doi:10.11959/j.issn.1000-436x.2017138

基于封闭环境加密的云存储方案

杜瑞忠[1,2]，王少泫[1,2]，田俊峰[1,2]
（1. 河北大学计算机科学与技术学院，河北 保定 071002；2. 河北省高可信信息系统重点实验室，河北 保定 071002）

摘 要：针对保护云存储中用户数据机密性的问题，提出了一种在云服务提供商处加密数据的云存储方案。通过虚拟机隔离技术来构造封闭计算环境，改进 RSA 公钥加密算法使其不需要重新产生大素数就能实现密钥变化，并通过 SSL 安全链接传输数据以及密钥，将数据在封闭计算环境中安全加密后再存储至分布式文件系统。封闭计算环境能阻止操作系统中不良应用以及云管理员的攻击，有效防范数据泄露。实验结果表明，用户数据的机密性得到了提升，并且相较于其他在云端加密的云存储方案，所提方案所带来的性能损耗降低了许多。

关键词：云存储；分布式文件系统；封闭计算环境；RSA
中图分类号：TP393.08 **文献标识码**：A

Cloud storage scheme based on closed-box encryption

DU Rui-zhong[1,2], WANG Shao-xuan[1,2], TIAN Jun-feng[1,2]
(1. College of Computer Science and Technology, Hebei University, Baoding 071002,China;
2. Key Lab on High Trusted Information System in Hebei Province, Baoding 071002,China)

Abstract: Aiming at protecting the confidentiality of data for cloud storage users, a scheme that encrypt data in cloud service providers was presented. The scheme constructed a closed-box computing environment by virtual machine isolation technique, improved algorithm of RSA to change keys without having to produce large prime numbers, transfer data and keys through SSL and encrypted data in the closed computing environment before storing to the distributed file system. Closed-box computing environment can prevent attacks from cloud administrators and malicious applications in the operating system. It also can guard against data leakage effectively. The result of experiment shows that the confidentiality of data improved and the performance loss is decreased considering to other cloud storage scheme that encrypt data in cloud.

Key words: cloud storage, distributed file system, closed-box computing environment, RSA

1 引言

云计算[1]具有有效降低应用成本、充分利用资源、提高计算能力、使用方便等优点而颇受欢迎。云存储[2]的概念是由云计算的概念发展而来的，云存储不单是一种存储方式，而是一种建立在互联网上的服务，它具有本地存储所不具备的众多优点，如海量存储、资源共享、成本低廉，但是云存储的安全问题却成为其最大的短板。数据存储的安全问题成为云存储甚至云计算推广过程的绊脚石。根据云存储应用调查报告[3]，只有 50%的人愿意使用云存储来进行数据备份等作业。Verizon 在 2016 年度数据泄露调查报告中指出，除了黑客攻击、木马病毒、钓鱼网站等外部因素，缺乏整套行之有效的安全管理系统，内部员工泄密以及内部管理等内部因素成为引发数据泄密事件的主要因素[4]。由此可见，

收稿日期：2017-01-12；**修回日期**：2017-03-30
基金项目：国家自然科学基金资助项目（No.61170254, No.60873203）；河北省自然科学基金资助项目（No.F2014201098）；河北省高等学校科学技术研究基金资助项目（No.ZD2016043）；河北省物联网数据采集与处理工程技术研究中心基金资助项目（No.河北 065201）
Foundation Items: The National Natural Science Foundation of China (No.61170254, No.60873203), The Natural Science Foundation of Hebei Province (No.F2014201098), The Science and Technology Research Project in Colleges and Universities of Hebei Province (No.ZD2016043), Hebei Engineering Technology Research Center for IoT Data Acquisition & Processing, North China Insitute of Science and Technology (No.Hebei 065201)

云存储的用户很难完全信任云服务提供商。

因此，保障数据安全的重要性尤为突出。云存储的安全性问题已引起学术界和企业界的广泛关注，相关研究主要围绕数据保密存储、安全审计和密文访问控制 3 个领域展开[5]。本文主要侧重于数据保密存储。

数据加密是保护云端用户数据机密性的基本手段，根据数据加密位置的不同，又分为在客户端进行加密、在云端进行加密。

在客户端加密，即在上传数据前在用户自己的设备上将数据进行加密。典型的有 ACCC[6]、HCRE[7]、AB-ACCS[8]等，将数据上传到云端之前进行加密能充分保障数据的安全，但是缺点也很明显：在用户客户端上加密，将会增加客户端的负担，并且随着移动设备的推广，将来在移动设备上上传数据也会变得越来越流行，这种加密手段会对移动设备使用云存储带来极大的障碍。如果客户端性能较差，加解密密钥又比较复杂，将会使用户的操作更加繁琐、耗时，客户端的计算负荷大大加重，而云平台强大的计算能力却没有得到充分利用。

在云端加密，即将数据上传至云端后再进行加密。典型的有文献[9]提出的在分布式文件系统服务器上部署安全虚拟监督系统的云存储方案，若用户要求保护单块数据的安全性，则在块数据服务器上部署安全虚拟监督系统，若用户要求保护整体数据的安全性，则在元数据服务器上部署安全虚拟监督系统，并且为了保障数据在网络中的传输安全，均采用了 SSL 协议。优点是使云平台的计算能力能够得到充分发挥，并且将云端的操作系统和分布式文件系统进行了隔离。其缺点是数据对于分布式文件系统是明文，因此，用户数据还是可能泄露给云服务提供商，并且多次使用 SSL 协议带来了性能损耗。

通过以上方法的分析，本文主要讨论了一种可以安全利用云服务提供商对数据进行加密的方法，即在云中构造一个封闭的计算环境，一次加密，多点安全存储，并改进 RSA 算法，实现密钥更新，在保障用户数据机密性的同时，充分发挥云计算平台的强大计算能力。

2 方案设计

2.1 云存储结构介绍

云存储的核心在于分布式文件系统，分布式文件系统是一项可以通过对远程设备进行存储和访问的技术，它可以将远程大量的存储设备整合起来作为一个整体以供存储[10]。通过对现有的一些分布式文件系统如 GFS[11]、HDFS[12]等的研究，可以发现其结构主要分为 3 个部分，分别是分布式文件系统客户端、元数据服务器、块数据服务器，如图 1 所示。

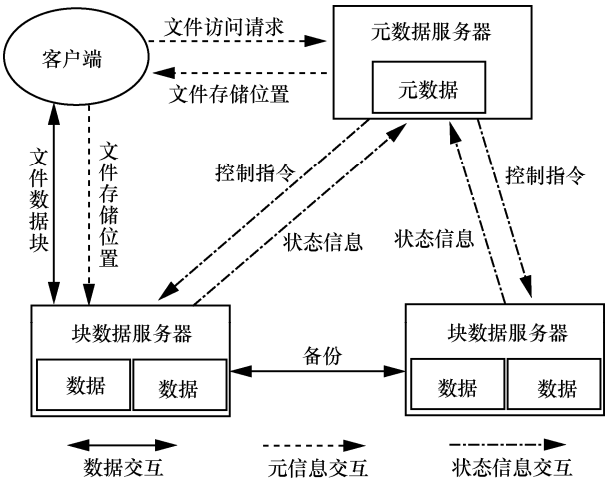


图 1 分布式文件系统结构

一般元数据服务器由一台或多台服务器构成，而块数据服务器则由许多服务器共同组成。通常元数据服务器负责分配和记录每个文件存储在块数据服务器的位置及每个文件的元数据信息；分布式文件系统客户端负责向元数据服务器提出存储请求，然后根据元数据服务器回馈的文件分块信息将数据传输至相应块数据服务器，读取文件时则依据元数据服务器存储的文件位置从指定的块数据服务器读取相应数据块；块数据服务器的工作则是存储文件块，并且根据需求可能会有备份。

2.2 封闭计算环境

2003 年，Tal[13]首次提出了封闭计算环境（closed-box）的概念，为了构造封闭计算环境需要用到隔离机制。隔离机制[14]是保障系统安全的重要手段，传统的一些操作系统因为不能为各种应用程序的运行提供一个相对安全或相对封闭的环境而存在很大的安全隐患。

最为典型的就是提出封闭计算环境概念的 Tal 设计的 Terra[15]系统，Terra 是一种使用虚拟机技术来实现隔离的系统，它分为普通虚拟机 CVM 和安全虚拟机 SVM，普通虚拟机对应的计算环境是

open-box VM，安全虚拟机对应的计算环境是closed-box VM，closed-box VM 中的程序数据无法被操作系统管理员窥探以及操作，作为一个相对独立的计算环境，它是安全的，除了其构建者外，系统中的其他主体均无法探测或修改其内容。Overshadow[16]和 Daoli[17]使用了类似的技术来达到内存和 I/O 的隔离保护的目的。Overshadow 不需要对操作系统和应用程序做任何修改，Daoli 相较于 Overshadow 来说，会对操作系统作少量的修改，这样的修改使其性能损失更小，并且 Daoli 支持 Docker 容器[18]。Daoli 保护下的系统的整体结构如图 2 所示。

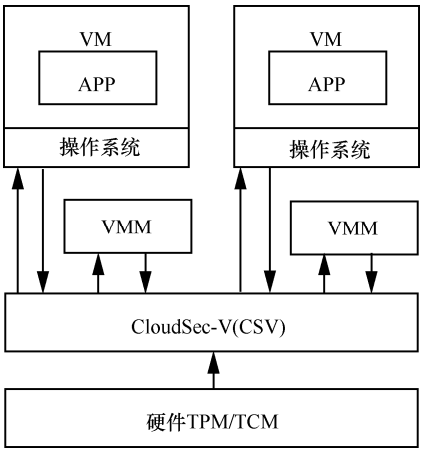


图 2　Daoli 保护下的系统结构

部署在硬件上的可信安全芯片（TPM）对 CloudSec-V（CSV）进行完整性保护，通过在 VMM 和硬件之间插入 CSV，使其运行在虚拟化扩展（VMX）根模式，具有软件栈中最高优先权，有效监控虚拟平台安全任务。与此同时，将 VMM 运行在虚拟化扩展非根模式，处于软件栈非特权层，仍然可以有效管理虚拟化任务，但无权进入 VM 计算环境，若在该层执行特权指令（如进入 VM 计算环境），则会造成指令流下限至根模式层。基于以上优点故采用 Daoli 来构造封闭计算环境。Daoli 现为 GitHub 上的开源项目，部署以及使用说明可参考 GitHub 上 Daoli 项目相关文档。

2.3　保护方案模型

对于一般的分布式文件系统来说，用户文件元数据信息的交互位于分布式文件系统客户端和元数据服务器之间，用户数据的交互位于分布式文件系统客户端和块数据服务器之间。为了保障数据和密钥传输的安全性，本文方案引入 SSL 安全连接。但引入 SSL 安全连接传输大文件带来的性能损耗是巨大的，侯清铧[9]在清华大学 Corsair 平台上测试，他的方案因多次引用 SSL 连接，以至 700 MB 的文件传输会给服务器端带来 11 倍多的性能损失，给客户端带来 6 倍多的性能损失，因此，本文尽量减少需要 SSL 安全连接传输的过程，并且在基于以上的分布式文件系统特点和要求下，可以得到如图 3 所示的保护方案模型。

当用户需要上传数据时，其 SSL 模块首先将数据明文加密，数据在网络传输过程中为密文。到达云端后，在封闭计算环境中先通过 SSL 模块解密出明文，再通过改进的 RSA 算法加密成复杂度高的密文，而后存储至分布式文件系统中的块数据服务器。具体的元数据信息交互过程如图 4 所示，云图案代表这个网络传输过程依然是不可信的。具体流程如下。

1) 用户首先通过安全存储申请界面向分布式文件系统提出存储请求，并上传数据基本信息，如文件大小和文件类别等。

2) 存储请求通过互联网传输至分布式文件系统。

3) 分布式文件系统根据文件大小决定是否需要分块并向元数据服务器请求各文件块存储位置。

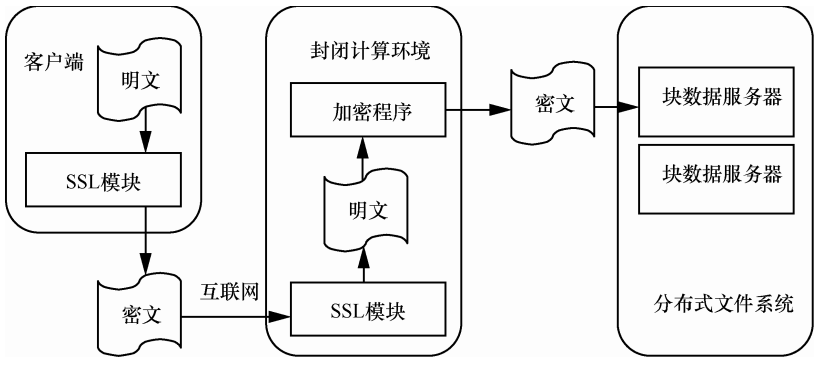


图 3　保护方案模型

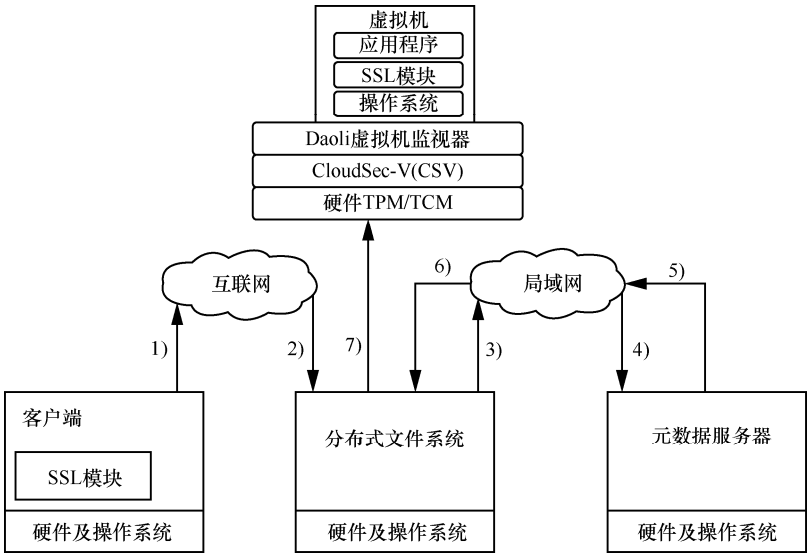


图4　元数据信息交互过程

4) 存储请求通过云端局域网传输至元数据服务器。

5) 元数据服务器向分布式文件系统返回相关信息。

6) 文件元数据信息通过云端局域网传输至分布式文件系统。

7) 分布式文件系统将文件元数据信息转发至封闭计算环境，至此文件元数据信息交互完成。

元数据信息交互完成后，便会进行块数据信息交互，具体过程如图5所示。

1) 用户明文数据经过 SSL 模块加密后得到密文，将数据密文以及 RSA 公钥上传。

2) 用户密文数据经过互联网传输至封闭计算环境，在封闭计算环境中的 SSL 模块将其解密后，再使用用户上传的 RSA 公钥对数据进行加密。

3) 封闭计算环境 VM 根据指令将文件分块加密或直接加密，然后发送文件块到元数据服务器指定的块数据服务器。

4) 加密后的文件在云端局域网传输至相应块数据服务器。

5) 封闭计算环境将存储完成信息汇报给分布式文件系统，至此文件块数据信息交互完成。

用户数据在进入封闭计算环境前是经过 SSL 协议加密的，经过封闭计算环境加密后，在整个云端传输、存储都是密文形式，可有效防止云管理员以及不良应用的窥探。读取文件则是申请封闭计算

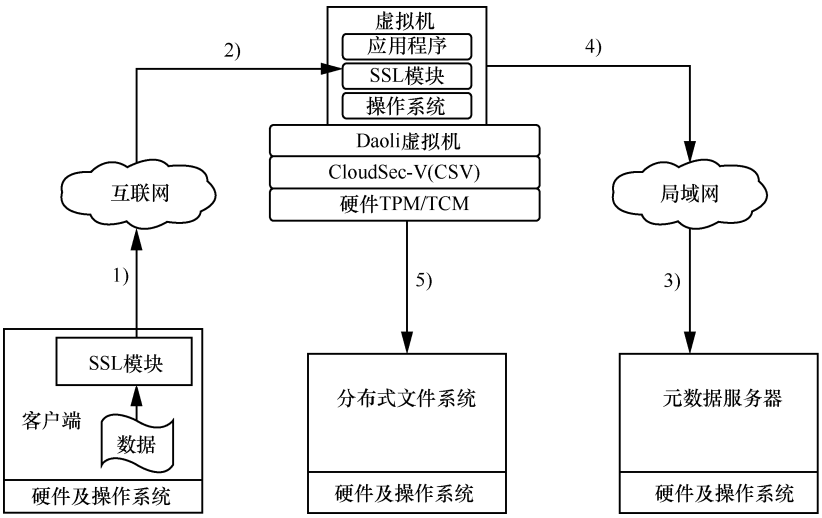


图5　块数据信息交互过程

环境将数据解密后再经过 SSL 协议传输至用户处，解密密钥由用户提供。

针对以上方案模型，采用如下密钥策略：用户存放数据时，由用户密钥管理模块生成一对公钥和私钥，将数据与公钥同时发送至云端。在封闭计算环境内用公钥加密后存储，每次需要读取数据时，生成一对新的 RSA 密钥，将新的公钥与旧的私钥发至云端，新的私钥存放在用户的密钥管理模块中。在云端将数据用旧的私钥解密后通过 SSL 协议加密发送至客户端，再将数据用新的公钥再次加密。这样即使旧的私钥被截获，也无法得到数据。具体过程如图 6 所示。

定义 1　*SSL*（*x*）代表 *x* 通过 SSL 协议加密传输。

定义 2　*PK* 代表 RSA 公钥。

定义 3　*SK* 代表 RSA 私钥。

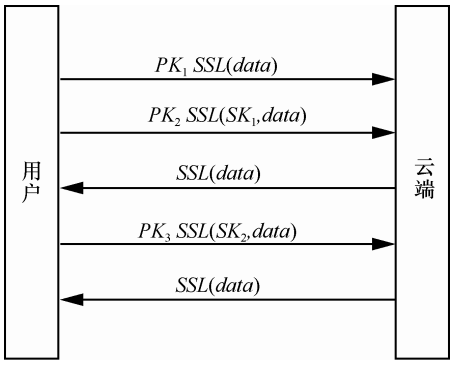


图 6　密钥及数据交换示意

2.4　算法描述

对于 RSA 公私钥体系，每次实现密钥的变化，最根本的是通过重新产生大素数来实现一次一密，但是这样实现一次一密的密钥长度比密文还长，性能损耗巨大。因此，通过对 RSA 的公钥和私钥上分别附加一个加密密钥和解密密钥来实现密钥变化，安全性提升虽不如重新产生大素数的高，但是效率却高出很多。

2.4.1　密钥产生

1) 选取 2 个大素数 p 和 q，计算 $n=pq$，n 可以公开，p 和 q 必须保密。

2) 计算 $\Phi(n)=(p-1)(q-1)$，任取 $2\leqslant e\leqslant\Phi(n)$，且 $gcd(e,\Phi(n))=1$，e 为加密密钥，公开。其中，$\Phi(n)$是 n 的欧拉函数值。

3) 计算 d，使 $ed=1\bmod(\Phi(n))$，d 为 e 的模反元素，其中，d 为解密密钥，需要保密。

4) 初始化 e_0 和 d_0，$2\leqslant e_0<n$，取 $d_0=tn-e_0(t\in N^*)$。t 保密，t 不参与加解密只参与 e_0、d_0 生成和改变。

5) 以{e，e_0，n}为公开钥，{d，d_0，t}为密钥。

6) 密钥改变。

生成随机整数 v，加密完成后

$$e_{0_{i+1}}=e_{0_i}(e+v)\bmod tn,i=0,1,2,\cdots \tag{1}$$

解密为

$$d_{0_{i+1}}=d_{0_i}(e+v)\bmod tn,i=0,1,2,\cdots \tag{2}$$

2.4.2　加密

将待加密的明文分成多组，每个分组 m 转换成的十进制数不大于模数 n，加密运算如式(3)所示。

$$c=(m^e+e_0)\bmod n \tag{3}$$

2.4.3　解密

对密文 c 的解密运算为

$$m=(c+d_0)^d\bmod n \tag{4}$$

2.5　适用场景

基于封闭环境加密的云存储方案主要适用的场景有以下几种。

1) 个人用户。客户端电脑性能不够强，为了安全存储数据，在自身客户端上加密数据会造成更大的负担，效率更低，因此，需要在云端完成数据加密工作。

2) 中小型企业。希望减少采购客户端，缩减开支并安全存储数据。

3) 移动端用户。为了及时修改或增加数据需要在移动端完成操作。因移动端性能现阶段与电脑相比仍然差距不小，完成数据加密工作负担太重，需要云端安全加密存储。

3　安全性分析

云存储让用户失去了对数据的绝对控制权，与此同时也带来了其特有的安全隐患，本文将其安全隐患主要分为 3 个方面，分别是传输过程中被截获、存储过程中被获得操作系统权限的应用或拥有操作系统权限的云服务商或使用同一台主机的不同租户窃取、存储在块数据服务器上的数据被窃取或破解即加密密钥的安全性问题。

3.1　传输过程

为了保障数据在网络传输中的安全性，本文方

案使用了 SSL 安全协议即安全套接字层[19]，SSL 在交换数据前通过握手协议来进行安全审查，在交换数据时提供具有机密性、可靠性、完整性的信道，其中，机密性由 DES、MD5 等加密技术保证，完整性由 X.509 数字证书鉴别，并且在 SSL 3.0 后包含了对 Diffie-Hellman 密钥交换进行短暂加密的支持，通过对服务器端的 Diffie-Hellman 指数的鉴别，可以抵御中间人攻击。

SSL协议已被各大厂商用于Web浏览器和服务器之间的加密数据传输。相较于由用户完成轻量级加密，云端完成强度较大的二次加密来保障传输过程中数据的安全性来说，采用 SSL 不仅能进行安全审查，并且使用起来更加方便，无论安全性还是便捷性都得到了提升。

3.2 存储过程

存储过程中的安全问题主要分为 2 个方面：1) 操作系统中应用程序良莠不齐，可能有些程序会获取操作系统权限来盗取数据；2) 需要对本身就拥有权限的云管理员进行防范。

传统操作系统通过地址空间隔离来实现进程间的隔离。进程的地址空间分为内核态和用户态，所有的进程共享一个内核空间，每个进程却拥有不同的用户地址空间。为了实现 2 种地址空间的链接，操作系统对每个进程分配了一套独立的页表，页表记录了内核空间向用户空间转换的地址，来完成线性地址到物理地址的转换。传统操作系统是极为复杂的，内部包含了上百万行的代码，代码越多，漏洞越多，因此传统操作系统很难为程序提供一个可靠的运行环境。但是，Daoli 使用的可信基是一个仅有 5 000 行静态代码的 CloudSec-V，通过可信计算方法有效将其固化（对其二进制代码取散列值，存入 TPM/TCM），并可将固化状态报告给检查者。要访问进程真正的物理地址需要获得映射列表，由于内核地址空间的特权等级要比用户空间高，只有获得操作系统权限才能访问映射列表，因此要访问映射列表必须切换到内核态，使用特权指令则会被 Daoli 截获，环环相扣实现安全。由 CloudSec-V 专门管理安全，VMM 管理平台虚拟化任务，实现安全与高效的统一。

相较于直接使用云服务商提供的机器进行加密，引入虚拟机监视器虽带来了一定的性能损失，但安全性也得到了提升。

3.3 密钥策略

数据存储在块数据服务器上后，加密密钥保证数据不被破解是保障数据安全性的关键。要证明改进的 RSA 算法的安全性，首先要保证其可用即证明算法的正确性。

3.3.1 算法的正确性

检验算法的正确性主要看能否从密文及密钥恢复出原文，即查看 $m = D(c) = (c + d_0)^d \bmod n$ 是否正确。

证明 由 $c = (m^e + e_0) \bmod n$，设 $c = (m^e + e_0) - yn (y \in Z)$

因为 $e_0 + d_0 = tn \ \ (t \in Z)$ 由解密式

$$\begin{aligned} D(c) &= (c + d_0)^d \bmod n \\ &= ((me + e_0) - yn + d_0)^d \bmod n \\ &= (me + (e_0 + d_0) - yn)^d \bmod n \\ &= ((me + tn - yn)^d \bmod n \\ &= ((me + (t - y)n)^d \bmod n \\ &= m^{ed} \bmod n \end{aligned}$$

因为 $ed = 1 \bmod \Phi(n)$，设 $ed = \Phi(n)k + 1 \ \ (k \in Z)$

所以

$$\begin{aligned} D(c) &= m^{ed} \bmod n \\ &= m^{\Phi(n)k+1} \bmod n \end{aligned}$$

下面，分 2 种情况来分析。

1) m 与 n 互素，那么由 Euler 定理可知 $m^{\Phi(n)} \equiv 1 \bmod n$，$m^{\Phi(n)k} \equiv 1 \bmod n$，$m^{\Phi(n)k+1} \equiv m \bmod n$，即 $D(c) = m^{\Phi(n)k+1} \bmod n = m \bmod n$

因为 $m<n$，所以 $D(c) = m \bmod n = m$。

2) $gcd(m,n) \neq 1$，由于 $n = pq$，因此 $gcd(m,n) \neq 1$ 意味着 m 是 p 或 q 的倍数，不妨设 $m = cp \ \ (c \in N^*)$。此时，必有 $gcd(m,n) = 1$，否则 m 也是 q 的倍数,从而是 pq 的倍数，与 $m < n = pq$ 矛盾。

由 $gcd(m,q) = 1$ 及 Euler 定理得

$$m\Phi(q) \equiv 1 \bmod q$$

所以

$$m^{k\Phi(q)} \equiv 1 \bmod q$$

$$\left[m^{k\Phi(q)}\right]\Phi(p) \equiv 1 \bmod q$$

$$m^{k\Phi(n)} \equiv 1 \bmod q$$

因此，存在一整数 r，使 $m^{k\Phi(n)} \equiv 1 + rq$，两边同乘 $m = tp$ 得

$$m^{k\Phi(n)+1} = m + rtpq = m + rtn$$

$$m^{k\Phi(n)+1} \equiv m \bmod n$$

因为 $m<n$，所以 $D(c)=m^{k\Phi(n)+1} \bmod n = m \bmod n = m$。

故算法的解密式是正确的，密文可以正确地由解密式恢复成明文。

3.3.2 算法的安全性

RSA 算法从提出到现在已经 30 多年了，广泛的应用证明 RSA 的安全性是非常可靠的。本文在其基础上新增加密密钥 e_0 和解密密钥 d_0，$e_0+d_0=tn\ (t\in Z)$，t 只有用户知道。从理论上攻破 RSA 已实属不易，现实中更是难上加难，本文对加密密钥又附加上 e_0，用户每次使用数据都会使用新的密钥进行加密，在用户短暂的 2 次使用间隔中破解解密密钥是极为困难的。因此，可以认为用户数据存储在块数据服务器上是安全的。

4 性能测试

本文采用由美国国家航空航天局和 Rackspace 合作研发并发起的以 Apache 许可证授权的自由软件和开放源代码项目 OpenStack[20]来进行性能测试，OpenStack 是一个免费开源的软件平台，通常被部署为云基础架构服务（IaaS, infrastructure as a service），OpenStack 具有部署简单、功能丰富、拓展性强等优点。

测试环境：由于本文只进行了数据传输存储测试，并没有高强度的多用户测试，因此，实验设备服务器端和客户端均使用实验室普通电脑。服务器硬件配置为戴尔 OptiPlex 3020 Mini Tower 台式机，处理器为英特尔 Core i5-4590 @ 3.30GHz 四核，内存 8 GB（三星 DDR3 1 600 MHz/海力士 DDR3 1 600 MHz），主硬盘为影驰 GX0128ML106-P （128 GB 固态硬盘），加希捷 ST500DM002-1BD142（500 GB 机械硬盘），网卡为瑞昱 RTL8168/8111/8112 Gigabit Ethernet Controller 千兆网卡。客户端硬件配置与服务器端配置相同。部署 Openstack 的 Linux 系统为 CentOS 6.5，Openstack 版本为 juno，通过在服务器端负责 cinder 块存储的物理主机上部署 Daoli 系统来进行对比得出性能损耗。测试的数据大小从 2~1 024 MB 不等，其中“origin”表示不使用 SSL 安全连接且不部署 Daoli 时的性能；“with SSL”表示使用 SSL 安全连接时的性能；“With SSL and Daoli”表示使用 SSL 安全连接并在服务器端部署 Daoli 后的性能。为了使实验数据更加直观，性能指标 *PI* 定义为 $\frac{\text{CPU使用率}}{\text{传输速率}}$，由于直接使用 $\frac{\text{CPU使用率}}{\text{传输速率}}$ 得出的值过小，不便于对比，因此将 CPU 使用率的“%”去掉，使用数字进行比值。CPU 使用率越高，传输速度越小，说明此时性能最差，反之，CPU 使用率越低，传输速度越大，说明此时性能最优。

表 1~表 3 分别表现了服务器端在原始、加入 SSL（with SSL）、加入 SSL 和 Daoli（with SSL and Daoli）保护这 3 种状态下的 CPU 使用率、传输速度以及 *PI*。通过对比本文可以发现，随着文件大小的增长，服务器端 CPU 使用率是线性增长的，传输速度在上升达到某一个值之后也趋于平缓，并且 SSL 安全连接在小文件处带来的时间开销比例更大一些，而 Daoli 则对传输速度影响不大。

表 1　服务器传输文件时性能数据

数据大小/Mbit	CPU 使用率	传输速率/(MB·s^{-1})	*PI*
2	3.46%	1.26	2.74
8	6.69%	4.78	1.39
32	11.65%	8.77	1.32
128	16.85%	9.94	1.69
512	20.68%	10.35	1.99
1 024	22.39%	10.75	2.08

表 2　服务器使用 SSL 连接传输文件时性能数据

数据大小/Mbit	CPU 使用率	传输速率/(MB·s^{-1})	*PI*
2	6.67%	0.47	14.19
8	14.59%	2.59	5.61
32	31.29%	3.64	8.59
128	37.24%	4.16	8.93
512	43.21%	4.34	9.93
1 024	58.87%	4.54	12.95

表 3　服务器使用 SSL 连接和 Daoli 保护传输文件时性能数据

数据大小/Mbit	CPU 使用率	传输速率/(MB·s^{-1})	*PI*
2	8.44%	0.42	20.11
8	18.08%	2.57	7.02
32	38.09%	3.57	10.66
128	44.44%	4.16	10.66
512	50.56%	4.31	11.73
1 024	70.17%	4.50	15.57

从图7可以得知，在服务器端引入SSL安全连接相较于原始情况来说带来了3~5倍的性能损失，而Daoli保护相较于引入SSL安全则带来了1.18~1.41倍的性能损失。

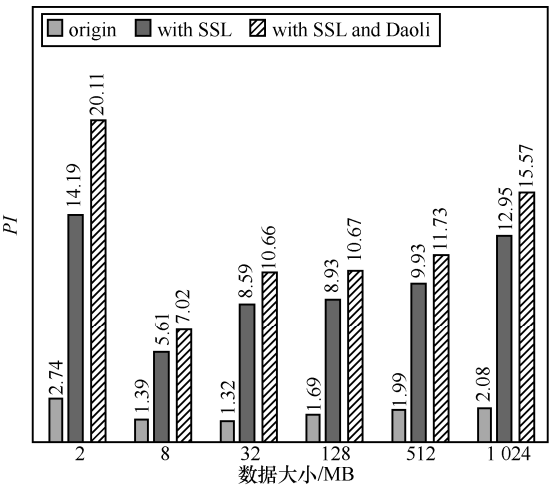


图7　服务器性能数据

表4和表5分别表现了客户端在原始、加入SSL安全连接这2种状态下的CPU使用率、传输速度以及*PI*。通过对比可以发现，与服务器端类似，随着文件大小的增长，客户端CPU使用率也是线性增长的，传输速率在上升到某一个值之后也趋于平缓，并且SSL安全连接在小文件处带来的时间开销比例更大一些。

表4　客户端传输文件时性能数据

数据大小/MB	CPU使用率	传输速率/(MB·s^{-1})	*PI*
2	4.34%	1.26	3.44
8	6.65%	4.78	1.39
32	12.28%	8.77	1.40
128	21.58%	9.94	2.17
512	24.57%	10.35	2.37
1 024	27.65%	10.75	2.57

表5　客户端使用SSL连接传输文件时性能数据

数据大小/MB	CPU使用率	传输速率/(MB·s^{-1})	*PI*
2	6.87%	0.47	14.61
8	11.58%	2.59	4.46
32	26.23%	3.64	7.21
128	31.98%	4.16	7.67
512	36.63%	4.34	8.42
1 024	49.68%	4.54	10.92

从图8可以看出，在客户端引入SSL安全连接相较于原始情况带来了3.21~5.14倍的性能损失。从性能测试的结果可以看出，Daoli保护引入的性能损失远小于使用SSL安全连接引入的性能损失，并较文献[9]的测试结果性能损失降低了许多。

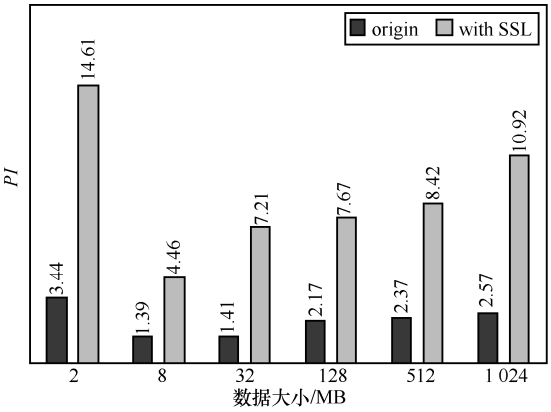


图8　客户端性能数据

5　相关工作

本文与相关研究工作的对比如表6所示。从云存储的发展历程来看，出现较早并被广泛使用的分布式文件系统主要有GFS[11]和HDFS[12]，GFS只有单一文件存储功能，HDFS虽然具有安全模式，但是进入安全模式后不允许客户端进行任何修改文件的操作，包括上传、删除文件和重命名等。

CFS[21]是最早的加密文件系统之一，在CFS的基础上研究开发了Cepheus[22]，Cepheus提出利用可信第三方来进行用户密钥管理，并引入了锁盒子机制用于用户分组管理。在Cepheus的思想上，衍生了更适合网络存储的文件加密系统Plutus[23]，该系统将文件基于其共享属性的相似性进行分组，引入锁盒子机制由客户端负责密钥分发与管理，但是，由于其密钥数随着文件分组数的增加会不断增加，系统规模很难扩展。Corslet[24]是一个栈式文件系统，可以直接架在已有的云存储系统之上，可提供端到端的数据私密性、完整性保护及访问控制等。但是，以上安全云存储系统其实已经将存储结构分为客户端、服务器和云存储服务提供商3个部分，若用户不能完全信任云存储服务提供商，那么对于第三方实现安全服务的服务器依然心存顾虑，且云平台强大的计算能力并未得到使用。

鉴于云环境虚拟化的特点，衍生出着眼于对虚拟机监视器（hypervisor）可信增强的系统，如

表 6　　云数据安全存储方案安全性对比

方案	技术特点	加密机制	加密位置	传输安全	内存安全	外存安全	不足
文献[9]	将云端操作系统隔离，由虚拟机监控系统完成加密	未提及	云端	部分解决	部分解决	部分解决	数据对分布式文件系统是明文，仍可能泄露给云服务提供商
文献[23]	将文件基于共享属性相似性分组	公私钥	客户端	解决	解决	解决	密钥随文件数线性增加，系统规模难以扩展
文献[24]	引入第三方实现密钥管理及加密	对称加密	第三方	部分解决	未解决	部分解决	增加第三方来完成加密工作，加密强度低
文献[25]	数据隐私性保护及彻底销毁	对称加密+公私钥	客户端	部分解决	解决	解决	只侧重于数据销毁，数据加密工作仍由客户端完成
文献[26~28]	加密算法支持密文查询	可搜索加密	客户端	解决	解决	解决	效率较低，查询语句不够灵活，实用性较差
本文	将云端操作系统与分布式文件系统隔离，在封闭计算环境中完成加密	改进的公私钥	云端	解决	解决	解决	暂不支持密文检索

Terra[15]，通过可信 hypervisor 实现对虚拟机中的应用程序进行度量。但 Terra 只能对已开启的虚拟机进行保护，且度量精度较粗，只能以虚拟机镜像文件为单位进行度量，且未考虑对通过完整性度量后的可执行程序进行保护。Overshadow[16]也是通过增强 hypervisor 可信来为虚拟机中的指定程序提供私密的运行空间，使用 Overshadow 虽然不用对操作系统以及应用程序做任何修改，但是其性能损失相对来说也会大一些。Dissolver[25]通过 Hypervisor 可信技术来实现云端数据隐私性保护和彻底销毁，但是数据加密过程需要在客户端完成。本文方案中的 Daoli 通过 TPM 对 CloudSec-V 进行完整性保护，以 CloudSec-V 作为可信基，由 CloudSec-V 管理 hypervisor 的安全，来解决 Terra 无法将可信延伸至未开启的 VM 的问题，并通过对操作系统的少量修改实现比 Overshadow 更高的效率。

在云端的数据加密后查询变得困难，因此一些研究机构提出了支持查询的数据加密方法。文献[26]提出了云密文数据的模糊查询方法，不需要对输入的关键字严格匹配，但是其仅支持布尔型查询，难以实现排名排序等操作。文献[27,28]则提出了支持排名查询的 SE 机制，但是仅支持单个关键字查询且其相关度量化模型的正确性还有待验证，目前，SE 机制的效率还比较低，且查询语句不够灵活，数据明文中的语义不能准确转化为查询条件，实用性还有些欠缺。本文的加密算法基于成熟的 RSA 公钥加密算法，更注重于安全性和高效性，相信随着 SE 机制的逐步完善，越来越多的云存储系统会将 SE 机制加入进来。

6　结束语

本文描述了一般分布式文件系统的特征，并结合文献[9]的测试结果，提出了一种安全的在云服务提供商处加密用户数据的云存储方案。为了保障数据在网络中传输的安全性使用 SSL 协议来传输数据，数据传输至云端后使用 Daoli 构造封闭计算环境来对数据进行加密再存储至分布式文件系统，保障用户数据在云端加密的安全性，改进 RSA 算法使其每次均可实现密钥变化来加强在分布式文件系统块数据服务器上存储数据的安全性。

但由于时间有限，本文并未进行多用户高强度下对服务器性能影响的测试，并且通过实验可以发现性能损耗低值并非在最小文件或最大文件处，而是处于之间的某个值，因此，在将来进行更为详细的测试之后，将分块功能在客户端就完成，提高传输效率。

参考文献：

[1] MELL P, GRANCE T. The NIST definition of cloud computing[R]. National Institute of Standards and Technology(NIST), Washington, USA：Technical Report Special Publication, 2011: 800-145.

[2] 李晖, 孙文海, 李凤华, 等. 公共云存储服务数据安全及隐私保护技术综述[J]. 计算机研究与发展,2014,51(7):1397-1409.
LI H, SUN W H, LI F H, et al. Secure and privacy-preserving data storage storage service in public cloud[J]. Journal of Computer Research and Development, 2014, 51(7):1397-1409.

[3] 傅颖勋, 罗圣美, 舒继武. 安全云存储系统与关键技术综述[J].计算机研究与发展, 2013, 50(1):136-145.
FU Y X, LUO S M,SHU J W. Survey of Secure Cloud Storage System and Key Technologies[J]. Journal of Computer Research and Development ,2013, 50(1):136-145.

[4] 张宇. Verizon 报告称两成数据泄露皆因“杂项错误”[J]. 计算机与网络, 2016,42(8):52.

ZHANG Y. Verizon report shows fifth of breaches caused by “miscellaneous errors”[J]. Computer and Network Journal, 2016,42(8):52.

[5] TAKABI H, JOSHI J B D, AHN G. Security and privacy challenges in cloud computing environments[J]. IEEE Security & Privacy, 2010, 8(6): 24-31.

[6] 吕志泉, 张敏, 冯登国. 云存储密文访问控制方案[J]. 计算机科学与探索, 2011, 5(9): 835-844.
LV Z Q, ZHANG M, FENG D G. Cryptographic access control scheme for cloud storage[J]. Journal of Frontiers of Computer Science and Technology, 2011, 5(9):835-844.

[7] 洪澄, 张敏, 冯登国. 面向云存储的高效动态密文访问控制方法[J]. 通信学报, 2011, 32(7): 123-132.
HONG C, ZHANG M, FENG D G. Achieving efficient dynamic cryptographic access control in cloud storage[J]. Journal on Communications, 2011, 32(7): 123-132.

[8] 洪澄, 张敏, 冯登国. AB-ACCS：一种云存储密文访问控制方法[J]. 计算机研究与发展,2010,47(z1):259-365.
HONG C, ZHANG M, FENG D G. AB-ACCS: A cryptographic access control scheme for cloud storage[J]. Journal of Computer Research and Development,2010,47(z1):259-365.

[9] 侯清铧, 武永卫, 郑纬民. 一种保护云存储平台上用户数据私密性的方法[J]. 计算机研究与发展, 2011, 48(7): 1146-1154.
HOU Q H, WU Y W, ZHEN W M. A method on protection of user data privacy in cloud storage platform[J]. Journal of Computer Research and Development, 2011, 48(7): 1146-1154.

[10] EKASIT K. Improving the communication performance of distributed animation rendering using BitTorrent file system[J]. Journal of Systems & Software,2014,97:178-191.

[11] GHEMAWAT S, GOBIOFF H, LEUNG S T. The google file system[C]//Proc of the 19th ACM Symp on Operating Systems Principles.New York : ACM , 2003: 29-43.

[12] 何高峰. 基于 HDFS 的多租户小文件存储系统的研究与设计[D]. 杭州: 浙江大学, 2016.
HE G F. Research and design of multi-tenant small file storage system based on HDFS[D]. Hangzhou: Zhejiang University, 2016.

[13] TAL G, MENDAL R, DAN B, A broader vision for trusted computing[C]//9th Workshop on Hot Topics in Operating System,2003.

[14] EMMETT W, JUNGHWAN R, KRSTE A. Mondrix: memory isolation for linux using mondriaan memory protection[C]//The 20th ACM Symposium on OS Principles. Brighton, UK, 2005.

[15] TAL G, BEN P, JIM C, et al. Terra: a virtual machine-based platform for trusted computing[J]. ACM Sigops Operating Systems Review, 2003, 37(5):193-206.

[16] CHEN X X, TAL G, LEWIS E C, et al. Overshadow: a virtualization-based approach to retrofitting protection in commodity operating systems[J]. ACM Sigplan Notices,2008,43(3): 2-13.

[17] 毛文波. 可信云安全的原理与应用[C]//第三届中国云计算大会, 2011.
MAO W B. Trusted and secure cloud principle and practice[C]//3rd China Cloud Computing Conference,2011.

[18] DUA R, RAJA A R, KAKADIA D. Virtualization vs containerization to support PaaS[J]. IEEE International Conference on Cloud Engineering, 2014, 41: 610-614.

[19] 佘冉君. SSL 安全研究及实现[D]. 西安: 西安电子科技大学, 2015.
SHE R J. The SSL security research and implementation[D]. Xi’an: Xidian University, 2015.

[20] 王彬. 基于 OpenStack 的云平台管理系统的设计与实现[D]. 北京: 北京交通大学, 2016.
WANG B. Design and implementation of cloud platform management system based on OpenStack[D]. Beijing Jiaotong University. 2016.

[21] BLAZE M. A cryptographic file system for UNIX[C]//The 1st ACM Conf on Communications and Computing Security.1993: 9-16.

[22] RIVEST R L, SMITH A C, FU K E. Group sharing and random access in cryptographic storage file systems[M]. Massachusetts Institute of Technology. 1999.

[23] KALLAHALLA M, RIEDEL E, SWAMINATHAN R, et al. Plutus: scalable secure file sharing on untrusted storage[C]//The 2nd Conf on File and Storage Technologies. Berkley:USENIX Association, 2003.

[24] 薛矛, 薛巍, 舒继武, 等. 一种云存储环境下的安全存储系统[J]. 计算机学报, 2015, 38(5): 987-998.
XUE M, XUE W, SHU J W, et al. A secure storage system over cloud storage environment[J]. Chinese Journal of Computers, 2015, 38(5): 987-998.

[25] 张逢喆, 陈进, 陈海波. 云计算中的数据隐私性保护与自我销毁[J]. 计算机研究与发展, 2011, 48(7): 1155-1167.
ZHANG F Z, CHEN J, CHEN H B, et al. Lifetime privacy and self-destruction of data in the cloud[J]. Journal of Computer Research and Development, 2011, 48(7): 1155-1167.

[26] LI J, WANG Q, WANG C, et al. Fuzzy keyword search over encrypted data in cloud computing[C]//The INFOCOM 2010 Mini-Conference. San Diego. 2010: 1-5.

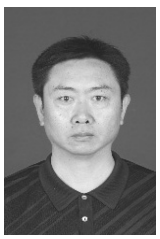
[27] WANG Q, CAO N, LI J, et al. Secure ranked keyword search over encrypted cloud data[C]//The 30th International Conference on Distributed Computing Systems. Genoa, Italy, 2010: 253-262.

[28] WANG C, CAO N, REN K, et al. Enabling secure and efficient ranked keyword search over outsourced cloud data[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(8): 1467-1479.

作者简介：

杜瑞忠（1975-），男，河北献县人，博士，河北大学教授、硕士生导师，主要研究方向为可信计算与信息安全等。

王少泫（1990-），男，河北涉县人，河北大学硕士生，主要研究方向为可信计算与信息安全等。



田俊峰（1975-），男，河北蠡县人，博士，河北大学教授、博士生导师，主要研究方向为分布计算、可信计算与信息安全。