第34卷 第22期 计 算 机 工 程 2008年11月
Vol.34 No.22 Computer Engineering November 2008
·安全技术· 文章编号：1000—3428(2008)22—0159—03 文献标识码：A 中图分类号：TP309

TCG命令验证协议的改进方法

康新振，王震宇，徐 锐
(解放军信息工程大学信息工程学院，郑州 450002)

摘 要：经过分析显示可信计算联盟(TCG)命令验证协议会受到一种基于 Dolev-Yao 模型的中间人攻击，对系统的可信性和安全性造成影响。针对该攻击，文章提出一种协议改进方法。在改进后的协议中，可信平台模块(TPM)和访问者能对会话状态进行有效的沟通，从而抵御中间人攻击。

关键词：可信计算联盟；命令验证协议；中间人

Improvement Method of TCG Command Validation Protocol

KANG Xin-zhen, WANG Zhen-yu, XU Rui
(Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002)

【Abstract】 This paper shows that the Trusted Computing Group(TCG) command validation protocols are exposed to a Dolev-Yao Man in The Middle(MiTM) attack, which will tamper with the security and the trustworthiness of the entire system. In order to avoid such attack, this paper proposes a countermeasure which makes an effective way through which the caller and TPM can well understand the session state of each other.

【Key words】 Trusted Computing Group(TCG); command validation protocols; Man in The Middle(MiTM)

1 概述

计算机信息的安全问题很难单靠软件解决。可信计算是在计算设备硬件平台上引入安全芯片——可信平台模块(TPM)[1-3]，通过TPM提供的安全特性来提高整个系统的安全性。为了实现资源的授权访问，TPM执行一系列精心设计的协议来抵御各种形式的攻击，即命令验证协议。命令验证协议向TPM证明访问者有权执行一个操作或访问一个对象。验证主要针对可能会影响安全、隐私或者会暴露平台秘密的TPM命令。验证的证据来自一个共享的秘密——授权数据(AuthData)。

在命令验证协议设计过程中充分考虑了可能受到的安全威胁，通过HMAC和非对称加密技术保证授权数据传递过程中的完整性和机密性、并且通过“rolling nonce”滚动随机数机制防止重放攻击和中间人攻击。对于可能受到的DoS攻击，规范中并没有定义保护措施，要求厂商在产品设计过程中自己解决这个问题。即使有这些安全措施，本文的分析显示协议仍然会受到Dolev-Yao[4]模型的中间人(MiTM)攻击，造成访问者和TPM所认为的授权数据不一致、资源无法被访问的严重后果。

2 命令验证协议流程分析

命令验证协议分为授权协议和授权数据管理协议2类。授权协议用来安全地将证据从访问者传递到TPM，证明访问者拥有授权数据，从而有权执行一个操作或访问一个对象。TPM1.2规范中包括3个命令授权协议：对象无关的授权协议(Object-Independent Authorization Protocol, OIAP)，对象特定的授权协议(Object- Specific Authorization Protocol, OSAP)和代理特定的授权协议(Delegate-Specific Authorization Protocol, DSAP)。授权数据管理协议用来创建或修改授权数据。TPM1.2规范中包括3个授权数据管理协议：授权数据插入协议(AuthData Insertion Protocol, ADIP)，授权数据修改协议(AuthData Change Protocol, ADCP)和非对称授权数据修改协议(Asymmetric Authorization Change Protocol, AACP)。

TPM1.1定义了OIAP和OSAP。TPM1.2新增了DSAP，用来在代理模式下进行授权验证。大多数的命令验证都能用OIAP和OSAP中任意一个。OIAP是为了提高效率而设计的，一次会话可以用来验证多个实体。OSAP协议过程与OIAP协议过程大体相似，区别只是它用Auth Data和随机数产生一个临时会话秘密，由临时秘密和随机数做HMAC产生的值作为证据。OSAP的这个特点，可以被用在授权数据管理协议中加密新的授权数据。ADIP协议用来在一个实体的创建期间插入新授权数据。ADCP和AACP允许一个实体改变授权数据。

本文以ADIP为例对命令验证协议的流程进行分析。ADIP用于在生成对象时生成与之相联的授权数据。创建一个新对象，需要对它的父实体进行验证，且必须使用OSAP对话。OSAP会话首先验证父实体，用父实体的授权数据产生一个临时的会话秘密，并用临时会话秘密加密子实体的授权数据，使新的授权数据不会直接在会话中传输，确保其机密性。ADIP会话过程如图1所示。

步骤说明如下：

(1)1~2表示访问者通过TCS(TSS Core Service)发送TPM_OSAP命令给TPM，同时发送*nonceOddOSAP*，并通过*parenthandle*指定父实体。

(2)3~4表示TPM创建会话并将其和一个验证句柄

基金项目：国家“863”计划基金资助项目(2007AA01Z483)
作者简介：康新振(1983－)，男，硕士研究生，主研方向：可信计算，信息安全；王震宇，副教授；徐 锐，硕士研究生
收稿日期：2008-03-19 **E-mail：**kxz_121@163.com

authHandle 联系起来；产生随机数 *authLastNonceEven* 并和 *authHandle* 保存在一起；产生 *nonceEvenOSAP*，产生会话秘密

sharedSecret= HMAC(*key.usageAuth, nonceEvenOSAP, nonceOddOSAP*)

其中，*key.usageAuth* 是父实体的授权数据；将验证句柄和 2 个随机数 *nonceEvenOSAP* 和 *authLastNonceEven* 返回给 TCS。

(3)5~9 表示 TCS 收到 OIAP 命令的返回值，保存验证句柄 *authHandle* 和 2 个随机数；用同样的方法产生会话秘密

sharedSecret= HMAC(*key.usageAuth, nonceEvenOSAP, nonceOddOSAP*)

产生随机数 *nonceOdd* 并和 *authHandle* 联系并存储在一起；计算

inAuth= HMAC(*sharedSecret, inParamDigest, inAuthSetupParams*)

其中，*inParamDigest* 是 XOR(*entityAuthData, SHA*1(*sharedSecret, authLastNonceEven*))的结果；*entityAuth Data* 是为新实体创建的授权数据；*inAuthSetupParams* 是按顺序排列的命令参数{*auth Handle, authLastNonceEven, nonce Odd, continueAuthSession*}；最后发送命令、参数和 *HMAC* 到 TPM：(*tag, paramSize, ordinal, inArgOne, inArgTwo, auth Handle, nonceOdd, continueAuthSession, inAuth*)。

(4)10~12 表示 TPM 加载验证句柄和授权数据，按照同样的方法计算 *HMAC* 并和来自 TCS 的 *HMAC* 做比较，如果一致则继续，不一致则返回 *TPM_AUTHFAIL*；计算出新实体的授权数据，*entityAuthData* = XOR(*inParamDigest, SHA*1(*sharedSecret, authLastNonceEven*))；执行命令，产生 *non ceeven* 代替 *authlastnonceeven*；计算 *resAuth*＝*HMAC*(*sharedSecret, outparamdigest, outauthsetupparams*)；返回输出结果：(*tag, paramSize, returnCode, outArgOne, nonceEven, continueAuthSession, resAuth*)。

(5)13~14 表示访问者接收到结果，保存 *nonceeven*；计算 *HM=MAC*(*sharedSecret, outparamdigest, outauthsetupparams*)；比较 *HM* 和 *resAuth*，如果一致则接受命令结果。

(6)15~16 表示结束会话。

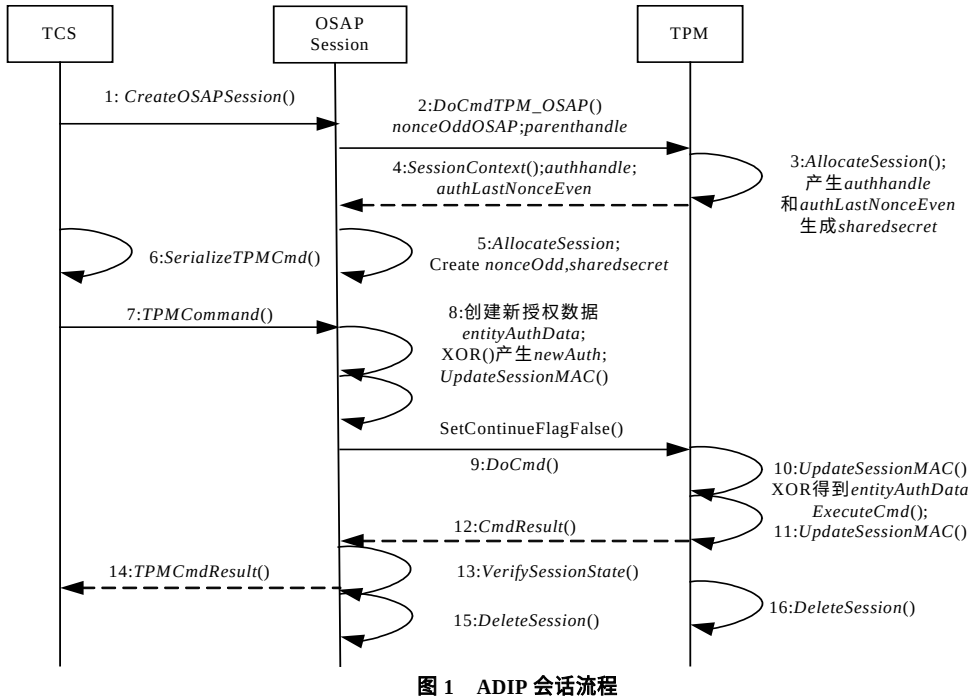

图 1　ADIP 会话流程

3　攻击方法

为防止重放攻击和其他的包伪造攻击，协议采用 rolling nonces 和 *HMAC* 方法。TPM 可以通过 *HMAC* 方法检查数据包是否被修改过，但不能将正常的网络通信错误和一个由 MiTM 发起的真正的包伪造攻击区分开。这就为将要提到的攻击提供了一个可行的切入点。

另外，在所有的可信计算联盟(Trusted Computing Group, TCG)命令验证协议中，TPM 执行完命令并返回结果给访问者后，并不对访问者是否收到消息做确认，很容易造成会话参与双方所认为的会话状态不一致。

Dolev-Yao模型是安全协议形式化分析与验证研究中使用最广泛的协议攻击者模型[5-6]，它假设攻击者的知识和能力不能够低估，攻击者可以控制整个通信网络。该模型中攻击者具有如下能力：(1)窃听所有经过网络的消息；(2)阻止和截获所有经过网络的消息；(3)存储所获得或自身创造的消息；(4)根据存储的消息伪造消息，并发送该消息；(5)作为合法的主体参与协议的运行。

文献[7]证明，在 OIAP 协议执行过程中存在 Dolev-Yao 模型的中间人攻击。该攻击造成访问者认为会话失败，但 TPM 认为会话处于成功状态。本文的分析显示，TCG 的其他命令验证协议也存在这个弱点，而且，Dolev-Yao 模型中的中间人的攻击所造成的会话状态不一致的情况，对授权数据管理协议造成的危害更为严重。

ADIP 会话过程中，TPM 收到插入授权数据的命令，验证访问者身份正确，然后执行命令，保存插入的对象和其授权数据。此时 TPM 返回命令执行结果给访问者，但协议中，TPM 并不对访问者是否收到命令执行结果做确认。在 TPM 返回命令的执行结果时，中间人攻击者截取这个消息，存储并改变消息中一些 bit 位，然后将改变后的消息发送给访问者来进行欺骗。这个消息类似一个合法的重放消息，但是在进行 HMAC 检查消息完整性时不通过，给用户一个假相，好像一个网络错误。进而，访问者认为对象和授权数据产生未成功，而 TPM 这一端则认为命令成功执行完毕，造成会话双方所理解的会话状态不一致。

此外，2 个授权数据管理协议 ADCP 和 AACP 同样存在这样的弱点，会遭到类似的攻击。在会话的最后一步，TPM 将改变授权数据命令的执行结果返回过程中，中间人进行攻击，这样访问者认为授权数据没有改变，仍用原来的授权数据，而 TPM 则认为更新成功，接受新的授权数据。对于受到攻击的会话中所操作的资源，访问者认为的授权数据和 TPM 认为的授权数据不一致。由于协议规定，TPM 将知道授权数据视为资源的合法授权使用者的唯一证据，如果丢失授权数据，下次访问者试图访问该资源时，将不能提供 TPM 认为的正确的授权数

据。实际上，该资源将没有“合法授权使用者”，因为在 TPM 外部没有任何实体知道该授权数据。之后，该资源除了占用空间和浪费资源外，没有其他用途，用户也永久失去了自己的数据。

4 改进措施

以上分析显示，造成攻击的原因是会话双方缺少关于会话状态信息的沟通机制，访问者和 TPM 可能会认为会话状态处于不同的状态。本文以 ADIP 协议为例提出的改进措施是，TPM 和访问者都向对方报告会话状态，访问者和 TPM 暂时记录原来的授权数据等资源的有关信息，以便当会话最后处于失败状态时，回滚到原来的状态，以免造成授权数据的不一致或资源的浪费。

改进后的协议中，访问者在收到 TPM 执行命令的结果后，发送一个消息给 TPM 报告访问者的所有会话状态，并认为会话成功，结束会话。TPM 收到报告消息后，才最终认为会话成功，结束会话；如果收不到消息或消息错误则认为会话失败，结束会话。

本文定义了 1 个表示所有会话状态的位图，该位图由 2 个 bit 位表示 1 个会话状态的数组，每个状态的计算规则如下：

(1)设为 0 表示会话处于打开或 Unknown 状态；

(2)设为 1 表示会话处于 Failed 状态；

(3)设为 2 表示会话处于 Success1 状态；

(4)设为 3 表示会话处于 Success2 状态。

Failed 状态表示会话失败。除了 Success1, Success2 和 Failed 之外的状态称为 Unknown 状态。TPM 成功执行完命令返回结果之后，收到访问者的会话状态报告消息之前，TPM 认为会话处于 Success1 状态 ;访问者收到命令返回结果之后，下一次会话中 TPM 报告上一次会话状态之前，访问者认为会话处于 Success1 状态。实际上，Success1 状态并不代表会话真正处于成功状态，它只是一个“准成功状态”。

TPM 收到访问者的会话状态报告消息之后，将会话置于 Success2 状态，结束会话并在下一次会话时向访问者报告会话状态；访问者下一次收到 TPM 的会话状态消息之后，如果 TPM 会话成功则将自己的会话置为 Success2 状态，否则置为 Failed 状态。

图 2 描述了改进后的协议流程，其中，Bc 和 Bt 分别表示访问者和 TPM 的会话状态位图。第 12 步、第 15 步和第 17 步的消息均由新的授权数据保护。访问者发送第 15 步的消息后，如果收到第 17 步消息，则会话置为 Success2 状态并结束会话。如果在一定时间内收不到第 17 步消息都会关闭会话，将会话状态置为 Success1，待下一次会话中，TPM 报告会话状态时，再决定是否将会话置为 Success2 状态。TPM 正确收到第 15 步消息，结束会话并将会话状态置为 Success1，发送第 17 步的消息，并在下一次会话中报告的会话状态域时包含上次会话状态。

在改进后的 ADIP 协议中，访问者和 TPM 能对会话状态进行很好的沟通，不管会话状态是成功还是失败，都不会造成会话状态和授权数据的不一致，从而不会造成用户密钥或数据的丢失。

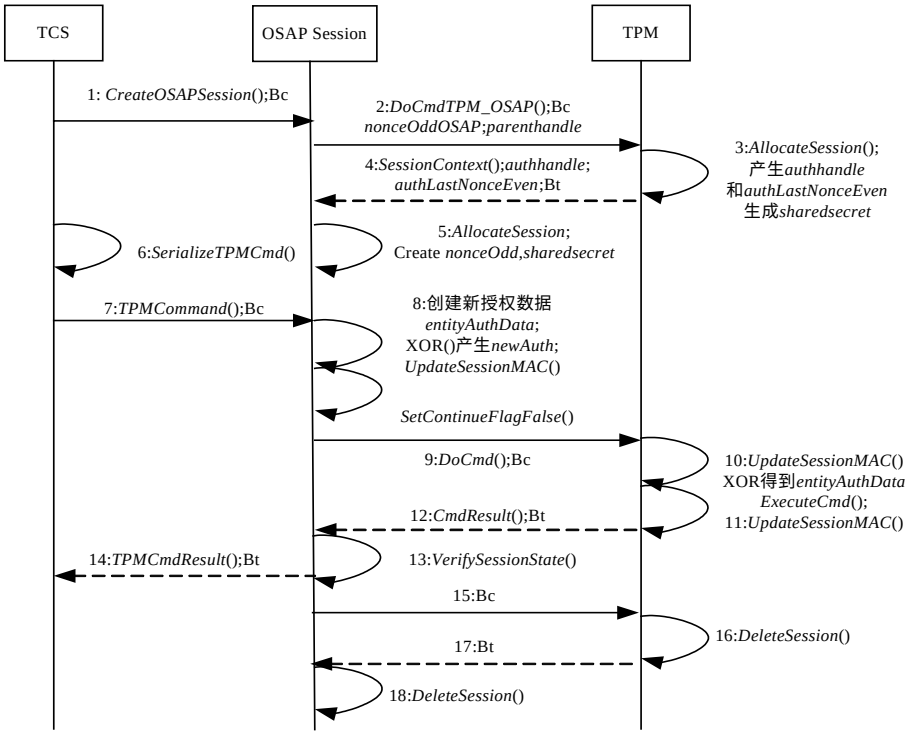

图 2 改进后的 ADIP 会话

5 结束语

TCG 的命令验证机制在保护资源机密性方面的具有重要作用。虽然 TCG 规范采用了相应措施保护协议不受重放攻击和中间人攻击，但本文的分析显示了中间人攻击的可能性，对此提出一种改进措施。改进后的协议，无论会话状态是成功还是失败，都不会造成会话状态和授权数据不一致。对改进前后协议安全性的形式化分析是今后研究的重点。

参考文献

[1] TCG Specification Architecture Overview Specification Revision 1.2[EB/OL]. (2004-04-01). http://www.Trustedcomputinggroup.org.

[2] Trusted Computing Platform Alliance Main Specification Version 1.1b[EB/OL]. (2002-02-01).http://www.trustedcomputinggroup.org.

[3] TPM Main Part 1, 2, 3[EB/OL]. (2006-03-02). http://www.trusted computinggroup.org.

[4] Dolev D, Yao A C. On the Security of Public Key Protocols[J]. IEEE Transactions on Information Theory, 1983, 29(2):198-208.

[5] 薛 锐, 冯登国. 安全协议的形式化分析技术与方法[J]. 计算机学报, 2006, 29(1): 1-20.

[6] 卿斯汉. 安全协议的设计与逻辑分析[J]. 软件学报, 2003, 14(7): 1300-1309.

[7] Bruschi D, Cavallaro L, Lanzi A, et al. Replay Attack in TCG Specification and Solution[C]//Proc. of ACSAC'05. Singapore: [s. n.], 2005.