基于 Merkle 树的安全移动代理路由协议及其推广

柳 毅, 姜正涛, 王育民
(西安电子科技大学综合业务网国家重点实验室, 陕西西安 710071)

摘 要: 本文对 Domingo J. 等人提出的一个高效的移动代理路由协议进行了分析, 指出其具有一个很严重的安全缺陷: 不能抵制路由主机间的共谋攻击. 在此基础上, 利用 hash 函数, 提出了一个基于 Merkle 树的安全移动代理路由协议, 并分析了其安全性、计算复杂度以及信息传输量. 结果表明该方案不仅弥补了原有协议的缺陷, 而且保持了原协议高效的特点. 最后, 将该方案其推广至动态路由.

关键词: 移动代理; Merkle 树; 计算复杂度; 动态路由
中图分类号: TN915.08 文献标识码: A 文章编号: 0372-2112 (2005) 07-1250-04

Mobile Agent Route Protection Based on Merkle Trees and Its Extension

LIU Yi, JIANG Zheng-tao, WANG Yu-min
(*National Key Laboratory on Integrated Server Networks, Xidian University, Xi'an, Shaanxi 710071, China*)

Abstract: The characteristics of mobile agents make them ideal for electronic commerce applications in open networks. Security is a fundamental precondition for the acceptance of mobile agent systems. In this paper, we analyze the effective route protection based on mobile agents given by Domingo J. and point that the protocol has security threat: it is not against collusion of malicious hosts. After that, using hash function, a secure itinerary protection of mobile agents based on Merkle trees is presented. Its and computational cost are discussed in detail. The results show that our method not only satisfies all security properties but also keeps the characters of the protocol existed. Finally, Our idea is extended to protect flexible itineraries too.

Key words: mobile agents; Merkle trees; computational cost; flexible itineraries

1 引言

移动代理技术由于其自身所具有的优异特性, 已经在很多新兴领域得到了的应用. 但是安全问题一直是移动代理技术得到更广泛应用的一个限制因素. 移动代理技术的主要特点就在于移动代理可以离开代理主人, 按照主人的意愿在网络中漫游, 而不需要与主人保持长期联系, 从而避免大量网络数据的传送. 这样, 如何使移动代理按照代理主人设定的路由信息漫游, 保证路由信息不会被非法主机恶意修改也就自然而然的成为了移动代理技术面临的主要安全问题之一.

对此, 有很多路由方案被提了出来[1~4]. 其中, Domingo J. 等首次将 Merkle 树引入到移动代理路由协议中, 利用 Merkle 树的一次签名特点来构造高效的移动代理路由协议[1]. 但是我们通过分析可以看出, 虽然该协议在计算复杂度和网络信息传输量方面具有较好的性能, 可是它存在一个很严重的安全隐患: 不能抵制主机间的共谋攻击.

本文在对该协议进行详细分析的基础上, 结合 hash 函数提出了一个安全的基于 Merkle 树的移动代理路由协议, 不仅弥补了原有协议的安全缺陷, 而且保持了原有协议在计算复杂度和网络信息传输量方面的良好性能, 并且将该思想推广至动态路由[4].

2 Merkle 树

Merkle 树是指这样一个树结构: 树的每一个叶结点是一条指令加上该指令的 hash 值构成; 每个父结点下面的所有子结点的 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, 直至得到树的根结点.

Merkle 树的主要优点是仅需通过对树根结点的一次签名运算就可以对树上所有的叶结点独立的提供完整性认证. 文[5, 6]都曾利用 Merkle 树的这一特点来构造公钥证书.

移动代理路由协议中采用 2- 3 元 Merkle 树, 即 Merkle 树的每一父结点下包含 2 个或 3 个子结点. 采用 2- 3 元 Merkle 树的原因是为了保证: 对于任意数目的路由主机 $n > 1$ ($n = 1$ 时, 即只有一个路由主机的情况比较简单, 这里不予考虑), 所有主机的路由信息都能包含在 Merkle 树的叶结点中. 这是由于对于任意正整数 $n > 1$, 存在非负整数 k 和 l 使得: $2k + 3l = n$. 不妨如此选取 k 和 l: 当 n 为偶数时, $k = n/2$, $l = 0$; 当 n 为奇数时, $k = (n-3)/2$, $l = 1$.

收稿日期: 2003-09-29; 修回日期: 2005-03-04
基金项目: 国家自然科学基金重点资助项目(No. 19931010)

3　原有的基于 Merkle 树的移动代理路由协议

3.1　符号标记

H_0: 移动代理主人(用户);

H_i: 移动代理路由经过的第 i 个主机;

PK_i: H_i 的公钥;

$S_i(m)$: 用 H_i 的私钥对消息 m 进行签字;

$E_{PK_i}(m)$: 用 H_i 的公钥对消息 m 进行加密;

$h(\cdot)$: 单向抗碰撞的安全 hash 函数;

$\oplus$: 异或运算;

t: 时戳.

3.2　基于 Merkle 树的路由协议

设移动代理需要路经 n 个主机, 用户 H_0 的身份对路由主机是预先已知的.

(1) 路由计算

H_0 计算 $U_i = E_{PK_i}(H_{i-1}, H_i, H_{i+1}, t)$　$i = 1, 2, \cdots, n-1$ 以及 $U_n = E_{PK_n}(H_{n-1}, H_n, H_0, t)$.

(2) 计算 Merkle 树

将 $(U_i, h(U_i))$ 作为叶结点, H_0 按照 Merkle 树的构造方法(见第 2 节)计算 2-3 元 Merkle 树.

(3) 签字

H_0 对 Merkle 树的根结点 RV 进行签字 $S_0(RV)$.

定义 Merkle 树的反向路径为这样一条路径: 从叶结点 $(U_i, h(U_i))$ 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). 显然, 反向路径的长度等于 Merkle 树的高度.

(4) 移动代理的漫游

H_0 把对根结点签名后的 Merkle 树发送给主机 H_1, 开始移动代理的漫游. H_i ($i = 1, 2, \cdots, n$) 收到 H_{i-1} 传送过来的 Merkle 树后, 抽取叶结点 $(U_i, h(U_i))$ 的反向路径, 证实反向路径上的所有 hash 值, 然后通过 U_i 和它的反向路径计算得到根结点值 RV_i, 检查它是否与由 H_0 签名的 RV 相同(否则, H_0 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, 获得 H_{i-1}, H_i, H_{i+1} ($i = n$ 时为 H_{n-1}, H_n, H_0). 移动代理在 H_i 完成它的任务后, H_i 将 $U_{i+1}, \cdots, U_n$ 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}. 最终, 移动代理由 H_n 传回 H_0.

3.3　安全性分析

移动代理的安全路由协议必须满足以下几条性质[1,2]:

$S1$、防修改: 恶意主机不能擅自修改代理路由信息, 即使它们共谋也不能达到目的.

$S2$、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.

$S3$、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.

$S4$、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.

$S5$、具有时戳性: 代理路由不会被恶意主机进行重放攻击.

下面我们对协议安全性进行分析:

$S2$: 主机 H_i 解密 U_i 后可以得到 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私钥.

$S3$: 主机 H_i 解密 U_i 后可以得到三个主机的地址, 其中应该包括它自己的地址 H_i. 如果没有, 则它表明不在代理主人 H_0 建立的路由信息中.

$S4$: 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.

$S5$: 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.

但是我们可以指出, 该协议不能抵制路由主机通过共谋而发动的修改路由信息攻击, 即协议并不满足安全性质 $S1$. 分析如下:

如果主机 H_i 与 H_{j+1} ($i < j$) 共谋, 希望使得不经过路由主机 $H_{i+1} \sim H_j$, 直接由 H_i 将移动代理传送给 H_{j+1}, 然后继续传送给 H_{j+2} 而不会被其他主机发现. 主机 H_i 与 H_{j+1} 只需保留路由信息 $U_{i+2}, \cdots, U_n$ 以及证实这些路由信息所需的相应反向路径, 而将其他信息删除. 这样, 当 H_{j+1} 将所剩余的信息传送给 H_{j+2} 时, 即使 H_{j+2} 或者其后的主机正确执行协议也仍然不能发现移动代理并没有经过路由主机 $H_{i+1} \sim H_j$.

4　改进方案

我们在原有方案的基础上, 给出一个安全基于 Merkle 树的移动代理路由协议, 在保持原有方案特点的同时, 弥补了其缺陷.

4.1　改进的路由协议

(1) 初始化

H_0 产生安全随机数 $R_1, R_2, \cdots, R_n$, 并计算.

$h(R_1)$、$h(R_1 \oplus R_2) \cdots h(R_1 \oplus R_2 \oplus \cdots \oplus R_n)$.

然后, H_0 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 2-3 元 Merkle 树

把叶结点置为 $(U_i, h(U_i))$, $i = 1, 2, \cdots, n, H_0$ 按照 2-3 元 Merkle 树的构造方式构造 2-3 元 Merkle 树.

(3) 签字

构造完 Merkle 树后, H_0 对树的根结点 RV 进行签字 $S_0(RV)$.

(4) 移动代理的漫游

H_0 把对根结点签名后的 Merkle 树发送给主机 H_1, 开始移动代理的漫游. H_1 收到 H_0 传送过来的 Merkle 树后, 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, H_2 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 $E_{PK_2}(S_2(R_1, t))$, 连同 Merkle 树剩余部分(将证实 U_2, U_3, …, 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 $E_{PK_i}(S_{i-1}(R_1 \oplus \cdots \oplus R_{i-1}, t))$ 后, 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, H_i 解密并验证 $E_{PK_i}(S_{i-1}(R_1 \oplus \cdots \oplus R_{i-1}, t))$ 得到 $R_1 \oplus \cdots \oplus R_{i-1}$, 然后解密 U_i, 获得 H_{i-1}, H_i, H_{i+1}($i=n$ 时为 H_{n+1}, H_n, H_0)和 $h(R_1 \oplus \cdots \oplus R_{i-1})$. 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, 代理在 H_i 完成它的任务后, H_i 计算 $E_{PK_{i+1}}(S_i(R_i \oplus \cdots \oplus R_i, t))$($i=n$ 时则计算 $E_{PK_0}(S_n(R_1 \oplus \cdots \oplus R_n, t))$), 连同 Merkle 树剩余部分(将证实 U_{i+1}, …, 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).

4.2 改进方案的安全性分析

由于改进方案继承了原方案的特点, 类似分析可以得到, 它满足安全性质 $S2$~$S5$. 下面我们重点分析协议的安全性质 $S1$.

恶意主机如果想修改路由信息, 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} 共谋, 企图删除 H_{i+1}~H_j 的相关路由信息. 但由于没有 H_{i+1}~H_j 的帮助, H_i 与 H_{j+1} 就无法得知 R_{i+1}~R_j. 这样, 即使 H_j 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.

因此, 改进方案满足路由安全性质 $S1$~$S5$.

4.3 计算复杂度

由于计算 hash 函数所需的计算复杂度与计算公钥体制下加密、签名相比微不足道. 所以, 这里主要考虑公钥体制下计算复杂度.

(1)用户方面: 从协议中可以看到, H_0 计算路由 r 时加密处理的总计算复杂度为 n, 而由于采用了 Merkle 树的特点, 需要进行签字处理的总计算复杂度仅为 1.

(2)路由主机方面: 在移动代理漫游过程中, 每一路由主机 H_i($i=1, 2, \cdots, n$)处理路由信息时, 需要进行的两次基本解密、两次验证签字、一次签字以及一次加密, 与主机在路由中的位置无关. 这里假设采用的公钥体制(如 RSA)中, 加密和验证签字所需的计算量相同, 而解密和签字所需的计算量相同. 因此, 路由主机总的计算复杂度为: 签字(解密)为 $3n$, 加密(验证签字)为 $3n$.

4.4 信息传输量

在网络信息传输量方面, 传输的信息除了包括路由信息外, 还要包含 Merkle 树上的所有结点上的 hash 函数信息.

为了讨论方便, 假设公钥体制取模长度为 1024 比特的 RSA, hash 函数取输出为 160 比特的 SHA-1. 则用户发出的移动代理路由信息长度为: $1024\times(n-1)+160\times(2n-1)$.

4.5 不同路由协议之间的性能比较

下面我们通过列表方式总结不同的安全路由协议性能.

表 1 不同路由协议之间性能比较

计算复杂度及信息传输量 / 路由方案	用户		路由主机		网络信息传输量(比特)
	加密(验证签字)	签字(解密)	加密(验证签字)	签字(解密)	
基于嵌套加密嵌套签字的方案[3]	$\frac{n(n+1)}{2}$	$\frac{n(n+1)}{2}$	$\frac{n(n+1)}{2}$	$\frac{n(n+1)}{2}$	$1024n$
基于嵌套加密基本签字的方案[4]	$\frac{n(n+1)}{2}$	n	n	$\frac{n(n+1)}{2}$	$1024n$
本文基于 Merkle 数的方案	n	1	$3n$	$3n$	$1024(n+1)+160(2n-1)$

4.6 协议的推广

2002 年, Joan Mir 和 Joan Borrell 第一次提出了移动代理动态路由的概念[4]. 动态路由: 设移动代理将路由经过的主机为 $H_0, H_1, \cdots, H_i, \langle(H_{i+1}, \cdots, H_{i+m1}), \cdots, (H_{i+m1+\cdots+m(k-1)+1}, \cdots, H_{i+m1+\cdots+mk})\rangle, H_{i+m1+\cdots+mk+1}, \cdots, H_n, \langle(H_{i+1}, \cdots, H_{i+m1}), \cdots, (H_{i+m1+\cdots+m(k-1)+1}, \cdots, H_{i+m1+\cdots+mk})\rangle$ 中每个"()"内包含的主机表示移动代理路由中的一个支路主机组, 移动代理经过"< >"中主机时, 只需要选择其中的一个支路主机组经过. 代理在从 H_i 发出时将进行路由支路的选择. 当选定一条支路后, 代理又将按照一定的主机顺序进行漫游. 如下图所示:

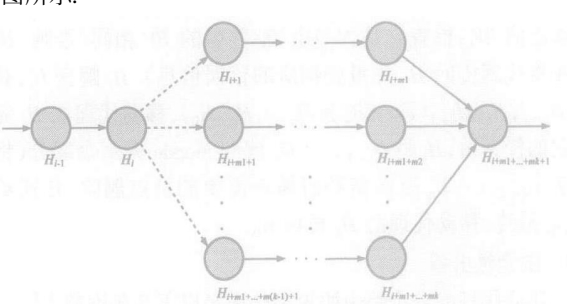

图 1 动态路由

图中实线表示移动代理必经的路由, 虚线表示代理可选择的路由. 从图 1 中可看出, 代理在从 H_i 发出时将进行路由支路的选择. 当选定一条支路后, 代理又将按照一定的主机顺序进行漫游.

基于 Merkle 树的移动代理动态路由协议只是在协议第一步初始化时有所不同, 下面给出协议初始化过程.

动态路由协议初始化:

1、用户 H_0 产生安全随机数 $R_0, R_1, \cdots, R_n$, 其中要求满足:

$R_{i+1} \oplus \cdots \oplus R_{i+m1} = R_{i+m1+1} \oplus \cdots \oplus R_{i+m2} = \cdots = R_{i+m1+\cdots+m(k-1)+1} \oplus \cdots \oplus R_{i+m1+\cdots+mk} = R_{动}$. 然后计算

$h(R_0)$, $h(R_0 \oplus R_1)$, …, $h(R_0 \oplus R_1 \oplus \cdots \oplus R_i)$,

$h(R_0 \oplus \cdots \oplus R_i \oplus R_{i+1})$, …, $h(R_0 \oplus \cdots \oplus R_i \oplus R_{i+1} \oplus \cdots \oplus R_{i+m1})$,

$h(R_0 \oplus \cdots \oplus R_i \oplus R_{i+m1+1})$, …, $h(R_0 \oplus \cdots \oplus R_i \oplus R_{i+m1+1} \oplus \cdots \oplus R_{i+m1+m2})$,

$h(R_0 \oplus \cdots \oplus R_i \oplus R_{i+m1+\cdots+m(l-1)+1})$, …, $h(R_0 \oplus \cdots \oplus R_i \oplus R_{i+m1+\cdots+m(l-1)+1} \oplus \cdots \oplus R_{i+m1+\cdots+ml})$, $l = 3, \cdots, k$.

$h(R_0 \oplus \cdots \oplus R_i \oplus R_{动} \oplus R_{i+m1+\cdots+mk+1})$, …, $h(R_0 \oplus \cdots \oplus R_i \oplus R_{动} \oplus R_{i+m1+\cdots+mk+1} \oplus \cdots \oplus R_n)$.

2、H_0 计算 $U_1 = E_{PK_i}(H_0, H_1, H_2, R_1, t)$

$U_l = E_{PK_l}(H_{l-1}, H_l, H_{l+1}, h(R_1 \oplus \cdots \oplus R_{l-1}), R_l, t)$

$l = 2, 3, \cdots, i-1$;

$U_i = E_{PK_i}(H_{i-1}, H_i, \langle H_{i+1}, H_{i+m1+1}, \cdots, H_{i+m1+\cdots+m(k-1)+1} \rangle, h(R_1 \oplus \cdots \oplus R_{i-1}), R_i, t)$;

$U_{i+k+1} = E_{PK_{i+k+1}}(H_i, H_{i+k+1}, H_{i+k+2}, h(R_1 \oplus \cdots \oplus R_i), R_{i+k+1}, t)$, $k = 0, m_1, \cdots, m_{(k-1)}$;

$U_{i+k+q} = E_{PK_{i+k+q}}(H_{i+k+q-1}, H_{i+k+q}, H_{i+k+q+1}, h(R_1 \oplus \cdots \oplus R_i \oplus R_{i+k+1} \oplus \cdots \oplus R_{i+k+q+1}), R_{i+k+q}, t)$,

当 $k = 0$ 时, $q = 2, \cdots, m_1 - 1$

当 $k = m_1$ 时, $q = 2, \cdots, m_2 - 1$

⋮

当 $k = m_{(k-1)}$ 时, $q = 2, \cdots, m_k - 1$

;

当 $i+k+q = i+m_1, i+m_1+m_2, \cdots, i+m_1+\cdots+m_k$ 时:

$U_{i+k+q} = E_{PK_{i+k+q}}(H_{i+k+q-1}, H_{i+k+q}, H_{i+m1+\cdots+mk+1}, h(R_1 \oplus \cdots \oplus R_i \oplus R_{i+k+1} \oplus \cdots \oplus R_{i+k+q-1}), R_{i+k+q}, t)$;

以及

$U_{i+m1+\cdots+mk+1} = E_{PK_{i+m1+\cdots+mk+1}}(H_{i+p}, H_{i+m1+\cdots+mk+1}, H_{i+m1+\cdots+mk+2}, h(R_1 \oplus \cdots \oplus R_i \oplus R_{动}), R_{i+m1+\cdots+mk+1}, t)$,

$p = m_1, m_1+m_2, \cdots, m_1+K+m_k$;

$U_j = E_{PK_{i+m1+\cdots+mk+1}}(H_{j-1}, H_j, H_{j+1}, h(R_1 \oplus \cdots \oplus R_i \oplus R_{动} \oplus R_{i+m1+\cdots+mk+1} \oplus \cdots \oplus R_{j-1}), R_j, t)$,

$j = i+m_1+\cdots+m_k+2, \cdots, n-1$;

$U_n = E_{PK_n}(H_{n-1}, H_n, H_0, h(R_1 \oplus \cdots \oplus R_i \oplus R_{动} \oplus R_{i+m1+\cdots+mk+1} \oplus \cdots \oplus R_{n-1}), R_n, t)$.

协议在构造 Merkle 树阶段同前文类似, 移动代理漫游时, 在 H_i 将进行路由支路的选择, 然后处理路由信息的步骤与前文类似, 限于篇幅, 此处就不详细讨论了.

在移动代理的动态路由协议中, 不论选择哪条路由支路, 因为存在等式 $R_{i+1} \oplus \cdots \oplus R_{i+m1} = R_{i+m1+1} \oplus R_{i+m2} = \cdots = R_{i+m1+\cdots+m(k-1)+1} \oplus \cdots \oplus R_{i+m1+\cdots+mk} = R_{动}$, 所以, $H_{i+m1+\cdots+mk+1}$都可以通过验证 $R_0 \oplus \cdots \oplus R_i \oplus R_{动}$ 来证实移动代理是否经过了正确的路由.

因此, 类似于前面协议安全性的分析, 可以很容易的得出动态路由协议满足路由安全性质 $S1 \sim S5$.

5　结论

本文对 Domingo J. 等提出的一个高效移动代理路由协议进行了安全分析, 指出了该协议所存在的安全缺陷: 不能发现路由主机的共谋攻击. 在此基础上, 本文利用 hash 函数提出了改进方案, 在保留原有协议特点的同时, 改进方案弥补了它们的安全缺陷. 详细分析了协议的安全性、计算复杂度和网络信息传输量, 并将协议推广至动态路由中.

参考文献:

[1] Josep Domingor Ferrer. Mobile agent route protection through hashr based mechanisms [A]. LNCS 2247 [C]. Berlin: Springer-Verlag, 2001. 17-29.

[2] D Westhoff, M Schneider, C Unger, F Kaderali. Methods for protecting a mobile agent's route [A]. ISW'99 LNCS 1729 [C]. Berlin: Springer-Verlag, 1999. 57-71.

[3] D Westhoff, M Schneider, C Unger, F Kaderali. Protecting a mobile agent's route against collusions [A]. LNCS 1758 [C]. Berlin: Springer-Verlag, 2000. 215-225.

[4] J Mir, J Borrell. Protecting general flexible itineraries of mobile agents [A]. ICICS 2001, LNCS 2288 [C]. Berlin: Springer-Verlag, 2002. 382-396.

[5] I Gassko, P S Gemmell, P MacKenzie. Efficient and fresh certification [A]. LNCS 1751, Berlin: Springer-Verlag, 2000. 342-353.

[6] Josep Domingor Ferrer, M Alba, F Sebe. Asynchronous large scale certification based on certificate verification trees [A]. IFIP Communications and Multimedia Security'2001 [C]. Boston MA: Kluwer, 2000. 185-196.

作者简介:

柳　毅　男, 1976 年 11 月生于河北省邯郸市, 博士生, 主要研究方向是信息安全、移动代理. E-mail: liuyi_xd@126.com.

姜正涛　男, 1977 年出生于山东省青岛市, 博士生, 主要研究方向是信息安全、密码学.

王育民　男, 1936 年出生于北京市, 教授, 博士生导师, 主要研究方向是编码理论、信息安全与密码学.