文章编号:2095-6134(2015)05-0701-07

在桌面虚拟化系统中实施国产密码算法*

林雪燕[1,2,3], 林璟锵[1,2†], 管 乐[1,2,3], 王 雷[1,2]

(1 中国科学院数据与通信保护研究教育中心, 北京 100093;
2 中国科学院信息工程研究所 信息安全国家重点实验室, 北京 100093; 3 中国科学院大学, 北京 100049)

(2014 年 9 月 1 日收稿; 2015 年 3 月 3 日收修改稿)

Lin X Y, Lin J Q, Guan L, et al. China standard cryptographic algorithm implementation in virtual desktop system [J]. Journal of University of Chinese Academy of Sciences, 2015,32(5):701-707.

摘 要 在分析现有主流桌面虚拟化方案的基础上,综合考虑中国在通信安全方面的法律要求,对虚拟桌面传输协议的特性进行分析与总结,同时对 KVM 方案的 SPICE 协议进行基于国产密码算法的安全性改造. 通过在 OpenSSL 中添加国产密码算法 SM3/SM4,用以替换 SPICE 协议中 SSL 使用的 SHA1/AES 算法,提供机密性和完整性保护. 实验表明,该方案不仅能保证传输的安全性,同时能保持其性能,推广了国产密码算法的应用,符合国家相关管理条例.

关键词 国产密码算法; 桌面虚拟化; SPICE 协议; OpenSSL

中图分类号:TP309 **文献标志码**:A **doi**:10.7523/j.issn.2095-6134.2015.05.018

China standard cryptographic algorithm implementation in virtual desktop system

LIN Xueyan[1,2,3], LIN Jingqiang[1,2], GUAN Le[1,2,3], WANG Lei[1,2]

(1 *Data Assurance and Communication Security Center, Chinese Academy of Sciences, Beijing* 100093, *China*;
2 *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093, *China*;
3 *University of Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract Currently desktop virtualization technology has become a focal point of the cloud computing technology and we analyze the main virtual desktop systems. Considering the legal requirement in communication security, we summarize the characteristics of the virtual desktop transmission protocols and choose the SPICE protocol, based on KVM, to improve the transmission security. In SPICE, the communication between the client and server can be secured by using OpenSSL. We propose to support the China standard cryptographic algorithms including SM3 and SM4 in the open-source project OpenSSL to ensure the security of virtual desktop system. The experimental results show that our scheme can not only ensure the safety of the transmission, but also keep good performance.

* 国家重点基础研究发展(973)计划(2014CB340603)、国家高技术研究发展(863)计划(2012AA013104,2013AA01A214)和中国科学院战略性先导专项(XDA06010702)资助

† 通信作者,E-mail:linjingqiang@iie.ac.cn

Key words China standard cryptographic algorithms; desktop virtualization; SPICE protocol; OpenSSL

随着云计算和虚拟化技术的快速发展,桌面虚拟化技术应运而生. 桌面虚拟化系统与传统 PC 相比,具有成本低、管理方便等显著优势,目前已被广泛使用. 虚拟桌面传输协议作为桌面虚拟化系统的重要组成部分,负责客户端和服务器之间的信息传递,包括图像、光标、键盘鼠标输入等信息. 通常这些信息会包含用户的隐私数据, 必须采取必要的措施,如将信息进行加密传输等,来保证这些信息的安全. 目前主流的虚拟桌面传输协议都不同程度地实现了数据传输安全保护,但是这些解决方案都是由国外厂商提供,而且很多都是采用公开源代码的软件包,其安全性有待进一步验证. 特别是, Snowden 事件和 OpenSSL HeartBleed 漏洞与 CCS 漏洞表明开源软件包容易被广泛分析并利用其漏洞来实施攻击,而且无法确定其是否留有后门. 因此在安全性要求较高的环境中,不适合采用这些已有的、国外厂商提供的、基于国外密码算法的安全解决方案. 近几年来,中国的密码算法研究进展很大,先后破解了全球两大密码算法 MD5[1] 和 SHA1[2]. 国家密码管理部门也先后发布了中国自主研发的商用密码算法标准,包括非对称算法 SM2、哈希算法 SM3 和对称算法 SM4.

本文提出使用安全、高效的国产密码算法对虚拟桌面的数据传输过程提供机密性和完整性保护,加强传输的安全性,同时保持其良好的性能.

1 桌面虚拟化系统和虚拟桌面传输协议

1.1 桌面虚拟化系统

桌面虚拟化系统是对数据中心的服务器 Host 进行服务器虚拟化,来运行大量的独立的桌面操作系统,每个操作系统都可以按需安装各种应用,并根据专有的虚拟桌面传输协议将需要显示的图像、声音等数据发送给终端设备,图 1 描述了典型的桌面虚拟化系统结构. 用户可以使用多种终端设备,通过网络连接到虚拟桌面管理平台 Manager,身份认证成功后,用户请求连接自己的虚拟机,Manager 将用户的连接请求参数发送给虚拟机服务器 Host,Host 进行相关管理操作,比如虚拟机的端口映射等,并向 Manager 返回虚拟机的相关参数,Manager 将虚拟机的参数传递给用户,用户与虚拟机服务器通过虚拟桌面传输协议直接建立连接.

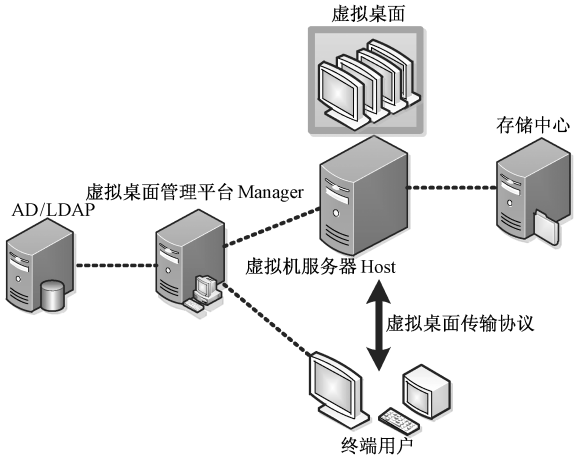

图 1 典型的桌面虚拟化系统

Fig. 1 Typical virtual desktop system

与传统的本地 PC 架构相比,桌面虚拟化系统具有如下优势[3]:

1)用户通过互联网就可以访问自己的桌面系统,而且桌面虚拟化系统的客户端支持多种终端设备,比如台式机、手机、平板电脑、瘦客户机等,用户可以随时随地地访问自己桌面的数据和信息.

2)桌面虚拟化系统采用集中管控的方法,系统管理员不必频繁地对大量终端进行补丁更新、病毒查杀等管理和维护工作,减少了管理成本.

3)用户所有数据的存储和计算都在服务器端的数据中心实现,本地客户端并不保存这些数据,能够有效避免由于本地客户端的安全隐患而导致的信息泄露,增加了安全性.

目前桌面虚拟化系统已被广泛使用,典型的解决方案有 Microsoft 的 Med-v 方案、VMware 的 VMware View 方案、Citrix 的 Xen Desktop 桌面虚拟化方案和 RedHat 的 Enterprise Virtualization 方案.

1.2 虚拟桌面传输协议

虚拟桌面传输协议是虚拟桌面服务器和用户终端之间的传输通信协议,为终端用户交付虚拟

桌面环境. 协议主要对用户终端的操作等输入信息以及服务器图像显示、声音等输出信息进行传输,同时处理其他与虚拟桌面相关的功能,如实时迁移、剪贴板共享等. 虚拟桌面显示的内容从服务器端传输到用户终端之前一般要进行数据压缩,以提高传输效率.

目前,主流的虚拟桌面传输协议有:remote display protocol (RDP)、independent computing architecture (ICA)、PC-over-IP (PCoIP)、simple protocol for independent computing environment (SPICE). 表 1 对这些虚拟桌面传输协议的性能进行了分析与比较[4-5]. 在桌面虚拟化系统应用中,用户终端在与虚拟化服务器数据交换的过程,数据通过互联网进行传输,存在数据被窃取、篡改的风险. 虚拟桌面服务器和用户终端之间的数据传输的安全性主要是通过虚拟桌面传输协议实现的. 所以,选择一个高效、安全的传输协议,对保障桌面虚拟化系统的安全性至关重要.

表 1　各虚拟桌面传输协议比较

Table 1　Virtual desktop transmission protocols comparison

	RDP	ICA	PCoIP	SPICE
带宽传输要求	高	低	高	中
图像展示体验	中	中	高	中
双向音频支持	高	高	低	高
视频播放支持	高	中	低	高
用户外设支持	低	高	低	中
支持厂商	Microsoft	Citrix	VMWare	Red Hat

目前,上述 4 个传输协议都采用加密的方式保证数据的安全性,但是实现方式差别较大[6]:RDP 传输采用 TCP 协议,使用 RC4 对称加密算法加密所有的数据来保护数据安全[7],目前支持的加密长度有 40 bit,56 bit 和 128 bit;PCoIP 协议的实现同时使用 UDP 和 TCP 协议,会话建立与控制采用 TCP 协议,而优化流媒体的内容则使用 UDP 协议,传输过程中采用 128 bit 的 AES 算法进行加密;ICA 协议采用 RC5 对称加密算法进行加密[8],目前支持的加密长度有 40 bit,56 bit 和 128 bit,同时还支持 SSL/TLS 传输;SPICE 协议自身没有提供数据加密处理,但是可以根据需要选择是否采用 SSL 传输,保证数据安全. SPICE 协议使用 OpenSSL 实现 SSL 传输,选用 128 bit 的 AES 算法加密传输中的数据信息,保护数据的安全.

使用密码算法保护数据的安全是普遍采用的数据保护方案,而密码算法的选取对于数据信息的安全至关重要. 如果选取留有后门的密码算法对数据进行保护,即便算法强度再高、密钥长度再长,也无法保证数据信息的安全. 2013 年,Snowden 给全球信息安全界扔下了一枚重磅炸弹,揭露了美国国安局(NSA)与 RSA 公司的交易丑闻. 据称,RSA 公司将 NSA 提供的方程式设定为 BSafe 安全软件的优先或默认随机数生成算法,在移动终端的加密技术中放置后门[9]. 目前的桌面虚拟化方案主要由国外厂商提供,使用的密码算法也都是由国际发布的密码算法,如 AES、RSA、SHA1. 因为无法确定在密码算法实现中是否留有后门,所以如果国内的单位采用已有的桌面虚拟化解决方案,对该单位的敏感信息、商业机密数据存在极大的安全威胁. 所以,需要在桌面虚拟化方案的关键位置使用自主实现的密码算法实施通信保护. 同时,这也符合中国的商用密码管理条例.

通过分析,RDP 协议不能在非微软的平台下使用、使用局限性较大,也难以进行改造;PCoIP 协议与 RDP 协议一样是运行在 Windows 下;ICA 是 Citrix 专有的协议,协议规范不开放;相对其他 3 个协议,SPICE 协议有如下优点:首先,SPICE 是一个基于 TCP 的完全开放的协议;其次,SPICE 的架构是跨平台的,允许更大的互操作性,包括 Linux 和 Windows;而且,SPICE 独特的架构设计也为其传输性能提供了很好的支持. 目前,越来越多的虚拟桌面开发商使用这种实现方式,SPICE 客户端也具有各种形式的广泛支持. 所以,我们选择使用 SPICE 协议进行改造.

2　SPICE 虚拟桌面传输协议

2.1　红帽企业桌面虚拟化

红帽企业桌面虚拟化解决方案包括 3 个部分:红帽企业虚拟化 Hypervisor、红帽企业桌面虚拟化管理平台和 SPICE 传输协议. 红帽企业虚拟化 Hypervisor 又称为虚拟机监视器,它使用全虚拟化技术,以基于 KVM 技术的红帽企业 Linux 内核为基础,允许多个虚拟机并发地运行在一台计算机上,是一种高性能、安全的系统管理程序. 红帽企业桌面虚拟化管理平台是红帽企业桌面虚拟化系统的控制中心,它是一个带有图形化管理控

制台和编程界面的集中化企业级虚拟化管理引擎,它拥有一套全面的管理工具,管理员可以用它创建、监控和维护虚拟桌面. SPICE 传输协议是红帽企业桌面虚拟化系统使用的远程传输协议,将用户终端与其虚拟桌面连接,能够提供与物理桌面近乎相同的最终用户体验,它是红帽桌面虚拟化系统的一个重要组成部分.

2.2 SPICE 协议框架结构

SPICE 传输协议是一个高性能的、动态的、具有自适应能力的虚拟桌面传输协议,它主要包括 SPICE 客户端、SPICE 服务器和相应的 QXL 设备及 QXL 驱动[10],如图 2 所示.

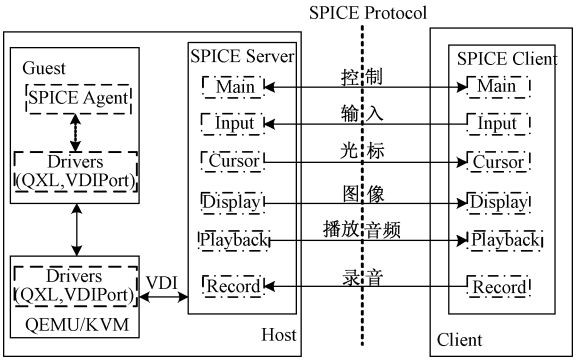

图 2 SPICE 协议框架结构图

Fig. 2 SPICE framework

由图 2 可知,SPICE 客户端与服务端使用 SPICE 协议进行通信,通过通道进行连接,每一种通道专门负责一种特定类型数据的传输和通信. 每个通道用一个专用的 TCP 套接字来传输一种数据,这个套接字可以是安全的(使用 SSL)或不安全的. 协议中主要包括 6 个通道:主通道、显示通道、输入通道、光标通道、播放通道、记录通道. 随着 SPICE 协议的不断完善,又增加了新的通道,如 USB 通道、SmartCard 通道. 在客户端,每个通道有一个专用的线程,通过不同的线程优先级,可以给每个通道提供不同的服务质量(Quality-of-Service,QoS).

SPICE 客户端是用户终端上的跨平台软件组件,支持 Windows、Linux、Mac OS 等. SPICE 客户端具有多种形式的广泛支持:有基于 GTK 的 spice-gtk 实现,以插件方式运行于 Firefox 的 spice-xpi 实现和 Android 系统的 SPICE 客户端 aSpice 实现. spice-gtk 是基于 glib 对象的 SPICE 客户端实现,在 spice-gtk 中,SpiceMainChannel 类负责处理主通道的消息,它负责发起和管理其他的通道,如创建、连接和断开等.

SPICE 服务端的功能是用 libspice 库实现的,它通过 SPICE 协议与客户端进行通信,通过 VDI 接口与 QEMU[11]虚拟设备进行交互,来处理用户的鼠标、键盘等输入. QEMU 为每一个虚拟桌面提供一个虚拟机进程,由于服务端直接与虚拟设备进行交互,不需要经过虚拟机中的 Guest 操作系统,因此可以在 SPICE 服务器上运行多种 Guest 操作系统.

除客户端与服务端外,SPICE 协议提供了其他组件用于增强功能,比如运行于 Guest 操作系统上的 QXL 图形显示驱动,用来提供更好的远程显示效果. SPICE 协议还提供安装在 Guest 操作系统上的可选软件组件 spice-agent,它能够执行面向客户机的管理任务来增强用户的体验,如显示参数设置、远程剪贴板等.

2.3 SPICE 传输协议的安全机制

SPICE 传输协议的安全机制主要包括 2 个方面. 第一,使用公钥算法 RSA 建立连接和进行身份认证. 建立会话主要由主通道进行,其他通道在主通道建立后才进行连接. 首先客户端向服务器发送请求连接 SpiceLinkMess 消息,服务器收到消息后向客户端发送应答 SpiceLinkReply 消息,并生成 1 024 bit 的 RSA 密钥对,并将公钥发给客户端. 客户端收到应答消息后,检测其中的错误码. 如果没有错误,客户端就用接收到的公钥加密用户的口令,并将加密结果发送给服务器. 服务器解密获得用户的口令,校对后将连接的结果发送给客户端. 客户端检查连接结果如果是 SPICE_LINK_ERR_OK,就建立正常的连接,用户通过身份认证;如果应答消息显示出现错误,客户端就必须重新连接进行身份认证. 第二,虽然 SPICE 协议中并没有提供数据加密处理,但是可以选择是否采用 SSL 传输来满足不同的安全需求. 目前协议采用 OpenSSL 进行加密. OpenSSL 是 SSL 协议的一个开源项目实现,主要为应用层通信提供身份认证和加密敏感数据,以及保护数据的完整性,目前是互联网上应用最广泛的安全传输方法.

3 SM2/3/4 算法在 SPICE 中的实施方案

3.1 SM2/3/4 密码算法介绍

SM2 算法是由国家密码管理局发布的基于

椭圆曲线的公钥密码算法[12]. 算法标准包括 4 个部分:第 1 部分是算法介绍,第 2—4 部分分别从数字签名、密钥交换、公钥加密方面介绍 SM2 算法的应用,这 3 部分都可为安全产品生产商提供产品和技术的标准定位以及标准化的参考,提高安全产品的可信性与互操作性.

SM3 算法是一个哈希算法,能对长度为 $L(L<2^{64})$ bit 的消息 m,经过填充和迭代压缩,生成哈希值,哈希值的长度为 256 bit[13]. 该算法能用于密码应用中的数字签名和验证、消息认证码的生成与验证以及随机数的生成,可满足多种密码应用的安全需求.

SM4 算法是一个分组密码算法,它的分组长度为 128 bit,密钥长度为 128 bit[14]. 加密算法与密钥扩展算法都采用 32 轮非线性迭代结构. 解密算法与加密算法的结构相同,只是轮密钥的使用顺序相反,解密轮密钥是加密轮密钥的逆序.

3.2　SPICE 传输安全机制改进方案

通过分析 SPICE 协议的源代码,得到 SPICE 协议的客户端与服务器的相关组件图,如图 3 所示. 其中与传输相关的主要在 Spice-common 组件,spice-protocol 以头文件的形式提供,主要定义传输的消息格式,SSL 模块负责进行安全传输,其他相关 API 包括内存分配等. 由于 SSL 模块是嵌在 Spice-common 这个公共模块的,通过调用 OpenSSL 的密码算法库 libcrypto. so 和 SSL 协议库 libssl. so,所以对 OpenSSL 进行修改后,需要重新生成自签名证书,并对 SPICE 的源代码进行重新编译、安装.

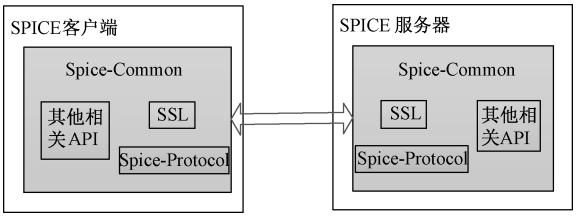

图 3　SPICE 相关组件

Fig. 3　Related components of SPICE

由于 SPICE 协议采用 OpenSSL 实现 SSL 传输,所以可以通过改进 OpenSSL 的算法支持列表,使得 OpenSSL 实现对国产密码算法的支持,从而实现 SPICE 协议采用国产密码算法进行安全传输. 数据传输过程中默认使用的对称、哈希、非对称算法分别为 AES 算法、SHA1 算法和 RSA 算法. 因此,我们提出 SPICE 传输协议的改进方案,即仍然采用 OpenSSL 进行传输,但是使用 SM2 算法进行身份认证和会话密钥协商,使用 SM3 算法保证数据的完整性,使用 SM4 算法对双方通信数据进行加密,提供机密性保护.

为实现上述方案,我们在生成自签名证书的时候,使用 SM2 算法,生成 256 bit 的公私钥对,并采用 PEM 格式以方便 OpenSSL 中库函数的处理. 该方案的流程图如图 4 所示.

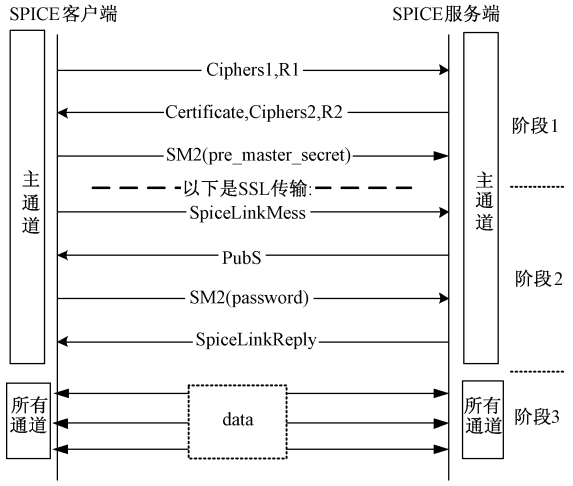

图 4　SPICE 安全通信过程

Fig. 4　Secure communication process of SPICE

该方案分为 3 个阶段,阶段 1 主要是 SSL 连接的过程. 双方创建主通道,并由客户端发起请求连接 SSL_connect( ). 阶段 1 和阶段 2 都是在主通道进行的. 客户端将算法列表 Ciphers1 和用作产生主密钥的随机数 R1 发给服务器. 服务器将自己的证书及证书链、算法列表 Ciphers2、用作产生主密钥的随机数 R2 发送给客户端. 客户端对服务器的证书进行验证,再产生随机密码串 pre_master_secret,并用服务器的公钥对其进行加密,将加密结果 SM2(pre_master_secret) 发送给服务器. 客户端和服务器分别根据 pre_master_secret 以及客户端和服务器的随机数 R1、R2 计算出主密钥 master_secret,再根据 master_secret 计算对称加密密钥 K1 和 MAC 密钥 K2,并互相发送连接完成消息,SSL 连接建立. 之后的通信过程 SM4 算法所需的对称加密密钥和 SM3 算法所需的密钥就是从 master_secret 计算得到的 K1 和 K2.

阶段 2 主要对客户端的身份进行验证. 客户端向服务器发起 SPICE 连接请求 SpiceLinkMess,服务器收到请求后生成 256 bit 的 SM2 临时公私

密钥对,并将临时公钥 PubS 发给客户端. 客户端用收到的公钥加密自己的 password,将结果 SM2(password)发给服务器. 服务器用临时私钥 PriS 解密,获得用户的 password,校对后将应答消息返回给客户端 SpiceLinkReply. 如果 SPICE 连接成功,说明客户端即用户通过身份认证.

阶段 3,用户已经成功登录,客户端执行 spice_channel_iterate( ),调用 OpenSSL 接口 SSL_read( )和 SSL_write( )与服务器进行通信. 在阶段 2 和阶段 3,所有客户端和服务器之间的通信数据 data 都要先用 SM3 算法求哈希值 SM3(data),并附在 data 后,再用 SM4 将其加密,即 SM4[x,SM3(x)].

OpenSSL 的每种密码算法都定义了自己的接口,同时还使用 EVP 技术封装了所有的密码算法[15]. 其中,对称算法使用统一的 API 接口 EVP_Encrypt 和 EVP_Decrypt 进行数据的加解密,提供代码的可重用性;使用非对称算法进行密钥交换或者公钥加密时,使用统一的接口 EVP_Seal 和 EVP_Open 进行加密和解密,而进行数字签名时,则使用 EVP_Sign 和 EVP_Verify 进行签名和验证;采用 EVP_Digest 接口作为信息摘要算法统一的 EVP 接口,对所有信息摘要算法进行封装. 这些接口的定义都包含在"/crypto/evp/evp. h"文件里面,通过这样的统一封装,使得只需要在初始化参数的时候做很少的改变,就可以使用相同的代码但采用不同的加密算法进行数据的加密和解密,为算法替换工作提供了方便.

在对称算法中,EVP_Encrypt 的应用架构使用循环的结构用 EVP_EncryptUpdate( )函数进行加解密操作,它的函数声明是 intEVP_EncryptUpdate (EVP _ CIPHER _ CTX * ctx, unsigned char * out, int * outl, const unsigned char * in, int inl). 实际的加密操作是调用 EVP_CIPHER_CTX 成员加密算法结构 EVP_CIPHER 的成员函数 init( )、do_cipher( )进行加解密计算. 实验方案就是对 AES 的 init( )、do_cipher( )操作替换上 SM4 算法的对应代码.

在哈希算法中,EVP_Digest 的应用架构与 EVP_Encrypt 类似,使用循环的结构用 EVP_DigestUpdate( )函数进行哈希操作,它的函数声明是 int EVP_DigestUpdate(EVP_MD_CTX * ctx, const void * d, size_t cnt). 实际的哈希操作是调用 EVP_MD_CTX 成员哈希算法结构 EVP_MD 的成员函数 init( )、update( )进行哈希计算. 实验方案就是对 SHA1 的 init( )、update( )操作替换上 SM3 算法的对应代码. 并且,由于 SM3 算法的哈希值长度为 32 bytes,而 SHA1 为 20 bytes,故需要在"crypto/sha/sha. h"文件中 SHA _ DIGEST _ LENGTH 宏定义中做修改,相应地,SHA_CTX 的结构中存放哈希结果也要从 5 个寄存器改为 8 个寄存器.

替换 SM2 算法需要涉及的工作包括:SSL 服务器的证书解析和公钥获取、SM2 算法的加解密和签名验签调用. 由于 RSA 算法是基于大数分解难题,而 SM2 算法是基于有限域的椭圆曲线难题,它们的计算过程差异很大. 所以针对 SM2 算法,我们的方案是利用 OpenSSL 提供的 Engine 机制来替换 RSA 算法,具体做法是开发一个 Engine 对象. 我们将会在后续的工作中实现 SM2 算法的替换.

3.3 实验实现

根据上述方案,我们搭建了自己的桌面虚拟化系统. 通过在网上下载修复了 HeartBleed 漏洞和 CCS 漏洞的 OpenSSL 的源代码,版本号是 1.0.1h,修改其关键代码,重新编译安装. 使用 OpenSSL 自带的命令生成 SPICE 服务器的自签名证书,并将 SPICE 服务器的网络设置成桥接模式. 然后利用源码编译生成的 OpenSSL 动态链接库,编译和安装 SPICE 客户端 spice-gtk、SPICE 服务器 spice-server 和 KVM 模块 qemu-kvm,搭建桌面虚拟化系统. 为了验证实验方案的有效性,我们在 SPICE 服务器上建立了 2 个不同系统的虚拟机,分别是 Windows7 系统和 RHEL6.5 系统. 如图 5 所示.

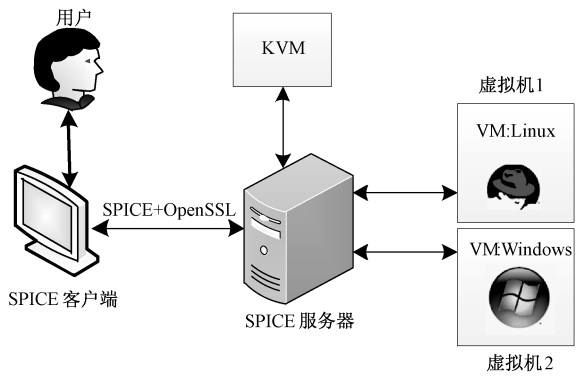

图 5 实验环境

Fig. 5 Experimental environment

我们通过测试用户进行文本输入情况下的显

示延迟来比较改进前后系统的性能及用户体验. 为了排除其他因素的影响,我们使用自动输入工具 AutoKey 模拟用户的重复输入行为,来测试改进方案前后的用户体验. 通过记录系统记录字符输入的时间点 t_1 和显示在屏幕上的时间点 t_2,得到往返的时间 RTT(round trip time),则 RTT = $t_2 - t_1$. 由于改进方案前后所处的网络环境是一样的,故可以认为 2 种情况下数据包在网络中传输的时间是一样的,唯一的差别就是客户端和服务器端的加解密处理过程. 所以测得的 RTT 指标即可反映出用户的体验. 为了获得可靠的实验结果,我们进行了一个小时的重复文本输入,并去掉刚开始 5 min 和结束前 5 min 的实验数据,以防止开始和结束这 2 个时间点可能引起的数据偏差. 测试结果如图 6 所示,RTT 的值在 0 ~ 1 s 之间,我们将其平均分成 10 个区间,统计每个区间的文本输入次数,横轴是 RTT 的值,纵轴是其对应数量的百分比.

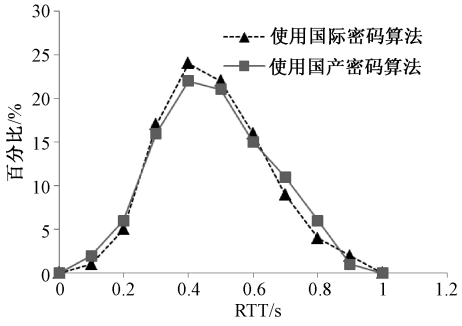

图 6　实验结果

Fig. 6　Experimental result

从实验结果可以看出,使用改进后的 OpenSSL 的桌面虚拟化系统,在文本编辑情况下与改进前的方案相比性能上差别不大.

我们还通过在虚拟桌面进行图片编辑,发现在图片更新时有稍微的延迟;进行视频观看时,丢帧现象比较明显,这主要是受网络环境影响.

4　总结与展望

本文介绍桌面虚拟化系统和虚拟桌面传输协议,并分析 SPICE 协议的安全机制. SPICE 协议采用 OpenSSL 实现数据的安全传输,我们提出用国产密码算法替换 OpenSSL 中的密码算法,在桌面虚拟化系统中实施国产密码算法,来保证传输的安全性. 我们实现了该方案的原型系统,并对其功能、性能进行了测试实验,实验结果表明该方案在提高安全性的基础上,仍然能保证良好的性能.

我们将在后续的工作中完成 SM2 算法的替换,使国产密码算法 SM2/SM3/SM4 都能应用到我们的桌面虚拟化系统中,替换现有 SPICE 协议使用的密码算法. 进一步,由于桌面虚拟化系统中的客户端相当一部分是移动终端,例如手机、平板电脑,因此,在后续工作中我们将本文提出的方案扩展到移动终端,用于保护移动终端用户数据的传输安全.

参考文献

[1] Wang X, Yu H. How to break MD5 and other hash functions [C]// Advances in Cryptology-EUROCRYPT 2005. Springer Berlin Heidelberg, 2005: 19-35.

[2] Wang X, Yin Y L, Yu H. Finding collisions in the full SHA-1 [C] // Advances in Cryptology-CRYPTO 2005. Springer Berlin Heidelberg, 2005: 17-36.

[3] 孙宇, 陈煜欣. 桌面虚拟化及其安全技术研究[J]. 信息安全与通信保密, 2012 (6): 87-88.

[4] 石屹嵘, 龚德志. 基于 SPICE 开源协议的云桌面技术架构研究[J]. 电信科学, 2013, 29(8): 162-167.

[5] Schlosser D, Binzenhofer A, Staehle B. Performance comparison of windows-based thin-client architectures[C]// Telecommunication Networks and Applications Conference, 2007, ATNAC 2007. Australasian, IEEE, 2007: 197-202.

[6] Wang J, Liang L. Survey of virtual desktop infrastructure system [EB/OL]. (2011-05-13) [2014-08-15]. https://tools.ietf.org/html/draft-ma-appsawg-vdi-survey-00#page-30.

[7] 罗 鹏, 祝跃飞. Windows 下 RDP 协议的安全性[J]. 计算机工程, 2007, 33(20): 145-147.

[8] Boca I. Citrix ICA Technology Brief[R]. Technical White Paper, 1999.

[9] Pfleeger S L. Taking action to build trust in security[J]. Security & Privacy, 2014, 12(2): 3-4.

[10] SPICE. SPICE sources and documentations[CP/OL]. [2014-08-15]. http://www.spice-space.org/.

[11] QEMU. QEMU sources and documentations [CP/OL]. (2012)[2014-08-15]. http://www.qemu.org/.

[12] 国家密码管理局. SM2 椭圆曲线公钥密码算法[CP/OL]. (2010-12-22)[2014-08-15]. http://www.oscca.gov.cn/UpFile/20101222214822692.pdf.

[13] 国家密码管理局. SM3 密码哈希算法. [CP/OL]. (2010-12-22)[2014-08-15]. http://www.oscca.gov.cn/UpFile/20101222141857786.pdf.

[14] 国家密码管理局. SM4 密码哈希算法. [CP/OL]. (2009-11-31) [2014-08-15]. http://gm.gd.gov.cn/upfile/2009113105257460.pdf.

[15] OpenSSL Sources and Documentations [CP/OL]. (2014) [2014-08-15]. http://www.openssl.org/.