第 33 卷 第 5 期 计 算 机 工 程 2007 年 3 月
Vol.33 No.5 Computer Engineering March 2007
·安全技术· 文章编号：1000—3428(2007)05—0112—02 文献标识码：A 中图分类号：TP393.08

基于逻辑编程的 EKE 协议分析

王全来[1,2]，韩继红[1]，王亚弟[1]
（1. 解放军信息工程大学电子技术学院，郑州 450004；2. 解放军防空兵指挥学院，郑州 450052）

摘 要：基于逻辑编程规则及 Spi 演算提出了一种验证密码协议安全性的新方法，利用该方法可以对密码协议的安全性质以程序化的方式进行验证。通过对 EKE 协议进行的分析，不但证明了协议已知的漏洞，而且发现了针对 EKE 协议的一个新的攻击——并行会话攻击。很好地验证了该新方法对密码协议的分析能力。

关键词：进程演算；逻辑编程；自动验证；密码协议

EKE Protocols Analysis Based on Logic Programming

WANG Quanlai[1,2], HAN Jihong[1], WANG Yadi[1]
(1. Institute of Electronic Technology, PLA University of Information Engineering, Zhengzhou 450004;
2. Air Defense Forces Command College of PLA, Zhengzhou 450052)

【Abstract】 Based on the Spi calculus and the logic programming rules, a new technique is presented to verify cryptographic protocols. The technique makes it possible to verify security of the protocols, in a fully automatic way. By analyzing the EKE protocol, it finds a new attack——parallel session attack, and this result demonstrates the analysis power of the technique for protocol security.

【Key words】 Process calculus; Logic programming; Automatic verification; Cryptographic protocols

密码协议的目标是在一个不安全的网络中安全地交换敏感数据，如电子商务中的信用卡号码。为防止攻击，密码协议利用密码技术将网络上的数据保护起来。同时密码协议的设计是非常困难的，容易产生错误，例如 Needham-Schroeder 公钥协议，在其发表 17 年后才由 G Lowe 在文献[1]中发现了它的一个漏洞。之所以这样，原因在于攻击只出现在那些试图破坏协议的恶意用户中，且在协议的测试和人工验证过程中，不会检测到这样的攻击，这就是密码协议验证受到更多重视的原因。

本文主要研究了一种基于逻辑编程的验证方法，该方法具有以下特征：(1)可以处理许多密码学运算，如对称密钥、公开密钥和 Diffie-Hellman 密钥协商模型；(2)对协议运行次数和消息大小没有任何限制；(3)自动验证，用户只需给出协议规范和所要验证的安全性质。

1 密码协议的形式化描述

Spi演算[2]的语法由项和进程组成，其中项表示进程中的名字、变量和数据集合，定义分别如下：

L,M::= 项标识
N 名字
X 变量
$\{L\}_{L_0}$ 对称加密
$\{|L|\}_{L_0}$ 非对称加密

P,Q::= 进程标识
out(c, L).P 输出
in(c, L).P 输入
P|Q 并行合成
(ν n)P 生成名字
case x of $\{L\}_{L_0}$ in P 对称解密
case x of $\{|L|\}_{L_0}$ in P 非对称解密

其中，名字表示现时和密钥。(ν n)P生成一个新名字n，并将n限定到进程P上。out(c, L).P输出L，然后作为P继续。in(c, L).P接收L，然后作为P继续。case x of $\{L\}_{L_0}$ in P表示用对称密钥L_0进行解密，case x of $\{|L|\}_{L_0}$ in P表示用非对称密钥L_0进行解密。

2 密码协议的验证方法

验证时，将协议表示为逻辑编程规则(horn 子句)集，其语法如图 1。其中项表示协议参与者之间交换的消息；变量表示任意的项；名字表示原子数据，如密钥和现时；函数用于建立项，如加密或 hash 函数；谓词表示这些消息的事实；规则表示如果所有的事实为真，则 F 为真。

M,N ::=		项
		变量
	a [M1, … , Mn]	名字
	f (M1, … , Mn)	函数
F ::=		事实
	P (M1, … , Mn)	谓词
R ::=		规则
	F1 … Fn → F	推论

图 1 协议表示的语法

协议验证方法如图 2，主要思想是基于 Spi 演算来研究新的验证方法。Spi 演算不仅具有加密和 hash 函数等运算，而且等价性提供了一个通用的表示密码学运算以及推理安全性的方法。在新验证方法中给出两个新的结构：构造器和析构器。构造器用于建立新的项，如加密 encrypt；析构器对项

作者简介：王全来(1970－)，男，博士生，主研方向：密码学，信息安全等；韩继红，副教授；王亚弟，教授、博导
收稿日期：2006-03-20 **E-mail：**wql_lai@126.com

进行操作，如解密 decrypt。析构器定义为简化关系，如对称密钥解密可定义为 decrypt(encrypt(x, y), y) → x，即用相同的密钥 y 解密密文 encrypt(x, y)，就可以得到明文 x。

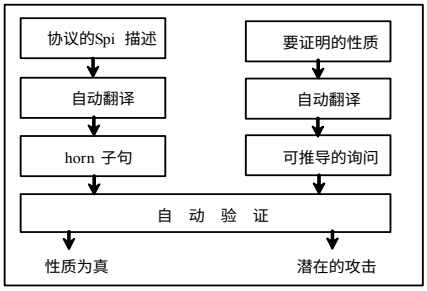

图 2 协议验证方法

协议的形式化表示随后被自动翻译成 horn 子句即逻辑程序。主要思想是使用谓词 attacker，attacker(M)表示攻击者可能具有的所有消息 M。如攻击者可以加解密，如果他所具有的密钥为下面两个 horn 子句时：

attacker(x) ∧ attacker(y) → attacker(encrypt(x, y))

attacker(encrypt(x, y)) ∧ attacker(y) → attacker(x)

当攻击者具有明文和密钥时，可以构建密文 encrypt(x, y)；当具有密文和密钥时，可以得到明文。协议的保密性由不可推导性得到：如果 attacker(M)是不可推导的，则攻击者不会得到 M，即 M 是秘密的；反之，协议是不安全的。因为对 attacker(M)的推导可以用于重构一个攻击，但这样的重构也有可能失败，如果系统已经发现了一个错误攻击。协议的认证性如下描述：如果参与者 A 相信他已经与 B 完成了一次会话，则 B 已经开始了与 A 的会话，而且 A、B 达成了协议所约定的参数，如会话密钥。下述判定给出协议的认证性：如果 A 执行事件 end(M)，则 B 已经执行了事件 begin(M)，begin 事件发生的次数至少是 end 事件发生的次数。

基于 Spi 演算和编程逻辑规则，利用 Prolog 语言设计和开发了一个自动验证系统。利用该系统，验证了很多协议，找到了已知的攻击或证实了协议的安全性。

3 EKE 协议分析

3.1 EKE 协议

加密密钥交换协议[3]以一种新奇的方法同时使用对称密钥和公开密钥体制来确保安全性，EKE协议利用共享密钥加密随机产生的公开密钥。非形式化描述如下：

消息 1 A→B：$\{K_A\}_P$

消息 2 B→A：$\{\{|K|\}_{K_A}\}_P$

消息 3 A→B：$\{N_A\}_K$

消息 4 B→A：$\{N_A, N_B\}_K$

消息 5 A→B：$\{N_B\}_K$

其中，P为共享口令，K为随机会话密钥，K_A为A的公开密钥。EKE可以用各种公开密钥算法实现，如：RSA，Diffie-Hellman等。基于RSA实现的EKE协议，由于K_A的一次随机性导致了一种攻击，即使K_A不是一次随机性的，EKE协议也是不安全的：攻击者可以利用公钥加密算法的弱点来搜寻P，即对RSA加密进行中间人攻击。Diffie-Hellman密钥交换体制[4]非常适合实现EKE协议。设 α 为一个群的生成元，群的阶大于 $2^{64} > 2^{|P|}$，则消息 1 中A随机选取 $x \in (0,2^{64})$ 并计算 $K_A = \alpha^x$；消息 2 中B随机选取 $y \in (0, 2^{64})$，并计算 $\{K\}_{K_A} = \alpha^y$；随后达成会话密钥 $K = \alpha^{xy}$。在这个实现中，群的生成元 α 可以公开达成：A在一个预协商中向B发送群的描述(包括群的生成元 α)。这里仅要求由 α 生成的群的阶大于 2^{64}，这是一个很小的数，即公钥体制中群的阶的最小值，该协议非常有效。较小的阶数导致易于计算离散对数，也导致易于解决可计算Diffie-Hellman问题。然而，如果没有 α^x、α^y 和 α^{xy}，可计算Diffie-Hellman问题的易于解决不会有助于找到口令：P在大小为 2^{64}的空间中仍是统计无关的。如果消息 3、消息 4 和消息 5 中的随机现时足够大，K在阶数大于 2^{64}的群中仍是统计无关的，离线字典攻击或者在线密钥猜测仍是很困难的。

3.2 EKE 协议的 Spi 演算描述

协议形式化描述如下：

```
Free c (定义通信通道)
Fun pk/1 (定义公开密钥函数)
Fun encrypt/2 (定义公钥加密函数)
Reduce decrypt(encrypt(x, pk(y)),y) = x (定义公钥解密函数)
Fun host/1 (定义主机名字函数)
Fun sencrypt/2 (定义对称密钥加密函数)
Reduce sdecrypt(sencrypt(x, y), y) = x (定义对称密钥解密函数)
Not pkA; not P (保密性假设)
private free secretA, secretB
query attacker: secretA; attacker: secretB (验证保密性)
query ev:endBparam(x) ⇒ ev:beginBparam(x) (验证认证性)
query
ev:endBfull(x1,x2,x3,x4,x5) ⇒ ev:beginBfull(x1,x2,x3,x4,x5)
query ev:endAparam(x) ⇒ ev:beginAparam(x)
query
ev:endAfull(x1,x2,x3,x4,x5) ⇒ ev:beginAfull(x1,x2,x3,x4,x5)
query evinj:endBparam(x) ⇒ evinj:beginBparam(x)
query
evinj:endBfull(x1,x2,x3,x4,x5) ⇒ evinj:beginBfull(x1,x2,x3,x4,x5)
query evinj:endAparam(x) ⇒ evinj:beginAparam(x)
query
evinj:endAfull(x1,x2,x3,x4,x5) ⇒ evinj:beginAfull(x1,x2,x3,x4,x5)
let processA = (定义发起者 A 进程)
  new secretA; in(c, pkA); in(c, hostX)
  event beginBparam(hostX)
  out(c, sencrypt(pkA, P); in(c, m)
  let (m1, =hostX) = sdecrypt(m, P) in
  let k=decrypt(m1, pkA) in
  new Na；out(c, encrypt((Na, k)); in(c, m2)
  let (=Na, NX2) = decrypt(m2, k) in
  event beginBfull(Na, hostA, hostX, pkA, NX2)
  out(c, sencrypt(NX2, k))
  if hostX = hostB then
  event endAparam(hostA)
  event endAfull(Na, hostA, hostX, pkA, NX2)
  out(c, sencrypt(secretA, k))
let processB = (定义响应者 B 进程)
  new secretB; in(c, m3)
  let pkY = decrypt(m3, P) in
  event beginAparam(hostY)
  new k; out(c, sencrypt(encrypt(k, pkY), P)
  in(c, m4)
  let (NY, =hostY) = sdecrypt(m4, k) in
  new Nb; event beginAfull(NY, hostY, hostB, pkY, Nb)
  out(c, sencrypt((NY, Nb), k)); in(c, m5)
  if Nb = sdecrypt(m5, k) then
```

（下转第 116 页）