维普资讯 http://www.cqvip.com

基于 Merkle 树的安全移动代理路由协议*)

柳 毅 姜正涛 王育民
(西安电子科技大学 ISN 国家重点实验室 西安710071)

摘 要 移动代理是一种软件程序,它漫游在计算机网络中,经过不同的主机代替用户执行一定的任务。它被认为在未来的电子商务中将起到很重要的作用,但安全问题一直是移动代理得到实际应用的一个很大障碍。本文利用二元 Merkle 树结合 Hash 函数给出了一个安全的基于 Merkle 树的移动代理路由协议。相比原有的嵌套签字加密协议,它显著地降低了移动代理主人的计算复杂度。

关键词 移动代理,嵌套加密,Merkle 树,计算复杂度

Mobile Agent Route Protection Based on Merkle Trees

LIU Yi JIANG Zheng-Tao WANG Yu-Min
(National Key Laboratory on ISN, Xidian University, Xi'an 710071)

Abstract Mobile agents are software programs that can autonomously migrate from host to host to fulfill their goals. It is believed to be playing an important role in future e-commerce systems, but the security problems have been an obstacle for mobile agents to be practical. Based on Merkle trees, a route protection of mobile agents is presented in this paper which reduces the computation cost of the agent owner compared with the protocol based on nested encryptions.

Keywords Mobile agents, Nested encryptions, Merkle trees, Computational cost

1 引言

随着 Internet 的不断扩容,网上信息的不断增加,用户对所需信息的查找、利用和处理也变得越来越困难。移动代理的出现为用户解决这方面的困难提供了便利的工具。移动代理是一种软件程序,它漫游在计算机网络中,经过不同的主机代替用户执行一定的任务。比如:收集电视节目单、购买商品、预定车票、监视股市行情等等。

电子商务更加扩展了移动代理的应用领域。移动代理可以代替用户进行谈判、购买性价比最好的商品、在线拍卖、旧货交易等等。但同时,由于代理的移动特点,也带来了许多新的安全问题。这些问题主要集中在两个方面:(1)防止恶意代理攻击网络主机;(2)防止恶意主机攻击移动代理。目前,针对这两个方面的大量安全问题已提出了许多的解决方案。一些已经得到了实际应用,但更多的方案还远未成熟[2]。这已被认为是移动代理还不能得到广泛应用的主要障碍之一。

对于移动代理的路由问题,文[4,6]提出安全的移动代理路由应该满足以下几条性质:

P1、防修改:恶意主机不能擅自修改代理路由信息,即使它们相互勾结也不能达到目的。

P2、信息的最小泄露:路由上的任意主机仅知道它的先驱者和后继者,路由上其他主机的信息对它来说是不可知的。

P3、可证实性:路由上的任意主机都能证实它是代理路由的一部分。

P4、代理来源确认性:路由上的每一主机都能确认,代理确实来自于它的先驱主机。

P5、具有时戳性:代理路由不会被恶意主机进行重放攻击。

文[6,7]给出了一系列移动代理路由的安全协议,其中具有嵌套加密的方案可以满足性质 P1～P5,但是给移动代理主人带来了很高的计算复杂性。文[11]利用 Merkle 树来构造公钥证书,采用 Merkle 树的主要优点是仅使用对树根结点的一次签名就可以对树上所有的叶结点独立地提供完整性认证[4]。本文利用 Merkle 树的这一优点,在文[4]的基础上给出了一个安全的基于 Merkle 树的移动代理路由协议。相比原有的采用嵌套加密的方案,此协议大大降低了移动代理主人的计算复杂度,同时满足路由安全性质 P1～P5。

2 原有的方案

文[6]给出了四个有关移动代理路由的安全协议。其中,采用嵌套加密的方案提供了较好的安全性。基本思想是:对主机的地址进行反复的嵌套签字和加密。设 PK_i 是第 i 个主机的公钥,$S_0(\cdot)$是用户 H_0(移动代理主人)的签字函数,H_i 代表第 i 个主机的 IP 地址,t 为时戳。整个路由 r 表示为:

$$r=E_{PK_1}[H_2,S_0(H_1,H_2,t,E_{PK_2}[\cdots]),E_{PK_2}[\cdots]]$$

其中 $E_{PK_i}[\cdots]=E_{PK_i}[H_{i+1},S_0(H_i,H_{i+1},t,E_{PK_{i+1}}[\cdots]),E_{PK_{i+1}}[\cdots]],i=1,\cdots,n-1$

$E_{PK_n}[\cdots]=E_{PK_n}[H_0,S^0(H_n,H_0,t)]$。代理漫游过程中,第一个主机 H_1 首先收到 r,用它的私钥对其进行解密得到 $S_0(H_1,H_2,t,E_{PK_2}[\cdots])$,$H_2$ 和 $E_{PK_2}[\cdots]$。通过用户的公钥,H_1 能够证实下一主机 H_2 和 $E_{PK_2}[\cdots]$ 是代理路由的一部分,并且能够证实 H_1 本身包含在路由中。时戳 t 可以防止恶意主机进行路由的重放攻击。除此以外,代理路由的其余部分用 H_2

*)基金项目:国家自然科学基金(No. 19931010)资助重点项目。柳 毅 博士研究生,研究方向为电子商务安全,网络安全。姜正涛 博士研究生,研究方向为电子商务安全,密码学。王育民 教授,博士生导师,长期从事信息论,编码与密码学的教学与科研工作。

维普资讯 http://www.cqvip.com

的公钥 PK_2 加密，H_1 不能获得有关它们的任何信息。H_1 把代理和路由 $r_1=E_{PK_2}[H_3,S_0(H_2,H_3,t,E_{PK_3}[\cdots]),E_{PK_3}[\cdots]]$ 发送给 H_2，H_2 重复以上的解密和验证过程，这样进行下去直到 H_n。最后，代理经由 H_n 传回 H_0。如果路由过程中 H_i 发现任何不相符的信息，H_i 会直接将代理传回 H_0 并报告相应的错误信息。

该协议满足路由安全性质 P1～P5，但这个方案的主要不足是移动代理主人在进行路由 r 的计算时，要进行多次的签字和加密处理。如果代理主人需要计算多个路由时，处理它们将需要大量的时间。

在公钥协议中，验证签字和加密所需的时间远远低于解密和签字所需的时间[13]。以 RSA 为例，若选择较低的公钥指数，设解密和签字所需时间为 t，那么验证签字和加密所需的时间可以只为 $t/100$，而计算一个 *Hash* 函数所需时间仅为 $t/10000$[4]。所以，代理主人计算路由 r 所需的时间主要由 r 中签字的次数来决定。而签字的次数等于代理路经主机的数目，因此计算 r 所需的时间随着路经主机数目的增加线性增长。当主机数目很大时，将会给代理主人带来巨大的计算负担。

本文给出的基于 Merkle 树的移动代理路由协议在满足路由安全性质 P1～P5的基础上，利用 Merkle 树结合 *Hash* 函数(这里假设 *Hash* 函数是单向抗碰撞的)，使得在计算路由 r 时仅需进行一次签字，这样就大大降低了代理主人计算路由所需的时间。

3 基于 Merkle 树的路由协议

本节利用二元 Merkle 树结合 *hash* 函数作为工具，来构造安全的移动代理路由协议。二元 Merkle 树的构造如下：树的每一个叶结点是一条指令加上该指令的 *hash* 值构成；每对兄弟结点的 *hash* 值组合到一起，再进行 *hash* 运算就得到它们的父结点；这个过程一直进行下去，直到得到树的根结点 *RV*。

移动代理路由协议如下：

(1) 初始化：移动代理主人 H_0 产生秘密随机数 $R_1,R_2,\cdots,R_n$，并计算 $h(R_1)$、$h(R_1\oplus R_2)\cdots h(R_1\oplus R_2\oplus\cdots\oplus R_n)$，其中 $h(\cdot)$ 为单向抗碰撞的 *hash* 函数，$\oplus$ 表示相异或。

移动代理主人计算 $U_1=E_{PK_1}(H_0,H_1,H_2,R_1,t)$

$U_i=E_{PK_i}(H_{i-1},H_i,H_{i+1},h(R_1\oplus\cdots\oplus R_{i-1}),R_i,t)i=2,3,\cdots,n-1$

以及 $U_n=E_{PK_n}(H_{n-1},H_n,H_0,h(R_1\oplus\cdots\oplus R_{n-1}),R_n,t)$

(2)构造二元 Merkle 树：把叶结点置为$(U_i,h(U_i))$ $i=1,2,\cdots,n$，代理主人按照二元 Merkle 树的构造方式构造二元 Merkle 树。

(3)签字：构造完二元 Merkle 树后，代理主人对树的根结点 *RV* 进行签字 $S_0(RV)$。

定义 U_i 的反向路径为这样一条路径：从包含$(U_i,h(U_i))$的叶结点向上直到根结点 *RV*，其中包括需要证实该路径所必须 *hash* 值(例如：沿着该路径兄弟结点的 *hash* 值)。反向路径的长度等于 *Merkle* 树的高度，并且仅随叶结点个数呈对数增长。

移动代理主人 H_0 把对根结点签名后的 Merkle 树发送给主机 H_1。

(4)移动代理的漫游：H_1 收到 H_0 传送过来的 *Merkle* 树，首先从中抽取 U_1 的反向路径，重新计算从叶结点$(U_1,h(U_1))$直到根结点 *RV* 一路上所有的 *hash* 值，检查通过 U_1 和它的反向路径计算得到的根结点值 RV_1 是否与由 H_0 签名的 *RV* 相同。若一切合法，H_1 解密 U_1，获得 H_0，H_1，$H2$ 和 R_1，得知代理是由 H_0 传送过来，并且它的后继主机是 H_2(否则，H_1 直接将代理传回 H_0 并报告相应的错误信息)。代理在 H_1 完成它的任务后，H_1 计算 $E_{PK_2}(R_1,t)$，连同 Merkle 树剩余部分(将证实 $U_2,U_3,\cdots,U_n$ 反向路径所不需要的结点删除)传送给 H_2。

对于 $i=2,3,\cdots,n$，H_i 收到 H_{i-1} 传送过来的剩余 Merkle 树以及 $E_{PK_i}(R_1\oplus\cdots\oplus R_{i-1},t)$，首先从剩余 Merkle 树中抽取 U_i 的反向路径，重新计算从叶结点$(U_i,h(U_i))$直到根结点 *RV* 一路上所有的 *hash* 值，检查通过 U_i 和它的反向路径计算得到的根结点值 RV_i 是否与由 H_0 签名的 *RV* 相同。若一切合法(否则，H_i 直接将代理传回 H_0 并报告相应的错误信息)，H_i 解密 U_i 和 $E_{PK_i}(R_1\oplus\cdots\oplus R_{i-1}t)$，获得 H_{i-1}，H_i，$H_{i+1}(i=n$ 时为 H_{n-1}，H_n，$H_0)$、$h(R_1\oplus\cdots\oplus R_{i-1})$ 和 $R_1\oplus\cdots\oplus R_{i-1}$。$H_i$ 知道代理是由 H_{i-1} 传送过来，并且它的后继主机是 $H_{n+1}(i=n$ 时为 $H_0)$，同时 H_i 检查 $R_1\oplus\cdots\oplus R_{i-1}$ 是否是 $h(R_1\oplus\cdots\oplus R_{i-1})$ 的原像。若一切合法(否则，H_i 直接将代理传回 H_0 并报告相应的错误信息)，代理在 H_i 完成它的任务后，H_i 计算 $E_{PK_{i+1}}(R_1\oplus\cdots\oplus R_i,t)(i=n$ 时则计算 $E_{PK_0}(R_1\oplus\cdots\oplus R_n,t))$，连同 Merkle 树剩余部分(将证实 $U_{i+1},\cdots,U_n$ 反向路径时所不需要的结点删除)传送给 $H_{i+1}(i=n$ 时则直接传送给 $H_0)$。

4 路由协议的安全性

下面，我们逐条检查路由安全性质 P1～P5：

P1. 恶意主机若想修改路由信息，例如修改 U_i 为 $U'_i\neq U_i$。若要不被发现，则要求通过 U'_i 的反向路径计算所得的根结点值与通过 U_i 的反向路径计算所得的根结点值相同。由于选择的 *hash* 函数是单向、抗碰撞的，因此这在计算上是不可行的。

而如果恶意主机 H_i 与 H_{j+1} 相勾结，企图删除 $H_{i+1}\sim H_j$ 的相关路由信息以及证实它们所需的 Merkle 树中的反路径。但由于没有 $H_{i+1}\sim H_j$ 的帮助，即使 H_i 与 H_{j+1} 相勾结仍然得不到 $h(R_0\oplus\cdots\oplus R_{j+1})$ 的原像 $R_0\oplus\cdots\oplus R_{j+1}$，很快 H_{j+2} 就会发现至少 H_{j+1} 参与修改了路由信息。

P2. 主机 H_i 用自己的私钥解密 U_i 可以得到 H_{i-1} 和 H_{i+1}，除此以外它不能得到路由信息中其他主机的信息，因为不能得到其他主机的个人私钥。

P3. 主机 H_i 用自己的私钥解密 U_i 可以得到三个主机的 *IP* 地址，其中包括它自己的 *IP* 地址 H_i。如果没有，则表明它不在代理主人 H_0 建立的路由信息中。

P4. 只要主机间具有相互的 IP 协议，就可以满足这条性质。一方面，主机 H_i 通过 IP 协议了解到代理所来自主机的 IP 地址 H'_{i-1}；另一方面，H_i 解密 U_i 后得到代理应该来自的主机 IP 地址 H_{i-1}。H_i 可以比较 H'_{i-1} 和 H_{i-1}，判断两者是否相等。

P5. U_i 中的时戳 t 可以防止恶意主机进行路由信息的重放攻击。

从以上可见，路由协议满足安全性质 P1～P5。

5 路由协议的计算复杂度

前面已经分析得出，计算路由信息所需的时间主要由信息中签字的次数来决定。原来的嵌套加密方案中，签字的次数等于代理路经主机的数目 n，因此，计算量将随着路经主机数目的增加呈线性增长。当主机数目很大时，将会给移动代理主人带来巨大的计算负担。

而本文给出的基于 Merkle 树的路由协议，只需要对树的

(下转第89页)

维普资讯 http://www.cqvip.com

(2)IDS 分析　IDS 提供了 IDS 预警、报文载荷和会话文件三种信息。IDS 预警的优点在于通过其特征匹配功能，能识别出具体的行为(如缓冲区溢出)。报文载荷提供了有关入侵行为的详细信息。会话文件非常有利于对 FTP、TELNET 和 IRC 等明文会话的迅速分析。

(3)系统日志　详细记录了入侵者是如何攻破系统和攻破系统后的行为，但它也是入侵者攻破系统后最先修改和删除的部分。所以系统日志应保存在三个地方：本地系统、远程 syslog 服务器和发往 syslog 服务器的全部信息的 IDS 报文载荷。通过三者的比较就可判别出入侵者是否对系统日志进行过修改。

(4)被动指纹[4](Passive Fingerprinting)　主动指纹(Active Fingerprinting)是通过使用工具(例如 Nump 或 Queso)主动检测远程主机以判断其操作系统，而被动指纹只是被动地嗅探远程主机的数据包来判断其操作系统。其可通过检查 TCP/IP 报文首部中的四个部分(TTL、窗口大小、DF 和 TOS)和 ICMP 回应请求来判断远程主机的操作系统、服务和应用。显然被动指纹不易被入侵者觉察，但是仍要防范入侵者的签名欺骗。

(5)取证[5](Forensic)　入侵者攻破系统以后，通常修改或删除了入侵的攻击程序、系统的日志文件、管理程序、配置文件甚至内核等，取证技术就是在被攻破系统的映像上，尽可能地恢复被入侵者破坏的数据，以重构入侵行为和提取入侵证据。The Coroner's Toolkit[6](TCT)包是取证过程中最常用的工具。

(6)统计(Statistics)　统计是指在陷阱网络长时间和多地域运行后，采集其所有的攻击数据，然后利用统计学原理对攻击数据进行分析，并对未来的攻击趋势作出预测。例如：可以统计某类探测特征与其攻击成功的时间间隔，以便将来此类特征重现时能够准确预测攻击将发起的时间。

(7)Sebek2[7]　入侵者目前也大量地使用了 SSH 或 3DES 等加密信道与被攻破的系统通信，这就给数据捕获工作带来了难度。不论是否使用加密信道，Sebek2是运行在陷阱机上直接捕获入侵者击键信息和上传文件的内核模块，其不断地将捕获的本地数据发向远程日志服务器，但这些数据包不会被入侵者的 Sniffer 捕获，因为所有陷阱机的内核都基于本地 MAC 地址做了相应的隐藏。

5　陷阱网络的风险

陷阱网络的确可以帮助研究入侵行为，但是同时它也包含着风险。尽管使用了防火墙、路由器和其他技术对入侵行为进行数据控制以降低风险，但是数据控制仍然可能会被入侵者躲过。另外，利用加密和反 IDS 技术也可躲过基于 Sniffer 的数据捕获。所以要真正地降低风险，就需要对陷阱网络进行持续的管理和维护。

结束语　陷阱网络的成功应归功于其简捷性：一个仿真实际系统的高度受控网络。所有进出陷阱网络的流量都会被捕获和分析，根据分析就可以了解入侵者的行为，对入侵者的行为了解的越多，对所面临威胁的觉悟和知识就会越高。但是入侵者的行为是不断变化的，这就需要陷阱网络的体系结构和计算机取证技术也要不断地动态变化。

参 考 文 献

1 Honeynet Project. Know Your Enemy: Honeynets [EB/OL]. http://project.honeynet.org,2003
2 Spitzne L. Honeypots: Tracking Hackers[EB/OL]. http://www.tracking-hackers.com
3 Dike J. User-Mode Linux [EB/OL]. http://user-mode-linux.sourceforge.net
4 Dittrich D. Computer Forensics[EB/OL]. http://staff.washington.edu/dittrich/forensics.html
5 Farmer D. Venema W. The Coroner's Toolkit (TCT)[EB/OL]. http://www.porcupine.org/forensics
6 Spitzer L. Passive Fingerprinting[EB/OL]. http://www.enter-,act.com/~lspitz/finger.htm
7 Honeynet Project. Know Your Enemy: Sebek2[EB/OL]. http://project.honeynet.org,2003

(上接第86页)

根结点进行一次签字，与路经主机的数目无关。虽然增加了一些 *hash* 函数的计算，但它们所占用的时间与签字相比是微不足道的。因此，该协议在保证了安全性的同时，大大降低了代理主人的计算量。

结论　本文给出了一个基于 Merkle 树的安全移动代理路由协议。它在满足路由安全性质 P1～P5的基础上，利用二元 Merkle 树结合 *Hash* 函数，使得在计算路由信息时仅需移动代理主人进行一次签字，相比原有的嵌套加密方案，该协议大大降低了代理主人计算路由信息所需的计算量。

该协议也可以很方便地推广到可变路由协议[9]。在可变路由协议中，对于二元 Merkle 树来说，不必改变结构；只是对随机数异或结果进行 Hash 运算时，在可变路由阶段，保证随机数异或结果相一致即可。

参 考 文 献

1 Das A, Gongxuan Y. A Secure Payment Protocol Using Mobile Agents in an Untrusted Host Environment[A]. ISEC 2001, LNCS 2040[C], Springer-Verlag Berlin Heidelberg, 2001. 33～41
2 Hohl F. Time limited blackbox security: Protecting mobile agents from malicious host[A]. Mobile agent and security, LNCS 1419 [C], Berlin: Springer-Verlag, 1998. 92～113
3 Perry M, Zhang Q. SITA: Protecting Trade Agents from Malicious Host[A]. Mobile Agents for Telecommunication Applications, LNCS 2164[C], Berlin: Springer-Verlag, 2001. 173～183
4 Domingo-Ferrer J. Mobile Agent Route Protection through Hash-Based Mechanisms[A]. LNCS 2247[C], Springer-Verlag, Berlin Heidelberg, 2001. 17～29
5 Belmon S G, Yee B S. Mobile Agent and Intellectual Property Protection[A]. Mobile Agents, LNCS 1477[C], Springer 1998. 172～183
6 Westhoff D, Schneider M, Unger C, Kaderali F. Methods for Protecting a Mobile Agent's Route[A], ISW'99 LNCS 1729[C], Springer-Verlag, Berlin Heidelberg, 1999. 57～71
7 Westhoff D, Schneider M, Unger C, Kaderali F. Protecting a Mobile Agents Route Against Collusions[A]. LNCS 1758[C], Springer-Verlag Berlin Heidelberg, 2000. 215～225
8 Sander T, Tschudin C F. Protecting Mobile agents Against Malicious Hosts[J]. Mobile agent and security, LNCS 1419[C], Berlin: Springer-Verlag, 1998. 44～60
9 Mir J, Borrell J. Protecting General Flexible Itineraries of Mobile Agents[A]. ICICS 2001, LNCS 2288[C], Springer-Verlag, Berlin Heidelberg, 2002. 382～396
10 Sander T, Tschudin C. Protecting Mobile Agents Against Malicious Hosts[A]. In Mobile Agents and Security[19], pp. 137～153
11 Gassko I, Gemmell P S, MacKenzie P. Efficient and fresh certification[A]. LNCS 1751, Berlin: Springer-Verlag, 2000. 342～353
12 Karnik G, Asokan N, Gülcü C. Protecting the Computation Results of Free-Roaming Agents[A]. LNCS 1477, Springer-Verlag, 1998. 194～207
13 王育民，刘建伟. 通信网的安全——理论与技术[M]. 西安：西安电子科技大学出版社，1999