doi:10.11959/j.issn.1000-436x.2017183

# 基于 SGX 的虚拟机动态迁移安全增强方法

石源[1,2]，张焕国[1,2]，赵波[1,2]，于钊[1,2]
（1. 武汉大学计算机学院，湖北 武汉 430072；2. 武汉大学空天信息安全与可信计算教育部重点实验室，湖北 武汉 430072）

**摘 要**：针对虚拟机动态迁移面临的虚拟机信息泄露的安全问题，引入内存动态保护技术 SGX，基于 KVM 虚拟化环境，提出一种动态迁移安全增强方法。在迁移两端构建以 SGX 技术为核心的硬件隔离的安全执行环境，保障加密、完整性度量等安全操作和秘密数据的安全。通过迁移双方的安全执行环境之间的远程证明，建立一个用于传输迁移数据的加密信道，并在此基础之上实现迁移双方的平台完整性的相互验证。最后分析该方法的安全增强效果，并通过实验验证了 SGX 技术的引入不会对迁移造成过多的性能损耗。

**关键词**：虚拟化；动态迁移；Intel SGX；远程证明；完整性度量
**中图分类号**：TP391 **文献标识码**：A

# Security-enhanced live migration based on SGX for virtual machine

SHI Yuan[1,2], ZHANG Huan-guo[1,2], ZHAO Bo[1,2], YU Zhao[1,2]
(1. School of Computer, Wuhan University, Wuhan 430072, China;
2. Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, Wuhan University, Wuhan 430072, China)

**Abstract:** The virtual machine may face the problem of information leakage in live migration. Therefore, a dynamic memory protection technique SGX was introduced and a security enhancement live migration method based on KVM environment was proposed. Firstly, on both sides of migration, a hardware-isolated secure execution environment centered SGX was built. It guaranteed the security of operations like encryption and integrity measurement and also ensured the security of private data. An encrypted channel to transfer migration data based on the remote attestation between the secure execution environments of both migration sides was constructed. And the mutual authentication of both sides' platform integrity was realized. Finally, the security enhancement effect and did the experiment was analyzed. The results shows that the introduction of SGX won't cause much negative effect to the migration performance.

**Key words:** virtualization, live migration, Intel SGX, remote attestation, integrity measurement

## 1 引言

随着云计算的兴起与发展，虚拟化技术得到了广泛的研究与应用。虚拟化技术为云平台实现资源抽象、隔离以及资源的按需分配提供技术支撑。在云虚拟化架构中，虚拟机是系统资源虚拟化的直观体现，也与云用户密切相关。

虚拟机动态迁移是构建可靠云平台的重要需求之一。通过动态迁移技术，可以将虚拟机从负载过高的主机迁移到负载较小的主机，以实现负载均衡；当某一主机出现问题时，可以将该主机的虚拟机动态地迁移至其他主机，保障虚拟机的正常运行。对于虚拟机动态迁移的研究大都集中在提升性能方面[1]，主流的虚拟化平台包括 Xen、KVM 都有

**收稿日期**：2016-11-10；**修回日期**：2017-02-22
**通信作者**：张焕国，liss@whu.edu.cn
**基金项目**：国家自然科学基金重点资助项目（No.61332019）；国家重点基础研究发展计划（“973”计划）基金资助项目(No.2014CB340600)；国家高技术研究发展计划（“863”计划）基金资助项目(No.2015AA016002)
**Foundation Items:** The National Natural Science Foundation of China (No.61332019), The National Basic Research Program of China (973 Program) (No.2014CB340600), The National High Technology Research and Development Program of China (863 Program) (No.2015AA016002)

适用于自身平台的完善的动态迁移技术，但是随着网络空间安全问题的日益严峻[2]，虚拟机迁移的安全问题也开始受到人们的重视。

虚拟机动态迁移的安全问题涉及了云计算系统的多个层次。文献[3]通过实验验证了虚拟机动态迁移的安全问题主要源于3个层次：虚拟机监控器（VMM，virtual machine monitor）层、数据层和迁移模块层。其中，VMM层属于虚拟化的核心层，攻击者可能利用VMM的漏洞获得其控制权限[4]以达到控制虚拟机迁移甚至控制虚拟机的目的。数据层包含了迁移数据在迁移两端的存储和处理以及迁移双方的数据传输，攻击者可以利用内存泄露[5]或窥探传输信道来获取虚拟机的隐私信息。迁移模块的安全性直接决定着迁移的过程是否可信，攻击者可以利用迁移模块的漏洞实现对迁移的控制或者直接窃取虚拟机的隐私信息。

针对VMM的安全问题，文献[6]提出了一种基于网络安全引擎（NES-H, network security engine-hypervisor）的动态迁移框架，NES-H是对hypervisor的一种扩展，它包含防火墙、入侵检测等防御机制，可以在一定程度上增强迁移本地环境的安全。针对虚拟机迁移数据传输不安全的问题，文献[7]提出基于SSH通道的在线迁移方案，保证迁移过程的安全；文献[8]提出使用RSA算法和SSL来保证数据传输的安全；文献[9]提出使用IPsec实现迁移主机之间的安全传输；文献[10]在迁移流程中加入了混合随机变换编码机制来保障迁移数据的安全。为了增强迁移整体的可信性，文献[11]提出了基于可信令牌的方案，只有满足安全策略的云平台才能执行迁移操作；文献[12]设计了一种基于角色/策略的迁移方法，该方法通过安全组件实现迁移主机之间的安全认证，确保迁移双发的真实可信。

现有的动态迁移安全增强方案虽然能在一定程度上保障动态迁移的安全性，但是普遍存在一些不足：若VMM、操作系统以及迁移模块等特权软件存在漏洞，可能会引发内容泄露攻击，此时包括加密、完整性度量等安全措施以及其他安全组件都会受到攻击的影响而失效；即使通信信道被加密，由于加密的过程可能在一个不安全的环境下进行，那么密钥很可能在攻击者探测信道之前就已经泄露，同理，平台完整性度量的过程和度量信息的交互也就得不到安全保障。针对上述问题，本文基于内存动态保护技术SGX提出了虚拟机动态迁移安全增强方法。该方法基于SGX提供的保护机制，为虚拟机动态迁移的安全相关模块提供硬件隔离的执行环境Enclave，保障迁移数据加密和平台完整性度量等操作的安全；通过迁移双发Enclave之间的双向远程证明建立一个加密信道，并通过该信道交互迁移双方的平台完整性信息，以实现双向的平台完整性验证，验证通过后再通过加密信道传输加密的迁移数据；最后分析了本文方法的安全性并通过实验验证了该方法不会对虚拟机迁移产生过多额外的性能开销。

## 2 背景知识与威胁模型

### 2.1 SGX Enclave保护机制

Intel于2013年推出了一套CPU架构的新扩展SGX，它增加了一组新的指令集和内存访问机制，旨为用户空间提供由硬件安全保障的可信执行环境（TEE, trusted execution environment）。如图1所示，新的扩展允许将合法的应用程序的安全操作封装在一个Enclave[13~15]的容器中，在应用程序地址空间划分出一块被保护的区域，为容器内的代码和数据提供机密性和完整性保护，免受恶意软件破坏。只有位于容器内部的代码才能访问该Enclave所在的内存区域，而容器之外的软件包括特权软件（如VMM、BIOS、操作系统）和非特权软件都不能访问Enclave内部数据。除了提供内存隔离与保护安全属性，SGX架构还支持远程认证和密封的功能，可用于安全软件应用和交互协议的设计。

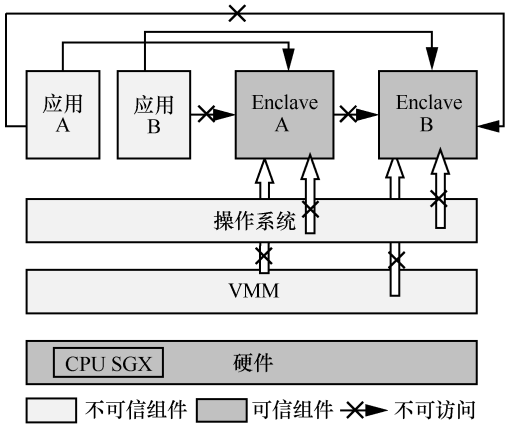



图1 SGX基本原理

Enclave保护机制主要包括2个部分：1) Enclave内存访问语义，2) 程序地址映射关系的保护。这2项功能共同完成对Enclave的机密性和完整性的保护。

1) 内存访问语义。在系统内分配一块被保护的物理内存区域 EPC（Enclave page cache），用来存放重要的数据结构。内存保护机制在物理上锁住 EPC 内存区域，将外部的访问请求视为引用了不存在的内存，使外部的实体无法访问。

2) 地址映射保护。EPC 内存以页为单位进行管理，将页的控制信息保存在控制结构 EPCM 里，这类似于操作系统内的页表，存放 EPC 页面的基本信息，如是否已被使用、页的拥有者、页类型、地址映射和权限属性等。EPCM 结构在保护模式的段/页机制的基础上，执行 Enclave 页的访问控制。

应用程序在申请创建一个 Enclave 时，会进行页面分配、复制程序代码与数据和度量操作，最后会对 Enclave 的完整性进行验证，判断特权软件在创建过程中是否篡改了程序数据。成功执行了初始化过程后，才能进入 Enclave 执行代码，此后内存保护和地址映射保护使外界无法访问 Enclave 内存，从而保证 Enclave 的机密性和完整性。

### 2.2 SGX Enclave 证明

为了向第三方证明当前的某个程序正在被平台的 Encalve 所保护的，平台通过向第三方提供能够反映当前平台 Encalve 的可信性和签名的凭证，使第三方可以通过验证凭证信任平台的程序是可信的受 SGX 保护的，之后第三方可以向该程序提供秘密信息和需要保护的数据。SGX 支持 2 种类型的证明方式[13]：1)平台内部 Enclave 间的证明，同一平台的某个 Enclave 可以向另一个 Enclave 证明自己的身份和真实性；2)平台间的远程证明，用于远程的认证者验证 Enclave 的身份信息。

在平台内部的本地证明过程中，当 Enclave A（简称 A）向 Enclave B（简称 B）进行证明时，A 首先向硬件请求产生一个报告结构 REPORT，该结构主要由 A 的身份信息和属性、平台硬件 TCB 等安全相关信息、用户数据和以上内容的签名组成。REPORT 生成之后交由 B 来验证 A 的身份和真实性。

基于 Enclave 的远程证明需要引入一个特殊的 Quoting Enclave。如图 2 所示，Quoting Enclave 创建平台认证的签名密钥 EPID，这个密钥不仅代表平台还代表着底层硬件的可信度，并且绑定处理器固件的版本，当 Enclave 系统运行时，只有 Quoting Enclave 才能访问到 EPID 密钥。远程证明的过程中，假设远程平台 B 要认证程序 Enclave A，A 同样要向硬件请求产生一个报告结构 REPORT，Quoting Enclave 首先通过 REPORT 验证 A 是否运行于同一平台，验证通过后，由 Quoting Enclave 将 REPORT 封装成一种能够代表 Enclave 和平台状态信息的 QUOTE 结构，该结构的主要组成实际上就是 REPORT，只是包含了更详细的 Enclave 的信息。然后 Quoting Enclave 使用 EPID 密钥对 QUOTE 结构进行签名并发送到远程平台供远程证明使用。

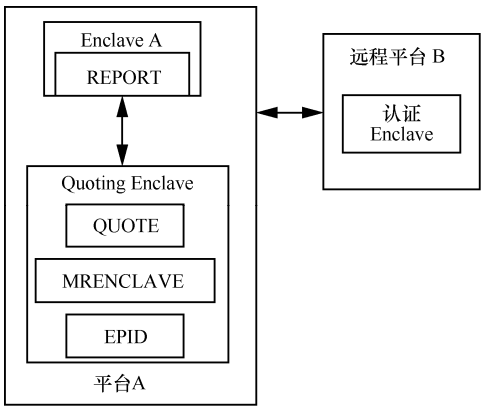



图 2　Enclave 远程证明交互流程

## 3 威胁模型

如何为迁移相关操作提供一个安全可信的执行环境以及如何有效保护虚拟机迁移数据是亟待解决的问题。基于上述问题，本文假设平台的处理器是可信的，不会被篡改的，假设 CA 是可信的，系统中的 Enclave 公钥证书默认都是可信的，而 BIOS、操作系统以及 VMM 等软件系统是不可信的。假设攻击者可以监听、控制网络流量。攻击者可以进行除物理攻击之外的攻击，例如，利用系统或软件漏洞对 BIOS、操作系统、VMM 等进行攻击，攻击者可以修改系统配置以及植入恶意程序，进而窥探或篡改内存、破坏系统的正常运行流程。假设 Enclave 机制以及 Enclave 内部的程序和数据都是可信的，而外部的都是不可信的。

## 4 基于 SGX 的动态迁移安全增强系统

系统的基本思路将与迁移安全相关的关键模块（迁移加密模块、平台度量模块）放到安全隔离的 Enclave 中并将相关的密钥等隐私数据绑定至 Enclave。基于 SGX 以及 Enclave 是绝对安全的，即使系统内部被攻击或篡改，也可以保障上述模块和操作的安全隔离，防止密钥等隐私信息被窃取。

在 KVM/Qemu 的虚拟化环境下原有的迁移框

架的基础之上，结合 SGX 的安全特性，本文设计了安全增强的迁移系统，如图 3 所示。以迁移源主机为例，在硬件资源层，CPU 支持 Intel SGX，EPC 是物理内存的一部分，通过 SGX 技术将其划分为 Enclave 独占的存储空间，用来存放 Enclave 所属应用程序的代码和数据。在 Linux 内核层，包含 KVM 模块等虚拟化组件以及为上层应用提供创建、增加、删除以及销毁 Enclave 等功能的 SGX 驱动。在用户层，虚拟机迁移的相关工作由 Qemu 来实现，将 Qemu 中与迁移相关的模块运行在隔离的可信迁移 Enclave 中，即可对迁移的相关操作进行一个动态的隔离的保护，有效防止内存泄露的攻击。

SGX 程序设计需要对软件架构重新进行划分，每个软件都需要被划分成 2 个逻辑部分[15]：可信部分（Enclave 部分）和不可信部分（除去可信部分剩下的部分）。可信部分用于执行密钥生成、密码运算、保存敏感数据等安全相关的操作；不可信部分则负责程序剩余流程。可信部分的代码段和数据段位于 EPC 中，受到 SGX 保护，而不可信部分的代码和数据则位于一般的内存中。因此，本文将原本由 Qemu 实现的平台度量、迁移加密 2 个安全功能隔离出来，将它们置于可信的 Enclave 中（这里命名为可信迁移 Enclave，TM-Enclave），以实现安全操作的隔离。

TM-Enclave 中的度量模块负责实现对平台的完整性度量，在迁移双方的 TM-Enclave 相互远程证明之后，通过 Enclave 之间的加密信道可以安全地传输度量信息，以验证双方平台的完整性是否满足迁移的安全条件。加密模块负责对与隐私相关的数据进行隔离的安全加密，为了保证迁移效率，优先考虑加密那些涉及虚拟机密钥的设备，如虚拟可信平台模块[16,17]（vTPM, virtual trusted platform module）。vTPM 的引入增强了虚拟化环境的安全性，但是 vTPM 虚拟机迁移需要额外考虑到 vTPM 实例数据的安全迁移[18]，在 KVM 架构下的 vTPM 设备包含一个用于存储 vTPM 密钥的 Qcow2 格式的镜像文件[17]，而且虚拟机进程也包含了该设备的隐私信息，因此，该设备的镜像文件和状态信息在迁移的时候都需要得到保护。

### 4.1 基于 Enclave 的平台完整性双向验证

在开始虚拟机迁移之前，迁移双方应该确认对方平台是安全可靠的，而判别的依据则是迁移系统的关键数据和代码的完整性，包括迁移模块、内核关键模块以及 VMM 的关键模块的数据和代码的完整性。上述模块除更新之外都不会被系统的正常流程而更改，并且初始度量值保存在度量日志中，度量日志仅能被 TM-Enclave 读写。

与一般的度量过程不同，本文的所有度量操作都是在隔离的 TM-Enclave 中实现，确保操作的安全可靠。同时，迁移双发的度量日志的交互也基于 Enclave 实现的。在迁移双方传递度量信息之前，首先双发的 TM-Enclave 之间要进行远程证明，只

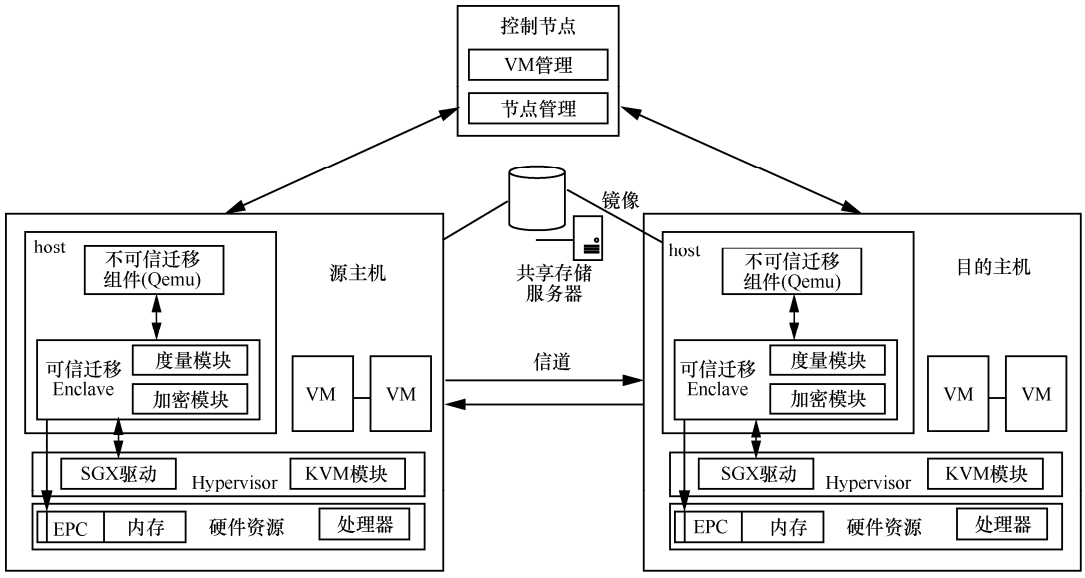



图 3 基于 SGX 的虚拟机动态迁移安全增强系统框架

有验证通过才能开始交互。在远程证明的过程中，双发交互 2 个共享的密钥种子，用于生成加密通信信道的密钥。随后迁移双方各自基于 TM-Enclave 进行平台完整性度量，并将度量结果发送给对方，相互验证对方的完整性，确保通信双方都是可信实体，未受到攻击者的破坏。基于 Enclave 的平台完整性双向验证分为 2 个阶段。

第 1 阶段包括双向的 Enclave 远程证明以及安全信道的建立，以单向远程证明的过程为例说明交互的过程，如图 4 所示，具体描述如下。

1) 源主机向目的主机的迁移管理工具（Qemu）发起一个挑战，挑战中包含 nonce；目的主机的 Quoting Enclave 向 Qemu 提供它的身份信息（Enclave identity），Qemu 将该信息和源主机发送过来的挑战一起发送给 TM-Enclave。

2) 目的主机的 TM-Enclave 响应挑战并附加一个用于生成加密密钥的种子 *Seed_A*，将上述数据生成一个散列摘要，并将其作为用户数据域（userdata）的参数，然后由 EREPORT 指令，将上述参数和目的端 Enclave 的身份信息组合生成一个将上述数据绑定至 TM-Enclave 的 REPORT 结构；TM-Enclave 将 REPORT 发送给 Qemu，再转发给 Quoting Enclave。

3) Quoting Enclave 验证 REPORT，然后创建一个 Quote 结构并利用它的 EPID 密钥对该结构进行签名。Quoting Enclave 将签名后的 Quote 返回给 Qemu；Qemu 再将 Quote 结构和所有与源主机相关的信息发送到源主机。

4) 源主机使用 EPID 的公钥证书来验证 Quote 签名。然后再通过 userdata 来验证响应信息的完整性；当所有验证通过之后，源主机的 TM-Enclave 就将目的主机发送过来的密钥种子 *Seed_B* 密封至 TM-Enclave。

5) 目的主机再向源主机发起一个挑战，然后重复类似于步骤 1)~步骤 4)，目的主机获得源主机发送过来的密钥种子 *Seed_B*。至此，迁移双方都拥有了密钥种子 *Seed_A* 和 *Seed_B*，双方使用相同的密钥种子和算法生成相同的密钥，该密钥即用来加密迁移信道。

双向证明的主要目的是为了使迁移双方信任对方的 Enclave 环境，再分别将用于生成加密密钥的种子通过各自 TM-Enclave 的私钥加密，然后发送给对方。因为 TM-Enclave 中的私钥对都被隔离保护，所以可以确保密钥种子不会被窃取，也就保障了加密密钥的机密性。

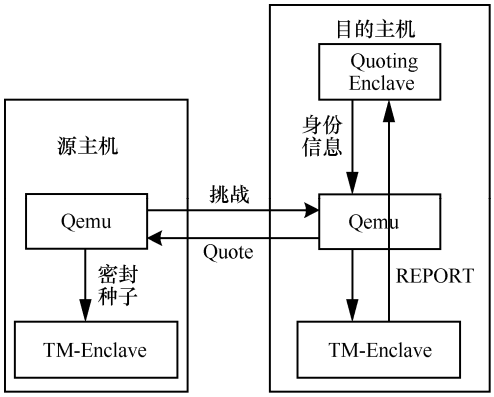



图 4　Enclave 远程单向认证流程

第 2 阶段为平台完整性验证。该阶段使用第 1 阶段共享的种子生成的迁移密钥作为通信密钥，这里命名为 $Key_{\text{Enclave}}$。过程如图 5 所示，详细描述如下（加密、度量操作都在 TM-Enclave 中执行）。

1) 源主机向目的主机发送远程度量请求 $Rquest_{\text{meas}}(S)$，附加源主机 TM-Enclave 生成的随机数 $nonce_1$，即 $M_1=Key_{\text{Enclave}}(Request_{\text{meas}}(S)||Nonce_1)$。将 $M_1$ 进行散列运算并将结果附在 $M_1$ 上，加密后发送到目的主机。

2) 目的主机先解密消息，然后计算并验证 $M_1$ 的摘要值；验证通过后，开始本机内核与 VMM 关键代码和数据完整性的度量，将度量结果保存至度量日志，并由 TM-Enclave 对度量日志进行签名。

3) 将签名后的度量日志、目的端的度量请求 $Request_{\text{meas}}(D)$、目的端的 TM-Enclave 生成的随机数 $Nonce_2$ 和收到 $Nonce_1$ 发送给源主机，记为 $M_2=Key_{\text{Enclave}}(Key_{sign(D)}(Logs_1)||Request_{\text{meas}}(D)||Noce_1||Nonce_2)$。

4) 源主机收到 $M_2$ 后，对目的主机的度量结果进行校验，验证通过之后，源主机开始本机内核与 VMM 中关键数据和代码完整性的度量，将度量结果扩展到度量日志，并由源主机的 TM-Enclave 对度量日志进行签名。

5) 将签名后的度量日志、虚拟机的迁移请求 $Request_{\text{mig}}$ 和 TM-Enclave 产生的随机数 $Nonce_3$ 发送给目的主机，记为 $M_3=Key_{\text{Enclave}}(Key_{sign(S)}(Logs_2)||Request_{\text{mig}}||Nonce_3)$。

6) 目的主机成功解析 $M_3$，并校验源主机的平台完整性度量结果后，目的主机向源主机发送完整性校验成功标志 $Measure_{\text{comp}}$、生成的随机数 $Nonce_4$ 和收到的随机数 $Nonce_3$，记为 $M_4=Key_{\text{Enclave}}(Measure_{\text{comp}}||Nonce_4||Nonce_3)$。

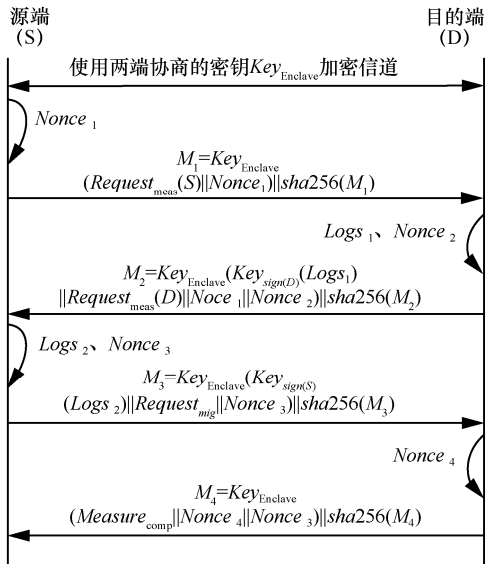



图 5 平台完整性双向验证流程

源主机接收 $M_4$，验证完整性校验成功标志之后，即可开始后续迁移的步骤。

### 4.2 安全操作与隐私数据的隔离保护

虚拟机迁移需要的文件内容主要包括 2 种：第一种是存储在磁盘的文件（虚拟机镜像、其他设备镜像、配置文件等）；另一种则是虚拟机内存、设备状态信息等。云系统通常都会采用共享存储机制，所以像虚拟机镜像、设备镜像、配置文件等不需要迁移，只要在本地利用 TM-Enclave 的加密模块加密数据，然后将密文写回共享服务器即可，如上文所述，迁移两端的 TM-Enclave 共享着加密密钥，因此目的端虚拟机可以成功读取文件。

虚拟机内存迁移由 Qemu 负责实现，通过将虚拟机的内存信息封装在一个符合 Qemu 迁移规则的结构体中，Qemu 会自动完成内存预拷贝，迭代传输内存等过程。本文利用 TM-Enclave 对迁移结构体文件进行加解密处理，除了可以保证主机上虚拟机的内存信息不会被窃取，还可以利用迁移双方的 Enclave 建立起的安全信道，保障迁移数据在传输过程中不会出现明文信息的泄露。封装并加密迁移结构体要根据需求修改与设备相关的 Qemu 代码，指定保存并加密哪些设备的信息。

如前文所述，SGX 程序分为可信部分和不可信部分。在本文中，可信部分对应 TM-Enclave，不可信部分为 Qemu 的其他部分，因为 Qemu 并非所有的代码都放在 Enclave 中执行，所以这里还存在 Qemu 的未受 Enclave 保护的部分与受保护的迁移部分代码交互的问题。可信部分与不可信部分之间的交互需要按照 SGX 规范来编写接口文件定义访问接口。不可信部分调用可信部分功能的方式叫做 ECALL；可信部分调用不可信部分功能时要通过 OCALL。

下面，以安全加密的交互过程为例，说明安全操作以及隐私数据是如何得到隔离保护的。如图 6 所示，加密模块作为可信部分运行于安全内存区域 EPC 中，Qemu 的其他部分作为不可信部分位于一般内存区域中。当 Qemu 执行运行迁移加密功能时，创建 Enclave 并调用可信函数，程序流程进入隔离的 Enclave。EPC 内存的加密相关的代码从磁盘上获取密封的密文，解密之后用该密钥对待迁移的数据进行加密处理，然后将迁移数据密文写回磁盘中，并通过 OCALL 将执行结果返回给 Qemu，等待后续的操作。迁移模型中涉及密钥以及加密操作的代码段和数据段都应该放入 SGX 提供的安全区域执行。

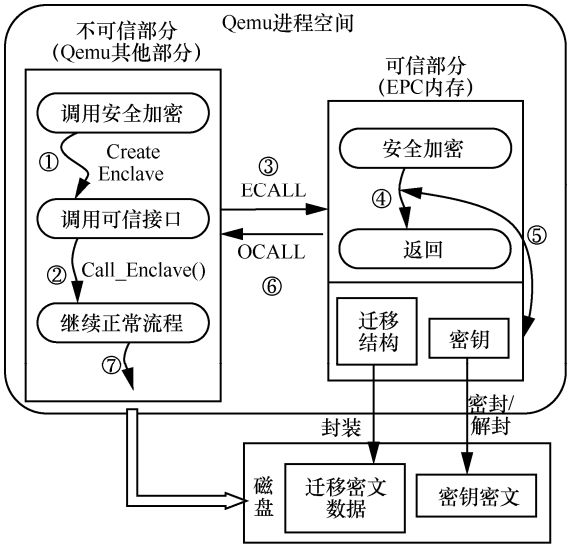



图 6 安全加密执行流程

在加密处理迁移数据的流程之后，迁移的最后阶段就是数据传输与虚拟机恢复。Qemu 将虚拟机的所有内存数据全部传输到目的主机，然后将传输过程中新产生的内存脏页迭代复制到目的主机，导入到新建的空虚拟机，直到满足虚拟机停机迁移条件为止。虚拟机停机后，将剩余内存脏页、虚拟机 vCPU 与外设状态传输到目的主机，其中，外设状态用与前面一样的 Enclave 密钥加密保护，将密文整个加密过程依然是在隔离的 Enclave 中执行。目的主机将接收到的密文信息解密，将暂停阶段迁移的数据导入到虚拟机中，至此，源主机上所有虚拟机的相关数据都传输到目的主机上的空虚拟机中，虚拟机可在目的主机恢复运行。

### 4.3　安全性分析

本文方法能够有效防止迁移信息的泄露。SGX Enclave 能够给迁移的相关代码和数据提供硬件隔离的执行环境，不依赖于固件和特权软件的安全性。即使 VMM 等特权软件出现漏洞并且被利用，受 Enclave 保护的软件也不会受到影响，攻击者也无法访问到被 SGX 保护的内存。通过 SGX 对迁移相关的加密、认证、完整校验等操作的隔离，可以在系统遭受攻击的情况下防止虚拟机隐私信息的泄露。因为涉及隐私的迁移数据通常都要加密处理，即使攻击者窃取网络中的传输信息，也无法获得虚拟机隐私数据的明文信息，而本文的加密密钥由源主机和目的主机中的安全 Enclave 保护，常规手段根本无法从中获取密钥，因此迁移数据就不存在明文在传输信道被窃取的风险。

基于迁移双方的 Enclave 的远程证明，同时双方共享一个用于加密信道的密钥，源主机和目的主机可以通过它来对两者交互的数据和迁移的数据进行加密保护。之后双方可以基于 Enclave 实现平台完整性的相互校验，以确保双方平台符合迁移的安全要求，可有效防止不可信实体参与到虚拟机动态迁移过程中。由于加密密钥以及平台完整性校验都是受 Enclave 保护的，攻击者难以伪造 Enclave 身份、篡改平台完整性信息和窃取密钥。

## 5　实验与结果分析

本节首先介绍本文方案的实现环境以及关键点，随后将对所提出的基于 SGX 的虚拟机动态迁移保护方法进行功能和性能评估。功能测试首先以实例说明虚拟机迁移过程中存在的内存泄露等问题，然后验证与分析在原有迁移框架的基础之上增加了 SGX 防护是否能够有效地防止内存信息的泄露。性能评估是为了评价迁移安全增强的方法造成的性能损耗。性能评估采用对比的方式进行，将本文的迁移方案与以往的方案进行对比，以此说明本文方案的性能代价。

### 5.1　实验环境与方案实现

为了更加接近真实的云虚拟机迁移场景，本文基于 openstack（Havana 版本）搭建了包含 3 个节点的简易云平台，其中，以 1 台主机作为控制节点，2 台支持 SGX 的笔记本（CPU 为 Intel i5 6300HQ，内存为 8 GB）作为计算节点，安装 Intel SGX SDK 1.5.80。主机和虚拟机操作系统为 Ubuntu 14.04 64 bit，计算节点还安装 libtpms 和 Qemu（Qemu 1.7.50 with PATCH）以支持 vTPM 虚拟机，2 个节点通过交换机组成局域网。

测试对象包括 KVM 架构下的普通 vTPM 虚拟机、包含原始加密方案的 vTPM 虚拟机[17]，以及本文基于 SGX 增加的 vTPM 虚拟机。采用 vTPM 虚拟机的主要原因是该类虚拟机包含虚拟的 TPM 设备，该设备可以产生密钥、证书等隐私数据供用户使用，该设备的非易失性信息（密钥、证书等）都是存储在一个镜像文件中，在迁移的时候是需要加密处理的。针对 vTPM 非易失性数据的迁移，本文在原有的加密方案[17]（以下简称 KVM 方案）的基础之上，基于 SGX 改进加密程序。由于采用了共享存储，所以只需在迁移之前由源主机将镜像文件加密并写入共享存储服务器，迁移结束阶段目的主机读取镜像文件并解密恢复。

Qemu 本身是将 vTPM 定义为一个不可迁移设备，因此为了迁移 vTPM 的易失性状态（密钥句柄、授权信息等），如表 1 所示，本文增加对这些状态的读写函数以及加解密函数，将易失性状态保存至一个符合 Qemu 迁移规则的状态保存模块中，该模块可将 vTPM 定义为一个可迁移设备，由 Qemu 来实现对其迁移的操作，可有效降低 vTPM 迁移的开销。这样 vTPM 设备就会随着虚拟机的迁移周期执行初始化、设备注册、状态保存、数据迁移以及状态恢复等操作。

参照 SGX 的编程方法[15]，根据本文需要保护的程序和对象，本文将 Qemu 程序分成可信和不可

**表 1　定义 Enclave API**

| API | 含义 |
|---|---|
| void tpm_state_read(unsigned char**buffer, uint32_t *buflen) | 读取 vTPM 易失性状态 |
| void tpm_state_write(unsigned char**buffer, uint32_t *buflen) | 写入 vTPM 易失性状态 |
| void qemu_volatilestate_enc(ObjectData *volatiledata, void *data) | 易失性状态加密 |
| void qemu_volatilestate_dec(ObjectData *volatiledata, void *data) | 易失性状态解密 |

信部分，并生成相应的接口，使程序的不可信部分与可信部分可以通过 ECALL 和 OCALL 进行互相之间的调用。本文实验主要针对 vTPM 设备对迁移组件进行了增强，所以保护的范围有限，但是通过本文方案足以说明利用 SGX 增强迁移是可行的，未来通过更加完善的设计可以增加方案的保护能力和范围，致力于达到保护整个虚拟机进程的目的。此外，SGX 提供了包含有 AES、ECC、SHA256 等常见的密码算法。本文使用 ECC 算法生成 Enclave 的密钥对，并使用 SGX 提供的随机数生成函数生成 nonce。

### 5.2 功能评估与分析

首先在没有保护机制的情况下验证虚拟机（以 vTPM 虚拟机为例）待迁移的内存信息是否存在泄露的风险。如图 7 所示，这是在 vTPM 虚拟机内部执行 tpm_getpubek 命令后的显示效果，这里显示出了 vTPM 的背书密钥（EK, endorsement key）的信息。通过导出虚拟机的内存数据，如图 8 所示，本文可以发现在地址段 0360360 开始的部分就是和虚拟机内部得到了 EK 的信息一致。在执行 tpm_getpubek 命令之前，需要首先执行 tpm_takeownership 操作，这时候需要用户自定义授权口令，这些信息都会保存在内存当中，在虚拟机迁移的通过截获未加密的迁移数据分组，就有可能恢复出这些敏感信息[10]。通过上述分析，可以得出在没有保护机制的情况下，在虚拟机执行动态迁移时，无论是当前用户输入的命令、密码还是虚拟机运行的文本文档进程和内容等都可能通过读取内存或捕获迁移的数据分组而还原出来。

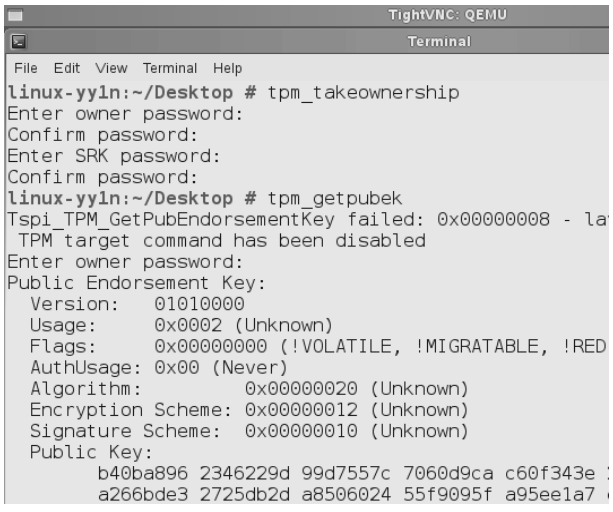



图 7 vTPM 虚拟机内查看 vTPM 密钥信息

```
0360220 bb2e e4af 8a4d b893 8516 0000 b549 65df
0360240 4e60 0000 e083 0000 664a 6c0f 0000 4234
0360260 ea0c b661 40c8 9e3e 9dc8 0000 900b e381
0360300 2a01 4f41 43e8 0000 0000 5730 d411 7bb0
0360320 565f 0000 0000 a780 005b 0000 0000 0000
0360340 0000 0003 0000 0000 0c00 0000 0000 0100
0360360 0000 0000 0000 0001 0bb4 96a8 4623 9d22
0360400 d799 7c55 6070 cad9 0fc6 3e34 6925 f02a
0360420 0d2e 7d75 1160 3380 66a2 e3bd 2527 2ddb
0360440 50a8 2460 f955 5f09 5ea9 a7e1 e9e2 612b
0360460 951c 89e0 3a16 0e29 3a8a 18c9 34f9 82aa
```

图 8 导出 vTPM 内存数据

在使用了安全增强方案后，首先本地 vTPM 隐私信息的存储都是加密，通过 od 命令查看存储 vTPM 隐私的镜像文件，加密之前如图 9 所示，从地址 0000240 开始就找到与虚拟机内对应的明文（图 7 中的 public key 信息），加密后如图 10 所示，可见 vTPM 实例数据在写入存储文件时都已经被加密处理。与原始加密方案不同的是，此处的加解密的密钥是迁移双发协商共享的，是绑定至 Enclave 的并且不需要迁移，所以无法通过本地的内存读取密钥，也不存在通信信道泄露密钥明文的问题。

```
0000000 0000 3506 0100 2200 0000 0000 0000 0000
0000020 0000 0000 0000 0000 0000 0000 0000 0000
*
0000140 0000 0000 0000 0000 0000 0000 0001 0000
0000160 0000 0001 0001 0000 0000 0000 0100 0001
0000200 0000 0011 0000 0100 0000 0100 0301 0100
0000220 0000 0c00 0000 0008 0000 0200 0000 0000
0000240 0000 0001 0bb4 96a8 4623 9d22 d799 7c55
0000260 6070 cad9 0fc6 3e34 6925 f02a 0d2e 7d75
0000300 1160 3380 66a2 e3bd 2527 2ddb 50a8 2460
```

图 9 od 命令查看未加密前的 vTPM 实例数据

```
0000140 1bcc df9a 887d 9029 39f2 c66b 3a47 c356
0000160 1389 e5b5 9a92 5c4f 66d4 dbe3 6985 aa34
0000200 87b0 a6ce 24e1 3e31 9007 2ef1 260a 9f54
0000220 3e09 d791 825d 498f 041e d706 2127 560f
0000240 05f5 e5b2 8647 2d23 1816 7e28 9abf e900
0000260 395a f956 bd06 87d4 57d2 9302 afca 955b
0000300 61be c718 6fd9 45ac f45a 4d11 f1b8 6287
```

图 10 od 命令查看加密后的 vTPM 实例数据

然后再尝试导出 vTPM 的内存数据，发现包含的本文定义的迁移数据结构字段以及相关的 vTPM 敏感信息的内存数据无法导出来，因为这些地址属于 Enclave 保护的范围，除了迁移 Enclave 关联的结构，其他系统组件都无法获取其中的信息。

以上是 SGX 对于迁移时本地内存的数据保护，在迁移过程中迁移数据也可以得到一定程度的安全防护。在迁移之前，利用绑定至安全 Enclave 的安全密钥加密待迁移的数据，整个过程都在 Enclave 中执行，外部获取不到也无法篡改这一过程。在迁移过程中，本文可以通过控制通信信道获取传输的信息流，但是由于数据被加密了，而加密密钥被 Enclave 绑定，无法获取密钥

也就无法获得迁移的明文信息。虽然恶意篡改依然可能发生，但是攻击的数据都是密文，无法针对特定数据进行特定的篡改攻击，也就达不到预期的攻击效果。

### 5.3　性能评估

首先对引入 Enclave 之后的加密程序进行性能评估。与普通虚拟机相比，vTPM 虚拟机动态迁移过程额外包含了对 vTPM 镜像文件的加密读写（保存与恢复 vTPM 的非易失性状态）以及 vTPM 易失性状态的保存与恢复。本文方案的 vTPM 镜像的读写使用了对称加密算法和 SGX 密封操作，因此本文对无加密方案、原始加密方案（简称 KVM 方案）以及本文基于 SGX 安全增加方案的 vTPM 镜像读写操作分别进行了 100 次测试并统计了平均耗时，如图 11 所示，无论是读操作还是写操作，原始方案的耗时都是最小的，相比之下，KVM 的方案和 SGX 方案都存在一定的额外的性能损耗，其中，SGX 方案略高于 KVM 方案，但是平均损耗很小，相对于虚拟机整体迁移时间来说是可以接受的。

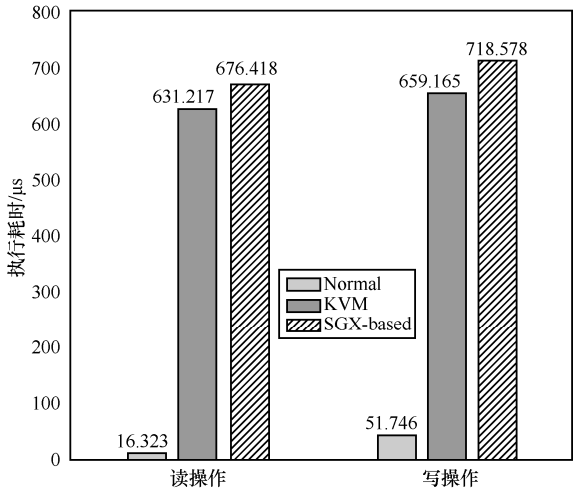



图 11　读写操作耗时对比

针对 vTPM 易失性状态的保存与恢复，分别测试表 1 的 4 个命令在普通环境下和采用 SGX 技术的情况下执行时间，每个命令重复执行 100 次，计算其平均耗时。实验结果如图 12 所示，无论是对于 vTPM 状态的读操作还是写操作，采用 SGX 保护的时候，命令执行的耗时略高于普通环境下的时间，但是增加的时间平均不超过 12 μs，几乎可以忽略不计。如图 13 所示，对于加解密操作，无论是否采用 SGX 保护，其执行时间都普遍高于读写的操作。采用 SGX 保护的时候，加解密命令执行的耗时都高于普通环境下的时间，增加的时间平局不超过 70 μs，相对于运算的整体时间来看也是很小的。

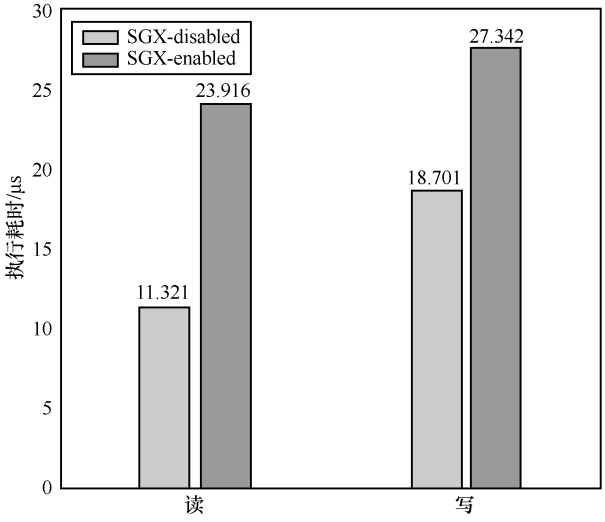



图 12　读写操作耗时对比

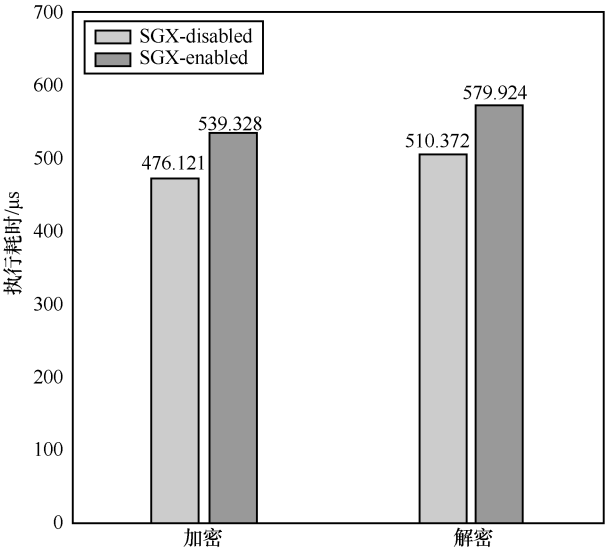



图 13　加解密操作耗时对比

最后，针对 vTPM 虚拟机统计迁移的总体时间，迁移过程主要从源主机预处理迁移数据以及迭代传输内存数据开始，到虚拟机短暂停机，再到虚拟机在目的端恢复运行。测试环境同样包括普通的迁移环境、包含原始加密方案的迁移环境以及基于 SGX 保护的迁移环境，创建 5 台相同配置的虚拟机，划分为 5 组分别上述 3 种环境下进行迁移测试，每组执行 20 次迁移操作并计算平均迁移耗时，如表 2 所示。

表 2 总体迁移时间

| 分组 | Normal/ms | KVM/ms | SGX-based/ms |
|---|---|---|---|
| 第 1 组 | 23 132 | 23 476 | 23 563 |
| 第 2 组 | 23 298 | 23 301 | 23 381 |
| 第 3 组 | 23 231 | 23 562 | 23 633 |
| 第 4 组 | 23 138 | 23 368 | 23 456 |
| 第 5 组 | 23 272 | 23 591 | 23 657 |

总体迁移时间的对比效果如图 14 所示，在普通环境下，由于 vTPM 虚拟机没有增加额外的保护，所以整体迁移时间都是最低的；增加了加密保护的迁移方案的耗时略高于无保护方案，增加的时间都在 0.5 s 以内；基于 SGX 环境的方案由于对迁移系统和加密程序进行了改进以及增加了对 vTPM 易失性信息的保护，因此其迁移时间是最长的，主要是由于程序运行过程中 SGX 环境切换以及 Enclave 内部密码学运算所引起的耗时，但是总体来看，增加的耗时相比于总迁移时间还是比较小，本文方法不会给迁移带来过多额外的性能损耗。

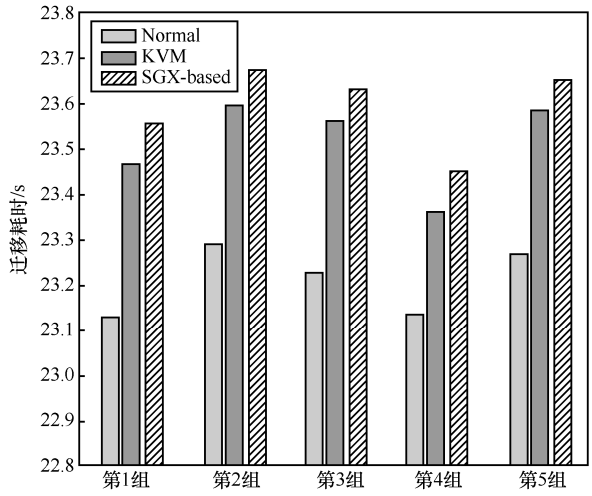



图 14 迁移总耗时对比

## 6 结束语

虚拟机动态迁移是虚拟化应用的关键技术之一，其安全性对于构建安全可靠的云平台也至关重要。本文针对虚拟机动态迁移过程中存在的内存泄露等安全问题，提出了基于 SGX 技术的迁移安全增强方法。利用 SGX 技术提供的基于硬件隔离的 Enclave，为虚拟机迁移的数据加密、平台完整性度量等安全操作提供可信的执行环境，防止迁移隐私数据的内存泄露。通过 Enclave 远程认证技术，迁移双方交互密钥种子以生成共享的密钥，该密钥用来加密迁移信道。信道加密密钥由 Enclave 保护，可以防止密钥的泄露，因此也可以阻止攻击者通过窥探信道获取迁移的隐私信息。然后通过实验分析了虚拟机迁移面临的内存泄露等安全问题并验证本文方法能够有效地解决上述问题，最后通过性能测试验证了本文提出的迁移安全增强方法不会过多的影响迁移性能。本文主要针对虚拟机特定设备的存储信息和内存进行保护，因此下一步工作将考虑利用 SGX 对整个虚拟机进程进行保护，并将 SGX 环境与云管理平台进行整合。

## 参考文献：

[1] 邹庆欣, 郝志宇, 云晓春. 基于运行阶段特征的虚拟机实时迁移技术[J]. 通信学报, 2016, 37(1):170-179.
ZOU Q X, HAO Z Y, YUN X C. Live migration based on the characteristics of operation stages for virtual machine[J]. Journal on Communications, 2016, 37(6):170-179.

[2] ZHANG H G, HAN W B, LAI X J, et al. Survey on cyberspace security[J]. Science China Information Sciences, 2015, 58(11):1-43.

[3] OBERHEIDE J, COOKE E, JAHANIAN F. Empirical exploitation of live virtual machine migration[C]//BlackHat DC convention. 2008.

[4] VAN C A, PIETERS W, WIERINGA R. Security implications of virtualization: a literature study[C]//IEEE International Conference on Computational Science and Engineering (CES). 2009:353-358.

[5] YAMUNADEVI L, ARUNA P, DEVI D S, et al. Security in virtual machine live migration for KVM[C]//International Conference on Process Automation, Control and Computing (PACC). 2011:1-6.

[6] CHEN X, GAO X, WAN H, et al. Application-transparent live migration for virtual machine on network security enhanced hypervisor[J]. China Communications, 2011, 8(3):32-42.

[7] NAGIN K, HADAS D, DUBITZKY Z, et al. Inter-cloud mobility of virtual machines[C]//International Conference on Systems and Storage. 2011:1-12.

[8] PATIL V P, PATIL G A. Migrating process and virtual machine in the cloud: load balancing and security perspectives[J]. International Journal of Advanced Computer Science & Information Technology, 2012, 1(1):11-19.

[9] BIN S N A, MASUDA H. Evaluation of a secure live migration of virtual machines using IPSEC implementation[C]//IEEE International Conference on Advanced Applied Informatics. 2014:687-693.

[10] 范伟, 孔斌, 张珠君, 等. KVM 虚拟化动态迁移技术的安全防护模型[J]. 软件学报, 2016, 27(6):1402-1416.
FAN W, KONG B, ZHANG Z J, et al. Security protection model on

live migration for KVM virtualization[J]. Journal of Software, 2016, 27(6):1402-1416.

[11] WANG W, ZHANG Y, LIN B, et al. Secured and reliable VM migration in personal cloud[C]//International Conference on Computer Engineering and Technology. 2010:705-709.

[12] ASLAM M, GEHRMANN C, BJORKMAN M. Security and trust preserving VM migrations in public clouds[C]//IEEE International Conference on Trust, Security and Privacy in Computing and Communications. 2012:869-876.

[13] ANATI I, GUERON S, JOHNSON S, et al. Innovative technology for CPU based attestation and sealing[C]//International Workshop on Hardware and Architectural Support for Security and Privacy (HASP). 2013.

[14] HOEKSTRA M, LAL R, PAPPACHAN P, et al. Using innovative instructions to create trustworthy software solutions[C]//International Workshop on Hardware and Architectural Support for Security and Privacy (HASP). 2013.

[15] Mckeen F, ALEXANDROVICH I, BERENZON A, et al. Innovative instructions and software model for isolated execution[C]// International Workshop on Hardware and Architectural Support for Security and Privacy (HASP). 2013.

[16] BERGER S, CACERES R, et al. vTPM: virtualizing the trusted platform module[C]//The 15th conference on USENIX Security Symposium. 2006: 305-320.

[17] SHI Y, ZHAO B, YU Z, et al. A security-improved scheme for virtual TPM based on KVM[J]. Wuhan University Journal of Natural Sciences, 2015, 20(6): 505-511.

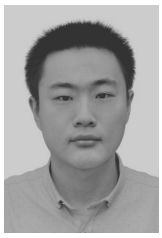
[18] FAN P R, ZHAO B, SHI Y. An improved vTPM-VM live migration protocol[J]. Wuhan University Journal of Natural Sciences, 2015, 20(6): 512-520.

**作者简介：**

**石源**（1991-），男，江西九江人，武汉大学博士生，主要研究方向为信息安全和可信计算。

**张焕国**（1945-），男，河北元氏人，武汉大学教授、博士生导师，主要研究方向为信息安全、可信计算、容错计算与计算机应用等。

**赵波**（1972-），男，山东青岛人，武汉大学教授、博士生导师，主要研究方向为信息安全、可信计算、嵌入式体系结构等。



**于钊**（1991-），男，河南郑州人，武汉大学硕士生，主要研究方向为信息安全和可信计算。