基于串空间的可信计算协议分析

冯 伟[1),2)] 冯登国[1)]

[1)](中国科学院软件研究所可信计算与信息保障实验室 北京 100190)

[2)](中国科学院大学 北京 100190)

摘 要 可信计算技术能为终端、网络以及云计算平台等环境提供安全支撑,其本身的安全机制或者协议应该得到严格的形式化证明.该文基于串空间模型对其远程证明协议进行了分析.首先,扩展了串空间的消息代数和攻击者串,使其能表达可信计算相关的密码学操作,并对衍生的定理进行了证明;并且提出了 4 个新的认证测试准则,能对协议中的加密、签名、身份生成和哈希等组件进行推理.其次,基于扩展的串空间模型对远程证明协议的安全属性(隐私性、机密性和认证性)进行了抽象和分析.最后,给出了对发现攻击的消息流程,并基于 ARM 开发板对其中的布谷鸟攻击进行了实现,验证了串空间的分析结果.

关键词 可信计算;远程证明;串空间;认证测试;形式化验证

中图法分类号 TP309 **DOI 号** 10.3724/SP.J.1016.2015.00701

Analyzing Trusted Computing Protocol Based on the Strand Spaces Model

FENG Wei[1),2)] FENG Deng-Guo[1)]

[1)](*Trusted Computing and Information Assurance Laboratory, Institute of Software Chinese Academy of Sciences, Beijing* 100190)

[2)](*University of Chinese Academy of Sciences, Beijing* 100190)

Abstract Trusted Computing technology can provide security supports for terminals, networks and cloud computing platforms; and its secure mechanisms or protocols should be proved formally and rigorously. This paper tries to analyze its remote attestation protocol based on the strand spaces model. Firstly, we extend the term algebra and penetrator's model in the strand spaces in order to express the cryptographic operations used in trusted computing, and the derived theorems have also been proved. We also propose four new authentication test principles accordingly, which can be used to reason about the encryption, signature, identity-making and hash components in the protocol. Then, we use the extended strand spaces model to describe and analyze the security properties (including privacy, secrecy and authentication) of remote attestation protocol. Finally, we describe the message flow of the attacks captured by the strand space analysis, and implement the cuckoo attack using ARM development board, which experimentally proved the results inferred from strand spaces model.

Keywords trusted computing; remote attestation; strand spaces; authentication test; formal verification

1 引 言

随着信息化技术的发展,网络变成了人们生活中不可或缺的一部分,很多安全操作也通过网络完成.例如,网络购物正在不断普及,人们已经接受通过网上支付的方式进行金融交易;公司或者政府机构经常通过内部网络管理各个终端,进行资料和数

收稿日期:2014-02-19;最终修改稿收到日期:2014-09-19.本课题得到国家自然科学基金(61202414,91118006)和国家"九七三"重点基础研究发展规划项目基金(2013CB338003)资助.**冯 伟**,男,1986 年生,博士研究生,主要研究方向为可信计算、网络与系统安全.E-mail:vonsky2010@hotmail.com.**冯登国**,男,1965 年生,博士,教授,博士生导师,主要研究领域为信息安全与密码学、可信计算与信息保障.

据的分发与共享(如 ftp 资源);云计算环境通过网络能为用户提供各种安全服务,如云存储和云租用平台等.这些安全操作通常涉及对敏感数据的处理,如用户的支付账号和密码,公司的机密文档和云隐私数据等.但是,由于网络环境的开放性和复杂性,使得其容易变成攻击者的目标:通过远程注入恶意软件,攻击者可以在网上交易时截获用户的支付账号和密码;恶意的终端可能连接公司内部网络,获取机密信息;客户存储在云中的数据可能面临机密性和完整性等威胁.

如果在安全操作执行之前,网络中的各个实体能相互验证平台身份和平台软件配置等信息,能有效防止攻击的发生.可信计算技术提出的可信平台模块 TPM (Trusted Platform Module)[1] 和远程证明协议①,能为网络实体提供基于硬件的可信证据,保证只有拥有合法身份并运行着预期软件的实体才能通过验证.因此,远程证明协议可以用于构建安全支付、可信网络接入[2] 和可信云服务[3] 等安全系统,防止机密信息和数据的泄露.安全支付中,远程证明协议可以向金融机构证明用户终端的软件配置,防止恶意软件截获支付信息;可信网络接入[2] 系统通过远程证明协议验证接入的终端设备,只允许合法的终端连接网络,可以有效保护内部网络资源;在可信云服务[3] 系统中,客户在上传数据之前,先使用远程证明协议验证云节点的身份和完整性状态,可以保证数据只存储在可信的云节点中.

由此可见,远程证明协议的应用很广泛,并对很多系统的安全性起关键作用.但是,目前远程证明协议并没有得到严格的形式化证明.如果协议本身存在安全隐患,将会严重威胁依赖该协议的安全系统.Coker 在文献[4]中指出了远程证明协议在灵活性方面的不足,提出了 5 条认证准则,并依此设计了一个灵活的认证体系;在文献[5]中进一步基于提出的认证体系设计了一个新的远程证明协议,称为 CAVES 协议.文献[6]对 CAVES 协议进行了形式化分析,基于串空间 CPSA 自动化工具[7-8] 证明了协议的安全性.但是,CAVES 协议及其分析存在如下几个问题:(1)没有考虑远程证明协议的平台身份密钥建立过程以及可信平台证书的关联性,遗漏了对隐私性的分析;(2)没有将安全芯片 TPM 独立成一个角色进行分析,容易忽视针对 TPM 本身的攻击;(3)CPSA 工具无法对远程证明协议依赖的一些 TPM 命令进行分析,如 TPM_Makeidentity 身份密钥生成、TPM_Extend 完整性扩展和 TPM_Quote 引证等操作.

针对上述问题,本文对安全芯片和证明者进行了区分,总结了一个完整的远程证明协议,包含平台身份密钥建立和平台证明两个过程;并对串空间理论和认证测试方法进行了扩展,使得其能描述安全芯片及其命令.本文的具体贡献如下:

(1)根据可信计算规范,本文对远程证明协议进行了完整的总结,通过消息流的形式给出了协议交互流程,方便对协议进行分析;并根据对 TPM 命令接口[1] 的理解,用密码原语对协议中的 TPM 命令进行了抽象描述.

(2)对串空间的消息代数和攻击者能力进行了扩展,证明了扩展模型范式引理的正确性,并基于扩展模型给出了远程证明协议的串空间定义;提出了 4 个认证测试准则,能对远程证明协议中的加密、签名、身份生成和哈希等组件进行推理分析.

(3)根据串空间扩展模型对远程证明协议的安全属性进行了定义,并基于认证测试准则逐一分析了这些安全属性;通过分析发现了一些攻击,并针对攻击提出了改进方法;最后基于扩展模型给出了对改进方法的正确性说明.

(4)对发现的攻击进行了详细的描述,并基于 ARM 开发板对其中的布谷鸟攻击进行了实现,从实验上验证了串空间的分析结果.

本文第 2 节给出可信计算和串空间方面的背景知识以及相关研究成果;第 3 节对远程证明协议进行总结,抽象协议交互流程,并提炼安全目标;第 4 节针对远程证明协议,给出扩展的串空间理论和新的认证测试方法,使得串空间模型能描述安全芯片的密码学操作;第 5 节基于扩展的串空间模型对远程证明协议进行分析;第 6 节详细描述发现的攻击,并给出攻击实验.

2 背景和相关工作

2.1 可信计算

可信计算技术[9] 通过在计算平台引入硬件安全芯片来保证计算机甚至网络环境的可信.这种可信是指计算机平台拥有合法的身份,并按照预期的软件配置执行.国际上,安全芯片以满足 TCG(Trusted Computing Group)规范的可信平台模块 TPM[1] 为

① TCG specification architecture overview, revision 1.4, 2007. http://www.trustedcomputinggroup.org/resources/tcg_architecture_overview_version_14

主，目前该规范已经升级到 TPM 2.0①. 国内安全芯片以满足国家密码管理局规范的可信密码模块 TCM②(Trusted Cryptography Module)为主，其采用我国自主知识产权的密码算法(如 SM2、SM3、SMS4 等)，并使用兼容我国公钥基础设施的双证书体系.

远程证明协议是可信计算的核心技术之一，其主要作用是允许计算机平台创建关于其身份和完整性状态的报告，该报告基于安全芯片生成，能被远程方所信赖. 平台身份由安全芯片保护的密钥提供，而完整性状态指对所有可执行程序或者软件进行密码操作后的度量值. 远程证明协议又称为完整性报告(Integrity Reporting)协议，TCG 规则中流程主要侧重平台完整性的报告，而忽视了平台身份密钥 AIK (Attestation Identity Key)的生成. 不过，文献[10]的图 2 和文献[11]的图 5 分别给出了平台身份密钥 AIK 的创建流程和具体 TPM 命令调用情况. 本文依据这些文献，对远程证明协议的完整流程和密码细节进行了总结.

可信计算的形式化分析是一种重要技术. 一般形式化方法由于缺少对 TPM 相关命令的描述而难以用于分析可信计算的协议. 文献[5-6]使用串空间 CPSA 分析工具对远程证明协议进行了研究，设计并证明了一个新的 CAVES 协议. 不过该研究缺少对平台身份密钥建立的分析，也没有对安全芯片单独进行建模；因此，其分析结果不能全面阐述远程证明协议的安全性. 本文通过扩展 CPSA 工具依赖的串空间理论和认证测试准则，对可信计算远程证明协议进行了全面的分析. 文献[12]提出了一个形式化基础和框架，能对可信平台进行模型检测分析，发现了可信平台的一些逻辑缺陷. Chen 等人[13]为 TPM 2.0 设计并实现了新的数字签名方法，并在 SDH 假设和随机预言机模型下证明了 TPM 签名原语的安全性. Kremer 和 Kunnemann[14]提出了新的形式化分析工具，能处理协议中的非单调全局状态，适合对 TPM 的 API 和 PCR 寄存器进行分析. 文献[15]首次提出了可以处理 Diffie-Hellman 幂运算、双线性映射和 AC-运算符的符号分析算法，并基于 tamarin 进行了实现，该方法有助于分析可信计算的直接匿名认证协议.

2.2　串空间

串空间模型建立在 Dolev-Yao 基础[16]之上，该模型假定密码系统本身是完好的，攻击者无法进行密码分析，不过攻击者可以完全控制网络，甚至能使用自己的身份参与到协议会话中. 串空间的发展大致经历了 3 个阶段：串空间理论[17]、认证测试方法[18-20]和自动化分析工具 CPSA[7-8]. 串空间理论是纯手工推理，对于一个简单协议(如 NS 协议)的分析，其推理和证明过程都比较复杂，容易出错；认证测试方法提炼出了一些高级测试准则，能直接针对协议的消息组件推理认证属性，并且对所有协议是通用的，简化了手工推理的复杂性，降低了分析的难度；CPSA 工具进一步基于认证测试方法给出了对协议的自动推理，能以图的方式输出协议所有可能的执行流程，安全属性需要根据所有输出的图进行分析判断，因此，CPSA 实际上是一个半自动化工具，不过其避开了复杂的串空间理论，分析者容易上手，而且也提高了协议分析的效率.

串空间通过图的方式，能比较直观的描述协议的执行或者攻击流程，已经被成功用于分析 CAVES 认证协议[5-6]、TLS 安全通道[21-23]和 IKE 密钥交换[24]等. 文献[21]中 Kamil 为了分析 TLS 协议对串空间进行了扩展，其扩展方法为本文的扩展研究提供了参考思路. 由于串空间的消息代数、攻击者串和认证测试方法描述能力有限，在分析一些复杂协议时可能遇到障碍，文献[21]和本文的方法提供了一种通用的扩展思路，能增强串空间的描述能力，使其能用于对更多协议进行形式化分析.

Guttman[25]对串空间模型进行了扩展，使得其能分析公平交换协议. Ramsdell[26]在 CPSA 工具的基础上提出了形状分析语句，能够提取一阶逻辑中的语句来描绘 CPSA 的运行，并通过逻辑推理的方式来确定协议的安全目标. 串空间的协议主体维护的本地状态在各个协议会话中是独立的，在处理跨会话的状态时会遇到困难，于是 Ramsdell 等人[27]提出了一种混合分析方法，使用定理证明来推理基于状态的计算，并使用 CPSA 来处理消息传递部分. 基于该混合分析方法，Ramsdell 对使用 TPM API 访问 PCR 寄存器的情况进行了建模分析. Guttman 在文献[28]中基于串空间理论提出了新的标号转换系统分析方法.

3　可信计算远程证明协议简介

完整的远程证明协议 RA(Remote Attestation)

① Trusted Platform Module Library, Family "2.0", 2014. http://www.trustedcomputinggroup.org/resources/tpm_library_specification

② Functionality and Interface Specification of cryptographic supporting platform for trusted computing, 2007. http://www.oscca.gov.cn/UpFile/File64.PDF

分为两个阶段:平台身份密钥建立[10-11]和平台证明过程[1,5].前者基于 PrivacyCA[10]产生一个平台身份密钥,在认证平台身份的同时保护用户的隐私;后者基于平台身份密钥向远程方证明平台的软件状态.本节首先对这两个阶段进行总结,最后给出协议的安全目标.

3.1 平台身份密钥建立

安全芯片 TPM[1]使用背书密钥 EK(Endorsement Key)作为其唯一标识,其私钥长期保存在 TPM 芯片内部(不会泄露),公钥由芯片厂商签名生成 EK 证书.由于 EK 可以唯一标识一个 TPM 及其所在的硬件平台,通常无法作为平台身份使用;否则,会泄露平台隐私.因此,TCG 使用安全芯片生成的身份密钥 AIK 代替 EK 作为平台身份.为了证明 AIK 属于一个合法的 TPM 芯片,同时不泄露具体是哪个 TPM,TCG 规定采用可信第 3 方 PrivacyCA 来认证 AIK 并生成 AIK 证书.通过 AIK 证书,只有 PrivacyCA 能确定 TPM 所在的硬件平台,而其他验证者无法追踪到该平台,因此平台隐私得到保护.

平台身份密钥 AIK 的建立过程见消息 1~6.具体包含 3 个主要的角色:安全芯片 TPM/TCM(T),用户进程(U)和可信第 3 方 PrivacyCA(PCA):

消息 1. U→T: L, PCA

消息 2. T→U: AIK, I

消息 3. U→PCA: Cert_EK, AIK, L, PCA, I

消息 4. PCA→U: $\{\text{Cert\_AIK}\}_{EK}$

消息 5. U→T: $\{\text{Cert\_AIK}\}_{EK}$

消息 6. T→U: Cert_AIK

消息 1 和 2:用户选取 1 个身份标签 L 和指定可信第 3 方 PCA,通过调用 TPM_MakeIdentity 命令[24],由 TPM 芯片生成 1 个新鲜的 AIK 密钥对,将私钥 AIK^{-1} 加密保存在 TPM 内部,而将公钥 AIK 和 TPM 生成的身份内容 I 返回给用户.注意,L 只是 1 个标签,由用户自由选取,可以用于查找 AIK 证书,而无法作为安全芯片或者用户的唯一标识名称使用.我们使用如下函数来表示身份内容 I 的生成:

$$I=[AIK, L, PCA]_{SK(T)}$$

该函数通过安全芯片 T 的 AIK^{-1} 对 AIK,L 和 PCA 进行签名,表示这些数据与该安全芯片 T 关联(这里 $SK(T)=AIK^{-1}$).

消息 3 和 4:用户进程将公钥 AIK 以及签名的身份内容 I,连同 EK 证书一起发送给可信第 3 方 PCA.PCA 验证 EK 证书的有效性后,用其私钥 SK(PCA)对 AIK 公钥进行签名生成 AIK 证书.PCA 使用 EK 加密保护 AIK 证书,将加密包返回给用户.规范中,PCA 会生成一个临时对称密钥 K 来加密保护 AIK 证书,而使用 EK 加密 K,从安全角度而言,与直接使用 EK 加密 AIK 证书一样.为了方便分析,本文省去 K,采取简化的形式.

我们使用 $[p, PK(p), v]_{SK(v)}$ 来表示一个证书,p 是证书拥有者,$PK(p)$ 是 p 的公钥,v 是签发者,$SK(v)$ 是 v 的私钥.因此,EK 证书和 AIK 证书可以分别表示如下(MF 表示生产安全芯片 T 的厂商):

$$\text{Cert\_EK}=[T, EK, MF]_{SK(MF)}$$

$$\text{Cert\_AIK}=[L, AIK, PCA]_{SK(PCA)}$$

如果攻击者拥有相应的私钥,也可以产生伪造的证书,因此具体分析时,采用如下形式来描述证书:

$$[p, k_p, v]_k$$

基于一些假设(如 v 可信且 $k=SK(v)$ 是安全密钥),需要证明 $k_p=PK(p)$.

消息 5 和 6:用户进程将 EK 生成的加密包发送给安全芯片 T,并通过 TPM_ActivateIdentity 命令来激活 T 内部的新身份密钥 AIK,同时使用 T 内部的 EK 私钥解密获得相应的 AIK 证书.

3.2 平台证明过程

平台证明过程主要是向远程方证明本地平台的身份和完整性状态.身份由 AIK 表示,完整性状态由安全芯片的平台配置寄存器 PCR(Platform Configuration Register)保证.通过 TPM_Quote 命令[1],AIK 可以对 PCR 进行签名,签名的结果可以作为远程方依赖的证据.平台证明过程包含 3 个主要角色:安全芯片 TPM/TCM(T),用户进程 Attester(U)和远程验证者 Verifier(V).具体协议流程如下:

消息 7. V→U: N_V

消息 8. U→T: N_V

消息 9. T→U: Q

消息 10. U→V: Q, ML, Cert_AIK

首先,远程验证方 V 产生一个新鲜的随机挑战值 N_V,向本地证明者 U 请求可信证据(消息 7).U 将随机值转发给安全芯片 T(消息 8).然后 T 使用 TPM_Quote 命令对 PCR 值和 N_V 进行签名,并将签名结果 Q 返回给用户进程 U(消息 9).最后 U 将 Q、AIK 证书和度量日志 ML 一起发送给 V(消息 10).V 通过验证这些值,可以确定 U 所在平台的可信状态:AIK 确定平台是否拥有合法的安全芯片;度量日志 ML 确定平台是否运行预期的软件;Q 关

联 AIK 和 ML，并保证 ML 的完整性.

度量日志 ML 记录了用户计算平台的所有可执行程序及其度量值(程序二进制代码的哈希值). PCR 是这些度量值的一个哈希聚集值，采用哈希链(HashChain)的方式生成，即

$$\text{PCR\_New}=Hash(\text{PCR\_Old}\parallel m)$$

m 是一个新的度量值，PCR_Old 是旧的 PCR 值，而 PCR_New 是扩展(通过 *TPM_Extend* 命令) m 后的新 PCR 值. PCR 的主要目的是保证 ML 的完整性. 我们采用如下两个函数来表示 Q 的生成：

$$\text{PCR}=\text{HashChain}(\text{ML})$$

$$\text{Q}=[\text{PCR},\ \text{N}_\text{V}]_{SK(\text{T})}$$

HashChain 是度量日志 ML 的一个完整性标识，拥有密码哈希函数的属性，为了简化，具体分析时将 HashChain 作为一个哈希函数处理. Q 是一个基于 T 内部 AIK 私钥的数字签名.

3.3　安全目标

根据远程证明协议的相关工作[4-5,10]以及其在安全系统(安全支付、可信网络接入和可信云服务等)中的应用，本文总结了如下 5 个安全目标：

(1) 隐私性. 平台证明过程中，验证者可以确定一个合法的 TPM 芯片，但是无法确定具体是哪个 TPM 芯片. 简单的说，Cert_EK 只能提供给可信第 3 方 PrivacyCA，而其他方需要的身份认证性由 Cert_AIK 提供；Cert_EK 会泄露具体的 TPM 芯片，而 Cert_AIK 不会.

(2) PrivacyCA 认证性. 作为可信第 3 方的 PrivacyCA 能够验证具体 TPM 芯片的身份合法性(基于 Cert_EK)，只有确定其合法后才会给其签发 AIK 证书(Cert_AIK).

(3) Verifier 认证性. 平台证明过程中，验证者 Verifier 能确定一个合法的证明者平台(包含安全芯片 T 以及用户进程 U)，能同时验证平台的完整性状态.

(4) ML 的机密性. 度量日志 ML 记录了证明者平台的所有软件配置，通常会包含该平台的一些敏感信息，不能泄露.

(5) 用户的认证性. 用户能确保其与一个合法的验证者 Verifier 交互，而不是在与一些恶意方交互，防止平台的可信数据被恶意攻击者获取.

4　串空间扩展模型

本节主要给出串空间理论[17]以及我们为了描述远程证明协议所作的一些扩展. 同时，对扩展模型中衍生的一些定理和准则进行了证明.

4.1　串空间基础

4.1.1　消息代数

设 A 是协议执行中所有消息的集合，A 中的元素称为消息项(term). 原始串空间理论[17]定义了两类原子消息项：一类是明文消息项(如参与者标识、随机数等)，构成集合 M；另一类是密钥消息项(如对称密钥、加密密钥等)，构成集合 K. 为了描述远程证明协议的安全芯片，我们增加了一个集合 X，表示所有安全芯片的集合.

原始串空间[17]在消息集 A 上定义了两个二元运算(加密运算 encr 与级联运算 join)、一个一元运算(密钥求逆 inv). 可信计算安全芯片对密钥有严格的分类，如 EK 只能用于加密，而 AIK 只能用于签名. 因此，我们将 K 划分为 3 个不相交子集：加密密钥 K_{enc}、签名密钥 K_{sig} 和对称密钥 K_{sym}. 为了描述普通签名消息，增加了一个二元运算 sig. 为了描述身份消息，增加了一个二元运算 mid. 为了描述 *PCR* 值，增加一个 hash 运算. 扩展的消息集合和运算定义如下.

定义 1.　消息集合 A 中的原子消息项是 M，K 和 X 中的元素(其中 M，K 和 X 两两互斥)；复合消息项由如下运算生成：加密 encr：$K_{\text{enc}}\times A\rightarrow A$；签名 sig：$K_{\text{sig}}\times A\rightarrow A$；级联 join：$A\times A\rightarrow A$；身份生成 mid：$X\times A\rightarrow A$；哈希 hash：$H\times A\rightarrow A$(其中 H 是所有哈希函数的集合).

根据习惯，记 encr(k,m) 为 $\{m\}_k$，记 sig(k,m) 为 $[m]_k$，记 join(a,b) 为 $a\parallel b$，记 mid(T,m) 为 $[m]_T$，记 hash(h,m) 为 $h(m)$.

密钥的逆由 inv：$K\rightarrow K$ 进行定义，记 $\text{inv}(k)=k^{-1}$. 如果 $k\in K_{\text{sym}}$，则 $k=k^{-1}$. 对于主体名称 $a\in M$，$PK(a)$ 和 $SK(a)$ 分别表示主体 a 的公钥和私钥. 如果一个非对称加密密钥 $k\in K$ 由安全芯片 $T\in X$ 生成，则私钥记为 $k=SK_{\text{enc}}(T)$，公钥记为 $k^{-1}=PK_{\text{enc}}(T)$. 同理 T 中有非对称签名密钥 $SK_{\text{sig}}(T)$ 和验证密钥 $PK_{\text{sig}}(T)$. $SK_{\text{enc}}(T)$ 和 $SK_{\text{sig}}(T)$ 处于安全芯片 T 的硬件保护，永不泄露，统一记为 $SK(T)$. 注意，一般主体 a 并不区分签名密钥和加密密钥，而安全芯片 T 对密钥类型有严格的限制.

本文针对远程证明协议的变量约定如下：$T\in X$；$U,V,PCA,MF,L,ML,N_{\text{V}}\in M$；$AIK,AIK^{-1}\in K_{\text{sig}}$；$EK,EK^{-1}\in K_{\text{enc}}$. 如果 AIK 和 EK 属于某个安全芯片 T，则 $AIK=PK_{\text{sig}}(T)$，$AIK^{-1}=SK_{\text{sig}}(T)$，

$EK=PK_{enc}(T)$,$EK^{-1}=SK_{enc}(T)$.

消息项之间存在子项关系($\sqsubseteq$),定义如下.

定义 2. 子项关系$\sqsubseteq$递归定义如下:$a\sqsubseteq a$;若 $a\sqsubseteq b$,则 $a\sqsubseteq\{b\}_k$,$a\sqsubseteq[b]_k$,$a\sqsubseteq[b]_T$;若 $a\sqsubseteq b$ 或者 $a\sqsubseteq c$,则 $a\sqsubseteq b\parallel c$. 这里 $a,b,c\in A$,$k\in K$,$T\in X$. 如果 $a\sqsubseteq b$ 且 $a\neq b$,则称 a 为 b 的真子项.

注意,子项关系的意义为:给定一些密钥和安全芯片的知识集合,消息项 m 的子项可以很容易从 m 中提取出来. 显然,$h(a)$中的 a 以及$\{b\}_k$,$[b]_k$,$[b]_T$中的 k 和 T 并不属于子项,这是由完美密码系统(Dolev-Yao 假设)的安全性质决定的,如哈希是不可逆的,密文不会泄露密钥信息等.

定义 3. 如果 $a\sqsubseteq b$,且 a 不是一个级联项($g\parallel h$ 的形式),那么称 a 为 b 的一个组件,b 由 a 与任意的消息项级联而成. 因此,组件可以是一个原子项,加密项,哈希项,签名项或者身份项.

组件可以将一个非级联形式的消息项与协议的某个执行主体进行关联. 如只有拥有签名私钥的主体才能产生某个签名项,只有拥有加密私钥的主体才能解密某个加密项等. 组件是串空间认证测试方法[18-20]提出的一个非常重要的概念.

4.1.2 串、串空间和丛

协议参与者可以发送消息项,也可以接受消息项. 在串空间中,使用正号表示发送,负号表示接受. 串(strand)是指协议参与者发送和接受消息项的一个序列. 串空间(strand space)是所有串的集合.

定义 4. $\langle\sigma,a\rangle$是一个带符号项,其中 $\sigma\in\{+,-\}$,$a\in A$. 一个带符号项记为 $+a$ 或者 $-a$. $(\pm A)^*$ 是所有带符号项的有限序列集合,其一个典型的元素可以记为$\langle\langle\sigma_1,a_1\rangle,\cdots,\langle\sigma_n,a_n\rangle\rangle$. A 上的一个串空间是一个集合 Σ,该集合具有一个轨迹映射 $tr:\Sigma\rightarrow(\pm A)^*$. 对一个固定的串空间 Σ,满足如下性质:

(1) 节点(node)n 是$\langle s,i\rangle$,其中 $s\in\Sigma$,i 是一个整数,满足 $1\leq i\leq length(tr(s))$. 节点的集合为 N,每个节点 $n=\langle s,i\rangle$属于唯一一个串 s.

(2) 如果节点 $n=\langle s,i\rangle\in N$,那么 $index(n)=i$,$strand(n)=s$,$term(n)=(tr(s))_i$,即串 s 中的第 i 个带符号消息项.

(3) 如果 $n_1,n_2\in N$,则 $n_1\rightarrow n_2$ 表示 $term(n_1)=+a$,$term(n_2)=-a$,即 n_1 发送一个消息 a 到 n_2,给出了串的因果关系.

(4) 如果 $n_1,n_2\in N$,则 $n_1\Rightarrow n_2$ 表示 n_1 和 n_2 出现在同一个串中,且 $index(n_2)=index(n_1)+1$,即 n_2 是 n_1 的直接后继,n_1 是 n_2 的直接前驱. $n_1\Rightarrow^+ n_2$ 表示 n_1 是 n_2 的前驱(不一定是直接前驱),可以经过一个或者多个$\Rightarrow$边到达 n_2.

(5) 一个无符号项 t 发生在 $n\in N$ 中,当且仅当 $t\sqsubseteq term(n)$,即 t 是节点 n 的消息项的子项.

(6) 设 I 是无符号项的集合,节点 $n_1\in N$ 是 I 的入口点,当且仅当对于某个 $t\in I$,$term(n_1)=+t$ 并且对于任意 $n_2\Rightarrow^+ n_1$,$term(n_2)\notin I$.

(7) 一个无符号项 t 源于一个节点 n,当且仅当 n 是集合 $I=\{t_0\mid t\sqsubseteq t_0\}$的入口点.

(8) 一个无符号项 t 唯一源于一个节点集合 $S\subset N$,当且仅当存在唯一的节点 $n\in S$ 满足 t 源于 n. 通常一个唯一源于节点的消息项可以表示一个随机数 nonce 或者会话密钥.

(9) 一个无符号项 t 非源于一个节点集合 $S\subset N$,当且仅当不存在节点 $n\in S$ 满足 t 源于 n. 通常长期安全密钥都是非源于的.

(10) 如果消息项 t 是 $term(n)$的一个组件,并且对于每个 $n_0\Rightarrow^+ n$,t 不是 $term(n_0)$的组件,那么称 t 是节点 n 的一个新组件.

基于串和串空间的概念,下面给出远程证明协议正规参与者串的定义. 基于第 2 节的描述,协议主要有 4 个参与者角色 T,U,V 和 PCA,同时增加 1 个厂商角色 MF 来生成 EK 证书. 记 EK 证书为 $cert(EK)=[T\parallel EK\parallel MF]_{SK(MF)}$,AIK 证书为 $cert(AIK)=[L\parallel AIK\parallel PCA]_{SK(PCA)}$,安全芯片 T 生成的身份为 $I=[AIK\parallel L\parallel PCA]_T$,安全芯片生成的 Quote 签名为 $Q=[h(ML)\parallel N_V]_{inv(AIK)}$.

注意,这里 I 和 Q 的表示不一致,因为 I 的生成需要安全芯片内部 EK 的参与,只能由安全芯片 T 进行;而 Q 的签名可能由软件生成(基于伪造的密钥),不一定在 T 内部进行.

定义 5. 设 $TPM[T,U,V,PCA,MF,L,AIK,EK,ML,N_V]$是安全芯片串的集合,其轨迹(trace)为$\langle -L\parallel PCA,+AIK\parallel I,-\{cert(AIK)\}_{EK},+cert(AIK),-N_V,+Q\rangle$.

定义 6. 设 $User[T,U,V,PCA,MF,L,AIK,EK,ML,N_V]$是用户进程串的集合,其轨迹(trace)为$\langle +L\parallel PCA,-AIK\parallel I,+L\parallel PCA\parallel AIK\parallel I\parallel cert(EK),-\{cert(AIK)\}_{EK},+\{cert(AIK)\}_{EK},-cert(AIK),-N_V,+N_V,-Q,+ML\parallel cert(AIK)\parallel Q\rangle$.

定义 7. 设 $Verifier[T,U,V,PCA,L,AIK,ML,N_V]$是远程验证者串的集合,其轨迹为$\langle +N_V,-ML\parallel cert(AIK)\parallel Q\rangle$.

定义 8. 设 $PrivacyCA[T,PCA,MF,L,AIK,$

$EK]$是隐私证书权威串的集合，其轨迹为$\langle -L \parallel PCA \parallel AIK \parallel I \parallel cert(EK), +\{cert(AIK)\}_{EK}\rangle$.

定义 9.　设 $Manufacturer[MF,T,EK]$是安全芯片厂商串的集合，其轨迹为$\langle +cert(EK)\rangle$.

为了简化分析过程，假定在远程证明协议之前，厂商颁发的 EK 证书已经被用户进程和 PrivacyCA 得到. 同时，使用星号（$*$）作为串参数的通配符，如 $TPM[T,U,V,PCA,MF,L,AIK,*,*,N_V]$表示 EK 和 ML 可以取任意值.

结点集合 N 和边 $n_1 \rightarrow n_2$ 及 $n_1 \Rightarrow n_2$ 的集合构成一个有向图$\langle N, (\rightarrow \cup \Rightarrow)\rangle$. 一个丛(bundle)通常定义为$\langle N,(\rightarrow \cup \Rightarrow)\rangle$的一个子图，丛表示协议的一个可能执行，其边解释了节点之间消息项的因果关系. 下面根据原始串空间的描述给出关于丛的定义和定理，定理的正确性可以参考 Thayer[17].

定义 10.　1 个丛 $C=\langle N_C,(\rightarrow_C \cup \Rightarrow_C)\rangle$是$\langle N,(\rightarrow \cup \Rightarrow)\rangle$的子图，其中$\rightarrow_C \subseteq \rightarrow$，$\Rightarrow_C \subseteq \Rightarrow$. 丛 C 满足下列性质：(1) C 是一个非空的有限无环图；(2) 如果 $n_1 \in N_C$，且 $term(n_1)$是负号，则存在唯一的 n_2 满足 $n_2 \rightarrow n_1$ 是 C 中一条边；(3) 如果 $n_1 \in N_C$，$n_2 \Rightarrow n_1$，则 $n_2 \Rightarrow_C n_1$.

例如，图 1 给出了可信计算远程证明协议正常执行一次的丛.

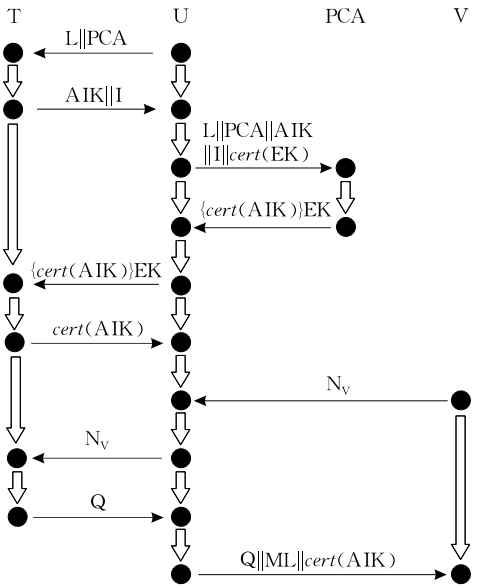


图 1　远程证明协议正常执行的丛

定义 11.　1 个节点 $n \in N_C$，那么称该节点在丛 $C=\langle N_C,(\rightarrow_C \cup \Rightarrow_C)\rangle$中，记为 $n \in C$. 对于 1 个串 s，满足$\langle s,i\rangle \in C$ 的最大正整数 i 称为串 s 在 C 的丛高（C-height）.

定义 12.　如果 C 是一个丛，定义关系$\leq_C = (\rightarrow_C \cup \Rightarrow_C)^*$，其表示节点之间的因果关系.

定理 1.　对于一个丛 C，关系$\leq_C$是一个偏序关系，即满足自反、反对称和传递性质. 丛 C 中节点的每个非空子集都存在$\leq_C$-最小元.

4.1.3　攻击者能力

串空间基于 Dolev-Yao 模型[16]，攻击者能力由其初始知识（初始密钥集合 K_P 以及其他初始原子消息集合 M_P），和攻击者能执行的动作（称为攻击者串）来决定. 本文对串空间的消息代数进行了扩展，对攻击者能力也进行了相应的扩展. 主要增加了与签名、身份生成和哈希相关的攻击者串，即定义 13 的 SI，VE，MI 和 H 串.

定义 13.　攻击者串的轨迹可以是如下的形式之一：

(1) M. 散发明文消息：$\langle +m\rangle$，$m \in M_P$.

(2) K. 散发密钥：$\langle +k\rangle$，$k \in K_P$.

(3) C. 级联消息：$\langle -g,-m,+g \parallel m\rangle$.

(4) S. 拆分消息：$\langle -g \parallel m,+g,+m\rangle$.

(5) E. 加密消息：$\langle -k,-m,+\{m\}_k\rangle$，$k \in K_{enc}$.

(6) D. 解密消息：$\langle -k^{-1},-\{m\}_k,+m\rangle$，$k \in K_{enc}$.

(7) SI. 签名消息：$\langle -k,-m,+[m]_k\rangle$，$k \in K_{sig}$.

(8) VE. 验证消息：$\langle -k^{-1},-[m]_k,+m\rangle$，$k \in K_{sig}$.

(9) MI. 身份生成：$\langle -T,-m,+[m]_T\rangle$，$T \in X$.

(10) H. 哈希：$\langle -m,+h(m)\rangle$，$h \in H$.

我们假定只有安全芯片才能生成平台身份信息，不过，任何攻击者只要能控制一个安全芯片，其也能生成平台身份信息，因此增加了攻击者 MI 串. 读取安全芯片 PCR 值不需要任何授权，Quote 签名只要拥有私钥就可以进行，因此增加了 SI 串描述攻击者伪造签名或者证书的能力. 只要能获取平台度量日志 ML，攻击者也能生成其完整性标识 h(ML)，因此增加了攻击者 H 串.

如果 1 个节点位于攻击者串，则称其为攻击者节点；否则，称为正规节点. 通常一个渗透的串空间既包含正规串（所有合法参与者的串），也包含这里定义的攻击者串.

4.2　范式引理

攻击者行为可以通过攻击者串的任意组合来实现. 不过，这会导致很多冗余的丛，使得协议的分析变得复杂. 在认证测试方法中，Guttman 等人[18-19]提出了丛等价和范式引理来限制攻击者串的组合顺序，可以在不消弱攻击者能力的情况下，删除攻击者的一些冗余行为. 由于我们对攻击者串进行了扩展，

引入了新的冗余情况，本节主要给出冗余消除方法以及证明扩展后的范式引理依然成立.

定义 14. 串空间 Σ 中的两个丛 C_1 和 C_2 等价，当且仅当它们拥有相同的正规节点.

定义 15. 丛中包含 4 种类型的冗余：E-D 冗余和 C-S 冗余(参考认证测试[18-19])；SI-VE 冗余和 MI-VE 冗余(如图 2 所示).

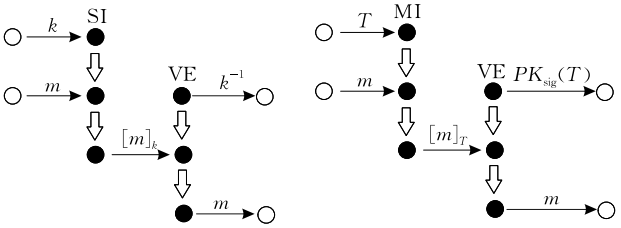


图 2 SI-VE 冗余和 MI-VE 冗余

定理 2. 每个丛 C_1 都存在一个没有任何冗余的等价丛 C_2.

证明. 关于 E-D 冗余和 C-S 冗余的消除方法见参考文献[19]. 而关于 SI-VE 冗余和 MI-VE 冗余的消除如图 3 所示. 通过删除冗余中的攻击者串 VE，并增加相应的边(虚线箭头表示)，删除冗余后，正规节点没有变化，而且新生成的图满足丛的定义(定义 10). 因此，根据定义 14，删除冗余生成的丛与原始丛等价. 证毕.

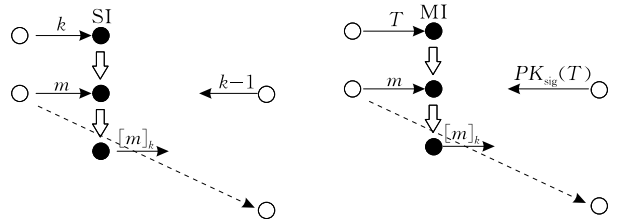


图 3 SI-VE 冗余和 MI-VE 冗余消除方法

攻击者的行为在认证测试中通过一条路径(path)来描述. 一条攻击者路径中，除了首尾节点，所有其他节点都是攻击者节点. 为了限制攻击者路径的形式，使用构造边(constructive edge)和析构边(destructive edge)的概念来对攻击者串进行分类. 由于引入了新的攻击者串，我们对构造边和析构边进行了重新定义.

定义 16. 如果一个 $\Rightarrow^+$ 边是 E,C,SI,MI 或者 H 串的一部分，称其为构造边；如果其是 D,S 或者 VE 串的一部分，称其为析构边. 如果 1 个攻击者节点是 K 或者 M 节点，称该节点为初始节点(initial).

定义 17. 如果对于丛 C 中的每条攻击者路径，每个析构边都领先于(precede)每个构造边，则称该丛是正常丛(normal bundle).

定理 3. 攻击者范式引理. 每个丛 C_1 都存在 1 个等价的正常丛 C_2.

证明. 没有 SI,MI,VE 和 H 串的范式引理在文献[19]中已经得到证明. 我们主要考虑扩展的攻击者串的影响. 由于哈希函数的不可逆特性，H 串的输出无法传给一个析构边. 而对于 SI 串，其输出只能传给 D 串或者 VE 串，但是由于 K_{sig} 和 K_{enc} 是不相交的，D 串无法处理签名的形式，故 SI 串的输出只能传给 VE 串. MI 串中，$SK_{sig}(T)$ 是签名密钥，故其输出也只能传给 VE 串. 而对于 VE 串，其只接受签名类型的密钥和消息，故其输入无法由 E,C 和 H 串提供. 因此，新增加的攻击者串除了 4 个冗余外，没有引入新的构造边立即领先于析构边的情形. 而根据定理 2，4 个冗余消除后可产生等价丛. 证毕.

4.3 认证测试方法

本节介绍的技术可以用于证明协议的机密性和认证性. 首先定义安全消息，然后提出 4 个认证测试准则. 由于对串空间模型进行了扩展，本节的定义和准则与原始串空间不同.

4.3.1 安全消息

在扩展的串空间模型中，1 个密钥是安全的应该至少满足如下 3 点：首先，该密钥不能在攻击者的初始知识集合中；其次，协议消息交互时只会出现在加密的形式或者哈希的形式中；最后，如果其存储在安全芯片中，且不会在安全芯片外部使用，那么该密钥是安全的. 下面采用递归的方式定义安全密钥.

定义 18. 安全密钥. 设 S_0 是满足如下条件的密钥 k 的集合：(1) $k\notin K_P\cup M_P$ 或者 $k\in\{SK(T)\mid T\in X\}$；(2) 不存在正号正规节点 $n\in\Sigma$ 和消息项 t，满足 t 是 n 的一个新组件且 $k\subseteq t$. 设 S_{i+1} 是满足如下条件密钥 k 的集合：(1) $k\notin K_P\cup M_P$ 或者 $k\in\{SK(T)\mid T\in X\}$；(2) 对于每个正号正规节点 $n\in\Sigma$ 及其新组件 t，k 在 t 中的发生只能是在加密($\{\cdots k\cdots\}_{k0}$，$k0^{-1}\in S_i$)或者哈希($h(\cdots k\cdots)$)的保护形式中. 设 $S=\cup_i S_i$，如果 $k\in S$，那么 k 在串空间 Σ 中是安全密钥.

定义 19. 首先安全密钥是安全消息；其次如果消息 m 在串空间 Σ 中只发生在安全密钥加密保护的消息项或者哈希项 $h(m)$ 中，则 m 也为安全消息.

定理 4. 设 C_0 是一个丛，r 是一个原子项，且满足 r 是一个安全密钥或者安全消息，那么对于所有的等价丛 C_1，不存在满足 $term(n)=r$ 的节点 n.

定理 4 的证明在文献[21]中已经给出，只需要增加一点，即如果安全密钥或者安全消息 r 长期存

储在安全芯片 T 中，则显然消息项中不会出现 r.

通常协议的机密性可以通过证明相关密钥在 S_i 中来获得，对于大部分协议，$i=0$ 或者 1；或者通过证明相关消息是安全消息（如后面度量日志 ML 的机密性分析）.

4.3.2　认证测试

认证测试是基于串空间理论提炼出来的高级推理准则，该准则使得安全协议的认证性分析变得简单. 认证测试[18-20]阐述了丛中推理正规节点和正规串的条件，主要包含传出测试（outgoing test）和传入测试（incoming test）. 安全协议通常通过挑战/响应的方式来获得认证性，认证测试是基于该事实建立的. 本节给出并证明扩展的串空间模型的认证测试方法.

加密测试. 设 n_0 和 n_1（$n_0 \Rightarrow^+ n_1$）是丛 C 的两个节点，a 唯一源于节点 n_0，且 a 在 n_0 处只发生在其组件 $t_0=\{h\}_k$，$k^{-1}\in K_{\text{enc}}\cap S$ 中，同时 a 在 n_1 处只发生在其一个新组件 t_1 中. 那么，一定存在一个正号正规节点 m_1 满足 t_1 源于 m_1；而且存在一个节点 m_0 满足 $a\sqsubseteq term(m_0)$，且 $n_0\leq_C m_0\Rightarrow^+ m_1\leq_C n_1$.

该加密测试与传出测试一致，证明过程已经在文献[9]中给出. 该测试主要说明只有拥有安全私钥的正规串才能解密 a，并生成 a 的一个新组件.

签名测试. 设 n 是丛 C 的一个负号节点，$t=[h]_k\sqsubseteq term(n)$ 是 n 的一个新组件且满足 $k\in K_{\text{sig}}\cap S$. 那么，一定存在一个正规节点 $m\leq_C n$ 满足 t 源于 m.

证明.　设集合 $D=\{n_0\in C\mid t\sqsubseteq term(n_0)\}$. 显然 D 是一个非空集合，因为 $n\in D$，由定理 1 可知，D 存在 $\leq_C$-最小元，记为 m. 下面采用反证法，假设 m 是一个攻击者节点，由 $t\sqsubseteq term(m)$ 可以推出 m 只能是 SI 串上的正号节点，其密钥边为 k，这显然与 $k\in S$ 矛盾. 因此，m 只能是一个正规节点. 而且 m 是 D 的最小元，由定义 4 可知 t 源于 m.　　证毕.

身份测试. 设 n 是丛 C 的一个负号节点，$g=[h]_T\sqsubseteq term(n)$ 是 n 的一个新组件且满足 $T\in X$，$SK_{\text{sig}}(T)\in K_{\text{sig}}\cap S$. 那么，一定存在一个正规节点 $m\leq_C n$ 满足 g 源于 m.

证明.　身份测试类似签名测试，其证明过程与签名测试类似. 不过身份测试主要用来确认一个合法安全芯片 T 的存在.　　证毕.

哈希测试. 设 n 是丛 C 的一个节点，$t=h(g)\in term(n)$ 是其一个新组件，且 g 是安全消息. 则一定存在 1 个正规节点 $m\leq_C n$ 满足 t 源于 m.

证明.　设集合 $D=\{n_0\in C\mid t\sqsubseteq term(n_0)\}$. 显然 D 是一个非空集合，因为 $n\in D$，由定理 1 可知，D 存在 $\leq_C$-最小元，记为 m. 如果 m 是一个攻击者节点，则只可能发生在 H 串中，而 m 是安全消息，攻击者无法生成 $h(m)$，否则违背了哈希函数的抗碰撞攻击特性. 因此，m 只能是一个正规节点.　　证毕.

4.3.3　机密性和认证性分析

一个正规串表示协议执行中一个合法参与者的本地视图，即该参与者发送和接受消息的所有行为. 串空间分析的思路是从一个合法参与者的角度出发，推断其他参与者串和可能的攻击者行为.

认证性是推断其他合法参与者已经执行的行为. 推断过程依赖 4 个方面的元素[19]：该合法参与者对应的正规串、协议的规范描述（即所有定义的正规串）、攻击者能力和一些源假设（如唯一源于的随机数和非源于的安全密钥等）. 丛中定义的因果关系可以用来推理其他串的轨迹. 一个典型的认证性定义如下.

定义 20.　认证性. 对于所有的丛 C 和串 s_0，如果 s_0 在 C 中的丛高为 i，并且一些源假设成立，那么丛 C 中存在一个丛高为 j 的正规串 s_1.

机密性是推断攻击者不可能执行的行为（如无法获取某个值），一个典型的机密性定义如下.

定义 21.　机密性. 对于所有的丛 C 和所有正规串 s，如果 s 丛高为 i，并且一些源假设成立，那么不存在任何节点 $n\in C$ 使得 $term(n)=t$.

认证性是丛等价下的不变量，因为其只断言一些正规串的存在. 如果机密性在所有 C 的等价丛中成立，那么 t 值的机密性就成立. 根据定理 4，t 值的机密性主要依赖于证明 $t\in S_i$.

5　远程证明协议分析

本节的分析在远程证明 RA 串空间中进行. RA 串空间包含定义 5 到定义 9 给出的正规串以及定义 13 给出的攻击者串. 首先给出可信计算远程证明协议安全目标的形式化定义（基于定义 20 和 21）；其次，分析远程证明协议依赖的公钥基础设施的正确性；最后，分别从证明者（本地平台的安全芯片与用户进程）和验证者的角度对远程证明协议进行分析.

5.1　安全属性

本节基于扩展的串空间模型对 3.3 节描述的远程证明协议安全目标进行了形式化定义，如下所示（目标 1～目标 5）：

目标 1（隐私性）. 设 C 是 RA 串空间 Σ 中的一个

丛，且 $s_u \in User[T,U,*,PCA,MF,L,AIK,EK,*,*]$ 是一个正规用户进程串，满足 $SK(PCA) \notin K_P$. 那么：(1) 丛中不存在节点 n 满足 $term(n)=EK \| AIK$；(2) 如果丛中存在节点 m 满足 $term(m)=\{\cdots EK \cdots\}_k$，则 $k=PK(PCA)$.

目标 2(PrivacyCA 认证性). 设 C 是 RA 串空间 Σ 中的一个丛，$s_p \in PrivacyCA[T,PCA,MF,L,AIK,EK]$ 是一个丛高至少为 1 的正规隐私证书权威串，且 $AIK^{-1} \in \{SK_{sig}(T') | T' \in X\}$. 那么必定存在唯一的丛高至少为 2 的正规安全芯片串 $s_t \in TPM[T,*,*,PCA,MF,L,AIK,EK,*,*]$.

目标 3(Verifier 认证性). 设 C 是 RA 串空间 Σ 中的一个丛，且 $s_v \in Verifier[T,U,V,PCA,L,AIK,ML,N_V]$ 是一个丛高至少为 2 的远程验证者串，N_V 唯一源于 $\langle s_v,1\rangle$ 且 $SK(PCA) \notin K_P$，$AIK^{-1} \in \{SK_{sig}(T') | T' \in X\}$. 那么必定存在唯一的丛高至少为 6 的正规安全芯片串 $s_t \in TPM[T,U,V,PCA,*,L,AIK,*,ML,N_V]$ 和一个丛高至少为 10 的正规用户进程串 $s_u \in User[T,U,V,PCA,*,L,AIK,*,ML,N_V]$.

目标 4(ML 的机密性). 设 C 是 RA 串空间 Σ 中的一个丛，且 $s_u \in User[T,U,V,PCA,MF,L,AIK,EK,ML,N_V]$ 是一个正规用户进程串，满足 $SK(V) \notin K_P$. 那么，丛中不存在节点 n 满足 $term(n)=ML$.

目标 5(用户的认证性). 设 C 是 RA 串空间 Σ 中的一个丛，且 $s_u \in User[T,U,V,PCA,MF,L,AIK,EK,ML,N_V]$ 是一个丛高大于 0 的正规用户进程串，满足 $SK(V) \notin K_P$. 那么，丛中必定存在丛高大于 0 的正规远程验证者串 $s_v \in Verifier[T,U,V,PCA,L,AIK,ML,N_V]$.

下面将分别针对这些目标，对可信计算远程证明协议进行分析.

5.2 证书分析

基于 PKI 证书，可以建立协议参与主体与其公钥的关联性. EK 证书可以将 EK 公钥与一个唯一的安全芯片 T 进行关联，在扩展的串空间模型中，可以使用如下定理来描述一个 EK 证书：

定理 5. 设 C 是 Σ 中的一个丛. 对于每个节点 $n \in C$ 和每个 EK 证书 $[a \| pk \| b]_{sk} \sqsubseteq term(n)$，$a \in X, b \in M$，那么或者 $pk=PK_{enc}(a)$，$sk=SK(b)$ 或者 $sk \in K_P$ 成立.

证明. 一个 EK 证书 $[a \| pk \| b]_{sk}$ 可以源于如下两个串：

(1) 一个正规厂商串 $Manufacturer[b,a,pk]$. 正规厂商串表示一个合法的厂商角色，其私钥不会泄露，即 $sk=SK(b) \notin K_P$，同时厂商可以担保 pk 和 a 的关联性，即 $pk=PK_{enc}(a)$.

(2) 一个攻击者 SI 串. SI 串需要攻击者拥有相应的签名私钥，即 $sk \in K_P$. 证毕.

AIK 证书可以保证 AIK 公钥来自一个合法的安全芯片，但无法关联具体的安全芯片. 通过如下定理来描述一个 AIK 证书：

定理 6. 设 C 是 Σ 中的一个丛. 对于每个节点 $n \in C$ 和每个 AIK 证书 $[a \| pk \| b]_{sk} \sqsubseteq term(n)$，$a, b \in M$，那么或者存在一个 $t \in X$ 满足 $pk=PK_{sig}(t)$，$sk=SK(b)$ 或者 $sk \in K_P$ 成立.

证明. 一个 AIK 证书 $[a \| pk \| b]_{sk}$ 可以源于如下两个串：

(1) 一个正规 PrivacyCA 串 $PrivacyCA[*,b,*,a,pk,*]$. Privacy 作为一个可信方，可以担保 pk 来自一个合法的安全芯片，即存在一个 $t \in X$ 满足 $pk=PK_{sig}(t)$.

(2) 一个攻击者 SI 串. SI 串需要攻击者拥有相应的签名私钥，即 $sk \in K_P$. 证毕.

下面从 PrivacyCA 的角度分析其认证性，即目标 2 描述的安全属性.

目标 2 分析(PrivacyCA 认证性). 设节点 $n=\langle s_p,1\rangle$. 由 $AIK^{-1} \in \{SK_{sig}(T') | T' \in X\}$，可知存在安全芯片 T_0 满足 $AIK^{-1}=SK_{sig}(T_0)$，$I=[AIK \| L \| PCA]_{T_0}$. 显然 I 是节点 n 的一个新组件，根据“身份测试”准则，一定存在一个正规节点 $m \leq_C n$ 满足 I 源于 m. 正规节点 m 一定位于定义 5 到定义 9 规定的正规串中，显然 m 是某个正规安全芯片串的第 2 个节点，即存在丛高至少为 2 的串 $s_t \in TPM[T_0,*,*,PCA,MF_0,L,AIK,EK_0,*,*]$. 由于 AIK 由安全芯片唯一生成，即 AIK 唯一源于安全芯片串，故串 s_t 是唯一的.

另外，$cert(EK)=[T \| EK \| MF]_{SK(MF)} \sqsubseteq n$，安全芯片厂商 MF 是可信方，故 $SK(MF) \notin K_P$. 由定理 5 可知，$EK=PK_{enc}(T)$，PrivacyCA 可以确定 EK 是厂商 MF 为安全芯片 T 签发的身份. 但是，我们无法确定 $T=T_0$ 是否成立，这样可能会出现攻击者使用其他芯片的 EK 证书来欺骗 PrivacyCA.

改进方法：使用 $I=[EK \| AIK \| L \| PCA]_T$ 作为安全芯片生成的身份信息，将 EK 和 AIK 进行捆绑认证.

改进方法证明：通过 I 可以确定丛高至少为 2 的串 $s_t \in TPM[T_0,*,*,PCA,MF_0,L,AIK,EK,$

$*,*]$的存在(根据身份测试准则)，即 $EK=EK_0$；而通过 $cert(EK)$，可以进一步确定 $T=T_0$ 和 $MF=MF_0$，这样目标 2 成立.　　完毕.

5.3　从证明者的角度分析

从证明者的角度，其主要关心隐私性、ML 的机密性和用户的认证性. 下面依次分析这 3 个属性.

目标 1 分析(隐私性).　根据定理 4，我们只需要证明 $EK\in S_i$ 或者 $AIK\in S_i$ 在串空间 Σ 中成立，就可以证明丛 C 不存在满足 $term(n)=EK\,\|\,AIK$ 的节点 n. 首先，EK 是公钥，不满足 $EK\in S_0$；其次 $EK\sqsubseteq[T\,\|\,EK\,\|\,MF]_{SK(MF)}$ 发生在节点$\langle s_u,3\rangle$中，没有加密或者哈希的保护形式，不符合 S_i 的递归定义(定义 18). 同理 $AIK\in S_i$ 也不成立. 因此，隐私性并不成立.

改进方法：在节点$\langle s_u,3\rangle$中发送 $cert(EK)$ 时，使用公钥 $PK(PCA)$ 来加密保护 $cert(EK)$.

改进方法证明.　首先丛中不存在节点 n 满足 $term(n)=EK$；其次，EK 只出现在消息 $\{cert(EK)\}_{PK(PCA)}$ 中，由于 $SK(PCA)\in S_0$，根据定义 18，故 $EK\in S_1$. 这样隐私性才成立. 即便 EK 证书本身是公开的，但攻击者无法将其与一个特定的 AIK 关联，也就无法破坏隐私性.　　完毕.

目标 4 分析(ML 的机密性). 根据定义 18 和定理 4，类似隐私性的分析，很容易确定 ML 的机密性不成立. 如果在丛 C 中增加一个攻击者 s 串$\langle -Q\,\|\,ML\,\|\,cert(AIK),+Q,+ML,+cert(AIK)\rangle$，并增加一个攻击者 c 串$\langle -Q,-ML,-cert(AIK),+Q\,\|\,ML\,\|\,cert(AIK)\rangle$，且没有影响到任何正规节点，显然得到丛 C 的等价正常丛(定义 14 和定义 17)，这样 $term(\langle s,3\rangle)=ML$，破坏了 ML 的机密性.

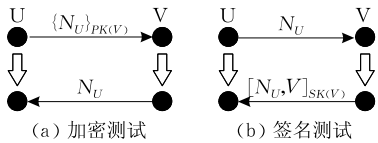
改进方法：在用户进程 U 和远程验证者 V 之间协商一个会话密钥 K 来加密传送 ML，或者直接使用 V 的公钥来加密传送 ML.

改进方法证明.　只要 $K\in S$，或者 $SK(V)\in S$ 成立，根据定义 19 和定理 4，可以确定对于 C 的所有等价丛 C_1，ML 只出现在安全密钥的加密形式中，不存在满足 $term(n)=ML$ 的节点 n. 这样 ML 的机密性成立.　　完毕.

目标 5 分析(用户的认证性).　协议中用户进程串 s_u 接收到的唯一与 V 相关的组件为 N_V，该组件无法使用任何认证测试准则，即无法确认一个正规远程验证者串 s_v 的存在.

改进方法：基于“加密测试”或者“签名测试”准则，可以设计一个满足用户认证性的消息交互流程. 将如图 4 所示的加密测试或者签名测试加入远程证明协议流程中，可以达到 U 认证 V 的目的. 需要假设 N_U 唯一的源于 U 且 $SK(V)\notin K_P$.

改进方法证明.　基于图 4(a)，测试组件为 $\{N_u\}_{PK(V)}$，使用“加密测试”准则，可以确定$(v,1)$的存在；基于图 4(b)，测试组件为 $[N_u,V]_{SK(V)}$，使用“签名测试”准则，也可以确定节点$(v,1)$的存在. 另外基于 N_U 的唯一性，可以确定 V 串的唯一性. 完毕.



(a) 加密测试　　(b) 签名测试

图 4　加密和签名测试流程图

5.4　从验证者的角度分析

验证者主要关心其认证性，下面对其进行分析.

目标3分析(Verifier认证性). 设节点$n=\langle s_v,2\rangle$，根据定义 7，节点 n 存在 3 个新组件：Q，$cert(AIK)$ 和 $h(ML)$. 前两个组件可以使用“签名测试”进行分析，最后一个组件可以使用“哈希测试”进行分析.

由 $AIK^{-1}\in\{SK_{sig}(T')\,|\,T'\in X\}$，即存在某个安全芯片 T_0 满足 $AIK^{-1}=SK_{sig}(T_0)$，由定义 18 知 $AIK^{-1}\in S_0$. 对于新组件 $Q=[h(ML)\,\|\,N_V]_{inv(AIK)}\sqsubseteq term(n)$，根据“签名测试”，一定存在一个正规节点 $m\leqslant_C n$ 满足 Q 源于 m，显然 m 是某个安全芯片串的第 6 个节点，即存在丛高至少为 6 的串 $s_t\in TPM[T_0,U_0,V,PCA_0,*,L_0,AIK,*,ML,N_V]$. 由于 AIK 唯一的源于安全芯片 T_0，故串 s_t 是唯一的.

对于新组件$cert(AIK)=[L\,\|AIK\|PCA]_{SK(PCA)}$，由于 $SK(PCA)\notin K_P$，根据“签名测试”可以知道该组件源于 PrivacyCA 串，同时根据定理 6，存在一个安全芯片 $T_1\in X$ 满足 $AIK=PK_{sig}(T_1)$. 由于 $AIK\in s_t$，可以确定 $T_1=T_0$. 同时根据 $cert(AIK)$ 组件，可以确定 $L=L_0$，$PCA=PCA_0$.

对于新组件 $h(ML)$，由于 ML 的机密性不成立(见目标 4 分析)，“哈希测试”无法保证其来自于一个正规串，攻击者 H 串可以根据 ML 伪造 $h(ML)$.

目前可以确定串 $s_t\in TPM[T_0,U_0,V,PCA,*,L,AIK,*,ML,N_V]$，但是无法确定 $T=T_0$ 和 $U=U_0$ 是否成立，而且也无法确定正规用户进程串 $s_u\in User[T,U,V,PCA,*,L,AIK,*,ML,N_V]$ 的存在. 这样可能导致两类攻击的存在：(1) 一个没有安全芯片的攻击者平台可以利用可信平台的安全芯

片 T 来欺骗验证者 V(攻击丛如图 5 所示);(2) 如果攻击者平台也拥有一个安全芯片 T_1,攻击者根据 ML 度量日志先扩展 T_1 的 PCR 后生成 $h(ML)$,这样可以代替其他任意平台完成远程证明(称为布谷鸟攻击[29],其攻击丛如图 6 所示).

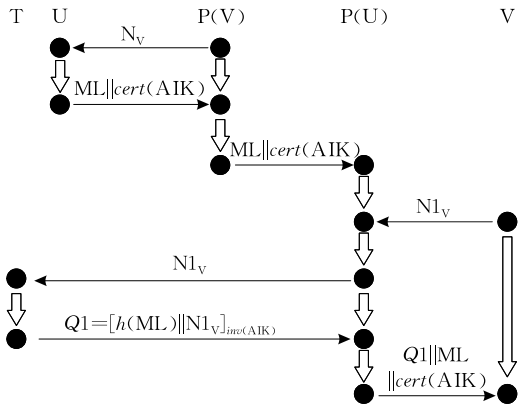


图 5 远程证明协议的一个攻击丛

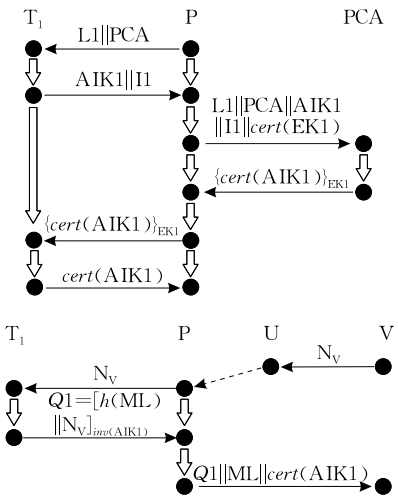


图 6 远程证明协议的一个布谷鸟攻击丛

改进方法:为了预防攻击(1),可以向远程验证者证明 T 和 U 的绑定关系,保证只有 U 才能使用 T,通常可信芯片通过授权会话协议来建立这种关系.而为了避免攻击(2),我们需要向远程方 V 证明具体的安全芯片 T(如使用 EK 证书),但是这样会破坏隐私性,文献[29]中的解决方法是直接在 T 和 V 之间建立安全通道,需要在可信芯片 T 上增加一个特殊安全接口,但从厂商角度而言这种方法比较不切实际.因此,目前还没有有效的方法在保护隐私性的同时防止布谷鸟攻击.

改进方法说明:针对攻击(1)的改进方法依赖可信计算的授权会话协议,其分析需要先证明授权会话协议的安全性,是我们下一步的工作目标.针对攻击(2)的改进方法需要在 T 和 V 之间直接建立安全通道.如果是密码安全通道,可以采用串空间先分析该通道的安全性;如果是硬件安全通道,则需要对串空间进行进一步扩展.如何建立安全通道以及证明安全通道的安全性在本文的分析范围之外. 完毕.

6 攻 击

6.1 攻击描述

本节根据对每个安全属性的分析结果,给出对应攻击的简单描述.这些攻击可以破坏对应的安全属性.在下面消息交互过程中,以角色 P 表示攻击者.构建一个可信平台,平台身份密钥建立过程(消息 1~6)可以执行多次,生成多个 AIK;平台证明过程(消息 7~10)可以基于不同 AIK,与不同的验证者建立可信关系.

攻击 1(破坏隐私性). 攻击者只要截获消息 3 和消息 10 就可以破坏隐私性.对于每个平台的 AIK 建立过程,攻击者截获消息 3,并存储(Cert_EK,AIK)的二元组;当一个平台在消息 10 进行平台证明时,攻击者截获 Cert_AIK,可以查询存储的二元组找到对应的 Cert_EK,进而确定具体的安全芯片 T.而且,基于存储的(Cert_EK,AIK)二元组,攻击者可以将同一个平台的所有 AIK 进行关联,进一步破坏其隐私性.文献[5]中协议只分析平台证明过程,忽略了平台密钥建立过程,因此遗漏了该攻击.

攻击 2(破坏 PrivacyCA 认证性). 攻击者截获消息 3 的 Cert_EK,并随机生成一个 AIK 可以欺骗 PrivacyCA,如对消息 3 进行修改:

消息 2.3.a. U→P(PCA): Cert_EK,AIK,L,PCA,I

消息 2.3.b. P(U)→PCA: Cert_EK,AIK_P,L_P,PCA,I_P

消息 2.4. PCA→U: $\{Cert\_AIK_P\}_{EK}$

该攻击主要是理论上存在风险,实际中就算 PrivacyCA 生成 $Cert\_AIK_P$,攻击者也无法解密,因为其没有 EK 私钥.不过该攻击在实际中可以用于对 PrivacyCA 进行 DoS 攻击,让其不断生成 AIK 证书,消耗其资源.

攻击 3(欺骗攻击,破坏 Verifier 认证性). 一个没有安全芯片 T 的攻击者平台利用一个可信平台的消息流来欺骗验证者 V,从而获得 V 的验证,攻击流如下:

信息 3.7.a. P(V)→U: N_P

消息 3.10.a. U→P(V): ML,Cert_AIK

消息 3.7.b. $V \rightarrow P(U): N_V$

消息 3.8. $P(U) \rightarrow T: N_V$

消息 3.9. $T \rightarrow P(U): Q$

消息 3.10.b. $P(U) \rightarrow V: Q, ML, Cert\_AIK$

该攻击过程中，攻击者首先冒充 V 与一个合法的 U 交互，获取其完整性度量日志 ML 和 AIK 证书 Cert_AIK（消息 3.7.a～3.10.a）；然后，当远程验证者 V 向攻击者平台请求验证数据时，攻击者冒充 U 从安全芯片 T 获得签名数据 Q（消息 3.7.b～3.9）；最后，攻击者组合消息骗取 V 的验证（消息 3.10.b）.

攻击 4（布谷鸟攻击，破坏 Verifier 认证性）. 攻击者购买一个 TPM 芯片，能代替其他任何恶意平台完成远程证明过程，假设攻击者的安全芯片为 T_P，攻击流程如下：

消息 4.1. $P \rightarrow T_P: L_P, PCA$

消息 4.2. $T_P \rightarrow P: AIK_P, I_P$

消息 4.3. $P \rightarrow PCA: Cert\_EK_P, AIK_P, L_P, PCA, I_P$

消息 4.4. $PCA \rightarrow P: \{Cert\_AIK_P\}_{EK}$

消息 4.5. $P \rightarrow T_P: \{Cert\_AIK_P\}_{EK}$

消息 4.6. $T_P \rightarrow P: Cert\_AIK_P$

消息 4.7.a. $P(V) \rightarrow U: N_P$

消息 4.10.a. $U \rightarrow P(V): ML$

消息 4.7.b. $V \rightarrow P(U_P): N_V$

消息 4.8. $P(U_P) \rightarrow T_P: N_V$

消息 4.9. $T_P \rightarrow P(U_P): Q_P$

消息 4.10.b. $P(U_P) \rightarrow V: Q_P, ML, Cert\_AIK_P$

该攻击过程中，攻击者首先使用安全芯片 T_P 向 PrivacyCA 申请一个合法的身份 $Cert\_AIK_P$（消息 4.1～4.6）；然后，攻击者冒充 V 向一个合法的用户平台 U 发起挑战，获取其合法度量日志 ML（消息 4.7.a～4.10.a）；基于 ML，攻击者可以在 T_P 平台上通过 TPM_Extend 命令扩展出预期的 PCR 值；最后，当攻击者控制的任意恶意平台 U_P 想要获取 V 的验证时，可以使用 T_P 所在的攻击者平台代替 U_P 完成平台证明过程（消息 4.7.b～4.10.b）. 因为 $Cert\_AIK_P$ 是合法的身份，Q_P 是对合法完整性状态 PCR 值的签名，因此可以通过 V 的验证. 该攻击体现了 TPM 安全芯片本身也是一把双刃剑，能被攻击者利用. 文献[5]中由于没有区分安全芯片 T 和用户进程 U 两个角色，因此没有发现此攻击. 由于该攻击与 CMU 学者[29]发现的布谷鸟攻击一致，因此采用这种攻击称谓.

攻击 5（破坏 ML 的机密性）. 由于远程证明协议设计时只考虑了认证性，使用了基于 AIK 的签名，而没有考虑机密性，并且 ML 是明文传送的，任何攻击者都可以获得 ML，则 ML 的机密性缺失可能被攻击用来破坏其他安全属性，如攻击 4 中攻击者使用获取的 ML 伪造合法的完整性状态，破坏了 Verifier 的认证性.

攻击 6（破坏用户的认证性）. 远程证明协议本身是一个单向认证协议，即只有验证者 V 来检查证明者平台的身份和完整性状态，而证明者平台并没有获得远程验证者 V 的任何认证信息. 因此，任意攻击者都可以随意地向证明者平台发起认证挑战，平台会不断调用 TPM_Quote 命令进行应答. 一方面消耗 TPM 的计算资源，进行 DoS 攻击；另一方面，截取可信平台的数据，进行其他攻击（如攻击 3 和攻击 4 中都有利用这一点，攻击者 P 冒充 V 与 U 交互）.

6.2　攻击实验

本节对以上攻击中的布谷鸟攻击（攻击 4）进行了模拟实现. 在文献[30]中，我们提出的可信平台模块 TEEM，通过修改安全芯片的驱动，使得计算平台的可信功能通过 USB 发送给 TEEM 完成. TEEM（相当于 T_P）可以作为代替平台上原有 TPM/TCM 使用的布谷鸟攻击实例. 攻击者获得 TEEM 后，将 TEEM 的 PCR 和 ML 固定为 V 信任的值，通过 USB 连接任何计算平台，即可代替该平台完成证明，并获得 V 的信任. 具体攻击实现如图 7 所示，包含如下两个过程：

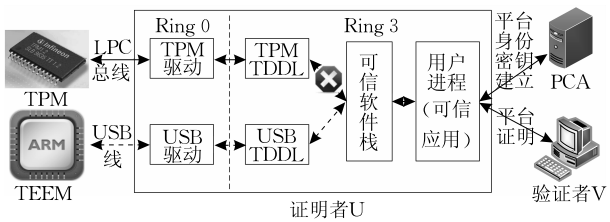


图 7　布谷鸟攻击实现

(1) 模拟一个安全芯片

基于 ARM 开发板，我们模拟了一个安全芯片，称为 TEEM. TEEM[30]使用 ARM TrustZone 安全扩展技术提供隔离执行环境，使用开源的软件 TPM Emulator① 提供可信功能（包括密钥管理，身份生成，引证，签名等）. TPM 安全芯片一般通过 LPC 总线固定在主板上，而 TEEM 通过 USB 线与计算机

① http://tpm-emulator.berlios.de

(如证明者 U 所在平台)连接,TEEM 位于 ARM 嵌入式环境,通过 Linux gadget 驱动将 ARM 开发板模拟为一个 USB 外设,证明者平台的 USB 驱动可以识别该外设.具体实现时,ARM 端内核配置 gadget serial 驱动,计算机主机安装 CDC/ACM① 驱动可识别 TEEM 所在的 ARM 开发板.通过 USB 线连接 TEEM 所在的 ARM 开发板和证明者主机平台,主机上所有通过该 USB 驱动请求的数据都会转发给 TEEM 进行处理,因此,TEEM 所在的 ARM 开发板可以作为一个模拟的安全芯片使用.

(2)使用模拟安全芯片实施布谷鸟攻击

根据可信计算标准规范[1],可信应用通过可信软件栈调用可信计算的功能;而可信软件栈通过可信设备驱动库 TPM TDDL(Trusted Device Driver Library)将可信计算功能的命令请求交给 TPM 驱动,进而交给 TPM 安全芯片处理(如图 7 所示).类似地,我们实现了 USB TDDL 设备驱动库,该库通过 USB 驱动直接与 TEEM 通信.USB TDDL 主要包含 3 个基本接口:TDDL_Open()、TDDL_Close()、TDDL_TransmitData().这 3 个接口通过访问 USB 驱动完成.

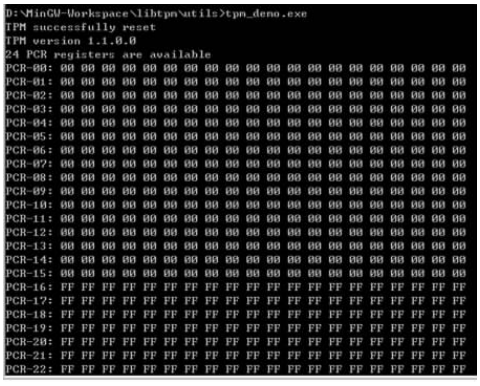


图 8 通过可信软件栈 libtpm 访问 TEEM 的响应结果

在主机平台上通过使用 USB TDDL 库替换 TPM TDDL 库,并通过 USB 线连接 TEEM.这样可信软件栈的所有可信功能请求将会交给模拟安全芯片 TEEM 处理,而不是实际的 TPM 安全芯片.图 8 显示了在 Windows 平台上替换 TDDL 库后通过可信软件栈 libtpm② 访问 TPM 的功能时的响应结果,不过其显示出来的 TPM 信息实际上是来自 ARM 开发板的 TEEM 模拟芯片.

通过篡改证明者平台 TDDL 库和设备驱动,当用户进程与 PCA 建立平台身份密钥,以及和远程验证者 V 进行平台证明时,实际都是使用 TEEM 模拟的可信功能.可信计算命令在 TEEM 中的执行性能(包含主机与 TEEM 之间 USB 通信的时间)在文献[30]中也进行了详细的评估.通过在 Windows 平台上的实验,对于基于 TEEM 的远程证明协议,身份生成命令 TPM_MakeIdentity 需要 3.593 s,AIK 证书激活命令 TPM_ActivateIdentity 需要 421 ms,引证命令 TPM_Quote 需要 359 ms,一个 PCR 扩展需要 62.5 ms,读取 PCR 值也需要 62.5 ms.

显然,该攻击需要 TEEM 拥有一个合法的 EK 证书.只要攻击者控制一个安全芯片,其就拥有一个有效的 EK 证书.另外,随着可信芯片的发展,支持 USB 接口的便携式可信芯片是一个趋势.这样,只要攻击者拥有一个合法的便携式可信芯片,其可以在任何主机上完成远程证明协议来进行布谷鸟攻击.因此,在发展可信安全芯片时,需要提出相应的机制来防止攻击的发生,本文基于串空间的分析方法可以提供参考.

7 结束语

本文对串空间模型进行了扩展,使得其能分析可信计算远程证明协议,并证明了扩展后范式引理的正确性.通过扩展的模型对远程证明协议及其安全属性进行了形式化定义.同时,本文提出了 4 个认证测试准则,并使用这些准则对远程证明协议进行了分析.经过分析发现,期望的安全属性只有部分成立,不成立的部分容易导致攻击(如布谷鸟攻击等).针对攻击,本文提出了相应的改进方法.同时为了验证串空间的分析结果,本文基于 ARM 开发板对发现的布谷鸟攻击进行了模拟实现.

本文的工作弥补了可信计算理论研究的不足,为其在实际中的使用提供信心保障.本文的串空间扩展方法针对可信安全芯片进行,同样也适合分析拥有类似功能的密码硬件.下一步工作主要包含 4 个方面:基于本文提出的改进方法,设计并分析新的远程证明协议,促进可信计算的发展;进一步分析可信计算的授权会话协议(如 TPM2.0 中新的授权协议等),明确安全芯片和计算平台的绑定认证性;扩展开源 CPSA[8] 工具实现对本文方法的自动化验证.

参 考 文 献

[1] Trusted Computing Group. TPM Main Specification Level 2

① http://www.thesycon.de/eng/usb_cdcacm.shtml
② http://ibmswtpm.sourceforge.net/#libtpm

Version 1.2, Revision 116, 2011

[2] Feng Wei, Qin Yu, Yu Ai-Min, Feng Deng-Guo. A DRTM-based method for trusted network connection//Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications. Changsha, China, 2011: 425-435

[3] Santos N, Rodrigues R, Gummadi K P, Saroiu S. Policy-sealed data: A new abstracton for building trusted cloud services//Proceedings of the 21st USENIX Security Symposium. Bellevue, WA, 2012: 175-188

[4] Coker G, Guttman J, et al. Attestation: Evidence and trust//Proceedings of the 10th International Conference on Information and Communications Security (ICICS08). Birmingham, UK, 2008: 1-18

[5] Coker G, Guttman J, et al. Principles of remote attestation. International Journal of Information Security, 2011, 10(2): 63-81

[6] Ramsdell J D, Guttman J D, et al. An Analysis of the CAVES Attestation Protocol using CPSA. Bedford: The MITRE Corporation, 2009

[7] Doghmi S F, Guttman J D, Thayer F J. Searching for shapes in cryptographic protocols//Proceedings of the 13th TACAS. LNCS 4424. Berlin: Springer, 2007: 523-537

[8] Ramsdell J D, Guttman J D. CPSA Primer. Bedford: The MITRE Corporation, 2012

[9] Feng Deng-Guo, Qin Yu, Wang Dan, Chu Xiao-Bo. Research on trusted computing technology. Journal of Computer Research and Development, 2011, 48(8): 1332-1349(in Chinese)
(冯登国，秦宇，王丹，初晓博. 可信计算技术研究. 计算机研究与发展，2011，48(8)：1332-1349)

[10] Pirker M, Toegl R, Hein D, Danner P. A PrivacyCA for anonymity and trust//Proceedings of the Trust and Trustworthy Computing(Trust 2009). Oxford, UK, 2009: 101-119

[11] Ryan M. Introduction to the TPM 1.2. Draft of October 31, 2008

[12] Bai Guang-Dong, Martin Andrew, et al. TrustFound: Towards a formal foundation for model checking trusted computing platforms//Proceedings of the 19th International Symposium on Formal Methods. Berlin: Springer-Verlag, 2014(8442): 110-126

[13] Chen Li-Qun, Li Jiang-Tao. Flexible and scalable digital signatures in TPM 2.0//Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS 2013). New York, USA, 2013: 37-48

[14] Kremer S, Kunnemann R. Automated analysis of security protocols with global state//Proceedings of the 25th IEEE Symposium on Security and Privacy (IEEE S&P). Oakland, CA, 2014

[15] Schmidt B, Sasse R, et al. Automated verification of group key agreement protocols//Proceedings of the 25th IEEE Symposium on Security and Privacy (IEEE S&P). Oakland, CA, 2014

[16] Dolev D, Yao A C. On the security of public key protocols. IEEE Transactions on Information Theory, 1983, IT-29(2): 198-208

[17] Fabrega F J T, Herzog J C, Guttman J D. Strand spaces: Why is a security protocol correct?//Proceedings of the IEEE S&P. Oakland, CA, 1998: 160-171

[18] Guttman J D, Thayer F J. Authentication tests//Proceedings of the IEEE S&P. Los Alamitos, CA, 2000: 96-109

[19] Guttman J D, Thayer F J. Authentication tests and the structure of bundles. Theoretical Computer Science, 2002, 283(2): 333-380

[20] Doghmi S F, Guttman J D, Thayer F J. Completeness of the authentication tests//Proceedings of the 12th European Symposium on Reaearch in Computer Security (ESORICS). Dresden, Germany, 2007: 106-121

[21] Kamil A, Lowe G. Analysing TLS in the strand spaces model. Journal of Computer Security, 2011, 19(5): 975-1025

[22] Kamil A, Lowe G. Specifying and modelling secure channels in strand spaces//Degano P, Guttman J eds. Formal Aspects in Security and Trust (FAST 2009). LNCS 5983. Berlin: Springer, 2009: 233-247

[23] Kamil A, Lowe G. Understanding abstractions of secure channels//Degano P, Etalle S, Guttman J eds. Formal Aspects of Security and Trust (FAST 2010). LNCS 6561. Berlin: Springer, 2010: 50-64

[24] Jiang Rui, Hu Ai-Qun, Li Jian-Hua. Research on formal design of ESIKE based on Authentication Tests. Chinese Journal of Computers, 2006, 29(9): 1692-1699(in Chinese)
(蒋睿，胡爱群，李建华. 基于 Authentication Test 方法的高效安全 IKE 形式化设计研究. 计算机学报，2006，29(9)：1692-1699)

[25] Guttman J D. State and progress in strand spaces: Proving fair exchange. Journal of Automated Reasoning, 2012, 48(2): 159-195

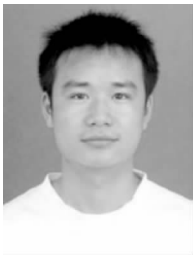
[26] Ramsdell J D. Proving Security Goals with Shape Analysis Sentences. Bedford: The MITRE Corporation, 2013

[27] Ramsdell J D, Dougherty D J, et al. A Hybrid Analysis for Security Protocols with State. Bedford: The MITRE Corporation, 2014

[28] Guttman J D. Establishing and preserving protocol security goals. Journal of Computer Security, 2014, 22(2): 203-267

[29] Parno B. Bootstrapping trust in a trusted platform//Proceedings of the 3rd Conference on Hot Topics in Security(HotSec). San Jose, CA, 2008: 1-6

[30] Feng Wei, Feng Deng-Guo, et al. TEEM: A user-oriented trusted mobile device for multi-platform security applications //Proceedings of the Trust and Trustworthy Computing (Trust2013). London, UK, 2013: 133-141



FENG Wei, born in 1986, Ph. D. candidate. His research interests include trusted computing, network and system security.

FENG Deng-Guo, born in 1965, Ph. D., professor, Ph. D. supervisor. His research interests include information security and cryptology, trusted computing and information assurance.

Background

Trusted Computing is an emerging technology to enhance the security of various computing platforms by a dedicated secure chip (TPM/TCM), which is accepted by industrial and academic world. The key technologies of trusted computing include trust chain, remote attestation, trusted network connection, OS measurements and so on. And these technologies can be applied to many practical secure scenarios, like online payment, digital rights management, secure storage with encryption and network access control. However, the theory of trusted computing lacks the formal security analysis, and thus the security of its applications cannot be proved. This paper tries to analyze the security of trusted computing using formal methods (like strand space model), and prove or improve its security protocols and secure mechanisms. Specifically, this paper extends the strand space model to analyze the remote attestation protocol of trusted omputing. By analyzing, we find some attacks to remote attestation and also suggest possible improvement methods. This paper also focuses on the practical attack experiments (like cuckoo attack) to secure chip TPM/TCM.

This work is partly supported by the National Natural Science Foundation of China (under Grant Nos. 91118006 and 61202414) and the National Basic Research Program (973 Program) of China (under Grants No. 2013CB338003). These projects aim to provide formal analysis methods and test tools for trusted computing protocols and systems. The research group has been working on the trusted computing, network and system security. Many papers have been published in respectable international conferences, such as TRUST, ISC, ICICS, NSS, Trustbus, TrustCom, etc.