基于进程代数的安全协议分析与验证*)

李梦君　李舟军　陈火旺
(国防科技大学计算机学院　长沙410073)

Analysis and Verification of Security Protocols Based on Process Algebra

LI Meng-Jun　LI Zhou-Jun　CHEN Huo-Wang
(School of Computer Science, National University of Defense Technology, Changsha 410073)

Abstract Process algebra is the theories for concurrency. Now the methods based on process algebra for the analysis and verification of security protocols becomes one of the important methods for the study of the security protocols. In the paper, we outline these methods, especially the symbolic trace analysis method the bisimulation method.

Keywords Process algebra, Cryptographic protocol , Analysis ,Verification

1　概述

安全协议用于在开放的互连网络上实现身份认证,分发会话密钥等安全性功能,开放的互连网络是一个复杂的运行环境,网络攻击者的恶意攻击和多个安全协议运行实例的并发运行,导致安全协议难以真正实现分发会话密钥,完成身份认证的意图。因此,对安全协议进行分析与验证成为一项十分必要的工作。

形式化分析与验证方法被人们普遍认可为安全协议分析与验证的一种有效方法。安全协议的形式化分析与验证建立在以下的前提假设之上:(1)安全协议使用的加密体系是安全的,协议攻击者无法攻破加密算法,一个网络攻击者只有拥有一个密文对应的解密密钥时,他才能得到正确的明文;(2)若$\{x\}_u=\{y\}_v$,则$x=y\wedge u=v$,即如果两个密文相同,则对应的明文和加密密钥都相同。因此,形式化分析与验证方法独立于具体的密码算法,研究安全协议是否满足安全性质。安全协议的安全性质主要指协议的保密性和认证性,保密性是指一个秘密不被非授权者获得,认证性是指非授权者不能伪装成合法参与者参与协议的运行。

进程代数是一类使用代数方法研究通信并发系统的理论的泛称,它包括CSP、CCS、π-演算[14]等,这些代数理论的共同特征为:1. 均使用通信,而不是共享存储,作为进程之间相互作用的基本手段;2. 在语法上,用一组算子作为进程构件,算子的语义用结构化操作语义方法定义,进程可看为标号转移系统;3. 把并发性归结为非确定性,将并发执行的进程的行为看作各单个进程的行为所有可能的交错合成,即并发的交叠语义。对安全协议进行分析和验证采用的进程代数理论主要为CSP、CCS、π-演算。

在安全协议的形式化分析与验证研究中,进程代数方法是一个新的研究方向,它借鉴了进程代数理论中许多有益的思想和方法(比如符号化和互模拟方法),能够利用进程代数已有的验证工具,并且成功地发现了一些安全协议原先没有被发现的攻击手段。基于进程代数的安全协议分析与验证,它将安全协议中每一个参与者的行为都描述为一个单独的进程,这些子进程使用通信作为进程之间相互作用的基本手段,安全协议被描述为由这些子进程组成的一个通信并发系统,多个安全协议运行实例的并发运行被描述为多个通信并发系统的并发运行;同时,安全协议应当满足的安全性质被描述为一些断言,或者被描述为一个理想的通信并发系统;对安全协议的分析与验证,就是分析与验证安全协议对应的并发系统是否满足安全性质断言,或者是否与安全性质对应的并发系统互模拟等价。

基于进程代数的安全协议分析与验证可以分为两类,一类研究安全协议对应的并发系统所有的踪迹是否满足安全性质断言,称之为踪迹分析方法;另一类研究安全协议对应的并发系统是否与安全性质对应的理想并发系统互模拟等价,称之为互模拟验证方法。本文的第2部分综述踪迹分析方法,第3部分综述互模拟验证方法,最后是全文总结,并给出了基于进程代数的安全协议分析与验证进一步的研究方向。

2　踪迹分析方法

基于踪迹分析方法的安全协议分析的主要研究工作包括M. Boreale[1~3]、S. Schneider[4,5]和G. Lowe[6,7]。其中,S. Schneider和G. Lowe的研究将协议攻击者的行为显式地用CSP进程描述。无穷多个安全协议运行实例的并发运行和协议攻击者生成无穷多的消息都将导致协议分析时出现无穷状态的问题,为了克服这个问题,他们的研究限制了并发运行的安全协议运行实例数目和协议攻击者生成消息的数目,采用FDR作为协议分析工具,他们发现了Needham-Schroeder协议和其它一些安全协议的攻击手段。M. Boreale的研究将协议攻击者的行为隐式地用扩展了加/解密原语的π-演算进程描述,采用符号化方法,不再限制协议攻击者生成消息的数目,得到了有穷多个安全协议运行实例并发运行时的分析方法,并实现了协议分析工具STA[3]。M. Boreale在文[1]中的研究主要针对对称密钥体系的加密协议,在文[2]中将研究对象扩展到非对称密钥体系的加密协议和包含Hash函数的加密协议。M. Boreale在文[1]中的研究最有代表性,我们以文[1]为主,描述基于踪迹方法的安全协议分析。

*)本课题得到国家自然科学基金(90104026,60073001)和863计划资助。

2.1 描述语言的语法与语义

名集合 $N: m, n, \cdots$	变量集合 $V: x, y, \cdots$
局部名集合 $LN: a, b, \cdots, h, k, \cdots$	环境名集合 $\epsilon N: \underline{a}, \underline{b}, \cdots, \underline{h}, \underline{k}, \cdots$
标号集合 $L: a, b, \cdots$	
消息集合 $M: M, N ::= u \mid \{M\}_u \mid \langle M, N\rangle$	
项集合 $Z: \eta, \zeta ::= u \mid \{\zeta\}_\eta \mid \langle \zeta, \eta\rangle$	
主体集合 $A: A, B ::=$	$\mid 0 \mid a(x).A \mid \bar{a}\langle\zeta\rangle.A$
	$\mid case\ \zeta\ of \{y\}_u\ in\ A \mid [\zeta=\eta]A$
	$\mid pair\ \zeta\ of\ \langle x, y\rangle\ in\ A \mid A \| B$

图1 语法

其中 N、L、V 是三个不相交的可数集，$N = LN \cup \epsilon N$。图1描述的语法针对对称密钥体系的加密协议。

消息集合和消息之间的推理关系 $\vdash$ 是由图2的消息推理规则生成的最小二元关系，其中 $S \subseteq_{fin} M$，$M \in M$，$\hat{V}$ 表示带标记变量集合(定义见下文)，消息推理规则(MVAR)表示任意消息集合可以推导出任意的带标记变量。

$$(AX)\frac{}{S \vdash M}\ M \in S \qquad (ENV)\frac{}{S \vdash \underline{a}}\ a \in \epsilon N$$

$$(MVAR)\frac{}{S \vdash \hat{x}}\ \hat{x} \in \hat{V} \qquad (PROJ_1)\frac{S \vdash \langle M, N\rangle}{S \vdash M}$$

$$(PROJ_2)\frac{S \vdash \langle M, N\rangle}{S \vdash N} \qquad (PAIR)\frac{S \vdash M\ S \vdash N}{S \vdash \langle M, N\rangle}$$

$$(DEC)\frac{S \vdash \{M\}_u\ S \vdash u}{S \vdash M} \qquad (ENC)\frac{S \vdash M\ S \vdash u}{S \vdash \{M\}_u}$$

图2 消息推理规则

踪迹 s 是 $\{a\langle M\rangle, \bar{a}\langle M\rangle\}^*$ $(a \in L, M \in M)$ 上的序列串，$v(s) = \phi$，并且对于每一个满足 $s = s_1 \cdot a\langle M\rangle \cdot s_2$ 的组合 $s_1, a\langle M\rangle, s_2$，皆有 $s_1 \vdash M$。其中 $v(s)$ 表示 s 中的变量集合，$s \vdash M$ 表示 $msg(s) \vdash M$，即 s 中的消息集合可推导出消息 M。进程是不包含自由变量的主体。格局 C 是踪迹 s 和进程 P 构成的二元组 $\langle s, P\rangle$，不包含环境名的格局称为初始格局。给定一个格局 $\langle s, P\rangle$ 和一个踪迹 s'，如果存在进程 P'，使得 $\langle s, P\rangle \rightarrow^* \langle s', P'\rangle$，则称格局 $\langle s, P\rangle$ 产生踪迹 s'，记作 $\langle s, P\rangle \searrow s'$。符号化踪迹 σ 是 $\{a\langle M\rangle, \bar{a}\langle M\rangle\}^*$ $(a \in L, M \in M)$ 上的序列串，σ 中不含有环境名，并且对于每一个满足 $\sigma = \sigma_1 \cdot \alpha \cdot \sigma_2$ 的组合 $\sigma_1, \alpha, \sigma_2$，如果 $x \in v(\alpha) - v(\sigma_1)$，则 α 是一个输入动作。符号化格局是由符号化踪迹 σ 和主体 A 组成的二元组 $\langle \sigma, A\rangle$，并且主体 A 中不含有环境名，$v(A) \subseteq v(\sigma)$。

下面给出描述语言的操作语义，基操作语义定义了进程之间的迁移关系，符号操作语义定义了主体之间的迁移关系。与 π-演算的迁移语义不同，描述语言的基操作语义定义在格局上，符号操作语义定义在符号化格局上。

基操作语义

(INP)	$\langle s, a(x) \cdot P\rangle \rightarrow \langle s \cdot a\langle M\rangle, p[[M/x]]\rangle\ s \vdash M, v(M) = \phi$
(OUT)	$\langle s, \bar{a}\langle M\rangle \cdot P\rangle \rightarrow \langle s \cdot \bar{a}\langle M\rangle, p\rangle$
(CASE)	$\langle s, case\{\zeta\}_k\ of\ \{y\}_k\ in\ P\rangle \rightarrow \langle s, P[\zeta/y]\rangle$
(MATCH)	$\langle s, [\zeta=\zeta]P\rangle \rightarrow \langle s, P\rangle$
(PAR)	$\dfrac{\langle s, P\rangle \rightarrow \langle s', P'\rangle}{\langle s, P \| Q\rangle \rightarrow \langle s', P' \| Q\rangle}$

符号操作语义

(INP)	$\langle \sigma, a(x) \cdot A\rangle \rightarrow \langle \sigma \cdot a(x), A\rangle$
(OUT)	$\langle \sigma, \bar{a}\langle M\rangle \cdot A\rangle \rightarrow \langle \sigma \cdot \bar{a}\langle M\rangle, A\rangle$
(CASE)	$\langle \sigma, case\{\zeta\}_v\ of\ \{x\}_u\ in\ A\rangle \rightarrow \langle \sigma\theta, A\theta\rangle\theta = mgu(\{\zeta\}_v, \{x\}_u)$
(SELECT)	$\langle \sigma, pair\ \zeta\ of\langle x, y\rangle in\ A\rangle \rightarrow \langle \sigma\theta, A\theta\rangle\theta = mgu(\zeta, \langle x, y\rangle)$
(MATCH)	$\langle \sigma, [\zeta=\eta]A\rangle \rightarrow \langle \sigma\theta, A\theta\rangle\theta = mgu(\zeta, \eta)$
(PAR)	$\dfrac{\langle \sigma, A\rangle \rightarrow \langle \sigma', A'\rangle}{\langle \sigma, A \| B\rangle \rightarrow \langle \sigma', A' \| B'\rangle} B' = B\theta$ 如果 $\sigma' = \sigma\theta$

图3 基操作语义与符号操作语义

符号操作语义中的 mgu 表示最广义合一化算子[1]。

2.2 安全性质断言与符号化踪迹分析

基于踪迹分析方法的安全协议分析用对应性断言描述安全性质，对应性断言形如 $\alpha \leftarrow \beta, \alpha, \beta \in \{a\langle M\rangle, \bar{a}\langle M\rangle\}^*, a \in N, M \in M$，并且 $v(\alpha) \subseteq v(\beta)$，对应性断言的直观意义为行为 α 总是先于行为 β 发生。安全协议的保密性和认证性都可以用对应性断言描述[1]。

设 σ 是一个符号化踪迹，ρ 是一个基变换，如果 $\sigma\rho$ 是一个踪迹，则称 ρ 满足 σ，并称 $\sigma\rho$ 是 σ 的一个解，σ 是协调的。设 C 是一个初始格局，s 是一个踪迹，则 $C \searrow s$ 当且仅当存在一个符号化踪迹 σ，使得 $C \searrow \sigma$ 并且 s 是 σ 的一个解。

使用进程描述语言后，安全协议对应的并发系统就可以看作符号化格局迁移系统，对安全协议的踪迹分析就转换为对符号化格局迁移产生的符号化踪迹进行分析，由于不协调的符号化踪迹不能实例化，对符号化踪迹的分析就可以缩小范围，只针对协调的符号化踪迹进行。通过定义精化关系可以得到符号化踪迹协调性的判定方法。

2.3 精化关系与符号化踪迹协调性的判定

将符号化踪迹中出现的变量分为两类，一类可以用环境生成的消息作实例化，称为带标记变量，带标记变量用 $\hat{x}$ 表示，一类不能用环境生成的消息作实例化，称为普通变量，普通变量的实例化需要通过匹配完成。包含带标记变量的符号化踪迹称为带标记符号化踪迹。设 σ 是一个带标记符号化踪迹，ρ 是一个基变换，如果 $\sigma\rho$ 是一个踪迹，并且对每一个带标记变量 $\hat{x}$，皆有 $(\sigma/\hat{x})\rho \vdash \rho(\hat{x})$，则称 ρ 满足 σ，并称 $\sigma\rho$ 是 σ 的一个解，σ 是协调的。其中 $\sigma/\hat{x}$ 表示 σ 中不包含 $\hat{x}$ 的最长前缀。带标记符号化踪迹中，有一类本身就是协调的，称为可解带标记符号化踪迹。设 σ 是一个带标记符号化踪迹，如果对每一个 $\sigma_1, a\langle M\rangle, \sigma_2$，使得 $\sigma = \sigma_1 \cdot a\langle M\rangle \cdot \sigma_2$，都有 $\sigma_1 \vdash M$，则称 σ 是可解带标记符号化踪迹。每一个可解带标记符号化踪迹都是可实例化的，基置换 $\rho: v(\sigma) \rightarrow \epsilon N$ 使 $\sigma\rho$ 是 σ 的一个解。

为了判定带标记符号化踪迹与消息之间的可推导关系，需要得到带标记符号化踪迹和消息中不能再分的原子消息集合。设 σ 是一个带标记符号化踪迹，则 σ 中不能再分的原子消息集合 $I(\sigma) = \{M; \sigma \vdash M, M \in LN \cup V$ 或者 $M = \{N\}_u$，若 $\sigma \nvdash u\}$。对于 $M \in M$，M 中不能再分的原子消息集合 $[M]_e$ 递归定义如下：

$$[u]_e = \{u\} - (\hat{V} \cup \epsilon N)$$
$$[(M, N)]_e = [M]_e \cup [N]_e$$
$$[\{M\}_u]_e = \{M\}_u \text{ 如果 } \sigma \nvdash u, \text{否则} [\{M\}_u]_e = [M]_e$$

图4 消息的分解

由上述定义，$\sigma \vdash M$ 当且仅当 $\{M\}_e \subseteq I(\sigma)$。消息分解为消息集

合与消息之间的推理关系提供了一个判定方法。

设 σ 是一个带标记符号化踪迹，$\sigma=\sigma'\cdot a\langle M\rangle\cdot\sigma''$，其中 σ' 是 σ 中为可解的最长前缀，设 $N\in[M]_{\mathcal{E}}-I(\sigma')$，精化关系 $\succ$ 就是由 REF_1 和 REF_2 两条规则生成的带标记符号化踪迹上的最小二元关系。精化关系定义的精化过程通过不断地把变量变为带标记变量和在项之间作匹配，把带标记符号化踪迹句逐步化为可解形式的带标记符号化踪迹。设 s 是一个踪迹，则 s 是 σ 的一个解当且仅当 s 是某个 $\sigma'\in SF(\sigma)$ 的解，其中 $SF(\sigma)=\{\sigma'\mid\sigma(\succ)^*\sigma'$ 并且 σ' 为可解形式$\}$。符号化踪迹 σ 是协调的当且仅当 $SF(\sigma)\neq\phi$。

$$REF_1\quad\frac{N\notin V, N\in_{\mathcal{E}} I(\sigma'), \tilde{y}=\{x\mid\hat{x}\in v(\sigma)\text{并且}(\sigma\theta)/\hat{x}\text{比}\sigma/\hat{x}\text{短}\}}{\sigma\succ\sigma\theta[\tilde{y}/\tilde{y}]}$$

$$REF_2\quad\frac{N=x\text{或者}N=\{N'\}_x}{\sigma\succ\sigma[\hat{x}/x]}$$

图5 精化规则

2.4 安全协议分析

踪迹分析方法对验证用对应性断言 $\alpha\leftarrow\beta$ 描述安全性质时，可以进一步减少需要分析的协调的符号化踪迹数目，只需分析 β 在其中出现的协调的符号化踪迹。设 σ 是一个符号化踪迹，$pr=\alpha\leftarrow\beta$ 是一个安全性质，$v(\alpha)\subseteq v(\beta)$ 并且 $v(\beta)\cap v(\sigma)=\phi$，则 $\sigma\vdash pr$ 当且仅当：对每一个 θ 使得 $\beta\in_{\mathcal{E}} act(\sigma)$，对每一个 $\sigma'\in SF(\sigma\theta)$，记 $\sigma'=\sigma\theta\theta'$，都有 $\alpha\theta\theta'$ 先于 $\beta\theta\theta'$ 在 σ' 中出现。

有了以上结论，就得到安全协议的符号化踪迹分析方法。设 P 是一个安全协议，C 是 P 对应的初始格局，计算 C 生成的所有的符号化踪迹 σ，验证每一条符号化踪迹 σ 是否都满足安全性质。若每一条符号化踪迹 σ 都满足安全性质，则安全协议 P 满足安全性质。如果存在一条可解形式的符号化踪迹不满足安全性质，则它的每一个实例化都是安全协议 P 的一个攻击手段。

3 互模拟验证方法

基于互模拟方法的安全协议验证的主要研究工作包括 Abadi，Gorden[8,9]，Michele Boreale[10,11] 和 Veronique Cortier[12]。Abadi 和 Gorden 最早提出使用互模拟方法验证安全协议[8]，并定义了一个协议描述语言 Spi 演算，给出了 Spi 演算的形式语义，定义了 Barbed 互模拟和 Barbed 等价的概念。Barbed 等价蕴涵测试等价，对安全协议进行验证，就只需证明安全协议经 Spi 演算描述得到的并发系统与安全性质所对应的理想并发系统 Barbed 等价。Abadi 和 Gorden 定义的 Barbed 等价要求两个要证明 Barbed 等价的进程与所有进程的并发合成是 Barbed 互模拟的，这使得 Barbed 等价的证明十分复杂。Michele Boreale 在文[10]中使用环境灵敏的标号迁移系统，定义了弱互模拟，证明了弱互模拟与 Barbed 等价的蕴涵关系，如果两个进程是弱互模拟的，则它们是 Barbed 等价的。弱互模拟的证明不需要穷举所有进程，因而简化了 Barbed 等价的证明。弱互模拟的证明支持组合推理，这使得安全协议的互模拟验证能够应用组合推理技术。Veronique Cortier 扩展了 Michele Boreale 的研究结果，使它不再局限于对称密钥体系的加密协议中，将它扩展到非对称密钥体系和包含 hash 函数的加密协议中。我们以文[10]为主，描述基于互模拟方法的安全协议验证。

3.1 描述语言的语法与语义

语法

名 N	$a,b,\cdots,h,k,\cdots,x,y,z\cdots$
消息 M	$M,N::=a\mid\langle M_1,M_2\rangle\mid\{M\}_k$
表达式 Z	$\eta,\zeta::=a\mid\{\eta\}_\zeta\mid dec_\eta(\zeta)\mid(\zeta_1,\zeta_2)\mid\pi_1(\zeta)\mid\pi_2(\zeta)$
公式 Φ	$\phi,\psi::=tt\mid name(\zeta)\mid[\zeta=\eta]let\ z=\zeta\ in\ \phi\mid\phi\wedge\psi\mid\neg\phi$
进程 P	$P,Q::=0\mid\eta(x).\ p\mid\overline{\eta}\zeta.\ P\mid P+Q\mid P\mid Q\mid(\nu a)P\mid!P\mid\phi P\mid$ $let\ z=\zeta\ in\ P$

图6 描述语言的语法

语义：

$$(INP)\ a(x).\ P\xrightarrow{a(M)}P[M/x]\qquad(OUT)\ \bar{a}M.\ P\xrightarrow{a(M)}P$$

$$(SUM)\ \frac{P\xrightarrow{\mu}P'}{P+Q\xrightarrow{\mu}P'}\qquad(IDE)\ \frac{P[\tilde{v}/\tilde{x}]\xrightarrow{\mu}P'}{A(\tilde{v})\xrightarrow{\mu}P'}\ A(\tilde{x}\Leftarrow P)$$

$$(PAR)\ \frac{P\xrightarrow{\mu}P'}{P\|Q\xrightarrow{\mu}P'\|Q}\qquad(COM)\ \frac{P\xrightarrow{(\nu\tilde{b})\bar{a}(M)}P'\ Q\xrightarrow{a(M)}Q'}{P\|Q\xrightarrow{\tau}(\nu\tilde{b})(P'\|Q')}$$

$$(RES)\ \frac{P\xrightarrow{\mu}P'}{(\nu c)P\xrightarrow{\mu}(\nu c)P'}\ c\notin n(\mu)\qquad(OPEN)\ \frac{P\xrightarrow{\mu}P'}{(\nu c)P\xrightarrow{\mu}P'}\ c\neq a,\ c\in n(M)-\tilde{b}$$

$$(GUARD)\ \frac{[[\phi]]=tt\ P\xrightarrow{\mu}P'}{\phi P\xrightarrow{\mu}P'}\qquad(LET)\ \frac{\zeta\neq\perp\ P[\zeta/z]\xrightarrow{\mu}P'}{let\ z=\zeta\ in\ P\xrightarrow{\mu}P'}$$

图7 描述语言的语义

进程与环境的交互语义

$$(E\text{-}OUT)\ \frac{P\xrightarrow{(\nu\tilde{b})a-(M)}P'\quad\widehat{\eta\sigma}=a}{\sigma\triangleright P\overset{(\nu\tilde{b})\bar{a}(M)}{\underset{\eta(x)}{\longmapsto}}\sigma[M/x]\triangleright P'}$$

$$(E\text{-}TAU)\ \frac{P\xrightarrow{\tau}P'}{\sigma\triangleright P\overset{\tau}{\longmapsto}\sigma\triangleright P'}$$

$$(E\text{-}INP)\ \frac{P\xrightarrow{a(M)}P'\quad\widehat{\eta\sigma}=aM=\widehat{\zeta\sigma}\quad\tilde{b}\overset{def}{=}(n(\zeta)-dom(\sigma))}{\sigma\triangleright P\overset{a(M)}{\underset{(\nu\tilde{b}\eta(\zeta))}{\longmapsto}}\sigma[\tilde{b}/\tilde{b}]\triangleright P'}$$

图8 进程与环境的交互语义

在语义中出现的 $\hat{\cdot}$ 和 $[[.]]$ 分别是表达式和公式的赋值函数。表达式赋值函数 $\hat{\cdot}:Z\to M\cup\{\perp\}$ 递归定义如下：

$\hat{a}=a$

$\widehat{\{\zeta_1\}_{\zeta_2}}=\{M\}_k$，如果 $\hat{\zeta}_1=M$，并且 $\hat{\zeta}_2=k\in N$，否则为 $\perp$

$\widehat{dec_{\zeta_2}(\zeta_1)}=M$，如果 $\hat{\zeta}_1=\{M\}_k$ 并且 $\hat{\zeta}_2=k\in N$，否则为 $\perp$

$\widehat{\langle\zeta_1,\zeta_2\rangle}=\langle M_1,M_1\rangle$，如果 $\hat{\zeta}_1=M_1$ 并且 $\hat{\zeta}_2=M_2$，否则为 $\perp$

对 $i=1,2$，$\widehat{\pi_i(\zeta)}=M_i$，如果 $\zeta=\langle M_1,M_2\rangle$，否则为 $\perp$

图9 表达式的赋值

公式的赋值函数 $[[.]]:\Phi\to\{tt,ff\}$ 在 Φ 上递归定义，其中 $[[let\ z=\zeta\ in\ \phi]]=[[\phi[\zeta/z]]]$ 如果 $\zeta\neq\perp$，否则为 ff。

3.2 Barbed 等价和安全性质的描述

进程上的对称二元关系 $S\subseteq P\times P$ 称为 Barbed 互模拟，如果对任意的 $(P,Q)\in S$ 蕴涵：

1. 对每一个 P'，如果 $P\xrightarrow{\tau}P'$，则存在 Q'，使得 $Q\Rightarrow Q'$，并且 $P'SQ'$。

2. 对每一个 a，如果 $P\downarrow a$ 则 $Q\Downarrow a$。

最大的 Barbed 互模拟关系称为 Barbed 互模拟的，记作 $\dot{\cong}$。两个进程 P 和 Q，如果对所有的进程 R，都有 $P\|R \dot{\cong} Q\|R$ 成立，则称进程 P 和 Q 为 Barbed 等价的，记作 $P \cong Q$。

在基于互模拟方法的安全协议验证中，安全性质是用进程之间的 Barbed 等价描述的。设 P 是待验证的进程，其中 M 是进程 P 的一个秘密，用 $P(M')$ 表示将 P 中的 M 置换为 M'，则保密性描述为对任意的 M'，都有 $P(M) \cong P(M')$。认证性验证非授权者不能伪装成合法参与者参与协议的运行，即验证协议参与者接收到的消息只能来自合法参与者。设 $Inst$ 是待验证的进程，$Inst_{spec}$ 是进程 $Inst$ 的理想描述，其中协议参与者接收到的消息都被实例化为合法的协议参与者发送的对应的消息，则认证性被描述为 $Inst \cong Inst_{spec}$。

3.3 环境灵敏的标号转移系统和弱互模拟

在 π-演算中，进程与它所处的环境在任何时刻都拥有相同的名集合[11]，在扩展了加/解密算子的 π-演算中，进程与它所处的环境并不总是拥有相同的名集合，一个加密消息只有当环境拥有对应的解密密钥时，环境才能获得明文，这就使得进程与它所处的环境在信息拥有上是不对称的。传统的标号转移系统适合于描述 π-演算，由于它没有刻画环境，因此不适合于描述扩展了加/解密算子的 π-演算。

环境灵敏的标号转移系统使用格局 $\sigma \triangleright P$ 作为状态，其中 P 为进程，σ 为置换，用于描述环境的知识。格局之间的迁移形如 $\sigma \triangleright P \overset{\mu}{\underset{\delta}{\mapsto}} \sigma' \triangleright P'$，其中 μ 表示进程动作，δ 表示与进程动作 μ 对应的环境动作，它们是输入/输出或是输出/输入动作对，表示进程与环境之间的交互，或者是 $\tau/-$对，表示此时进程与环境之间没有交互。

两个置换 σ 和 σ' 如果满足 $dom(\sigma)=dom(\sigma)'$，并且对每一个满足 $fn(\phi) \subseteq dom(\sigma)$ 的公式 ϕ，都有 $[[\phi\sigma]]=[[\phi\sigma']]$，则称它们是等价的，记作 $\sigma \sim \sigma'$。两个置换等价的定义中使用了全称量词，文[11]中给出了置换的特征刻画和两个置换等价的判定方法。两个格局 $(\sigma_1 \triangleright P, \sigma_2 \triangleright Q)$ 称为协调的，如果置换 $\sigma_1 \sim \sigma_2$。一个关系如果只包含协调的格局对，则称该关系是协调的。弱互模拟是格局上最大的对称二元协调关系，记作 $\approx$，对任意的 $(\sigma_1 \triangleright P, \sigma_2 \triangleright Q) \in \approx$，记为 $(\sigma_1, \sigma_2) \vdash P \approx Q$，如果 $\sigma_1 \triangleright P \overset{\mu}{\underset{\delta}{\mapsto}} \sigma_1' \triangleright P'$，则存在 μ', σ_2', Q'，使得 $\sigma_2 \triangleright Q \hat{\mu}, \overset{\ddot{\mu}}{\underset{\delta}{\Rightarrow}} \sigma_2' \triangleright Q'$，并且 $(\sigma_1', \sigma_2') \vdash P' \approx Q'$。在文[11]中证明了弱互模拟和 Barbed 等价之间的关系：对任意的进程 P 和 Q，$(e_V, e_V) \vdash P \approx Q$ 蕴涵 $P \cong Q$，其中 $V=fn(P,Q)$，e_V 为 V 上的恒等置换。证明两个进程 P 和 $QBarbed$ 等价，可以通过证明 $(e_V, e_V) \vdash P \approx Q$ 得到。

3.4 组合推理

组合推理是克服状态空间爆炸问题的一个有力武器，它将一个大系统的分析与验证问题分解为多个子系统的分析与验证问题以及这些子系统之间可组合性条件的验证问题。弱互模拟的可组合性条件与进程所处的环境密切相关，环境是用它包含的消息集合的解密闭包描述的。设 W 是一个消息集合，W 的解密闭包 $dc(W)$，是如下递归定义的消息集合：

(1) $W \subseteq dc(W)$

(2) 如果 $k \in dc(W)$，并且 $\{M\}_k \in dc(W)$，则 $M \in dc(W)$

(3) 如果 $\langle M_1, M_2 \rangle \subseteq dc(W)$，则 $M_1, M_2 \in dc(W)$

对每一个 s，如果 $\sigma \triangleright R \overset{'}{\underset{u}{\Rightarrow}} \sigma' \triangleright R' \overset{(\alpha\delta)\vec{a}(M)}{\underset{\psi(x)}{\Rightarrow}}$，都有 $M \in dc(\sigma)$，则称进程 R 是 σ-$safe$ 的。进程 R 是 σ-$safe$ 的，表示进程 R 不会增加环境 σ 的知识集。

弱互模拟的组合定理描述如下：设 $(\sigma_1, \sigma_2) \vdash Q_1 \approx Q_2$，设 $(\sigma_1, \sigma_2) \vdash R_1 \approx R_2$，并且对 $i=1,2$，Q_i 和 R_i 是 σ_i-$safe$ 的，则 $(\sigma_1, \sigma_2) \vdash Q_1 \| R_1 \approx Q_2 \| R_2$。弱互模拟的组合定理表明满足可组合性条件的两组进程 (Q_1, Q_2) 和 (R_1, R_2)，它们对应分量的并发合成仍然是弱互模拟的。弱互模拟的组合定理将进一步简化进程 Barbed 等价的证明。

小结 安全协议分析与验证的难点在于状态空间爆炸问题，安全协议运行在开放的互连网络上，网络攻击者的恶意攻击和多个安全协议运行实例的并发运行，使得安全协议分析与验证的过程中不可避免地出现了状态空间爆炸问题。基于进程代数的安全协议分析与验证是安全协议分析与验证最重要方法之一，它运用进程代数理论研究中已有的一些理论成果，比如符号化方法、互模拟等价等，成功地解决了一些安全协议的分析与验证问题。踪迹分析方法是一种可实现自动化分析的方法，现有的安全协议自动分析工具绝大多数是基于踪迹分析方法的，如 FDR 和 STA。互模拟验证方法实现自动化验证比较困难，但是它能较为有效地克服状态空间爆炸问题，开发基于互模拟验证方法验证安全协议的定理证明器需要进一步研究。

基于进程代数的安全协议分析与验证在以下几个方面需要进一步的研究：

(1) 对无穷多组安全协议运行实例并发运行情况下的安全性判定。

(2) 对存在攻击手段的安全协议进行校正。

(3) 将基于进程代数的安全协议分析与验证技术运用到对电子商务协议的分析与验证中。电子商务协议是电子商务实施的技术基础，电子商务协议应满足原子性，不可否认性，匿名性等性质[15]，需要对电子商务协议进行相应的分析与验证。

参 考 文 献

1 Boreale M. Symbolic trace analysis of cryptographic protocols. In: Proc. of ICALP'01, LNCS～2076, Springer, 2001
2 Boreale M, Buscemi M. A Framework for the Analysis of Security Protocols
3 Boreale M, Buscemi M. Experimenting with STA, a Tool for Automatic Analysis of Security Protocols
4 Schneider S. Verifying Authentication Protocols in CSP. IEEE Transactions on Software Engineering, 1998
5 Schneider S. Using CSP for protocol analysis: the Needham-Schroeder Public-Key Protocol: [Technical Report]. 1996
6 Lowe G. Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR. In: Proc. of TACAS'96, LNCS 1055, Springer-Verlag, 1996
7 Lowe G. A Hierarchy of Authentication Specifications. In: 10th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press, 1997
8 Abadi M, Gorden A D. A calculus for cryptographic protocols: The spi calculus. Information and Computation, 1999
9 Abadi M, Gorden A D. A Bisimulation Method for Cryptographic Protocols. Nordic Journal of Computing, 1998
10 Boreale M, Nicola R D, Pugliese R. Proof techniques for cryptographic processes. In Logic in computer science, 1999
11 Boreale M, Nicola R D, Pugliese R. Process Algebraic Analysis of Cryptographic Processes
12 Cortier V. Observational equivalence and trace equivalence in an extention of Spi-calculus. Application to cryptographic protocols analysis. Extended version. 2002
13 Milner R. Communication and Concurrency. Prentice-Hall International. 1989
14 Milner R, Parrow J, Walker D. A calculus of mobile processes, parts I and II. Information and Computation, 1992
15 周龙骧. 电子商务协议研究综述. 软件学报, 2001, 12(7)