可信计算增强 P2P 网络的安全性研究

李向前[1],宋 昆[2],金 刚[3]
LI Xiang-qian[1],SONG Kun[2],JIN Gang[3]

1.北京交通大学 计算机与信息技术学院,北京 100044
2.中国电子工程设计院 北京 307 信箱,北京 100840
3.华中科技大学 图像识别与人工智能研究所,武汉 430074
1.School of Computer and Information Technology,Beijing Jiaotong University,Beijing 100044,China
2.China Electronics Engineering Design Institute,Beijing 307 Mail Box,Beijing 100840,China
3.Institute for Pattern Recognition and AI,Huazhong Univ. of Sci. & Tech.,Wuhan 430074,China
E-mail:lixq@computer.njtu.edu.cn

LI Xiang-qian,SONG Kun,JIN Gang.Research of enhancement security of P2P network based on trusted computing. Computer Engineering and Applications,2007,43(17):137-139.

Abstract: A trusted agent with using Platform Configuration Register(PCR) in trusted computing specifications is introduced into our architecture.The hierarchy of the agent in the whole software and hardware system is pointed out,which consists of three main modules.This method explains the content of exchange protocol in order to build trust in mutual system platform.Also give a statement of the enhancement of the security of peer-to-peer network computing.An approach of constructing network trusted path in secure operating system is addressed.Point out some related future research fields at last.
Key words: trusted computing;PCR authentication;platform access control;TCG;TPM

摘 要:基于可信计算技术提出了一个具有时间自校验功能的信任代理模块,并将它引入到提出的框架中。给出了它在系统中的层次位置,同时指出了该模块包括的 3 个主要部分以及它们的功能。框架可以保证平台协议交换的可靠性,并能保证检测环境提供信息的可靠性。指出了该算法如何加强平台访问控制和应用程序的安全性,并说明了该算法如何增强对等网络计算的安全性以及设计安全操作系统的网络可信路径方法,最后提出了一些相关的未来研究方向。
关键词:可信计算;PCR 验证;平台访问控制;TCG;TPM
文章编号:1002-8331(2007)17-0137-03 **文献标识码**:A **中图分类号**:TP393.08

1 研究背景

TCG 规范将在未来的几年内得到普遍应用,结合平台的安全性,最初的 Intel 的 LaGrande 技术和微软下一代安全计算基(Next-Generation Secure Computing Base,NGSCB)显示,它们的目标是建立基于 TCG 规范的核心硬件架构和操作系统组件。随着接受可信计算规范提供强大的安全特性,我们将不断提出这样的疑问: 如何应用可信计算技术到目前的系统中去。这篇文章将讨论可信计算技术在对等网络中的用途。

随着 P2P 技术在应用程序中的广泛使用以及 P2P 商业模式的产生,P2P 网络安全性变得更加严峻, 因为它们缺乏任何集中的授权。论文将说明 TCG 协议 DAA (Direct Anonymous Attestation)如何加强应用的稳定性和依靠平台的匿名性,从而减少 P2P 网络中的匿名攻击。更进一步,将阐述利用 DAA 协议来建立实体匿名验证,并能在已知终端之间建立一个安全通道(Secure Channels)。

2 可信计算中的验证机制

TCG 规范[1]的核心内容就是为各种计算平台提供了一整套基于可信平台模块 TPM 及平台可信构造块 TBB 建立信任及可信性的机制。TPM 作为平台的信任根,可以提供完整性检测,创建和使用数字签名以及隐私保密等机制。

TPM 中保存的根密钥是 EK(Endorsement Key),生产 TPM 的时候,制造厂商将 EK 与两个信任状绑定,分别为认可信任状(Endorsement Credential)和平台信任状(Platform Credential)。经过 TCG 评估机构评估后, 给 TPM 颁发一致性信任状(Conformance Credential)。在此基础上,TPM 将建立可信报告根 RTR、可信度量根 RTM 和可信存储根 RTS。可信报告根 RTR 提供密码机制对 TPM 的状态及信息进行数字签名, 可信存储根 RTS 提供密码机制保护保存在 TPM 之外的信息, 可信度量根 RTS 提供对平台的状态进行度量。TPM 交付用户后,所有者获得 TPM 的所有权,同时生成存储根密钥。

2.1 平台配置寄存器 PCR

TCG 规范[1]要求在一个 TPM 中最少有 16 个 PCR(Platform Configuration Register)。每个 PCR 是 TPM 内部的一个 20 字节存储区域,总长度为 160 个比特位。保存测量值的累计摘要,由历史数据组成,PCR 摘要更新机制至关重要, 称之为 “扩展(Extending)PCR”。

$PCR[n]_i \leftarrow SHA1(PCR[n]_{i-1} \| \text{measured data})$

其中,n 是 PCR 被更新的次数。与完整性检测相关的,除 PCR

还有一个是组件是存储检测日志 SML (Stored Measurement Log),它负责记录时间日志,并负责维护完整性变化的事件数据库。对于平台所有 PCR 的所有事件都将被 SML 所记录,因此 SML 和 PCR 是紧密联系的。

因为 SML 存储了平台所有历史信息,因此 SML 可能变得非常庞大,TPM 的存储能力是有限的,所以通常 SML 被存储在 TPM 之外,同时利用 PCR 中当前的数据和 SML 中的历史事件数据来检测当前平台的状态。

平台完整性的检测过程是:将 PCR 当前数据和 SML 中存储的与该 PCR 相关的事件序列,重新哈希以后,再和存储在 PCR 中的值进行比较,如果一致则说明平台完整性合法。

2.2 平台密钥与检测过程

EK 作为平台的标识由厂商提供,AIK 密钥由 TPM 生成,TPM 只可以有一个 EK,但是可以有多个 AIK。AIK 用来保护 EK 的私密性,TCG 规范要求 EK 永远不能被 TPM 泄漏。除了 TPM 及所在平台外,要求任何实体都不能确定 EK 与 AIK 的绑定关系,否则 AIK 信任状的获得者就可能掌握 TPM 的私密性标识信息,从而进行伪造攻击。AIK 信任状功能是让其他实体相信用 AIK 签名的信息确实来自一个可信平台,这样就在平台的层次上实现对私密性的保护。

在 TCG 规范的平台下,TPM 每次使用不同的 AIK 来签名 PCR 的值,这就能保证验证的不相关性。当一个请求验证者(Challenger)要求检查某平台,它可以请求相关 PCR 的值,通过 TPM 对 PCR 签名来保证其有效性,协议验证过程为:

步骤 1 Challenger 希望要检查一个平台的一个或者多个 PCR 的值。

步骤 2 平台 Agent 收集和请求 PCR 值相关的 SML 记录。

步骤 3 TPM 验证请求 PCR 值并用自己的 AIK 私钥签名。

步骤 4 TPM 发送请求的 PCR 值给平台 Agent。

步骤 5 平台 Agent 组合 TPM 证明过的信任状,并将签名过的 PCR 值和相应的 SML 记录以及相关的信任状发送给 Challenger。

步骤 6 Challenger 检查发送来的数据,并计算得到的 SML 记录和比较签名过的 PCR 值,从而评估平台信任状。

所以,要验证平台所处的状态,Challenger 需要通过检查获得 SML 的一部分和一个或者多个 PCR 的值来完成。

2.3 信任状验证方式

TCG1.1 规范和 TCG1.2 规范[1]提供了不同的验证拥有相关信任状的方式。

在规范 1.1 中,需要通过一个可信第三方,比如 Private-CA,来获得信任状。信任状来自 CA 颁发的数字证书,CA 必须知道 AIK 和 EK 的绑定,这种模式的安全性依赖 CA 的安全性、可靠性和可信性。CA 方法与传统密钥证书颁发过程类似,必须首先假设 CA 可信,验证协议过程为:

步骤 1 由 TPM 生成 AIK 密钥对。

步骤 2 TPM 将 EK 信任状、平台信任状、一致性信任状、用 EK 签名的 AIK 私钥发给 Private-CA。

步骤 3 Private-CA 对材料进行检查后,为 TPM 颁发 AIK 信任状。

版本 1.2[1]中依然支持 CA 这样的机制,但是引入了一种新的验证方法直接匿名验证 (Direct Anonymous Attestation, DAA)。DAA 使用可靠的密码技术来保证用户身份的私密性,而且并不需要引入可信的第三方。DAA 方法包括 Join 与 Sign 两个过程,需要 TPM、TPM 所在平台、DAA Issuer、Verifier 四方共同参与。

Join 过程:DAA Issuer 生成公钥 IKey 及相应私钥;TPM 收到 IKey 证书;TPM 生成 DAA 私钥(privDK),使用 EK 对自己进行标识,将相关信任状传给 Issuer;Issuer 验证通过后,向 TPM 方返回 CertDK,拥有(privDK,certDK),用它们对消息进行的签名可以用 Ikey 证书验证。

Sign 过程:TPM 生成 AIK 对,用(privDK,certDK)对 AIK 公钥进行签名,Verifier 获得 IKey 证书后确定 AIK 私钥在一个 TPM 中,为其颁发信任状。

3 完整性检测框架

完整性检测框架包括 3 个主要模块。如图 1 所示。

(1)检测机制,在一个运行的平台上,决定什么时候开始验证,如何安全地进行检测。

(2)完整性请求响应机制(An Integrity Challenge Mechanism)允许授权的请求者检查一个平台的检测列表,并验证是不是最新的,是不是全面的。

(3)一个完整性验证机制,验证检测列表是否完整,是否被篡改,并验证所有检查记录描述的确实是可信的。

图 1 显示了这些机制是如何进行远程验证的,检查由所谓的检查代理完成(Measurement Agents),它将创建一个检测文件,保存两个方面内容:(1)在内核中保存一个有序的检查列表;(2)给 TPM 汇报扩展的检查列表信息。

完整性请求响应机制允许远程请求者(challenger)获得检

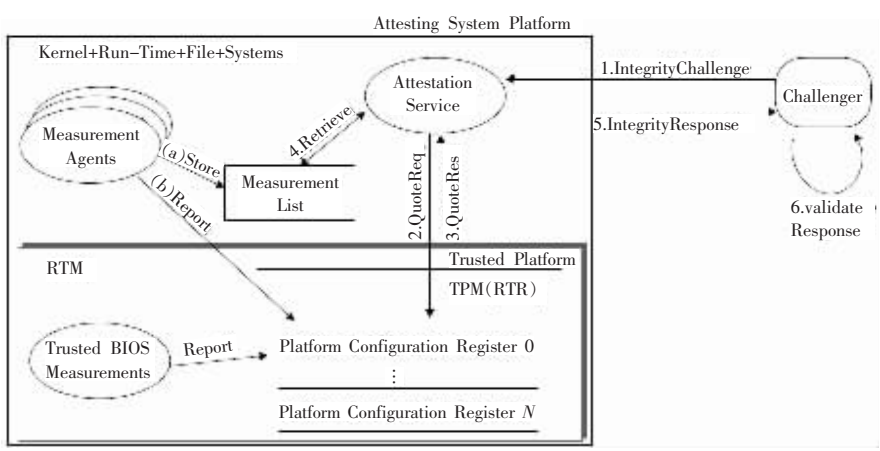

图 1 完整性检测框架

查列表和 TPM 签名的检查列表集(步骤 1),当收到这样的请求时,验证系统首先签名 TPM 中集合(步骤 2 和步骤 3)和内核中的签名列表(步骤 4),然后将两者全部发送给验证方(步骤 5),最后,验证方可以验证运行环境的完整性和可信性(步骤 6)。

3.1　检查机制

检测机制是一个新的可执行的程序被加载一个基本的检测,检测可执行内容和敏感性数据的能力。

BIOS 和 bootloader 可以检测内容代码的初始化,并能够让内核检测自身变化(比如加载模块),并能创建用户级的进程。内核利用同样的方法检查用户级的进程,可以检测可执行代码的进程(比如:dynamic loader 或者通过 execve 加载 htppd)。然后,这个代码可以检查后来的安全敏感输入(比如 dynamic loader 的 libraries 和 httpd 的配置脚本)。

请求验证者的信任决定于:被检查的代码对于安全敏感性输入的检查,并能够保护自己,防止没有检测的输入。操作系统可以通过 MAC(Mandatory Access Control)策略进一步保护应用程序,防止恶意源,没有检查的输入,并保护数据不受修改。为了能唯一标识任何一个可执行的组件,通过文件的全部内容计算一个 SHA-1 的哈希值。这个 160 bits 的哈希值将唯一标识文件的内容。不同的文件类型、版本以及扩展,可以通过唯一的指纹(fingerprints)区分。

每一个哈希值将被整理收集到一个检查列表(measurement list),代表系统的完整性历史。检查列表不允许被修改,否则攻击者可能隐藏完整性相关的行为。因为框架不带入侵检测特性,因此它不能够防止对检查列表的破坏和篡改。然而,为了能够阻止这些恶意的行为,使用硬件扩展来进行验证,这就是 TPM,可以为请求验证方提供检查列表的修改。

TCG 的 TPM 提供保护数据寄存器,称之为 PCR。可以通过两个函数修改它。一个是重新引导平台,这个时候所有 PCR 的值将被清 0,另一个函数是 TPM_extend 函数,PCR 利用 160 bits 的数字 n 和参数 i,通过计算 SHA1($PCR[i]||n$)得到当前的 PCR 的值 $PCR[i]$。新值存储到 $PCR[i]$中,根据 TCG 的规范和没有直接对硬件的攻击,这样没有任何其他途径来修改 PCR 寄存器的值。我们使用 PCR 来维护框架中所有检测的完整性检测,在检测组件发生作用或者潜在危害系统之前,所有检测将利用 TPM_extend 保存到 TPM 的 PCR 中。因此,任何检测软件将在得到直接(可执行程序)和非直接(静态的配置数据文件等)控制权之前已经被记录了。

比如:如果进行了 i 次检查,值分别为 $m_1 \cdots m_i$,在选定的 PCR 中包含的值为

$$SHA1(\cdots SHA1(SHA1(0||m_1) \| m_2 \cdots \| m_i))$$

TPM 的安全存储防止其它设备和系统软件修改这些值,当一个软件被一个恶意的系统扩展,并赋予了其他的值,扩展的过程也将被利用 SHA1 计算,然后阻止恶意系统修改 PCR 中的值来代表一个指定的系统。

3.2　完整性请求验证机制

完整性请求验证协议(Integrity Challenge protocal)描述了请求方如何安全地从验证系统中获得检查和验证的信息。当获取验证信息的时候,协议必须防止下面的威胁。重放攻击(Replay Attacks):一个恶意的系统可以重放验证信息(检查列表+TPM 数据集)。篡改攻击(Tampering Attacks):一个恶意系统或者一个攻击者在验证系统到达请求验证方之前篡改检查列表和 TPM 数据集。替换攻击(Masquerading):一个恶意系统或者一个攻击者利用另一个系统的或者系统历史的检查列表和 TPM 数据集替换当前的值。

3.3　完整性验证机制

配置和检查列表已经通过完整性请求验证协议,这里假设它们都是合法的。决定是否信任或者不信任一个验证系统,是基于独立地检查每一个检查列表,检查它的值和一系列信任的检查值。

更加成熟的验证模型可以关联多次检测得到一个评价结果,这个思想是记录的数据和完整性机制中定义好的数据相匹配。

新程序版本、未知程序或者其他代码将产生未知指纹(Fingerprint),程序指纹的更新可以被请求验证方检测,并添加到数据库中去。而且,包括一些已知脆弱性特定的旧版本程序将需要被重新分类成为不可信程序。

请求验证方必须拥有一个策略,来决定如何分类指纹,如何处理未知的和不信任的指纹。如果没有额外的策略加强机制进行隔离执行,一个不可信的指纹将导致不可信的完整性。

3.4　可信计算增强对等计算的安全性

假定每个节点平台都支持 TCG 规范,并具有相应的 DAA 证书,对等网络拥有一个唯一的标识。P2P 网络中节点代号是形式为 $N_p=\zeta^{fp}$ 的字符串,其中 ζ 的来自网络的名字,利用一个哈希函数进行转换,f_p 是 TPM 对节点 P 的代号密值[2,3]。

如果一个节点声称它具有某特定代号,其他节点可以检查声称者提供在一个随机消息 M 上 DAA 签名,利用这个特性,通过 M 附加其它信息来提供其它的安全服务。验证的节点可以检查并使用 DAA 验证算法,通过验证在形成代号 N_p 时的密值 f_p 和当前信任状的 f 值是否一致,来检查声明者是否拥有合法的信任状。

使用不可预测的随机信息 M 可以防止重放攻击,因为恶意节点可能捕获并重放信任状。节点 P 声明代号 N_p 是固定的,该值由网络名函数和在 Sign 阶段的特定 f_p 来决定。通过 DAA 算法建立的匿名验证机制,节点可以被其它节点验证,但是平台身份并没有在验证过程中泄漏。

这种匿名方法需要确保平台可以从可识别的 Issuers 处得到一个信任状。否则就很难阻止一个节点通过不同的 f 值得到多个信任状,并通过这些信任状创建多个代号。如果假设初始信任状是在制造的时候颁发,则可以阻止平台获得多个信任状。这样,平台制造商是授权颁发者,节点的软件可以利用它们的公钥进行配置,因此制造商就是 DAA 的信任根。另一个可行的方案是由可信服务提供者提供信任根。

通过使用匿名代号机制,某个节点的行为将具有一定的相关性。因为 DAA 验证并不泄漏平台标识(EK),所以节点的行为和特定的 TPM 无关。因为在 Sign 阶段代号和 DAA 签名时候的值是不一致($\zeta_1 \neq \zeta$),因此代号可以有效地保护节点的行为,而且代号只和特定的平台的 TPM 相关。一个给定的节点可能需要在不止一个 P2P 网络中活动,通过网络名来确定密值 ζ 得到代号,可以保证在不同的网络中,节点的行为依旧保持不相关性。

另外一种方法是通过私有 CA 来建立验证机制,这种思想

(下转 240 页)