TPM的两个主要密码授权协议的安全性分析与改进

刘 皖[1] 谭 明[2] 陈兴蜀[3]

(解放军信息工程大学电子技术学院 郑州 450004)[1] (北京市 7227 信箱 北京 100072)[2]

(四川大学计算机学院 成都 610064)[3]

摘 要 可信计算的主要功能是由可信平台模块(TPM)完成的。对象无关授权协议(OIAP)、特定对象授权协议(OSAP)是TPM在可信计算平台中运行的基础,确保这些协议的安全运行是极为重要的。本文对这两个授权协议进行了逻辑描述并对其安全性进行了分析,针对协议的安全隐患提出了相应的改进方法。

关键词 可信计算,可信平台模块,授权协议

Security Analysis and Improvement of the TPM's Two Main Authorization Protocols

LIU Wan[1] TAN Ming[2] CHEN Xing-Shu[3]

(Institute of Electronic Technology, the PLA Information Engineering University, Zhengzhou 450004)[1]

(The 7227th Mail Box, Beijing 100072)[2] (Collge of Computer, Sichuan University, Chengdu 610064)[3]

Abstract The main function of trusted computing is done by the trusted platform module. For the working of TPM is based on the OIAP and OSAP protocol, it is most important to ensure the secure working of the two protocols. This article does logic description and security analysis on the two protocols, then presents some ways to protect the two protocols from being attacked.

Keywords Trusted computing, Trusted platform module, Authorization protocol

1 前言

可信计算的主要功能是由可信平台模块(TPM)完成的。TPM是可信计算技术的核心,是一个含有密码运算部件和存储部件的小型片上系统。作为可信平台的构件,TPM的组件被信赖能够正常工作,但TCG组织制定的TPM规范只包括评论信息和正规陈述,并没有实施部分,其实施由具体厂家实现。TPM规范也没有安全性分析和安全机制,这是TPM规范的弱点所在。本文对TPM规范中的两个主要的授权协议进行了形式化描述与安全性分析,针对协议的安全漏洞提出了相应的解决方法。

2 OIAP和OSAP概述

TPM所有者和TPM控制的每个实体(例如密钥)都有授权数据(共享秘密、口令),知道授权数据,意味着可以使用该实体。使用需要授权的实体,必须提供该实体的授权数据。TPM授权协议的目的是让请求者向TPM证明自己知道实体的秘密,因而可以使用这些资源(如密钥)。

对象无关授权协议(OIAP)、特定对象授权协议(OSAP)是TPM规范中最为常用的基本授权协议,绝大多数TPM命令都要经这两个授权协议授权执行。OIAP可以向任意实体提供多个授权会话,一旦一个OIAP会话被创建,就可以用来授权使用被TPM管理的任何实体,该会话直到一方请求会话终止才关闭,OIAP会话由请求者通过TPM_OIAP( )命令创建。OSAP只向单个实体提供一个授权会话,但能在不增加会话创建的开销的前提下对这个特定实体授权多个命令的执行。另外,OSAP在会话中创建临时秘密,将临时秘密当作共享秘密,而不是像OIAP一样直接将实体授权数据当作共享秘密。OIAP和OSAP均能保护授权数据不被泄漏,并都使用了"新鲜值nonce"机制,以预防重放攻击和中间人攻击。TPM规范中OIAP及OSAP会话过程的描述见文[5]。

3 TPM的两个主要授权协议OIAP和OSAP的安全性分析及改进

3.1 对象无关授权协议(OIAP)的安全性分析及改进

OIAP可以向任意实体提供多个授权会话。当上层调用者请求TPM执行受保护的TPM命令时,TPM接收到上层调用者发送给自己的TPM命令序数、参数以及上层调用者和TPM共有的秘密后,在OIAP中通过对该秘密的验证来判决上层调用者是否有权请求该TPM命令。

3.1.1 OIAP协议的逻辑描述与分析

根据TPM规范,首先对OIAP协议进行标记,然后进行协议的逻辑描述与分析。其中USER表示请求者、TPM直接用TPM表示。令TPM_OIAP协议的标识符为OIAP,消息类型标识符tag= RQU或RSD,ordinal为使用OIAP进行授权的命令,USER的新鲜值标识符 N_u = nonceOdd,TPM的新鲜值标识符 N_t = authLastNonceEven, N'_t = nonceEven, USER和TPM的共享秘密 S_{UT} = key. usageAuth或ownerAuth,输入参数摘要 inParamDigest = H(ordinal, inArgOne, inArgTwo),输出参数摘要 outParamDigest = H(returnCode, ordinal, outArgOne),输入授权创建参数 inAuthSetupParams = authHandle, N_t, N_u, continueAuthSession0,

刘 皖 博士研究生,主要从事信息安全工程领域的研究与开发;谭 明 硕士,工程师,主要从事信息安全工程领域的研究与开发;陈兴蜀 博士,副教授,主要研究领域:信息安全、计算机网络领域。

输出授权创建参数 outAuthSetupParams = authHandle, N'_t, N_u, continueAuthSession1, 输入参数授权消息认证码 inAuth = HMAC(S_{UT}, inParamDigest || inAuthSetupParams), 输出参数授权消息认证码 resAuth = HMAC(S_{UT}, outParamDigest || outAuthSetupParams), $U \Rightarrow T : M$ 表示 U 将消息 M 发送给 T, 则 OIAP 协议的逻辑描述如下:

(1)USER⇒TPM: RQU, OIAP

(2)TPMU⇒SER: N_t, authHandle

(3)USER⇒TPM: RQU, ordinal, keyHandle , inArgOne, inArgTwo, authHandle, N_u, continueAuthSession0, HMAC(S_{UT}, inParamDigest ||inAuthSetupParams)

(4) TPM⇒ USER: RSD, returnCode, outArgOne, N'_t, continueAuthSession1, HMAC(S_{UT}, outParamDigest || outAuthSetupParams)

在步骤(1)中, USER 向 TPM 发送请求标识符 RQU 和授权协议的标识符 TPM_OIAP, 表示 USER 请求 TPM 以 OIAP 协议对随后的命令进行授权验证, 以执行 TPM 的命令或使用资源。

在步骤(2)中, TPM 向 USER 发送新鲜值 N_t, 以备随后证明 USER 所发包含该新鲜值的消息的新鲜性, 防止重放攻击。

在步骤(3)中, USER 向 TPM 发送的该消息包含 RQU, ordinal, 用来指示 KEY 或者 OWNER 的 keyHandle, 输入参数 inArgOne 和 inArgTwo, N_u 以及它们的消息认证码 inAuth = HMAC (S_{UT}, inParamDigest || inAuthSetupParams))。TPM 用 keyHandle 找到相应的 KEY 的授权数据 key. usageAuth 或者 OWNER 的授权数据 ownerAuth 用作 S_{UT}。消息认证码 inAuth 的作用是检验消息的完整性和消息发送主体的真实性, 如果验证通过, TPM 就可以相信线上所传送来的数据是由所期望的 USER 传送来的。

在步骤(4)中, TPM 向 USER 发送的该消息包含响应标识符 RSD、操作返回码 returnCode、输出参数 outArgOne、N'_t 以及它们的消息认证码 resAuth= HMAC(S_{UT}, outParamDigest ||N'_t||N_u)。resAuth 与上一步中 inAuth 的作用相似, 如果验证通过, USER 就可以相信线上所传送来的数据是由所期望的 TPM 传送来的。

从步骤(3)、步骤(4)可以看出, OIAP 协议可以保证 TPM 和 USER 双方对线上传送的数据达成相互的信任。尽管 keyHandle 没有直接参与 HMAC 计算, 但由于 keyHandle 和参与 HMAC 计算的 S_{UT} 有着直接的映射关系, 因此 keyHandle 的完整性也得到了保证。OIAP 较好地实现了实体使用上的授权保护。

3.1.2 OIAP 协议的改进

在以上对 OIAP 的逻辑描述与分析的过程中, 我们在 OIAP 协议的执行过程中似乎没有发现明显的安全漏洞, 但我们注意到 OIAP 协议授权目的的获得是以双方所拥有的授权数据是机密的为前提。在授权协议的授权数据管理中, 授权数据是由与授权数据相关的实体的创建者设置的, 创建者口令的泄露会引起该协议的全部失败。如果要使得用户口令的泄露不影响安全协议中密钥的机密性, 可以对 OIAP 的授权数据管理做以下改进: 原来授权数据就是用户口令的转换值, 现在将授权数据改为用户口令、预设秘密和 TPM 生成的新鲜值的转换值。这样在用户口令泄露的情况下, 由于预设秘密和新鲜值的不同, 攻击者也不能生成与合法用户相同的授权数据, 从而预防了攻击者通过窃取用户口令来攻破 OIAP 协议的攻击手段。同时, 由于在授权数据中添加了新鲜值, 使得授权数据在每次对不同实体授权时动态变化, 从而预防了通过窃听收集大量明文/密文对进行的已知明文攻击与选择明文攻击。

3.2 特定对象授权协议(OSAP)的安全性分析及改进

OSAP 向单个实体提供一个授权会话, 并使新的授权信息能够机密地传输。OSAP 与实体相关, 它与 OIAP 在理念和设计上是相似的, 需要认证目标实体并提供额外的 nonce。OSAP 通过创建被用来保护会话通信安全的临时共享秘密进行初始化, 使用 TPM_OSAP 命令创建会话。发送者在需要多次使用实体时, 只需要获得一次该实体的授权数据, 此时的授权数据来自输入的口令。在这一点上, 可以同对 OIAP 协议的改进一样对 OSAP 协议中授权数据的产生进行改进。

3.2.1 OSAP 协议的逻辑描述与分析

和 OIAP 协议标记一样, 使用 USER 表示请求者, TPM 直接用 TPM 表示。TPM_OSAP 协议的标识符为 OSAP, 消息类型标识符 tag = RQU 或 RSD, handle 为使用者请求的资源句柄, ordinal 为使用 OSAP 进行授权的命令, N_{up} = nonceOddOSAP, N'_t = authLastNonceEven, N_{tp} = nonceEvenOSAP, N_u = nonceOdd, N_t = nonceEven, 输入参数摘要 inParamDigest = H(ordinal, inArgOne, inArgTwo), 输出参数摘要 outParamDigest = H (returnCode, ordinal, outArgOne), 输入授权创建参数 inAuthSetupParams = N'_t||N_u, 输出授权创建参数 outAuthSetupParams = N_t||N_u, USER 和 TPM 的共享秘密 S_{UT} = key. usageAuth 或 ownerAuth, 临时共享秘密 S_{temp} = HMAC(S_{UT}, N_{tp}||N_{up}), 输入参数授权消息认证码 inAuth= HMAC(S_{temp}, inParamDigest || inAuthSetupParams), 输出参数授权消息认证码 resAuth= HMAC (S_{temp}, outParamDigest || outAuthSetupParams), $U \Rightarrow T : M$ 表示 U 将消息 M 发送给 T, 则 OSAP 协议的运行步骤如下:

(1) USER⇒TPM: RQU, OSAP, keyHandle, N_{up}

(2) TPM⇒USER: N'_t, N_{tp}, authHandle

(3)USER⇒TPM: RQU, ordinal, keyHandle, inArgOne, inArgTwo, authHandle, N_u, continueAuthSession0 , HMAC (S_{temp}, inParamDigest ||N'_t||N_u)

(4) TPM⇒ USER: RSD, returnCode, outArgOne, N_t, continueAuthSession1 , HMAC(S_{temp}, outParamDigest ||N_t||N_u)

和 OIAP 相比, OSAP 使用临时共享秘密 S_{temp} 代替直接使用共享秘密 S_{UT}, 这主要是为了下一次再使用同样的资源的时候, 不需要再使用实体的授权数据, 而直接使用共享的临时秘密, 因而是一种可以对同一实体进行多次授权操作的协议。根据以上协议运行的步骤, TPM 在第一步后就创建了授权会话并用 authHandle 来指示, 同时将 S_{UT} 作为 HMAC 计算中的共享秘密生成了临时共享秘密 S_{temp} 并存于授权会话数据区中。TPM 对 OSAP 协议进行授权验证的时候, 是通过授权句柄 authHandle 找到共享秘密 S_{temp}, 计算 HM = HMAC (S_{temp}, inParamDigest || inAuthSetupParams), 并把 HM 和 inAuth 比较, 进行授权验证的。由于 S_{temp} 和输入参数没有直接的联系, 这就造成了如下的替换攻击。

根据对 OSAP 协议的分析, 可以得出如下的攻击方法:

(1)攻击者获得实体 entity-n 的授权数据 authData-n 并使用实体 entity-n 的授权数据 authData-n, 执行 OSAP 会话,

获得授权句柄 authHandle-n 和临时共享秘密 S_{temp-n}。

(2)攻击者对于需要获得实体 entity-m 的授权数据才能执行的命令,输入 authHandle-n 作为授权句柄,并用临时共享秘密 S_{temp-n} 计算输入参数授权消息认证码 inAuth = HMAC(S_{temp-n}, inParamDigest || inAuthSetupParams)。

上述攻击中,TPM 根据 authHandle-n 找到的临时共享秘密 S_{temp-n} 计算出的 HM 和攻击者所计算出的 inAuth 是一样的,因而能够通过 TPM 的验证使 TPM 认为该命令得到授权,从而执行攻击者本无权执行的命令。由于 TPM 中的大部分命令都可以通过 OSAP 进行授权执行,这个漏洞的安全隐患是很大的。

3.2.2 OSAP 协议的改进

基于以上分析,对 OSAP 协议的实施可以做如下修改:

(1)执行命令 TPM_OSAP( )的过程中,同时保存临时共享秘密 S_{temp} 与共享秘密 S_{UT}。

(2)执行 OSAP 协议授权验证时,将与命令参数直接关联的实体授权数据和 TPM 执行命令 TPM_OSAP()时保存的共享秘密 S_{UT} 相比较。如果相等,再用 S_{temp} 进行 HMAC 验证。

这样就把共享秘密 S_{UT} 和命令输入参数直接关联起来了,使得该替换攻击无从下手。

结束语 对象无关授权协议(OIAP)、特定对象授权协议(OSAP)是 TPM 在可信计算平台上运行的基本协议,确保这些协议的安全运行极为重要。本文对这两个授权协议进行了逻辑描述并对其安全性进行了分析,针对协议的安全隐患提出了相应的改进方法,对于研发 TPM 芯片时的工程实现提供了参考。

参 考 文 献

1 谭兴烈. 可信计算平台中的关键部件 TPM. 信息安全与通信保密,2005-02

2 Graeme Proudler (Hewlett Packard). Trusted Computing, TCG Technical Committee,2004(报告)

3 科学技术信息研究所. 中国 TCG 标准进入倒计时[EB/OL]. 2005-11-25

4 TCG Glossary. http://www trustedcomputinggroup. org, 2004 (6):4~7

5 Trusted computing work group. TPM Main Specification. http://www. trustedcomputinggroup. org. 2005

6 范红. 安全协议形式化分析理论与方法:[博士论文]. 解放军信息工程大学,2003-02

7 可信终端成就安全体系. http://www. ccw. com. cn/applic/tech/htm2004/20040613_1340T. asp

8 陈军. 可信平台模块安全性分析与应用:[博士论文]. 中科院,2006-03

9 叶定松. 计算机可信改造与实践. 见:中国信息安全发展趋势与战略高层研讨会文集,2005,6:51~53

(上接第 263 页)

QinHan $\vdash_C$ Bell, Bo, Drum

……

其中 XiaShang $\vdash_C$ Cymbals, Bo, Drum 可以读做两个数据库中夏商时期的乐器有铙、镈和鼓。

结论 信息流理论是一门信息科学理论,它是关于信息如何在分布式系统中流动的数学理论模型,为我们研究分布式系统的语义互操作问题提供了理论基础。而分布式系统的分析形成具有相对性,且既可以是具体的,也可以是抽象的,所以它的应用将非常广泛。

参 考 文 献

1 Devlin K. Logic and Information. Cambridge,1991

2 Devlin K. Introduction to Channel Theory. ESSLLI 2001, Helsinki, Finland,2001

3 Dretske. Knowledge and the Flow of Information. Oxford. Basil Blackwell,1981

4 Floridi L. What is the Philosophy of Information? Mataphilosophy, 2002,33(1-2): 123~145

5 Barwise J, Seligman J. Information Flow: the Logic of Distributed Systems. Cambridge: Cambridge University Press,1997

6 Checkland P. Systems Thinking, Systems Practice. Chichester: John Wiley & Sons,1981

7 Kent R E. The Information Flow Framework. Starter document for IEEE P1600. 1, the IEEE Standard Upper Ontology Working Group, 2001. http://suo. ieee. org/IFF/

8 Kent R E. The IFF Approach to Semantic Integration. In: the Boeing Mini-Workshop on Semantic Integration, November 2002

9 Kalfoglou Y, Schorlemmer. Using Information Flow Theory to Enable Semantic Interoperability. In:Proceedings of the 6th Catalan Conference on Artificial Intelligence (CCIA '03), Palma de Mallorca, Spain, October 2003

10 Kalfoglou Y, Schorlemmer M. IFMap: an ontology mapping method based on information flow theory. Journal on Data Semantics,2003, 1(1):98~127

11 Liu Hongzhe,Bao Hong,Feng Junkang. IF based Semantic Interoperability for Distributed Digital Museums. Computing and Information System Journal,2006,10

12 Liu Hong-zhe, Bao Hong, Wu Jing, et al. An Information Flow Based Approach to Semantic Integration of Distributed Digital Museums. ICMLC06,Dalian,China,2006