



---

# TPMc: A New TPM System for Trusted Cloud Computing

代炜琦  
华中科技大学

---

服务计算技术与系统教育部重点实验室 (SCTS)  
集群与网格计算湖北省重点实验室 (CGCL)

# VMG

- TPM已经在计算机系统中得到了广泛的应用。
  - 保障计算机启动流程的安全，维护系统软件栈的完整性，进行对系统进行远程证明，安全地密封数据等等
- 虚拟机组(VMG) :这是因为在云计算环境中，用户往往有多台需要一同协作的虚拟主机，同时这些用户往往也需要和其他用户协作来共同完成一些任务。
- 虽然vTPM与能将物理主机虚拟为多个虚拟主机的主机级别的虚拟机技术相兼容，但vTPM并不足以支持通常单个用户会有多台虚拟机构成的虚拟机组（VMG）的云计算级别的虚拟化技术。为了保证云服务的安全性，必须针对VMG构建一个整体可信的执行环境。

# Background

## ● vTPM

**vTPM**由**IBM**研发，用于解决传统**TPM**模块无法直接应用与虚拟化平台的问题，成功地将可信计算平台带到了**Xen**虚拟机当中。传统基于**TPM**开发的软件可以基本无障碍地运行在**vTPM**环境下，这对于可信计算的应用起到了促进作用。

**vTPM**是伴随着**DomU**建立而建立，销毁而销毁

。

# Background

## ● TCG Software Stack

- ❖ **TCG**协议栈（**TSS**），从应用上来讲也就是一般应用程序用来访问以及利用**TPM**功能的一套接口协议。
- ❖ 在**Linux**中，常用的**TSS**的实现是**TrouSerS**.通过**TrouSerS**用户应用程序可以直接利用**TSS API**去访问**TPM**模块。

# Background

## ● NVM (Non-volatile memory)

- ❖ NVM (Non-volatile memory) 是一系列非易失性存储器的统称。在TPM中用于存放一些断电之后保存的信息，例如SRK等。
- ❖ vTPM由于是软件虚拟实现，其NVM由Dom0文件系统中的文件所取代。其功能与物理TPM中的NVM类似。

# Background

## ● vTPM\_manager

- ❖ **vtpm\_manager**维护一个**vTPM**实例和一个**VM**之间的关系。在**Dom0**中，每一个从**DomU**发来的**TPM**请求都要经历接收、存储、转发、处理、再发送这一完整流程。其中最为关键的就是存储转发功能。因此需要一个可以接收并处理**TPM**指令的程序，它既可以对**TPM**指令进行识别，判断类型，还可以辨清并转发不同**DomU**送来的**TPM**指令到不同的**vTPM**进行处理。这就是**vtpm\_manager**的主要功能。

# Background

## ● NFS

- ❖ **NFS (Network Filesystem)** 是由**Sun Microsystems**开发的一种网络文件系统协议。它的存在使得多台**Linux**主机共享文件系统成为可能。在**NFSv4**协议中，还引入了**Lock File**机制，这是的跨主机的进程同步成为了可能。

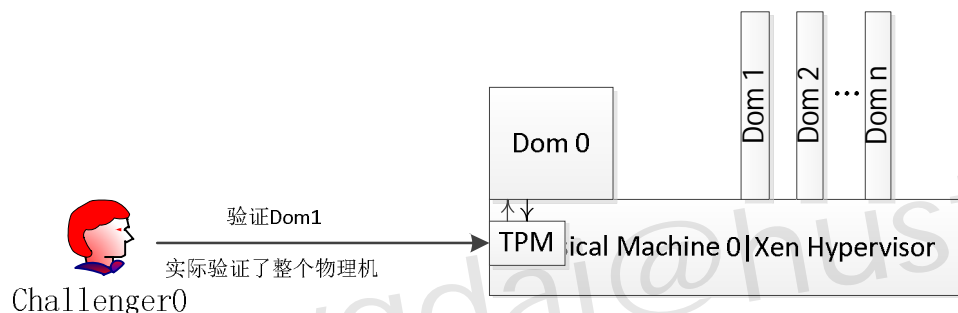
## 传统可信计算平台存在的问题

- 硬件TPM模块无法满足大量虚拟机的需求

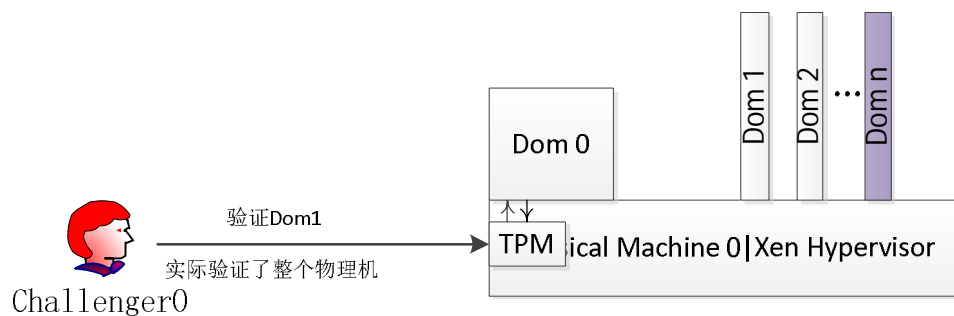
通常情况下，一个硬件TPM与一台物理计算机绑定。而在这台物理计算机上运行虚拟机则无法享受到TPM带来的相关技术支持，即便有，也只是通过物理机上的TPM相关应用实现，这极大地限制了TPM的作用。一旦某个验证者希望验证某台虚拟机的安全状况，它只能对整个物理计算机进行远程证明。这会带来问题。



# 传统TPM下的虚拟机

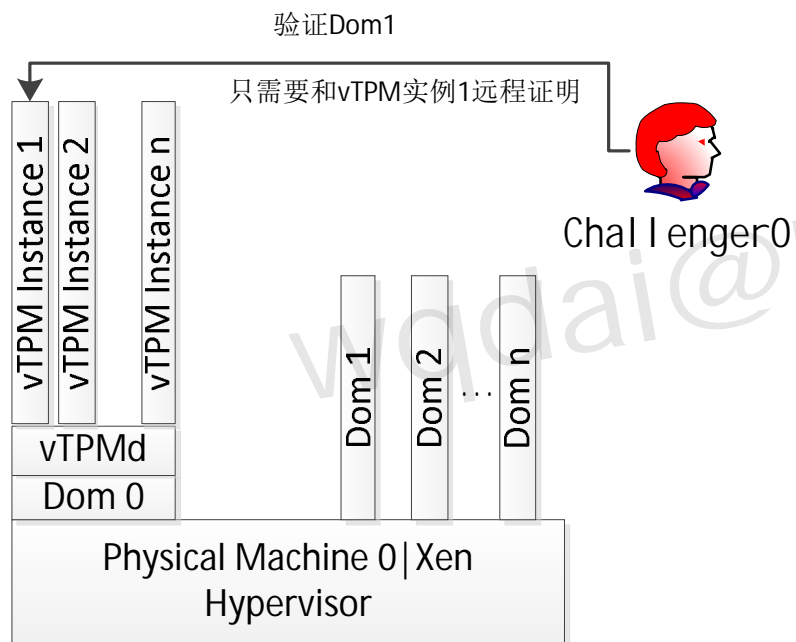


当一个远程验证这想验证 Dom 1 的安全状态，他将只能验证整个物理机。当这些虚拟机的状态都符合要求时，验证将会成功。



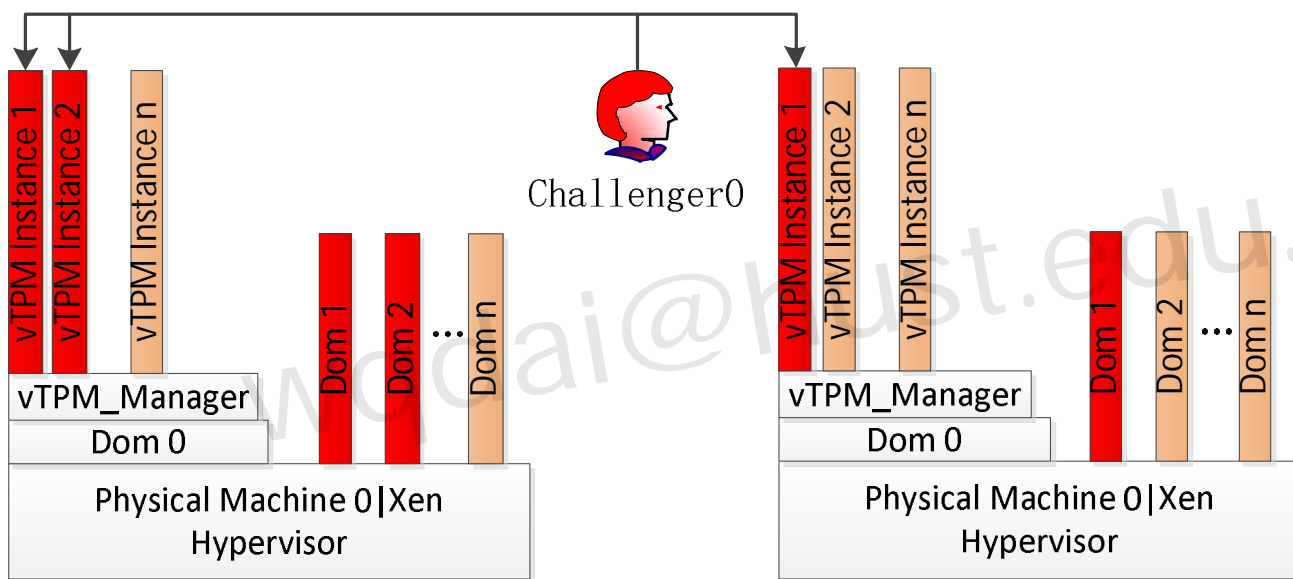
然而，如果一台虚拟机的状态不符合要求，如图中的 Dom n，此次远程证明将失败。然而事实上，远程证明方想要验证的 Dom1 的状态是安全的。

# vTPM所带来的变化



在使用vTPM的系统中，每一个使用vTPM的虚拟机都有一个与其对应的vTPM实例。当远程用户要验证Dom 1是否处于安全状态时，只需要对vTPM 实例1进行远程证明即可。

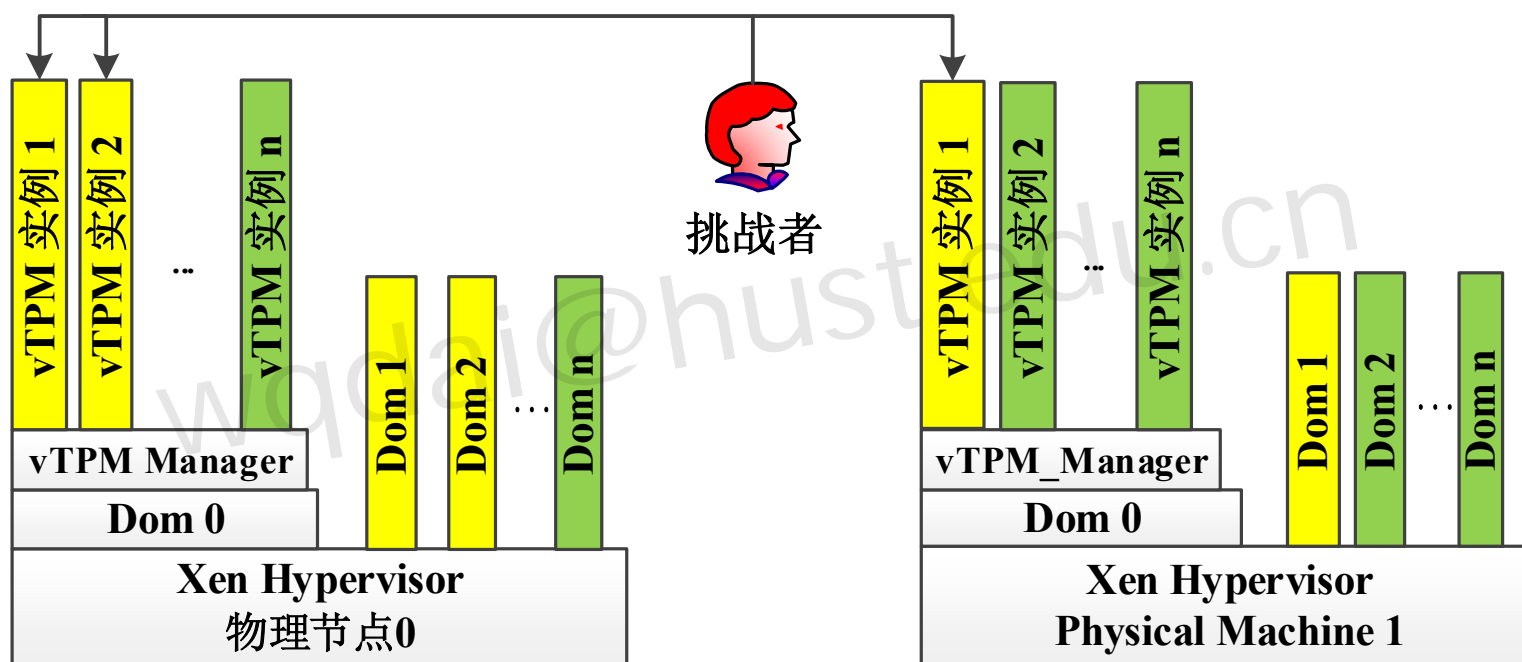
## vTPM下对域进行远程证明



如图所示，同样颜色的虚拟机表示属于同一个TVD。当远程用户希望验证红色TVD是否处于安全状态时，他只能分别对这个TVD中的三个虚拟机所属的vTPM实例提出远程证明。这个过程将非常耗

时，也增加了相关应用的设计难度。

# 云计算环境下vTPM的缺陷



- **性能缺陷。** 每台虚拟机只会将自己的安全状态以特定的顺序扩展到PCR中。当挑战者需要验证这一组虚拟机的安全状态时，计算复杂度和通讯开销都都与虚拟机数目成线性相关，无论对于服务端还是客户端而言。同时这些开销会进一步被远程证明的次数所放大。

## vTPM的缺陷2

- 安全缺陷：在一个接一个的远程证明过程中，某一台虚拟机也许会更改其PCR数值（重启虚拟机），然后在远程证明结束前还原PCR数值。这代表着从一个恶意状态，在远程证明进行时转换为安全状态，然后在远程证明完成之后还原回恶意状态的一个过程。在vTPM架构下，这种攻击形式是无法被检测到的。
- 传统单个虚拟机可以通过信任链记录软件服务启动顺序，对于一组虚拟机也同样需要确保虚拟机内服务的启动顺序，比如防火墙服务、蜜罐之类的安全服务需要最先启动，然后启动其他SaaS虚拟机的服务，如果顺序不对，虚拟机组就有被攻击的危险。

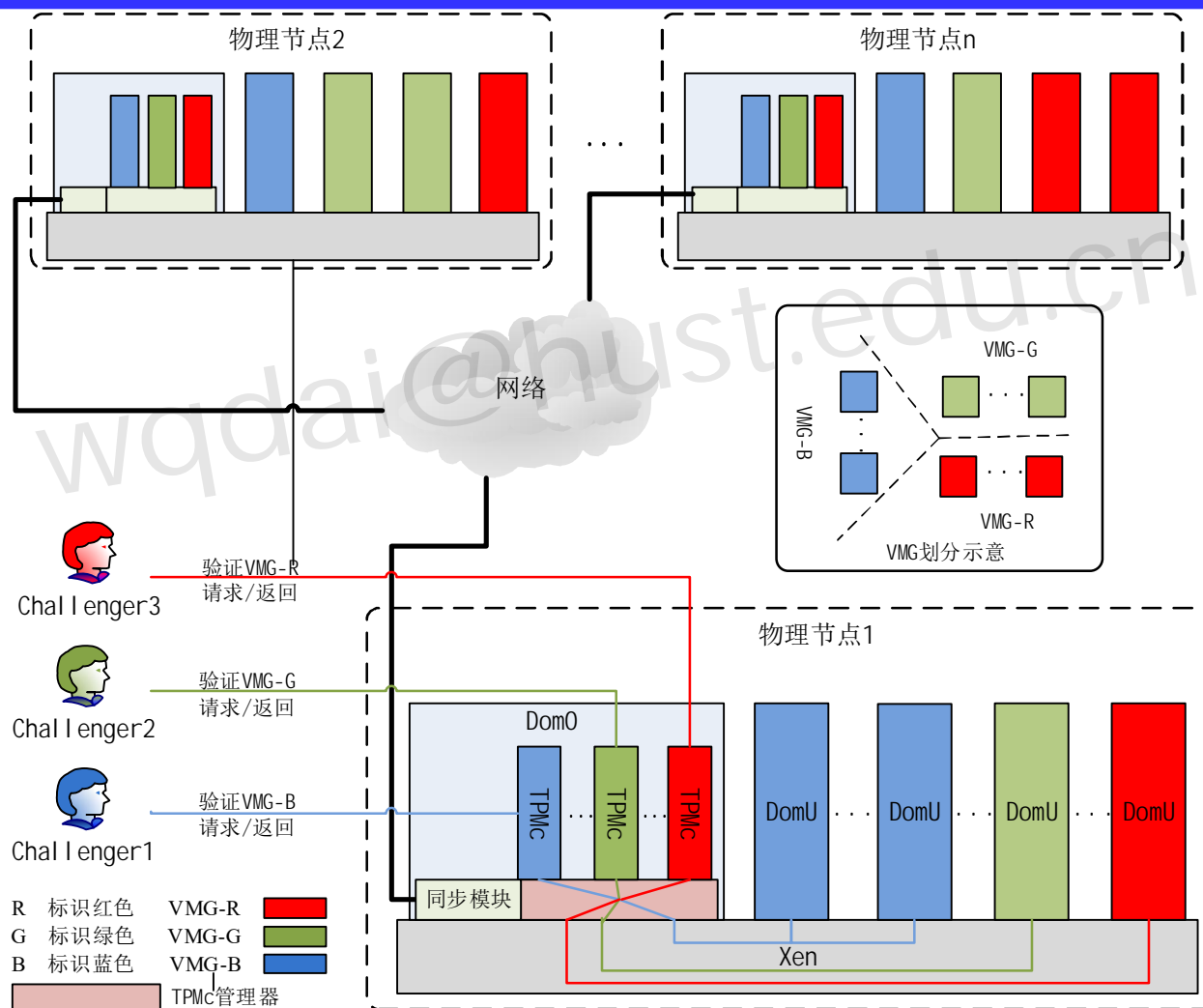
## vTPM的缺陷3

- 云中多虚拟机的时序问题：一组虚拟机，分别为VMa, VMb, ... VMg. 其中VMa运行GMail服务，VMb运行Twitter服务，每一台这样的虚拟机都负责接受和发送以及删除来自某项服务的信息。假设此时有一台特别的虚拟机，被定义VMz,它负责将未读的信息，邮件，Twits发送给用户的智能手机。一旦被发送，这些将会被mark为已读，并且VMa... VMg将会将这些信息删除。假设这有一个时间顺序，  
(i)VMa..VMg收到邮件/信息/Twits. (ii)VMz将所有信息发送给用户。  
(iii)VMa...VMg删除所有信息。在某些情况下，这个顺序也许会被打乱(i)->(iii)->(ii)。即一些信息会在用户读之前删除。由于每个vTPM实例服务一台VM，他们并不能互相知道彼此的顺序。这些vTPM中的PCR将可能以任意可能的执行顺序被扩展。
- 安全状态是不同的，在传统vTPM中只能知道VM都到了对应的A和B状态。但操作的顺序错误可能带来安全隐患。

## vTPM的缺陷4

- 一个VMG，被一台VM及其所属的vTPM所seal的数据无法被与它协作的另一台VM去unseal，即便这两台虚拟机都分配给了同一个云计算用户。这限制了虚拟机间数据共享。
- 假设VMG中多台虚拟机希望能够访问共享存储，他们需要运行一套基于密码学的密钥管理协议，基于此架设多台主机间的加密信道。多用户的密码管理的资源开销非常的大。

# TPMc系统架构

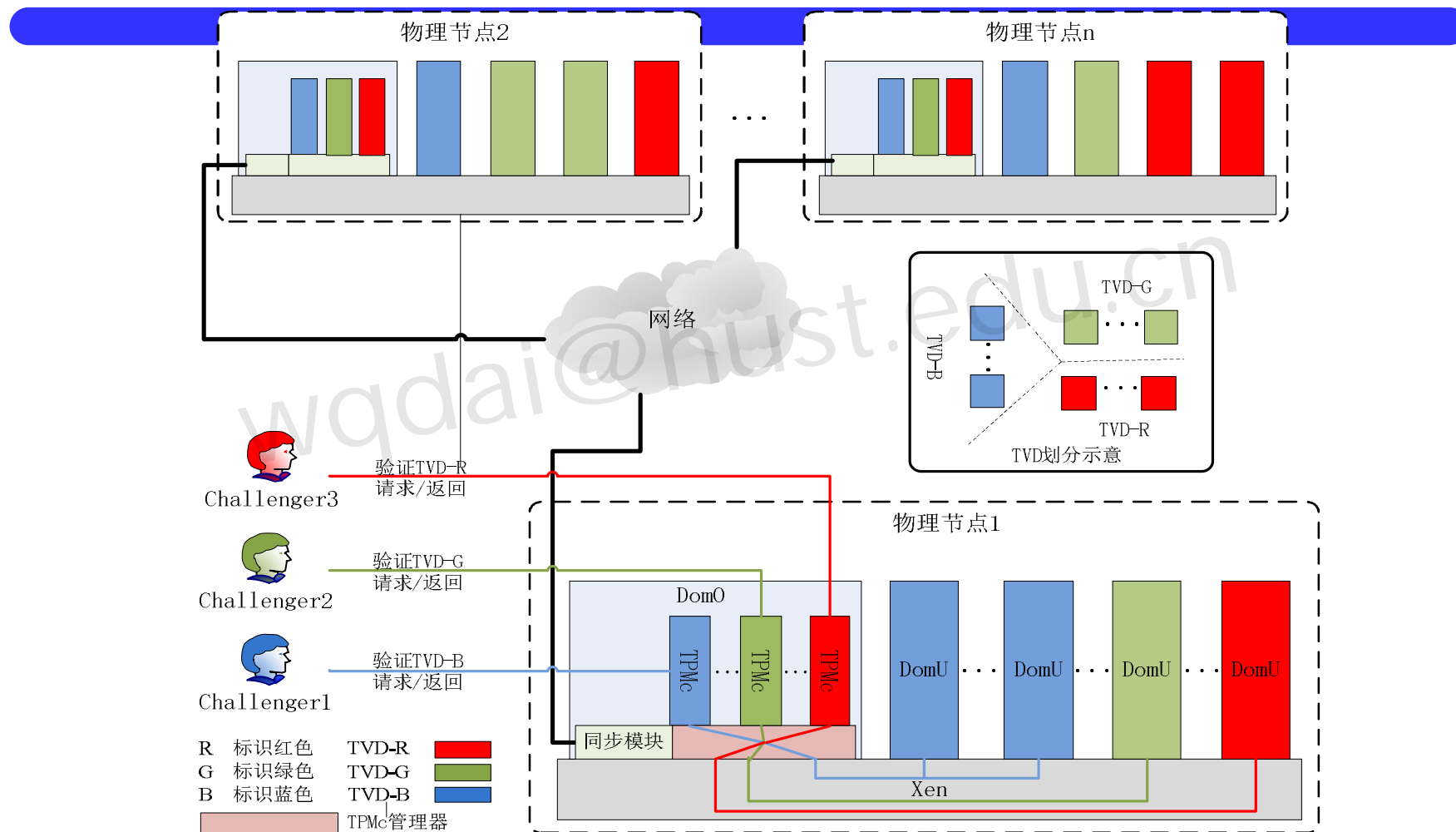




### — 需解决问题：

- 如何使跨物理节点的TVD公用TPMc?
- 用户如何请求使用TPMc?
- 公用TPMc时如何解决多个请求的冲突和TPMc的同步问题（每个虚拟机看到的TPMc都是相同的）？
- 每个用户使用TPMc创建的密钥如何共享？
- 一个TPMc如何体现整个TVD的安全性？
- TVD如何识别域中的虚拟机？
- 虚拟机如何区分使用TPMc和自己的vTPM？

# 1. 如何使跨物理节点的TVD公用一个TPMc?



如上图所示，相同颜色的虚拟机构成一个TVD，对应相同颜色的TPMc。

由于TPMc和普通的vTPM的功能相同，唯一的不同的就是一个TPMc被多个虚拟机使用，而一个vTPM实例只被一个虚拟机使用。因此可以使用vTPM实例作为TPMc。TPMc可以像普通vTPM实例那样通过TPMc管理器（修改过的vTPM\_Manager）创建。TPMc部署在每个物理节点中的Dom0，接下来我们需要解决的是这样部署带来的问题。

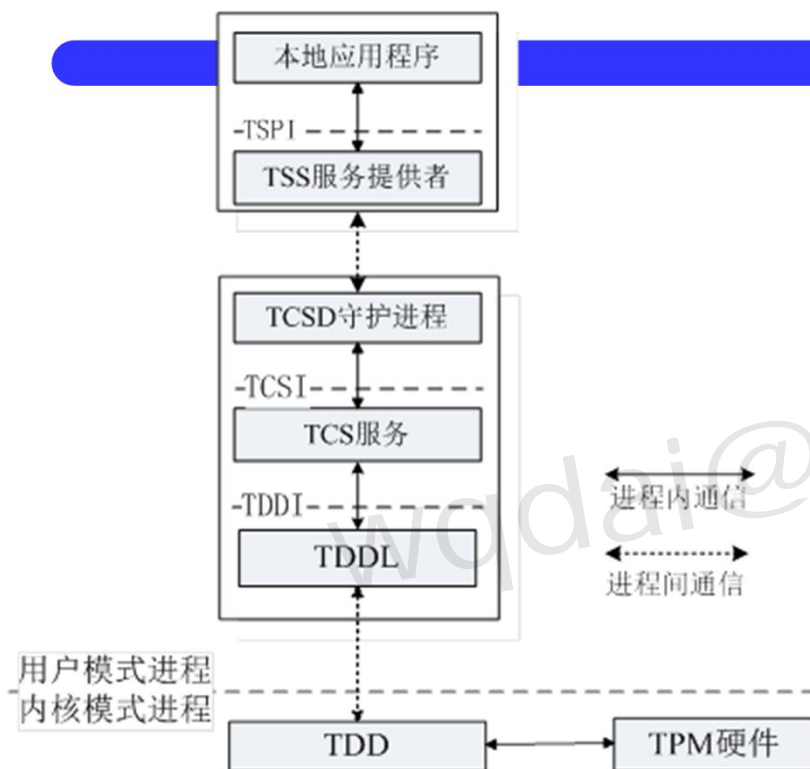
## 2.用户如何请求使用TPMc?

用户请求使用传统的物理TPM或者vTPM需要通过TCG软件栈。由于TPMc只是一个特殊的vTPM，所以为了保持兼容性，用户对TPMc的使用也必须遵循TCG规范，使用TCG软件栈。用户请求使用TPMc时，发送命令到虚拟机所在物理节点的Dom0中的TPMc。

### 3. 公用TPMc时如何解决多个请求的冲突和TPMc的同步问题？

由于我们是使用vTPM作为TPMc，vTPM不存在多个虚拟机同时发送命令的情况而且各个vTPM实例之间是相互独立的，所以原有的vTPM自身无法解决多个请求冲突的问题，也无法解决多个实例之间同步的问题。

将TVD中每个物理节点上的TPMc对应的NVM文件都存放到一台nfs服务器上（假设该服务器是安全的）。当有用户需要使用TPMc处理TPM命令时，TPMc首先从nfs上加载该文件到内存中，然后执行tpm密令，最后将TPMc的内存写到文件中。整个过程利用nfs固有的锁和排队机制，避免冲突。同时这个方法也实现了TPMc之间的同步，保证了各个虚拟机请求的TPM命令执行前，看到的TPMc是最新的。

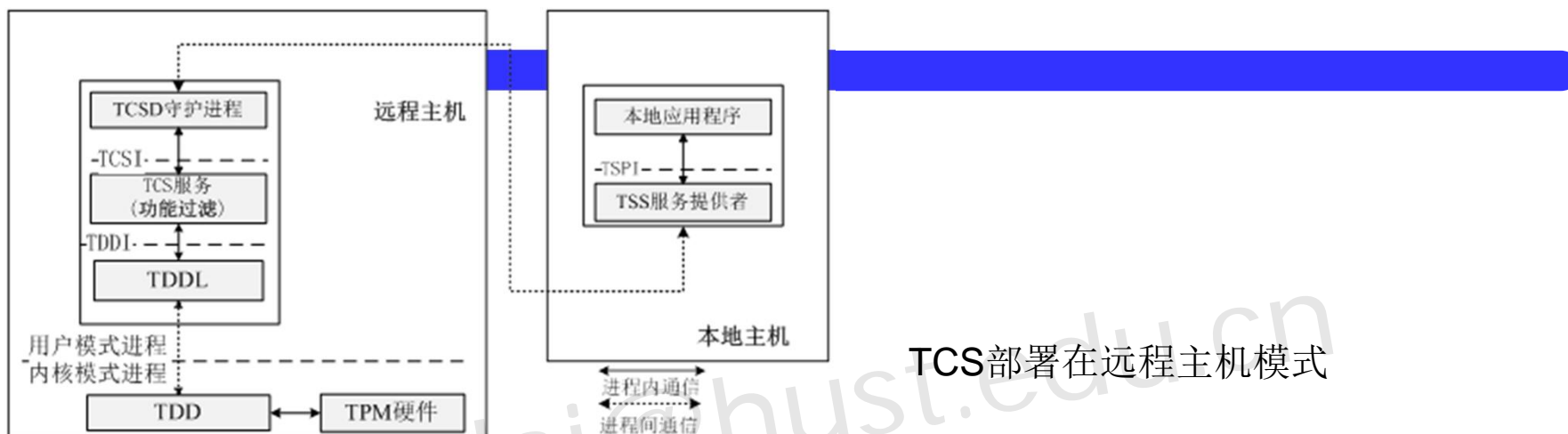


TCS部署在本地模式

#### 4. 每个用户使用TPMc创建的密钥如何共享？


如前面所说，用户使用虚拟机通过TCG软件栈使用TPMc。密钥是由TCG软件栈的TCS维护的。只要将TCG维护的密钥放到NFS上就可以实现共享。但因此会带来新的问题。

这个问题与TCG的架构有关系。TCG软件栈中的TCS既可以部署在本地也可以部署在远端



(1) 若采用TCS部署在本地模式，意味着每个虚拟机都部署TCS，这样用户通过TPMc产生的密钥就保存在DomU中。由于我们使用了nfs，所以在这种部署方式下，将DomU中TCS维护的密钥都放到nfs中实现共享。但是这就意味着DomU对nfs有修改的权限，这显然是不安全的（DomU不可信）。所以这种部署方式不合适。





(2) 采用TCS部署在远端的方式如何呢？由于整个TVD中，可信的只有Dom0和nfs，所以TCS只能部署在Dom0或者nfs中。考虑到TCS是TCG软件栈的一部分，需要和TPMc通信，所以将TCS部署在Dom0更加方便和安全。然后将Dom0中TCS维护的密钥放到nfs上来实现共享。

# 跨物理节点的VMG公用TPM<sub>Mc</sub>

由于TPM<sub>Mc</sub>和普通的vTPM的功能相同，唯一的不同就是一个TPM<sub>Mc</sub>被多个虚拟机使用，而一个vTPM实例只被一个虚拟机使用。因此可以使用vTPM实例作为TPM<sub>Mc</sub>。TPM<sub>Mc</sub>可以像普通vTPM实例那样通过TPM<sub>Mc</sub>管理器（修改过的vTPM\_Manager）创建。修改vTPM\_manager成为TPM<sub>Mc</sub>管理器，在原有vTPM\_manager的3个线程基础上新建一个线程，监控所有使用TPM<sub>Mc</sub>实例的TPM命令，然后转发给相应的TPM<sub>Mc</sub>实例。

# TPMc的同步问题

- 将VMG中每个物理节点上的TPMc对应的NVM文件都存放到一台nfs服务器上（假设该服务器是安全的）。当有用户需要使用TPMc处理TPM命令时，TPMc首先从nfs上加载该文件到内存中，然后执行tpm命令，最后将TPMc的内存写到文件中。整个过程利用nfs固有的锁和排队机制，避免冲突。同时这个方法也实现了TPMc之间的同步，保证了各个虚拟机请求的TPM命令执行前，看到的TPMc是最新的。这种方式任意Domain 0都无法知道VMG的分布状况。

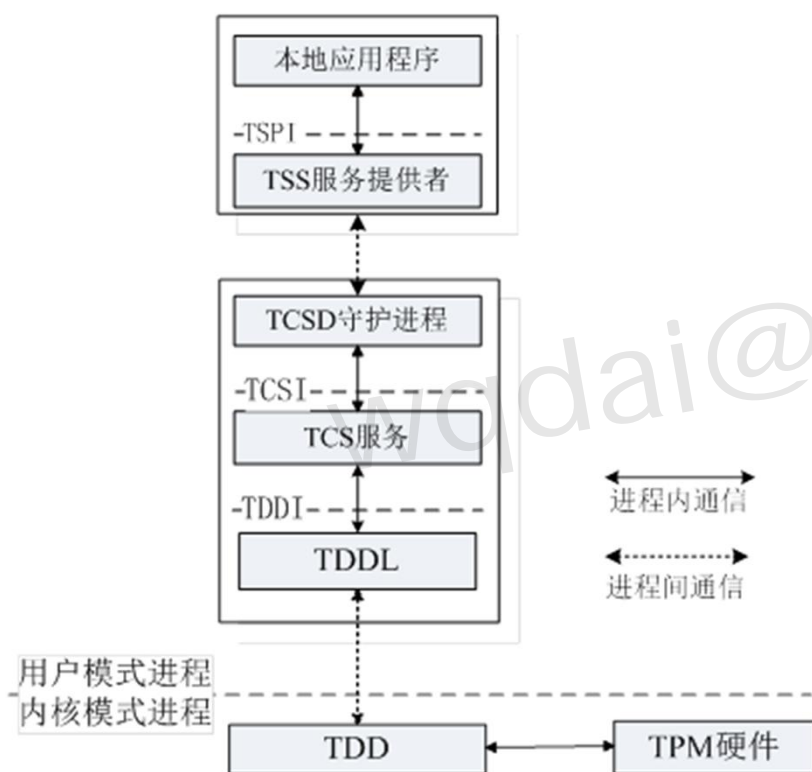
# VM的密钥共享

每个VM使用TPM<sub>CM</sub>创建的密钥如何共享？

如前面所说，用户使用虚拟机通过TCG软件栈使用TPM<sub>CM</sub>。密钥是由TCG软件栈的TCS维护的。只要将TCG维护的密钥放到NFS上就可以实现共享。但因此会带来新的问题。

这个问题与TCG的架构有关系。TCG软件栈中的TCS既可以部署在本地也可以部署在远端。

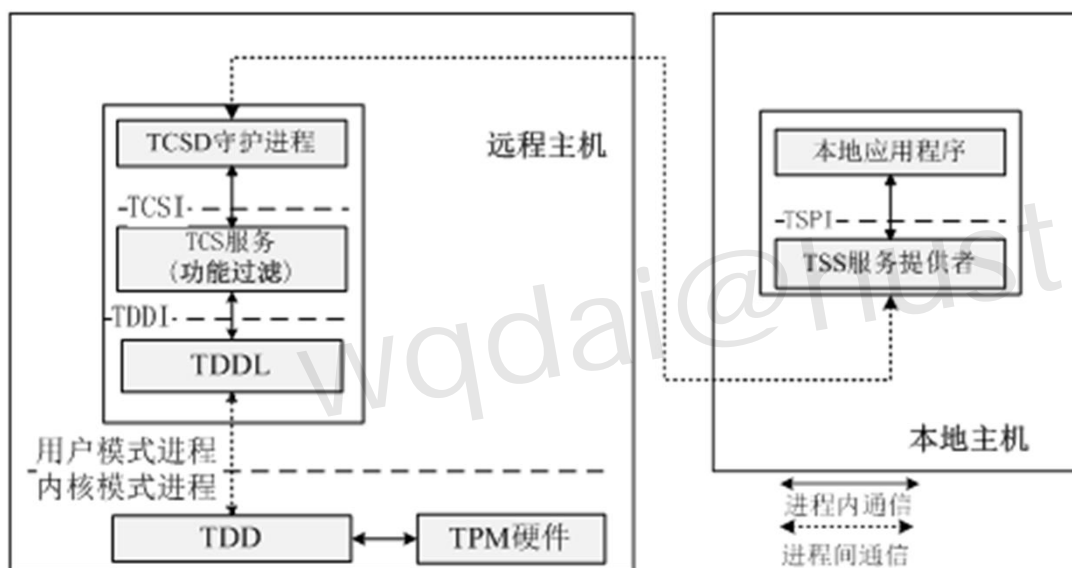
# VM的密钥共享



TCS部署在本地模式

(1) 若采用TCS部署在本地模式，意味着每个虚拟机都部署TCS，这样用户通过TPMc产生的密钥就保存在DomU中。由于我们使用了nfs，所以在这种部署方式下，将DomU中TCS维护的密钥都放到nfs中实现共享。但是这就意味着DomU对nfs有修改的权限，这显然是不安全的（DomU不可信）。所以这种部署方式不合适。

# VM的密钥共享



TCS部署在远程主机模式

(2) 采用TCS部署在远端的方式如何呢？由于整个VMG中，可信的只有 Dom0 和 nfs，所以 TCS 只能部署在 Dom0 或者 nfs 中。考虑到 TCS 是 TCG 软件栈的一部分，需要和 TPMc 通信，所以将 TCS 部署在 Dom0 更加方便和安全。然后将 Dom0 中 TCS 维护的密钥放到 nfs 上来实现共享。

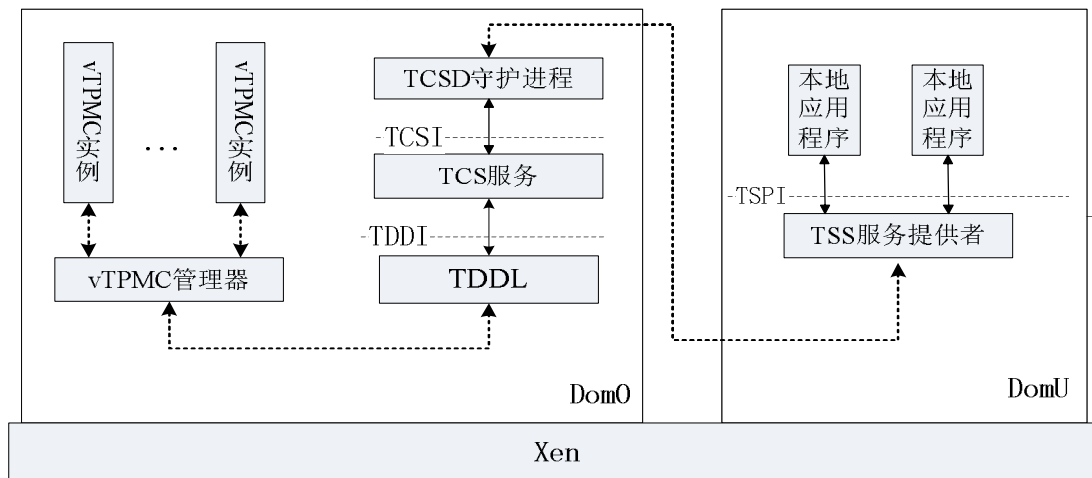
# VM的密钥共享

1.采用TCS部署在远端的方式——TCS部署在Dom0中，需要对tcsd有一些修改。

i) 取消原来tcs对TPM命令的过滤

ii) 由于无法采用原来的前后端机制（因为tcs部署在dom0中），所有需要将TDDL的TPM指令直接发送给TPMc管理器而不是原来的前端驱动。修改前后的架构图对比如下图所示。

2.部署在Dom0中的TCS维护的密钥，需要跨节点共享，因此找到存储这些密钥的地方，然后转存到NFS上，利用NFS实现共享和同步。



Tcs部署在dom0中，绕过了前后端驱动

# TPMc体现整个VMG的安全

TPMc不仅要给VMG提供TPM功能，更要体现VMG的整体安全性。我们可以在原有24个pcr的基础上，新增加24个pcr（pcr24~pcr47）用来度量整个VMG的安全性。pcr24监控VMG的所有pcr0，当有任意一个pcr0被扩展，那么TPMc的pcr24就做相应的扩展。同样的，pcr25监控VMG的所有pcr1，等等。因此TPMc可以监控到VMG中任意一个VM的安全状态变化。

共享存储密钥seal到VMG整体状态以及服务启动顺序上，如果安全服务未启动，则PCR的值和seal的不匹配，敏感数据无法解封。



# 如何区分TPM<sub>Mc</sub>和vTPM

- 由于要使用TPM<sub>Mc</sub>，必须通过部署在远端的TCS，而要使用本地的vTPM，得通过部署在本地的TCS。这个在应用程序中就可以体现和区分。
- TPM<sub>Mc</sub>：TCS放到Dom0中，绕过了Xen的前后端驱动。
- vTPM：本地TCS，通过前后端驱动。