方法的发展及其分类。对于每一种理论，我们都选择具有代表性的分析方法进行了介绍，并讨论了其优缺点。此外，我们还介绍了该领域研究的新热点。

第二. 对模态逻辑与时间概念相结合的 CS 逻辑[58]进行了深入研究，针对发现的该逻辑及其改进方案的不足之处，对 CS 逻辑进行了重新改进和扩展。改进后的 CS 逻辑不仅能更准确地表达知识与信念之间的关系，而且可提高对协议分析的效率。

第三. 针对 BAN 类逻辑的不足，构造了一个新的形式化分析工具—MBL 逻辑。该逻辑与 BAN 类逻辑相比：1. 具有严格的证明机制，在给出得语义模型下可证明推理规则的正确性。2. 可分析消息的保密性，减轻了主体对于可信中心的依赖。3. 可对单向函数保护的消息进行推理，能有效地分析协议的认证性。4. 易于扩展。最后我们还实现了针对该逻辑的自动化分析工具。

第四. 研究了设计安全的认证密钥交换协议所需注意的事项，指出了设计协议时所必备的协议消息及各类消息所需的安全保护，并分别讨论了在公钥体制下与在对称钥体制下各应采用何种安全级别的加密算法。然后提出两个攻击游戏，根据主体在游戏中的成功概率给出了协议认证性的概念。

第五. 结合符号理论中的模态逻辑法与计算理论下的矛盾归约法，设计了一种分析安全协议的调和分析法，该方法为安全协议的形式化分析提供了一条新的思路。它不仅能分析协议的逻辑可靠性，而且能分析协议的计算可靠性。最后我们给出了采用该方法分析协议的一个实例。

针对本文提出的理论，我们认为在以下方面还需要进一步研究：建立一种设计安全协议的逻辑方法，然后利用 prolog 语言建立该逻辑方法的自动化方案，最后通过自动化设计方案由协议目标来定位协议的缺陷；研究在各种攻击模型下对称钥体制的安全性，以便提出更好的用于安全协议的对称钥加密方案，以及更全面地分析加密方案的安全性。

参考文献

[1] Bruce Schneier. Applied cryptography: protocols, algorithms, and source code. New York : C. John Wiley & Sons, 1996.

[2] R.Canetti and H.Krawczyk.Analysis of key-exchange protocols and their use for building secure channels.In:Eurocrypt'01,volume 2045 of Lecture Notes in Computer Science,Springer-Verlag,May 2001,453~474.

[3] 冯登国.国内外安全协议研究现状及发展趋势.见:信息安全国家重点实验室编.安全协议研讨会文集.北京: 信息安全国家重点实验室,2004,1~9.

[4] W.Diffle and M.E.Hellman.Multiuser Cryptographic Techniques,AFIPS Conference Proceedings,1976,45:109~112

[5] W.Diffle and M.E.Hellman.New Directions inCryptography.IEEE Transactions on Information Theory,1976,22:644~654.

[6] R.Blom.An Optimal Class of Symmetric Key Generation Schemes.Advances in Cryptology-Eurocrypt'84,Springer-Verlag,1985,335~338.

[7] T.Matsumoto,Y.Takashima,and H.Imai.On seeking Smart Public-key Distribution Systems. Transactions of IECE,Japan,1986,69:99~106.

[8] M.Girault.Self-certified Public Key.Advances in Cryptology Crypto' 92, Springer-Verlag, 1991,490~497.

[9] U.Feige,A.Fiat,and A.Shamir.Zero Knowledge Proofs of Identify.In Proceedings of STOC,1987,210~219.

[10] C.P.Schnorr.Efficient Signature Generation for Smart Cards.Journal of Cryptology,Vol.4, No3,1991,161~174.

[11] T.Okamoto.Provably Secure and Practical Identification Schemes and Corresponding Signature Scheme.Advances in Cryptology-Crypto'92,Springer- Verlag,1993,31~33.

[12] Lawrence C. Paulson. Relations between secrets: Two formal analyses of the yahalom protocol. J. Computer Security, 2001.

[13] R.M.Needham and M.D.Schroeder.Using encryption for authentication of large networks of computers.Communications of the ACM ,1978 ,21(12):- 993~999.

[14] Yahalom R, Klein B, Beth T. Trust relationships in secure systems: A distributed authentication perspective. In: Proceedings of the 1993 IEEE Symposi- um on Security and Priv-

acy. Los Alamitos: IEEE Computer Society Press, 19- 93,150~164.

[15] 冯登国,裴定一.密码学导引.科学出版社,1999.

[16] C.M.Campbell. Design and Specification of Cryptographic Capabilities.IEEE Comm.Soc. Mag,1978,16(6):15~19.

[17] W.Diffle and M.E.Hellman.Privacy an Authentication.An Introuction to Cryp- tography.In Proceedings IEEE,Mar 1979,67(3):397~427.

[18] 万哲先.代数和编码.北京:科学出版社,1985.

[19] NBS.Data Encryption Standard.FIPS PUB 46.National Bureau of Standards, Washingto-n.D.C,1977.

[20] H.Feistel.Cryptography and Computer Privacy.Scientific American,1973,288- (5):15~23.

[21] A.Sorkin and Lucifer.A Cryptographic Algorithm.Cryptologia,1984,8:22~41.

[22] Katzan and H.Jr.The Standard data encryption Algorithm.Petrocell:Books Inc, 1977.

[23] E.Biham.New Types of Cryptanalytic Attacks using related keys.Advances in Cryptology-Eurocrypt'93,Springer-Verlag,1994,398~409.

[24] W.E.madryga.Ahigh PerformanceEncryption Algorithm.Computer Security: A Global Challenge.North Holland:Elsevier Science Publishers,1984,557~570.

[25] H.Gustafson,E.Dawsonm,and B.Caelli.Comparison of Block Ciphers.Advances in Crypto-logy-Auscrypt'90,Springer-Verlag,1990,208~220.

[26] R.Scott.Wide Open Encryption Design Offers Flexible Implementations.Cryptologia,19-85,9(1):75~90.

[27] A.Shimizu,S.Miyaguchi.Fastdata Encipherment Algorighm FEAL.Advances in Cryptolo-gy-Eurocrypt'87,Springer-Verlag,1988,267~280.

[28] R.L.Rivest,A.Shamir,and L.Adleman.A Method for Obtaining Digital Signatures and Pub-lic-key Cryptosystem.Comm.ACM,1978,21(2):120~126.

[29] L.ElGamal.A Public Key cryptosystem and a Signature Scheme Base on Discrete Logarit-hm.IEEE Transactions on Information Theory,1985,31:469~472.

[30] W.Diffie.The first Ten Years of Public-key Cryptography.In Contemporary Cryptology,th-e Science of Information Integrity.IEEE Press,1992,135~175.

[31] M.O.Rabin.Digitalized Signatures and Public-key Functions as Intractable as Factorization. MIT/LCS/TR-212,MIT Laboratory for Computer Science,1979.

[32] J.Pieprzyk and B.Sadeghiyan.Design of Hashing algorithms,Springer-Verlag, 1993.

[33] A.J.Menezed,P.C.V.Oorschot and S.A.Vanstone.Handbook of Applied Cryptography,IEE-E Press,1996.

[34] D.Knuth. The Art of Computer Programming. Massachusetts: Addison-Wesley,1981.

[35] S.Heber and W.S.Stornetta.How to Timestamp a Digital Document.Journal of Cryptology. 1991(3):99~111.

[36] D.Bayer,S.Haber and W.S.Stornetta.Improving the Efficiency and Reliability of Digital Time-stamping.In Sequences II,Methods in Communication,security, and Computer Science,Springer-Verlag,1993,309~334.

[37] Paul Syverson.A taxonomy of relay attacks.In Proceedings of 7th IEEE Computer Security Foundations Workshop,97-1001.New York:ACM Press,1994,131 ~136.

[38] G.Low.An attack on the Needham-Schroeder public-key authentication protocol.Information processing Letters,1995,56(3):131~133.

[39] B.C.Neuman and S.G.Stubblebine.A note on the use of timestamps as nonces. ACM Operating System Review,1993,27(2):10~14.

[40] U.Carlsen.Cryptographic protocol flaws:know your enermy.In proceedings of The Computer Security Foundations Workshop VII.IEEE Computer Society Press,1994,192~200.

[41] P.Syverson.On key distribution protocols for repeated authentication.ACM Operating Systems Review,1993,27(4):24~30.

[42] M.Abadi and R.Needham.Prudent engineering pratice for cryptographic protocols.Technical Report DEC SRC Technical Report 125,Digital Equipment Corporation,Novement 1995.

[43] T.Y.C.Woo and S.S.Lam.A lesson on authentication protocol design.Operati- ng System Review,1994,28(3):24~37.

[44] T.Y.C.Woo and S.S.Lam.Authentication for distributed systems.Computer,1992,25(1):39 ~52.

[45] ISO/IEC.Information Technology-Security Techniques-Entity Authentication Mechanisms-Part 2:Entity authentication using symmetric techniques.International Organization for Standardization and International Electro-technical Commission,1992.ISO/IEC JTC 1/SC 27 N489 CD 9798-2,1992-06-09.

[46] ISO/IEC.Information Technology-Security Techniques-Entity Authentication Mechanisms -Part 2:Entity authentication using symmetric techniques.International Organization for Standardization and International Electro-technical Commission,1993. ISO/IEC JTC 1/SC 27 N739 DIS 9798-2, 1993-08-13.

[47] W. B. Mao. Modern Cryptography: Theory and Practice. Beijing: Publishing House of Electronics Industry, 2004(in Chinese).

[48] R.A.DeMillo,G.L.Davida,D.P.Dobkin,M.A.Harrison,and R.J.Lipton.Applied Cryptology Cryptography Protocols,and Computer Security Models,volume 29.Proceedings of Symposia in Applied Mathematics.Providence:American Mathematical Society,1983.

[49] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. Digital Systems Research Center, Research Report: 39, 1989.

[50] D. Dolev, A. C. Yao. On the security of public key protocols. IEEE Transactions on Information Theory, 1983, 29(2): 198~208.

[51] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In: Proceedings of the 1990 IEEE Computer Society Symposi- um on Research in Security and Privacy. Los Alamitos: IEEE Computer socie- ty Press, 1990, 234~248.

[52] M.Abadi,M.R.Tuttle.A semantics for a logic of authentication.In:proceedings of the 10th ACM Symposium on Principles of Distributed Computering.ACM Press,1991,201~216.

[53] P.C.V.Oorschot. Extending cryptographic logics of belief to key agreement p- rotocols. In: Proceedings of the 1st ACM Conference on Computer and Com- munications Security. ACM Press, 1993, 233~243.

[54] P. F Syverson,P.C.V.Oorschot. On unifying some cryptographic protocol logics. In: Proceedings of the 1994 IEEE Computer Society Symposium On Rese- arch insecurity and Privacy. Los Alamitos: IEEE Computer Society Press, 199- 4,14~28.

[55] P. F Syverson,P.C.V.Oorschot.A unified cryptographic protocol logic.Manus- cript,1996.

[56] F.J.T.Fábrega, J.C.Herzog, and J.D.Guttman. Strand spaces: Why is a security protocol correct?In: Proc of the 18th IEEE Symposium on Security and Priva- cy. Los Alamitos: IEEE Computer Society Press, 1998, 160~171.

[57] S.A.Schneider. Using CSP for protocol analysis: the Needham-Schroeder public-key protocol. University of London, Technical Report: CSD-TR-96-14, 1- 996.

[58] T.Coffey and P.Saidha.Logic for verifying public-key cryptographic protocols.IEEE Proc. Computers and Digital Techniques,Jan 1997,144(1):28-32.

[59] Mart´ýn Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. Information and Computation, 148(1):1–70, January 19- 99. An extended version appeared as Digital Equipment Corporation Systems Research Center report No. 149, January 1998.

[60] James W. Gray, III and John McLean. Using temporal logic to specify and verify cryptographic protocols (progress report). In Proceedings of the 8th IEEE Computer Security Foundations Workshop, 1995:108~116.

[61] R. Kemmerer, C. Meadows, and J. Millen. Three system for cryptographic Protocol analysis.Journal of Cryptology, Spring 1994, 7(2):79~130.

[62] Richard A. Kemmerer. Analyzing encryption protocols using formal verification techniques. IEEE Journal on Selected Areas in Communications,May 1989, 7(4):448~457.

[63] Catherine Meadows. A system for the specification and analysis of key management protocols. In Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy, pages 182~195, 1991.

[64] John C. Mitchell, Mark Mitchell, and Ulrich Stern. Automated analysis of cr- yptographic protocols using Mur_. In Proceedings of the 1997 IEEE Symposi- um on Security and Privacy, pages 141~151, 1997.

[65] S.Gritzalis,D.Spinellis,P.Georgiadis.Security protocols over open networks and distributed systems:formal methods for their analysis,design,and verification. Computer Communications,May 1999,22(8):695~707.

[66] S.Gritzalis,D.Spinellis.Cryptographic protocols over open distributed system: a taxonomy of flaws and related protocol analysis tools.16th International Con- ference on Computer Safety, Reliability and Security. NewYork, U.K.:SAFE- COMP'97.123~137.

[67] C.Meadows.Applying formal methods to the analysis of a key management protocol .Journal of Computer Security,1992(1):5~35.

[68] Rubin,P.Honeyman.Formal methods for the analysis of authentication protocols.Technical Report,CITI,Nov 1993:93~97.

[69] C.A.Meadows.A formal verification of cryptographic protocols:a survey.Advances in Cryptology-ASIACRYPT'94 Proceedings.Springer-Verlag,1995: 133~ 150.

[70] R.Kailar and V.D.Gligor. On belief evolution in authentication protocols.Pro- ceedings of the Computer Security Fouucations workshop IV,1991:103~ 116.

[71] P.Bieber.A logic of communication in a hostile environment.Proceedings of the Computer Security Foundations workshop III,1990,14~20.

[72] P.Syverson.A logic for the analysis of cryptographic protocols.Naval research Laboratory Formal Report 9350,December 1990.

[73] CCITT.CCITT draft recommendation X.509.The directory-authentication framework,Version 7,1987.

[74] R.Kailar. Accountability in electronic commerce protocols. IEEE Trans. on Software Engineering, 1996,22(5):313-328.

[75] G.Lowe.Breaking and fixing the Needham_Schroeder public-key protocol using FDR.Pr-

oceedings of TACAS,1996,147~166.

[76] R.Kemmerer,C.Meadows,J.Millen.Three systems for cryptographic protocol analysis.Journal of Cryptology,1994,7(2):79~130.

[77] C.Meadows.Analysis of the Internet Key Exchange Protocol using the NRL protocol analyzer.Proceedings of the 1999 IEEE Symposium on security and privacy,Oakland,CA, IEEE Computer Society Press,May 1999.

[78] J.Millen,S.Clark,and S.Freedman.The Interrogator:protocol security analysis. IEEE Trans. Software Engineering,SE-13(2),Feb 1987.

[79] Will Marrero,Edmund Clarke,and Somesh Jha.A model checker for authentication protocols.In Proc.DIMACS Workshop on Design and Formal Verificati- on of Security Protocols 1997.

[80] C.A.R.Hoare.Communicating sequential processes. Englewood Cliffs NJ:Pre- ntice-Hall International,1985.

[81] A.W.Roscoe.The Theory and Practice of Concurrency. Englewood Cliffs NJ: Prentice-Hall,1997.

[82] S.Schneider.Concurrent and real-time Systems:the CSP Approach.New York: John Wiley & Sons, 1999.

[83] G.Lowe.CASPER: A complier for the analysis of security protocols. Journal of Computer Security, 1998,6:53~84.

[84] L.C.Paulson.The inductive approach to verifying cryptographic protocols.Jo- urnal of Computer Security,1998:85~128.

[85] Tobias Nipkonw,C.Paulson,Markus wenzel.A Proof Assistant for Higher-Order Logic.Springer Verlag:volume 2283 of Lecture Notes in Computer Science in the Tutorials subseries,2002.

[86] D. X. Song. Athena: A new efficient automated checker for security protocol analysis. In proceedings of the 12th IEEE Computer Security Foundations W- orkshop. IEEE Computer Society Press, 1999.

[87] Joshua D. Guttman, F. Javier Thayer Fábrega. Authentication tests. In Procee- dings of the 2000 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, 2000, 96~109.

[88] D.Dolev,C.Dwork,and M.Naor.Non-malleable cryptography.In Proceedings of 23rd Annunal ACM Symposium on Throey of Computing,1991,542~552.

[89] Mihir Bellare, Anand Desai, Eron Jokipii, and Phillip Rogaway. A concrete security

treatment of symmetric encryption: analysis of the DES modes of operation. In Proceedings of 38th Annual Symposium on Foundations of Computer Science (FOCS 97), 1997.

[90] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Advances in Cryptology—CRYPTO '94, volume 839 of Lecture Notes in Co- mputer Science,Springer-Verlag,1994,341~358.

[91] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS 82),1982,112~117.

[92] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing,1987, 218~229.

[93] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or All languages in NP have zero-knowledge proof systems. Journal of the A- CM, 1991,38(3):691~729.

[94] Shafi Goldwasser, Silvio Micali, and Ronald Rivest. A digital signature sch- eme secure against adaptive chosen-message attack. SIAM Journal on Comp- uting,1988,17: 281~308.

[95] A.C.Yao. Theory and applications of trapdoor functions. In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS 82),19- 82,80~91.

[96] S.Goldwasser and S.Micali.Probabilistic encryption.journal of Computer and System Sciences,1984,28:270~299.

[97] M.Bellare and P.Rohaway.Entity authentication and key distribution.In D.Stinson,editor, Advances in Cryptology-proceedings of CRYPTO'93,Lecture Notes in Computer Science 773.Springer-Verlag,1994:232~249.

[98] M.Bellare,R.Canetti,H.Krawczyk.A modular approach to the design and analysis of authentication and key exchange protocols.In:Proceedings of 30th Annual Symposium on the Theory of Computing,ACM,1998: 419~428.

[99] M. Abadi, P. Rogaway. Reconciling two views of cryptography.Journal of Cryptology, 2-002,15(2): 103~127.

[100] D.Micciancio,B.Warinschi.Completeness theorems for the Abadi-Rogaway logic of encrypted expressions. Journal of Computer Security,2004,12(1): 99~ 129.

[101] V. Gligor, D. O. Horvitz. Weak Key Authenticity and the Computational Co- mpleteness of Formal Encryption. The 23th Annual International Cryptology Conference, Santa Barbara,2003.

[102] D. Micciancio, B. Warinschi. Soundness of Formal Encryption in the Presenc- e of Active Adversaries. The 1st Theory of Cryptography Conference(TCC), LNCS2951, Massachusetes Institute of Technology, 2004.

[103] A.Galton.Logic for information technology.Wiley,1990.

[104] S.Brackin.Automatic formal analysis of cryptographic protocols.Proceedings of the 19th National Conference on Information Systems Security,1996.

[105] 范红,冯登国.一种分析 Timed-Release 公钥协议的扩展逻辑.计算机学报,2003,7:832~838.

[106] Ronald L.Rivest Adi Shamir,and David A.Wagner:Time-lock puzzles and ti- med-release cryptographic protocol.Mit Labortary for Computer Science,1996, 545~554.

[107] J.Millen. CAPSL: Common Authentication Protocol Specification Language, Technical Report MP 97B48, The MITRE Corporation, 1997.

[108] G.Denker and J.Millen. CAPSL Integrated Protocol Environment.In:Proceedi- ngs of the DARPA Information Survivability Conference and Expo 2000,1:207~221.

[109] E.A.Campbell,R.Safavi-Naini.On automating the BAN logic of authenticati on.In Proceedings of 15th Australian Computer Science conference 1992.

[110] R.C.Hauser,E.S.Lee.Verification and modeling of authentication protocols.In Computer Security—ESORICS'92,1992.

[111] A.Mathuria,R.Safavi-Naini,P.Nickolas.Exploring minimal BAN logic proofs of authentication protocols.In Proceedings of 10th International Conference on Information Security, May,1994.

[112] A.Mathuria,R.Safavi-Naini,P.Nickolas.On the automation of GNY logic.In P- roceedings of 18th Australasian Computer Science Conference,February,1995.

[113] S.Brackin.Automatic formal analysis of cryptographic protocols.Proceedings of the 19th National Conference on Information Systems Security,1996.

[114] S.Brackin.An interface specification language for automatically analyzing cr- yptographic protocols.Proceedings of the 1997 Symposium on Network and Distributed System Security,1997:40~51.

[115] S.Brackin.Using checkable types in automatic protocol analysis.Proceedings of the 15th Annual Computer Security Applications Conference,1999.

[116] CHEN huiping,ZHAO yaohua,QIAN xu.Tutorial of artificial intelligence. Beijing:Publishing House of Electronics Industry,2001.

[117] M.Blum,S.Goldwasser.An efficient probabilistic public-key encryption sche- me which hi-

des all partial information.In G.R.Blakley and D.Chaum,editors, Advances in Cryptology-Proceedings of CRYPTO'84,Lecture Notes in Computer Science 196,Springer-Verlag,19-85,289~299.

[118] M.Bellare and P.Rogaway.Optimal asymmetric encryption.In A.deantis,editor, Advances in cryptology-Proceedings of EUROCRYPT'94,Lecture Notes in Computer Science 95-0,Springer-Verlag,1995,92~111.

[119] R.Cramer and V.Shoup.A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack.In H.Krawczyk.editor,Advances in Cryptology-Proceedings of CRYPTO'98,Lecture Notes in Computer Science 1462,Springer-Verlag,1998,13~25.

[120] Luby M, Rackoff C. How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal on Computing, 1988, 17(2): 373-386.

[121] O.Goldreich,S.Goldwasser and S.Micall.How to construct random functions.Journal of the ACM.1986,33(4):210~217.

[122] 卿思汉.安全协议.北京:清华大学出版社,2005.

[123] W.Mao and C.Boyd.Methodical use of cryptographic transformations in authentication protocols.IEE Proceedings,Comput.Digit.Tech,July 1995,142(4): 272~278.

感 谢

三年的博士生学习生活很快就要结束了，当本文付梓之时，回首三年的学习生活，尤其是写论文的“痛并期望着”的四个月，有着诉不尽的感慨，道不完的感谢。

首先感谢尊敬的导师李大兴教授对我的辛勤培养和悉心指导。李老师严谨的治学态度，渊博的知识，一丝不苟的治学态度以及深邃的思想使我受益匪浅。他不仅在学校中给我提供了宽松的治学环境，并且在得安计算机技术有限公司给我营造了良好的实习环境，使我得以在实践中加深了对理论知识的理解。三年来，我取得的每一点进步无不凝聚着李老师的心血。李老师的勇于开拓、积极进取的实干精神也将激励我勇敢的面对今后研究工作中的每一个困难。

感谢马绍汉老师对论文的指导及提出的宝贵意见。马老师思维敏捷、学识渊博、治学严谨且平易近人，精辟的学术观点常常使我感觉醍醐灌顶、如浴春风，一丝不苟的学术态度时时令我领悟到追求真理的真谛。

感谢论文评审组的专家们对论文的审阅及对论文提出的宝贵意见！

感谢秦静老师在多年的学习和生活中给我的指导与帮助，她严谨的学术态度，渊博的专业知识是我受益匪浅。

感谢刘红英老师在学习与生活上给予我的热情帮助和支持。她周到细致的工作让我在研究所里感到了家一样的温暖。

感谢三年来朝夕相处的同学和师弟师妹。他们对我的真心帮助和关怀让我感受到了手足情深。感谢于佳，王洪涛，李国文，孔繁玉，蔡准，李如鹏几位博士在学习生活中给予我的支持；感谢研究所的所有师弟师妹，如王倩、林宇翰、迟小军等，与他们的融洽相处给我留下了许多美好的回忆。

感谢我的妻子刘月及岳父岳母在生活与精神上给我的帮助与支持，特别是我的妻子，是她在我最困难的时候守在我身边，给了我往前走的勇气。

我要深深地感谢含辛茹苦抚养我成人的父母，我所取得的一切都源于他们对我深深的爱。

最后，我向所有给予我真诚关心和帮助的老师、同学和亲友致以诚挚的谢意！

博士学习期间发表的论文情况

1、赵华伟,李大兴,秦静.一种时间相关的分析安全协议的扩展逻辑.计算机应用,2005,25(10):2272~2275

2、赵华伟,李大兴.单向函数在公钥认证协议中的作用.计算机应用,2005,25(11):2509~2511

3、赵华伟,李大兴.加入时间因素的 Needham-Schorder 协议及其形式化分析.计算机工程与应用,2005,41(36):108~110

4、赵华伟,李大兴.一种调和两种观点的安全协议分析法.计算机研究与发展,2006,43(7):1121~1127

5、赵华伟,李大兴. 密钥交换协议的安全性分析.山大学报,2006,41(4),已录用

已投稿的论文:

6、赵华伟,李大兴.一种可分析保密性与认证性的模态逻辑.计算机学报.

在读期间参与科研项目情况

973 项目：网络安全认证协议研究（G1999035802）

863 项目：PKI/PMI 高速加密和验证设备研制(2003AA141120)