基于 MTS/COM+ 的 OPC 安全机制研究

乔加新

(安徽财经大学 信息工程学院,安徽 蚌埠 233041)

摘 要:在深入分析安全机制基础:Windows 操作系统的安全机制、COM/DCOM 的安全机制和 MTS/COM+ 的安全机制基础上,提出了基于 MTS/COM+ 的 OPC 安全机制,包括安全调用机制和角色安全机制,并研究了各机制的作用范围,给出了它们程序实现的具体过程。企业不仅可以根据 OPC 接口标准统一访问系统低层资源,OPC 安全机制又可以保证 OPC 客户端与 OPC 服务器之间访问的安全性。

关键词:MTS/COM+;OPC;安全机制

中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2007)06-0151-04

Research on OPC Security Mechanism Based on MTS/COM+

QIAO Jia-xin

(School of Information Engineering, Anhui University of Finance & Economics, Bengbu 233041, China)

Abstract: In thorough analysis security mechanism: Windows operation system security mechanism, COM/DCOM security mechanism and MTS/COM+ security mechanism, the paper presents an OPC security mechanism based on MTS/COM+, including security call mechanism and role security mechanism, researches on function scope of each mechanism and gives the process of each mechanism. Enterprise not only interview the system information by OPC interface, but also ensure OPC client communication with OPC server by OPC security mechanism.

Key words: MTS/COM+; OPC; security mechanism

0 引 言

OPC (OLE for Process Control,用于过程控制的 OLE)是一个工业标准,它是许多世界领先的自动化和软、硬件公司与微软公司合作的结晶。这个标准定义了应用 Microsoft 操作系统在基于 PC 的客户机之间交换自动化数据的方法[1]。管理该标准的组织是 OPC 基金会。随着 1997 年 2 月 Microsoft 公司推出 Windows95 支持的 DCOM 技术,1997 年 9 月新成立的 OPC Foundation 对 OPC 规范进行修改,增加了数据访问等一些标准,OPC 规范得到了进一步的完善。OPC 是基于 Microsoft 公司的 Distributed interNet Application (DNA)构架和 Component Object Model(COM)技术的,根据易于扩展性而设计的。OPC 规范定义了一个工业标准接口,这个标准使得 COM 技术适用于过程控制和制造自动化等应用领域。OPC 是以 OLE/COM 机制作为应用程序的通讯标准。OLE/COM 是一种客户/服务器模式,具有语言无关性、代码重用性、易于集成性等优点。OPC 规范了接口函数,不管现场设备以何种形式存在,客户都以统一的方式去访问,从而保证软件对客户的透明性。OPC 的跨进程甚至跨网络的客户/服务器软件结构,尤其随着 Internet/Intranet 的飞速发展,则其安全性是不可缺少的保护机制。

根据 OPC 规范编写的 OPC 客户端与服务器应用程序,任何依照 OPC 接口标准实现的客户端都可以以统一的形式访问 OPC 服务器,为了保证基于微软的 OPC 系统安全性,可以利用 Windows 本身提供的安全机制。但是,Windows 本身的安全机制很复杂,建立在 Windows 操作系统,RPC 以及 COM/DCOM 安全机制的基础上的 MTS/COM+ 安全机制,无需编程人员编写大量的程序代码就可以应用系统的安全保护,为了让企业内部人员以及 Internet/Intranet 上的用户安全调用 OPC 系统提供的服务,文中在 MTS/COM+ 基础上,建立 OPC 安全机制,实现企业控制系统的安全保证。

收稿日期:2006-08-25

基金项目:安徽财经大学校青年科研项目(ACKYQ0620);信息工程学院青年科研项目(xgky2006008)

作者简介:乔加新(1975-),男,安徽蚌埠人,硕士研究生,讲师,研究方向为计算机控制、网络协议。

1 安全机制基础

实现系统的安全机制主要包括以下三个方面：Windows 操作系统的安全机制、COM/DCOM 的安全机制和 MTS/COM+ 的安全机制。

1.1 Windows 操作系统的安全机制

Windows 操作系统具有完全的保护机制，系统的所有资源都是受保护的，这些资源包括文件、外设、进程、线程，甚至同步对象、共享内存、注册表中的键等等。应用用户帐号数据库 (User Account Database) 来管理，当用户登录之后必须接受 PDC(Primary Domain Controller)的检查。客户端的操作系统会把一些用户登录的信息，例如用户名称、用户的安全标识符 SID (Security Identifier) 以及用户的机器信息等传递给 *PDC* 或 *BDC* (Backup Domain Controller)检查。

客户和服务器通过 RPC 连接进行对话，服务器就可以临时假冒该客户的身份，使得它可按该客户的权限发一个访问请求。在访问之后，服务器恢复自己的身份。因此，有必要在 RPC 层次上进行鉴定，以保护 RPC 调用。主要分为 5 个鉴定层次：无鉴定操作即正常的 RPC 调用；连接时进行 RPC 鉴定；每一个接口调用时进行鉴定；对每个请求进行鉴定，并对接收到的数据包进行完整性检验；进行所有的鉴定并对数据包加密[2]。

1.2 COM/DCOM 的安全机制

当应用程序调用 COM / DCOM 对象时，调用的应用程序就必须接受 COM / DCOM 安全机制的管理。通过 RPC(Remote Procedure Call)以及安全支持提供者 SSP(Security Support Provider)的检查。RPC 以及 SSP 会检验调用者的身份，并且根据此信息来授予激活远程对象以及存取远程对象的权利[2]。

激活安全性是进程一级的安全性，即进程中所有的对象和所有对象的成员函数共享的安全性，它分两种情况：静态安全性和动态安全性。

●静态安全性通过系统注册表实现，在系统注册表中，有两组注册表名字值用于控制对象或者组件进程的安全性：

(1) KEY_ LOCAL_ MACHINE \ Software \ Microsoft \ OLE 键下的值用于设置操作系统一级缺省安全特性；

(2)HKEY_CLASSES_ROOT \ CLSID \ {clsid} 下指定 AppID 键用于设置特定组件程序的安全特性。

●动态安全性控制通过调用 CoInitiallizeSecurity 函数，则不再使用注册表的静态设置。使用 CoInitiallizeSecurity 可以接受三种安全性设置：

(1)指定安全描述符，利用 Win32 API 函数创建安全描述符；

(2)指定进程使用特定的 AppID 中注册的安全设置，这种方式与静态安全设置结合起来；

(3)指定 IAccessControl 接口，IAccessControl 包括了设置进程允许访问或者禁止访问的各项特性。

调用安全性是指在已经建立连接的基础上，客户调用组件程序的安全特性，客户和组件程序也可以针对特定的接口指定专门的安全特性。

1)从客户端来看，客户程序通过代理对象间接调用组件程序，客户程序可在调用代理对象前控制接口调用的安全性，IClientSecurity 是实现代理选择的接口，其中 QueryBlanket 函数用于获得安全属性，SetBlanket 函数用于设置安全属性；

2)从组件程序来看，组件程序可以调用 CoGetCallContext 获得 ISeverSecurity，由 COM 提供的存根管理器实现了缺少的 ISeverSecurity 接口，利用其接口函数实现安全属性的设置与访问。

1.3 MTS/COM+ 的安全机制

MTS/COM+ 的安全机制是建立在 Windows 操作系统和 COM/DCOM 的安全机制之上，以这两者的安全机制为基础而发展出了一套新的、但是非常方便的安全机制。并用“角色”来取代它们。MTS/COM+ 提供了安全控制机制：激活控制(Activation Control)、访问控制(Access Control)、认证控制(Authentication Control)和鉴定控制(Identity Control)，其中的激活控制和存取控制是 COM / DCOM 提供的安全机制，认证控制是操作系统的安全机制，而鉴定控制则是 MTS/COM+ 额外提供的安全机制[3]。鉴定控制除了可以安全地管理系统资源的存取之外，主要的目的也是为了提供 MTS/COM+ 对于资源对象池的功能。其中组件、套件组件安全控制以及资源安全控制则是 MTS/COM+ 的主要安全控制。

2 基于 MTS/COM+ 的 OPC 安全机制

基于 MTS/COM+ 的 OPC 安全机制，以 MTS/COM 的安全机制为基础，主要包括安全调用机制和角色安全机制来提供 OPC 系统的安全保证。

2.1 安全调用机制

MTS/COM+ 所定义的安全调用模型是由一组相关的 COM 接口所定义的。由于 MTS/COM+ 的安全模型只是一组相关的 COM 接口，所以当其执行在不同的操作系统时，操作系统就会实现这一组安全接口，并提供 MTS/COM+ 在这个操作系统中的安全控管[4,5]。

(1)激活控制：是指控制哪些 OPC 客户端可以激

活 OPC 服务器。当 OPC 客户端试着在远程机器上激活 OPC 服务器时,MTS/COM+ 的安全机制会检查该客户端是否有权限激活远程服务器的安全控制。

(2)访问控制:是指在 OPC 服务器上由适当的 OPC 客户端激活后,控制哪些 OPC 客户端可以存取 OPC 服务器所提供的服务。

(3)认证控制:主要是控制 OPC 客户端和 OPC 服务器端之间传递数据的安全性。严格的认证控制可以让数据在传输时先经过加密,以避免无权限用户的访问。同时必须决定使用哪一层次的数据安全性,当然设置的安全层次越高,数据越安全,但是程序执行的效率就会受到影响。

(4)鉴定控制:是指一个 MTS/COM+ 套件组件中的 OPC 服务器在需要访问其他的系统低层资源时,使用什么系统帐号来登陆到这些资源。只有使用一个合法的鉴定控制身份来登陆这些资源,才能取得它需要的资源。鉴定控制能够控制 OPC 服务器是使用特定的用户权限执行的,或是使用目前在 OPC 客户端登陆的用户的权限执行的。

2.2　角色安全机制

为了减少在实现 OPC 安全机制时花费大量时间,利用 MTS/COM+ 提供了一个新的机制来管理系统安全的问题,即"角色"(Role)。所谓"角色"就是代表一群用户的字符串名称[4,5]。在 MTS/COM+ 中,可以为每一个 OPC 服务器设定一群可以访问它的"角色"。因此在 MTS/COM+ 中要访问 OPC 服务器的安全机制,就等于先建立一群"角色",然后在每一个"角色"中加入相对的用户,最后再把不同的"角色"指定给不同的 OPC 服务器。当一个用户要访问远程 OPC 服务器时,利用 MTS/COM+ 会检查指定给它的"角色",然后再检查这个用户是否属于任何指定给它的"角色"。如果用户属于其中一个指定给它的"角色",那么用户就可以访问这个 OPC 服务器,否则就会发生存取错误。

如果需要更为精确地控制访问的安全,可以用程序代码来做更详细的控制。MTS/COM+ 允许以三种不同的层次来设定什么"角色"可以存取什么 OPC 服务器:

(1)组件层次:只在 OPC 客户端访问 OPC 服务器时接受检查,一旦 OPC 客户端通过组件层次的检查,那么用户就可以访问 OPC 服务器的任何接口以及调用接口中的任何方法。

(2)接口层次:OPC 客户端不但在访问 OPC 服务器时接受检查,而且访问 OPC 服务器的任何接口时也要接受检查。因此为 OPC 服务器不同的接口,设定不同的"角色"。

(3)方法层次:即 OPC 客户端在每次调用 OPC 服务器的任何方法时都必须接受角色检查。这个层次的安全控制非常严格,允许对 OPC 服务器做非常精确的设定。但是这个安全层次也是执行速度最慢的,因为每一个远程调用都需要进行一次角色检查。

"角色"安全机制有两种建立方式:一是宣告式安全机制。使用宣告式安全机制的好处是方便、有弹性。因为系统管理人员可以随时根据目前的需要而动态地设定 MTS/COM+ 组件的安全存取机制。不过这也可能会造成系统的问题,因为系统管理人员可以随时改变安全机制;二是程序代码控制安全机制。必须撰写一些和安全访问机制相关的程序代码,使用程序代码控制安全机制,程序员可以对于 MTS/COM+ 做最精确的安全管制,也可以使用程序代码控制安全机制来辅助宣告式安全机制的不足之处。

3　OPC 安全机制的程序实现

以轻工行业的啤酒生产公司为例,包括为生产提供动力的动力车间,生产过程中起重要作用的酿造车间以及对产品包装的灌装车间,在动力车间又有水处理控制系统和锅炉控制系统,酿造车间有糖化过程控制系统和大罐冷却控制系统,灌装车间有洗瓶系统、酒机 PLC 系统和打包机控制系统。为了实现企业集成控制系统,利用 OPC 的接口的规范化,按照一种统一的方式访问各低层控制系统。由于该公司生产工艺复杂,保密性强,一般员工只允许访问生产工艺流程的某些资源,这就要求为系统设置很严格的权限管理,在本系统中采用基于 MTS/COM+ 的 OPC 安全机制来保证系统的安全性,可以在 OPC 服务器的组件层次、接口层次以及方法层次设定安全存取的角色。当一个 OPC 客户端应用程序调用远程 OPC 服务器,在通过 MTS/COM+ 的激活和存取安全检查后,MTS/COM+会检查该 OPC 客户端的角色权限。基于 MTS/COM+ 的 OPC 安全机制模型如图 1 所示。角色定义时,如果以产品等实体为角色,每一种产品就要加很多角色和接口才能保证系统的安全性,因此,以产品为角色是不现实的。本系统是采用车间、小组为角色。

结合宣告式和程序代码两种安全机制的优点,用 MTS/COM+ 为 OPC 系统设置安全性,其过程如下[4,5]:

(1)启用 MTS/COM+ 的安全机制,必须先激活 MTS/COM+ 的安全访问检查。要激活 MTS/COM+ 的安全访问检查,执行 MTS Explorer 或 COMponent Services,点选要设定的 OPC 服务器,再点选鼠标右键。接着点选快捷菜单中的内容(Properties)选项,并切换

到安全设定（Security），OPC 客户端应用程序访问此 OPC 服务器时，就必须通过 MTS/COM+ 的访问和激活的安全检查，并且也必须接受稍后介绍的角色存取检查。除了设定安全等级之外，也可以设定认证控制等级。认证等级分为：无（ None ）、联机（Connect）、调用（Call）、封包（Packet）、封包完整性（Packet Integrity）、封包保密性（Packet Privacy）。

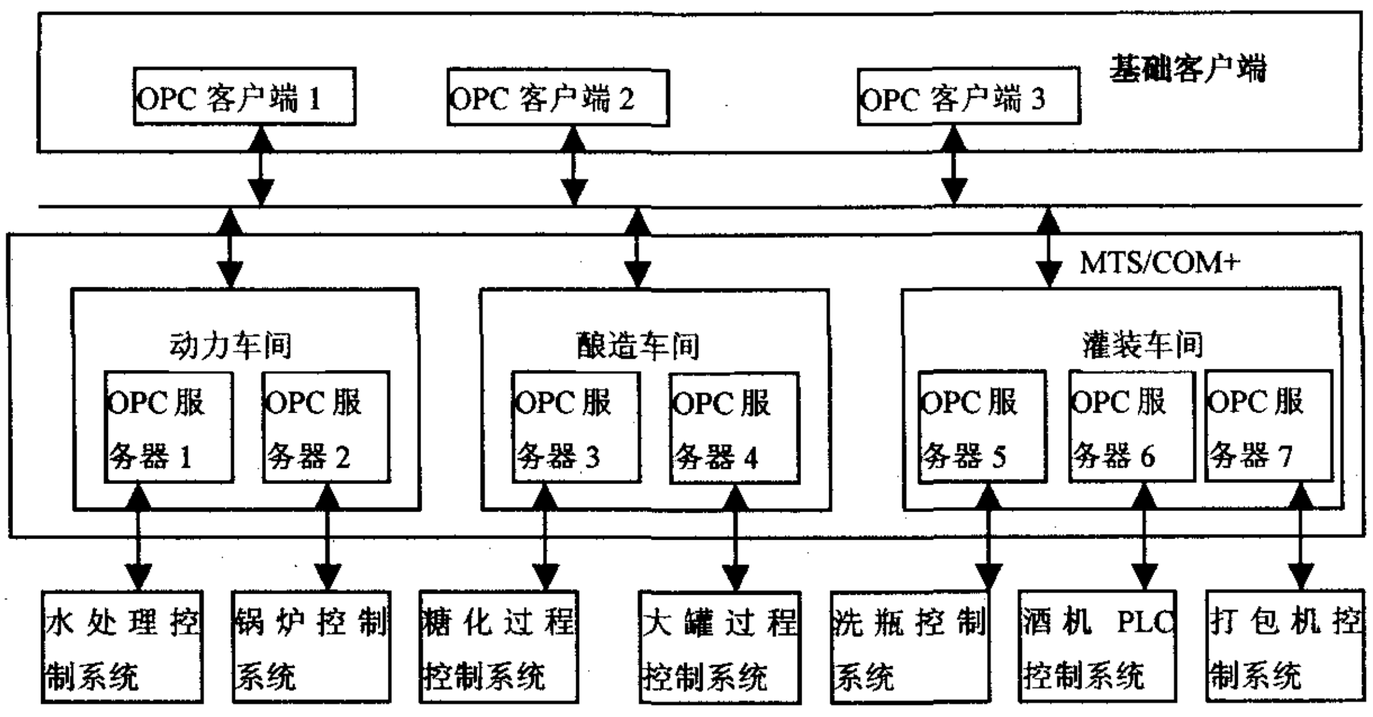

图 1 基于 MTS/COM+ 的 OPC 安全机制模型

(2)定义每一个组件、接口以及方法以便能够合法访问角色，由于 MTS/COM+ 是使用角色作为控制安全机制的单位，因此在定义系统合法的角色之前必须使用操作系统的系统管理工具定义用户或用户群组。然后再建立角色并把用户和用户群组加入到适当的角色中。

(3)把 Windows 操作系统的用户或用户群组加入到 MTS/COM+ 的角色中。使用操作系统的管理工具定义用户或用户群组。然后再为系统建立各种角色：为动力车间定义水处理角色和锅炉角色，为酿造车间定义糖化角色和大罐冷却角色，为灌装车间定义洗瓶角色、酒机角色和打包机角色，然后使用 MTS·Explorer 或 Component Services 把 Windows 操作系统的用户或用户群组加入到 MTS/COM+ 的角色中。

(4)指定角色到组件、接口和方法中。在组件层次上，指定动力角色给动力车间组件，酿造角色给酿造车间组件，灌装角色给灌装车间组件。指定角色以后，所有属于指定角色的用户都可以存取相应车间 OPC 服务器提供的接口和方法。在接口层次上，由于糖化温化工艺和大罐冷却工艺安全性要求很高，在 OPC 服务器 3 和 OPC 服务器 4 的接口层次也进行设定安全机制。最后，在方法层次上，如冷却接口有两个方法：冷却设置和冷却信息。冷却温度设置参数为保密技术的信息，同时对会产品质量有重大影响，希望只有工艺设计师才可以调用该方法。所以，只把高级酿造角色指定给该方法。将酿造角色指定给冷却信息方法。

整个系统配置完后，运行某一 OPC 客户端时，例如以属于酿造角色的用户“XX”登录时，客户可以通过与酿造车间组件相关 OPC 服务器的检查，访问酿造车间组件的信息。对于属于不同组件层次、接口层次和方法层次上的角色用户根据系统的安全设置进行安全访问，实现整个企业的 OPC 系统的安全运行。

4 结束语

利用 MTS/COM+ 的角色机制和调用安全机制，实现企业 OPC 系统的安全访问，企业不仅可以根据 OPC 接口标准统一访问系统低层资源，“角色”机制又可以保证 OPC 客户端与 OPC 服务器之间访问的安全性，保证企业安全运行。同时也为 OPC 规范在企业中进一步使用提供了安全保证。

参考文献：

[1] OPC Foundation. OPC Data Access Custom Interface Specification Version 2.05[S]. 2001.

[2] 潘爱民. COM 原理与应用[M]. 北京：清华大学出版社，1999.

[3] David. 深入理解 COM+[M]. 潘爱民译. 北京：清华大学出版社，2000.

[4] 李 维. Delphi5. X 分布式多层应用——高级程序设计篇[M]. 北京：机械工业出版社，2000.

[5] 刘才德，吕汉兴. 基于 MTS/COM+ 安全机制的设计与实现[J]. 计算机工程，2002(3)：137－138.

(上接第 126 页)

aided by UML[R/OL]. UK: School of Electronics and Computer Science, University of Southampton. 2004. eprints. ecs. soton. ac. uk/10169/.

[4] Laleau R, Mammar A. An overview of a Method and its support tool for generating B specification from UML notations [C]//The Fifteenth IEEE International Conference on Automated Engineering. [s. l]: [s. n.], 2000.

[5] Ledang H, Souquières J. Modeling class operations in B: a case study on the pump component[R/OL]. A01－R－011, Laboratoire Lorrain de Recherche en Informatique et ses Applications, 2001－03. http://www. loria. fr/Vledang/publications/UML01. ps. Z.

[6] Leuschel M A, Butler M, Lo Presti S. ProB User Manual[EB/OL]. 2005. //http://www. ecs. soton. ac. uk/mal/systems/prob. html.