一种大数据平台敏感数据安全共享的框架

董新华,李瑞轩,何亨,周湾湾,薛正元,王聪

华中科技大学计算机科学与技术学院,武汉 430074

摘要　大数据平台存储了海量的用户敏感数据,这些敏感数据的共享有助于企业降低为用户提供个性化服务的成本,实现数据增值,而数据的安全共享是一个亟待解决的问题。通过分析敏感数据安全现状,提出了一个大数据平台敏感数据安全共享系统框架,包括数据平台上敏感数据的安全提交、存储、使用和销毁;研究了基于密文异构转化的代理重加密算法和基于虚拟机监控器的用户进程保护方法等关键技术,为系统功能的实现提供了支撑。该框架能够保护用户敏感数据的安全性,有效实现这些数据的安全共享,同时使数据拥有者完全掌握自身数据的控制权,从而有利于营造现代互联网信息安全的良好环境。

关键词　安全共享;敏感数据;代理重加密;进程空间保护;大数据

中图分类号　TP309　　**文献标志码**　A　　**doi**　10.3981/j.issn.1000-7857.2014.34.006

A Framework for Secure Sharing of Sensitive Data on Big Data Platform

DONG Xinhua, LI Ruixuan, HE Heng, ZHOU Wanwan, XUE Zhengyuan, WANG Cong

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract　Vast amounts of users´ sensitive data are stored on the big data platform. The sharing of sensitive data will help enterprises to reduce the cost of providing users with personalized service, and achieve value-added services of data. However, the secure sharing of data is an urgent problem. By analyzing the present security situation of sensitive data, this paper proposes a framework for secure sharing of those data on big data platform, including security submission, storage, use and destruction of sensitive data on the semi-trusted big data sharing platform. Relevant key technologies were studied, such as the proxy re-encryption algorithm based on heterogeneous cipher-text transformation and user process protection methods based on the virtual machine monitor, which provides support for the realization of system functions. The framework well protects the security of users´ sensitive data, and shares these data effectively and safely. At the same time, the data owners have complete control of their own data, which is conducive to foster a sound environment for modern Internet information security.

Keywords　secure sharing; sensitive data; proxy re-encryption; process space protection; big data

随着信息数据化步伐的加快,结构化、半结构化和非结构化的海量数据在快速产生。通过对这些大数据的收集、整理、分析和挖掘,又能得到海量的个人用户敏感数据。这些数据除了满足本企业自身需求外,也可以存储在大数据平台上为其他企业提供租用服务,实现数据资源共享和数据资产化。因传统云存储只是被动地存储明文数据或加密过的数据,并不会对密文数据进行计算操作,可以认为这些数据是“死”的数据。而大数据平台是数据(包括敏感数据)交换的场所,可提供海量数据的存储和计算服务。计算服务主要是指参与方利用平台对数据进行操作(如密文数据转换,功能加密等),即可以把“死”数据盘活。为了展示大数据平台上敏感数据的流动过程,一个应用敏感数据的实例如图1所示。

收稿日期:2014-09-25;修回日期:2014-11-03

基金项目:国家自然科学基金项目(61300222,61173170,60873225);华中科技大学自主创新基金项目(2012TS052,2012TS053,2013QN120, CXY13Q019)

作者简介:董新华,博士研究生,研究方向为云计算和大数据管理、信息检索,电子信箱: xhdong@hust.edu.cn;李瑞轩(通信作者),教授,研究方向为系统安全、分布式计算、数据管理,电子信箱: rxli@hust.edu.cn

引用格式:董新华, 李瑞轩, 何亨, 等. 一种大数据平台敏感数据安全共享的框架[J]. 科技导报, 2014, 32(34): 47-52.

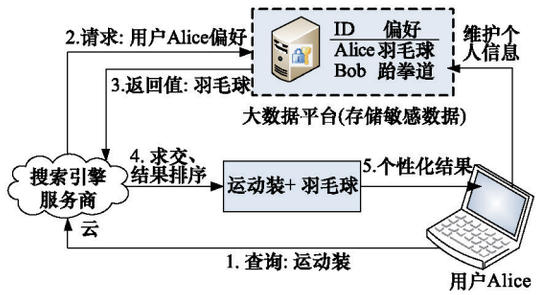

图1 敏感数据的应用实例(用户偏好)

Fig. 1 Application of sensitive data (user´s preferences)

图1中将用户偏好作为敏感数据。当用户Alice提交一个查询“运动装”时,搜索服务商首先从大数据平台上查找Alice的个人偏好,如果大数据平台上收集并共享了该用户的个人偏好信息“羽毛球”,则搜索引擎会返回“运动装+羽毛球”这样的个性化结果给Alice。当Alice看到自己喜欢的羽毛球运动装,则可能会产生一次愉快的购物过程。借助这种方式,数据参与各方(收集数据并共享的企业,搜索服务商,服务平台,个人用户)能实现多赢。然而,敏感数据在多用户和多个数据平台流转的过程中,在给企业带来资产增值的同时,由于互联网非安全因素的困扰及敏感数据易泄漏的特殊性,使得敏感数据的共享还存在一些安全问题。

从数据产生到消亡的全生命周期的角度,敏感数据的安全共享主要存在4个方面的安全问题:1) 敏感数据从数据拥有者本地服务器提交到大数据平台上的数据提交安全问题;2) 大数据平台上敏感数据的计算和存储安全问题;3) 云平台上的敏感数据提供给其他企业使用过程中的数据使用安全问题;4) 数据最终的安全销毁问题。为了解决这些安全问题,国内外一些研究机构和学者对敏感数据安全共享技术进行了一些积极的探索研究,主要有云存储加密、访问控制、可信计算和数据安全销毁等技术。

对云存储数据加密和访问控制,在加密技术方面,基于属性的加密算法(attribute-based encryption,ABE)包括密钥策略的KP-ABE[1]和密文策略的CP-ABE[2]等。ABE将解密规则蕴含在加密算法中,可以免去密文访问控制中频繁出现的密钥分发代价。但当访问控制策略发生动态变更时,就要求对数据进行重加密。Li等[3]提出了基于代理重加密PRE(proxy re-encryption)方法,即一个拥有代理重加密密钥的半可信代理[4]可以将一份密文转换为该重加密密钥对应的密文,而这个代理者不能获得明文,而且由代理重加密密钥无法计算出授权双方中任何一方的解密密钥。Centry等[5]提出了一种完全同态加密(fully homomorphic encryption)机制,它允许人们对密文进行特定的代数运算得到仍然是加密的结果。但目前完全同态加密方案计算量巨大,在现有计算技术条件下难以实现。在密文检索方面,文献[6]~[8]针对云中数据隐私保护展开研究,分别提出了在云中进行密文检索的解决方案。在访问控制方面,洪澄等[9]提出一种新的访问控制方法AB-ACCS,其基于密文属性的加密算法为用户私钥设置属性,为数据密文设置属性条件,通过私钥属性和密文属性的匹配关系确定解密能力。分布式数据流向访问控制[10](distributed systems with information flow control,DIFC)基于一组简单的数据跟踪规则,用标签跟踪数据。对于隐私数据,DIFC允许不被信任的软件使用隐私数据,但由受信任的代码控制是否透露该隐私数据。Zhang等[11]针对云中大量用户细粒度访问控制的复杂性,提出一种安全有效的撤销方案。即通过改进CP-ABE算法,去构建一种细粒度访问控制方法,该方法基于Shamir秘密共享理论实现用户撤销,同时实现了数据拥有者和云服务商开销最小化。利用单点登录,任何一个授权用户都可以通过标准的公用应用接口登录云存储系统,享受各种云存储服务。

对可信计算和进程保护技术,国际可信计算组织TCG在现有体系结构上引入硬件安全芯片TPM,利用TPM的安全特性来保证通用计算平台的可信。在学术界,主要研究思路是首先基于安全芯片建立终端平台信任,然后通过远程证明建立平台间的信任,最后将信任延伸到网络。建立终端信任平台的主要技术手段是完整性度量。虚拟平台度量技术的研究成果包括HIMA[12]和HyperSentry度量架构[13]。HIMA利用虚拟平台的隔离特性,通过对虚拟机内存的监控实现对虚拟机的完整性度量。而HyperSentry采用硬件机制,在Hypervisor无法感知的情况下对其进行度量。TCG组织于2005年发布了可信网络连接TNC(trusted network connection)架构规范1.0版本[14],其特点在于将终端完整性作为网络接入控制的判定之一。中国学者基于TNC架构也开展了可信网络连接的研究工作[15]。冯登国等[16]从构建可信终端的信任入手,建立了基于信任度的信任模型,给出了基于信息流的动态信任链构建方法。张逢喆等[17]提出一种透明、向后兼容的方法去保护虚拟基础设施上客户虚拟机的隐私和完整性。基于Xen虚拟机监控器和CHAOS系统[18~21],实现了Dissolver原型系统,Dissolver保证了明文形式的用户数据只存在于私密运行空间中,用户密钥只存在于虚拟机监控器的内存空间,在用户指定的时间点,内存中的数据以及用户密钥将被强制销毁。

对数据销毁,Wang等[22]针对电子数据提出了安全的自销毁方案。Zeng等[23]为保护隐私数据提出了一种改进的数据自销毁机制Selfvanish。董亮等[24]针对紧急情况下防止敏感信息泄漏的问题,研究了一种实时触发的敏感数据安全销毁系统方案。秦军等[25]针对开源云计算存储系统HDFS中的数据不能彻底销毁从而可能导致数据泄露的问题,设计了HDFS的多安全级数据销毁机制。张逢喆等[26]提出了用户数据全生命周期的隐私性管理以及强制性数据销毁协议,为存储在云端的用户数据提供了控制机制。

现有技术从不同角度,部分解决了数据的共享和隐私保护问题,但没有考虑数据全生命周期中整个过程的安全。本文通过分析共享敏感数据全生命周期的安全问题,首先构建

大数据平台敏感数据安全共享系统模型,然后利用代理重加密技术保障大数据平台上敏感数据的存储安全,利用基于虚拟机监控器的用户进程保护方法,保障敏感数据共享后的使用安全性。最后借助安全插件和数据自销毁机制,解除用户对个人敏感数据泄漏的担忧。

1　大数据平台敏感数据安全共享系统框架

在大数据平台半可信情况下,若企业对外有偿共享敏感数据并发布到大数据平台之上,必须要有安全保障的机制。从发展的角度,服务提供商可把使用敏感数据服务的程序运行在公有云平台上。处于不同信任域的多参与方共享和使用数据的场景中,敏感数据会经多次流转,因此需要构建敏感数据全生命周期的安全通道,主要包括安全提交、安全存储、安全使用和安全销毁4个方面的安全问题。大数据平台敏感数据安全共享系统框架如图2所示。

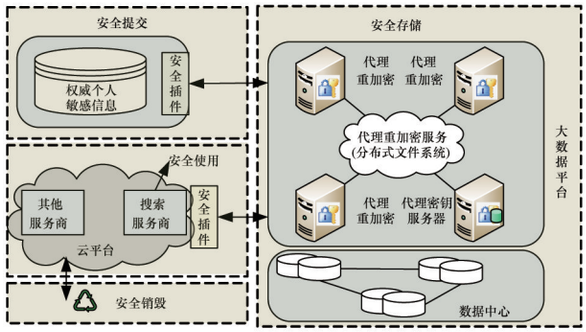

图2　大数据平台敏感数据安全共享系统架构

Fig. 2　A systematic framework of secure sharing of sensitive data on big data platform

在大数据平台半可信的情况下,通用的做法是把数据加密后再上传到大数据平台上存储,以保证数据提交的安全。安全插件能提供一些操作(如加密,解密,授权等),云平台上的服务商(如搜索引擎)使用大数据平台中的数据,需要下载安全插件,以保证数据的安全。

为保证数据存储安全,本文设计了密文异构转化的代理重加密算法H-PRE(heterogeneous proxy re-encryption),该算法支持IBE(identity-based encryption)到PKE(public-key encryption)密文的异构转化,并能兼容传统的密码体制。其目的是将数据拥有者上传的数据密文转化为数据使用者能够用他自己的私钥解密的密文。

在云不可信的情况下,解密后的明文会泄露用户的隐私信息,因此需要采用基于虚拟机监控器的用户进程保护技术,通过可信的虚拟机监控器层,跳过客户操作系统直接向用户进程提供数据保护。虚拟机监控器上的密钥管理模块用作存储用户注册新程序组的公钥。在程序运行时,主程序文件末尾的对称密钥会由密钥管理模块动态解密。所有的程序公钥和对称密钥都存放在虚拟机监控器的内存中。

云存储的归档、复制及备份机制导致多份冗余存在,要彻底删除用户的个人隐私数据,需要采用一种适用于云环境下的数据销毁安全方案。从高安全性出发,还需要设计一种基于租期的机制可控地彻底销毁私密数据和密钥,保证租期到期后,在云端任何地方都不再存在明文的用户数据和密钥。

该框架的基本流程是:1)由拥有海量个人用户敏感信息的企业,通过本地安全插件,预先设置好需要共享该敏感信息的服务商,随后把相应数据加密提交并存储到大数据平台;2)在大数据平台上对提交的数据进行代理重加密(转化)操作;3)云平台上使用该敏感信息的服务商借助安全插件下载并解密相应数据于进程私密空间中,敏感数据在进程私密空间中运行;4)对于使用过的还暂存于云上的数据,通过安全机制销毁。总之,该框架能够保护用户敏感数据的安全性,有效实现这些数据的安全共享,同时使数据拥有者完全掌握自身数据的控制权。本文重点研究基于密文异构转化的代理重加密算法和基于虚拟机监控器的用户进程保护方法。

2　基于代理重加密的数据安全提交和存储

2.1　基于密文异构转化的代理重加密算法

密文异构转化的代理重加密(H-PRE)涉及3类算法:传统基于身份的加密算法(主要包含$Setup_{IBE}$,$KeyGen_{IBE}$,Enc_{IBE},Dec_{IBE}函数);重加密(主要包含$KeyGen_{RE}$,$ReEnc$,$ReDec$函数);传统的公钥密码算法(主要包含$KeyGen_{PKE}$,Enc_{PKE},Dec_{PKE}函数)。H-PRE的基本过程是,企业所拥有的个人信息数据先通过本地安全插件进行加密处理,而后上传到大数据平台,经代理重加密服务后,转换为指定用户能够解密的密文。若搜索引擎服务商是被指定的用户,则可以通过自己的私钥解密,得到所需要的明文。基于密文异构转化的代理重加密算法实现按以下步骤进行。

1)$Setup_{IBE}(k)$:输入安全参数k,随机产生一个主安全参数mk,计算出系统参数集$params$。

2)$KeyGen_{IBE}(mk, params, id)$:当用户向密钥产生中心申请私钥时,密钥产生中心获取用户合法的身份信息id后,使用$params$和主安全参数mk为用户生成公私钥(pk_{id}, sk_{id})。

3)$KeyGen_{PKE}(params)$:当用户提交申请时,除了产生基于身份类型的公私钥对外,还产生一对传统公钥体制的公私钥对(pk_{id}', sk_{id}')。

4)$Enc_{IBE}(pk_{id}, sk_{id}, params, m)$:在用户加密数据时执行,数据拥有者使用自己的基于身份类型的公私钥对(pk_{id}, sk_{id}),随机数$r \in RZ_p^*$。将数据明文m加密为密文$c=(c_1,c_2)$。

5)$KeyGen_{RE}(sk_{idi}, sk_{idi}', pk_{idj}', params)$:在数据拥有者授权时执行,使用数据拥有者的基于身份类型的私钥sk_{idi}和传统类型的私钥sk_{idi}',以及代理受理者传统类型的公钥pk_{idj}',计算出从用户i到用户j的代理重加密密钥$(rk_{idi-idj})$。

6)$ReEnc(c_i, rk_{idi-idj}, params)$:在大数据平台透明地执行,此函数将用户$i$加密的原始密文重加密为用户$j$能解密的密

文,使用用户i加密的密文$c_i=(c_{i1},c_{i2})$及用户i到j的代理重加密密钥及相关系统参数作为输入,经过计算输出重加密密文$c_j=(c_{j1},c_{j2})$。

7) $Dec_{PKE}(c_j, sk_{idj}', params)$:重加密密文的解密函数,代理受理者在接收到代理服务器发来的密文$c_j=(c_{j1},c_{j2})$后,使用传统密码体制的sk_{idj}'计算出数据明文$m'=m$。

2.2 系统的敏感数据提交、存储和提取

数据拥有者在本地加密数据。数据拥有者首先使用AES对称密钥加密所提交的数据,然后再使用H-PRE代理重加密加密对称密钥,最后把产生的结果密文存储在分布式数据中心。与此同时,如果数据拥有者分享敏感数据给其他用户,数据拥有者需要在本地进行授权并产生代理重加密密钥,这些密钥存储在授权密钥服务器中,以方便大数据平台进一步的操作。

在大数据平台上,代理重加密服务器利用代理重加密密钥,对原密文进行重加密(转化),生成共享用户(数据使用者)能解密的密文。数据使用者要使用大数据平台上的数据时会向平台发送数据请求,然后到共享空间中查询是否有最新版的数据,如果有,则获取并下载该数据。该操作独立在大数据平台上完成,对用户是透明的,而且大数据平台上的计算资源相对客户端而言更强大,这样可以将代理重加密的计算开销放在大数据平台上,提高了用户的体验。代理重加密保护系统流程包括数据提交、存储和提取等子操作。

数据提交和存储(共享)操作。收到提交数据请求后,通过浏览器调用安全插件为数据拥有者提供提交数据的服务,详细步骤为:

1) 读取数据拥有者提交的数据文件,随机产生一个AES透明加密密钥SEK,并使用AES算法对文件数据进行加密;

2) 使用H-PRE代理重加密算法对SEK进行加密,并将数据密文和加密SEK的密文一起存入数据中心;

3) 数据拥有者选择要共享数据的用户(如搜索服务商);

4) 安全插件读取数据拥有者的私钥数据,并从大数据平台获取数据使用者的公钥;

5) 使用安全插件生成对应的代理重加密密钥,然后由安全插件将代理重加密密钥上传到大数据平台的授权密钥服务器中保存;

6) 在大数据平台上,通过代理重加密服务对数据进行重加密,产生重加密密文。

数据提取操作。收到提取数据请求后,通过浏览器调用安全插件为数据使用者提供提取数据服务,详细步骤为:

1) 大数据平台在重加密密钥服务器中查询该数据使用者是否已被授权,若已授权,则进入第2)步;

2) 安全插件向大数据平台发送数据下载请求,大数据平台在数据中心查找重加密密文数据;

3) 大数据平台将重加密密文推送给数据使用者的安全下载插件;

4) 数据使用者的下载插件读取用户私钥准备进行解密;

5) 数据使用者插件解密接受到的密文中SEK的密文,获取AES对称加密密钥;

6) 数据使用者使用AES对称密钥对数据密文进行解密,得到所需数据明文。

3 基于虚拟机监控器的数据安全使用

3.1 基于虚拟机监控器的用户进程保护方法

为确保程序运行中的安全,采用基于虚拟机监控器的用户进程保护方法。假定一些企业(如搜索引擎服务商)在云上租用了IaaS完成公司的业务,这些业务进程需要提取大数据平台中个人敏感数据。将企业自身拥有的敏感数据以及从大数据平台中提取数据的程序称为敏感进程,是需要保护的对象。程序运行中的保护模型如图3所示。敏感进程需要防范管理虚拟机和其下层不可靠操作系统层的威胁。租用的云上最底层的硬件采用可信平台模块(trusted platform module,TPM)方式,可保证上层的虚拟机监控器是可信的。这种情况下,租户(如搜索引擎服务商)的密钥管理机制需要和虚拟机监控器建立信任联系,就能保证在不可靠操作系统下的安全操作。

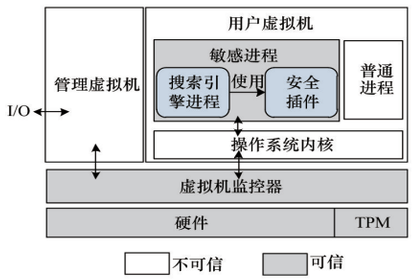

图3 程序运行中的保护模型

Fig. 3 Protection model of the sensitive process

引入虚拟化和可信计算技术,保障服务商的应用程序和安全插件运行在程序私有空间,让操作系统等外界无法对其进行干扰,以保护用户敏感数据的私密性。其安全运行过程如图4所示。

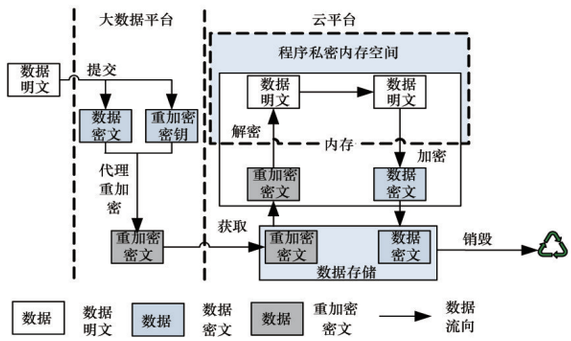

图4 敏感数据的安全运行过程

Fig. 4 Safe operation process of sensitive data

大数据平台计算出的重加密密文被提取到云平台上,云平台上的程序私密运行空间可以保证该数据在内存和硬盘两区域内的安全。首先,利用虚拟机监控器为虚拟机中的指定程序提供了一个私密运行空间,在这个运行空间中运行的程序,其内存是不能被操作系统或其他程序访问的。这种内存的隔离性保证了数据在内存中的高度私密性。然后,程序使用的数据在磁盘上的存储以密文形式,虚拟机监控器在读写数据时会分别为数据作解密和加密。因此结合上述两项保护措施,使用户的程序不管在内存中运行时,还是存储在磁盘时都能得到虚拟机监控器的保护。

3.2　系统敏感数据的安全使用

采用基于虚拟机监控器的用户进程保护技术,通过可信的虚拟机监控器层,跳过客户操作系统直接向用户进程提供数据保护。为保护云平台上交互过程中数据的提取、存储、运算等环节的安全,需要进行以下步骤完成。

3.2.1　可信环境和可信通道建立

云服务器在启动过程中需要借助可信计算技术对启动的软件进行度量。在提出的模型中,云上的应用者(搜索引擎服务商)需要保证的是虚拟机监控器的完整性,即保障虚拟机监控器的可信。云服务器在开机后需要将BIOS、GRUB和虚拟机监控器的度量值存入TPM芯片的平台配置寄存器(platform configuration register,PCR)中,在此之后向搜索引擎服务商作远程证言,以保证二者之间信任关系的建立。

搜索引擎服务商首先需要和云服务器上的虚拟机监控器建立可信通道,然后才能在云端安全地接收大数据平台传来的敏感数据。搜索引擎服务商与虚拟机监控器的远程证言和握手过程如图5所示。

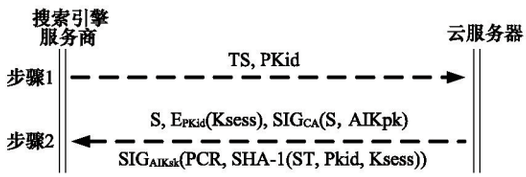

图5　与虚拟机监控器的远程证言和握手协议

Fig. 5　Remote attestation and hand shaking protocol between the SESP and the VMM

云服务器端由虚拟机监控器响应请求。第1步,搜索引擎服务商向云服务器发送一个完整性请求,包括搜索引擎服务商RSA公钥PKid和时间戳TS;第2步,虚拟机监控器生成一个会话密钥Ksess,将TS、搜索引擎服务商公钥和会话密钥Ksess一起用SHA1散列计算得到散列值。然后,虚拟机监控器调用TPM_quote指令,将这个散列值和平台配置寄存器值作为参数,得到一个用TPM私钥签名的Quote证言。另外,虚拟机监控器还将会话密钥Ksess用搜索引擎服务商公钥加密,并连同Quote证言和CA的证书一起发送给对方。当搜索引擎服务商接收这些信息后,对TS、PKid和Ksess值进行验证,如果一致,则本次通信安全。搜索引擎服务商通过与虚拟机监控器的握手,就协定了会话密钥Ksess,以后两者的通信都由该会话密钥加密。

3.2.2　数据上传和提取

云上的应用者(搜索引擎服务商)通过检索,向大数据平台上存储的个人信息库提取数据。假定云是不可信的,搜索引擎服务商在使用云之前需要将上传的可执行程序和数据加密,数据上传和提取协议如图6所示。

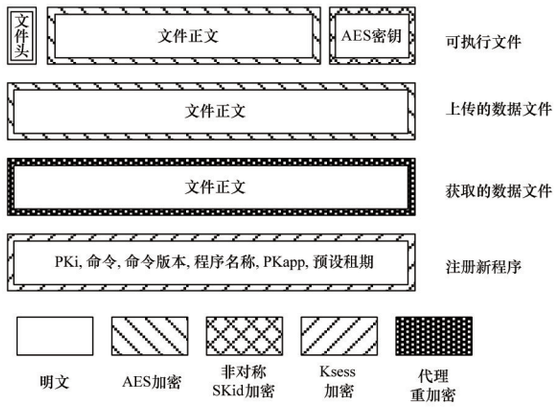

图6　数据上传和提取协议

Fig. 6　Upload and extract protocol of the data

图6中,搜索引擎服务商首先使用工具生成AES对称密钥和1对非对称密钥(PKapp,SKapp),用AES对称密钥对可执行文件和数据文件进行加密。用非对称密钥加密AES对称密钥后附加在应用程序文件的末尾。从大数据平台获取的文件仍然是代理重加密的密文,在运行时才能解密。注册新程序时需要标识其注册的命令格式,将搜索引擎服务商公钥PKid、程序注册命令、云上运行程序的名称、程序公钥PKapp以及预设租期等用会话密钥Ksess加密后发送给虚拟机监控器,然后将加密后的可执行文件和数据文件上传到云服务器中。

3.2.3　程序执行

企业在租用云中的应用程序执行过程中,动态的数据保护和加解密类似于程序运行空间保护[18~21]。如图4所示,在程序运行过程中,其私有运行空间中的内存是不能被其他进程和操作系统访问的。虚拟机监控器在操作系统与用户进程之间起到了数据交换的桥梁作用。在操作系统对用户内存空间作拷贝数据的读出和写入时,由于操作系统没有读写权限,虚拟机监控器会代替操作系统作拷贝动作。在拷贝数据到私密程序内存空间时,虚拟机监控器会用程序对应AES对称密钥对数据进行解密,这样程序能够正常对数据进行运算。反之,从私密程序内存空间拷贝数据到外部时,如在做写操作时,虚拟机监控器会首先使用AES密钥对数据进行加密。这样,存储在存储设备上的用户数据都是密文形式。

总之,在程序运行的私密空间中,来自大数据平台的重加密数据由安全插件解密,来自云上用户(搜索引擎服务商)的数据由虚拟机监控器(virtual machine monitor,VMM)解

密。程序运行完成后,产生的数据会被加密,并按基于租期的机制销毁。因而,程序运行的私密空间成为数据拥有者和数据使用者的一个平衡点,在维护两者利益的同时,还能防止敏感信息的泄露。

4 结论

针对敏感数据共享的安全问题,提出了一种大数据平台敏感数据安全共享系统框架,基于密文异构转化的代理重加密算法,确保了敏感数据提交和存储的安全。基于虚拟机监控器的用户进程保护方法确保了明文在云平台上的安全使用。该框架很好地保护了用户敏感数据的安全,同时数据拥有者有完全的数据控制权。在半可信条件下,该框架成为平衡参与各方利益的一种可行的解决方案。进一步的研究包括两个方面,一是优化代理重加密算法,提高加解密效率,二是降低参与各方之间交互的开销。

参考文献(References)

[1] Yu S, Wang C, Ren K, et al. Attribute based data sharing with attribute revocation[C]//Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. New York: ACM, 2010: 261-270.

[2] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]//Proceedings of the IEEE Symposium on Security and Privacy (S & P 2007). Piscataway, NJ: IEEE, 2007: 321-334.

[3] Li J, Zhao G, Chen X, et al. Fine-grained data access control systems with user accountability in cloud computing[C]//Proceedings of the 2nd International Conference on Cloud Computing Technology and Science (CloudCom). Piscataway, NJ: IEEE, 2010: 89-96.

[4] Wang L, Wang L, Mambo M, et al. New identity-based proxy re-encryption schemes to prevent collusion attacks[M]. Berlin: Springer, 2010: 327-346.

[5] Gentry C. A fully homomorphic encryption scheme[D]. California: Stanford University, 2009.

[6] Ananthi S, Sendil M S, Karthik S. Privacy preserving keyword search over encrypted cloud data[C]//Proceedings of the 1st Advances in Computing and Communications. Berlin: Springer, 2011: 480-487.

[7] Hu H, Xu J, Ren C, et al. Processing private queries over untrusted data cloud through privacy homomorphism[C]//Proceedings of the 27th IEEE International Conference on Data Engineering (ICDE). Piscataway, NJ: IEEE, 2011: 601-612.

[8] Cao N, Wang C, Li M, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data[J]. Parallel and Distributed Systems, 2014, 25(1): 222-233.

[9] 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(1): 259-265.
Hong Cheng, Zhang Min, Feng Dengguo. AB- ACCS: A cryptographic access control scheme for cloud storage[J]. Journal of Computer Research and Development, 2010, 47(1): 259-265.

[10] Zeldovich N, Boyd-Wickizer S, Mazieres D. Securing distributed systems with information flow control[C]//Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI). Berkeley, CA: Usenix Association, 2008, 8: 293-308.

[11] Zhang M, Lü Z, Feng D, et al. A secure and efficient revocation scheme for fine-grained access control in cloud storage[C]//Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom). Piscataway, NJ: IEEE, 2012: 545-550.

[12] Azab A M, Ning P, Sezer E C, et al. HIMA: A hypervisor-based integrity measurement agent[C]//Proceedings of the 25th Annual Computer Security Applications Conference(ACSAC). Piscataway, NJ: IEEE, 2009: 461-470.

[13] Azab A M, Ning P, Wang Z, et al. HyperSentry: enabling stealthy in-context measurement of hypervisor integrity[C]//Proceedings of the 17th ACM Conference on Computer and Communications Security. New York: ACM, 2010: 38-49.

[14] Trusted Computing Group. TNC architecture for interoperability[EB/OL]. 2005-03-03[2014-02-05]. http://www.trustedcomputinggroup.org/resources/tnc_architecture_for_interoperability_specification.

[15] 张焕国, 陈璐, 张立强. 可信网络连接研究[J]. 计算机学报, 2010, 33(4): 706-717.
Zhang Huanguo, Chen Lu, Zhang Liqiang. Research on trusted network connection[J]. Chinese Journal of Computers, 2010, 33(4): 706-717.

[16] 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 Dengguo, Qin Yu, Wang Dan, et al. Research on trusted computing technology[J]. Journal of Computer Research and Development, 2011, 48(8): 1332-1349.

[17] Zhang F, Chen J, Chen H, et al. CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization[C]//Proceedings of the 23rd ACM Symposium on Operating Systems Principles. New York: ACM, 2011: 203-216.

[18] Chen X, Garfinkel T, Lewis E C, et al. Overshadow: A virtualization-based approach to retrofitting protection in commodity operating systems [C]//Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). New York: ACM, 2008: 2-13.

[19] Yang J, Shin K G. Using hypervisor to provide data secrecy for user applications on a per-page basis[C]//Proceedings of the 4th ACM international conference on Virtual execution environments. New York: ACM, 2008: 71-80.

[20] Chen H, Zhang F, Chen C, et al. Tamper-resistant execution in an untrusted operating system using a virtual machine monitor[R]. Shanghai: Parallel Processing Institute, Fudan University, 2007.

[21] Dewan P, Durham D, Khosravi H, et al. A hypervisor-based system for protecting software runtime memory and persistent storage[C]//Proceedings of the 2008 Spring Simulation Multiconference. New York: ACM, 2008: 828-835.

[22] Wang G, Yue F, Liu Q. A secure self-destructing scheme for electronic data[J]. Journal of Computer and System Sciences, 2013, 79(2): 279-290.

[23] Zeng L, Shi Z, Xu S, et al. Safevanish: An improved data self-destruction for protecting data privacy[C]//Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom). Piscataway, NJ: IEEE, 2010: 521-528.

[24] 董亮, 庄毅, 高阳, 等. 一种实时触发的敏感数据安全销毁系统的研究[J]. 小型微型计算机系统, 2010, 31(7): 1323-1327.
Dong Liang, Zhuang Yi, Gao Yang, et al. Research on real-time trigger system for sensitive data safe destruction[J]. Journal of Chinese Computer System, 2010, 31(7): 1323-1327.

[25] 秦军, 邓谦, 张建平. HDFS 的多安全级数据销毁机制设计[J]. 计算机技术与发展, 2013, 23(3): 129-133.
Qin Jun, Deng Qian, Zhang Jianping. Design of multi-grade safety data destruction mechanism of HDFS[J]. Computer Technology and Development, 2013, 23(3): 129-133.

[26] 张逢喆, 陈进, 陈海波, 等. 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 Fengzhe, Chen Jin, Chen Haibo, et al. Lifetime privacy and self-destruction of data in the cloud[J]. Journal of Computer Research and Development, 2011, 48(7): 1155-1167.

(责任编辑 王媛媛)