虚拟平台环境中一种新的可信证书链扩展方法

谭良[1,2]，齐能[1]，胡玲碧[1]

（1. 四川师范大学计算机科学学院，四川 成都 610101；2. 中国科学院计算技术研究所，北京 100190）

摘 要：利用可信计算技术构建可信虚拟平台环境时，如何合理地将底层物理的可信平台模块（TPM, trusted platform module）的证书信任扩展延伸到虚拟机环境是值得关注的问题。目前，已有的证书信任扩展方案均不完善，有的方案存在违背 TCG 规范的情况，有的方案增加密钥冗余和 Privacy CA 性能负担，有的方案甚至不能进行证书信任扩展。因此，提出了一种新的可信证书链扩展方法。首先，在 TPM 中新增一类证书——VMEK（virtual machine extension key），并构建对 VMEK 的管理机制，该证书的主要特点是其密钥不可迁移，且可对 TPM 内和 TPM 外的数据进行签名和加密。其次，利用证书 VMEK 对 vTPM 的 vEK 签名构建底层 TPM 和虚拟机 vTPM 的证书信任关系，实现可信证书链在虚拟机中的延伸。最后，在 Xen 中实现了 VMEK 证书及其管理机制和基于 VMEK 的证书信任扩展。实验结果表明，所提方案可以有效地实现虚拟平台的远程证明功能。

关键词：可信计算；虚拟平台；可信平台模块；vTPM；证书链扩展

中图分类号：TP309

文献标识码：A

doi: 10.11959/j.issn.1000-436x.2018090

New extension method of trusted certificate chain in virtual platform environment

TAN Liang[1,2],QI Neng[1], HU Lingbi[1]

1. College of Computer Science, Sichuan Normal University, Chengdu 610101, China

2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Abstract: When using trusted computing technology to build a trusted virtual platform environment, it is a hot problem that how to reasonably extend the underlying physical TPM certificate chain to the virtual machine environment. At present, the certificate trust expansion schemes are not perfect, either there is a violation of the TCG specifications, or TPM and vTPM certificate results inconsistent, either the presence of key redundancy, or privacy CA performance burden, some project cannot even extend the certificate trust. Based on this, a new extension method of trusted certificate chain was proposed. Firstly, a new class of certificate called VMEK (virtual machine extension key) was added in TPM, and the management mechanism of certificate VMEK was constructed, the main feature of which was that its key was not transferable and could be used to sign and encrypt the data inside and outside of TPM. Secondly, it used certificate VMEK to sign vTPM's vEK to build the trust relationship between the underlying TPM and virtual machine, and realized extension of trusted certificate chain in virtual machine. Finally, in Xen, VMEK certificate and its management mechanism, and certificate trust extension based on VMEK were realized. The experiment results show that the proposed scheme can effectively realize the remote attestation function of virtual platform.

Key words: trusted computing, virtual platform, trusted platform module, vTPM, certificate chain extension

收稿日期：2017-03-29；**修回日期**：2018-03-14

基金项目：国家自然科学基金资助项目（No.61373162）；四川省科技基金资助项目（No.2014GZ0007）；可视化计算与虚拟现实四川省重点实验室基金资助项目（No.KJ201402）

Foundation Items: The National Natural Science Foundation of China (No.61373162), Sichuan Science and Technology Project (No.2014GZ0007), Sichuan Key Laboratory of Visual Computing and Virtual Reality Project (No.KJ201402)

1 引言

IT 资源服务化是云计算中最重要的外部特征[1-4]，云端虚拟平台包括物理硬件层、虚拟化管理层和客户虚拟机层，其中，客户虚拟机是资源服务化的直接依托环境。因此，云端虚拟平台和客户虚拟机的可信是云安全中最为核心的问题，是解决云租户与云服务提供商之间相互信任的基础[5-9]。文献[5-9]均认为，解决云端虚拟平台和客户虚拟机的可信问题是推动云计算在更广的范围内扩展、沿拓的重大安全问题。而可信计算[10-11]是保障计算平台可信的基础手段，它通过提供数据保护、身份证明以及完整性测量、存储与报告等功能以提高计算平台整体的可信性。因此，将可信计算技术融入云端虚拟平台已成为云安全研究领域的一大热点[12-16]，其中，可信平台模块（TPM, trusted platform model）的虚拟化是可信计算和虚拟化结合的关键技术之一[17-24]。

目前，TPM 的虚拟化可以采用 3 种方式：软件仿真型 TPM 虚拟化、硬件共享型 TPM 虚拟化和聚合型 TPM 虚拟化。所谓软件仿真型 TPM 虚拟化，就是指虚拟平台为每一个需要为用户提供可信执行环境的虚拟机创建一个软件仿真型虚拟 TPM 实例。所谓硬件共享型 TPM 虚拟化，就是各虚拟机分时访问物理 TPM。所谓聚合型 TPM 虚拟化，即在一个 TPM 内聚合出多个 TPM 虚拟功能（VF, virtual function）——vTPM，然后将 vTPM 直接分配或调度给需要的客户虚拟机使用。TPM 无论采用何种虚拟化方式，均需向虚拟机提供一个具有 TPM 功能的逻辑实体——vTPM。在云环境中，通过 vTPM 为客户虚拟机提供可信保障，使每个客户虚拟机在逻辑上都能拥有单个“独有”的 TPM，就像拥有一个真实的物理 TPM 一样。客户虚拟机环境可以使用 vTPM 提供的度量、存储和报告等功能，特别是可通过 vTPM 的完整性校验功能实现客户虚拟机环境的信任链传递，也可通过 vTPM 的数据保护功能实现客户虚拟机数据的密封存储，还可通过 vTPM 的远程证明功能实现客户虚拟机环境的身份证明。

然而，在虚拟环境下，当客户虚拟机进行远程证明时，不仅需要向挑战者证明顶层虚拟机环境的可信，还必须向挑战者证明底层虚拟平台的可信，因此，必须建立从物理 TPM 到 vTPM 的证书链，构建物理平台、虚拟平台到客户虚拟机平台的绑定关系[17]。目前，在如何将底层物理 TPM 的证书链扩展延伸到虚拟机方面已有多种方案，包括 vTPM vEK to hTPM AIK Binding[17]、TPM AIK signs vTPM vAIK[17]、vEK Certificate Signing CA[17]、vTPM vEK to hTPM EK Binding[25] 、vTPM vAIK to hTPM SK Binding[26]和 hTPM EPS Product vEK[27]等，但以上这些方面均不完善，有的方案存在违背 TCG 规范的情况，有的方案增加密钥冗余，有的方案增加 Privacy CA 的性能负担，有的方案甚至不能进行证书信任扩展。

针对这一问题，本文提出了对应的解决方案，并通过实验验证其功能，具体实现思路及实验方案将在后续篇章做详细介绍。

2 相关工作

所谓 vTPM 的证书信任扩展，是指如何将物理 TPM 的证书信任关系扩展到 vTPM，构造 TPM 到 vTPM 的证书信任链关系。

目前，国内外对 vTPM 的证书信任扩展进行了大量研究。文献[17]在研究中提出了 4 种构建 vTPM 证书链的设计思路，但最后一种与特殊硬件有关，为不失一般性，本文不讨论第 4 种，只讨论前 3 种。

1) vTPM vEK to hTPM AIK Binding

如图 1 所示，此方法中 vEK 和 vAIK 均由 vTPM 产生，vEK 由 TPM 的 AIK 签名绑定，vAIK 由 Privacy CA 的私钥签名，验证方用 Privacy CA 的公钥进行验证。此方法不仅将底层的证书信任扩展到了虚拟机，而且 vTPM 的证书结构与 TPM 一致，便于理解和已有成果的移植，但缺点包括 2 个方面，一是用 TPM 的 AIK 对 vEK 签名，违背了 TCG 规范，AIK 只能对 TPM 内部产生的信息进行签名，而 vEK 是 TPM 的外部信息；二是 AIK 的有效期通常很短，AIK 的失效导致对 vEK 的签名失效，从而导致 vAIK 失效，需要频繁向 Privacy CA 重新申请 vAIK，因此，Privacy CA 的性能负担重。

2) hTPM AIK signs vTPM vAIK

如图 2 所示，此方法用 TPM 的 AIK 直接对 vTPM 的 vAIK 进行签名，不需要 Privacy CA。此方法不仅将底层的证书信任扩展到了虚拟机，而且降低了 Privacy CA 的负担。但该方法依赖于 TPM 的 AIK 对 vAIK 签名，不仅同样违反了 TCG 规范，而且 AIK 的失效同样会导致其对 vAIK 的签名失效，从而会频繁更新 AIK 对 vAIK 的签名，增加性能负担。

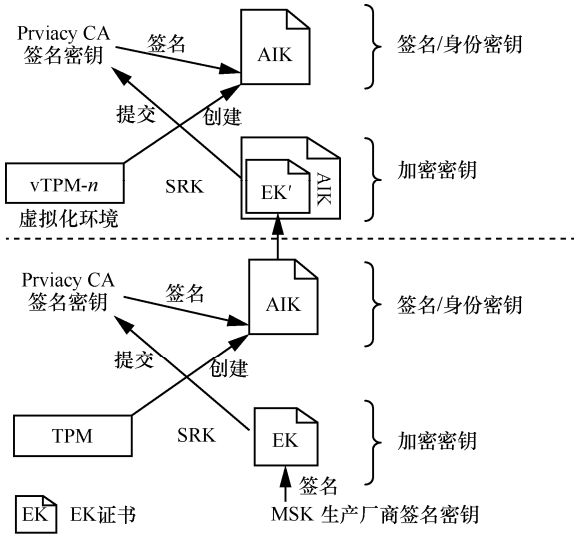


图 1　vTPM vEK to hTPM AIK Binding

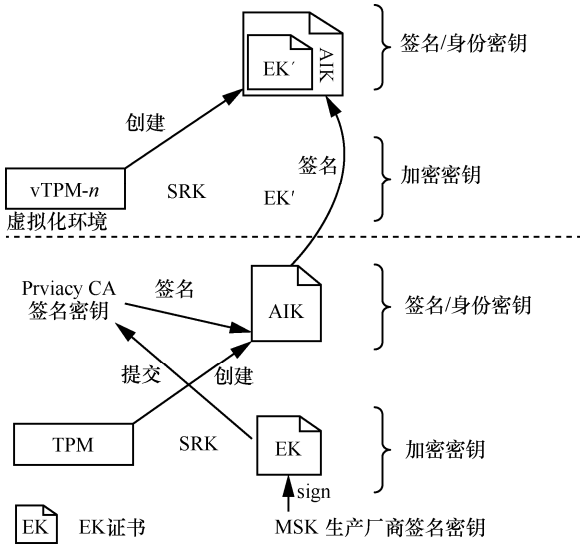


图 2　hTPM AIK signs vTPM vAIK

3) Local CA issue vEK Certificate

即由本地证书机关（注：不是 Privacy CA）为 vTPM 发布 vEK，如图 3 所示。本文方案的优点是 vEK 相对稳定，不会随着底层虚拟平台和 TPM 的变化而变化，缺点是不仅需要增加额外的证书机关，而且与 TPM 没发生绑定关系，即 TPM 的可信证书扩展没有传递到 vTPM。

文献[25]提出了 vTPM vEK to hTPM EK Binding 方案，即 vEK 由 TPM 的 EK 签名绑定，如图 4 所示。这个方案的优点是避免由于 AIK 的失效导致对 vEK 签名的失效，而且与 TPM 的绑定关系清楚、简单，将底层的 TPM 证书信任扩展到了虚拟机。但缺点是违反了 TCG 规范，即 EK 证书不能用于签名。

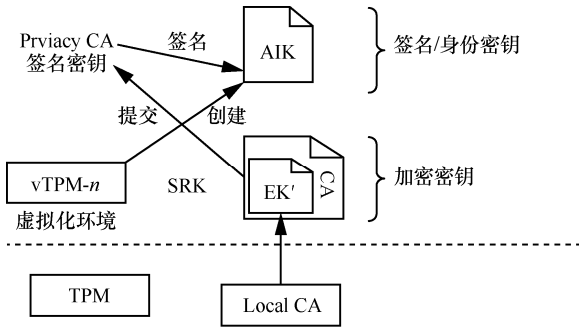


图 3　Local CA issue vEK Certificate

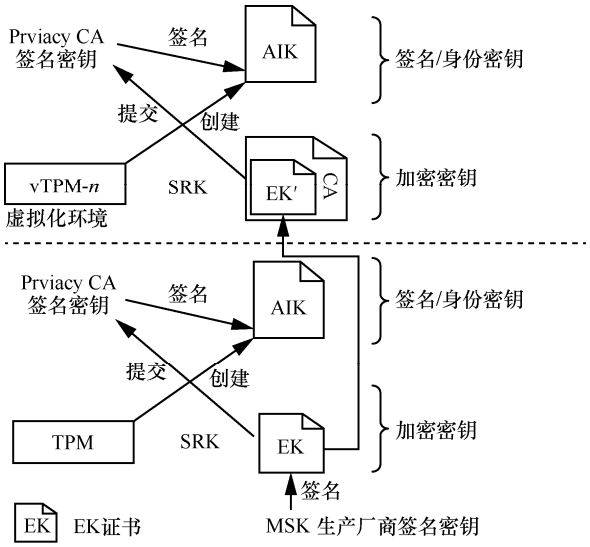


图 4　vTPM vEK to hTPM EK Binding

另外，为了对上述方案进行改进，文献[26]提出了 vTPM vAIK to hTPM SK Binding 方案，即引入了签名密钥 SK 作为中介，实现了 AIK 对 vAIK 的间接签名，使 AIK 不再对 TPM 外部数据进行签名，更好地满足 TCG 规范，而且 vAIK 的生成不依赖于 Privacy CA，降低了 Privacy CA 的负担，如图 5 所示。但该方案不仅增加了生成 vAIK 证书的复杂性，且没有解决 AIK 失效会导致 vAIK 失效的问题，而重构 vAIK 需要重新生成 SK，从而带来新的性能负担。另外，每一个 vAIK 证书对应一个 SK，会产生大量的密钥冗余。

文献[27]结合 TPM 2.0 的新特性，提出 hTPM EPS Product vEK 方案，如图 6 所示。在该方案中，vTPM 的身份证书 vAIK 由 EPS 推导产生的 vEK 向 Privacy CA 验证生成。文中认为，基于 vEK 和 EPS 的映射关系，就能够较为直接地标识虚拟机中的真实物理身份，建立从物理平台到虚拟平台的信任链，但实际上是不行的。EPS 仅仅是 Endorsement Key 的基础密钥种子，用 KDF 算法生成

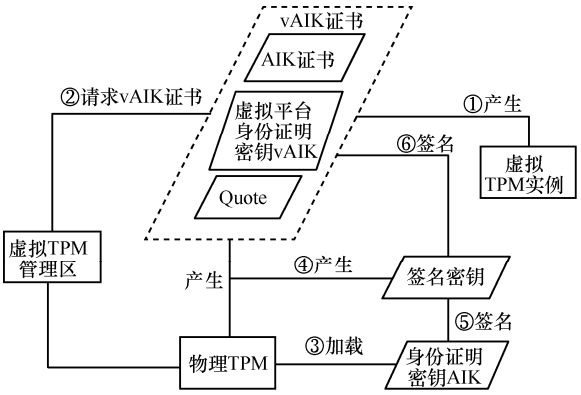


图 5　vTPM vAIK to hTPM SK Binding

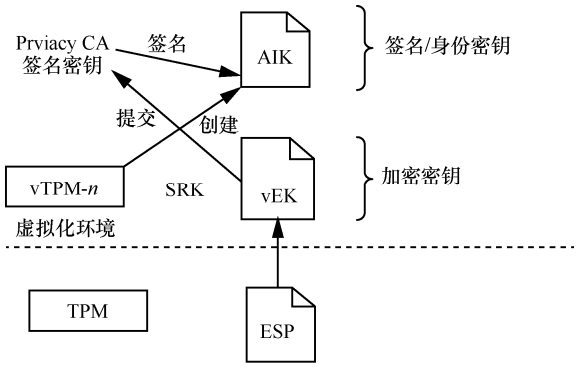


图 6　hTPM EPS Product vEK

Endorsement Key，虽然简单、容易，但仅仅是生成了 vEK 的密钥对而已，不存在信任扩展和传递的问题，外界不能通过这个密钥对的公钥推导出该公钥是底层 TPM 的 EPS 产生的。vTPM 证书信任链的扩展仍然需要将 EPS 产生的 vEK 的密钥公钥、底层虚拟化平台的签名信息和 vTPM 的相关信息传递给 Privacy CA 生成 vEK 证书，再由 vEK 证书向 Privacy CA 生成 vAIK 来实现。

从上面的分析可以看出，无论采用什么方式，将底层物理 TPM 的证书信任关系扩展到 vTPM 证书均不完善，有的方案存在违背 TCG 规范的情况，有的方案增加密钥冗余，有的方案增加 Privacy CA 的性能负担，有的方案甚至不能进行证书信任扩展。这将降低用户使用的方便性和信任度。

3　TPM 新证书——VMEK 的设计

为了解决以上问题，本文特为 TPM 增加一种可选的新证书——VMEK（virtual machine extension key），该证书由 TPM 拥有者通过 PCA 生成并激活，私钥由 SRK 保护，不可迁移，可对 TPM 内和 TPM 外的数据进行签名和加密，只用于虚拟机证书信任扩展和虚拟机迁移（或 vTPM 迁移）。由于本文只研究证书信任扩展，因此，在设计 VEMK 时不讨论迁移问题。

3.1　VMEK 证书结构与属性

TPM 中每种证书均要设定特定操作的数据段信息，包括背书证书、一致性证书、平台证书、确认证书和身份证书。VMEK 证书的数据段如表 1 所示，包含 VMEK 公钥、TPM 制造商、TPM 模块、TPM 一致性参考、平台类型、平台制造商、发布者、签名值、TPM 规范、PCR 值、源 VMEK 证书、有效期、策略参考和其他。

表 1　VMEK 证书的数据结构

数据段名称	说明	字段设置
VMEK 公钥	VMEK 公钥部分	必须
TPM 制造商	确定 TPM 制造商的标识符	必须
TPM 模块	确定 TPM 模块的信息	必须
TPM 一致性参考	指向 TPM 一致性证书的指针	必须
平台类型	确定平台的类型	必须
平台制造商	确定平台制造商的标志符	必须
发布者	确定证书的发布者	必须
签名值	发布者的签名值	必须
TPM 规范	确定 TPM 遵守的规范	必须
PCR 值	虚拟平台的 PCR 值	可选
源 VMEK 证书	源虚拟平台 VMEK 证书，便于回溯	可选
有效期	证书有效的时间段	可选
策略参考	证书的策略参考	可选
其他	扩展域（保留）	可选

其中，设置 PCR 值数据段的目的是通过该证书可以验证虚拟平台的可信性，通常包含 PCR[0]~PCR[15]，源 VMEK 证书数据段存放的是虚拟机迁移之前的虚拟平台 VMEK，设置该数据段的目的是通过该证书回溯原虚拟平台。

VEMK 密钥具有不可迁移性，在加密方面，具有与存储密钥相同的作用；在签名方面，具有与签名密钥相同的作用。具体比较如表 2 所示。

表 2　VMEK 证书的密钥属性与签名密钥和存储密钥的比较

名称	允许的签名方案	允许的加密方案
签名密钥	TPM_ES_NONE	TPM_SS_RSASSAPKCS1 v15_SHA1
存储密钥	TPM_ES_RSAESOAEP_ SHA1_MGF1	TPM_SS_NONE
VEMK 密钥	TPM_ES_RSAESOAEP_ SHA1_MGF1	TPM_SS_RSASSAPKCS1 v15_SHA1

3.2　VMEK 证书与 TPM 其他证书之间的关系

表 3 是 VMEK 与背书证书、一致性证书、平台证书、确认证书、AIK 证书和 VMEK 证书这 6 种证书的比较。

VMEK 证书与 EK 证书、平台证书和一致性证书之间还存在着一定的相互联系和相互制约关系，具体如图 7 所示。

VMEK 证书包括 EK 证书里的信息，如 TPM 制造商、TPM 模块等；VMEK 证书也包括平台证书里的信息，如平台类型、平台制造商等；VMEK 证书还包括一致性证书。VMEK 证书并不涉及 EK 公钥部分、平台证书公钥部分、一致性证书公钥部分等的私有敏感信息。

3.3　VMEK 证书管理

VMEK 证书管理包括 2 个方面的内容，一方面是 VMEK 证书的生成和使用，另一方面是 VMEK 证书的存储。

对于 VMEK 证书的生成和使用，本文定义 4 个接口，分别是 TPM_CreateVMEKKeyPair、TPM_ActiveVMEK、TPM_VMEKLoad 和 TPM_VMEK_Signing。特别需要说明的是，为了节省篇幅和便于理解阅读，接口中出现的类型、符号

表 3　　**TPM 中证书功能比较**

证书类型	发布者	证书作用	证书内容
背书证书	TPM 制造商	证明 TPM 身份	TPM 模块、发布者、TPM 规范、签名值、公钥等
一致性证书	可信第三方	指出评估者认可 TPM 的设计和实现符合评估准则	评估者名、平台制造商名、平台型号、平台版本号、背书证书
平台证书	平台制造商	确认平台的制造者并且描述平台的属性	背书证书、平台模型、发布者、平台规范、签名值等
确认证书	可信第三方	确认系统中某个硬件或软件	确认实体名、组件生产商名、组件型号、发布者、签名值等
AIK 证书	Privacy CA	证明 TPM 及平台的身份	AIK 公钥、TPM 模块、发布者、TPM 规范、签名值、身份标签（背书证书、验证证书和平台证书）等
VMEK 证书	Privacy CA	用来迁移 vTPM 及证书信任扩展	VMEK 公钥、TPM 模块、平台类型、一致性证书、发布者、签名值、源 PCR 值、源 VMEK 证书等

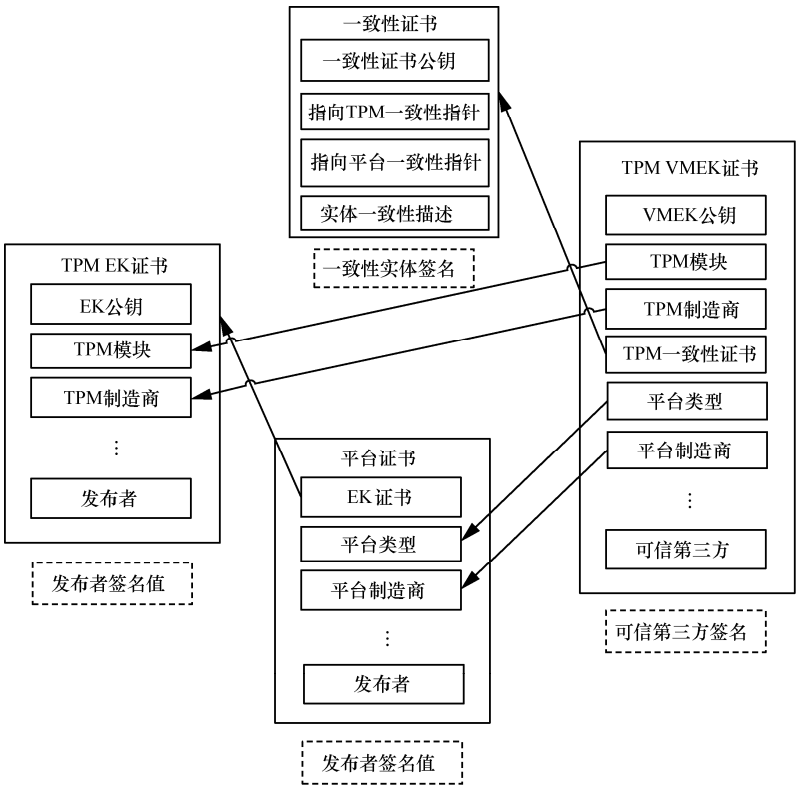


图 7　VMEK 证书与其他证书之间的关系

常量等与 TPM 规范一致。叙述中只定义与 VMEK 相关的接口和关键数据结构。

定义 1 TPM_Result TPM_CreateVMEKKeyPair (

TPM_ENCAUTH *VMEKAuth, //输入参数，VMEK 的加密授权数据

TPM_CHOSENID_HASH *labelPrivCADigest, //输入参数，PCA 身份标签摘要

TPM_KEY *VMEKKeyParams, //输入参数，VMEK 密钥的所有相关参数

TPM_AUTH *auth1, //输入输出参数，授权协议参数 1

TPM_AUTH *auth2, //输入输出参数，授权协议参数 2

TPM_KEY *VMEKKey, //输出参数，创建的 VMEK 密钥对

UINT32 *VMEKBindingSize, //输出参数，用 VMEK 签名的内容长度

BYTE **VMEKBinding //输出参数， VMEK 对 TPM_VMEK_CONTENTS 的签名值

);

该接口用于创建 VMEK 公私钥对（长度至少是 2 048 bit）。如果执行成功，返回 TPM_SUCCESS；否则，返回 TPM 错误代码。具体流程如图 8 所示。

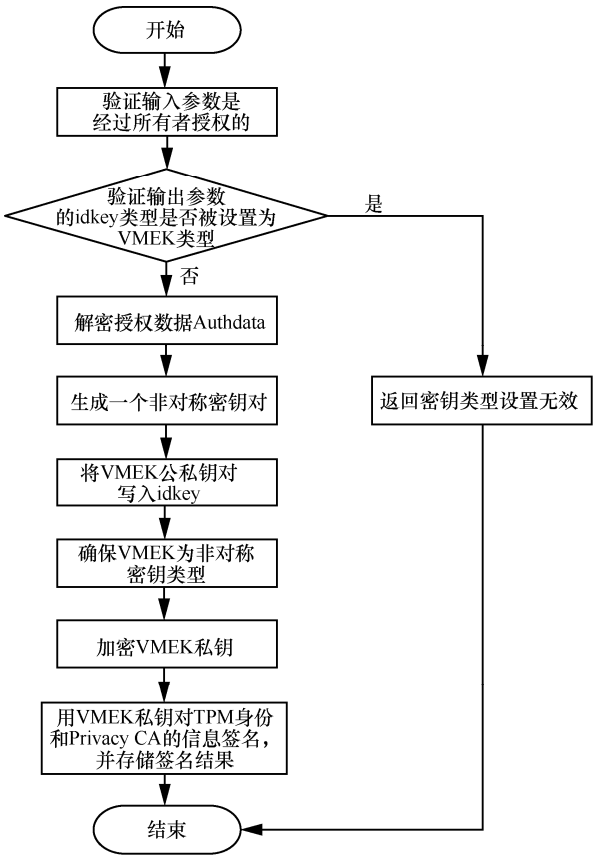


图 8 TPM_CreateVMEKKeyPair 接口实现流程

首先，TPM owner 使用 TPM_Create VMEKKeyPair 命令生成一对 VMEK 公私钥对（长度至少是 2 048 bit），同时产生一个 TPM_VMEK_ CONTENTS 结构，该结构中包括刚生成的 VMEK 的公钥部分以及 TPM 的一些标识信息。然后，使用 VMEK 的私钥部分对刚产生的 TTPM_VMEK_ CONTENTS 进行签名。最后，将签名值、TPM_VMEK_CONTENTS、背书证书、平台证书和一致性证书等一起发送给一个 Privacy CA 并等待其接受请求后生成 VMEK 证书，而 VMEK 的私钥部分则通过 SRK 加密保存在 TPM 内部。

其中，TPM_VMEK_CONTENTS 定义如下。

```
struct TPM_VMEK _CONTENTS {
    TPM_STRUCT_VER ver;//版本
    UINT32 ordinal; //序号
    TPM_CHOSENID_HASH labelPrivCADigest;
// PCA 身份标签摘要
    TPM_PUBKEY VMEKPubKey; //VMEK 公钥
};
```

随后，Privacy CA 产生一个会话密钥，并使用这个密钥对刚生成的 VMEK 证书进行加密，然后使用用户 TPM 的 EK 公钥对该会话加密，产生一个 TPM_ASYM_CA_ ATTESTATION 结构，其中，包括被加密的会话密钥、被加密的证书以及一些加密算法参数等。最后，Privacy CA 将 TPM_ASYM_CA_ ATTESTATION 发送给 TPM。

定义 2 TPM_Result TPM_ActiveVMEK (

TPM_KEY_HANDLE VMEKKeyHandle, //输入参数，VMEK 密钥句柄

UINT32 blobSize, //输入参数，来自 PCA 的 TPM_SYM_CA_ATTESTATION 的长度

BYTE *blob, //输入参数，来自 PCA 的 TPM_SYM_CA_ATTESTATION

TPM_AUTH *auth1, //输入输出参数，授权协议参数 1

TPM_AUTH *auth2, //输入输出参数，授权协议参数 2

TPM_SYMMETRIC_KEY *symmetricKey //输出参数，和 PCA 交互的会话密钥

);

该接口的主要作用是获取 VMEK 证书，并激活。如果执行成功，返回 TPM_SUCCESS；否则，返回 TPM 错误代码。具体流程如图 9 所示。

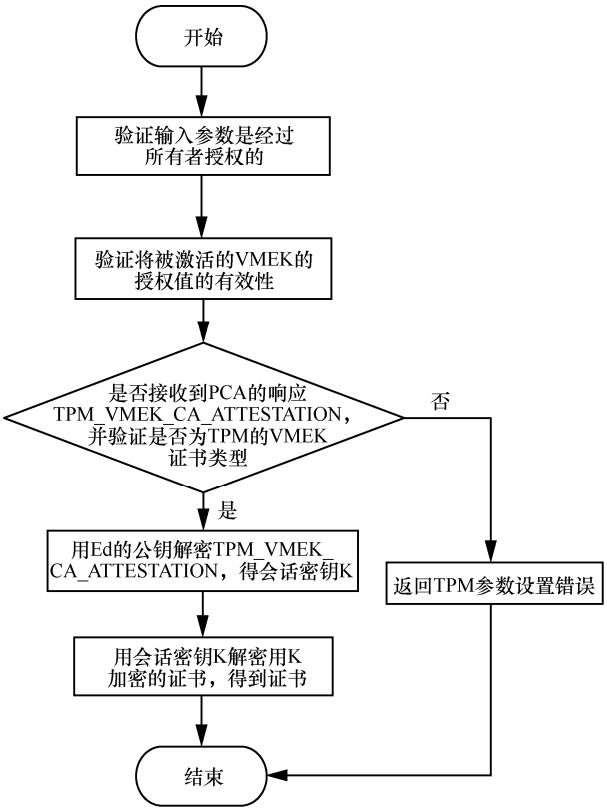


图 9　TPM_ActiveVMEK 实现流程

首先，TPM owner 验证从 PCA 接收到的 TPM_VMEK_CA_ATTESTATION 是否为当前自己申请的 VMEK 证书结构；然后，使用自己的 vEK 私钥解密加密证书的会话密钥；最后，使用该会话密钥解密证书密文，获得 VMEK 证书。

定义 3　TPM_Result TPM_VMEKLoad (

TPM_KEY_HANDLE SRKHandle, // 输入参数，SRK 的密钥句柄

TPM_KEY *inVMEK, //输入参数，VMEK 的私钥和公钥部分

TPM_AUTH *auth1, //输入输出参数，授权协议参数

TPM_KEY_HANDLE *inVMEKHandle, //输出参数，TPM 内部的 VMEK 句柄

);

该接口的主要作用是加载 VMEN 私钥到 TPM 内。如果执行成功，返回 TPM_SUCCESS；否则，返回 TPM 错误代码。具体流程如图 10 所示。

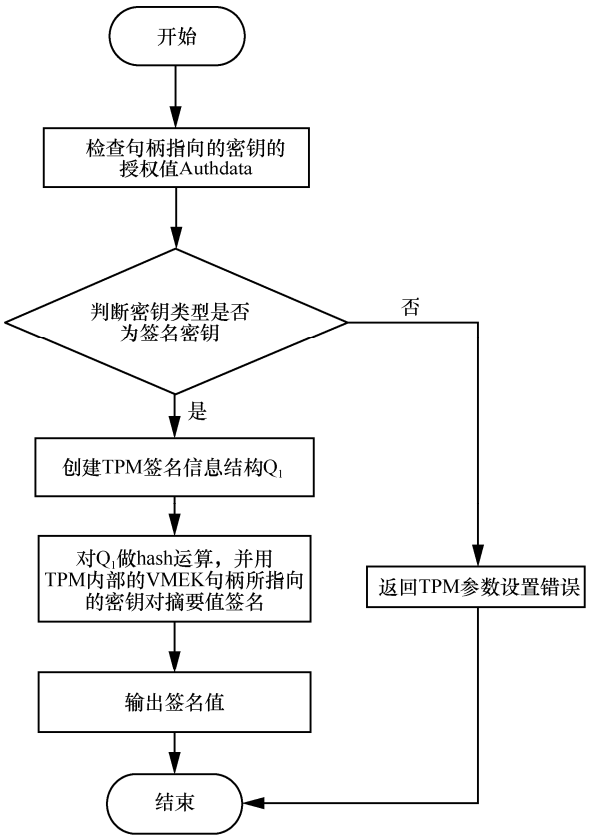


图 10　TPM_VMEKLoad 实现流程

首先，确保执行此操作是经过授权的；然后，验证准备加载的密钥的父密钥是否是 SRK，并用 SRK 对该密钥进行解密；接着，验证解密后密钥的属性；最后，把密钥装载到 TPM 内部存储器，并根据 TPM 规则给装载的密钥分配句柄。

定义 4　TPM_Result TPM_VMEK_Signing (

TPM_KEY_HANDLE VMEKHandle, //输入参数，VMEK 密钥句柄

TPM_NONCE *extrnalData, //输入参数，现时 nonce

UINT32 vEKbolb Size, //输入参数，vEK 密钥长度

TPM_PRIKEY *vEK, //输入参数，vEK 私钥

TPM_AUTH *auth1, //输入参数，授权协议参数

UINT32 *sigSize, //输出参数，签名值长度

BYTE **sig//输出参数，签名值

);

该接口的主要作用是用 VMEK 私钥对 vEK 签名。如果执行成功，返回 TPM_SUCCESS；否则，返回 TPM 错误代码。具体流程如图 11 所示。

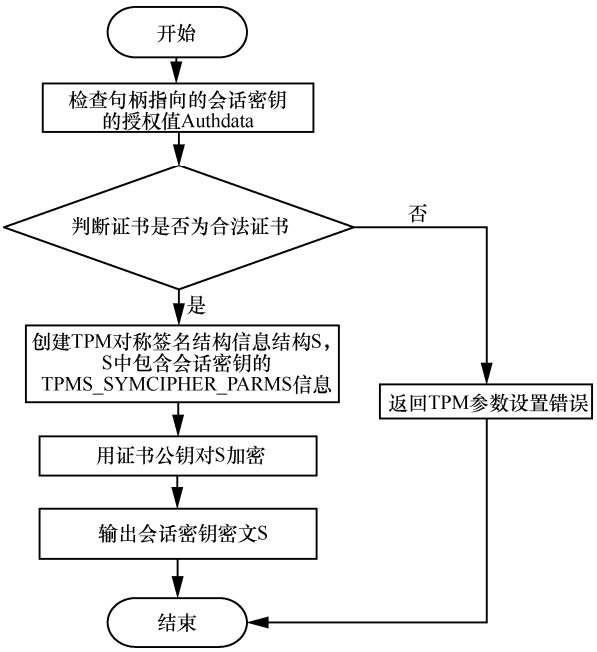


图 11 TPM_VMEK_Signing 实现流程

首先，确保执行 TPM_VMEK_Signing 操作是经过授权的；其次，确认 VMEK 的私钥是用作签名的；接着，创建 TPM_Sign_INFO 结构，记为 Q_1，并对 Q_1 做 hash 运算；最后，用 VMEK 的私钥对此 hash 结果签名，返回签名结果。

另外，VMEK 证书私钥的存储和使用如图 12 所示。从图 12 可以看出，VMEK 私钥存放在系统存储区，受 SRK 保护，TPM owner 可以通过相关命令使用 VMEK 私钥，应用程序则可以通过 TCS 使用 VMEK 私钥。特别需要指出的是，VMEK 的签名和加密可以由用户的应用程序调用实施。

4 基于 VMEK 的 vTPM 证书信任扩展

本节主要介绍基于 VMEK 的证书信任链扩展，称为 vTPM vEK to hTPM VMEK Binding 方案，并和已有 6 种方案进行比较。

4.1 vTPM vEK to hTPM VMEK Binding

对于 vTPM 的证书信任扩展，本文采用 hTPM VMEK signs vTPM vEK，即基于 VMEK 的 vTPM 证书信任扩展，如图 13 所示。该方案首先利用底层 TPM 的 VMEK 调用 TPM_VMEK_Signing 接口对 vTPM 的 vEK 签名，将底层信任传递到 vTPM，然后再由 vTPM 调用相关接口和 Privacy CA 一起生成 vAIK。本文方案的实质是用 VMEK 来替代 AIK，有以下 2 个优点，一是 VMEK 可以对 TPM 内外部信息进行签名，而 AIK 不能，不存在违反 TCG 规范的情况；二是相比 AIK，VMEK 不会频繁失效，从而不会引起其签名的失效，因而不会带来多余的性能负担。

具体的实施流程如下。

1) TPM 首先调用 TPM_CreateVMEKKeyPair 生成 VEMK 密钥对，然后调用 TPM_ActiveVMEK 生成 VMEK 证书，并激活。此步骤 TPM owner 只需操作一次。

2) 当虚拟平台产生新的 vTPM 时，会为 vTPM 生成 vEK，此时 TPM 调用 TPM_VMEK_ Signing 对 vEK 进行签名，将底层 TPM 的证书信任扩展到 vTPM。

3) vTPM owner 调用 TPM_MakeIdentity 命令生

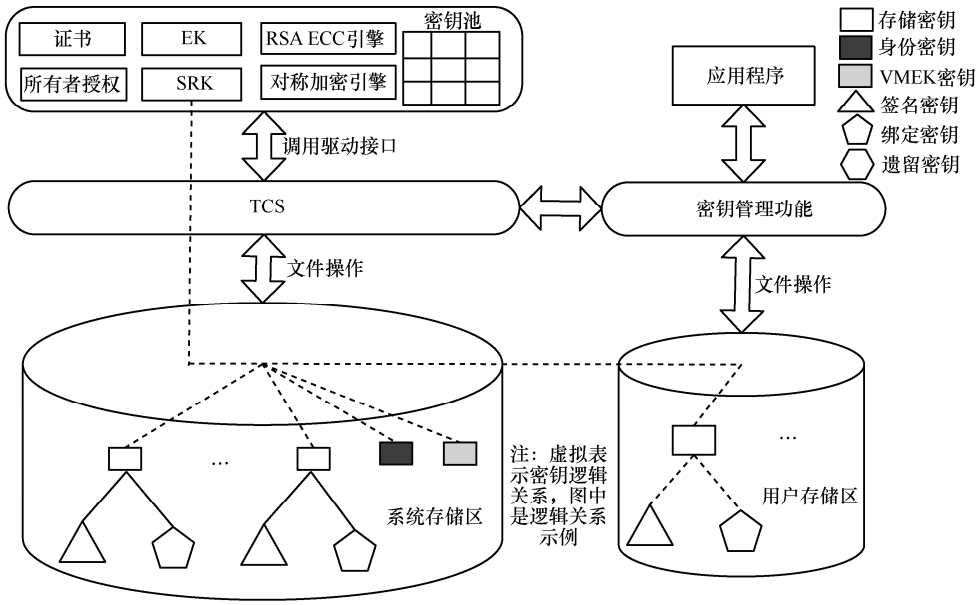


图 12 VMEK 证书私钥的存储和使用

成一对 vAIK 公私钥对（长度至少是 2 048 bit）。

4) vTPM owner 再调用 TPM_ActivateIdentity，获得 vAIK 证书。

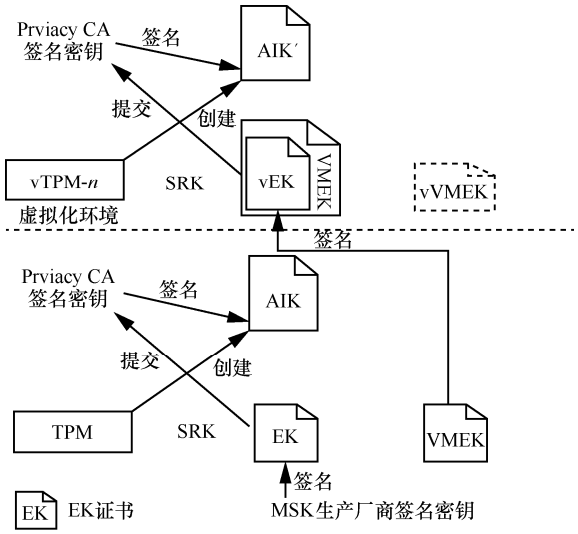


图 13　vTPM vAIKto hTPM VMEK Binding

4.2　本文方案与其他方案的比较分析

从 4.1 节可以看出，本文方案由于采用 vTPM vEK to hTPM VMEK Binding 方式进行证书信任扩展具有以下 4 个优点。

1) 不存在违背 TCG 规范的情况。

2) 没有产生密钥冗余。

3) 没有增加 Privacy CA 的性能负担。

4) 通过 VMEK 对 vTPM 的 vEK 进行签名，实现了底层 TPM 证书信任扩展到 vTPM。

vTPM vEK to hTPM AIK Binding 与本文方案相比，由于 vTPM vEK to hTPM AIK Binding 方案用 AIK 对 vEK 签名，存在违背 TCG 规范中 AIK 不能对 TPM 外部信息签名的要求，另外，AIK 容易失效，会导致 AIK 对 vEK 签名的失效，从而导致需重新向 Privacy CA 申请 vAIK，增加了 Privacy CA 的负担。而本文方案既不存在违背 TCG 规范的情况，也没有增加 Privacy CA 的负担。显然本文方案优于 vTPM vEK to hTPM AIK Binding。

hTPM AIK signs vTPM vAIK 与本文方案相比，由于 hTPM AIK signs vTPM vAIK 方案仍然用 AIK 对 vAIK 签名，同样存在违背 TCG 规范和增加 Privacy CA 负担的问题。显然本文方案优于 hTPM AIK signs vTPM vAIK。

Local CA issue vEK Certificate 与本文方案相比，由于 Local CA issue vEK Certificate 方案采用的是本地证书机关发放 vEK 证书，与 TPM 没发生绑定关系，即 TPM 的证书信任扩展没有传递到 vTPM，而本文方案不存在此类不足。

vTPM vEK to hTPM EK Binding 与本文方案相比，由于 vTPM vEK to hTPM EK Binding 方案用 EK 对 vEK 签名，违反了 TCG 对 EK 的使用规范，而本文方案不存在此类不足。

vTPM vAIK to hTPM SK Binding 与本文方案相比，由于 vTPM vAIK to hTPM SK Binding 方案需生成 SK 替代 AIK 对 vAIK 签名，这需要 SK 与 AIK 一一对应，当 AIK 失效时，需要生成新的 SK，带来较大的密钥冗余，增加了 TPM 的性能负担，而本文方案不存在此类不足。

hTPM EPS Product vEK 与本文方案相比，由于 hTPM EPS Product vEK 方案仅仅是通过 TPM 中的种子密钥 ESP 产生 vTPM 的 vEK，不存在信任扩展和传递的问题，即底层 TPM 的证书信任并没有通过 ESP 产生 vEK 传递到顶层的 vTPM 和虚拟机，而本文方案不存在此类不足。

表 4 是本文方案与其他 6 种方案的比较统计结果。

表 4　　本文方案与其他 6 种方案的比较统计结果

方案	是否遵守 TCG 规范	是否增加密钥冗余	是否增加 PCA 负担	信任是否扩展
vTPM vEK to hTPM AIK Binding	否	否	是	是
hTPM AIK signs vTPM vAIK	否	否	是	是
Local CA issue vEK Certificat	是	否	否	否
vTPM vEK to hTPM EK Binding	否	否	否	是
vTPM vAIK to hTPM SK Binding	是	是（多）	否	是
hTPM EPS Product vEK	是	否	否	否
本文方案	是	否（1 个）	否	是

5 在 Xen 平台中的实现

目前，在 Xen 4.4.0 实现了基于 VMEK 证书的信任链扩展。其中，特权管理域 Domain0 为 Ubuntu14.04 LTS，客户虚拟机操作系统为 Ubuntu14.04 LTS，并且为保证实验的可靠性和可操作性，选择 TPM_Emulator-0.7.4 对 TPM 环境进行仿真，TSS 软件栈为最新版本 TrouSerS0.3.14，并结合 IBM 发布的 IBM's TPM 2.0 TSS，对实验参数进行参考和调整。具体的实验设备配置如表 5 所示。

表 5 物理平台（Dom0）和用户虚拟机（DomU-Ubuntu）配置信息

配置项	物理平台（Dom0 特权域）	用户虚拟机（DomU-Ubuntu）
CPU	Intel Core i3 @3.4 GHz	Intel Core i3 @3.4 GHz
内核版本	Linux3.19.0	Linux3.19.0
内存	8 GB	4 GB
二级缓存	4 MB	4 MB
硬盘容量	1 TB	30 GB

5.1 VMEK 的实现

为了实现 VMEK 及其管理，对 TPM_Emulator-0.7.4 和 Xen 的源码做出了相应的增加，添加了相关的数据结构及管理接口。限于篇幅，本文仅列出数据结构和接口增加的具体位置，具体实现流程和接口功能描述可详见第 3 节和第 4 节。主要的实现如表 6 所示。

其中，在实现过程中涉及的一些辅助实现函数本文不再赘述。具体的实现步骤描述如下。

1) 在 tpm_emulator-0.7.4/tpm/tpm_structures.h 的 TPM_KEY_USAGE 中新增一种证书类型符号常量。

2) 在 tpm_emulator-0.7.4/tpm/tpm_structures.h 中添加 VMEK 证书，结构用于描述 VMEK 证书。

3) 在 tpm_emulator-0.7.4/tpm/tpm_structures.h 中定义数据结构 TPM_VMEK_CONTENTS。

4) 在 tpm_emulator-0.7.4/tpm/tpm_structures.h 中新增函数 TPM_CreateVMEKKeyPair、TPM_ActiveVMEK、TPM_VMEKLoad 和 TPM_VMEK_Signing 的接口定义。

5) 在 tpm_emulator-0.7.4/tpm/tpm_identity.c 和 tpm_emulator-0.7.4/tpm/tpm_vmekref.c 实现了 TPM_CreateVMEKKeyPair、TPM_ActiveVMEK、TPM_VMEKLoad 和 TPM_VMEK_Signing 函数。

6) 在 Xen 中//xen-4.4.0/xen/include/public/ xen.h，在虚拟机的信息结构体中添加 VMEK 的相关标识。

7) 在//xen-4.4.0/stubdom/下的 vtpm 和 vtpmmgr 中添加 VMEK 字段。

至此，TPM_Emulator-0.7.4 和 Xen 中 VMEK 及其管理实现完成。

5.2 基于 VMEK 的证书信任链扩展的实现

在 TrouSerS0.3.14 中对 TPM_CreateVMEKKeyPair、TPM_VMEKLoad、TPM_ActiveVMEK 和 TPM_VMEK_Signing 进行了封装。主要实现的接口如表 7 所示。

表 6 VMEK 在 Xen 中的实现

源码名称及版本	名称	文件路径	类型/作用
TPM_Emulator-0.7.4	TPM_KEY_USAGE	//tpm_emulator-0.7.4/tpm/tpm_structures.h	证书常量
	PM_KEY_VMEK		数据结构
	TPM_VMEK_CONTENTS		
	TPM_CreateVMEKKeyPair	//tpm_emulator-0.7.4/tpm/tpm_structures.h	函数定义
	TPM_ActiveVMEK		
	TPM_VMEKLoad		
	TPM_VMEK_Signing		
	TPM_CreateVMEKKeyPair	//tpm_emulator-0.7.4/tpm/tpm_identity.c	接口函数
	TPM_ActiveVMEK		
	TPM_VMEKLoad	//tpm_emulator-0.7.4/tpm/tpm_vmekref.c	
	TPM_VMEK_Signing		
Xen-4.4.0	VEMK_Info	//xen-4.4.0/xen/include/public/xen.h	vmek 标识
	VMEK_Info	//xen-4.4.0/stubdom/vtpmmgr/tpm.h	vmek 信息

表 7　　扩展 vTPM 管理器的 TSS 接口

应用层接口	TSP 层接口	TCS 层接口	TPM 命令
VTPM_CreateVMEKKeyPair	VTSP_CreateVMEKKeyPair	TCSP_CreateVMEKKeyPair	TPM_CreateVMEKKeyPair
VTPM_TPM_VMEKLoad	VTSP_VMEKLoad	TCSP_VMEKLoad	TPM_VMEKLoad
VTPM_ActiveVMEK	VTSP_ActiveVMEK	TCSP_ActiveVMEK	TPM_ActiveVMEK
VTPM_VMEK_Signing	VTSP_VMEK_Signing	TCSP_VMEK_Signing	TPM_VMEK_Signing

其中，VTPM_CreateVMEKKeyPair 函数用于在应用层创建 VMEK 公私钥对；VTPM_TPM_VMEKLoad 函数用于在应用层获取 VMEK 证书，并激活；VTPM_ActiveVMEK 函数用于在应用层加载 VMEN 私钥到 TPM 内；VTPM_VMEK_Signing 函数用于在应用层用 VMEK 私钥签名。

6　虚拟平台环境的远程证明测试

为验证 VMEK 设计和实现的有效性，本节对虚拟平台环境进行远程证明测试，主要分 VMEK 的生成、VMEK 对 vEK 证书的签名、vAIK 证书的生成和远程证明中证明方平台有效性验证 4 个方面。具体的物理平台配置参考表 5。

首先，测试了 VMEK 的生成，更改./tpm_emulator-0.7.4/tpm/tpm_testing.c 下的测试代码，测试 VMEK 的生成，实验结果如图 14 和图 15 所示。

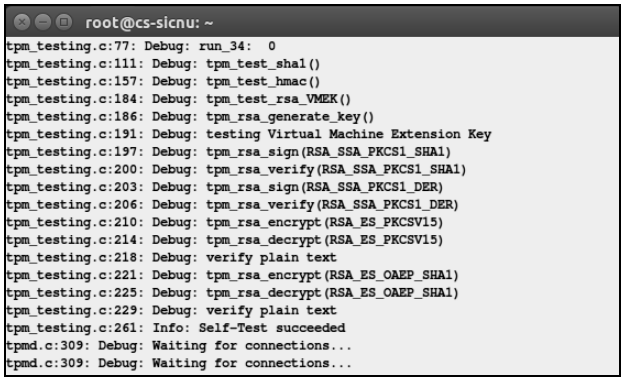


图 14　VMEK 相关函数测试

图 15　VMEK 证书部分内容（加密后）

其次，测试 VMEK 对 vEK 证书的签名，利用 TPM_VMEK_Signing()接口函数，对 vEK 进行签名，返回执行结果，如图 16 所示。

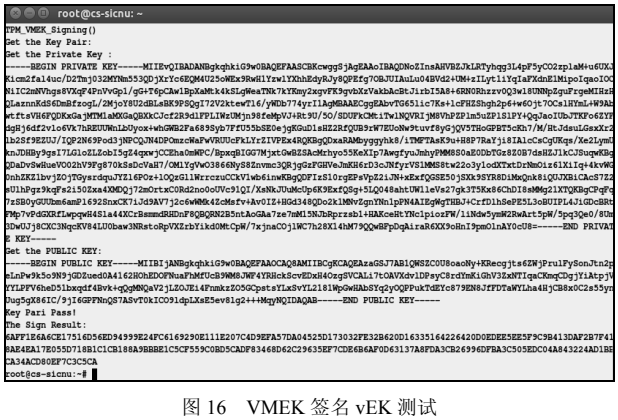


图 16　VMEK 签名 vEK 测试

第三，vAIK 证书的生成时间，并与物理 AIK 证书的生成时间做对比，其中，物理 AIK 证书的签发使用隐私 CA 测试网站，具体实验结果如表 8 所示。

表 8　生成 AIK 证书与 vAIK 证书时间对比

实验组序	生成 vAIK 证书时间/s			
	生成 VMEK 证书	签发 vEK 证书	签发 vAIK 证书	合计
第 1 组	10.03	3.11	3.20	16.34
第 2 组	10.02	3.10	3.20	16.32
第 3 组	10.04	3.10	3.21	16.35
第 4 组	10.03	3.10	3.19	16.32
第 5 组	10.03	3.09	3.20	16.32
平均值	10.03	3.10	3.20	16.33

最后，本节在虚拟计算环境下进行了远程证明实验，结果如图 17 所示。该远程证明采用基于二进制的直接模型，流程如图 18 所示。

验证方通过以下步骤确认证明方平台的可信性。

1) 通过验证 VMEK 对 vEK 的签名，可知 vAIK 是可信的，并且是由 vTPM 提供的。

2) 通过 VMEK 对 SK 的签名，可知 vTPM 是可信的，并且是受到物理 TPM 保护的。

```
root@cs-sicnu: ~
Are you sure to begin the remote attestation?(Y/N).....
Waiting......
Begin the Remote Attestation!
TPM_CreateVMEKKeyPair()              [SUCESS!]
TPM_ActiveVMEK()                     [SUCESS!]
TPM_VMEKLoad()                       [SUCESS!]
TPM_VMEK_Signing()                   [SUCESS!]
VMEK签名vEK                          [SUCESS!]
发送vEK                              [SUCESS!]
请求vAIK证书                         [SUCESS!]
验证vEK私钥                          [SUCESS!]
接收Quote原数据结构                  [SUCESS!]
接收SML                              [SUCESS!]
接收vAIK证书                         [SUCESS!]
接收隐私CA公钥                       [SUCESS!]
获取vAIK证书                         [SUCESS!]
验证Quote签名                        [SUCESS!]
验证CertifyInfo签名                  [SUCESS!]
验证签名密钥                         [SUCESS!]
验证vAIK公钥签名                     [SUCESS!]
验证DomU Quote签名                   [SUCESS!]
验证SML                              [SUCESS!]
Read the PCR
Waiting.........
Read From VMEK | PCR[0-9] values are :
Index  Content
PCR 00 : 61 51 74 84 E8 04 89 37 CF A3 AC EC 30 E0 84 CB 73 CF A7 D3
PCR 01 : 8E 01 EF 62 2A 03 7E 3A 07 10 49 C4 8E BA 08 BA 2F AB 56 0D
PCR 02 : 40 1B 43 5F 09 38 A8 4B 5F 17 E1 F0 72 FA 92 4B 89 A7 04 FE
PCR 03 : BA 9D BD 56 12 39 9D 2D 2D 54 4E B5 A5 EC B8 ED F7 21 89 57
PCR 04 : C5 9E 0C 5A 03 97 4C 69 20 D7 7A 36 2F 3A EB 93 F1 91 19 49
PCR 05 : D1 AB E4 E2 E5 5A 94 27 30 BD F6 41 A9 67 60 F1 73 19 A6 3B
PCR 06 : 42 D3 6B A7 1A F6 58 71 D0 B3 90 08 6B 05 E9 D3 8A 84 5F 96
PCR 07 : 3B 97 B5 10 DB 32 AF 8A 8E D0 62 82 A3 53 01 AD 0D AB 31 4B
PCR 08 : 21 CF 92 76 6F 76 28 F0 B4 15 7E 8D AD 7F 79 AB 78 BF 8A F6
PCR 09 : 42 BF 21 C9 D9 96 61 BE 72 E6 8C 8A 05 8A 4B ED 40 45 DE D2
Read From VM vTPM | PCR[10-15] values are :
Index  Content
PCR 10 : F8 79 91 CB CF BD B8 FB 62 C6 6B 34 5C DA EF 85 2A CF 61 7A
PCR 11 : B1 45 61 4A 39 8A 38 E1 18 48 8B 8D 76 AE E8 7D FC A0 19 4E
PCR 12 : 1E D3 F7 68 28 9D 40 28 07 5F 27 5D 21 F4 E6 9A 5A CE B5 8D
PCR 13 : D6 49 B2 45 62 FB 70 6D 6D A6 2D FD 68 F5 7C D8 E7 5E 92 26
PCR 14 : 7D C0 4F C5 32 C9 27 78 9D 70 F4 DB B3 4F 0E DE CA 86 D0 07
PCR 15 : 82 8B 1A 58 98 57 34 E4 A1 AE B2 6F B1 3E A7 3B A8 23 6A 24
验证Nonce                            [SUCESS!]
验证共享密钥                         [SUCESS!]
Waiting...
The Remote Attestation process has done!
root@cs-sicnu:~#
```

图 17　虚拟平台下的远程证明实现

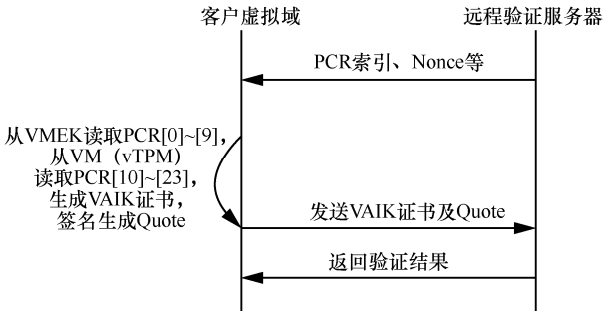


图 18　远程证明流程

3) 通过验证 VMEK 证书，证明物理 TPM 真实可信。

4) 通过验证物理平台的 Quote 可知，当前虚拟平台的运行环境（VMM 及特权域）是真实可信的。

5)验证客户虚拟域 Domain U 的 Quote 签名，以验证当前虚拟执行环境的可信性。

值得说明的是，在 TCG 规范中，TPM 共定义了 5 种证书，学术界认为其过于复杂，在我国的可信计算标准规范中，已减少到 3 种证书。本文又增加了一种新的证书，现任提升了管理的复杂度。不过，本文提出的 VMEK 证书主要针对虚拟平台环境，是一种可选的证书，因此，对非虚拟化平台环境，只要用户屏蔽不用，几乎没有额外负担。

7　结束语

通过新增一类证书——VMEK，解决了在虚拟平台环境中可信证书链扩展时存在的违背 TCG 规范、增加密钥冗余或 Privacy CA 的性能负担问题。

下一步的工作是将 VMEK 应用到 vTPM 或 TPM 的密钥迁移中。TPM 或 vTPM 允许其可迁移密钥从一个平台迁移到另一个平台，在迁移过程中，VMEK 可以用来确保该可迁移密钥公钥的正确性和安全性。

参考文献：

[1] ZHANG Y, ZHOU Y. 4VP: A novel meta OS approach for streaming programs in ubiquitous computing[C]//International Conference on Advanced Information NETWORKING and Applications. 2007: 394-403.

[2] ZHANG Y, ZHOU Y. Transparent computing: a new paradigm for pervasive computing[C]//International Conference on Ubiquitous Intelligence and Computing. 2006: 1-11.

[3] 陈康, 郑纬民. 云计算: 系统实例与研究现状[J]. 软件学报, 2009, 20(5): 1337-1348.
CHEN K, ZHENG W M. Cloud computing: system case and research status[J]. Journal of Software, 2009, 20(5): 1337-1348.

[4] 罗军舟, 金嘉晖, 宋爱波, 等. 云计算: 体系架构与关键技术[J]. 通信学报, 2011, 32(7): 3-21.
LUO J Z, JIN J H, SONG A B, et al. Cloud computing: architecture and key technologies[J]. Journal on Communications, 2011, 32(7): 3-21.

[5] 林闯, 苏文博, 孟坤, 等. 云计算安全:架构、机制与模型评价[J]. 计算机学报, 2013, 36(9):1765-1784.
LIN C, SU W B, MENG K, et al. Cloud computing security: architecture, mechanism and model evaluation[J]. Chinese Journal of Computers, 2013, 36(9): 1765-1784.

[6] 王国峰, 刘川意, 潘鹤中, 等. 云计算模式内部威胁综述[J]. 计算机学报, 2017, 40(2): 296-316.
WANG G F, LIU C Y, PAN H Z, et al. An overview of internal threats in cloud computing models[J]. Chinese Journal of Computers, 2017, 40(2): 296-316.

[7] MAHAJAN A, SHARMA S. The malicious insiders threat in the cloud[J]. International Journal of Engineering Research and General Science, 2015, 3(2): 245-256.

[8] BOUCHÉ J, KAPPES M. Attacking the cloud from an insider perspective[C]//Internet Technologies and Applications. 2015.

[9] 王焘, 张文博, 魏峻, 等. 一种基于故障预测的云计算系统自适应监测方法[P]. CN105677538A, 2016.
WANG H, ZHANG W B, WEI J, et al. An adaptive monitoring method for cloud computing systems based on fault prediction[P]. CN105677538A, 2016.

[10] 沈昌祥, 张焕国, 王怀民, 等. 可信计算的研究与发展[J]. 中国科

学: 信息科学, 2010(2): 139-166.
SHEN C X, ZHANG H G, WANG H M, et al. Research and development of trusted computing[J]. Chinese Science: Information Science, 2010(2): 139-166.

[11] 冯登国, 秦宇, 汪丹, 等. 可信计算技术研究[J]. 计算机研究与发展, 2011, 48(8): 1332-1349.
FENG D G, QIN Y, WANG D, et al. Research on trusted computing technology[J]. Journal of Computer Research and Development, 2011, 48(8): 1332-1349.

[12] CHEN Y, PAXSON V, KATZ R H. What's new about cloud computing security?[J]. 2014, 20.

[13] KO R K L, JAGADPRAMANA P, MOWBRAY M, et al. Trust cloud: a framework for accountability and trust in cloud computing[C]// Services. 2011: 584-588.

[14] 刘川意, 王国峰, 林杰, 等. 可信的云计算运行环境构建和审计[J]. 计算机学报, 2016, 39(2): 339-350.
LIU C Y, WANG G F, LIN J, et al. Trusted cloud computing operating environment construction and auditing[J]. Chinese Journal of Computers, 2016, 39(2): 339-350.

[15] 田俊峰, 常方舒. 基于 TPM 联盟的可信云平台管理模型[J]. 通信学报, 2016, 37(2): 1-10.
TIAN J F, CHANG F S. Trusted cloud platform management model based on TPM alliance[J]. Journal on Communications, 2016, 37(2): 1-10.

[16] 吴吉义, 沈千里, 章剑林, 等. 云计算: 从云安全到可信云[J]. 计算机研究与发展, 2011, 48(S1): 229-233.
WU J Y, SHEN Q L, ZHANG J L, et al. Cloud computing: from cloud security to trusted clouds[J]. Journal of Computer Research and Development, 2011, 48(S1): 229-233.

[17] BERGER S, GOLDMAN K A, PEREZ R, et al. vTPM: virtualizing the trusted platform module[C]//Conference on Usenix Security Symposium. 2006: 21.

[18] ENGLAND P, LOESER J. Para-virtualized TPM sharing[C]// International Conference on Trusted Computing and Trust in Information Technologies: Trusted Computing-Challenges and Applications. 2008: 119-132.

[19] STUMPF F, ECKERT C. Enhancing trusted platform modules with hardware-based virtualization techniques[C]// Second International Conference on Emerging Security Information, Systems and Technologies. 2008: 1-9.

[20] ALBELOOSHI B, SALAH K, MARTIN T, et al. Securing cryptographic keys in the IaaS cloud model[C]//IEEE/ACM International Conference on Utility and Cloud Computing. 2016: 42-56.

[21] YU Z, WANG Q, ZHANG W, et al. A cloud certificate authority architecture for virtual machines with trusted platform module[C]//IEEE International Conference on High PERFORMANCE Computing and Communications. 2015: 1377-1380.

[22] CHANG D, CHU X, QIN Y, et al. TSD: a flexible root of trust for the cloud[C]//IEEE International Conference on Trust, Security and Privacy in Computing and Communications. 2012: 119-126.

[23] WAN X, XIAO Z, REN Y. Building trust into cloud computing using virtualization of TPM[C]//Fourth International Conference on Multimedia Information NETWORKING and Security. 2013: 59-63.

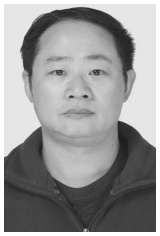
[24] XUE D, WU X, GAO Y, et al. TrustVP: construction and evolution of trusted chain on virtualization computing platform[C]//Eighth International Conference on Computational Intelligence and Security. 2013: 623-630.

[25] GOYETTE R. A review of "vTPM: virtualizing the trusted platform module"[R]. Network Security and Cryptography Symposium, 2007: 1-17.

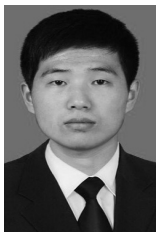
[26] 王丽娜, 高汉军, 余荣威, 等. 基于信任扩展的可信虚拟执行环境构建方法研究[J]. 通信学报, 2011, 32(9): 1-8.
WANG L N, GAO H J, YU R W, et al. Research on the construction method of trusted virtual execution environment based on trust extension[J]. Journal on Communications, 2011, 32(9): 1-8.

[27] 杨永娇, 严飞, 毛军鹏, 等. Ng-vTPM: 新一代 TPM 虚拟化框架设计[J]. 武汉大学学报(理学版), 2015, 61(2): 103-111.
YANG Y J, YAN F, MAO J P, et al. Ng-vTPM: a new generation of TPM virtualization framework design[J]. Journal of Wuhan University (Science Materials), 2015, 61(2): 103-111.

[作者简介]



谭良（1972-），男，四川泸州人，博士，四川师范大学教授，主要研究方向为可信计算、网络安全、云计算及大数据处理等。



齐能（1993-），男，河南商丘人，四川师范大学硕士生，主要研究方向为可信计算。

胡玲碧（1993-），女，四川威远人，四川师范大学硕士生，主要研究方向为可信计算。