文章编号:1001 - 9081(2006)05 - 1087 - 03

一种新的密码协议分析方法及其应用

文静华[1,2],张 梅[1],李 祥[2]
(1. 贵州财经学院 信息学院,贵州 贵阳 550004;
2. 贵州大学 计算机软件与理论研究所,贵州 贵阳 550025)
(jinghuawen@sohu.com)

摘 要:针对传统时序逻辑把协议看成封闭系统进行分析的缺点,提出一种新的基于策略的 ATL(Alternating-time Temporal Logic)逻辑方法分析密码协议。最后用新方法对 Needham-Schroeder 协议进行了严格的形式化分析,结果验证了该协议存在重放攻击。工作表明基于博弈的 ATL 逻辑比传统的 CTL 更适合于描述和分析密码协议。

关键词:密码协议;安全性;形式化分析;ATL
中图分类号:TP309.7 **文献标识码**:A

New approach for formal analyzing encryption protocols

WEN Jing-hua[1,2], ZHANG Mei[1], LI Xiang[2]
(1. *Information Department, GuiZhou Financial Institute, Guiyang Guizhou* 550004, *China;*
2. *Institute of Computer Science, Guizhou University, Guiyang Guizhou* 550025, *China*)

Abstract: Aiming at the shortcoming that traditional temporal logic regards protocols as close system to analyse, this paper proposes a ATL(Alternating-time Temporal Logic) logical method based on game to analyse cryptographic protocols. In the end, we make strict formal analysis for needham-schroeder protocol with this new method, as a result we validate there exists reply attacks. These works indicate that the ATL logic based on game is more suitable to describe and analyze cryptographic protocols than traditional CTL.

Key words: cryptographic protocols; security; formal analysis; ATL

0 引言

密码协议主要目的是利用密码技术实现密钥交换、身份认证和安全支付,在安全领域中协议同密码算法同样重要。若密码算法十分先进,但密码协议的设计有缺陷,该密码体系仍存在严重的安全隐患。当前关于安全问题的研究热点之一是涉及密钥分布、身份验证、数字签名和公证的密码协议的建模与形式化分析。

基于时序逻辑[1]的密码协议形式化分析方法已成为一个研究热点。其中,文献[3]提出的线性时序逻辑 LTL(Linear Temporal Logic)方法具有很强的描述能力,可以对协议系统进行形式化建模分析并开发了相应的检测工具 SPIN 进行自动验证,得到了广泛应用;文献[2]在 LTL 基础上提出了计算树逻辑 CTL 的方法能够对协议的并发性质进行更为准确的描述,同时也开发了自动模型检测工具 SMV 与 NuSMV。这些传统时序逻辑方法由于把协议看成封闭式并发系统进行研究,不能有效描述协议与外部环境(入侵、网络故障等)的联系,同时要求协议主体严格遵循既定步骤,不太适合日益复杂的密码协议的描述与分析。基于博弈的 ATL[3,4] 逻辑可以有效地解决上述问题,该逻辑能够对协议主体间的合作与竞争关系、协议内部与外部环境的关系等进行有效描述,系统各个主体可以选择相应策略行为。本文对 ATL 逻辑方法及其在密码协议形式化分析中的应用进行了研究。最后,我们利用本方法对 Needham-Schroeder 协议[5]进行了严格的形式化分析,结果验证了本文方法的有效性和方便性。

1 基于博弈的逻辑模型

本节我们将讨论一种新的用于分析密码协议的形式化模型。我们用交替转换系统 ATS[3,4](Alternating Transition Systems)描述密码协议,用时间交替时序逻辑 ATL 描述密码协议的性质并用模型检测工具 MOCHA 进行验证。交替转换系统 ATS 与时间交替时序逻辑 ATL 是 Alur 等人提出的适合于描述开放分布式系统的逻辑描述工具,其相应模型检测工具为 MOCHA,目前已在计算机理论界得到了一定应用[6]。

1.1 交替转换系统 ATS

ATS 是我们用来对交换协议建模的形式化工具,是普通 Kripke 结构的一个带博弈变量的扩展。其定义如下:

定义 1 一个交替转换系统 ATS 是一个六元组:

$S = < \Pi, \Sigma, Q, Q_0, \pi, \delta >$

其中:

Π 是命题集。

Σ 是参与者集。

Q 是状态集。

$Q_0 \subset Q$ 是初始状态集。

收稿日期:2005 - 11 - 14;**修订日期**:2006 - 01 - 27
基金项目:贵州省自然科学基金资助项目(20042111);贵州省教育厅自然科学基金资助项目(2004219)
作者简介:文静华(1975 -),男,贵州沿河人,博士研究生,主要研究方向:信息安全、协议分析; 张梅(1974 -),女,贵州沿河人,讲师,硕士,主要研究方向:信息安全、GIS; 李祥(1942 -),男,贵州安顺人,教授,博士生导师,主要研究方向:可计算性理论、密码学与网络安全.

$\pi: Q_0 \to 2^{\Pi}$ 是从状态到命题集的映射。

$\delta: Q_0 \times \Sigma \to 2^{2^Q} \setminus \{\theta\}$ 是一个从{状态 × 参与者}到非空的选择集合的转换函数,这里的每个选择是一个可能的下一个状态集合(可能包含一些约束)。当系统在状态 q 时,每个参与者选择一个集合 $Q_a \in \delta(q,a)$,这样,一个参与者 a 保证系统的下一个状态包含在它的选择 Q_a 中,具体选择其中的哪一个状态还要看系统中其他参与者的选择,因为 q 的后继存在于所有参与者选择的交集 $\cap_{a \in \Sigma} Q_a$ 里面。必须保证转换函数是无阻塞的而且所有参与者选择唯一的下一个状态,即:如果 $\Sigma = \{a_1, \cdots, a_n\}$,那么对每个状态 $q \in Q$ 和集合 $Q_1, \cdots, Q_n$,$Q_1 \cap \cdots \cap Q_n$ 是唯一的。若 $q_0 \in Q_0$ 是一个初始状态,由状态构成的无限序列 $q_0, q_1, \cdots, q_n$ 是一个计算。

1.2 时间交替时序逻辑 ATL

ATL 是与交替转换系统 ATS 对应的逻辑系统,下面给出 ATL 公式定义。

定义 2 一个 ATL 公式有如下形式:

1) p,其中,命题 $p \in \Pi$;

2) $\neg \varphi$ 或 $\varphi_1 \vee \varphi_2$,其中,φ_1 和 φ_2 是 ATL 公式;

3) $<<A>>\circ \varphi$, $<<A>>\Box \varphi$,及 $<<A>>\varphi_1 \cup \varphi_2$ 其中,$A \in \Sigma$ 是参与者集合,φ、φ_1 和 φ_2 是 ATL 公式。

$<<>>$ 是路径量词,$\circ$(下一个),$\Diamond$(可能),$\Box$(必然),$\cup$(直到)是时态算子,其定义见文献[1],其他 $\neg$,$\wedge$,$\vee$ 等与普通逻辑学中含义相同。

定义 3 策略,一个参与者的策略是一个映射:

$f_a: Q^+ \to 2^Q$,使得对所有 $\lambda \in Q^*$ 和所有 $q \in Q$,$f_a(\lambda \cdot q) \in \delta(q,a)$ 成立。

关于 ATL 和 ATS 的语法及语义详见文献[1]。

2 协议建模与分析

密码协议既要求满足保密性,同时要有足够的安全性,能够抵御重放等攻击。要用 ATL 逻辑进行密码协议分析,首先必须对协议系统进行建模,为了简化建模过程,我们不用直接建立系统的 ATS 模型,而是采用 Dijkstra 类型保护命令语言[7](guarded command language)方法进行建模,每个参与者 a 对应一个形如 $guard \to update$ 的保护命令集。一个计算步骤定义为:每个参与者选择它自己的命令集中 $guard$ 取值为真的一个命令,所有参与者选择的命令中 $update$ 部分相交得到的结果就是下一个状态。用保护命令语言建立系统的 ATS 模型,用 ATL 公式描述待验证的系统性质并输入到模型检测工具 MOCHA[2] 中运行,即可根据输出结果分析系统性质。另外我们必须增加一个入侵者 i。

2.1 协议基本假设

考虑到密码协议的一般情况,同时为了简化特定协议分析过程,我们对一些比较有共性的内容作一个基本假设,特定协议另有说明的除外。

通道:我们假定协议参与各方之间的通道是不可靠的,即其传输的信息可能延迟、丢失;而协议参与各方与可信第三方之间的通道是可恢复的,即其传输的信息可能延迟,但最终会在有限时间内到达目的地。当然,如果通道在被攻击的情况下,传输的信息有可能永远不会在有限时间内到达目的地,对于这样的情况,我们将在今后的研究中予以考虑。

协议主体:协议参与各方都可能是不诚实的,而可信第三方是诚实可靠的。

2.2 入侵者形式化建模

我们假定入侵者对信道有完全的控制能力,入侵者能够偷听、拦截、存储、插入、删除、生成、转发、重放消息。我们具体对入侵者的知识与能力作出如下假设:

a) 知道参与协议运行的各主体名及其公钥,并拥有自己的加密密钥和解密密钥;

b) 可窃听或中途拦截系统中传送的任何消息,增加自己的知识或在系统中可插入新的消息,并可运用他知道的所有知识;

c) 即使不知道加密部分的内容,也可重放他所看到的任何消息,同时可以改变明文部分内容。

入侵模型如图 1 所示。在协议运行中,入侵者将不依照协议的要求,而是根据自己拥有知识的情况,向所有的主体进行截获、生成、插入消息等。

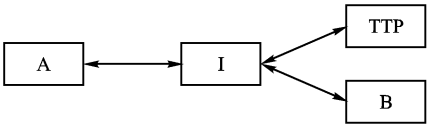

图 1 系统入侵模型

3 应用实例

3.1 Needham-Schroeder 协议简介

Needham-Schroeder 协议是 R. M. Needham 和 M. D. Schroeder 等人提出的应用于认证和密钥分配的协议,下面我们对其进行简要描述。

(1) $A \to S: A, B, N_a \ (m_1)$

(2) $S \to A: \{N_a, B, K_{ab}, \{K_{ab}, A\} K_{bs}\} K_{as} \ (m_2)$

(3) $A \to B: \{K_{ab}, A\} K_{bs} \ (m_3)$

(4) $B \to A: \{N_b\}_{K_{ab}} \ (m_4)$

(5) $A \to B: \{N_b - 1\} K_{K_{ab}} \ (m_5)$

协议共有 3 个参与者:协议发起人 A、协议响应者 B,及可信的第三方服务器 KDC。N_a,N_b 是 A 和 B 发布的具有新鲜性的随机数。协议目的是借助于服务器,主体 A 和主体 B 建立他们之间进行秘密通信的会话密钥。

协议的运行过程如下:

a) 保密密钥 K_{as} 和 K_{bs} 分别是 A 和 KDC、B 和 KDC 之间共享的密钥,本协议的目的就是要安全地分发一个会话密钥 K_{ab} 给 A 和 B;

b) A 在第 2 步安全地得到了一个新的会话密钥;第 3 步只能由 B 解密,并理解第 4 步表明 B 已知道 K_{ab} 了;第 5 步表明 B 相信 A 知道 K_{ab} 并且消息不是伪造的;

c) 第 4、5 步目的是为了防止某种类型的重放攻击,特别是如果敌方能够在第 3 步捕获该消息并重放之,这将在某种程度上干扰破坏 B 方的运行操作。

3.2 Needham-Schroeder 协议的建模

对协议建模的关键在刻画出协议各主体的基本行为转换关系,主要包括协议发起者、协议响应者和可信的第三方服务器。同时我们要对不诚实的协议参与者、可能存在的入侵者的行为进行描述,还要对通信信道进行合理的假设。

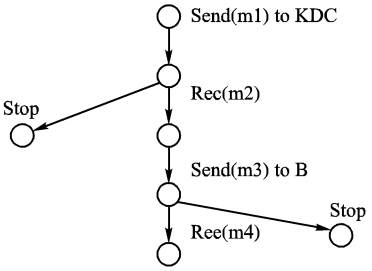

图 2　协议发起者 A 的基本行为

图 2 是协议参与者 A 在正常状态下的行为转换图,若 A 是不诚实的,它可以尝试与图 2 不同的行为以获取利益。如不按协议规定启动 Abort、Resolve 子协议,窃听 B 与 KDC 之间的通信内容等。B 同样可以尝试类似的行为以获取利益。

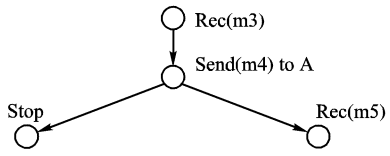

图 3　协议响应者 B 的基本行为

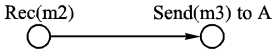

图 4　可信第三方服务器 KDC 的行为

除了对 Needham-Schroeder 协议的上述主要模块建模外,我们还要对不诚实的协议主体(参与者)、通道行为(按基本假设)、入侵者的各种可能的入侵行为等进行建模,然后将这些模型转化成 MOCHA 可以接受的保护命令语言[11](guarded command language)输入 MOCHA 系统进行验证,根据 ATL 公式是否成立来判断系统是否满足某个性质。与 SMV 等模型检测工具不同,MO CHA 系统在 ATL 公式不成立时并不输出对应反例,因而还需要对协议进行详细分析以找出其满足该性质的具体原因。

3.3　Needham-Schroeder 协议性质的 ATL 描述与分析

根据前面对 Needham-Schroeder 协议的描述,将用 ATL 公式分别对 Needham-Schroeder 协议的保密性、安全性等性质进行描述,并利用 MOCHA 工具进行验证。

3.3.1　保密性

对本协议来说,协议双方通信的内容是会话密钥 K_{ab},属于机密信息,协议要求入侵者 i 应该不能在会话结束之前获取,即如果协议双方 A 和 B 是诚实的,那么即使 i 可以控制通信信道 com,它也无法获取协议双方通信交换的关键信息 K_{ab} 及无法解密获取以此密钥加密的数据信息 m,保密性可以用以下 ATL 公式描述如下:

$$\neg \ < < i, com > > \Diamond (K_{ab} \vee m)$$

通过利用 MOCHA 工具对上述公式进行验证可知,Needham-Schroeder 协议满足保密性。

3.3.1　安全性

主要检验入侵者 i 能否通过重放等方法欺骗 A 或 B 使其在不知情的状况下以为是与合法协议主体(B 或 A)完成协议。经检验,下述公式成立:

$$\neg \ < < i, com, B_h > > \Diamond (stop \wedge < < i > > \Diamond m)$$

分析可知在协议中入侵者 i 可以通过偷听等手段获取以前某次协议的信息实施重放攻击而欺骗 B:

1)假定攻击方 C 已经掌握 A 和 B 之间通信的一个老的会话密钥;C 可以在第 3 步冒充 A 利用老的会话密钥欺骗 B,除非 B 记住所有以前使用的与 A 通信的会话密钥,否则 B 无法判断这是一个重放。

2)攻击如果 C 可以中途阻止第 4 步的握手信息,则可以冒充 A 在第 5 步响应从这一点起 C 就可以向 B 发送伪造的消息,而对 B 来说认为是用认证的会话密钥与 A 进行的正常通信。

上述攻击中,若 B 不保留以往通信副本,则入侵者 i 可以成功实现重放攻击,使得诚实的 B 以为是 A 发起的一个新的通信过程(如新合同等)而接受该会话密钥与入侵者 i 进行通信。由于入侵者 i 只能使用 A、B 以前成功进行的一个协议副本才能进行攻击,对整个协议的安全性构成威胁。我们可以通过加盖时间戳的方式有效地防止这种重放攻击。

4　结语

线性时序逻辑 LTL 和计算树逻辑 CTL 能够对协议的内部性质进行准确的描述,同时也具有自动模型检测工具 SPIN 和 SMV。但是传统时序逻辑方法由于把协议看成封闭式并发系统进行研究,不能有效描述协议与外部环境的联系,如:入侵、网络故障及协议主体之间的竞争与合作等,不太适合日益复杂的密码协议和大型网络协议的描述与分析[8,9]。本文提出的基于博弈的 ATL 逻辑方法,能够对协议主体之间的对抗和合作关系进行准确的描述,引入了入侵者模型以分析各种主要攻击方式,是有效的针对复杂密码协议的形式化分析方法。通过对一个典型的密钥分配协议 Needham-Schroeder 协议进行了严格的形式化分析并用 MOCHA 工具验证,发现了其存在重放攻击,结果表明了新方法的正确性和实用性。

参考文献:

[1] EMERSON EA. Temporal and modal logic[A]. VAN LEEUWEN J, ed. Handbook of Theoretical Computer Science, vol B: Formal Models and Semantics, chapter 16[C]. Elsevier Publishers B. V, 1990.995 - 1072.

[2] CLARKE EM, EMERSON EA. Design and synthesis of synchronization skeletons using branching time temporal logic[A]. Logic of Programs, volume 131 of Lecture Notes in Computer Science[C]. Springer-Verlag, 1981. 52 - 71.

[3] ALUR R, HENZINGER TA, KUPFERMAN O. Alternating - time temporal logic[A]. 38th Annual Symposium on Foundations of Computer Science[C]. IEEE Computer Society Press, 1997.100 - 109.

[4] ALUR R, HENZINGER T, MANG F, *et al*. MOCHA: modularity in model checking[A]. Proc. CAV '98[C], 1998. 512 - 525.

[5] NEEDHAM RM, SCHROEDER MD. Using Encryption for Authentication in Large Networks of Computers[J]. Communications of the ACM, 1978, 21(12): 993 - 999.

[6] SCHNEIDER SA. Formal analysis of a non - repudiation protocol [A]. 11th IEEE Computer Security Foundations Workshop[C], 1998. 54 - 65.

[7] HENZINGER T, MAJUMDAR R, MANG F, *et al*. Abstract interpretation of game properties[A]. Proc. SAS '00, 2000.220 - 239.

[8] 陈庆锋, 王驹, 白硕, 等. 电子商务安全协议的逻辑验证[J]. 软件学报, 2000, 11(3): 346 - 362.

[9] 肖德琴, 周权, 张焕国, 等. 基于时序逻辑的加密协议分析[J]. 计算机学报, 2002, 25(10): 1083 - 1091.