PDHA:一种移动 Agent 保护协议研究*)

胡建理　王嘉祯　杨素敏
(军械工程学院计算机工程系　石家庄 050003)

摘　要　针对移动 Agent 本身的安全性问题,提出了一种基于软件防护的移动 Agent 保护协议——PDHA(a protocol to detect malicious hosts' attacks),并对该协议进行了安全性分析和性能分析。该协议的核心思想是通过关联密钥链加密机制对移动 Agent 的数据进行加密,并应用参考状态分析机制对移动 Agent 的代码、状态进行保护,从而形成一种全面的移动 Agent 保护协议。实验表明,PDHA 协议功能是可以达到的,安全性也更强。

关键词　移动 Agent,安全性,机密性,完整性,Aglets 系统

PDHA:A Protocol for Mobile Agent Protection

HU Jian-Li　WANG Jia-Zhen　YANG Su-Min
(Dept. of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003)

Abstract　Focusing on the security problems of mobile Agents, a new software protection based protocol-PDHA(a protocol to detect malicious hosts' attacks) is proposed. Security analyses and performance analyses for this protocol are surveyed. The key idea of PDHA protocol is to encrypt data of mobile Agents with the mechanism of interrelated keys chains encryption, to protect codes and states of mobiles Agents with the mechanism of reference states analyses, and to present a full protocol for mobile Agent protection. The simulation experiments show the functionality of PDHA protocol can be implemented, and its security is higher.

Keywords　Mobile agent, Security, Confidentiality, Integrity, Aglets system

1　引言

Internet 技术的不断发展和软件方法学的不断演化促进了移动 Agent 技术的产生。移动 Agent(Mobile Agent)通常是指那些在网络上自由漫游,可迁移到除发起节点之外的远程节点上执行,以完成其所需任务的程序。在移动 Agent 系统中,在网络上漫游的不仅包括 Agent 的代码,还包括数据及其执行状态[1]。移动 Agent 技术解决了传统 Client/Server 计算模式的结构缺乏网络适应性的缺点,具有很好的应用前景。但是由于移动 Agent 在网络中的移动性,大规模地在开放的 Internet 中广泛应用移动 Agent,一个需要特别研究的问题就是移动 Agent 系统的安全性问题。

移动 Agent 的安全性问题分为 3 类[2~4]:

(1)保护运行移动 Agent 的主机或实体[亦称运行环境(Execution Environment)]不受恶意 Agent 的攻击;

(2)保护移动 Agent 间的通信过程交互信息的安全性;

(3)保护移动 Agent 不受恶意的运行环境以及不协作的或者有敌意的其他 Agent 的攻击。

第(1)和第(2)个问题在移动 Agent 模式出现之前就存在,只是形式不同,对它们的研究相对比较成熟,第(3)个问题尤其具有挑战性。对第(3)个问题,我们认为可以只记录移动 Agent 在各主机运行时的输入和运行后的结果,将这些记录发送给源主机或源主机的代理,即可达到检测恶意主机攻击的目的,从而提出新的检测恶意主机攻击的记录协议 PDHA。这一协议的检测开销有较大的降低,安全性更强。

本文第 2 节阐述了 PDHA 协议的详细流程;第 3 节和第 4 节分别对 PDHA 协议的安全性和复杂度进行了分析;第 5 节讨论了 PDHA 协议在 Aglets 系统中的应用,最后对全文进行了总结。

2　PDHA 协议详细流程

为了更好地描述,表 1 列举了本节所用符号的说明。

PDHA 协议的描述:

$KS_0=H(R1_0)$;

$KC_i=H(KS_{i-1},R1_i)$;

$KS_i=H(KC_i,R2_i)=H(H(KS_{i-1},R1_i),R2_i)$;

$K_i=EP_{KU(i+1)}(KS_i)$。其关联示意图如图 1 所示。

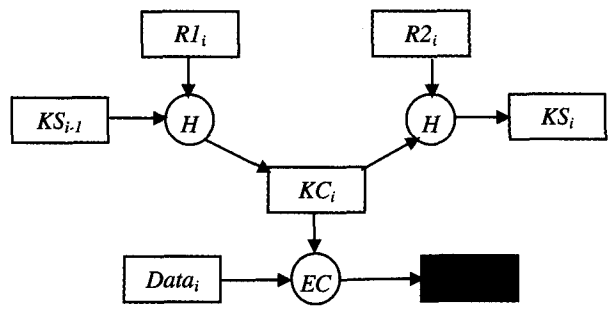

图 1　密钥关联关系示意图

*)国家 863 资助项目(编号:2004AA1Z2450)、智能技术与系统国家重点实验室开放课题资助项目(编号:99008)、河北省科技攻关计划资助项目(编号:WM05G13)。**胡建理**　讲师,博士生,主要研究方向为网络攻击与防御、分布式计算技术;**王嘉祯**　教授,博导,主要研究方向为网络安全、信息隐藏和图像处理;**杨素敏**　讲师,博士生,主要研究方向为信息隐藏、网络防御。

表 1　符号说明

符号	说明
KS_i	主机 S_i 的会话密钥(对称加密体制)
KC_i	主机 S_i 的加密密钥(对称加密体制)
KR_i	主机 S_i 的私钥(非对称加密体制)
KU_i	主机 S_i 的公钥(非对称加密体制)
EP_k	用公钥 k 加密(非对称加密体制)
DP_k	用公钥 k 解密(非对称加密体制)
EC_k	用密钥 k 加密(对称加密体制)
DC_k	用密钥 k 解密(对称加密体制)
SIG_i	用主机 S_i 的私钥数字签名
H	取报文摘要(哈希处理)
R_i	主机 S_i 用源主机 S_0 的公钥对相关信息进行加密所得信息链
D_i	主机 S_i 用传统密钥对其数据进行加密所得结果
MA_i	主机 S_i 上移动 Agent 唯一的序列号
$Code$	移动 Agent 代码
TS_i	主机 S_i 所用的时间戳
S_i	表示主机 i
G_i	主机 S_i 上包含上一主机名、代码及对报文摘要的签名信息
K_i	用主机 S_{i+1}的公钥加密主机 S_i 的会话密钥所得信息
$R1_i(R2_i)$	主机 S_i 所产生的随机数
ST_i	主机 S_i 的状态

加密形式：

$D_i=\{EC_{KCj}(Data_j)\mid 0<j\leqslant i\}$

$R_i=\{R1_j,R2_j,SIG_0(H(S_0,D_j,Code,MA_j,TS_j))\mid 0<j\leqslant i\}$

$G_i=\{S_{i-1},SIG_{i-1}(H(ST_{i-1},S_{i-1},S_i))\mid 0<j\leqslant i\}$

$U_i=\{EP_{KUi}(D_j,Code,ST_{i-1},S_0,TS_j,MA_i)\mid 0<j\leqslant i\}$

传输内容：

$S_i\rightarrow S_{i+1}:\{G_i,K_i,D_i,R_i,U_i\mid 0\leqslant j\leqslant i\}$

从上述描述中可以看出，PDHA 协议具有如下特点：

· PDHA 协议是一种基于检测的方法，属于软件保护方案；

· 通过记录移动 Agent 运行时的输入和运行后的结果对移动 Agent 的运行进行验证；

· 可以保护移动 Agent 的代码、状态的完整性；

· 能够防范对移动 Agent 篡改攻击、伪装攻击、DoS 和重放等多种攻击；

· 对移动 Agent 的数据采用常规加密，从而保护数据的机密性；

· 在移动 Agent 数据加密所用的密钥之间建立关联关系，形成密钥链，任何打破这种链关系的行为都将被检测到，从而保护数据的完整性；

· 密钥链中的密钥是通过对随机数和上一个密钥进行两次哈希而形成的，是一次性的。哈希函数具有单向性，密钥链是不可逆推的，即任何一个主机都无法逆推出上一个主机所使用的加密密钥。

整个协议分为 7 部分：

(1)在 Agent 所有者主机 S_0 上的初始化

(a)Generate Random Number:$R1_0(128)$

主机 S_0 上产生 128 位的随机数 $R1_0$，并保存好该随机数。

(b)Run:$K_0=EP_{KU1}(KS_0)=EP_{KU1}(H(R1_0))$

计算出初始的 KS_0，KS_0 是对 $R1_0$ 进行哈希得到的，采用的哈希算法为 MD5。把 KS_0 用下一个主机的公钥加密后生成 K_0。

(c)Initialize: $G_0=\{\text{void}\}$、$D_0=\{\text{void}\}$、$U_0=\{\text{void}\}$和 $R_0=\{\text{void}\}$

初始化 G_0、D_0、U_0 和 R_0，其初始值均为空。

(d)Transfer:G_0，K_0，D_0，R_0，U_0

将 G_0，K_0，D_0，R_0，U_0 传送给下一主机 S_1。

(2)在中间主机 S_i 上运行，其开始运行时的状态由 S_{i-1} 决定

(a)Generate Random Number:$R1_i(128)$，$R2_i(128)$

主机 S_i 产生两个 128 位的随机数 $R1_i$ 和 $R2_i$。采用 Agent 所有者主机 S_0 的公钥 KU_0 加密。这样，只有 S_0 才能解密它们，合法地得到这两个随机数的值。

(b)Run:$KC_i=H(R1_i+KS_{i-1})$

KC_i 是对 $R1_i+KS_{i-1}$ 进行哈希得到的，KS_{i-1} 是上一个主机所传送过来的，采用的哈希算法为 MD5。MD5 算法以任意长度的信息作为输入，生成 128 位的哈希值。

(c)Run:$Data_i$ compute on S_i

Agent 在主机 S_i 上运行产生自己的数据 $Data_i$。

(d)Append:$D_i=D_{i-1}+EC_{KCi}(Data_i)$

主机 S_i 加密自己产生的数据，采用的是常规加密算法 AES，密钥为 KC_i。设定 AES 算法的密钥长度为 128 位，正好和 MD5 算法的哈希值长度一致。加密后的数据存储到 Agent 的数据项中。

(e) Run: $K_i=EP_{KU(i+1)}(KS_i)=EP_{KU(i+1)}(H(R2_i+KC_i))$

KS_i 是对 $R2_i+KC_i$ 进行哈希得到的，采用的哈希算法为 MD5。把 KS_i 用下一个主机的公钥加密后生成 K_i。

(f)Append:$R_i=R_{i-1}+(R1_i,R2_i,SIG_0(H(S_0,D_i,Code,MA_i,TS_i)))$

主机 S_i 首先产生时间戳 TS_i，再对已有的主机名 S_0、数据项 D_i、代码 $Code$ 和时间戳 MA_i 组成信息的报文摘用源主机 S_0 的私钥签名，然后把该签名随机数 $R1_i$ 和 $R2_i$ 一起生成主机 S_i 的 R_i，其过程如图 2 所示。数字签名算法为 RSA，哈希算法为 MD5。

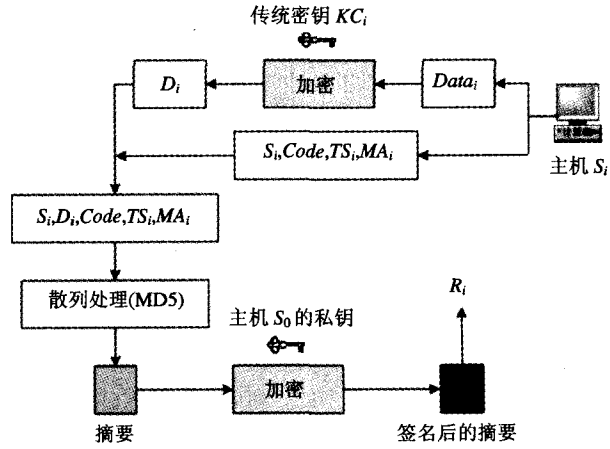

图 2　主机 S_i 生成 R_i 的过程

(g)Append:$G_i=G_{i-1}+(S_{i-1},SIG_{i-1}(H(ST_{i-1},S_{i-1},S_i)))$

将上一主机 S_{i-1} 的状态 ST_{i-1}、上一主机名 S_{i-1} 及当前

主机名 S_i 所组成的信息的报文摘要用 S_{i-1} 的私钥签名，然后签名信息与 S_{i-1} 生成主机 S_i 的 G_i。数字签名算法为 RSA，哈希算法为 MD5，其过程如图 3 所示。

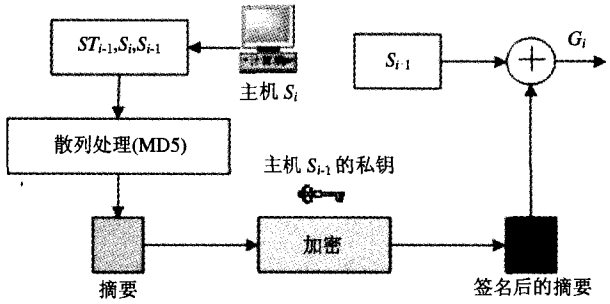

图 3　主机 S_i 生成 G_i 的过程

(h)Append:$U_i=U_{i-1}+EP_{KUi}(D_i,Code,ST_{i-1},S_0,TS_i,MA_i)$

主机 S_i 首先产生时间戳 TS_i，然后将经过加密的数据项 D_i、代码 Code、上一主机的输出状态 ST_{i-1}、源主机名 S_0、移动 *Agent* 唯一序列码 MA_i 及 TS_i 所组成的信息用主机 S_i 的公钥进行加密。所用的非对称加密算法为 RSA。

(i)Transfer:G_i,K_i,D_i,R_i,U_i

将 G_i,K_i,D_i,R_i,U_i 传送给下一主机。

(3)中间主机 S_i 收到上一主机 S_{i-1} 发来的信息，并检查其完整性。

S_{i-1} 向 S_i 发送信息。其中 G_i 中 S_{i-1} 的用以告知 S_i 发送该移动 Agent 的上一主机为 S_{i-1}，$SIG_{i-1}(H(ST_{i-1},S_{i-1},S_i)$ 为校验信息；U_i 用以将移动 Agent 的代码、数据和状态安全传输给 S_i，并告知 S_i 该移动 Agent 的源主机、时间戳和唯一序列号。由于移动 Agent 使用的 S_i 公钥加密，所以只有 S_i 能够恢复移动 Agent 并将其激活运行；R_i 为上一主机 S_{i-1} 向 S_i 发送的校验信息。

S_i 收到信息后，使用自身的私钥解密 U_i，得到移动 Agent 的 $D_j,Code,ST_{i-1},S_0,TS_j,MA_i$。

S_i 收到的 S_{i-1} 信息后，根据 G_i 中 S_{i-1} 的名字检索其公钥，然后根据移动 Agent 状态 ST_{i-1}、发送者的名字 S_{i-1}、本机名 S_i 以及该公钥进行验证。如果验证获得通过，说明该移动 Agent 的上一主机确实是 S_{i-1}，移动 Agent 状态在传输中没有被修改，本机确实是该信息的接收者。

(4)主机 S_i 对 $D_j,Code,ST_{i-1},S_0,TS_j,MA_i$ 用源主机 S_0 的公钥加密，生成信息 $V_i=\{EP_{KU0}(D_j,Code,ST_{i-1},S_0,TS_j,MA_i)|0<j\leqslant i\}$。

(5)主机 S_i 将 K_i,D_i,R_i 及 V_i 组成的信息返回到 Agent 所有者主机 S_0。如果 S_0 在生命期内没收到该消息，则检查问题。

(a)Run:$K_{S0}=H(R1_0)$

这是对 $R1_0$ 进行哈希得到 KS_0，采用的哈希算法为 MD5。

(b)Run:$KC_i=H(KS_0+R1_i)$ 和 $KS_i=H(KC_i+R2_i)$

由 R_i 可得到的 $R1_i$ 和 $R2_i$ 进行计算，得到 KC_i 和 KS_i。采用的哈希算法为 MD5。

(c)Run:$DC_{KCi}(D_i)$

解密 $EC_{KCi}(Data_i)$ 得到 $Data_i$，解密密钥为 KC_i。

(d)源主机 S_0 用其私钥对 V_i 进行解密，得到 $D_j,Code,ST_{i-1},S_0,TS_j,MA_i$。

(e)Verify:$SIG_0(H(S_0,D_j,Code,MA_j,TS_j))$。

源主机 S_0 用其公钥对 $SIG_0(H(S_0,D_j,Code,MA_j,TS_j))$ 进行解密得到报文摘要 $H(S_0,D_j,Code,MA_j,TS_j)$，并根据 S_0 经过其解密得到的 $D_i,Code,MA_j,TS_j$ 对 R_i 中的签名信息进行验证。如果通过验证，说明该移动 Agent 的源主机确实是 S_0，移动 Agent 的代码和数据在传输中没有被修改。并检查时间戳 TS_j 是否在移动 Agent 的生命期内，以及唯一的序列号 MA_j 是否有重复。其验证过程如图 4 所示。

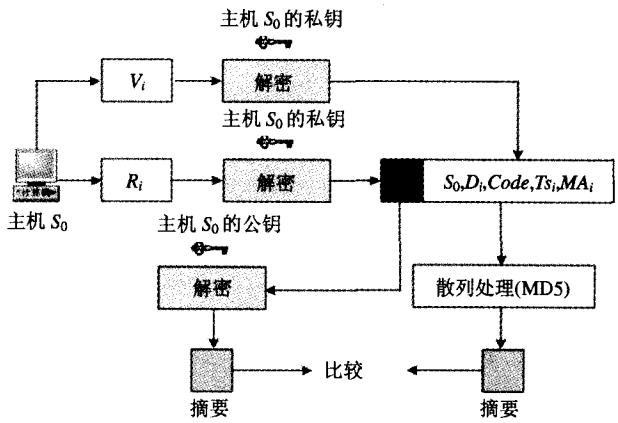

图 4　源主机 S_0 验证数据和代码完整性的过程

(6)如果移动 Agent 已完成预定任务，则结束；否则执行下一步。

(7)$i:=i+1$，重复执行(2)～(5)，直至任务最终完成或中止。

3　PDHA 协议安全性分析

PDHA 协议的安全性基于以下假设：

- 机制所用的对称加密算法、公钥加密算法和数字签名算法是安全的；
- 机制所用的随机数生成算法是安全的；
- 机制所用的哈希算法是安全的；
- 机制所涉及到的各个主体的私钥是安全的。

在以上假设的基础上，PDHA 协议能够防范被动攻击者、主动攻击者、被动骗子、主动骗子的攻击，能够保护移动 Agent 的安全性和完整性。分析如下。

3.1　被动攻击者

被动攻击者监听信道，试图窃取敏感信息。

假设被动攻击者能够截获在网络上传递的 G_i,K_i,D_i,U_i,R_i,V_i，试图窃取移动 Agent 数据，然而

$G_i=\{S_{i-1},SIG_{i-1}(H(ST_{i-1},S_{i-1},S_i))|0<j\leqslant i\}$。假设被动攻击者可以截获该信息：被动攻击者可以发现该信息发送者的名字 S_{i-1} 对 PDHA 协议的安全性没有任何影响；$SIG_{i-1}(H(ST_{i-1},S_{i-1},S_i))$ 只是有校验功能的签名信息，攻击者即使获得该信息，并不能从中得到移动 Agent 的信息，并且不能用这个信息来谎称他收到了该信息所代表的移动 Agent。因为该签名信息是被 S_{i-1} 签名的，同时包括接收者 S_i 的信息。

$U_i=\{EP_{KUi}(D_j,Code,ST_{i-1},S_0,TS_j,MA_i)|0<j\leqslant i\}$。这一部分是用接收主机 S_i 的公钥加密的，所以只有该主机可以解密并激活该移动 Agent。根据安全假设，主机的私钥是安全的，所以被动攻击者无法获取移动 Agent 的信息。也就是说，无法从 U_i 中获取任何信息。同理，也无法从 V_i 获取任

何信息。

$K_i=EP_{KU(i+1)}(KS_i)=EP_{KU(i+1)}(H(R2_i+KC_i))$。根据安全假设,主机的私钥是安全的,所以攻击者无法解密 K_i,无法从 K_i 中获取任何信息。

$D_i=D_{i-1}+EC_{KCi}(Data_i)$。主机 S_i 加密自己产生的数据 $Data_i$,采用的是常规加密算法 AES。根据安全假设,攻击者在没有密钥 KC_i 的情况下是无法从 D_i 中获取 $Data_i$ 的。

$R_i=\{R1_j,R2_j,SIG_0(H(S_0,D_j,Code,MA_j,TS_j))|0<j\leqslant i\}$。$R1_j,R2_j$ 为 S_i 产生的两随机数,对被动攻击者来说无任何意义;$SIG_0(H(S_0,D_j,Code,MA_j,TS_j))$只是有校验功能的签名信息,攻击者即使获得该信息,也不能从中得到移动 Agent 的信息。

3.2 主动攻击者

主动攻击者可能伪装成 PDHA 协议的参与者,在 PDHA 协议中引入新的信息、删除原有的信息、用另外的信息代替原有的信息等。

首先,主动攻击者无法做到引入新的信息或者删除原有信息而不被发现。假设主动攻击者可以拦截 G_i,K_i,D_i,U_i,R_i,V_i,并进行攻击。因为 G_i,K_i,D_i,U_i,R_i,V_i 都有明确的意义和形式,如果主动攻击者引入新的信息或者删除原有信息,则接受该消息的下一个主机立即能够发现格式的变化,知道被攻击了。

其次,主动攻击者也无法做到用另外的信息代替原有信息而不被发现。假设主动攻击者截获在网络上传递的 G_i,K_i,D_i,U_i,R_i,并恶意修改 Agent 数据。如攻击者冒充成 S_i,将数据项 $ECKC_i(Data_i)$ 修改成其他数据,但是由于主动攻击者没有 S_i 的私钥,因此无法伪造 S_i 的签名,只能用自己的私钥对 D_i 签名。当回到所有者主机时,检测数字签名就能发现存在恶意篡改。

3.3 被动骗子

被动骗子遵守 PDHA 协议,但企图获取 PDHA 协议以外的其他信息,试图窃取前面主机所产生的移动 Agent 数据、状态或代码。

假设被动骗子 S_{i+1} 收到上一个主机 S_i 传递的 G_i,K_i,D_i,U_i,R_i,试图窃取前面主机所产生的移动 Agent 数据,然而:

$K_i=EP_{KU(i+1)}(KS_i)=EP_{KU(i+1)}(H(R2_i+KC_i))$。这个本来就是给主机 S_{i+1},所以被动骗子 S_{i+1} 能够解密 K_i,得到 KS_i。但是 $KS_i=H(R2_i+KC_i)$,根据安全假设,被动骗子 S_{i+1} 无法获得 KC_i。同理,被动骗子 S_{i+1} 无法获取前面的主机的 $KC_0,KC_1,\cdots,KC_{i-1}$。

$D_i=D_{i-1}+EC_{KCi}(Data_i)$。主机 S_i 加密自己产生的数据 $Data_i$,采用的是常规加密算法 AES。根据安全假设,攻击者在没有密钥 KC_i 的情况下无法从 D_i 中获取 $Data_i$。同理,被动骗子 S_{i+1} 也无法获取前面的主机 $S_0,S_1,\cdots,S_{i-1}$ 所生成的数据。

$G_i=\{S_{i-1},SIG_{i-1}(H(ST_{i-1},S_{i-1},S_i))|0<j\leqslant i\}$。被动骗子可能获取 S_{i-1},对 PDHA 协议本身的安全性没有影响;$SIG_{i-1}(H(ST_{i-1},S_{i-1},S_i))$只是有校验功能的签名信息,对被动骗子不可用。

$R_i=R_{i-1}+(R1_i,R2_i,SIG_0(H(S_0,D_i,Code,MA_i,TS_i)))$。$R1_j,R2_j$ 为 S_i 产生的两随机数,对被动攻击者来说无任何意义;$SIG_0(H(S_0,D_j,Code,MA_j,TS_j))$只是有校验功能的签名信息,攻击者即使获得该信息,也不会获取移动 Agent 的任何信息。

$U_i=\{EP_{KUi}(D_j,Code,ST_{i-1},S_0,TS_j,MA_i)|0<j\leqslant i\}$。根据安全假设,主机 S_i 的私钥是安全的,所以攻击者无法解密 $EP_{KUi}(D_j,Code,ST_{i-1},S_0,TS_j,MA_i)$,也就是说,无法从 R_i 中获取任何信息。对 V_i 可做同样的分析。

3.4 主动骗子

主动骗子假装遵守 PDHA 协议,但企图通过篡改或删除前面主机的数据、重放等手段来破坏 PDHA 协议。

主动骗子无法做到篡改前面主机的数据而不被发现。假设主动骗子 S_i 出于某种目的将收到的 $G_{i-1},K_{i-1},D_{i-1},U_{i-1},R_{i-1}$ 中的某个数据项 $EC_{KCj}(Data_j)$ 修改成其他数据,生成新的 D_i,再生成 $EP_{KUi}(D_j,Code,ST_{i-1},S_0,TS_j,MA_i)$。但是由于保护信息中的相应项 $EP_{KUi}(D_j,Code,ST_{i-1},S_0,TS_j,MA_i)$是用发送者主机公钥加密的,主动骗子 S_i 无法修改。当回到所有者主机时,校验时检测到数字签名 $SIG_{i-1}(H(ST_{i-1},S_{i-1},S_i))$就能发现存在恶意篡改。

主动骗子无法做到截取掉某些数据项和保护信息而不被发现,如主动骗子 S_i 出于某种目的把它前面主机所产生的数据和保护信息都截掉,生成新的 D_i,再生成 $EP_{KUi}(D_j,Code,ST_{i-1},S_0,TS_j,MA_i)$。但由于主动骗子 S_i 无法获取前面主机的各个 $KS_j(0<j<i)$,因此主动骗子 S_i 只能用他收到的 KS_{i-1} 和他自己生成的随机数 $R1_i$ 和 $R2_i$ 来生成 KC_i 和 KS_i,密钥链关系被破坏。当 Agent 回到所有者主机时,进行校验,用 KS_0 和主动骗子 S_i 生成的随机数 $R1_i$ 和 $R2_i$ 生成解密密钥,这与当初主动骗子 S_i 生成的 KC_i 和 KS_i 不同,解密 D_i 的结果不可解,从而检测出篡改。部分截取的情况可类似分析。

同理,也可类似分析移动 Agent 的代码和状态的篡改和重放情况。

主动骗子也可能会保留目前结果,以后重新发出,但是 PDHA 协议采用时间戳防止了重放攻击。同时,由于常规加密密钥 KCi 是通过对随机数哈希生成的,而随机数也是一次一用的,也防止了重放攻击。

4 PDHA 协议复杂度分析

PDHA 协议对移动 Agent 数据进行保护是以增加执行和传输时间为代价的。执行时间是由于在每个主机上的随机数生成、哈希、加密、签名等操作所引起的,而传输时间是由于移动 Agent 带有保护信息而引起移动 Agent 容量增加而引起的。

4.1 执行时间分析

执行时间取决于两个:在中间站点的执行时间 T_{TOT_INT} 和 Agent 所有者确认 Agent 完整性的时间 T_{TOT_SNT}。在访问 N 个主机时,在中间主机上总的执行时间 T_{TOT_INT} 为:

$$T_{TOT\_INT}=N(4T_{HASH}+2T_{RAND}+T_{CRYPT\_AES}+3T_{CRYPT\_RSA}+3T_{SIGN})\approx N(T_{CRYPT\_AES}+3T_{SIGN})$$

其中:T_{HASH} 是哈希操作的时间;T_{RAND} 是生成随机数操作的时间;T_{CRYPT_AES} 是对主机所收集的数据进行 AES 加密的时间;T_{CRYPT_RSA} 是对保护信息和密钥链 K 进行 RSA 加密的时间;T_{SIGN} 是签名的时间。

哈希操作只是对主机所产生的大小固定的 128 位的随机数进行的。RSA 加密操作是对小容量的信息进行的。随机数生成采用的是普通的随机数生成算法。因此 T_{HASH},T_{RAND},T_{CRYPT_RSA} 与 T_{CRYPT_AES} 相比可以忽略不计。

T_{CRYPT_AES}依赖于应用数据，即中间主机所收集的数据的不同容量。当数据容量增大，T_{CRYPT_AES}也会增长，根据不同的应用会有所变化。

当 Agent 返回，Agent 所有者要确认移动 Agent 数据的完整性。执行时间 T_{TOT_SNT}为

$$T_{TOT\_SNT}=N(4T_{HASH}+T_{DECRYPT\_AES}+5T_{DECRYPT\_RSA}+T_{VER})\approx N(T_{DECRYPT\_AES}+T_{VER})$$

其中：T_{HASH}是哈希操作的时间；$T_{DECRYPT_AES}$是对主机所收集的数据进行 AES 解密的时间；$T_{DECRYPT_RSA}$是对保护信息进行 RSA 解密的时间；T_{VER}是确认签名的时间。

4.2 传输时间分析

假设传输时间随着 Agent 容量的大小线性地增长[5]。Agent 在两个站点之间的传输时间 T_{TX}由以下组成：$T_{TX}=aD_{CID}+bD_{AD}+cD_{PD}$。其中：$D_{CID}$是移动 Agent 的代码和初始数据等不变部分的容量大小；D_{AD}是移动 Agent 在主机上所收集的数据的容量大小；D_{PD}是移动 Agent 在主机上保护信息的容量大小；a,b,c是常数。

当访问 N 个主机时，移动 Agent 在每个主机上运行后，所增加的容量是：在主机上新收集的数据和保护信息。总的传输时间 T_{TOTTX}可以表述为：

$$T_{TOTTX}=NaD_{CID}+b\sum_0^N\Delta D_{AD}+c\sum_0^N\Delta D_{PD}$$

5 实验分析

为了分析 PDHA 协议的执行性能及其安全性，我们设计了两项实验：性能分析实验和安全性实验。实验运行环境为：主机为 Intel Pentium4 3.00GHz，OS 为 Windows2003，JSDK 为 J2sdk1.4.2_04 版，Aglets 系统为 Aglets-2.0.2 版，Cryptix 算法库为 cryptix-jce-20050328-snap[6]。

5.1 PDHA 协议的实现概述

我们将 PDHA 协议集成到 Aglets 系统中，主要在 Aglets 系统的两个方法中实现。

对移动 Agent 的代码、唯一序列码、时间戳及相关信息的加密、签名和报文摘要处理是在 LocalAgletRef.dispatch 中实现的。其主要过程是将移动 Agent 序列化后生成的字节数组进行签名处理，并对时间戳、Agent 唯一序列码及 Agent 源码进行加密、对其摘要进行签名后，统一生成一个字节信息流。因此可将 PDHA 协议中经过安全处理的信息生成部分放到 LocalAgletRef.dispatch 中实现；AgletContextImpl.receiveAglet 方法是在目的主机上完成的，主要是将收到的经过发送主机处理的完整信息流进行反序列化处理，然后对其中包含的 Aglet 代码、数据、Aglet 唯一序列码和时间戳信息进行对应的解密过程，获取 Aglet 原字节流、数据及其对应的报文摘要。并对相关校验信息进行验证，测试其完整性和唯一性。因此我们可将 PDHA 协议中的信息验证部分放在 AgletContextImpl.receiveAglet 方法完成。

为了表述方便，下面我们用 Aglets 表示原 Aglets 系统，用 IAglets(improved Aglets)表示集成了 PDHA 协议的 Aglets 系统。

5.2 性能分析实验

为了测试集成 PDHA 协议后系统的性能，实验分为两组进行：一组代表在信任环境中，采用 Aglets 进行实验。Aglet 在主机之间迁移，只保存一个 Aglet 认为最好的数据；另一组代表在不信任的环境中，采用 IAglets 进行实验。Aglet 在主机之间迁移，在 Aglet 源主机进行完整性校验。为实验方便，每个中间主机提供相同大小的数据量。实验结果如图 5 和图 6 所示。

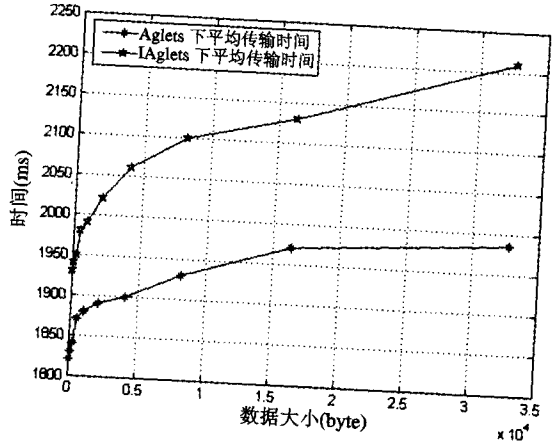

图 5 平均传输时间对比曲线

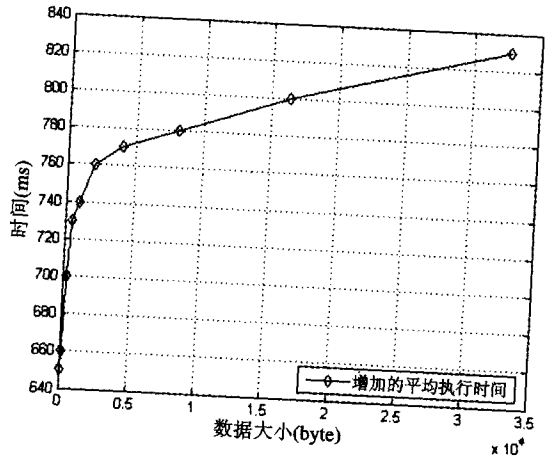

图 6 增加的平均执行时间曲线

图 5 对比了 Aglets 和 IAglets 条件下的平均传输时间。从图 5 可以看出，采用 PDHA 协议后，传输时间有所增加，但增加得并不多，可以满足实际应用的需要。图 6 记录了采用 PDHA 协议的情况下所增加的平均执行时间。在数据量较小的情况下，本实验结果为 680ms，而文[7]中的实验结果为 480ms。这是因为文[7]中采用 IKCE 机制来保护移动 Agent 数据的安全性，但主机间传输的信息量比 PDHA 协议要小。文[5]实验结果为 320ms，这是因为文[5]中只测试了完整性，没有测试机密性。在数据量较大的情况下，本实验所增加的平均执行时间比小数据情况下只增加了 180ms，说明协议在数据量变化较大的情况下对性能的影响不是很明显。

5.3 安全性分析实验

为了验证 PDHA 协议的安全性，我们开发了一个基于 Aglets 的应用系统——远程数据库查询系统(Remote Database Query System，RDQS)。该应用系统是在 Aglets 上开发的，其中没有应用任何数据保护方案。我们在可信任的环境中，在远程 Aglets 上运行 RDQS，Aglet 所返回的数据结果都是最原始的数据结果，图 7 是 RDQS 对远程数据库查询后的返回结果：

然后我们在非信任环境中，在远程 IAglets 系统中执行 RDQS。图 8 是其对全库查询后返回的结果。可以看到，在“查询结果显示”文件框中显示的内容分为两部分：上面显示的是基于 PDHA 协议中加密、数据签名、报文摘要处理后，并

经过验证、解密恢复后查询结果;下面显示的则是 Aglet 在目的主机对查询原始结果的报文摘要进行签名后返回到源主机后的结果,该结果与直接对明文的报文摘要进行数字签名后的结果是完全一致的。可以看出,RDQS 在远程 Aglets 上运行及在远程 IAglets 上运行后返回到源主机的结果完全一致,从而说明 IAglets 运行过程中数据的完备性和安全性是可以得到保障的,也间接说明了 PDHA 协议是安全的。

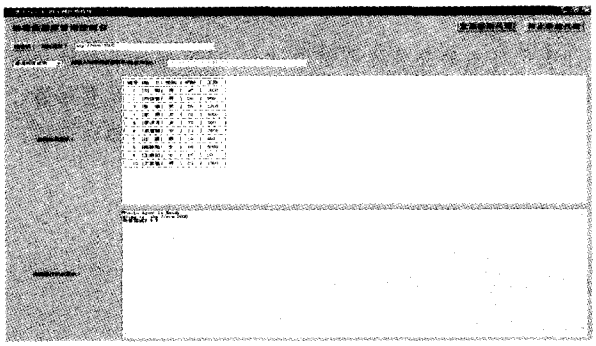

图 7 没提供任何数据保护的远程数据库系统查询结果

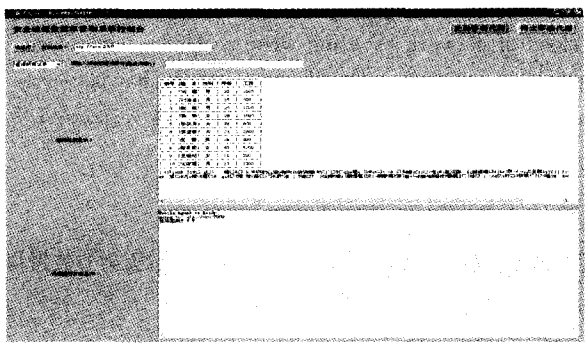

图 8 整合 PDHA 机制的应用系统在目的主机运行后所返回的查询结果

6 相关工作

PDHA 协议与相关工作对比如下:

• 与 TPE 方法[8]相比,PDHA 协议是一种基于检测的方法,属于软件保护方案。该方法不需要专用的硬件,开放性和扩展性比较好。

• 与 TTP 方法[9]相比,PDHA 协议的计算不借助于第三方实体,既没有由于移动 Agent 迁移到第三方实体而带来的时间消耗,又没有第三方实体的瓶颈问题。

• 与 Co-Signing 方法[10]相比,PDHA 协议的节点交互少,减少了由于节点交互带来的性能损失。

• 与 MH 协议[5]相比,PDHA 协议的思想也是在数据之间建立一种关联关系,以达到保护移动 Agent 数据的目的。不同的是,MH 协议采用路径哈希链,而 PDHA 协议是对加密密钥建立关联关系。MH 协议采用公钥加密来实现数据保护;PDHA 协议采用的是常规加密。一般来说,常规加密比公钥加密速度快,这在加密大数据时尤为明显。因此,PDHA 协议更加适用于移动 Agent 在中间主机收集的数据量比较大的场合,如数据安全性要求高且数据量比较大的电子商务应用、信息服务、网络管理等。

• 与 Vigna 方法[11]相比,Vigna 方法需要第三方可信主机,PDHA 协议使用源主机做记录,后者的方案更为现实,易于实现,而且不存在第三方主机所带来的安全性问题,安全性更强。

• 与 Hohl 协议[12,13]相比,Hohl 协议无法防止当前主机和下一主机的联合攻击,而 PDHA 协议则可以,安全性更强。

结论 本文针对移动 Agent 本身的安全性问题进行了研究,提出了一种基于软件的移动 Agent 防护协议。通过将 PDHA 协议集合到 Aglets 系统中进行的实验分析表明,PDHA 协议的模型是正确的,功能是可以达到的,协议是完备的,根据 SSMA 方案建设安全移动 Agent 系统的设想是可行的。

然而,移动 Agent 系统是一种新的分布式计算范型,其中还有许多安全性问题需要研究。一旦移动 Agent 系统的安全性问题得到了较好的解决,必将成为基于 Internet 和 Intranet 的分布式计算技术的主流。因此,必须进一步对移动 Agent 系统的安全性问题进行广泛深入的研究,使其不断朝着实用化、商业化的方向发展。

参 考 文 献

1 Torsten I, Frank K. Migration of Mobile Agents in Java: Problems, Classfication and Solutions [C]. In: MAMA'2000. Canada: International Computer Science Conventions Academic Press, Australia, December 2000. 362～369

2 Michael S, et al. Mobile Agents and Security [J]. IEEE Communications Magazine, 1998, 36(7): 76～85

3 Jansen W. Countermeasures for Mobile Agent Security [J]. Computer Communications, 2000, 23(17): 1667～1676

4 Moore J T. Mobile Code Security Techniques [R]: [Technical Report]. MS-CIS-98-28, University of Pennsylvania, 1998. 1～10

5 Corradi A, Montanar R. Mobile Agent Protection in the Internet Environment. In: Gibson RM, ed. Proceedings of the 23th Annual Int'l Computer Software and Applications Conf. Los Alamitos: IEEE Computer Society Press. 1999. 80～85

6 Systemics Ltd. Cryptix Library. 〈URL: http://www. cryptix. org/〉

7 潭湘,顾毓清,等. 一种用于移动 Agent 数据保护的机制. 软件学报,2005,16(3):477～484

8 Wilhelm G, Staamann M. A Pessimistic Approach to Trust in Mobile Agent Platforms. IEEE Internet Computing, 2000, 4(5): 40～48

9 Corradi A, Cremonini M. Mobile Agents Integrity for Electronic Commerce Applications. Information Systems, 1999, 24(6): 519～533

10 Cheng J S L, Victor KW. Defenses against the Truncation of Computation Results of Free-roaming Agents. In: Deng RH, Qing SH, Bao F, et al. eds. Information and Communications Security. London: Springer-Verlag, 2002. 1～12

11 Vigna G. Protecting Mobile Agents Through Tracing. Proceedings of the 3rd ECOOP Workshop on Mobile Object System. June 1997

12 Hohl F. A Framework to Protect Mobile Agents by Using Reference States. Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS 2000). 〈URL: http://mole. informatik. uni-stuttgart. de/papers/〉

13 Hohl F. A Protocol to Detect Malicious Hosts Attacks by Using Reference States. Universitat Stuttgart, Fakultat Informatik, Bericht Nr. 1999