整性，我们需要更精确的逆向特性（Biba，1977）。

1) 简单完整性原理：在安全等级k上运行的进程只能写同一等级或更低等级的对象（没有往上写）。

2) 完整性*规则：在安全等级k上运行的进程只能读同一等级或更高等级的对象（不能向下读）。

这些特性联合在一起确保了程序员可以根据公司总裁的要求更新看门人的信息，但反过来不可以。当然，有些机构想同时拥有Bell-La Padula和Biba特性，但它们之间是矛盾的，所以很难同时满足。

9.3.8 隐蔽信道

所有的关于形式模型和可证明的安全系统听上去都十分有效，但是它们能否真正工作？简单说来是不可能的。甚至在提供了合适安全模型并可以证明实现方法完全正确的系统里，仍然有可能发生安全泄露。本节将讨论已经严格证明在数学上泄露是不可能的系统中，信息是如何泄露的。这些观点要归功于Lampson（1973）。

Lampson的模型最初是通过单一分时系统阐述的，但在LAN和其他一些多用户系统中也采用了该模型。该模型最简单的方式是包含了三个运行在保护机器上的进程。第一个进程是客户机进程，它让某些工作通过第二个进程也就是服务器进程来完成。客户机进程和服务器进程不完全相互信任。例如，服务器的工作是帮助客户机来填写税单。客户机会担心服务器秘密地记录下它们的财务数据，例如，列出谁赚了多少钱的秘密清单，然后转手倒卖。服务器会担心客户机试图窃取有价值的税务软件。

第三个进程是协作程序，该协作程序正在同服务器合作来窃取客户机的机密数据。协作程序和服务器显然是由同一个人掌握的。这三个进程如图9-14所示。这一例子的目标是设计出一种系统，在该系统内服务器进程不能把从客户机进程合法获得的信息泄露给协作进程。Lampson把这一问题叫做界限问题（confinement problem）。

从系统设计人员的观点来说，设计目标是采取某种方法封闭或限制服务器，使它不能向协作程序传递信息。使用保护矩阵架构可以较为容易地保证服务器不会通过进程间通信的机制写一个使得协作程序可以进行读访问的文件。我们已可以保证服务器不能通过系统的进程间通信机制来与协作程序通信。

遗憾的是，系统中仍存在更为精巧的通信信道。例如，服务器可以尝试如下的二进制位流来通信：要发送1时，进程在固定的时间段内竭尽所能执行计算操作，要发送0时，进程在同样长的时间段内睡眠。

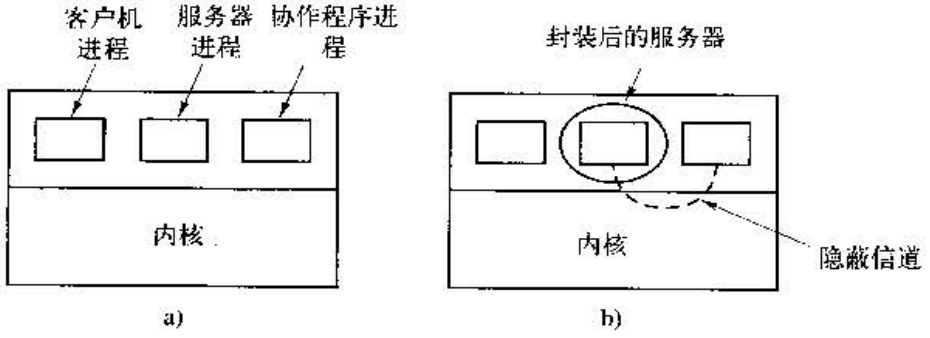

图9-14 a) 客户机进程、服务器进程和协作程序进程；b) 封装后的服务器可以通过隐蔽信道向协作程序进程泄露信息

协作程序能够通过仔细地监控响应时间来检测位流。一般而言，当服务器送出0时的响应比送出1时的响应要好一些。这种通信方式叫做隐蔽信道（covert channel），如图9-14b所示。

当然，隐蔽信道同时也是嘈杂的信道，包含了大量的外来信息。但是通过纠错码（如汉明码或者更复杂的代码）可以在这样嘈杂的信道中可靠地传递信息。纠错码的使用使得带宽已经很低的隐蔽信道变得更窄，但仍有可能泄露真实的信息。很明显，没有一种基于对象矩阵和域的保护模式可以防止这种泄露。

调节CPU的使用率不是惟一的隐蔽信道，还可以调制页率（多个页面错误表示1，没有页面错误表示0）。实际上，在一个计时方式里，几乎任何可以降低系统性能的途径都可能是隐蔽信道的候选。如果系统提供了一种锁定文件的方法，那么系统就可以把锁定文件表示为1，解锁文件表示为0。在某些系统里，进程也可能检测到文件处于不能访问的锁定状态。这一隐蔽信道如图9-15所示，图中对服务器和协作程序而言，在某个固定时间内文件的锁定或未锁定都是已知的。在这一实例中，在传送的秘密位流是11010100。