# 目录

·导言

·所需软件

·从tpm2工具项目使用的工具和实用程序

·认证目标

·隐私注意事项

·为什么匿名或隐私很重要？

·解决方案

·TPM认证

·什么是PCR？PCR值是如何产生的

·PCR的初始状态

·将值扩展到PCR指数

·金黄色或参考PCR

·系统软件状态

·报告信托基金（RTR）

·裸骨远程认证模型中的角色识别

·设备服务注册

·服务请求第1部分（平台匿名身份验证）

·服务请求第2部分（平台软件状态验证）

·服务交付

·使用tpm2工具进行简单认证

·实现简单认证框架的脚本

·设备节点

·服务提供商

·隐私CA

·常见问题

# 介绍

本文展示了如何使用tpm2工具项目中的实用程序/工具来建立tpm2平台配置寄存器（PCR）所反映的系统软件状态的基本远程证明。其目的是提供关于该主题的一般指导，而不是特别讨论或参考任何认证框架。

注意：示例认证框架的所有代码示例都严格用于演示目的。尚未对其进行生产使用评估。

## 所需软件

tpm2 tss v2.3.0版

tpm2 abrmd v2.2.0版

tpm2工具v4.0

ibmswtpm公司

openssl

## 从tpm2工具项目使用的工具和实用程序

tpm2\u主要

tpm2\U创建

创建tpm2

tpm2\u创建

tpm2\U证书

tpm2\U生成凭据

tpm2\U激活基准

胎压监测系统

胎压监测系统

tpm2报价

tpm2\U复选框

tpm2\u随机

tpm2\u读取公共

# 认证目标

“证明是某物的证据或证明。它是某事物存在或是事实的声明。它是一种见证或正式证明某事的行为。”

文学定义在本教程中非常适用，本教程讨论了TPM中PCR内容的证明。

证明有两个关键部分，即：

证明者身份：要相信某件事是真的，就需要证明被证明的东西的真实性。这本质上意味着证明者的身份是已知的或可信的。

证明数据完整性：为了相信某些事情是真实的，生成信息的过程和保护信息不被篡改的过程都需要内在的信任。

## 隐私注意事项

虽然用于签署认证blob的密钥的私有或敏感部分必须保持机密或机密，但是可以使用用于验证认证blob上的签名的公钥来识别唯一的签名者。这会破坏平台用户的匿名性或隐私。

### 为什么匿名或隐私很重要？

让我们使用一个具有嵌入式应用程序的智能锁设备的假设示例。它集成了来自不同提供商的多种微服务，提供各种服务，即：

双因素认证（2FA）。

面部识别。

设备日志。

如果设备在所有微服务中使用相同的密钥进行公钥认证和/或签名认证。服务提供商或第三方分析可能会通过跟踪同一公钥的使用情况，在知情或不知情的情况下进行隐私侵犯。因此可以确定设备所有者使用面部识别2FA并在一天中的某些时间一致地操作设备，从而更新设备日志。

### 解决方案

至少有四种可能的方法来解决这个问题。

每个用户都使用相同的私钥/公钥对认证blob进行签名。密码学上，这种方法与将私钥复制到多个平台的过程和控制一样强大。此外，强度还取决于在平台中导入私钥后存储私钥的方法和机制的健壮性。撤销密钥意味着所有拥有该密钥的平台都必须被撤销，并用新密钥重新配置。请参阅演示如何将外部密钥导入TPM的tpm2\U导入。

每个用户都有一个唯一的私钥，但是，所有私钥都映射到一个公钥。这可以通过Intel®EPD技术实现。EPID还提供了一种机制来撤销组中的单个密钥或所有平台。TPM中还提供了一个类似的基于椭圆曲线密码体制的直接匿名认证（DAA），称为ECDAA；本教程中不讨论它。

每次必须对认证blob进行签名时，每个用户都会生成并使用一个新密钥（临时密钥）。这方面有一些独特的挑战。如果每个认证blob都用一个全新的密钥签名，那么如何在最小程度上推断匿名身份来确定平台的真实性，从而确定认证。因此，我们需要一个匿名身份，该身份通过加密方式绑定到一个唯一的可信身份，并且该唯一身份永远不会透露给除Privacy-CA之外的任何实体。

#3的变体。验证证明的每一方一份AIK。允许抽象客户端ID，但验证器知道在后续验证中它是同一个客户端。这在银行等应用中很有用。

注意：如果平台的主机名和/或ip地址在每个连接上保持不变，那么平台匿名性也会被破坏。这不是这里讨论的主题，但是这里假设平台用户使用VPN或其他合理机制采取了足够的措施来解决这一问题。

# TPM认证

根据系统的健壮性和隐私规则，平台匿名性可能不是强制性的，因此隐私考虑不适用。如果是这种情况，或者如果上述方法1或2足以满足您的认证和隐私需求，那么后面详细介绍TCG认可密钥（EK）和认证标识密钥（AIK）信息的部分就不重要了。

注：关于TPM认证术语的更详细的讨论可以在这里找到。本文档对该讨论进行了扩展，以演示如何使用tpm2工具建立一个最小的或基本的认证框架。

记住这些定义，让我们直接进入TPM认证主题。在TPM中的各种对象类型中，下面的讨论仅限于如何使用方法3安全且匿名地证明TPM PCR数据。

## 什么是PCR？PCR值是如何产生的

PCR或平台配置寄存器是特殊的TPM2对象，只能通过哈希扩展机制进行修改或写入。其中，传入的新值与PCR中的现有值串联并散列。新的散列值现在将替换旧的值。这意味着尽管它是一个位置，但最终值反映了所有哈希扩展的历史。PCR按组排列，每个组一个用于特定的哈希算法SHA1、SHA256等。每个组最多有32个PCR。可信计算组发布了一个指南，说明系统软件的哪一部分应该扩展到PC客户机规范中的特定PCR索引。索引16处还有一个调试PCR，可以通过发出tpm2\u pcrreset命令将其重置为0。TPM重置时其他PCR重置。

修改PCR的唯一方法是扩展散列，而防止物理篡改值的健壮性机制使TPM成为存储信任根（RTS）。

### PCR的初始状态

PCR索引中的初始值由平台特定规范确定。可以使用tpm2\u pcrread工具并指定PCR选择字符串来读取给定PCR索引处的值。与我们的示例相关，让我们读取SHA1和SHA256库中的PCR索引0、1、2。

tpm2\u pcrread sha1:0,1,2+sha256:0,1,2

如果在模拟器上运行，PCR处于初始状态，因为PCR还没有扩展。因此，默认值（00/FF）应显示如下。

沙10:0000000000000000000000000000

1:0000000000000000000000000000

2:0000000000000000000000000000000000000000

SHA256 0:0000000000000000000000000000000000000000

1:0000000000000000000000000000000000000000000000

2:0000000000000000000000000000000000000000000000

### 将值扩展到PCR指数

PCR指标只能通过扩展来修改。也就是说，它不能像寄存器那样设置为绝对值。三种工具之一可以修改PCR值，即：

tpm2\u pcrextend：向TPM提供预先计算的数据摘要。

tpm2\u pcrevent：数据直接呈现给TPM，并且TPM在扩展之前计算数据摘要。

tpm2\u pcrreset:PCR索引值重置为零。并非所有PCR指标都是可重置的。

在tpm2\u pcrextend和tpm2\u pcrevent的情况下，TPM最终将传入的数据摘要与PCR索引处的当前值连接起来，散列连接数据并替换PCR索引值。

应该注意的是，扩展PCR的实体本质上是可信的。在TCG术语中，这个受信任的实体称为测量信任根（RTM）。Intel（R）Bootguard就是一个例子。防篡改代码扩展了预引导软件关键部分的初始测量。

下面与本教程中的示例脚本相关的是一个如何使用tpm2\u pcrextend修改PCR索引的示例。

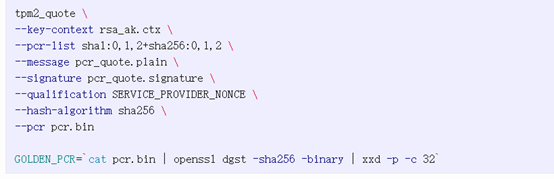


### 金黄色或参考PCR

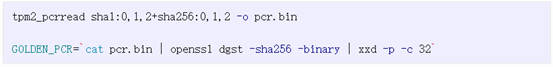
术语“黄金/参考”不是一个正式的术语。它只是PCR选择中PCR指标内容的摘要。选择是对所有PCR库的PCR指标的选择。PCR索引中的值本身是表示已知良好状态的二进制软件/数据块的摘要。单个PCR指标的摘要算法与特定库的摘要算法相同。然而，必须指定黄金/参考状态的摘要。

黄金/参考PCR状态可通过以下三种方式之一计算：

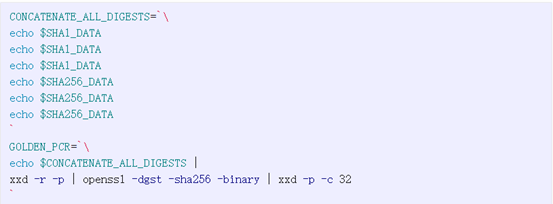
使用tpm2报价工具。用于对报价进行签名的签名方案确定报价的摘要算法。这应该在参考平台上进行一次。



如果只从参考平台读取PCR数据，也可以使用tpm2\U pcrread工具。此外，tpm2\u pcrread工具的输出可以直接传递给tpm2\u checkquote工具。但是，进行此操作时，必须将PCR选择信息指定给tpm2\ U checkquote工具。使用此方法计算黄金/参考PCR数据如下所示。



完全没有TPM，如下所示。



使用上述3种方法中的任何一种，黄金PCR值为“E756E3AF77A4F15A3F2ED489A7411A93D91D619506B6D1ED1121FAAEF45D8DE”。

## 系统软件状态

简而言之，这是一个单一的摘要，也就是对系统软件的有趣/关键部分的度量。为此，软件的受信任部分执行以下操作：

计算将加载以执行的软件模块的哈希值。

将摘要发送到TPM，并请求在PCR中对其进行扩展。

注：扩展度量的系统软件的受信任部分称为度量信任根（RTM）。这是天生的信任。

在整个平台引导过程中，将创建所有可执行代码和相关配置数据的日志，并将其扩展到pcr中。每次扩展PCR时，都会在TCG事件日志中创建一个日志条目。这使得挑战者能够看到最终的PCR消化是如何构建的。可以使用tpm2\u eventlog工具检查事件日志。事件日志通常存储在以下位置/sys/kernel/security/tpm0/binary\bios\u。

## 报告信托基金（RTR）

TPM密钥对象的私有或敏感部分受TPM保护。结合TPM设计提供的身份验证和增强授权模型，从这些密钥生成的签名被认为是可信的。如前所述，身份是签名和数据可信度的另一个方面。我们还确定，为了保护最终用户的隐私，需要匿名身份和加密绑定的唯一身份。这些需求共同构成了定义TPM对象的标准，这些属性使TPM成为报告信任（RTR）的根。

在TPM划分为的四个层次结构中，认可层次结构将由隐私管理员使用。在背书层次结构下创建的主键提供唯一标识，称为背书密钥（EK）。以下属性使EK在可在任何层次结构下创建的其他主键中具有特殊性：

TCG EK不能用作签名密钥。

TCG EK身份验证是一种引用认可层次结构授权的策略。

TCG EK由TPM制造商认证。有关密钥的更多详细信息，请参阅TCG EK凭证规范。如前所述，EK是一个主键，因此可以使用tpm2\u createprimary工具通过提供正确的属性和授权策略来创建。为了简化认证，还提供了一个工具tpm2\u createek，该工具具有TCG EK credential spec指定的所有默认值。

匿名身份是以EK作为其父级创建的认证身份密钥（AIK）。TCG没有强制要求的特定密钥模板来确定AIK密钥属性或授权模型。由于密钥通常用于引用/签名/认证等隐私敏感操作，因此密钥是在具有隐私管理员控件的背书层次结构下创建的签名密钥。因此，它的授权模型通常通过策略引用认可层次结构的授权。AIK与EK的关联是使用一种称为凭证激活的加密方法来完成的。与EK不同，EK是一个可以重新创建的常量/唯一主键，AIK键是短暂的。在这种情况下，每次创建AIK时，都会产生一个全新的密钥，从而使密钥匿名。上面提到的AIK是EK主键的子级，可以使用tpm2\u create工具创建。或者，也可以使用tpm2\u createak创建密钥，以便简单地证明。该工具使用最常用的AIK属性实现TPM2\u CC\u Create命令。

总之，

TPM制造商EK证书为唯一身份提供凭证；而凭证激活过程为AIK与EK的关联提供凭证，从而为TPM提供凭证。

tpm2\u createek和tpm2\u createak工具可用于创建EK和AIK。可以通过分别使用tpm2\u createprimary和tpm2\u create创建对象来对所选默认值之外的键进行任何进一步的自定义。

注意：系统软件状态也可以被视为系统的可信标识，而不需要签名密钥。这样的身份只对系统的自我证明有用。这也被称为本地认证。本地认证有一些用例，如密封加密机密或使用PCR状态作为代理认证模型，只要系统标识（即系统软件状态）不变，就可以使TPM对象的授权有效。

# 裸骨远程认证模型中的角色识别

最小裸骨远程认证的主要目标是：

证明证明引用的验证。

报价应包含表示系统软件状态的PCR值的摘要。

引用的签名者应该对验证者匿名；但是验证者应该能够确定引用是由有效的签名者签名的。

为了实现这些目标，有3个参与者就足够了，即：

设备节点：具有TPM的边缘平台，其系统软件状态是相关的。它通过引用中包含的PCR数据摘要生成认证结构。平台使用匿名的认证标识密钥对报价进行签名。AIK以加密方式绑定到同一平台上的唯一标识密钥。唯一标识密钥是认可密钥（EK）。

隐私CA：唯一可以证明AIK与有效EK关联而不向“服务提供商”披露EK的可信实体。这也是除了“设备节点”之外唯一一个从“设备节点”知道给定AIK的EK的实体。

服务提供者：“设备节点”与之通信以利用服务的实体。“服务提供商”需要确保：a.请求服务的实体已向“隐私CA”注册了其唯一身份。b、 匿名身份属于“隐私CA”存储的已注册唯一身份池。c、 “设备节点”的系统软件状态是可接受的。

然而，在实践中，这里显示的各种角色可以进一步细分。例如，匿名身份和系统软件状态的所有验证都可以交给另一个“验证者”角色。

现在让我们来看一下示例证明框架的各个阶段。

服务注册。

服务请求第1部分（匿名身份验证）。

服务请求第2部分（平台软件状态验证）。

提供服务。

## 设备服务注册

这是“设备节点”向“服务提供者”请求服务的阶段。“服务提供者”在这个阶段有两个要求。

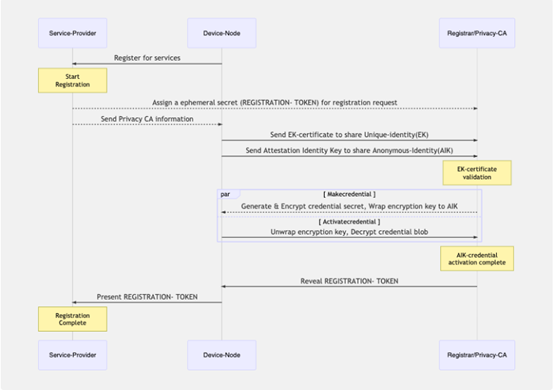
“设备节点”将其唯一密钥发送到“隐私CA”

“隐私CA”通过EK证书验证验证EK的真实性，并通过凭证激活过程验证EK在平台上的存在。

为了证实上面的#1&#2发生了，“服务提供商”创建了一个名为REGISTRATION-TOKEN的临时秘密，它只与隐私CA共享，而隐私CA反过来又向“设备节点”透露该秘密，当且仅当EK证书和凭证激活上的所有验证都通过时（仅当“设备节点”与平台EK和AIK通信时才可能）Privacy-CA。然后“设备节点”将REGISTRATION-TOKEN呈现给“服务提供商”以完成注册。

注：“服务提供商”在注册结束时没有关于平台的进一步信息记录。它只知道某个平台发出了注册请求，并且它的EK已在Privacy-CA的有效EK池中注册。

下面的序列图显示了注册过程中的交互。



## 服务请求第1部分（平台匿名身份验证）

这是注册平台向“服务提供商”发出服务请求并发送临时AIK以安全地获取服务的阶段。“服务提供商”需要了解以下内容：

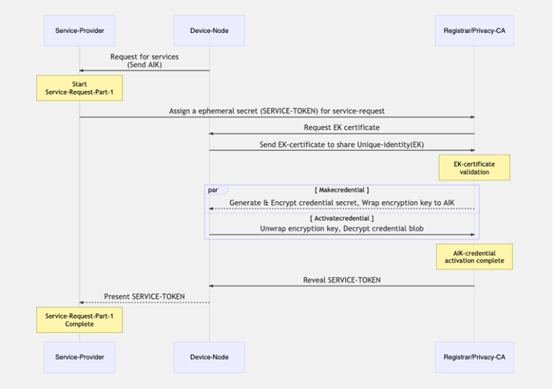
AIK是有效的，并且隐私CA可以证明AIK绑定到隐私CA的信任EK池中的EK。

AIK来自平台当前可访问的TPM。

为了实现这一点，隐私CA需要从平台请求EK，并通过凭证激活过程确保AIK存在于平台上。

为了验证在“隐私CA”和“设备节点”之间发生的这些交互，“服务提供者”创建了一个称为Service-TOKEN的临时秘密，它与“隐私CA”共享，后者在成功的凭证激活过程之后将其透露给“设备节点”。“设备节点”现在通过向“服务提供者”展示服务令牌来证明服务请求的有效性。

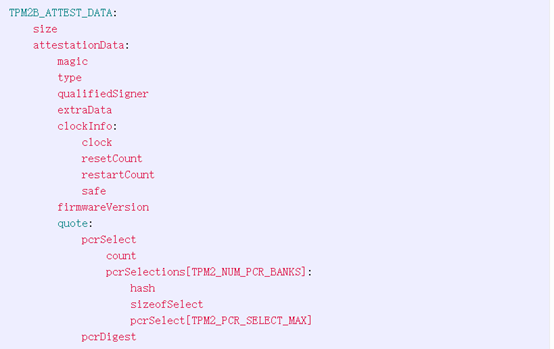
下面的序列图显示了服务请求期间的交互。



## 服务请求第2部分（平台软件状态验证）

在这个阶段，通过验证设备呈现的“服务令牌”，“服务提供商”已经确定AIK来自注册的平台并且它是可信的签名密钥。现在“服务提供商”需要确定“设备节点”的系统软件状态。为了实现这一点，“服务提供者”请求来自“设备节点”的认证报价，该报价必须与AIK签名。为了防止重放攻击，“服务提供商”生成一个NONCE，需要在签名之前将其添加到认证报价中。

下面是由tpm2报价返回的包含PCR的认证数据的YAML表示。



以上yaml陈述中显示的是认证报价中包含的所有信息。重要字段的简要说明如下：

魔力：表示此结构是由TPM始终TPM2生成的\U值创建的。

类型：证明结构的类型。这是TPM2\u ST\u detect\u对具有PCR信息的类型的引用，这里将详细讨论。

qualifiedSigner：签名密钥的限定名称。术语限定名是所有祖先键的所有名称的摘要，返回到层次结构根目录下的主种子。

外部数据：调用者提供的外部信息。由“服务提供者”生成的NONCE添加到此字段中。

时钟：TPM通电的时间（毫秒）。当存储主种子更改为TPM2\u Clear时，此值将重置为零。

resetCount：自上次清除TPM 2\U以来TPM重置的次数。

restarcount：自上次TPM重置或清除TPM 2\u后，发生TPM 2\u关闭或TPM\u哈希\u启动的次数。

安全：表示TPM以前未报告任何大于当前时钟值的时钟值。在清除TPM2时设置为“是”。

firmwareVersion:TPM供应商特定的值，用于标识固件的版本号。

pcrSelect：有关algID、PCR选择和摘要的信息。

计数：选择结构的数目。允许值为零。这表示所选PCR库（SHA1、SHA256等）的计数

PCR选择：这是PCR选择结构的列表。

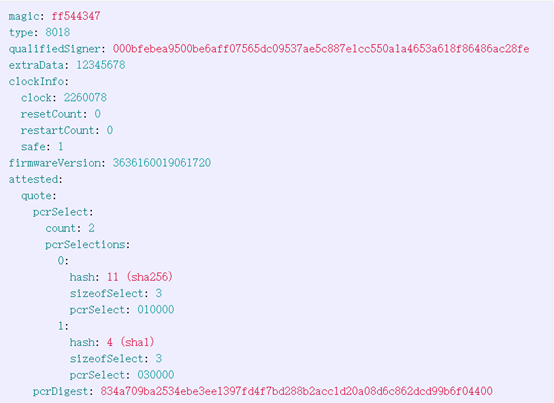
哈希：与选择库关联的哈希算法。

sizeofSelect:pcrSelect数组的大小（以八位字节为单位）。这表示表示一个组中所有PCR所需的字节数。每个PCR都表示为一个位。E、 g.对于每个银行24个PCR，sizeofselect应该是3个字节。

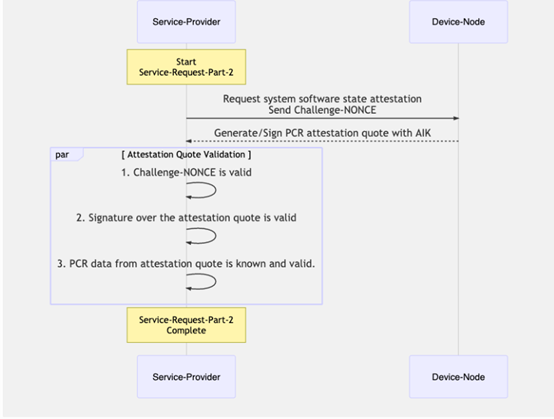
pcrSelect：所选PCR的位图（首先是最低有效字节）

pcrDigest：使用签名密钥的哈希算法对所选PCR的摘要。

下面是来自SHA1银行的PCR0认证报价和SHA256银行的PCR0、PCR1认证报价的转储示例。

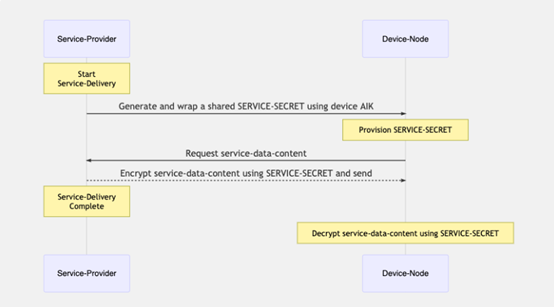


下面的序列图显示了这一阶段的相互作用。



## 服务交付

在这个阶段，通过验证“服务令牌”和它从“设备节点”接收到的认证引用，“服务提供者”已经确定了“设备节点”的匿名身份和系统软件状态。当系统处于已知状态时，“服务提供者”现在可以用AIK公钥包装“服务秘密”。

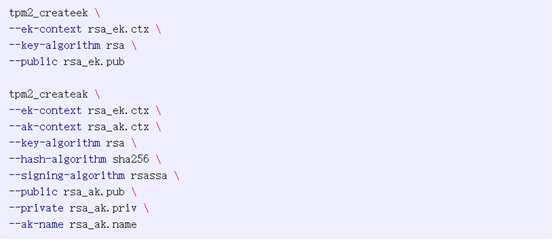


# 使用tpm2工具进行简单认证

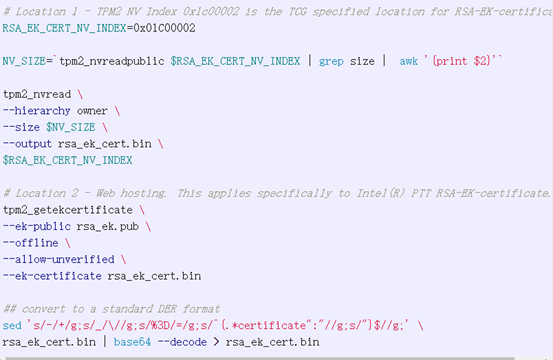
下面的演示显示了上述序列图中涉及的tpm2工具。在本节演示了tpm2工具之后，接下来的一节将展示不同角色的脚本及其在简单认证框架中的交互。它包括编写工具和逻辑脚本，以便在“服务提供商”、“设备节点”和“隐私CA”之间移动标识、引号、令牌、nonce等。

有了这篇序言，让我们直接进入tpm2工具。

“设备节点”创建认可密钥和认证标识密钥。



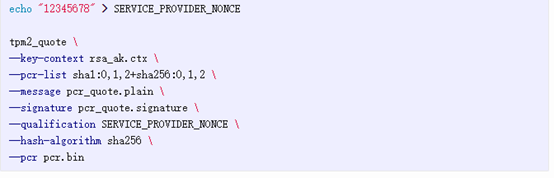
“设备节点”检索要发送到“隐私CA”的认可密钥证书。TPM制造商在两个可能的位置提供认可密钥证书。虽然大多数TPM制造商将它们存储在TCG指定的NV索引中，但也有一些通过web主机提供下载。让我们看看这两种方法。



“隐私CA”和“设备节点”执行凭证激活质询以验证AIK从最初由“设备节点”共享的EK证书绑定到EK。在所提出的简单证明框架中，这是在两个不同的实例中完成的——一次是“服务提供者”请求“设备节点”作为服务注册过程的一部分发送标识。第二次，当“设备节点”将其AIK发送给“服务提供商”时，“服务提供商”又将其发送给“隐私CA”以验证匿名身份。



“设备节点”根据“服务提供商”的请求生成PCR证明报价。“服务提供者”指定PCR库、PCR索引和临时临时临时数据。NONCE是为了确保报价验证和确认过程中不存在重放攻击的可能性。认证报价的签名密钥的有效性由“隐私CA”确定为有效密钥。



“服务提供商”验证由“设备节点”生成并签名的证明报价。为了确定“设备节点”的软件状态，在签名和nonce验证过程之后，“服务提供商”根据已知的有效值（先前确定的黄金/参考值）验证报价中PCR值的摘要。

