Z&G 协议公平性的形式化验证*)

龚德良[1]　殷建平[2]　詹宇斌[2]

(湘南学院计算机科学系　郴州 423000)[1]　(国防科技大学计算机学院　长沙 410073)[2]

摘　要　Zhou 和 Gollmann 设计的公平非否认协议(Z&G 协议)旨在为电子商务交易的双方提供非否认证据和公平性。提出一种基于状态转换的方法对其公平性进行分析。与以往方法不同,它是一种针对电子商务协议的专门分析工具:用状态转换系统为电子商务系统建模;用基于状态的形式系统描述协议,将协议的信任假设显式表示为协议的一部分。该方法按进程是否背离协议或者背离协议的程度将协议在系统中的执行序列定义为三类模式:遵守型、欺骗型和中断型。验证了 Z&G 协议在三种模式下的执行序列都满足公平性。

关键词　公平性,状态转换系统,遵守型,欺骗型,中断型

Analysis of Fairness of the Z&G Protocol

GONG De-liang[1]　YIN Jian-ping[2]　ZHAN Yu-bin[2]

(Computer Science Department of Xiangnan College, Chenzhou 423000, China)[1]

(School of Computer, National University of Defense Technology, Changsha 410073, China)[2]

Abstract　A non-repudiation protocol designed by Zhou and Gollmann is a typical electronic commerce protocol, which provides participants with fairness service. To verify the protocol this article models the electronic commerce system as the state transition system, specifies the protocol with state-based formalism, and formalizes fairness property in linear temporary logic. Protocol executions in the system are classified into three modes: compliance, deception and abortion, which have been analyzed and proved to satisfy fairness property.

Keywords　Fairness, Transition system, Compliance, Abortion, Deception

1　引言

传统密码协议,如认证和密钥分配协议,其目标是秘密性和认证性。协议的参与者诚实且拥有共同的目标,但他们之间的通信信道可能是不安全的,因此分析这类协议时应考虑对窃听者建模,分析通道不安全时可能出现的安全问题。而对于电子商务协议,其目标是保证证据有效性、公平性等。协议的参与者可能不诚实,并且各自都有不同的利益。因而这里出现了传统密码协议较少甚至几乎不予考虑的新情况。为了更有效地解决这个新的情况,我们不妨假定通信信道是安全(秘密、可认证、完整)的,由底层密码协议提供这些安全服务。这样,电子商务协议相对于传统密码协议就重在对参与者建模,分析考虑参与者欺骗时可能出现的安全问题。

Zhou 和 Gollmann 两人在 1997 年设计了公平性非否认协议(简称 Z&G 协议),其目的是保证参与者双方的公平性和提供非否认服务。1999 年,S. Schneider 最早用通信顺序进程(CSP)理论对协议进行分析,该方法对参与者、通信信道建模,但对不同属性的分析(证据的有效性和非否认性)需建立不同的模型;其后不久,Zhou 和 Gollmann 采用信任逻辑 SVO 非常简洁地验证了协议满足证据有效性,但 SVO 逻辑本身存在缺陷,因而无法分析协议公平性。2001 年,G. Bella 和 L. C. Paulson 应用定理证明器 Isabelle 和基于归纳的方法证明了协议满足证据的有效性和公平性,该方法用窃听者来模拟可能发生欺骗行为的协议参与者,但只能描述部分欺骗者行为。以上几种方法虽然从不同的角度对协议证据有效性及公平性进行了探讨,但它们基本上是在原有分析传统密码协议理论上的一种扩展,因而也不可避免地存在着一定的局限性。

本文尝试采用一种基于状态转换系统分析电子商务协议的方法来解决上述问题,该方法是一种分析电子商务协议的专门分析工具。它将进程表示为简单的状态转换系统,电子商务系统是多个并发、异步的分布式进程组成的组合状态转换系统。电子商务协议采用基于状态的形式化系统描述,协议表示为规则集合,每条规则由状态条件和动作组成,进程的状态满足状态条件才能执行相应的动作。协议的目标用时序逻辑描述。协议的执行序列按照其背离协议规则的程度分为 3 种模式:遵守型、欺骗型和中断型,其中,用遵守型对诚实参与者建模,也考虑远程系统的失效事件;欺骗型对不诚实参与者建模;中断型对本地系统崩溃、用户中断等事件建模。本文通过分析这三类执行序列证明了协议满足公平性,给出了部分定理的详细证明。模型和形式化机制采用的状态转换系统,模型简单灵活易于转换到其他的形式化机制。

2　电子商务系统模型

一个简单的电子商务系统由三类相互联系的主体(银行、商店、顾客)组成。主体之间通过消息传递信息。我们的模型假定每个主体一次只运行一个进程,并且用进程标识符来标识主体。每个进程有局部数据;进程能产生新的数据,如生成随机密钥和产生新的随机数;进程有计时功能判断超时;进程

*)国家自然科学基金(60473057)和湖南省教育厅重点科研基金(2006A006)资助。龚德良　副教授,主要研究方向为信息安全,计算机应用技术;殷建平　教授;博士生导师,主要研究领域为信息安全,模式识别;詹宇斌　博士生,主要研究领域为电子商务协议及形式化验证。

具有密码功能(如加密和解密),能从已有的数据推导出新的数据。我们将进程失效分为两类:进程失效停和欺骗。进程的失效停是指进程异常中止,即进程不按协议预定的步骤执行;进程欺骗是指表面上进程按协议预定的步骤执行,但实际上进程传送的消息并非协议规定的格式。我们假定电子商务系统中通信是安全的,即如果进程接收了一个消息,那么该消息是保密的、完整的和可认证的。

2.1 消息建模

定义 1(消息)

(1)原子消息 $m:t$ 是消息;

(2)如果 $m_1:t$ 和 $m_2:t$ 是消息,则两个消息的连接 $m_1m_2:t_1\times t_2$ 也是消息;

(3)如果 $f:t_1\times\cdots\times t_n\to t$ 是密码函数且 $m_i:t_i(i=1,\cdots,n)$ 是消息,则函数 f 对 $m_1,\cdots,m_n$ 的应用 $f(m_1,\cdots,m_n):t$ 也是消息。

定义 2 如果 m 和 m' 是消息,$m=m'$ 当且仅当

(1)m 和 m' 都是相同的原子消息;或者

(2)$\exists m_1,m_2,m_1',m_2'$ 且 $m_i=m_i'(i=1,2)m=m_1m_2,m'=m_1'm_2'$,或者

(3)$m=f(m_1,\cdots,m_n),m'=f(m_1',\cdots,m_n')$ 并且 $m_i=m_i'(i=1,\cdots,n)$。

定义 3 m 是消息,M 是消息集合,m 由 M 可构造,即 $m\in{}^*M$ 当且仅当

(1)$m\in M$;或者

(2)$m=m_1m_2,m_1\in{}^*M$ 且 $m_2\in{}^*M$;或者

(3)$m=m_i(i=1,2)$ 且 $m_1m_2\in{}^*M$;或者

(4)$m=f(m_1,\cdots,m_n)$ 且 $\forall i\in\{1,\cdots,n\},m_i\in{}^*M$。

2.2 进程建模

定义 4 $\mathcal{P}$ 是一个进程,表示为一个四元组 $\langle\Sigma,E,\tau,\Sigma_0\rangle$:

(1)Σ 是状态集合(可包含无穷状态);

状态包含两个状态变量:消息集合 MS 和历史事件序列 H

s(MS):表示在状态 s 下进程知道的消息集合;

s(H):表示到状态 s 为止进程已执行的历史事件序列。

(2)E 是事件集合,事件包括以下类型:

exit:进程正常退出;

faillocal:由于系统崩溃或本地强行中断引起进程的异常中止;

failremote:由通信失效引起进程的异常中止;

send(p,m):发送消息 m 给进程 p;

receive(p,m):从进程 p 接收消息 m;

timeout:超时;

random(b):随机产生原子消息 b。b 必须是系统未曾用过的并且其他进程无法计算。

(3)τ 是转换关系,其类型为 $\Sigma\times E\times\Sigma$;

(4)$\Sigma_0\subseteq\Sigma$ 是初始状态集合。

定义 5 若状态转换 $(s,e,s')\in\tau$ 满足:

(1)如果 $e=send(p,m)$,则 $m\in{}^*s(MS)$;

(2)如果 $e=receive(p,m)$ 或 $e=random(m)$,则 $s'(MS)=s(MS)\cup\{m\}$;

(3)如果 $e=random(b)$,则 $\exists m\in s(MS)$ 使得 $b\subseteq m$;

(4)如果 $e\in\{exit,faillocal,failremote,timeout,send(p,m)\}$ 则 $s'(MS)=s(MS)$。

则称 (s,e,s') 为有效的状态转换。

定义 6(进程计算)

$\sigma:s_0e_1s_1\cdots$ 是一个事件和状态的交替序列,且 $S=\langle\Sigma,E,\tau,\Sigma_0\rangle$ 是一个状态转换系统,σ 是 S 的一个计算当且仅当

(1)$s_0\in\Sigma_0$;

(2)对 σ 中的任意子序列 $s_ie_{i+1}s_{i+1}(i=0,1,\cdots,)$ 满足 $s_i,s_{i+1}\in\Sigma,e_{i+1}\in E$,且 $(s_i,e_{i+1},s_{i+1})\in\tau$;

(3)如果 σ 是有限序列,则 σ 最末的元素是一个状态;

(4)若 $s_ie_{i+1}s_{i+1}$ 是 σ 的一个子序列,如果 $e_{i+1}\in\{exit,faillocal,failremote\}$,则 σ 有限且 s_{i+1} 是最末的状态。

2.3 电子商务系统模型

定义 7(不相交并集)

给定集合 $S_1,\cdots,S_n$,我们定义 $S=S_1\uplus\cdots\uplus S_n$ 是 $S_1,\cdots,S_n$ 的不相交并集:$a\in S_i\leftrightarrow a^i\in S$ 表示 S 中的元素 a^i 来自于集合 S_i。

定义 8(系统模型)

$S_1,\cdots,S_n$ 分别是进程 $p_1,\cdots,p_n$ 的进程模型,如果 $\forall i\in\{1,\cdots,n\},S_i=\langle\Sigma_i,E_i,\tau_i,\Sigma_{0i}\rangle$ 则由 $p_1,\cdots,p_n$ 组成的系统模型 $S=\langle\Sigma,E,\tau,\Sigma_0\rangle$:

(1)$\Sigma=\Sigma_1\times\cdots\times\Sigma_n$;

(2)$E=E_1\uplus\cdots\uplus E_n$;

(3)$\tau:\Sigma\times E\times\Sigma$ 定义为:$(\bar{s},\bar{e},\bar{s}')\in\tau$ 当且仅当

$\exists i,\bar{s}=(s_1,\cdots,s_i,\cdots,s_n)\wedge\bar{s}'=(s_1,\cdots,s_i',\cdots,s_n)\wedge\bar{e}=e^i\wedge(s_i,e^i,s_i')\in\tau_i$

$\wedge e^i=random(b)\to(\exists m\in\bigcup_{k=1}^{n}s_k(MS)\mid b\subseteq m)$

(4)$\Sigma_0=\Sigma_{01}\times\cdots\times\Sigma_{0n}$

定义 9(系统计算)

$\sigma:s_0e_1s_1\cdots$ 是一个事件和状态的交替序列,且 $S=\langle\Sigma,E,\tau,\Sigma_0\rangle$ 是一个组合状态转换系统,σ 是 S 的一个计算当且仅当

(1)$s_0\in\Sigma_0$;

(2)对 σ 中的任意子序列 $s_ie_{i+1}s_{i+1}(i=0,1,\cdots,)$ 满足 $s_i,s_{i+1}\in\Sigma,e_{i+1}\in E$,且 $(s_i,e_{i+1},s_{i+1})\in\tau$;

(3)如果 σ 是有限序列,则 σ 的最末元素是状态;

(4)$s_ie_{i+1}s_{i+1}$ 是 σ 的一个子序列,如果 $e_{i+1}\in\{exit^p,faillocal^p,failremote^p\}$($e^p$ 表示进程 p 发生的事件),则不存在事件 $e_j,j>i+1$ 使得 $e_j=e^p$;

(5)如果 σ 是无限序列,则满足弱公平性;

(6)如果 $e_i=receive^q(p,m)\in\sigma$,则 $\exists e_j\in\sigma,j<i,e_j=send^p(q,m)$;

(7)对任意 $send^p(q,m)\in\sigma$,最多存在一个对应的 $receive^q(p,m)\in\sigma$。且每对 $send/receive$ 事件都有以下形式:$e_i=send^p(q,m),e_j=receive^q(p,m),i<j$;

(8)如果 $e_i=receive^q(p,m_1),e_j=receive^q(p,m_2),i<j$,则存在事件 $e_{i'}=send^p(q,m_1),e_{j'}=send^p(q,m_2),i'<j'$;

定义 10(投影)

S 是由 $S_i(S_i=\langle\Sigma_i,E_i,\tau_i,\Sigma_{0i}\rangle,i=1,\cdots,n)$ 组合而成的转换系统,$\sigma:s_0e_1s_1\cdots$ 是 S 的计算,$S_p(p\in\{1,\cdots,n\})$ 的投影 $\sigma\mid p$ 是指:若 $e_r(r\in N)$ 不是 S_p 的事件,则删除 σ 中的 e_rs_r,用 s_r^p(状态 s_r 中进程 p 的状态)来置换剩下的 s_r 得到的序列。

3 协议规范

形式化的协议规范包括以下几个部分:

(1)进程集合 $P=\{p_1,\cdots,p_n\}$;

(2)初始条件 I;

初始条件 I 表示系统开始执行协议时应满足的条件，例如协议中的公、私钥，进程的秘密等相关假设，初始条件用状态公式表示。

定义 11 s 表示状态，$P'\subseteq P$ 表示一个进程集合，m 表示消息（$s^p(MS)$ 表示进程 p 在状态 s 时知道的消息集合）。如果 $\forall p\in P-P', m\notin {}^* s^p(MS)$，则 $s\vdash shareable(m,P')$

(3)进程协议集合 $R=\{R_{p_1},\cdots,R_{p_n}\}$

$R_{p_i}(i=1,\cdots,n)$ 是进程 p_i 的协议，表示为协议规则的集合。规则 $r\in R_p$ 的形式为 $r:\eta\rightarrow\{E_1,\cdots,E_n\}$

η 称为使能条件，表示允许 E_i 发生的条件，用状态公式描述；E_i 表示在满足 η 的状态允许进程发生的事件。$E_i\in\{exit,timeout,send,receive,random\}$，$faillocal$ 和 $failremote$ 是异常中止不是由协议规定的。变量 H 记录进程的历史事件序列，若 $s(H)=e_1,\cdots,e_m,e_m$ 是进程最近执行的事件，$\sigma: s_0e_1s_1\cdots$ 是一个进程计算，则 $s(H)$ 满足：$s_0(H)=\in$ 且 $s_{i+1}(H)=append(s_i(H),e_{i+1})$（'$\in$'表示空序列，'$append$'表示在序列的末尾增加一个元素）。

(4)信任假设集合 $T=\{T_{p_1},\cdots,T_{p_n}\}$

若 p_i 是可信任的进程，$T_{p_i}=\{T_1,\cdots,T_n\}$，$T_i$ 用时序逻辑公式描述。若 p_i 不是可信任的进程，则令 $T_{p_i}=$true。

4 协议的执行序列

定义 12 E 表示事件模板，若对 E 中的所有自由变量 $x_1,\cdots,x_n$ 有代入 $[x_1/m_1,\cdots,x_n/m_n]$，使得 $e=E[x_1/m_1,\cdots,x_n/m_n]$，则称事件 e 是 E 的一个实例。

定义 13 (ses') 表示状态转换，$r:\eta\rightarrow\{E_1,\cdots,E_n\}$ 表示协议规则，(ses') 遵守协议规则 r 记为 $(ses')\vdash r$，当且仅当：

(1)$s\vdash\eta$ 并且

(2)a) $\exists E_i\in\{E_1,\cdots,E_n\}$，$e$ 是 $E_i\theta$ 的实例，θ 是用状态 s 下的取值对 E_i 中约束变量的一个代入；或者

b) $\exists E_i\in\{E_1,\cdots,E_n\}$，$E_i$ 是一个 $send$ 或者 $receive$ 事件，$e=failremote$。

定义 14 (ses') 表示状态转换，$r:\eta\rightarrow\{E_1,\cdots,E_n\}$ 是协议规则，(ses') 欺骗性的遵守协议规则 r

记为 $(ses')\vdash_d r$ 当且仅当

(1)$s\vdash\eta$ 并且

(2) $\exists E_i\in\{E_1,\cdots,E_n\}$，$e$ 是 E_i 但不是 $E_i\theta$ 中的实例，θ 是用状态 s 下的取值对 E_i 中约束变量的一个代入。

定义 15(进程协议的遵守型执行序列)

$\sigma: s_0e_1s_1\cdots$ 是进程 p 的计算，R_p 是 p 的协议，σ 是 R_p 的遵守型执行序列

当且仅当 $\forall(s_ie_{i+1}s_{i+1})\in\sigma$，$\exists r\in R_p$ 使得 $(s_ie_{i+1}s_{i+1})|=r$

定义 15 说明计算 σ 是 R_p 的遵守型执行序列当且仅当 σ 中的任何状态转换都能从 R_p 中找到一条应用于它的规则。

定义 16(协议的遵守型执行序列)

$\sigma: s_0e_1s_1\cdots$ 是系统计算，$\langle P,I,R,T\rangle$ 表示协议，σ 是该协议的遵守型执行序列当且仅当

1. $s_0\vdash I$；且

2. $\forall p\in P,\sigma|p$ 是 R_p 的遵守型执行序列。

定义 16 说明系统计算是协议的遵守型执行序列当且仅当它的初始状态满足协议的初始条件 I，并且该计算对各进程投影后的进程计算分别是该进程协议的遵守型执行序列。

定义 17(进程协议的中断型执行序列)

$\sigma: s_0e_1s_1\cdots s_n$ 是进程 p 的计算，R_p 是 p 的协议，σ 是 R_p 的中断型执行序列当且仅当

$s_0e_1s_1\cdots s_{n-1}$ 是 R_p 的遵守型执行序列且 $e_n=faillocal$。

定义 18(进程协议的欺骗型执行序列)

$\sigma: s_0e_1s_1\cdots$ 是进程 p 的计算，R_p 是 p 的协议，σ 是 R_p 的欺骗型执行序列当且仅当

只有当状态转换 $(s_ie_{i+1}s_{i+1})\in\sigma$ 使得 $(s_ie_{i+1}s_{i+1})\vdash_d r(r\in R_p)$ 时，σ 才不是 R_p 的遵守型执行序列。

定义 19(协议的中断型(欺骗型)执行序列)

$\sigma: s_0e_1s_1\cdots$ 是系统计算，$\langle P,I,R,T\rangle$ 表示协议，σ 是该协议的中断型(欺骗型)的执行序列当且仅当

(1)$s_0\vdash I$；且

(2)至少存在一个进程 $p\in P$，使得 $\sigma|p$ 是 R_p 的一个中断型(欺骗型)的执行序列。

(3)其他的进程 p'，$\sigma|p'$ 是 $R_{p'}$ 的遵守型执行序列。

定义 20(协议的最大执行序列)

σ 是协议 Π 的遵守/中断/欺骗型执行序列，σ 是协议 Π 的最大遵守/中断/欺骗型执行序列当且仅当不存在这样的 σ'，它既是协议 Π 的遵守/中断/欺骗型执行序列，又是 σ 的前缀。

定义 21(协议的可信任执行序列)

σ 是协议 $\langle P,I,R,T\rangle$ 的遵守/中断/欺骗型执行序列，$T=\{T_1,\cdots,T_k\}$，$\sigma\backslash s$ 表示 σ 对状态的投影，σ 是可信任执行序列当且仅当

$\sigma\backslash s\vdash {}^*\bigwedge_{i=\{1,\cdots,k\}\tau\in T_i}\bigwedge\tau$ 或者 $\sigma\backslash s\vdash {}^* T$

以后分别用 C，D 和 A 分别表示遵守、欺骗和中断模式，$E(\Pi)^C$，$E(\Pi)^D$ 和 $E(\Pi)^A$ 表示协议 Π 的最大遵守、欺骗和中断型执行序列，$E(\Pi)_p^x$ 表示协议 Π 的最大 $x(x\in\{C,D,A\})$ 型执行序列，并且该序列对进程 p 投影后得到的序列是协议 R_p 的遵守型执行序列。

5 Z&G 协议分析

5.1 Z&G 协议

Z&G 协议(图 1)：

1. $A\rightarrow B: f_{NRO},B,L,C,NRO$
2. $B\rightarrow A: f_{NRR},A,L,NRR$
3. $A\rightarrow TTP: f_{SUB},B,L,K,sub\_K$
4. $B\leftrightarrow TTP: f_{CON},A,B,L,K,con\_K$
5. $A\leftrightarrow TTP: f_{CON},A,B,L,K,con\_K$

协议中符号的含义：

A：交易的发起方。

B：交易的响应方。

TTP：可信任第三方。

M：A 发送给 B 的消息。

K：A 产生的消息密钥。

C：用密钥 K 对消息 M 加密后的密文。

L：交易的标识符，每次交易开始时由 A 产生一个新的交易标识符。

f：标记签名消息的类型。

X,Y：消息 X 和 Y 的连接。

$eK(X)$ 和 $dK(X)$：用 K 对 X 加密和解密。

P_x 和 S_x：x 的签名公钥和私钥。

$sS_x(X)$：用私钥 S_x 对 X 签名；

$x\rightarrow y:m$：x 发送消息 m 给 y；

$x\leftrightarrow y:m$：x 使用"ftp get"操作从 y 处取消息 m。

("ftp get"表示即使信道不可靠，x 最终也可以通过多次 ftp 操作从 y 处获取 m)

$NRO=sS_A(f_{NRO},B,L,C)$：C 的发方非否认证据。

$NRR=sS_B(f_{NRR},B,L,C)$：C 的收方非否认证据。

$sub\_K=sS_A(f_{SUB},B,L,K)$:$K$的提交证据。

$con\_K=sS_{TTP}(f_{CON},A,B,L,K)$:$TTP$发出的确认$K$的证据。

$(x,y\in\{A,B,TTP\})$

图1 Z&G协议

step1中,A发起一次新的交易,并产生新的交易标识符L和随机密钥k,用k加密M成C,接着A对C、L和B签名,生成A发送C的非否认证据NRO,然后将B、L、C和NRO传送给B;step2中B接收NRO和C后,验证NRO是否正确,若是则发送C的接收证据NRR给A。step3中A收到NRR后,检查NRR和NRO是否属于同笔交易,验证NRR,若都满足要求A发送密钥K和提交密钥证据sub_K给TTP。step4中TTP收到k和sub_K后,验证A的签名正确后将五元组(A,B,L,K,con_K)公布在其公共目录上,只对公众开放只读权限。元组的第一项表示密钥k的提供者。step5中B使用"ftp get"操作从TTP获取k和con_K。step6中A使用"ftp get"操作从TTP获取k和con_K。step5、step6没有先后,也可以同时进行。

协议假设TTP是绝对可信任的,TTP收到A提交的密钥和密钥提交证据后,一定会将A提交的密钥公布在其目录上,而且假设TTP一旦公布了该密钥,则A,B都能从TTP的目录上拿到密钥和密钥证据。

5.2 协议的抽象

5.2.1 基本密码块的抽象

(1)密码块类型

Z&G协议中的密码块抽象为以下类型:对称密钥类型t_symk;私钥类型t_prik;公钥类型t_pubk;签名类型t_seal;进程标识符类型t_pid;交易标识符类型t_tid;超类型t,有些情况下只需将消息看作位串流而非结构化的消息,例如加密一个消息,无需区分这个消息是密钥或者进程标识符等,这时消息属于类型t。

(2)函词和谓词

$enc:t\times t\_symk\to t$;$dec:t\times t\_symk\to t$;$seal:t\times t\_prik\to t\_seal$;$keypair:t\_prik\times t\_pubk\to boolean$;

$vseal:t\times t\_seal\times t\_pubk\to boolean$。

$keypair(k,k^{-1})=$true当且仅当k和k^{-1}是公钥系统中的一对密钥,k是公钥,k^{-1}私钥。

$$vseal(m,s,k)=\begin{cases}\text{true},\text{如果}\exists k^{-1}:t\_prik\mid s=seal(m,k^{-1})\wedge keypair(k,k^{-1});\\ \text{false},\text{否}\end{cases}$$

5.2.2 复合类型及相应的投影函数

(1)公钥证书类型$t_cert:t_pid\times t_pubk$。若$m=m_1m_2$类型为t_cert,则$\begin{cases}pid(m)=m_1\\key(m)=m_2\end{cases}$;

(2)消息的非否认证据类型$t_unr:t_pid\times t_tid\times t$。若$m=m_1m_2m_3$类型为t_unr,则$\begin{cases}pid(m)=m_1\\tid(m)=m_2\\eg(m)=m_3\end{cases}$;

(3)提交密钥的证据类型$t_usubk:t_pid\times t_tid\times t_symk$。若$m=m_1m_2m_3$类型为t_usubk,则$\begin{cases}pid(m)=m_1\\tid(m)=m_2\\key(m)=m_3\end{cases}$;

(4)发布密钥的证据类型$t_uconk:t_pid\times t_pid\times t_tid\times t_symk$。若$m=m_1m_2m_3m_4$类型为t_uconk,则$\begin{cases}oid(m)=m_1\\rid(m)=m_2\\tid(m)=m_3\\key(m)=m_4\end{cases}$;

(5)消息的非否认证据的签名类型$t_nr:t_unr\times t_seal$;

(6)提交密钥的证据的签名类型$t_subk:t_usubk\times t_seal$;

(7)发布密钥的证据的签名类型$t_conk:t_uconk\times t_seal$;

若$m=m_1m_2$类型为5,6或7,则$\begin{cases}msg(m)=m_1\\sig(m)=m_2\end{cases}$。

投影函数pid计算进程标识符;key计算证书中的公钥;tid计算交易的标识符;eg计算类型为t的字节流;oid计算交易的发起方的标识符;rid计算交易的响应方的标识符;msg计算被签名的消息;sig计算签名。

5.3 Z&G协议模型

为了描述Z&G协议中的"ftp get",引入一个新的事件类型notes(p,m):进程p将消息m记录在本地。从而关于系统计算定义9需要补充一条规则

(9)如果$e_i=notes(p,m)\in\sigma$,则一定存在$e_j=notes(TTP,m)\in\sigma(j<i,p\in\{A,B\})$。

5.3.1 Z&G协议的形式化规范

(1)协议进程集合$P=\{A,B,TTP\}$。

(2)协议的初始条件

$$\begin{aligned}I=&shareable(k_a^{-1},\{A\})\wedge shareable(k_b^{-1},\{B\})\wedge shareable(k_{ap}^{-1},\{TTP\})\\&\wedge shareable(\gamma,\{A\})\wedge keypair(k_a^{-1},\phi_a.key)\wedge keypair(k_b^{-1},\phi_b.key)\\&\wedge keypair(k_{ap}^{-1},k_{ap})\\&\wedge\{k_a^{-1},\Theta,\gamma,\phi_b,k_{ap}\}\subseteq MS^A\\&\wedge\{k_b^{-1},\phi_a,k_{ap}\}\subseteq MS^B\\&\wedge\{k_{ap}^{-1},\phi_a,\phi_b,\Theta\}\subseteq MS^{TTP}\end{aligned}$$

$k_a^{-1},k_b^{-1},k_{ap}^{-1}$分别是$A,B$和$TTP$的签名私钥;$\phi_a,\phi_b$是$A,B$的公钥证书;$\gamma$是待交换的消息;$\Theta$是旧的交易标识符集合。

(3)进程协议集合$R=\{R_A,R_B,R_{TTP}\}$

$R_A=\{R_{A1},\cdots,R_{A7}\}$

$R_{A1}.\ \exists x:t\_tid\mid new(\Theta,x)\in H\Rightarrow\{new(\Theta,y:t\_tid),exit\}$

$R_{A2}.\ \exists x:t\_symk\mid random(x)\in H\Rightarrow\{random(y:t\_symk),exit\}$

$R_{A3}.\ \exists x:t\_nr\mid(send(B,x)\notin H\wedge new(\Theta,x.msg.tid)\in H\wedge\exists y:t\_symk\mid random(y)\in H)\Rightarrow\{send(B,x),exit\}$

其中$x.msg.eg=enc(\gamma,y)$

$R_{A4}.\ \exists x:t\_nr\mid last(H)=send(B,x)\Rightarrow\{receive(B,y:t\_pid\times t\_tid\times t\_seal),timeout,exit\}$

$R_{A5}.\ \exists x_1x_2x_3:t\_pid\times t\_tid\times t\_seal\mid last(H)=receive(B,x_1x_2x_3)\wedge\exists y:t\_symk\mid random(y)\in H\wedge vseal(x_1x_2eg,x_3,\phi_b.key)\wedge new(\Theta,x_2)\in H\Rightarrow\{send(TTP,\phi_b.id\,x_2y\,seal(\phi_b.id\,x_2y,k_a^{-1})),exit\}$

其中$eg=enc(\gamma,y)$

$R_{A6}.\ \exists x:t\_subk\mid last(H)=send(TTP,x)\Rightarrow\{notes(A,y:t\_conk),timeout\}$

$R_{A7}.\ \wedge(\neg vseal(x_1x_2eg,x_3,\phi_b.key)))\vee last(H)=timeout\vee\exists z:t\_conk\mid notes(A,z)\Rightarrow\{exit\}$

其中$eg=enc(\gamma,y)$

进程A的7条协议规则分别表示:产生新的交易标识

符;产生随机会话密钥;如果 A 未给 B 发送过加密消息 C,则发送 C 或者退出协议;如果 A 已经给 B 发送过 C,则等待接收 B 的确认消息直至收到或超时或退出协议。如果 A 从 B 处收到了确认消息,并且验证了 B 的签名正确,则发送密钥 K 给 TTP 或退出协议。若 A 已将密钥 K 发送给 TTP,则等待接收 TTP 的确认消息直至收到或超时。若 A 从 B 收到了确认消息,但 B 的签名不正确,或者等待超时,或者收到了 TTP 处的密钥确认消息,则 A 退出协议。

$R_B=\{R_{B1},R_{B2},R_{B3},R_{B4}\}$

R_{B1}. $\exists x:t\_nr \mid receive(A,x)\in H\Rightarrow\{receive(A,x),timeout,exit\}$

R_{B2}. $\exists x:t\_nr \mid last(H)=receive(A,x)\wedge vseal(x.msg,x.sig,\phi_a.key)\Rightarrow\{send(A,\phi_a.id\ x.msg.tid\ seal(nrr,k_b^{-1})),exit\}$

其中 $nrr=\phi_a.id\ x.msg.tid\ x.msg.eg$

R_{B3}. $\exists x:t\_nr \mid last(H)=send(A,x)\Rightarrow\{notes(B,y:t\_conk),timeout\}$

R_{B4}. $(\exists x:t\_nr \mid last(H)=receive(A,x)\wedge\neg vseal(x.msg,x.sig,\phi_a.key))\vee last(H)=timeout\vee\exists y:t\_conk \mid last(H)=notes(B,y)\Rightarrow\{exit\}$

进程 B 的协议规则分别表示为:若 B 没有从 A 处收到类型为 t_nr 的消息,则等待接收直至收到或超时,B 也可以退出协议。若 B 收到了类型为 t_nr 的消息,且验证 A 的签名正确,则发送确认消息给 A 或者退出协议;若上一个事件是发送了消息给 A,则等待从 TTP 接收密钥 K 直至收到或者超时。如果 B 从 A 收到了消息,但是签名不正确,或者等待消息超时,或者从 TTP 拿到了密钥 K,则 B 退出协议。

$R_{TTP}=\{R_{T1},R_{T2},R_{T3}\}$

R_{T1}. $\exists x:t\_subk \mid receive(A,x)\in H\Rightarrow\{receive(A,x),timeout,exit\}$

R_{T2}. $\exists x:t\_subk \mid last(H)=receive(A,x)\in H\wedge vseal(x.msg,x.sig,\phi_a,key)\Rightarrow notes(TTP,conk,seal(conk,k_{up}^{-1}))$

其中 $conk=\phi_a.id\ x.msg.pid\ x.msg.tid\ x.msg.key$

R_{T3}. $\exists x:t\_subk \mid last(H)=receive(A,x)\in H\wedge\neg vseal(x.msg,x.sig,\phi_a.key)\vee\exists y:t\_conk \mid last(H)=notes(TTP,y)\Rightarrow\{exit\}$

进程 TTP 的协议规则表示为:若 TTP 没有从 A 处收到了密钥,则等待直至接收到密钥或者超时,TTP 也可退出协议。若 TTP 已收到密钥,并且验证 A 的签名正确,则 TTP 记录该密钥并公布。若 TTP 收到密钥但验证 A 的签名不正确或者 TTP 已经记录了该密钥,则 TTP 退出协议。

(4)信任假设集合

$T_{TTP}=\{T_1,T_2\}$,$T_A=$true,$T_B=$true。

T_1: $\forall x:t\_conk,\Box(notes(TTP,x)\in H\rightarrow(\exists y:t\_subk \mid receive(A,y)\in H\wedge x.msg.oid=\phi_a.id\wedge x.msg.rid=y.msg.id\wedge x.msg.tid=y.msg.tid\wedge x.msg.key=y.msg.key))$

T_2: $\forall x:t\_conk,\Box(notes(TTP,x)\in H\rightarrow vseal(x.msg,x.sig,k_{up}))$

T_1 表示若 TTP 记录密钥,则 TTP 一定从 A 处收到了该密钥(TTP 不会用错误的密钥,错误的 ID 号或者错误交易标识符)。T_2 表示 TTP 公布的密钥一定有 TTP 的正确的签名。

5.4 公平性

Z&G 协议中,A 需要的公平性是指在 A 严格遵守协议的前提下,若 B 收到了消息 M,则 A 有证据证明 B 接收了 M:1)A 收到 B 关于 M 的非否认证据 NRR;2)A 收到了 TTP 发布的密钥 K 和证书 conk;3)这里的 NRR 和 conk 属于同一笔交易,并且 NRR 中的加密消息确实是证书 conk 中的密钥加密 M 后的消息。

B 需要的公平性是指在 B 严格遵守协议的前提下,若 A 有 conk 和 NRR,则 B 确实收到了消息 M:1)B 收到了 A 发送 M 的非否认证据 NRO;2)B 收到了 conk;3)这里 NRO 和 conk 属于同一笔交易,用 conk 中的密钥解密 NRO 中的密文后确实是 M。

用时序逻辑公式描述协议的公平性:

$\Phi_A:\exists x_1x_2x_3:t\_pid\times t\_tid\times t\_seal,y:t\_conk \mid receive(B,x_1x_2x_3)\in H^A\wedge notes(A,y)\in H^A$

$x_2=y.msg.tid\wedge vseal(x_1x_2eg,x_3,k_b)\wedge vseal(y.msg,y.sig,k_{up})$

$\Phi_B:\exists x:t\_nr,y:t\_conk \mid receive(A,x)\in H^B\wedge notes(B,y)\in H^B$

$x.msg.tid=y.msg.tid\wedge dec(x.msg.eg,y.msg.key)=\gamma$

其中 $eg=enc(\gamma,y.msg.key)$

A 需要的公平性表示为 $P_A:\Box(\Phi_B\rightarrow\Diamond\Phi_A)$;B 需要的公平性表示为 $P_B:\Box(\Phi_A\rightarrow\Diamond\Phi_B)$

5.5 Z&G 协议的分析

分析 Z&G 协议采取的证明策略是先证明可信任的执行序列满足公平性,从而遵守型的最大执行序列也满足公平性,因为可以证明遵守型的执行序列属于可信任的执行序列。以下我们从三种模式(欺骗型、遵守型和中断型)分析协议的执行序列是否满足公平性。

notes 公理. $\forall\sigma\in E(\Pi)^C\cup E(\Pi)^D\cup E(\Pi)^A$,

$\sigma\vdash^*\forall x:t\_conk \mid \Box(notes(TTP,x)\in H^{TTP}\rightarrow(\Diamond notes(A,x)\in H^A\wedge\Diamond notes(B,x)\in H^B))$

定理 1 Π 是 Z&G 协议,$\mathcal{T}$ 是 Π 中的信任假设集合,则 $\forall\sigma\in E(\Pi)^C,\sigma\vdash^*\mathcal{T}$

证明:TTP 是唯一可信任的进程,$\mathcal{T}=T_{up}=\{T_1,T_2\}$。下面先证明 $\forall\sigma\in E(\Pi)^C,\sigma\vdash^* T_1$:

(1)假设 σ 中存在状态 s_i,并且 $\exists m_1:t\_conk$ 使得 $s_i\vdash notes(TTP,m_1)\in H^{TTP}$。检查进程 TTP 的协议,事件 $notes(TTP,m_1)$ 只能是规则 R_{T2} 的应用结果。

(2)应用规则 R_{T2},存在消息 $\exists m_2:t\_subk$ 使得 $s_i\vdash receive(A,m_2)\in H^{TTP}\wedge(m_2.msg.tid=m_1.msg.tid)\wedge(m_2.msg.pid=m_1.msg.rid)\wedge(m_2.msg.key=m_1.msg.key)\wedge(m_1.msg.oid=\phi_a.id)$

证明:$\forall\sigma\in E(\Pi)^C,\sigma\vdash^* T_2$:

(1)同上;

(2)应用规则 R_{T2},存在 $\exists b:t\_uconk$ 使得 $m_1=b\ seal(b,k_{up}^{-1})$;

(3)假设 $keypair(k_{up}^{-1},k_{up})=$true 则 $vseal(b,seal(b,k_{up}^{-1}),k_{up})=$true。 □

5.5.1 欺骗型模式

欺骗型模式的证明策略是先证明可信任的欺骗型执行序列满足进程需要的公平性,接着证明遵守型的执行序列满足进程需要的公平性。

命题 1 Π 是 Z&G 协议,$\sigma\in E(\Pi)_R^P$,$\mathcal{T}$ 是 Π 中的信任假设集合,$P_A=\Box(\Phi_B\rightarrow\Diamond\Phi_A)$ 是关于 A 的公平性,如果 $\sigma\vdash^*$

T,则 $\sigma \vdash^* P_A$

证明:$\forall \sigma \in E(\prod)_A^D$,且 $\sigma \vdash^* T$,假设 σ 中存在状态 s_i 使得 $s_i \vdash \Phi_B$,下面证明 $s_i \vdash \Diamond \Phi_A$:

(1)如果 $s_i \vdash \Phi_B$,则 $\exists m_1 : t\_conk$,

$s_i \vdash notes(B, m_1) \in H^B$,由系统计算定义的补充规则 9 可知

$$s_i \vdash notes(TTP, m_1) \in H^{TTP} \tag{1.1}$$

(2)由式(1.1)和信任规则 T_1,可知 $\exists m_2 : t\_subk$,

$$s_i \vdash receive(A, m_2) \in H^{TTP} \wedge m_2.msg.tid = m_1.msg.tid \wedge m_2.msg.key = m_1.msg.key \tag{1.2}$$

(3)根据系统计算定义规则 6 和式(1.2)可知

$$s_i \vdash send(TTP, m_2) \in H^A \wedge m_2.msg.tid = m_1.msg.tid \wedge m_2.msg.key = m_1.msg.key \tag{1.3}$$

检查进程 A 的协议,发现事件 $send(TTP, m_2)$ 只能是应用规则 R_{A5} 的结果。

(4)应用规则 R_{A5},则 $\exists m_3 m_4 m_5 : t\_pid \times t\_tid \times t\_seal$,

$$s_i \vdash receive(B, m_3 m_4 m_5) \in H^A \wedge vseal(m_3 m_4 eg, m_5, \phi_b.key) \wedge m_4 = m_2.msg.tid \tag{1.4}$$

其中 $eg = enc(\gamma, m_2.msg.key)$

由式(1.3)可知 $m_2.msg.tid = m_1.msg.tid$,所以 $eg = enc(\gamma, m_2.msg.key) = enc(\gamma, m_1.msg.key)$。

(5)由式(1.3)可知 $m_2.msg.tid = m_1.msg.tid$,由式(1.4)可知 $m_4 = m_2.msg.tid$,由传递性可知

$$m_1.msg.tid = m_4 \tag{1.5}$$

(6)由式(1.1)和信任规则 T_2,可知

$$s_i \vdash vseal(m_1.msg, m_1.sig, k_{up}) \tag{1.6}$$

(7)由式(1.1)和 notes 公理,可知 $\exists s_j (j \geqslant i)$,

$$s_j \vdash notes(A, m_1) \in H^A \tag{1.7}$$

由式(1.4),(1.5),(1.6),(1.7)可知 $s_j \vdash \Phi_A (j \geqslant i)$。

命题 2 $\prod$ 是 Z&G 协议,σ 是 $E(\prod)^C$ 中的一个执行序列,$P_A = \Box(\Phi_B \rightarrow \Diamond \Phi_A)$ 是关于 A 的公平性,则 $\sigma \vdash^* P_A$

证明:对任意 $\sigma \in E(\prod)^C$,由定理 3 可得 $\sigma \vdash^* \mathcal{T}$。在命题 7 的证明中可知,只要 σ 是可信任的并且对 A 的投影是协议 R_A 的遵守型执行序列,$\sigma \vdash^* P_A$,同理可证 $\sigma \vdash^* P_A$。

推论 1 欺骗型模式下的 Z&G 协议满足 A 需要的公平性。

证明:由命题 1 和命题 2 可证。

命题 3 $\prod$ 是 Z&G 协议,σ 是 $E(\prod)_B^D$ 中的一个执行序列,$\mathcal{T}$ 是 $\prod$ 中的信任假设集合,$P_B = \Box(\Phi_A \rightarrow \Diamond \Phi_B)$ 是关于 B 的公平性,如果 $\sigma \vdash^* T$,则 $\sigma \vdash^* P_B$。

证明:$\forall \sigma \in E(\prod)_B^D$,且 $\sigma \vdash^* T$,假设 σ 中存在状态 s_i 使得 $s_i \vdash \Phi_A$,下面证明 $s_i \vdash \Diamond \Phi_B$:

(1)如果 $s_i \vdash \Phi_A$,则 $\exists m_1 : t\_conk, m_2 m_3 m_4 : t\_pid \times t\_tid \times t\_seal$ 使得

$$s_i \vdash receive(B, m_2 m_3 m_4) \in H^A \wedge notes(A, m_1) \in H^A \wedge m_3 = m_1.msg.tid \wedge vseal(m_2 m_3 eg, m_4, k_b) \wedge vseal(m_1.msg, m_1.sig, k_{up}) \tag{3.1}$$

其中 $eg = enc(\gamma, m_1.msg.key)$

(2)由式(3.1)可知

$s_i \vdash receive(B, m_2 m_3 m_4) \in H^A$

由系统计算定义的规则 6 可知

$$s_i \vdash send(A, m_2 m_3 m_4) \in H^B \tag{3.2}$$

检查进程 B 的协议,$send(A, m_2 m_3 m_4)$ 只能是应用规则 R_{B2} 的结果。

3. 应用规则 R_{B2},则 $\exists m_5 : t\_nr$,

$$s_i \vdash receive(A, m_5) \in H^B \wedge m_3 = m_5.msg.tid \wedge m_4 = seal(\phi_a.id\ m_5.msg.tid\ m_5.msg.eg, k_b^{-1}) \wedge m_2 = \phi_a.id \tag{3.3}$$

(4)由式(3.1)可知 $m_3 = m_1.msg.tid$,由式(3.3)可知 $m_3 = m_5.msg.tid$,由传递性可知

$$m_5.msg.tid = m_1.msg.tid \tag{3.4}$$

(5)由式(3.1)可知

$s_i \vdash notes(A, m_1) \in H^A$

由系统计算定义规则 6 可知:

$$s_i \vdash notes(TTP, m_1) \in H^{TTP} \tag{3.5}$$

(6)由式(3.5)和 notes 公理,可知存在 $s_j (j \geqslant k)$:

$$s_j \vdash notes(B, m_1) \in H^B \tag{3.6}$$

(7)由式(3.1)可知 $s_i \vdash vseal(m_2 m_3 eg, m_4, k_b)$ 其中 $eg = enc(\gamma, m_1.msg.key)$,所以

$$m_4 = seal(m_2 m_3 eg, k_b^{-1}) \tag{3.7}$$

(8)由式(3.3)可知 $m_4 = seal(\phi_a.id\ m_5.msg.tid\ m_5.msg.eg, k_b^{-1})$,由式(3.6)可知 $m_4 = seal(m_2 m_3 eg, k_b^{-1})$,所以

$seal(\phi_a.id\ m_5.msg.tid\ m_5.msg.eg.\ k_b^{-1}) = seal(m_2 m_3 eg, k_b^{-1})$

由式(3.3)可知 $m_2 = \phi_a.id$,再由式(3.5)可知

$$m_5.msg.eg = enc(\gamma, m_1.msg.key) \tag{3.8}$$

由式(3.3)、(3.4)、(3.6)、(3.8)可知 $s_j \vdash \Phi_B (j \geqslant i)$。

命题 4 $\prod$ 是 Z&G 协议,σ 是 $E(\prod)^C$ 中的一个执行序列,$P_B = \Box(\Phi_A \rightarrow \Diamond \Phi_B)$ 是关于 B 的公平性,则 $\sigma \vdash^* P_B$。

推论 2 欺骗型的 Z&G 协议执行序列满足 B 需要的公平性。

定理 2 欺骗型的 Z&G 协议执行序列满足公平性。

5.5.2 遵守型模式

定理 3 遵守型的 Z&G 协议满足主体需要的公平性。

5.5.3 中断型模式

命题 5 $\prod$ 是 Z&G 协议,σ 是 $E(\prod)_A^A$ 中的一个执行序列,$\mathcal{T}$ 是 $\prod$ 中的信任假设集合,$P_A = \Box(\Phi_B \rightarrow \Diamond \Phi_A)$ 是关于 A 的公平性,如果 $\sigma \vdash^* T$,那么 $\sigma \vdash^* P_A$。

推论 3 中断型的 Z&G 协议执行序列满足 A 需要的公平性。

命题 6 $\prod$ 是 Z&G 协议,σ 是 $E(\prod)_B^A$ 中的一个执行序列,$\mathcal{T}$ 是 $\prod$ 中的信任假设集合,$P_B = \Box(\Phi_A \rightarrow \Diamond \Phi_B)$ 是关于 B 的公平性,如果 $\sigma \vdash^* T$,那么 $\sigma \vdash^* P_B$。

推论 4 中断型的 Z&G 协议执行序列满足 B 需要的公平性。

定理 4 中断型的 Z&G 协议执行序列满足公平性。

因篇幅有限,以上部分命题、定理没有给出详细证明,其证明方法类似已证命题和定理的证明方法,作者已经证明这些结论是成立的。

结束语 电子商务协议假设各进程之间的通信安全(即底层安全协议提供通信的秘密性、认证性和消息的完整性),因而协议分析主要是分析参与者发生欺骗行为时协议是否满足其安全目标。本文从电子商务协议的安全目标出发,根据电子商务系统的参与者的行为可能背离协议规则的程度,将系统行为分成三种模式:遵守型、欺骗型和中断型。于是,系统模型自然直观地描述了电子商务系统的特点。系统模型独立于协议,是一个抽象的、通用的模型,适用于各类不同的电子商务协议的分析。文中以 Z&G 协议为例,证明了三种模

式下的协议执行序列满足其公平性，在遵守型模式和可信任的背离模式下的证明过程基本相同，证明简洁易懂。然而我们的分析是建立在底层密码协议提供信道安全的基础之上，密码协议的前提是参与者必须诚实，但是电子商务协议中参与者却可能不诚实，因此采用协议分层的方法来验证电子商务协议属性还有待进一步研究。

参 考 文 献

[1] Syverson P F, Cervesato I. The logic of authentication protocols // Focardi R, Gorrieri R, eds. Foundations of Security Analysis and Design. volume LNCS 2171, Springer-Verlag, 2001

[2] Zhou Jianying, Gollmann D. Towards verification of non-repudiation protocols // Vickers T, Grundy J, Schwenke M, eds. International Refinement Workshop and Formal Methods Pacific 1998. Springer-Verlag, 1998: 370-380

[3] Wong H-C. Protecting Individuals' Interests in Electronic Commerce Protocols [D]. Computer Science Deptartment, Carnegie Mellon University, 2000

[4] Bella G, Paulson L C. Mechanical proofs about a non-repudiation protocol // Boulton R J, Jackson P B, eds. Theorem Proving in Higher Order Logics: TPHOLs 2001, LNCS 2152. Springer, 2001: 91-104

[5] Longo C, Bella G, Paulson L C. Verifying second-level security protocols // 16th International Conference on Theorem Proving in Higher Order Logics, volume LNCS 2758. Springer Verlag, 2003: 352-366

[6] Zhou Jianying, Gollmann D. An Efficient Non-repudiation Protocol // Proceedings of the 1997 IEEE Computer Security Foundations Workshop (CSFW 10). IEEE CS Press, 1997: 126-132

[7] Schneider S. Formal Analysis of a Non-repudiation Protocol // Proceedings of the 11th IEEE Computer Security Foundations Workshop. 1998: 54

[8] Qing S H, Li G C. A formal model of fair exchange protocols [J]. Science in China Series F-Information Sciences, 2005, 48 (4): 499-512

[9] Zhang L, Yin J P, Long J. Formal Analysis of Fairness of the ZG protocol[C] // Proceeding of Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (IEEE SecPerU 2005)

[10] Zhang L, Yin J P, Li M J. Formal Analysis of NetBill and Improvement[C] // Proceedings of the SKLOIS Conference on Information Security and Cryptology (CISC2005). Higher Education Press, 2005: 287-296

[11] 卿斯汉. 安全协议的设计与逻辑分析. 软件学报, 2003, 14(7): 1300-1309

[12] 周典萃，卿斯汉，周展飞. 一种分析电子商务协议的新工具. 软件学报, 2001, 12(9): 1318-1328

(上接第 55 页)

够在断链之前抢先发起路由维护，将有断链威胁的路径上的数据包平稳切换到其它路径，使对吞吐量的影响最低。

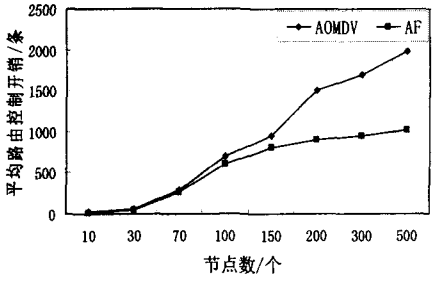

图 5 平均路由控制消息开销的比较

图 5 模拟节点数逐渐增加情况下，AOMDV 和基于 AF 的安全路由机制在平均路由控制消息开销上的比较。仿真结果显示，节点数达到 100 后，AOMDV 的 RREQ 消息数量会随节点数增加而大幅增加；而基于 AF 的安全路由机制由于有选择地向 AF 值低于安全阈值的周边节点转发 RREQ，并保证同一节点不会参与多条路径的形成，RREQ 消息数量增长较少。

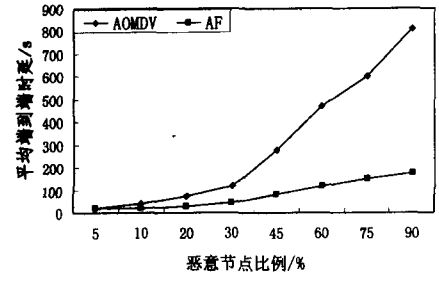

图 6 平均端到端时延的比较

图 6 模拟恶意节点比例逐渐升高情况下 AOMDV 和基于 AF 的安全路由机制在平均端到端时延上的比较。仿真结果显示，AOMDV 在攻击逐渐加剧时，大量路由控制消息在传输过程中丢失，导致端到端时延增加；基于 AF 的安全路由机制有效避开了有威胁的节点，保护了路由控制消息，即便恶意节点比例达到 90%，路由控制消息也只会经过那些安全节点，使得端到端时延保持在一个较低的水准上。

结束语 本文通过攻击因子 AF 衡量节点当前和未来遭受攻击的可能性，在此基础上建立了安全多路径集，并实现了抢先式路由维护。仿真实验证明，在基于 AF 的多路径安全路由机制的网络中，即使有相当比例的恶意节点存在，也不会对正常的路由进程产生太大影响。在保障多路径安全的同时，提高了路由建立的效率。将本文机制应用于动态环境下的大型网络中有较好的前景，值得进一步研究。

参 考 文 献

[1] Li Xuefei, Laurie G. Node-disjointness-based multipath routing for mobile ad hoc networks[C] // Proceedings of the 1st ACM International Workshop on PE-WASUN. 2004: 23-29

[2] Berton S, Yin H. Secure, Disjoint, Multipath Source Routing Protocol (SDMSR) for Mobile Ad Hoc Networks[C] // Proceedings of 5th Grid and Cooperative Computing (GCC' 2006). 2006: 387-394

[3] Mavropodi K. Secure Multipath Routing for Mobile Ad Hoc Networks[C] // Proceedings of Wireless On-demand Network Systems and Services (WONS'05). 2005: 89-96

[4] Zapata M G. Secure Ad hoc on-demand distance vector (SAODV) routing[Z]. Internet Draft, ddraft-guerrero-manet-saodv-06. txt, September 2006

[5] Kong Jiejun, Zerfos P, Luo Haiyun. Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks[C] // Proceedings of Ninth International Conference on Network Protocols. 2001: 251-260

[6] Marina M K, Das S R. On-demand Multipath Distance Vector Routing in Ad Hoc Networks[C] // Proceedings of the 1st International Conference for Network Protocols (ICNP). 2001: 14-23