计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
49(1): 210-216, 2012

一种基于完整性度量架构的数据封装方法

沈晴霓[1,2] 杜 虹[1,3] 文 汉[1,2] 卿斯汉[1,4]

[1](北京大学软件与微电子学院信息安全系 北京 102600)

[2](网络与软件安全保障教育部重点实验室(北京大学) 北京 100871)

[3](国家保密科学技术研究所 北京 100044)

[4](中国科学院软件研究所 北京 100190)

(qingnishen@ss.pku.edu.cn)

A Data Sealing Approach Based on Integrity Measurement Architecture

Shen Qingni[1,2], Du Hong[1,3], Wen Han[1,2], and Qing Sihan[1,4]

[1](*Department of Information Security, School of Software and Microelectronics, Peking University, Beijing* 102600)

[2](*Key Laboratory of Network and Software Assurance (Peking University), Ministry of Education, Beijing* 100871)

[3](*Institute of National Security Science and Technology, Beijing* 100044)

[4](*Institute of Software, Chinese Academy of Sciences, Beijing* 100190)

Abstract As an important capability of trusted computing platform, sealing can provide strong data storage security by combining data's encryption with the platform configuration, by which data can only be unsealed under specific configurations. However, sealing operation is hard to use for the complexity of modern OS, the randomness of the loading order of the booting components, the frequently changing configuration, software update and patches. IMA (integrity measurement architecture) implemented in operating system could measure the dynamic configurations and extend them to the trust chain of the whole trusted platform, and then support the data sealing. Therefore, a new approach to data sealing based on IMA is proposed here, which seals data to a relatively fixed configuration in PCR0—PCR7 (Platform Configuration Register) and then applies a list policy (black list policy or white list policy) to the measurement list (ML) in IMA for the variable configuration in PCR10 to determine whether the unseal operation can be performed. Finally, a prototype system "TPM Master" implemented in Linux is given and its performance and security analysis are both evaluated. The results show that the proposed approach could solve the issue of the PCR value varying with the OS complexity and make updating process much more flexible by the list policy, without re-sealing the original data.

Key words trusted computing; trusted platform module (TPM); integrity measurement architecture (IMA); data sealing; secure storage; platform configuration register (PCR)

摘 要 封装存储是可信计算平台的一项重要功能,它能将数据的加密存储与平台状态结合起来,提供了更强的安全存储保证.但现代操作系统结构越来越复杂,各种启动项的加载顺序也相对随机;平台配置的频繁改变、软件更新及系统补丁等都限制了封装存储的应用.而操作系统级的完整性度量架构(IMA)能将信任链扩展到整个计算平台,为封装存储提供了支持.为此,基于 IMA 提出一种新的数据

收稿日期:2011-05-18;修回日期:2011-08-02

基金项目:国家自然科学基金项目(60873238,60970135,61073156,61170282),国家科技支撑计划基金项目(2008BAH33B02)

封装方法,采用相对固定的标准状态来封装,结合易变 IMA 度量列表和结果以及经过签名的名单策略来评估平台状态,解决了操作系统复杂性带来的配置寄存器(PCR)的值不确定性和软件更新及系统补丁带来的频繁封装问题.

关键词 可信计算;可信平台模块;完整性度量架构;数据封装;安全存储

中图法分类号 TP309

为了保证数据的安全存储,TCG(Trusted Computing Group)[1-3]给出了一种基于 TPM(trusted platform module)的可信计算平台数据封装方法,将数据的加解密操作与平台的状态结合起来.TCG 封装方法具有较高的安全性,但由于其要求非常严格,在实际应用中受到很大限制[4-8].例如,现代操作系统越来越复杂,启动加载项顺序也相对随机,而封装操作所记录和验证的只是一个度量项的顺序聚集值,这样解封时刻的平台状态实际上很难在封装时刻精确描述.而且平台软件可能需要频繁更新,而每次更新都会导致状态改变,要适应这些改变,用户又不得不解封和重新封装数据,这些开销不容忽视.

针对 TCG 封装方法存在的问题,文献[4-5]提出了一种基于属性的封装方法.这种方法的主要思想是改变标准的封装流程,使用软硬件配置所具有的安全属性来代替标准方法中的平台配置,将其与待封装的数据进行绑定.这种方法能够解决标准方法中的封装失效问题,由于与数据绑定的是安全属性而不是平台配置本身,只要符合相应安全属性,即使平台配置发生变化数据仍然可以解封,这同时也避免了平台配置频繁变动所带来的解封/重新封装开销.但是这种方法也有一定的局限性,它需要引入一个第三方的更新认证权威(update certification authority, UCA)来对平台的安全属性进行认证.而实际应用中各平台配置差异很大,要找到一个能判断所有平台配置对应属性的机构是很难的,同时对基础架构的建设也提出了更高的要求.针对这个问题,文献[7]又提出了一种改进的基于安全属性的封装方法,它不需要第三方 UCA,而是让用户自己将安全属性与平台配置进行配对,并对配对值进行签名,再将这个签名值加入数据包中进行封装.当平台配置发生改变时,用户只需要自行签名新的"安全属性-平台配置"对,解封时只需提供新的签名值即可通过验证,执行解密操作.这种方法解决了 UCA 的部署问题,也减少了相应部分的开销,但是用户仍然需要确定频繁发生变化的配置与安全属性间的关系,并对其进行签名.此外,安全属性与平台配置的直接配对绑定的灵活性仍然显得不足,同时这种配对也没有解决软件环境的复杂性和加载顺序变化所带来的 PCRs(platform configuration registers)值不固定问题.

IBM 的 Sailer 等人[9-12]提出一种完整性度量架构(integrity measurement architecture, IMA),它通过修改 Linux 内核、使运行时系统可以自动对要装载的可执行内容进行完整性度量.此外,它在内核保存一张有序度量列表(measurement list, ML),TPM 保护这张内核列表的完整性而不是直接保护列表中的度量值.只要被度量的可执行内容没有改变,就能通过度量结果缓存机制避免重复运算,从而最小化给系统带来的性能损失.这种机制除了可以度量可执行文件以外还可以度量动态加载项、共享库和内核模块,而且这种方法是可扩展的,甚至允许应用程序度量他们指定的装载项.目前,IMA 已经作为操作系统内核组件实现.为此,本文基于 IMA 提出一种新的数据封装方法,使用可信计算平台中相对固定的度量结果(标准状态)来封装,并结合 IMA 度量列表及度量结果(易变状态),应用经过签名的名单策略对平台状态评估.该方法除了能够解决操作系统的复杂性带来的 *PCR* 值不确定性问题,还能通过更新名单策略的方式灵活应对平台的更新和软件的升级,而无需重新封装原有数据.

1 基于完整性度量架构的数据封装方法

TPM 具有多个 PCRs,分别用来记录不同组件的完整性信息.事实上,PCRs 不仅记录了各度量项的度量结果,而且还记录了度量项的度量顺序.考虑到平台软件不断更新和系统补丁频繁改变,本文使用平台配置中相对较为固定的标准配置来对数据进行传统的封装操作,并为封装密文附加相对易变的软件配置信息,在不降低安全性的同时增强封装的灵活性.

1.1 基本概念

为了能够区分不同平台状态信息的变化特性，并对它们采用不同的绑定方式以实现基于完整性度量架构的数据封装方法，本文给出如下定义：

标准状态：平台 PCR 状态信息中相对固定不变的标准状态(如 0～7 号 PCR).

易变状态：平台 PCR 状态信息中容易随着软件升级和系统补丁等而变化的易变状态(如 PC 终端上 IMA 产生的度量列表 ML 和默认用来记录操作系统度量项的 10 号 PCR).

名单策略：易变状态需要满足的安全策略(如包含指定度量项列表的黑名单或白名单).

1.2 基于完整性度量架构的数据封装方法

基于完整性度量架构的数据封装方法的主要思想如图 1 所示：先对数据使用标准状态进行封装，再在封装密文后附加易变状态需要满足的策略验证信息，这样即使易变状态改变，也不需要重新对数据执行开销较大的标准封装操作，而只需要更新相应的策略验证信息.

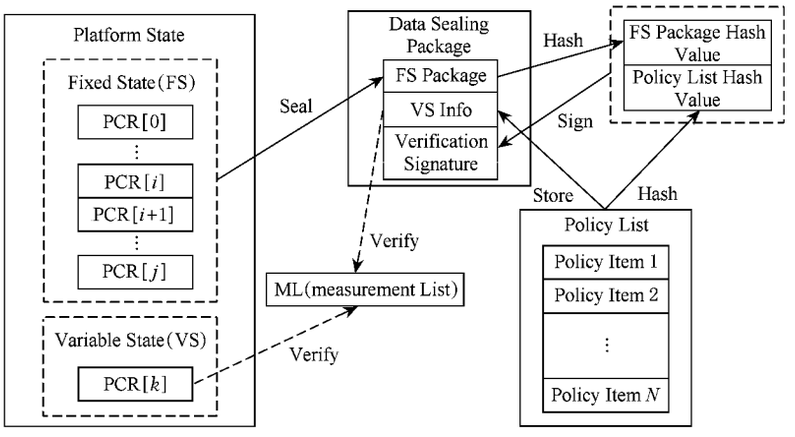


Fig. 1 IMA-Based sealing method.

图 1 基于完整性度量架构的封装方法

此外，由于 IMA 还提供一个与易变状态 PCR 值相对应的度量列表 ML，使用 PCR 值可以验证这个度量列表 ML. 因此本文不使用具体的 PCR 值来直接验证易变状态，而是以包含指定度量项列表的名单策略方式来实现这部分状态信息的验证，数据封包中只需要保存代表易变状态信息的这个策略列表，以及对标准状态封包和该策略列表的签名.

在解封时，首先验证签名是否有效，然后对平台度量列表 ML 应用指定的策略对易变状态部分进行验证，验证通过之后才提交标准封装的授权信息，让 TPM 完成标准解封. 如果软件配置发生变化，用户只需要将解封所需要满足的新策略与原封包的散列值一并重新签名即可.

这里代表易变状态信息的策略列表可以根据相应的安全需求灵活制定，如果安全级要求较高，可以使用白名单，直接将所有允许的加载项加入名单中，这样易变状态部分就只允许这些已知项；如果安全级要求较低，可以使用黑名单，解封数据时只检测已知的威胁项.

由于解封时是对度量列表应用策略，而不是传统封装中的直接比对 PCR，无论使用哪种策略，解封时的验证都不会受度量项加载顺序的影响. 此外，由于策略选择的灵活性，数据的一次封装可以对应多种安全的平台状态，这样就免去了针对多种可能的安全状态执行多次封装操作的时间和空间开销，并且能通过灵活的策略选择来满足多样的安全需求.

1.3 主要操作

为了规范封装方法，下面重点给出封装/解封(如图 2 所示)操作和策略更新操作. 其中的符号定义包括(PK, SK)表示存储公/私钥对；(SP, SS)表示签名公/私钥对；用 $Seal_{PK}(P, Auth, Config)$ 和 $Unseal_{SK}(C, Auth)$ 分别表示使用 PK、授权信息 Auth 和标准配置 $Config$ 对明文 P 进行标准封装及使用 SK 和授权信息 $Auth$ 对密文 C 进行解密；用 $Sig_{SS}(D)$ 和 $Ver_{SP}(Sig_{SS}(D))$ 来表示使用 SS 对数据 D 进行签名及使用 SP 来验证签名；用 $Hash(D)$ 来表示数据 D 的散列值.

1.3.1 封装操作

假设用户所要封装的明文数据为 P，并指定 P 只能在标准配置满足 $Config1$，且易变的软件配置度量列表符合名单策略 $Policy$ 的软硬件配置时打

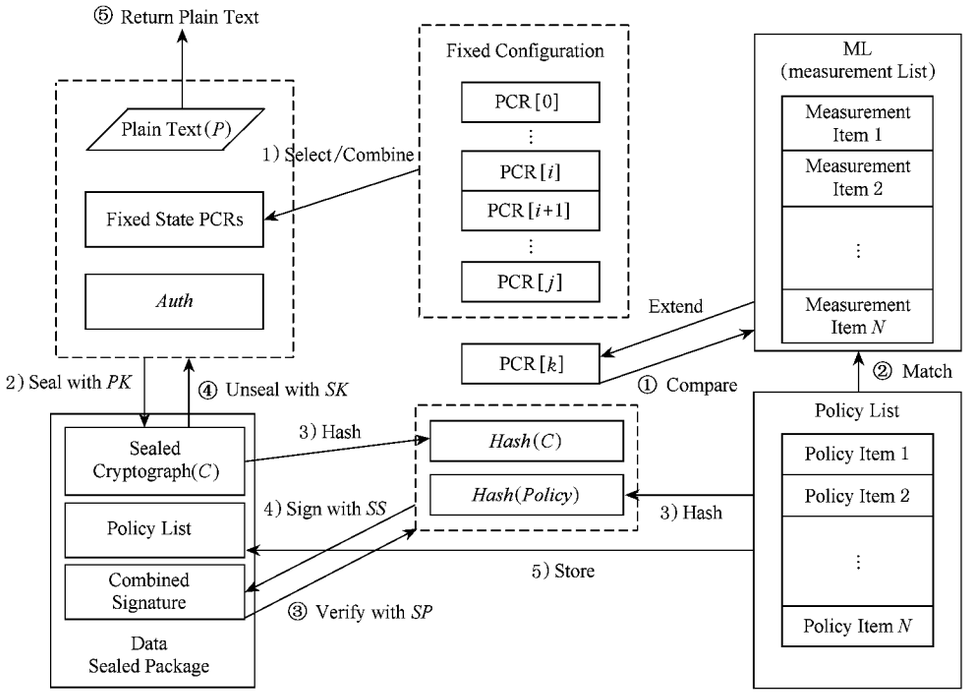


Fig. 2 Data sealing and unsealing operation process.

图 2 数据封装/解封过程

开.则对 P 进行封装操作的步骤如下.

1) 准备封装所需要的信息:用户授权信息 $Auth$、用户存储公私密钥对(PK,SK)、签名公/私钥对(SP,SS)、标准配置 $Config1$、名单策略 $Policy$;

2) 使用标准配置 $Config1$ 和用户设置授权信息 $Auth$,对 P 执行标准封装操作得到密文 $C=Seal_{PK}(P,Auth,Config1)$;

3) 计算封装产生密文 C 的散列值 $Hash(C)$以及名单策略 $Policy$ 的散列值 $Hash(Policy)$;

4) 对名单策略 $Hash(Policy)$和 $Hash(C)$进行签名:$Sig_{SS}(Hash(Policy),Hash(C))$;

5) 将封装数据包 C、名单策略 $Policy$ 与相应的签名 $Sig_{SS}(Hash(Policy),Hash(C))$一并存储.

1.3.2 解封操作

对封装的明文 P 解封的步骤如下:

① 准备要解封的密文 C, $Policy'$ 与 $Sig_{SS}(Hash(Policy'),Hash(C))$,以及密钥$(SK,PK)$;

② 取得平台当前 PCR 的 $Quote$ 值,验证 TPM 对 PCR 的签名无误;

③ 取得完整的度量列表,重新聚集生成 PCR 值与 2)中得到的进行比对;

④ 使用 SP 验证 $Ver_{SP}(Sig_{SS}(Hash(Policy'),Hash(C)))$无误,然后应用 $Policy'$策略对度量列表进行匹配,确保度量列表符合相应策略;

⑤ 验证无误后用户提交授权信息 $Auth$ 以解封封装数据包 C,返回 $P=Unseal_{SK}(C,Auth)$.

1.3.3 策略更新操作

假设平台合法更新,更新后解封密文 C 需要满足的策略为 $Policy'$,则对密文 C 更新策略 $Policy'$ 的步骤如下:

1) 提取密文 C;

2) 计算密文 C 的散列值 $Hash(C)$以及新名单策略 $Policy'$的散列值 $Hash(Policy')$;

3) 使用 SS 对 $Hash(Policy')$和 $Hash(C)$重新签名得到 $Sig_{SS}(Hash(Policy'),Hash(C))$;

4) 重新存储 C, $Policy'$ 与 $Sig_{SS}(Hash(Policy'),Hash(C))$.

2 原型实现

TPM Tools[13] 和 TPM Manger[14] 主要用于帮助用户完成 TPM 初始化及基本管理维护操作,比如状态查看、属主设置、更改授权等.为了验证第 1 节提出的数据封装方法,在现有工具功能基础之上,我们基于 Linux 实现了 TPM Master 综合应用系统[15],它扩展的功能包括:

1) 除了对 TPM 芯片进行管理,查询 TPM 芯片的各项信息外,还能够即时查询 TPM 中的 PCR

状态,并能将其导出以作为封装之用;

2) 能够直观地查看 TPM 密钥树的层次结构关系,在层次结构中自由地创建、删除和注册密钥并管理其相应授权;

3) 能够自由选择 TPM 的非对称密钥来封装随机生成的对称密钥,并使用后者来加密和解密文件,同时能使用当前 PCR 状态或之前导出的 PCR 状态来执行封装操作;

4) 能够对平台完整性状态进行评估,并能以单个度量项为粒度,把平台的完整性状态和文件加解密结合起来,并提供用户可自行设置的、灵活的名单策略.

其中的关键部分为封装/解封过程中采用的完整性评估和名单策略管理的实现方法.

2.1 完整性评估

TPM Master 以标准封装结合平台完整性和黑白名单验证的方式来实现平台状态级的可信性验证. 首先,使用度量列表 ML 来重构 PCR,并与经过 TPM 签名的当前 PCR 值比对. 如果无误则显示所有度量项,否则报告完整性信息异常. 其次,由于平台的完整性报告仅能报告平台的真实状态,并不能说明平台状态是否安全. 所以在平台的真实完整性状态报告基础之上,进一步应用签名的名单策略来进行验证. 其中黑名单策略用于记录一些可能产生安全威胁的程序特征值,当发现状态报告中存在其中程序特征值时就警告用户;白名单策略用于记录经过评估的安全程序的特征值,在安全要求较为严格的平台上,只允许运行白名单上的程序,如果发现状态报告中存在白名单中未记录的程序特征值时,就警告用户.

2.2 名单策略管理

黑白名单策略管理实际上就是对多个包含一定度量项的黑白名单列表的管理和维护,它允许对每个独立的黑白名单自由的增加和删除度量表项,新增表项的方式包括如下:

1) 从当前平台的完整性状态报告中选定度量项. 这些程序的指纹值已经经由 IMA 计算过,而且由 TPM 签名来保证真实有效,可以直接加入名单中.

2) 直接选择文件. 可以选择磁盘上任意一个文件,让 TPM Master 来计算其指纹值,并将其加入到名单中.

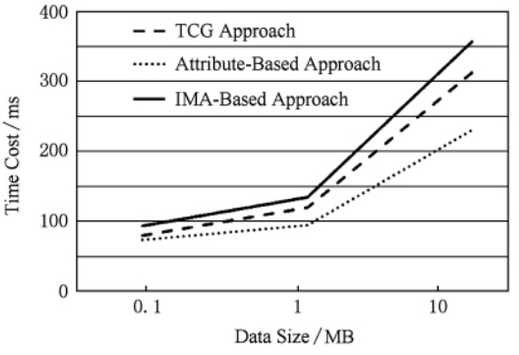
3) 直接输入程序的指纹值. 现在很多程序在发布时就已经在网站上给出了标准的指纹值,可以直接将其粘贴加入名单中.

3 效率与安全性分析

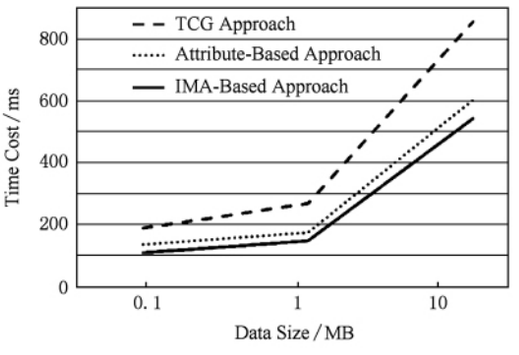
3.1 效率分析

本文实验选用 2 048 b 长度的封装密钥、256 b 长度的 AES 密钥和 CBC 密文反馈模式,分别在 Linux 动态加载自编译内核功能模块(加入策略名单)前后,对 100 KB,1 MB,10 MB 大小的文件使用 TCG 原有方法、改进的基于安全属性的方法[5]和本文方法执行封装操作,执行效率如图 3 所示,具体分析如下:

1) 第 1 次执行数据封装时,本文方法除了要使用标准状态对数据进行封装外,还要附加额外的易变状态信息,即相对于 TCG 方法,需要多执行一次散列和签名,因此性能会有所下降. 但是因为只需对一个很短的散列值进行签名,所以增加的开销较小,如图 3(a)所示,第 1 次对各文件执行封装操作(加载内核模块前)的平均开销增加仅为 12%. 而基于属性的方法比 TCG 原有方法的开销低,因为它仅需对固定安全属性值进行封装.



(a) Time cost of first sealing operation



(b) Time cost of Re-sealing upon policy update

Fig. 3 Performance comparison of sealing by old approaches and new approach.

图 3 原有方法与本文方法的封装操作的执行效率对比

2) 当平台状态发生变化时,用户只需对策略名单作相应的改动,并重新对其进行签名即可. 但是,TCG 封装方法需要先将已封装数据包进行解封,然后再重新执行一次封装. 因此,策略更新时,本文方法的执行效率比原有方法有显著提高,如图 3(b)所示,在平台因加载内核模块引起策略更新时带来的重新封装各文件的平均开销降低了约 52%. 而基于属性的方法,由于用户需要重新签名新的"安全属性-平台配置"对,并且平台配置值度量顺序不同带来了额外的验证开销,在效率提升方面低于本文方法,平均为 43%.

此外,TCG 原有封装方法和基于属性的方法针对多种可能的平台安全状态需要分别执行多次封装操作,而本文方法可以通过策略的选择让一次封装能够对应多种安全的平台状态,能够节省时间和空间上的大量开销. 本文方法尽管在解封时 TPM 需要多执行一个 Quote 操作来验证平台度量列表的完整性,但其他操作都不需要与 TPM 交互,开销可以忽略不计.

3.2 安全性分析

1) TPM 自身的安全性:本文方法无需改变 TPM 内部执行流程,TPM 内部的屏蔽存储单元和独立执行部件能够保证其作为可信根所需要具备的安全性.

2) 封装数据安全性:完整性度量架构可以保证度量列表的真实有效,而只有度量列表经过名单策略匹配通过后,用户才会提交授权信息 Auth,从而保证只有标准配置满足条件,但易变配置不满足相应策略时数据无法解封,在提高灵活性的同时又确保了高安全性.

3) 密钥安全性:本文方法提出的数据封装流程中用到的各密钥都是在 TPM 内部由其真随机数生成器生成的不可迁移密钥,不会在 TPM 外部被使用,从而保证了密钥的安全性.

4) 封装及解封安全性:加/解密、签名/验证操作都在 TPM 内部完成,授权信息的管理与平台绑定都由标准封装流程来保证,这样标准封装流程所具有的安全属性都得到了保留.

4 结 论

数据封装作为可信计算平台最为核心的功能之一,由于其操作条件过于严格而在实际应用中受到诸多限制. 本文提出了一种基于完整性度量架构 IMA 的数据封装方法,使用了相对固定的度量结果(标准状态)进行封装,并结合 IMA 的动态度量结果(易变状态),应用经过签名的名单策略对平台状态评估. 该方法在保持 TCG 标准封装方法高安全性的同时,使得数据封装能够实际应用于更加复杂与多变的可信计算平台环境中.

参 考 文 献

[1] Trusted Computing Group. TPM Main Specification Level 2 Version 1.2, Revision 103 [EB/OL]. [2011-02-19]. http://www.trustedcomputinggroup.org/resources/tpm_main_specification

[2] State Password Administration Committee in China. Functionality and interface specification of cryptographic support platform for trusted computing [EB/OL]. [2011-02-19]. http://www.oscca.gov.cn/Doc/6/News_1132.htm (in Chinese)
(中国国家密码管理局. 可信计算密码支撑平台功能与接口与规范[EB/OL]. [2011-02-19]. http://www.oscca.gov.cn/Doc/6/News_1132.htm)

[3] Sailer R, Doorn L, Ward J. The role of TPM in enterprise security, RC23363 [OL]. [2011-02-19]. http://www.oscca.gov.cn/Doc/6/News_1132.htm

[4] Poritz J, Schunter M, van Herreweghen E, et al. Property attestation scalable and privacy-friendly security assessment of peer computers, RZ 3548 [R]. New York: IBM Research, 2004

[5] Kühn U, Kursawe K, Lucks S, et al. Secure data management in trusted computing [G] //LNCS 3659: Proc of the Workshop on Cryptographic Hardware and Embedded Systems (CHES). Berlin: Springer, 2005: 324-338

[6] Kühn U, Selhorst M, Stüble C. Realizing property-based attestation and sealing with commonly available hard and software [C] //Proc of the 2007 ACM Workshop on Scalable Trusted Computing. New York: ACM, 2007: 50-57

[7] Lu Jianxin, Yang Shutang, Lu Songnian. A property-based sealed storage solution in trusted computing [J]. Information Technology, 2008, 32(1): 1-4 (in Chinese)
(陆建新, 杨树堂, 陆松年. 可信计算中一种基于属性的封装存储方案[J]. 信息技术, 2008, 32(1): 1-4)

[8] Wang Dan, Feng Dengguo, Xu Zhen. An approach to data sealing based on trusted virtualization platform [J]. Journal of Computer Research and Development, 2009, 46(8): 1325-1333 (in Chinese)

(汪丹, 冯登国, 徐震. 基于可信虚拟平台的数据封装方案[J]. 计算机研究与发展, 2009, 46(8): 1325-1333)

[9] Sailer R, Zhang X, Jaeger T, et al. Design and implementation of a TCG-based integrity measurement architecture [C] //Proc of the 13th USENIX Security Symposium. Berkeley: USENIX Association, 2004: 223-238

[10] Jaeger T, Sailer R, Shankar U. PRIMA: Policy-reduced integrity measurement architecture [C] //Proc of the 11th ACM Symp on Access Control Models and Technologies. New York: ACM, 2006: 19-28

[11] Loscocco P, Wilson P, Pendergrass J, et al. Linux kernel integrity measurement using contextual inspection [C] //Proc of the 2007 ACM Workshop on Scalable Ttrusted Computing. New York: ACM, 2007: 113-120

[12] Sourceforge. Linux integrity subsystem [EB/OL]. [2011-02-19]. http://linux-ima.sourceforge.net

[13] Sourceforge. TrouSerS/Tpm-Tools [EB/OL]. [2011-02-19]. http://trousers.sourceforge.net

[14] Sirrix. TPM Manager [EB/OL]. [2011-02-19]. http://projects.sirrix.com/trac/tpmmanager, 2009

[15] Wen Han. Design and implementation of an integrated trusted application-TPM master [D]. Bejing: Peking University, 2010 (in Chinese)
(文汉. Linux 下 TPM 可信综合应用系统的设计与实现[D]. 北京: 北京大学, 2010)

Shen Qingni, born in 1970. PhD. Associate professor and master supervisor. Her main research interests include operating system and virtualization security, trusted computing, cloud computing and cloud storage security.

Du Hong, born in 1953. Professor and PhD supervisor. His main research interests include information security.

Wen Han, born in 1984. Master and software engineer. His main research interests include operating system security and trusted computing.

Qing Sihan, born in 1939. Professor and PhD supervisor. His main research interests include secure operating system, trusted computing, cryptography algorithm, secure protocols and information system security criteria and evaluation.