附网存储安全模型及系统设计

韩德志　谢长生　付湘林　易发令
（华中科技大学外存储国家重点实验室　武汉464000）

The Design of Model and System for Network Attached Storage Security

HAN De-Zhi　XIE Chang-Sheng　FU Xiang-Lin　YI Fa-Ling
(State Key Lab. for Exterior Storage, Huazhong University of Science and Technology, Wuhan 464000)

Abstract　The Network Attached Storage (NAS) is one of the most effective projects for solving the network storage bottleneck at present. Its security mechanism is different from the common network security. This paper puts forward a NAS security model based on the Petri Nets and provides a NAS security system designed according to the model which has been proved effective and applicable by tests.

Keywords　Network attached storage, Petri nets, Authentication request, Application encryption

1　引言

附网存储(Network Attached Storage,NAS)是可以直接挂到网络上向用户提供文件级服务的存储设备,它有自己简化的实时操作系统,并将硬件和软件有效地集合在一起,用以提供文件服务。附网存储是目前解决网络存储瓶颈最有效的方案之一,其安全机制有自己的特点,不同于一般的网络安全。本文根据附网存储系统安全机制的特点,提出了一个基于Petri网(Petri Nets,PN)的附网存储安全系统模型,并在此模型的基础上,设计了一个附网存储安全系统,此系统通过试用是实用和高效的。

2　Petri网

Petri网(PN)源于西德计算机科学家C. A. Petri在1962年发表的博士论文,它提出了一种模拟"多人"游戏的建模方法,可方便地描述系统的并发、竞争和同步等特征。经过全世界众多计算机科学家的共同发展,丰富了它的内容。如今Petri网已成为一种描述和分析并发、分布式系统及工作流分析的强有力的数学模型工具,已广泛地应用于计算机系统建模、性能评价中。Petri网是一个具有两类结点的有向图,这两类结点叫做位置结点和变迁(Transition)结点,位置结点也叫库所(place),结点之间由有向弧(Directed Arc)连接,并且同类结点之间的连线是不允许的。通常库所由圆圈(○)表示,变迁由矩形(□)表示。

定义1　Petri网可由一个三元组表示 $PN=(P,T,A)$ 表示,其中:

(1) $P=\{P_1,P_2,\cdots,P_n\}$ 是一个有穷非空库所集;

(2) $T=\{t_1,t_2\cdots t_m\}$ 是一个有穷非空变迁集;

(3) $A\subseteq(PXT)U(TXP)$ 是有向弧集。它表示PN中的流关系,其中"X"表示笛卡尔积;

定义2　Petri网的库所中所含元素叫托肯,它可以是人、事件或状态等。在Petri网的运行过程中,托肯的数目可以改变,变迁是Petri网中的活动部件,它们根据以下变迁规则改变网的状态。

定义3　Petri网的变迁规则:

(1)一个变迁被授权(Enabled)当且仅当变迁t的每个输入库所p含有至少一个托肯。

(2)一个被授权的变迁可发生(fire/occur)。如果变迁t发生,那么t消耗它的每个输入库所p中的一个托肯,并且在它的每个输出库所p中产生一个托肯。

这里,我们给出经典Petri网的一个实例模型,该模型描述了一个简单的认证系统Authsys(图1)。

$Authsys=\{P,T,A\}$,其中 $P=\{in,out,free,busy\}$, $T=\{start,finish\}$, $A=\{(in,start),(free,start),(start\ busy),(busy,finish),(finish,free),(finsh,out)\}$。

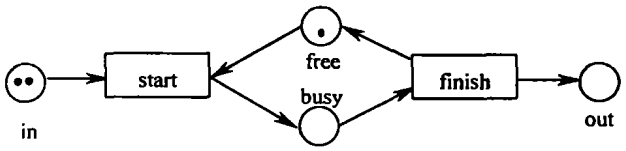

图1　一个经典Petri网描述简单的认证系统模型

库所in和free中各有2个托肯和1个托肯,库所in中的2个托肯表示有两个用户等待认证,库所free中的一个托肯表示系统处于空闲状态。当系统正在认证一个用户时,库所free中就没有托肯,而在库所busy中有一个托肯。库所out中的托肯表示已经被系统认证的用户。变迁start在初始标识下被授权,因为它的每个输入库所中都至少有一个托肯。而变迁finish在初始标识下没有被授权,因为它的输入库所busy中没有托肯,因此只有变迁start可以发生。正在发生的变迁start从两个输入库所中各消耗一个托肯,并在它的输出库所busy中产生一个托肯,表示系统正在工作。这时,变迁finish被授权发生,它从库所busy中消耗一个托肯,并在它的两个输出库所free和out中各产生一个托肯。这时,变迁start又被授权了。只要库所in中有等待认证的用户,两个变迁被交替授权发生的过程就一直持续下去。

*)基金项目:本文受国家自然科学基金项目(60173043)资助。谢长生　教授,博士生导师,研究方向为:基于网络的存储系统,网络安全。韩德志、傅湘林、易发令　博士生,研究方向为:网络安全,基于网络的存储系统,基于IP的存储区域网。

3 基于 Petri 网的附网存储安全系统模型

专用的附网存储服务器(Network Attached Storage Server,NASS),既是近程和远程访问的通信服务器,又是安全系统的主体。同时兼有二者的功能。故此它有比较复杂的状态变化。对同一局域网用户,由用户提供口令,经过"内置认证"模块来解决安全问题。对 Internet 用户的认证请求,经过安全系统的"认证请求"模块处理,生成新的认证请求包传给认证服务器。认证服务器处理后,经过"认证应答"模块把认证服器应答信息传给用户。不论是内置认证,还是认证服务器认证,只有认证被通过,用户才能对 NASS 进行读写操作。图2是基于 Petri 网的 NAS 安全模型。

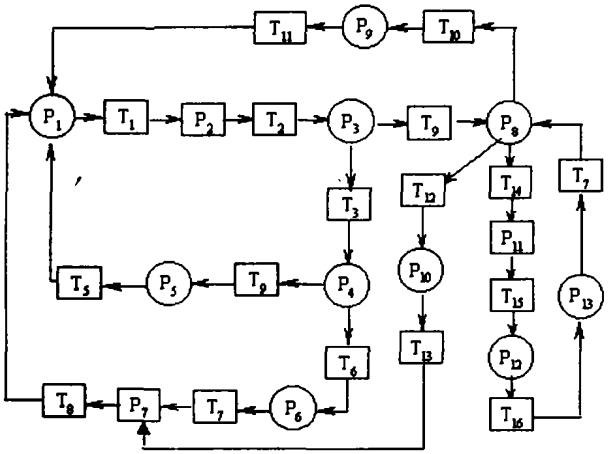

图2 基于 Petri 网的 NAS 安全模型

图2是由库所、变迁及有向弧组成,各库所及变迁具体含义解释如下:

库所集 P:

P_1:NASS 在等待客户认证信息;

P_2:NASS 接收到客户的认证信息;

P_3:NASS 得到客户认证信息分类处理结果;

P_4:NASS 等待内置认证结果;

P_5:NASS 准备断开与同一局域网用户的连接;

P_6:NASS 为同一局域网客户进行连接配置并提供连接服务;

P_7:NASS 准备结束与用户的信息交换;

P_8:NASS 等待认证 Server 的认证响应;

P_9:NASS 准备断开 Internet 用户的连接;

P_{10}:NASS 为 Internet 客户进行配置并提供连接服务;

P_{11}:NASS 等待下一个客户认证信息的到来;

P_{12}:NASS 等待客户对质询包的应答;

P_{13}:NASS 收到客户对质询信息的应答。

迁移集 T:

T_1:客户向 NASS 发请求认证信息;

T_2:NASS 对请求认证信息进行分类处理;

T_3:NASS 内置认证模块对同一局域网用户进行认证;

T_4:内置认证模块向 NASS 发访问拒绝数据包;

T_5:NASS 断开与同一局域网用户的连接;

T_6:NASS 向同一局域用户发接受访问消息;

T_7:NASS 与同一局域网客户进行信息交换;

T_8:NASS 收到新的认证请求;

T_9:NASS 产生请求认证数据包并发向认证 server;

T_{10}:认证服务器向 NASS 发访问拒绝数据包;

T_{11}:NASS 断开与 Internet 客户的连接;

T_{12}:认证服务器向 NASS 发接受访问数据包;

T_{13}:NASS 与 Internet 客户进行信息交换;

T_{14}:认证服务器向 NASS 发认证质询消息;

T_{15}:NASS 将认证质询消息发向客户;

T_{16}:客户将认证质询结果发向 NASS;

T_{17}:NASS 再向认证服务器发请求认证数据包。

4 附网存储安全系统的设计及组成

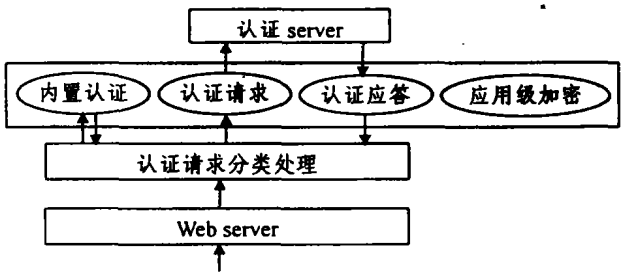

图3 附网存储安全系统结构图

图3是我们根据附网存储安全系统模型所设计的专用附网存储服务器安全系统的内部结构图。专用的附网存储服务器,主要是提供网络和存储功能,我们在安全系统中设置了五个服务进程,它们分别是:认证请求分类处理服务进程,内置认证服务进程,认证请求服务进程,认证应答服务进程,应用级加密解密服务进程,它们对来自客户端的认证信息及客户与 NASS 之间传递的信息进行处理。另外还需要设置一守护进程,负责接受的认证请求信息。

当我们打开附网存储服务器时,系统首先读配置文件并对系统进行各种配置,配置完毕后,创建并激活守护进程,守护进程通过监听 Web server 的80端口,接受用户的认证请求信息。其工作流程如图4所示。下面将分别介绍图3各模块的工作过程。

4.1 内置认证服务模块

其工作过程如下:

初始化;

等待认证分类处理进程传来的用户认证请求消息;

从认证请求消息中得到认证信息;

进行认证;

若为合法用户,向其发送接受访问消息;

返回;

若为非法用户,向其发送拒绝访问消息;

返回。

4.2 认证请求模块

其工作过程如下:

初始化;

等待来自认证分类处理进程传来的认证信息;

若为认证消息,生成 access_request 包;

将 access_request 包发向认证服务器;

返回;

若为用户质询应答消息,从质询应答消息中获取应答结果;

生成 access_request 包;

将该 access_request 包发向认证服务器；

返回。

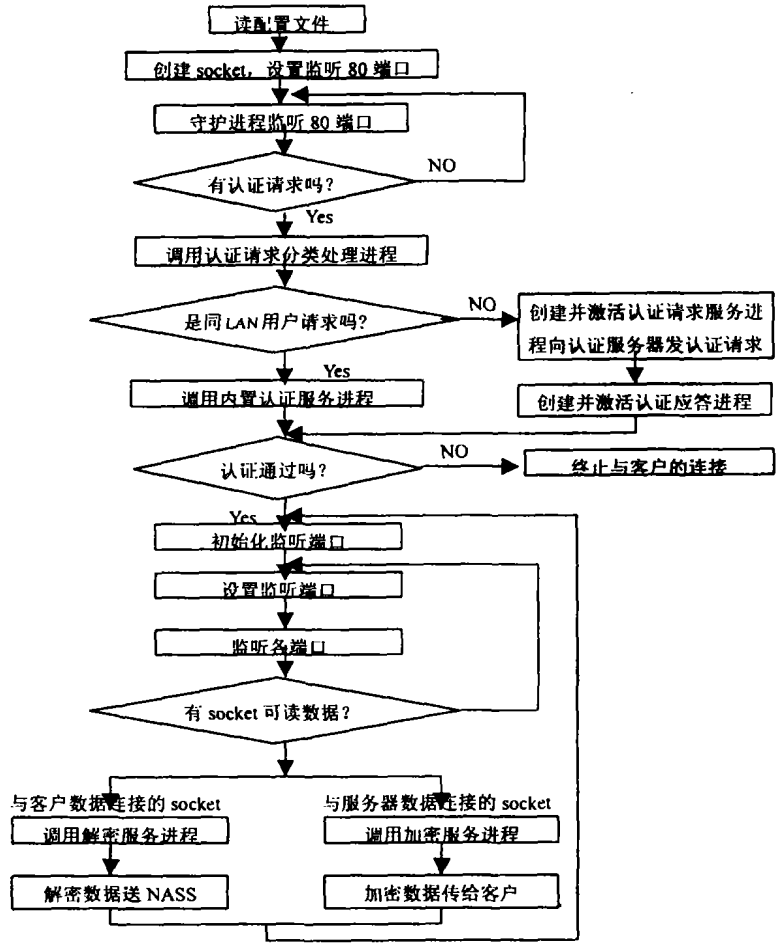

图4　附网存储安全系统框架流程

4.3　认证请求应答模块

其工作过程：

初始化；

等待认证服务器的认证应答数据包；

若为 access_accept 包，创建并激活加密/解密服务进程；

返回；

若为 access_reject 包，通过 web server 向用户发拒绝访问消息；

返回；

若为 access_challenge 包，提取该包中提示用户回答的信息；

生成用户提示消息并通过 Web server 发给用户；

返回。

4.4　应用级加密模块

其工作过程：

初始化；

若是用户写入 NASS 数据，则取解密算法对其写入数据解密；

返回；

若是用户从 NASS 读出数据，则取加密算法对其读出数据加密；

返回。

从图4附网存储安全系统主框架流程可了解到我们设计的附网存储安全系统的设计思想：与附网存储服务器处于同一局域网的用户(Intranet 用户)，其认证请求只需内置认证模块认证即可；处于附网存储服务器同一局域网以外的用户(Internet 用户)，其认证请求需认证服务器进行认证。这里认证服务器是一个较大的应用程序，也装在同一附网存储服务器上。对用户口令、用户与附网存储服务器传输的各种信息都采用密文传输。用户口令主要是对用户身份进行认证，以防止重发攻击，对用户口令加密主要是防止口令被中途窃听。这里的信息加密是采用对称密钥加密，其密钥是由用户口令通过密钥生成器生成的；用户口令和对称密钥是通过 NASS 的公钥加密，到达 NASS 后，由 NASS 的私钥解密；当用户向 NASS 写入信息时，用户的口令认证通过后，NASS 直接将用户所发送的加密信息解密存入相应的目录中；当用户从 NASS 读信息时，用户的口令被认证通过后，NASS 将用户要

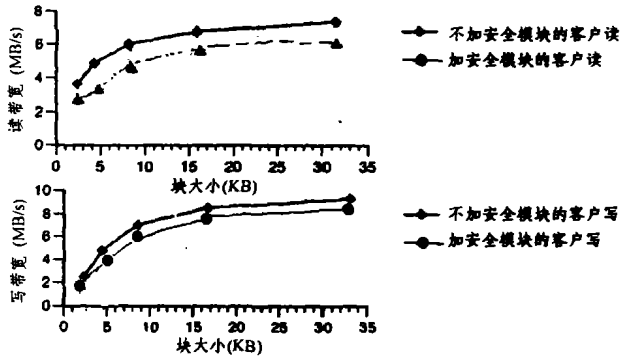

图5　NASS 加安全模块和不加安全模块的性能比较

(下转第88页)

如图4所示，ad 表示最近一次 SM-LDCQ 查询过程中静态对象到移动对象 o 之间的距离。在时间 t 内，移动对象运动的最大距离是 maxspeed×t，移动对象 mo 与静止对象间距离 ada 满足下列不等式：

$$|ad-mo.\max speed\times t|\leqslant ada\leqslant ad+mo.\max speed\times t$$

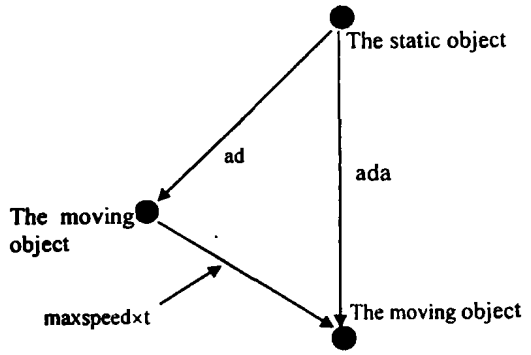

Figure 4 The maximum distance of between the objects after time t

由上面分析可得在时间 $|md-ad|/mo.\max speed$ 内移动对象 mo 一定满足 LDCQ 查询条件。对4元组〈SOID，ad，timer，ntime〉，令

$$ntime=|md-ad|/mo.\max speed$$

timer 的初值设为 t_L（对 SOID 进行 LDCQ 处理的时间点），其值随着时间而增加，当 timer 值等于 t_L+ntime 时，系统会按照4.3节的步骤(2)执行。

4.3.2 MM-LDCQ 处理中的 Min-Time 的计算方法

MM-LDCQ 功能是查询相对于移动被参照(referenced)对象的所有移动参照(referencing)对象的位置。设 ad 为某次执行 MM-LDCQ 时移动参照对象 rio 与移动被参照对象 rdo 间的距离，我们给出下列定义：

rio.maxspeed： 移动对象 rio 的最大速度

rdo.maxspeed： 移动对象 rdo 的最大速度

md： MM-LDCQ 中的查询距离

完成 MM-LDCQ 处理后 rio 与 rdo 间的距离应满足下面的不等式：

$$|ad-(rio.\max speed+rdo.\max speed)\times t|\leqslant ada\leqslant ad+(rio.\max speed+rdo.\max speed)\times t$$

在执行完 MM-LDCQ 后的时间 $|md-ad|/(rio.\max speed+rdo.\max speed)$ 内，对象 rio 一定满足 LDCQ 查询条件，所以当对象 rio 和 rdo 的位置更新时，不必重新执行 MM-LDCQ 操作。

MM-LDCQ 的查询结果内部也表示为4元组〈MOID，ad，timer，ntime〉。其中 MOID 表示移动对象，timer 是一个计数器，用以计数从最近一次对 MOID 处理完成后开始的时间，LDCQ 处理完后将 timer 值设为 t_L，timer 的值随时间增加。

$ntime=|md-ad|/(rio.\max speed+rdo.\max speed)$ 表示移动对象 MOID 能够连续满足 MM-LDCQ 条件的最小时间段。当 timer 等于 t_L+ntime 时，系统执行4.3节的步骤(2)。

结束语 为实现连续的位置相关的查询，移动计算系统要密切地跟踪移动对象的位置。使用传统的位置数据库组织模型和数据库技术，移动对象位置的频繁更新会造成无线带宽巨大的开销。本文在 MOST 数据模型的基础上提出了一种新的位置数据库组织模型，用以对移动对象动态位置建模。在该模型的基础上，我们还提出了两种 LDCQ 处理方法，以其实现减小 LDCQ 处理的 CPU 代价和无线带宽的开销。

目前，连续的位置相关查询的处理技术还远不成熟，有很多问题亟待解决。例如，对于 MM-LDCQ，被查询的对象和提交查询请求的对象都在移动，这两类移动对象间距离阈值的分布对系统性能有重要影响，这将是我们下一步要解决的问题。

参考文献

1 Wolfson O, Xu B, Chamberlain S, Jiang L. Moving objects databases: issues and solutions. In: Proc. of the 10th Intl. Conf. on Scientific and Statistical Database Management (SSDBM98), Capri, Italy, July 1998. 111～122

2 Wolfson O, et al. Cost and imprecision in modeling the position of moving objects. In: Proc. of the Fourteenth Intl. Conf. on Data Engineering (ICDE14), Orlando, FL, Feb. 1998

3 Wolfson O, et al. Updating and querying databases that track mobile units, invited paper, Special issue of the distributed and paralle databases Journal on Mobile Data Management and Applications, Kluwer Academic Publishers, 1999, 7(3): 257～287

4 Pitoura E, Samaras G. Locating objects in mobile computing. IEEE Transaction on Knowledge and Data Engineering. Accepted, 2000. To appear

5 Gok H G, Ulusoy O. Transmission Of Continuous Query Results In Mobile Computing Systems. Information Sciences, 2000, 125 (1-4): 37～63

6 Gök H G. Processing of continuous queries from moving objects in mobile computing systems: [PhD thesis]. Department of computer engineering and information science, Bilkent University, Jan. 1999

(上接第81页)

读的信息用用户同口令一起传来的密钥加密传给用户。这里用户可根据需要，定期更改口令。更改口令时，只需要把旧口令和新口令一起加密传给 NASS，NASS 就在用户口令登记表中用新口令替代旧口令。

5 初步实验结果

我们分别对 NASS 不加载安全模块的情况及加载安全模块的情况，分别测试其速度，其结果示于图5。从显示的结果看，加载安全模块写比不加载安全模块写慢7%～12%。加安全模块的读比不加安全模块的读慢11%～19%。

参考文献

1 Brocade whitepaper: comparing the Storage Area Network and Network Attached Storage

2 HITACHI whitepaper: Planning for Network Attached Storage and Storage Area Network Convergence

3 Blaze M. Key Management in an Encrypting File System. In: Proc. of the Summer 1994 USENIX Conf. 1994

4 Gobioff H. Security for a High Performance Commodity Storage Subsystem: [Ph. D. thesis]. Computer Science Department. Carnegie Mellon University, July 1999. Available as Technical Report CUS-CS-99-160

5 Krawczyk H, Bellare M, Canetti R. HMAC: Keyed-Hashing for Message Authentication. IETF Network working Group RFC2104, Feb. 1997

6 Reed B, Chron E, Burus R, Long D. Authenticating Network Attached Storage. IEEE Micro, 2000, 20(1): 49～57