doi:10.3969/j.issn.1000-436x.2013.12.019

基于 VPE 的可信虚拟域构建机制

王丽娜[1,2]，张浩[1,2]，余荣威[1,2]，高汉军[1,2]，甘宁[1,2]
(1. 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；2. 武汉大学 计算机学院，湖北 武汉 430072)

摘 要：针对现有可信虚拟域构建方式无法满足云计算灵活配置等特性的问题，结合云计算企业内部敏感数据的防泄漏需求，提出了基于 VPE 的可信虚拟域构建方法 TVD-VPE。TVD-VPE 利用分离式设备驱动模型构建虚拟以太网 VPE，通过后端驱动截获数据分组，并进行边界安全策略检查，最后对满足策略的数据帧进行加密。同时，还设计了可信虚拟域加入/退出协议确保用户虚拟机安全加入/退出，为边界安全策略的部署设计了面向可信虚拟域的管理协议，同时为高特权用户的跨域访问设计了跨域访问协议。最后，实现了原型系统并进行了功能测试及性能测试，测试结果证明本系统可以有效地防止非法访问，同时系统对 Xen 的网络性能的影响几乎可以忽略。

关键词：虚拟以太网；边界安全策略；可信虚拟域加入协议；可信虚拟域管理协议；跨域访问协议；分离式设备驱动

中图分类号：TP309 **文献标识码：**A **文章编号：**1000-436X(2013)12-0167-11

Building mechanism of trusted virtual domain via the VPE

WANG Li-na[1,2], ZHANG Hao[1,2], YU Rong-wei[1,2], GAO Han-jun[1,2], GAN Ning[1,2]
(1. Key Laboratory of Aerospace Information and Trusted Computing (Wuhan University)，Ministry of Education, Wuhan 430072, China;
2. Computer School, Wuhan University, Wuhan 430072, China)

Abstract：Due to lack of flexible networking control, most exiting trusted virtual domain deployment approaches fail to provide elastic and secure interconnection. A trusted virtual domain architecture TVD-VPE was proposed in cloud computing enterprises which greatly enhances sensitive data protection. TVD-VPE constructs a virtual private ethernet based on separate device driver, VPE captures network packets at the backend driver and checks whether the packets comply with border security strategy, and data frames are encrypted among trusted virtual domains to ensure the security of sensitive data. Simultaneously, four protocols were proposed, TVDJOP/TVDEXP protocol for any new VM joining in or exiting TVD securely, TVDMP protocol for deploying border y strategy, and Inter-TVD protocol for authorizing cross-domain access. Finally, the prototype system and tests of its functionality and performance were implemented. The experiment results reveal that the architecture can effectively prevent unauthorized access between these trusted virtual domains, while introduces little overhead to Xen network performance.

Key words: virtual private ethernet; border security strategy; TVD join protocol; TVD management protocol; inter-TVD access protocol; separate device driver

1 引言

云计算是信息技术领域的一次变革，也是互联网技术深度应用的必然趋势，已在诸多领域得到广泛应用[1]。在云计算模式下，虚拟机之间通过虚拟网桥或虚拟交换机连接，任意 2 个虚拟机之间可以通信，这种模式为虚拟机之间的网络安全通信带来了巨大挑战[2]，一方面会造成敏感数据的外泄，如未授权的虚拟机访问了有安全密级的数据；另一方面会影响虚拟机本身的隔离性[3]，比如一台虚拟机

收稿日期：2013-01-04；**修回日期：**2013-10-17
基金项目：国家自然科学基金资助项目(61373169, 61103219, 61303213)；教育部博士点基金优先发展领域基金资助项目(20110141130006)
Foundation Items: The National Natural Science Foundation of China (61373169, 61103219, 61303213); The Ph.D. Programs Foundation of Ministy of Education of China (20110141130006)

感染了木马，很可能通过网络通信而传播到其他虚拟机中。因此，对云平台上的虚拟机进行安全级别的划分以形成不同安全级别的虚拟网络，并制定安全策略防止不同安全级别虚拟域之间的非法访问，形成可信虚拟域 TVD[4]（trusted virtual domain），是解决云环境中敏感数据泄露问题的重要手段。

可信虚拟域是分布在不同物理平台上并且实施相同安全策略的虚拟资源的集合[4,5]。目前，构建可信虚拟域的研究主要分为：硬件虚拟网络隔离[6~10]和软件虚拟网络隔离[11~14]。硬件虚拟网络隔离是通过在硬件交换机中加入相应的网络隔离策略（如 VLAN 协议），当数据分组经过交换机进行转发时根据安全策略判断能否通过。Catuogno[15]等利用可信计算技术确保了虚拟域内部成员的安全性，在虚拟机通信过程中利用 VLAN 整合 IPSec 的交换机将不同虚拟域的数据分组分割到不同物理通道上，并利用 IPSec 确保通信的安全。然而硬件虚拟网络隔离需要交换机的硬件支持，交换机不能完全了解主机内部的情况，同时交换机的配置受制于网络拓扑结构，因此目前主要采用软件虚拟网络隔离的方式[13,14]。

由于传统的虚拟防火墙受制于网络拓扑结构，无法控制同一个主机上不同虚拟域的虚拟机网络隔离。为了解决这些问题，大量学者通过在宿主机的特权域 Dom0 中提供一个虚拟网络层来管理宿主机上的 VM(virtual machine)之间数据分组的转发[14,16,17]。Abhinav[16]利用 Xen[18]虚拟机监视器内省机制实现了软件防火墙 VMwall，通过截获数据分组并对数据分组进行分析，防止恶意程序的攻击实现网络隔离，然而其位于 Dom0 应用层软件，影响了系统效率。Renzo 提出了一个软件实现的虚拟交换机 VDE，该系统能够实现转发、路由以太帧，然而 VDE 并不支持管理功能[19]。Fabienne 提出了一个基于 Xen 的虚拟路由器，该虚拟路由器是通过软件实现，并且能够通过简单的转发策略确保虚拟机之间安全访问，然而，虚拟路由器却带来很大的处理开销和性能损失[20]。Ben Pfaff 提出了一种基于 Xen 的虚拟交换机:open vSwitch,open vSwitch 可以对转发进行粗粒度的访问控制来构建虚拟网络，同时实现虚拟域的动态管理，然而 open vSwitch 并不能确保虚拟域内部成员的安全以及通信过程的安全[21]。

现有可信虚拟域构建方式严重依赖路由器对 VLAN 及 IPSec 协议的硬件支持并受限于网络拓扑结构，并且不能灵活地根据虚拟机的用户属性构建可信虚拟域。本文针对上述的问题，提出了基于 VPE(virtual private ethernet)的可信虚拟域构建方法 TVD-VPE，并提供了相关安全策略(可信虚拟域的加入/退出协议，身份认证机制、跨域访问协议等)将可信虚拟域与用户绑定，为可信虚拟域的构建以及安全策略的部署提供了灵活的配置途径；为了实现上述方法，本文利用 Xen 分离式设备驱动模型，实现了虚拟网络 VPE，通过 VPE 截获网络数据分组并进行安全策略检测，构建可信虚拟域，确保虚拟机之间可控互联；最后对传输的数据分组进行加密，确保通信安全，为虚拟域用户提供灵活可靠的边界安全控制。这样确保高安全级别的敏感数据不会泄露到低安全级别虚拟域中，阻止不同安全级别虚拟域之间的非法访问，增强企业内部敏感数据的机密性、安全性。

2 基于 VPE 的可信虚拟域架构

为了构建基于 VPE 的可信虚拟域，本文利用分离式设备驱动模型构建虚拟网络 VPE，在特权域 Dom0 的网络后端驱动截获客户虚拟机 DomU 的数据分组并进行安全策略验证和 RBAC（role-based access control）访问控制，系统通过对使用 VM 的用户进行访问控制，实现对虚拟机数据分组的控制，同时对待发送的数据帧进行加密从而确保虚拟机通信安全，以确保不同安全级别虚拟域可信互联。

2.1 安全假设

本文考虑的主要防护对象是企业内部的恶意虚拟机或内部人员，以及网络中的窃听者。假设 TVDServer 与所有的物理机之间实现已经完成密钥协商，即 TVDProxy 与 TVDServer 共享密钥。

假设可信虚拟域中运行的虚拟平台均为可信的。本文通过可信计算基度量虚拟平台的完整性以确保其可信。TCB（trusted computing base）代理部署在各个物理平台的 Dom0 中，在虚拟机启动过程中，虚拟机创建器（domain builder）会对虚拟机内核进行完整性度量，并将度量值扩展至 vTPM（virtual trusted platform module）实例的 9 号 vPCR 中，确保虚拟机的完整性[22]。

2.2 基于 VPE 的可信虚拟域的设计目标

为了确保虚拟域的可信，制定相关的安全策略。首先介绍基本概念：虚拟机的集合定义为 VMS，可信虚拟域的集合定义为 $Realms$，用户的集合定义

为 *Users*。*Useri* 登录的虚拟机为 *VMi*，对应的角色为 *Rolei*，对应的可信虚拟域为 *Realmj*，*address* 为加入/退出可信虚拟域过程中的处理方式 *address*={*accept, reject*}，*handle* 为通信过程中的处理方式 *handle*={*pass, drop*}。具体安全策略如下所示。

1) 虚拟机加入/退出策略确保只有虚拟平台完整且经过认证的虚拟机才能加入到可信虚拟域中，防止企业内部的恶意攻击；虚拟机退出时，销毁认证凭证。加入可信虚拟域形式化表示为

$$integ(VMi) \wedge auth(Useri, Realmj) = accept$$
$$Useri \in Users\ VMi \in VMs\ Realmj \in Realms$$

其中，$integ(VMi)$：*VMi* 平台完整；$auth(Useri, Realmj)$：*Useri* 属于可信虚拟域 *Realmj*。

2) 虚拟机的访问策略确保可信虚拟域内访问以及跨域访问的合法性。虚拟机进行域内访问时，根据本地访问控制策略控制数据的转发；虚拟机进行跨域访问时，利用 RBAC 访控机制来确保访问合法。其中域内访问策略的形式化表示为

$$ACL(VMi, VMj) = pass \quad VMi, VMj \in VMs$$

跨域访问的策略表示为

$$\overline{ACL(VMi, VMj)} \wedge ACC(Rolei, T\arg etRealmj) = pass$$
$$VMi, VMj \in VMs\ Rolei \in Roles\ T\arg etRealmj \in Realms$$

$ACL(VMi, VMj)$：*VMi* 与 *VMj* 属于同一个虚拟域，$ACC(Rolei, T\arg etRealmj)$：角色 *Rolei* 能够访问目标虚拟域 $T\arg etRealmj$。

3) 网络通信安全策略实现数据帧的加密，确保物理网络平台之间通信信道的安全。

2.3　基于 VPE 的可信虚拟域架构

为了构建基于 VPE 的可信虚拟域，本文在物理平台部署 TPM（trusted platform module）模块，通过 TPM 模块度量虚拟机的完整性，在此基础之上，本文修改了网络后端驱动并部署访问控制策略确保 VM 之间可控通信及跨域访问的合法性，管理策略同步，以及虚拟机安全加入/退出虚拟域，从而增强企业内部敏感数据的机密性、安全性。

基于 VPE 的可信虚拟域系统位于特权管理域 Dom0 中，主要包括两大主要组件：可信虚拟域管理功能模块、数据逻辑控制转发模块。其中可信虚拟域管理功能模块由 TVD 管理服务器（TVDServer）和每台物理机上部署的 TVD 代理（TVDProxy）模块组成，前者存储、管理 TVD 策略，后者同步并实施 TVD 策略以管理虚拟机(如虚拟机的加入/退出，转发策略更新)。数据逻辑控制转发模块包括控制模块和数据封装转发模块。控制模块负责数据分组的地址解析，并利用 RBAC 机制控制数据分组转发及跨域访问；数据封装转发模块对 DomU 发送的普通数据分组进行封装，确保只有虚拟域内部的分组才能正常接收和发送，具体架构如图 1 所示。

1) TVD 代理模块负责可信虚拟域策略的部署与更新操作，以及本地虚拟机加入/退出虚拟域及跨域访问等方面功能。a)当虚拟机需要登录/退出可信虚拟域时，TVD 代理模块利用加入/退出可信虚拟域协议向 TVD 管理服务器发送加入请求/退出请求，通过身份认证/信息销毁完成虚拟域加入与退

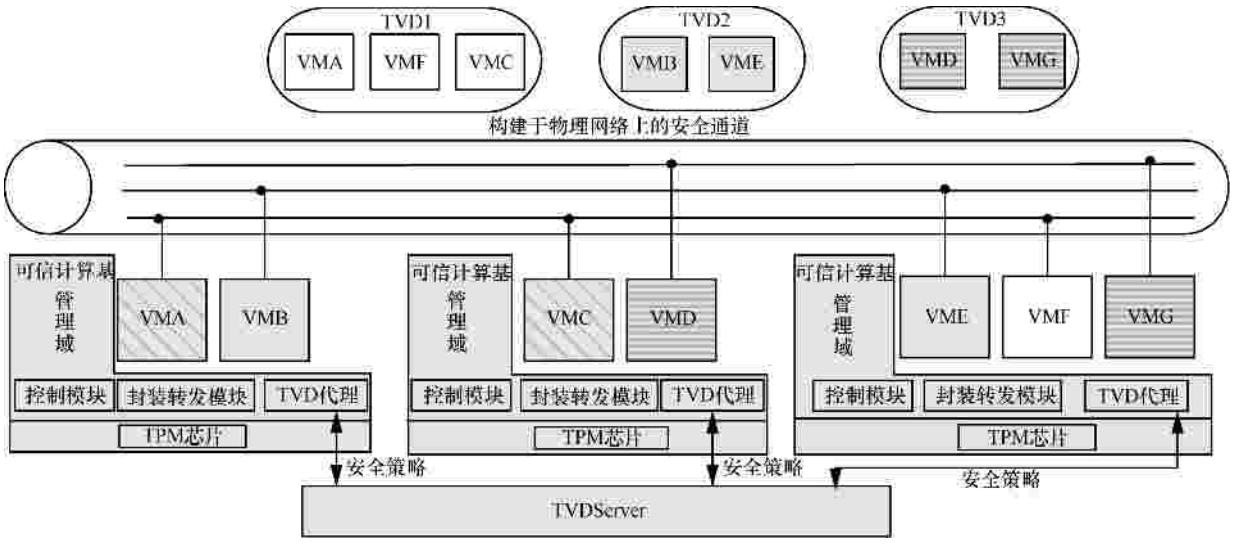

图 1　基于 VPE 的可信虚拟域的逻辑视图

出。b)在数据发送过程中，负责根据用户的角色查询 TVD 服务端的跨域 ACL 表，并对用户实现 Kerberos 跨域认证。c)当虚拟域策略发生改变时，TVD 代理模块负责接收 TVDServer 事件通知，并同步本地转发状态表与全局转发状态表。在远程管理服务器端上获取与本地主机上后端设备相关的虚拟机的信息，同时将全局状态转发表转化成本地状态转发表。

2) 控制模块主要负责基于 RBAC 的边界安全访问控制策略的实施。控制模块通过截断虚拟网络原有的网络通道，在数据帧被放入内核协议栈之前，对其进行 RBAC 访问策略验证，控制其转发状态。在网络数据发送端，当控制模块接收到网络后端驱动递交的数据后，获取相关地址及虚拟机对应的用户的角色信息，并根据验证该数据帧是否符合边界安全策略。在网络数据接收端，控制模块接收数据封装转发模块递交的网络数据帧，并对获取的数据帧进行 MAC 地址解析，并查找出对应的后端设备，最后通过网络后端把数据发送给对应的虚拟机。详细流程如图 2 所示。

3) 数据封装转发模块主要负责网络数据分组的加密与解密以及网络数据的封装与解封。在发送端，该模块接收通过策略安全验证后的数据帧后，加密数据帧，并将加密后数据二次封装成 UDP 数据分组。由于 UDP 协议是面向无连接的协议，相比 TCP 协议，更加简单、网络负载较小，因而更适合数据分组的传输。同时，为了提高系统通信效率，本文在内核层进行 UDP 数据分组的封装与发送。在接收端，本文创建网络数据接收的内核线程，该线程创建内核态的 SOCKET 并将自己阻塞以等待网络数据。当有网络数据到来时，将会触发数据接收软中断，该软中断将唤醒网络数据接收线程，该内核线程通过 SOCKET 接收网络 UDP 数据分组，同时解封该数据分组，获取数据分组中封装的数据帧，同时对数据帧进行解密，并将该数据帧提交给控制模块进行地址解析和转发判断处理。

2.4 关键技术挑战

为了实现基于 VPE 的可信虚拟域的数据截获与发送、安全加入/退出、策略的部署更新、跨域访问等相关功能，本文需要设计相关的协议和模型，关键技术如下。

1) 利用 Xen 的分离式设备模型实现可控的虚拟网络 VPE。在 Xen 架构下，利用分离式设备驱动模型，截获 DomU 发送的数据帧，并进行安全策略的验证。

2) 协议的设计与实现。协议包括面向可信虚拟域的管理协议 TVDMP(TVD management protocol)、可信虚拟域加入协议 TVDJOP (TVD join protocol)、可信虚拟域退出协议 TVDEXP (TVD exit protocol)，可信虚拟域跨域访问协议 Inter-TVD (Inter-TVD access protocol)。

3 TVD 协议设计

本节具体介绍上节涉及的相关协议：TVDJOP 协议、TVDEXP 协议、Inter-TVD 协议、TVDMP 协议。首先，给出协议基本的符号定义，如表 1 所示。

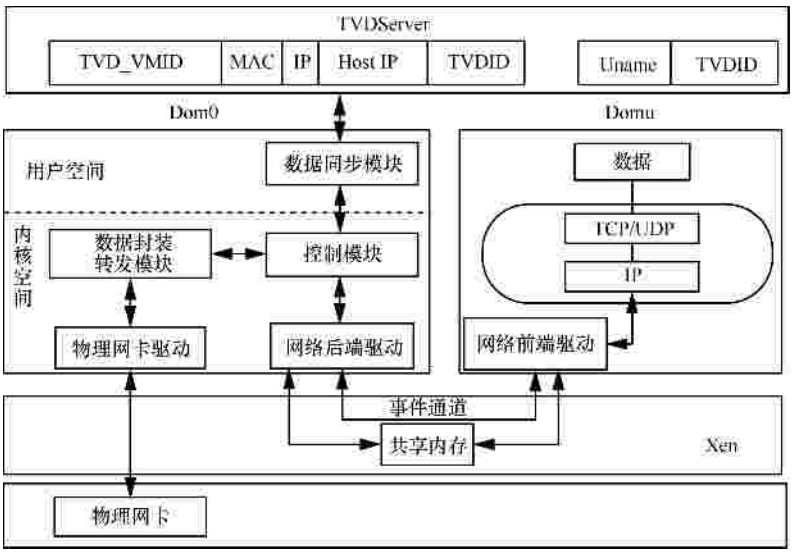

图 2 基于 VPE 的可信虚拟域系统架构

表 1　TVD 协议基本概念

符号表示	符号含义
ID(Client)	用户 ID
EK	背书密钥
Realm	用户所在可信虚拟域
AIK	身份证明密钥
AIKpub	身份证明的公钥
AIKpriv	身份证明的私钥
Integrity list	虚拟平台完整性列表
MAC	VMC 的物理地址
IP	VMC 的网络地址
VIF	VMC 的虚拟网络接口 ID
HostIP	VMC 的宿主机 IP
Role	用户的角色
TGS	TGT 票据服务器的 ID
$K_{c,tgs}$	Client 与 TGS 之间会话密钥
Lifetime	TGT 票据生存周期
TVDID	VMC 所在可信虚拟域
ExitSuccess	VMC 安全退出标志
K_h	宿主机与 TVDServer 共享的密钥
TargetIP	目标 VM 网络地址
TargetTVDID	目标 VM 所在可信虚拟域

3.1　TVDJOP 协议

TVDJOP 协议确保只有度量信息完整且经过身份认证的虚拟机才能加入到虚拟域中。协议的具体的设计如图 3 所示。

在 TVDJOP 协议中，参与的实体包括虚拟机中的认证客户端 Client、TVDProxy、TVDServer。Client 所在的虚拟机 VMC 在加入可信虚拟域之前，Client 需要向 TVDServer 注册其详细的身份信息，并加入到对应的可信域中。VMC 加入可信虚拟域的协议描述如下，其中利用远程证明协议[22]验证虚拟平台完整性。

1) Client 将 *TVD* 加入请求 *TVD_ADD_REQ* 传送给 TVDProxy，其中 *TVD_ADD_REQ*={ID(Client)||*Password*||*Realm*}。

2) TVDProxy 接受 *TVD_ADD_REQ*，并向 VMC 发送平台完整性挑战 *nonce*。

3) VMC 平台获取度量列表和 vTPM 内的 vPCR 值签名信息 $y=\mathrm{Sig}_{\mathrm{AIKpriv}}(\mathrm{vPCR}||\mathrm{nonce})$，并将 {*y*||AIKpub||Integrity list||$Cert_{EK}$(*AIKpub*)} 发送给 TVDProxy。

4) TVDProxy 利用 $Cert_{EK}$(*AIKpub*) 验证 *AIK* 的公钥 *AIKpub*，然后验证 ver_{AIKpub}(*vPCR*||*nonce*，*y*)=*True* 是否成立。如果成立，利用 vPCR 验证 Integrity list，否则“拒绝”。TVDProxy 计算

TVD_VMID = *Hash*(*vPCR*||*MAC*||*IP*||*Host_IP*)

利用 *Password* 生成 Master Key K_c(放入 TPM 中)，并生成 *KRB_AS_REQ*={*ID*(*Client*), *Authenticator*}，*Authenticator*=E_{Kc}{*TimeStamp*||*TGS*||*TVD_VMID*||*vPCR*||*MAC*||*IP*||*HostIP*||*VIF*}并传送 *KRB_AS_REQ* 给 TVDServer。

5) TVDServer 利用 Client 的密钥 K_c 验证 $\mathrm{ver}_{\mathrm{Kc}}$

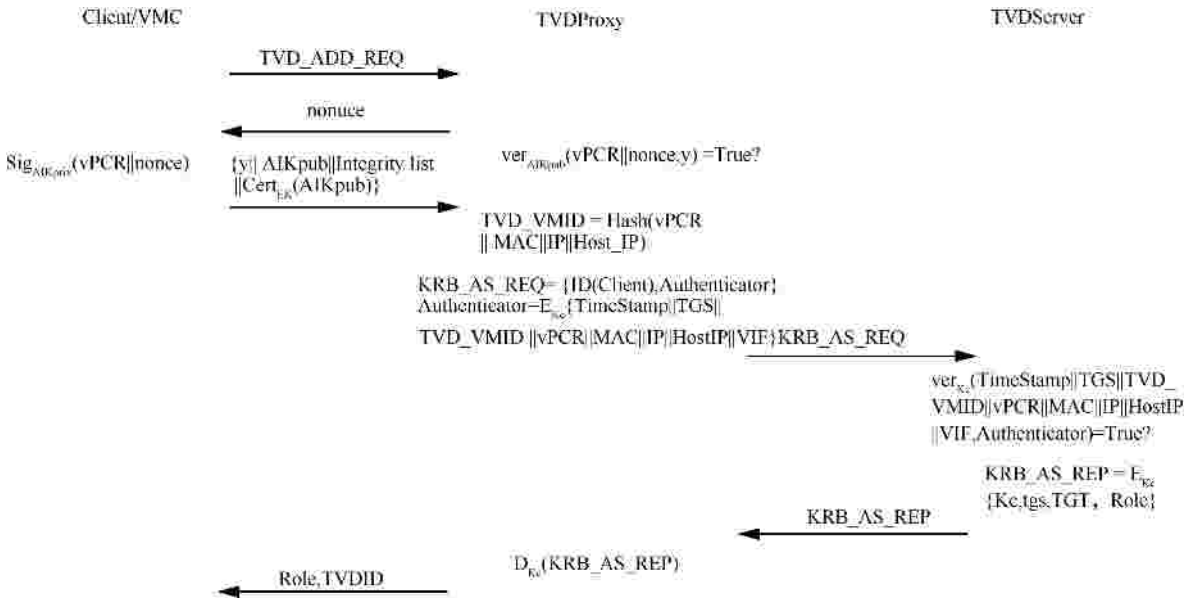

图 3　TVDJOP 协议

(TimeStamp‖TGS‖TVD_VMID‖vPCR‖MAC‖IP‖HostIP ‖VIF,Authenticator)=True 是否成立。如果不满足，则拒绝；否则，生成 $KRB\_AS\_REP = E_{Kc}\{K_{c,tgs}, TGT, Role\}$，$TGT = E_{Ktgs}\{TGS, ID(Client), TimeStamp, Lifetime, K_{c,tgs}\}$。注册 VMC 基本信息（若已有 VMC 信息则修改，其中 *TVDID* 修改为 *Realm*），并把 *KRB_AS_REP* 发送给 TVDProxy。

6） TVDProxy 利用 Kc 解密 KRB_AS_REP，得到 Role，TGT，并将其与 VMC 的 MAC、TVD_VMID 地址进行对应存储。最后，TVDProxy 发送 Role，TVDID 信息给 Client。

3.2 TVDEXP 协议

TVDEXP 协议确保虚拟机退出虚拟域之后，虚拟机对应的相关信息都要销毁。协议的具体设计如下所示。

1) Client 发送 TVD_EXIT_REQ，其中，TVD_EXIT_REQ={ID(Client),MAC,TimeStamp1}。

2) TVDProxy 验证 *TimeStamp*1，然后向 TVDServer 发送 *TVDS_EXIT_REQ*，*TVDS_EXIT_REQ* = {*TVD_VMID, TimeStamp*2}。

3) TVDServer 根据 *TVD_VMID* 值获得 VMC 信息，并将其 *TVDID*、*VIF* 字段清空，然后，TVDServer 将 *TVDS_EXIT_REP*={*ExitSuccess*，*TimeStamp*3}返回给 TVDProxy。

4) TVDProxy 销毁 Client 的 *TVD_VMID*、*TGT* 票据、*Role* 及 *Realm* 等信息，同时向 Client 发送 *TVD_EXIT_REP*={*ExitSuccess*，*TimeStamp*4}。

3.3 Inter-TVD 协议

Inter-TVD 协议确保数据传输过程中高权限用户跨域访问的合法性。协议的具体描述如下。

1) TVDProxy 确定当前时间 *TimeStamp*，并向 TVDServer 发送用户角色信息

$y_1 = E_{Kh}\{TVD\_VMID, Role, TargetIP, TimeStamp\}$

2) TVDServer 利用 K_h 解密 y，得到 *Role*，*TargetIP*，然后，TVDServer 根据 *TargetIP* 确定 *TargetTVDID*，并根据 ACL 表确认 *Role* 能否访问 *TargetTVDID*，如果满足条件，则在 *TVD_VMID* 对应的 *TVDID* 字段加入值 *TargetTVDID*，同时计算

$y_2 = E_{Kh}\{True, TimeStamp, TargetTVDID\}$

否则拒绝。最后将 y_2 发送给 *TVDProxy*。

3) TVDProxy 解密 y_2 获得 *TimeStamp*，*True*，*TargetTVDID* 并对时间进行验证，将 *TargetTVDID* 加入到 *TVD_VMID* 对应的缓存中。

3.4 TVDMP 协议

在执行完 TVDJOP、TVDEXP 或 Inter-TVD 协议后，TVDMP 协议负责及时同步全局状态转发表与本地状态转发表。

1) TVDServer 向 *TVDProxy* 发送同步通知 *TVD_SYN _INF*={*TVDID*，*TimeStamp*1}，其中 TVDID 代表需要更新的虚拟域。

2) TVDProxy 向 TVDServer 发送同步请求 *TVD_SYN_REQ*={*HOSTIP*,*TVDID*,*Timestamp*2}。

3) TVDServer 验证 HOSTIP 是否包含 TVDID 域的虚拟机，如果成立，将所有 TVDID 域内的 VM 信息返回给 TVDProxy，返回的信息为

$TVD\_SYN\_REQ = E_{Kh}\{MAC, IP, HOSTIP, TVDID, TimeStamp\}$

其中，K_h 为宿主机与 TVDServer 共享的密钥。

4) TVDProxy 利用 K_h 解密 TVD_SYN_REQ 获得数据，并转换成本地状态转发表。

4 基于 VPE 的可信虚拟域系统实现

通过上述的协议实现了虚拟机的可信接入与退出，在此基础之上，本系统利用 TVDMP 协议实现数据和安全策略的同步。同时，通过 Dom0 后端驱动截获虚拟机的网络数据帧，以进行边界安全策略验证，并将通过验证的数据帧进行 UDP 二次封装转发，利用加密机制确保通信过程中的安全传输，以达到构建可信虚拟域确保敏感数据安全的目的。

4.1 TVD 策略的部署及同步

TVD 策略的部署及管理由 TVDServer 和每台物理机上部署的 TVDProxy 共同完成，前者存储、管理 TVD 策略，后者同步并实施 TVD 策略以管理虚拟机（如虚拟机的加入/退出,转发策略更新），具体交互过程如图 4 所示。

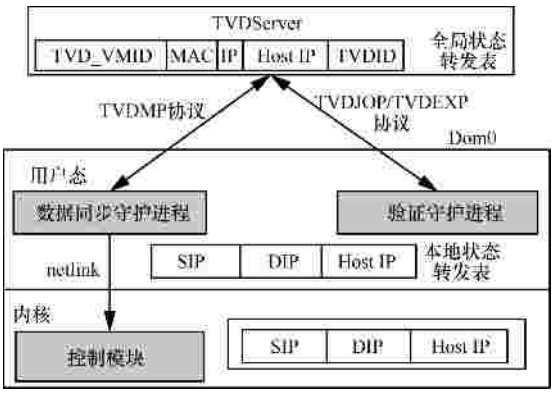

图 4 TVD 可信策略部署与管理

TVD 管理服务器部署在远端服务器上，主要包含 Kerberos 认证模块、LDAP 目录服务器及对应的管理模块守护进程。其中 Kerberos 认证模块负责监听 TVD 代理发送的加入虚拟域请求，管理模块守护进程负责监听虚拟机或用户信息的增删改查请求，LDAP 目录服务器根据虚拟域的信息组织存放用户信息及虚拟机的基础信息。

TVD 代理模块部署在本地宿主机上，主要包含数据同步守护进程和验证守护进程两部分。验证守护进程主要监听内部虚拟机的加入/退出虚拟域的请求，并验证虚拟平台的完整性，如果虚拟平台完整则向服务端发送加入/退出请求，并根据相应的 TVDJOP 及 TVDEXP 协议将对应的数据结构进行存储(带有特殊标志比特 Role，TGT，MAC、TVD_VMID)或销毁，否则，发送拒绝虚拟机加入虚拟域请求。数据同步守护进程监听 TVDServer 的同步通知，并通过 TVDMP 协议对可信虚拟域进行管理以及同步全局状态转发表与本地状态转发表。同步模块主要管理本地的状态转发表数据，当守护进程获取到 TVDServer 更新事件之后，同步模块从远程管理服务器上获取与本机相关的信息，并将获取的信息转换成本地状态表格式，并通过 netlink 将本地的状态转发表传递给内核态控制模块。

4.2　数据封装与发送

基于 VPE 的可信虚拟域系统利用 Xen 架构下网络设备的前后端驱动机制在 Dom0 的后端截获 DomU 发出的数据分组，并进行边界安全策略的验证，同时对经过策略验证的数据帧进行加密（AES-128 加密算法）处理，最后将加密后的数据封装为 UDP 分组，传递给物理网卡驱动进行数据发送。同时，为了提高系统的网络吞吐率，本文在内核中实现客户虚拟机数据帧的封装和发送。通过修改后端驱动的 net_tx_action 函数，添加基于 RBAC 的边界安全策略（域内访问控制策略及跨域访问策略）、数据帧加密机制及 UDP 数据分组的封装转发等功能，以实现虚拟机间网络通信的可控发送，具体算法如图 5 所示。

1) 截获 DomU 的网络数据帧并线性化。当数据帧到达后端驱动中，由于缓冲区大小的限制，该数据帧可能分布在多个分散缓冲区中。为了将完整的数据帧封装在 UDP 分组中，本文对数据帧进行线性化，将分散的数据缓冲区进行聚合操作。

```
函数原型：net_tx_action (unsigned long unused)
1)  get_data(skb);
2)  retval = skb_linearize(skb);
3)  ipaddr = search_vpe_table (skb)
4)  if ipaddr ==NULL then
5)          role = get_role(srcmac)
6)          retacl = search_targetdomin(role);
7)              if retacl != NULL then
8)                  encryptSkb(send_msg)
9)                  sendUDPdata
10)             else
11)                 return notFound;
12)             end
13) else
14)         encryptSkb(send_msg)
15)         sendUDPdata
16) end
```

图 5　基于 VPE 的可信虚拟域数据发送算法

2) 对 DomU 发出的数据分组进行转发判断。首先，解析数据帧获取数据帧的目的 IP、MAC 地址和源 IP、MAC 地址，并根据源 MAC 地址查询登录时存储在 Dom0 中用户角色信息。然后通过在源 IP 和目的 IP 对应的转发项，如果有对应的记录则返回相应的目的主机 IP 地址，否则利用 search_targetdomin 查看用户的角色是否包含特殊标志比特（标识该角色是否为跨域角色），以及对应的跨域角色能否访问目的虚拟域；如果满足跨域访问则返回目的主机的 IP 地址，否则返回空字符串。

3) 内核对数据分组进行 UDP 封装。在策略安全验证之后，创建内核态的 UDP 类型的 socket，通过目的 MAC 地址查询出其所在主机的 IP 地址，并将该地址赋值给 addr.sin_addr.s_addr。然后，将阶段 1）中经过线性化处理的数据帧拷贝到自定义的缓存区中，利用 AES-128 加密机制对缓冲区中数据进行加密，并将加密数据帧封装为 UDP 分组。最后调用内核态的数据发送函数 kernel_sendmsg 发送二次封装后的 UDP 数据分组。

4.3　数据接收

数据的接收模块以内核守护线程的形式存在，该线程将自己阻塞以等待网络数据，当有网络数据到来时，将会触发数据接收软中断 NET_RX_SOFTRQ，后者将唤醒网络数据接收线程，以将收到的网络数据转发给对应的虚拟机。数据接收守护线程的处理算法如图 6 所示。

```
函数：recv_from_kernel_thread(void *unused)
1)  while  exit_flag == 0  do
2)    prepare_to_wait(&sock->wait, &my_wait, TASK_ INTER-
RUPTIBLE);
3)    schedule();
4)    finish_wait(&sock->wait, &my_wait);
5)
while  !skb_queue_empty(&(sock->sk->sk_receive_queue)) do
6)        ret = kernel_recvmsg(sock, &msg, &vec, 1,
MAX_PAYLOAD, msg.msg_flags);
7)        if  ret > 0 then
8)          decrypt(vec.iov_base);
9)        if  (netdev_name=search_vif_by_mac(vec.iov_base) )
                                        == NULL then
10)           continue;
11)       end
12)         Send_msg_to_client(netdev_name);
13)       end
14)   end
15)   end
```

图 6 网络数据接收线程

基于 VPE 的可信虚拟域在初始化时创建接收端 SOCKET，并通过 inet_bind 将本地的地址和端口进行绑定，同时将该 SOCKET 地址传给网络驱动模块，最后创建网络数据接收守护线程，并将接收端 SOCKET 作为参数传递给该守护线程。

接收端守护线程在初始时阻塞自己等待网络数据。当有数据分组到来时，触发网络数据接收软中断 NET_RX_SOFTRQ，该中断通过内核函数 schedule 唤醒接收端守护线程，进而通过 kernel_recvmsg 接收 sock->sk->sk_receive_queue 队列中数据。对接收的 UDP 包进行解密获取原始的数据帧，将解密后的数据帧放入预先分配的 sk_buff 结构体中，并根据目的 MAC 查找出对应目的 VIF 对应的 netdevice，然后通过 dev_queue_ xmit 将数据帧发送给对应的虚拟机，完成数据接收及转发任务后接收线程阻塞自己，等待再次被唤醒来接收数据。

本系统修改软中断 NET_RX_SOFTRQ 的处理函数 net_rx_action，在该处理函数中添加唤醒网络数据接收线程的相关操作，以便及时接收网络数据分组。具体流程如图 7 所示。

```
函数原型：net_rx_action(struct soft_ation *h )
添加的代码如下：
if  my_sock_kern != NULL then
    wake_up_interruptible(&my_sock_kern->wait);
end
注释：my_socket_kern 是模块初始化时传递给网络驱动的内核态 socket
```

图 7 接收线程唤醒机制

4.4 系统安全性分析

下面对基于 VPE 的可信虚拟域构建系统的安全性简要分析。

重放攻击（RA）：系统可以抵抗重放攻击。在 TVDJOP 协议中，VM 登录时发送的登录请求为 KRB_AS_REQ ={ID(Client),Authenticator}，Authenticator = E_{Kc}{TimeStamp||TGS||TVD_VMID||vPCR||MAC||IP||HostIP||VIF}，因此，网络攻击者必须解密的数据分组并修改时间戳才能冒名认证成功，然而攻击者不可能解密 Authenticator。Inter-TVD 协议与 TVDMP 协议同样也能够有效地抵抗重放攻击。

网络窃听与篡改：系统可以抵抗网络窃听与篡改。在 VM 登录的过程中引入 TGT 票据，对会话密钥 $K_{c,tgs}$ 进行加密，避免口令的明文传输。TVDJOP 协议中对 Client 的信息进行加密，防止了攻击者伪造合法用户的网络地址。

5 实验设计与分析

为了验证基于 VPE 虚拟域的功能及可信虚拟网络的性能指标，本文进行了可信虚拟域划分功能测试及网络性能测试。

5.1 实验环境

整个实验在 OpenNebula 搭建的云平台环境中进行，云平台中有 6 台物理机和 12 台虚拟机，形成 3 个可信虚拟域，底层使用版本 3.3.0 的 Xen 虚拟监视器。在云平台中，每台物理机的操作系统是 Ubuntu-8.04-server，自带 TPM 模块和一块百/千兆网卡，主机之间通过百兆交换机进行交互，每台主机上有 3 个 Windows XP 的虚拟机，虚拟机中加载 VTPM 模块。TVDServer 可信虚拟域管理服务器为一台单独的服务器，操作系统为 redhatAS 5.5 版本，利用 MIT 的 krb5-1.5.4 版本，并在此版本上进行相关的字段扩展，加入对应的用户角色信息和权限信息，满足虚拟机加入协议的需求，最后将其与 openldap 进行集成部署。

5.2 实验场景

在上述的实验环境下，进行相关的功能测试和网络的性能测试。测试环境中只列出企业用户、角色、虚拟机的主要信息。表 2 中 RoleID 代表角色编号，RoleName 表示角色的名称，TVDID 表示角色所属的可信虚拟域，Access 表示用户能够访问的可信虚拟域的编号。

表 2　角色访问控制

RoleID	RoleName	TVDID	Access
1	普通职员	2	2
2	普通职员	3	3
3	管理员	2	2,3

表 3 中列出了用户的基本信息，Uname 表示用户名，RoleID 表示用户所属的角色编号，TVDID 表示用户所属的用户可信虚拟域。

表 3　用户基本信息

Uname	RoleID	TVDID
User1	1	2
User2	2	3
User3	1	2
User4	3	2

4 台虚拟机及相关的宿主机的信息如表 4 所示(TVDID 信息由登录用户决定)：其中 4 台虚拟机的 MAC 的地址前 40 bit 一致（00:25:11:12:3f:*），只有最后 8 bit 不一致；宿主机与虚拟机均在同一网段（192.168.1.*）。在表 4 中只列出虚拟机 MAC 地址、IP 地址的最后一比特。

实验中用户 User1、User2、Use3、User4 分别从虚拟机 1、2、3、4 上向对应的 TVD 代理发送登录虚拟域请求，经过 Kerberos 认证对应的虚拟机分别加入到用户所在的可信虚拟域中。

表 4　可信虚拟机的基本信息

VMID	MAC	IP	HOSTIP	TVDID
1	83	203	150	2
2	41	151	200	3
3	82	202	200	2
4	84	204	150	2, 3

5.3　功能测试

首先，对基于 VPE 虚拟域的原型系统进行相关功能测试，包括可信虚拟域的加入以及跨域访问等边界安全策略测试，在图 8 中区域 1 中显示的是 VM1 的 IP 地址，区域 2 中显示的是 VM1 不能与 VM2 进行通信(不属于同一虚拟域)。区域 3 中显示 VM1 与 VM3 能够正常通信(属于同一个虚拟域)。

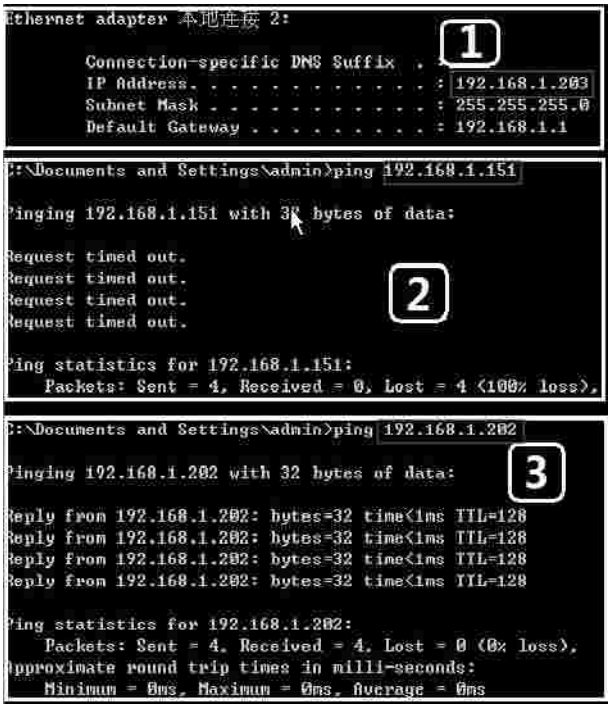

图 8　同一虚拟域内虚拟机间及不同虚拟域间通信

图 9 显示了可信虚拟域管理员用户 User4 登录的虚拟机 4 能够实现域内访问及跨域访问。当 User4 需要访问 VM2 时，经过 RBAC 跨域访问策略验证，将其映射成域 3 的普通用户，同时 VM4 对应的记录的 TVDID 字段增加域 3 的信息，即 VM4 同时属于域 2、3，经过认证与角色转换 VM4 可以同时与 VM2 和 VM3 正常通信。

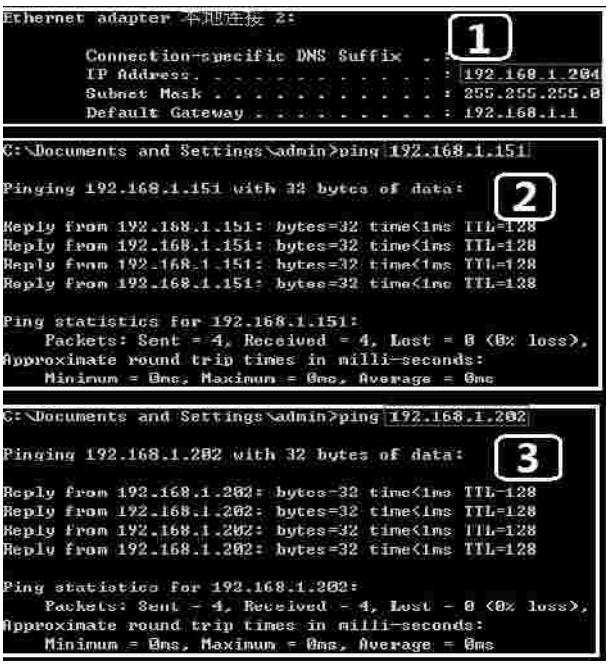

图 9　高权限用户的跨域访问

在图 10 中，通过 iperf 测试不同主机上虚拟机 2 种情况下 TCP 协议的网络吞吐量随着窗口大小变化的情况。本文进行了 3 组实验测试，分别为 TVD 数据加密传输、TVD 数据传输（数据分组未加密）、Xen 原始数据分组传输。从图中 3 组数据中可以看出加入 TVD 之后的网络性能较 Xen 原始数据分组传输网络性能略有下降，这是由于 TVD 系统数据传输的过程中进行数据分组的截获并进行相关的边界安全策略验证；同时，TVD 加密传输在完成边界策略验证之后，在内存中对数据分组进行加密传输，在提高了数据传输安全性的同时略微降低了传输效率。然而整体趋势下，TVD 虚拟网络、TVD 加密传输及 Xen 原始虚拟网络随着网络窗口尺度增大网络传输速度上升，当到达 128 KB 左右之后，网络速度趋于平衡，性能损耗大约为 18%以内。

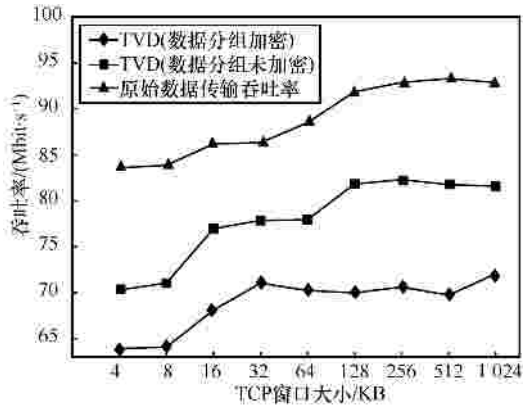

图 10 Xen 原始驱动与 VPE 在不同窗口下吞吐量对比

图 11 中描述了不同主机上虚拟机之间进行 UDP 的带宽利用率随着带宽的变化情况，实验环境链路中缓存设置为带宽时延积。与 TCP 的网络吞吐率测试相同，分别进行了 TVD 加密传输（数据分组加密传输）、TVD 数据传输（数据分组未加密）、Xen 原始数据分组传输。从图 11 中可以看出，在网络带宽较低的情况下，加入 TVD 系统后的虚拟网络、TVD 加密传输网络，与 Xen 原始情况相比，前 2 种网络带宽利用率较高；然而，随着网络带宽的增加（当带宽达到 70M 之后），UDP 的带宽利用率都会随着带宽的增大带宽利用率均呈现下降趋势，且原始 UDP 性能下降快于本系统。这可能是由于 Xen 原始网络数据分组需要经过网桥设备，而本系统不经过网桥，在高带宽情况下，终端网桥的处理能力成为性能主要瓶颈。加入 TVD 系统后的虚拟网络与 TVD 加密传输网络两者之间的带宽利用率差距在 4%左右，这是由于内存中进行数据分组加密的速度较快，对网络传输性能影响较小。本文的方案吞吐率整体性能与 Xen 原始性能差距不大。

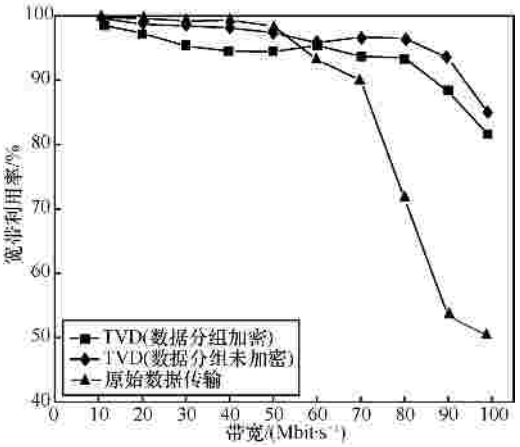

图 11 Xen 原始驱动与 VPE 的带宽利用率对比

6 结束语

本文提出了基于 VPE 的可信虚拟域构建机制 TVD-VPE，设计相关安全策略(可信虚拟域的加入/退出协议，身份认证机制、跨域访问协议等)将用户与虚拟域进行绑定，实现可信虚拟域的灵活配置。在此基础之上，设计基于分离式设备驱动的虚拟网络 VPE，其截获网络数据分组，并利用 RBAC 机制实现可控互联，这很好地解决了对硬件路由器和网络拓扑的依赖问题；同时对传输的数据分组进行加密和二次封装，确保通信安全。最后，在云平台下，实现了原型系统并进行了功能测试及性能测试，测试结果证明本系统可以有效地防止虚拟域间的非法访问，系统的加密方案略微降低了 Xen 的网络性能。本文研究了静态情况下虚拟域的可信构建机制，接下来，将深入研究虚拟机迁移过程中访问控制策略的同步机制等相关问题。

参考文献：

[1] 罗军舟，金嘉晖，宋爱波. 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. Cloud computing: architect nd key technology[J]. Journal on Communications, 2011, 32(7):3-21.

[2] BELLOVIN S M. Virtual machines,virtual security? [J]. munications of the ACM , 2006, 49(10): 104.

[3] 王丽娜，高汉军，刘炜. 利用虚拟机监视器检测及管理隐藏进程[J]. 计算机研究与发展, 2011, 48(8):1534-1541.
WANG L N, GAN H J, LIU W. Detecting and managing hidden

process VIA hypervisor[J]. Journal of Computer Research and Develop, 2011, 48(8):1534-1541.

[4] BUSSANI A, GRIFIN J L, JANSEN B, *et al*. Trusted Virtual Domains: Secure Foundation for Business and IT Services[R]. Research Report RC 23792, IBM Research, 2005.3-15.

[5] GRIFIN J L, JAEGER T, PEREZ R, *et al* . Trusted virtual domains: toward secure distributed services[A]. Proc of the 1st IEEE Workshop on Hot Topics in System Dependability(HotDep'05)[C]. Berkeley: USENIX, 2005.4.

[6] CASADO M, KOPONEN T, MOON D, *et al*. Rethinking packet forwarding hardware[A]. Proc of the Seventh ACM Workshop on Hot Topics in Networks[C]. Calgary, 2008.1270-1276.

[7] GREENHALGH A, HUICI F, HOERDT M, *et al*. Flow processing and the rise of commodity network hardware[J]. Sigcomm CCR, 2009,49(10):21-26.

[8] WANG Y, KELLER E, BISKEBORN B, *et al*. Virtual routers on the move: live router migration as a network-management primitive[A]. Pro of the ACM SIGCOMM 2008 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications[C]. Seattle, 2008. 231-242.

[9] CHEN X, MORLEY Z, JACOBUS M, *et al*. ShadowNet:a platform for rapid and safe network evolution[A]. Proc of the 2009 Conference on USENIX Annual Technical Conference (2009)[C]. Sandigeo, USENIX, 2009.3.

[10] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, *et al*. OpenFlow: enabling innovation in campus networks[A]. Proc of SIGCOMM CCR'08[C]. New York, 2008.69-74.

[11] BAVIER A, FEAMSTER N, HUANG M, *et al*. In VINI veritas: realistic and controlled network experimentation[A]. Pro of ACM SIGCOMM 2008 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications[C]. Pisa, 2006. 3-14.

[12] ANDERSON T, PETERSON L, SHENKER S, *et al*. Overcoming the Internet impasse through virtualization[J]. Computer, 38(4): 34-41.

[13] RIZZO L, CARBONE M, CATALLI G. Transparent acceleration of software packet forwarding using netmap[A]. Proc of 2012 INFOCOM[C].Pisa, 2012.2471-2479.

[14] KOPONEN T, CASADO M, GUDE N. Onix: a distributed control platform for large-scale production networks[A]. Proc of the 9th USENIX Symposium on Operating Systems Design and Imple ntation(OSDI 10)[C]. Vancouver: USENIX, 2010.351-364.

[15] CATUOGNO L, DMITRIENKO A, ERIKSSON K, *et al*. Trusted virtual domains-design, implementation and lessons learned[A]. Proc of the 2009 INTRUST[C]. Heidelberg, Springer, 2009. 156-179.

[16] SRIVASTAVA A, GIFFIN J. Tamper-resistant, application-aware blocking of malicious network connections[A]. Proc of '08 Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection[C]. Berlin, Springer-Verlag,2008.39-58.

[17] CASADO M, KOPONEN T, RAMANATHAN R, *et al*. Virtualizing the network forwarding plane[A]. Proc of the Workshop n Programmable Routers for Extensible Services of Tomorrow[C]. York, 2010. 908-914.

[18] BARHAM P, DRAGOVIC B, FRASER K, *et al*. Xen and the art of virtualization[A]. Proc of 19th ACM Symposium on Operating Systems Principles(SOSP-2003)[C]. Bolton Landing, 2003. 164-177.

[19] DAVOLI R. VDE: virtual distributed ethernet[A]. Proc of 1st International Conference on Testbeds & Research Infrastructures for the DEvelopment of Networks & Communities(TRIDENTCOM 2005)[C]. Trento, 2005.213-220.

[20] ANHALT F, PRIMET P V B. Analysis and Evaluation of a Xen Based Virtual Router[R]. Technical Report 6658, Inria, 2008.

[21] PFAFF B, PETTIT J, KOPONEN T, *et al*. Extending networking into the virtualization layer[A]. Proc of The 8th ACM Workshop on Hot Topics in Networks[C]. New York, 2009. 15-21.

[22] **王丽娜, 高汉军, 余荣威. 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. Research of constructing tr ted virtual execution environment based on trust extension[J]. Journal on Communications, 2011, 32(9):1-8.

作者简介：

王丽娜（1964-），女，辽宁营口人，武汉大学教授、博士生导师，武汉大学计算机学院副院长，武汉大学空天信息安全与可信计算教育部重点实验室主任，主要研究方向为云计算安全、信息隐藏、网络安全等。

张浩（1986-），男，湖北襄樊人，武汉大学博士生，主要研究方向为云计算安全、系统虚拟化、网络安全。

余荣威（1981-），男，江西九江人，武汉大学讲师，主要研究方向为可信计算和密码协议。

高汉军（1983-），男，武汉大学博士生，主要研究方向为系统虚拟化、可信计算和网络安全。

甘宁（1983-），男，辽宁辽阳人，武汉大学硕士生，主要研究方向为可信计算、网络安全。