doi:10.3969/j.issn.1000-436x.2014.08.013

基于可信芯片的平台身份证明方案研究

张倩颖[1,2]，冯登国[1]，赵世军[1,2]
(1. 中国科学院 软件研究所，北京 100190；2. 中国科学院大学，北京 100049)

摘 要：对基于可信第三方的平台身份证明方案进行了研究，提出了一种用证书和令牌标识可信计算平台并直接使用令牌证明平台身份的方案。与其他方案相比，该方案降低了证明过程的计算量和通信量，并且验证方验证平台身份的同时能够确认平台状态可信，获得了更高的安全性。利用协议组合逻辑证明了方案满足平台身份验证正确性和匿名性。原型系统实验结果表明，该方案平台身份证明效率高，特别适用于无线网络环境。

关键词：可信计算；平台身份证明；协议组合逻辑；匿名性
中图分类号：TP309 **文献标识码**：A **文章编号**：1000-436X(2014)08-0095-12

Research of platform identity attestation based on trusted chip

ZHANG Qian-ying[1,2], FENG Deng-guo[1], ZHAO Shi-jun[1,2]
(1. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: By studying the platform identity attestation base on trusted third parties, a scheme where a trusted computing platform is identified by a certificate and a token is proposed. In this scheme, only the token is used when the platform proves its identity. Compared to other schemes, this scheme not only has much lower calculation and communication, but also convinces the verifier of the trustworthiness of the client's platform state during the platform identity attestation. A detailed security proof of the proposed scheme is presented by using the protocol composition logic, and the proof shows that the scheme satisfies correctness and anonymity of platform identity verification. The experiment result in a developed prototype system shows that the proposed scheme provides good performances in computation and communication, and is especially suitable for the wireless network.

Key words: trusted computing; platform identity attestation; protocol composition logic; anonymity

1 引言

可信计算技术的基本思想是以可信平台模块（TPM，trusted platform module）为信任根建立终端平台的信任[1]，并通过远程证明将信任扩展至网络。平台身份证明是远程证明的一个重要方面，用于向远程验证方证明可信计算平台的身份，是建立平台间信任的基础。平台身份证明实质上是证明TPM 的身份，每个 TPM 在出厂时都绑定了一个背书密钥（EK，endorsement key），不同 TPM 拥有不同的 EK，所以 EK 可以唯一地标识 TPM。但是，如果所有的平台身份证明都基于 EK 进行，就无法保障平台的隐私性，因此需要为平台提供一种匿名身份证明机制，在不暴露平台身份的同时向验证方证明平台具有真实的 TPM。

TPM 标准采用 PCA（privacy CA）方案和 DAA（direct anonymous attestation）方案[2]来解决平台身份证明的隐私保护问题。在 PCA 方案中，TPM 首先生成一对身份证明密钥（AIK，attestation identity key）作为 EK 的别名，然后在证明自己拥有真实

收稿日期：2013-04-19；**修回日期**：2014-03-04
基金项目：国家自然科学基金资助项目(91118006, 61202414)；国家重点基础研究发展计划(“973”计划)基金资助项目(2013CB338003)
Foundation Items: The National Natural Science Foundation of China (91118006, 61202414); The National Basic Research Program of China (973 Program) (2013CB338003)

EK 的基础上，向作为可信第三方的 Privacy CA 申请 AIK 公钥证书，之后可以该证书为身份进行证明。但是，为防止身份被关联，每次证明 TPM 都要生成新的 AIK 向 Privacy CA 申请证书，导致 Privacy CA 负载过大，并且在证明过程中，平台向验证方发送 AIK 证书会造成较大的通信负担。DAA 方案是在群签名方案的基础上发展起来的，在 DAA 方案中 TPM 可以直接向验证方证明可信计算平台的真实性，无需可信第三方的参与，并且只申请一次 DAA 证书就可以进行多次身份证明并保证匿名性。但是，该方案非常复杂，只适用于具有较强计算能力的设备，并且很难进行跨域证明。

针对上述方案存在的问题，本文提出名为 PIA（platform identity attestation）的平台身份证明方案，其特点是以证书和令牌作为可信计算平台的身份，平台直接使用令牌即可基于可信第三方进行身份证明。对比其他方案，PIA 方案具有更高的证明效率和安全性。本文用协议组合逻辑（PCL，protocol composition logic）分析了方案的平台身份颁发—验证协议，证明协议满足平台身份验证正确性和匿名性 2 个安全属性，并设计实现了原型系统，实验结果表明，客户端能通过无线网络高效完成身份证明。

2 相关工作

TPM v1.2 标准[3]发布后，国内外研究人员围绕 PCA 和 DAA 两类平台身份证明方案开展了许多研究工作。

为防止 Privacy CA 为不具有 TPM 的平台颁发证书，文献[4]要求 TPM 在申请证书时用 EK 对请求信息进行签名，以此证明 TPM 的真实性，但这一要求违背了 EK 只进行加密操作的原则。Pirker 等根据建立可信服务的准则实现了一个可信的 Privacy CA 服务[5]，该服务是一个在 Xen hypervisor 上独立执行的自包含映像，能向客户端证明其状态，从而向客户端保证其能提供的隐私策略。Chen 等在文献[6]中分析了 PCA 方案的不可伪造性，即如果 TPM 不能向 CA 证明其身份那该 TPM 就不能获得证书，但只有所有 TPM 都未被攻破的情况下该分析才能成立，为抵抗来自被攻破 TPM 的攻击，作者提出一个改进的协议，要求 TPM 用 AIK 私钥对 EK 公钥签名，从而向 Privacy CA 证明 AIK 和 EK 属于同一诚实的 TPM，之后作者在文献[7]中分析了该协议的安全性。杨力等提出一种在无线网络环境中验证终端平台可信性的匿名认证协议[8]，该协议借助 Privacy CA 为移动节点签发 AIK 证书，节点首次接入本地网络时 Privacy CA 要验证其 AIK 证书，造成转发的消息量较多，因此作者之后又基于 DAA 方案对其进行了改进[9]。崔巍等将 PCA 方案应用于云计算环境[10]，提出一个可信的 IaaS 框架，使云服务提供商能够向用户远程证明云服务的安全性。Winkler 等提出一种保护用户隐私的视频监控方案[11, 12]，利用 Privacy CA 颁发的身份，集成 TPM 的摄像机可以向用户证明其状态信息。Pirker 等提出一个可信云节点加入协议[13]，该协议将 Privacy CA 功能集成到云控制端，使其能验证云节点具有真实的 TPM，并以此为基础进一步验证云节点的可信状态。Fongen 等提出将 Privacy CA 与身份管理服务相结合的方案[14]，该方案将用户身份与平台的 AIK 证书绑定，使认证双方能获得更强的软件完整性保证。Kraxberger 等提出一个适用于覆盖网络的可信认证方案[15]，该方案基于 Privacy CA 为覆盖网络的节点创建不可否认的、可验证的身份，并修改了 Privacy CA 的配置限制其为每个 TPM 只颁发一个 AIK 证书，从而保证每个节点都有唯一的身份。

针对 DAA 方案只适用于单信任域的问题，陈小峰等提出跨域 DAA 方案[16]，该方案有效解决了跨信任域的直接匿名证明问题，为不同厂商的可信计算平台间的匿名通信建立了基础，但方案以 TPM 标准中的 DAA 方案[2]为基础，而标准中的方案利用 RSA 密码体制实现，存在 DAA 签名长度较长、计算量大的缺点。2008 年，Brickell 等基于 CL-LRSW 假设提出首个基于椭圆曲线和双线性映射的 DAA 方案[17]，有效缩短了 DAA 证书和签名长度，大幅提高了计算和通信性能，之后 Chen 等在文献[18]中又对该方案做了进一步改进。此外，2008 年，陈小峰[19]和 Brickell[20]分别提出另一种基于 q-SDH 假设构造的基于椭圆曲线和双线性映射的 DAA 方案，极大地缩短了 DAA 证书的长度，进一步推动了 DAA 协议的改进研究，后续很多研究[21~23]都以该方案为基础。2012 年，杨力等提出一种移动环境下跨可信域的 DAA 证明方案[24]，实现了对移动终端在多可信域之间漫游时的可信计算平台认证。

尽管已有许多方案对 DAA 协议进行了改进，但是由于 DAA 方案在证明过程中需要进行零知识

证明，其效率和性能仍比较低，对实现而言复杂度仍较大，并且 DAA 方案只适用于单信任域的情况，跨信任域的证明问题也需要通过零知识证明解决，进一步增加了协议的复杂性，上述原因导致 DAA 方案很难应用于实际系统。因此，在实际应用中通常采用 PCA 方案作为 DAA 的轻量级替代方案。

PCA 方案如图 1 所示，Privacy CA 在验证客户端 TPM 具有真实 EK 的基础上，为其颁发 AIK 证书。获得证书后，TPM 通过使用 AIK 私钥签名证明数据，并将签名和 AIK 证书发送给验证方可以进行身份证明。验证方通过验证证书和签名可以确认可信计算平台的身份。分析 PCA 方案的相关研究成果可以发现，PCA 方案仍存在以下问题：每次证明 TPM 都需要申请新的 AIK 证书，证书颁发负担大；平台身份证明时 TPM 需要向验证方发送 AIK 证书，增加了通信负载。

为解决上述问题，本文提出 PIA 方案，用身份证书和身份令牌标识一个可信计算平台，平台申请一次证书即可以利用该证书多次申请身份令牌，降低了证书颁发负担。此外，平台直接使用令牌就可以进行身份证明，无需向验证方发送 AIK 证书，降低了证明过程的通信负载，并且免去了验证方验证 AIK 证书的计算负担。

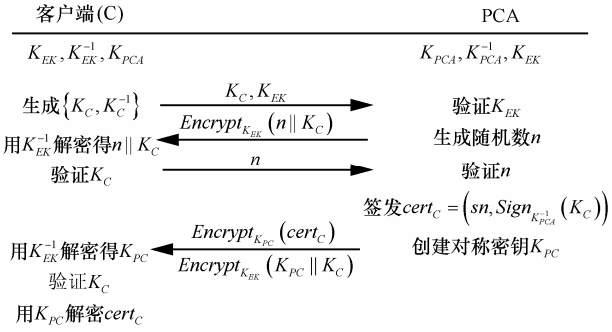

图 1　PCA 方案

3　平台身份证明方案

3.1　方案架构

方案架构如图 2 所示，包含以下 3 个参与方。

1) 客户端（C）。客户端是嵌入了 TPM 芯片的可信计算平台，向身份权威申请身份证书和令牌（token）后，在访问验证方服务时，用令牌向验证方证明身份。

2) 身份权威（IA，identity authority）。身份权威作为可信第三方负责颁发、管理和验证域内客户端平台身份。IA 颁发身份证书时会验证客户端 EK 证书，而颁发 token 时则会验证客户端平台配置和身份证书，并将颁发的 token 与对应的身份证书绑定。IA 收到验证方的身份验证请求后，基于 token 验证客户端平台身份并将结果返回给验证方。

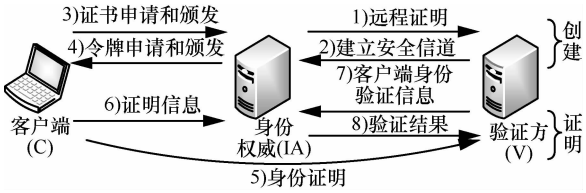

图 2　平台身份证明架构

3) 验证方（V）。验证方是 IA 管理域外的服务提供方，通过安全信道与 IA 交互验证客户端平台身份，得到验证结果后做出访问决策。

方案包含以下 2 个阶段。

1) 创建阶段。验证方通过远程证明验证 IA 可信后与其建立安全信道，委托 IA 认证和管理客户端。客户端利用 TPM 向 IA 申请身份证书，然后利用此证书向 IA 申请有使用期限的 token。token 过期或平台配置变化时，客户端可以利用身份证书重新申请 token：客户端向 IA 证明当前平台配置并出示身份证书信息，IA 验证证明信息和证书后为客户端颁发一个与证书绑定的新 token。

2) 证明阶段。首先客户端用 token 向验证方进行平台身份证明，然后验证方委托 IA 验证证明信息，收到验证结果后做出访问决策。

3.2　协议细节

PIA 方案包含 2 个协议：身份权威证明协议和平台身份颁发—验证协议，后者是 PIA 方案的核心，包含 3 个子协议：证书颁发协议、令牌颁发协议和身份验证协议。下面对各协议的交互细节进行说明。

身份权威证明。V 委托 IA 管理和验证客户端平台身份的前提是 V 确信 IA 的安全性，因此 V 定期要求 IA 进行基于硬件的安全性证明（如图 3 所示）。V 发送证明请求和随机数 *n* 给 IA。IA 加载可信代码度量平台配置并将结果扩展至 PCR，然后产生公私钥对，将私钥用当前平台配置封装，将公钥扩展至 PCR，之后向 V 远程证明平台配置。V 验证证明信息后，将 IA 加入信任列表并与之建立安全信道。

证书颁发（certificate issue）。为获得身份证书，C 向 IA 进行基于可信芯片的证明（如图 4 所示）。

图 3 身份权威证明协议

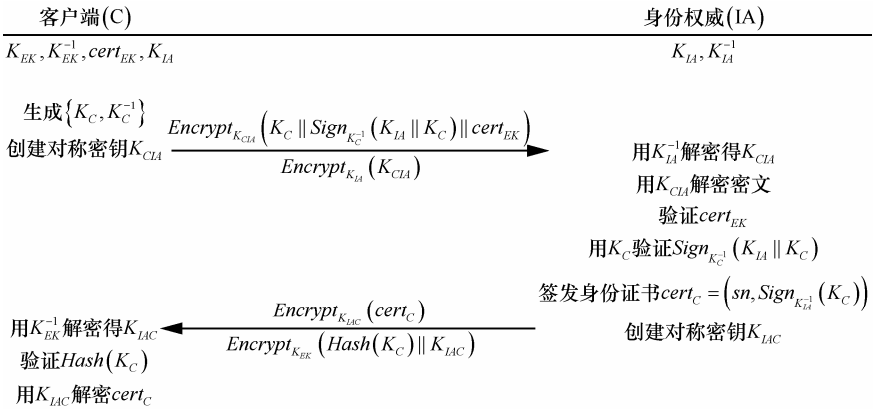

图 4 证书颁发协议

C 生成 AIK 密钥对 $\{K_C, K_C^{-1}\}$，用 K_C^{-1} 对 IA 公钥和 K_C 签名，将 K_C、签名和 EK 证书加密发送给 IA。IA 解密消息验证 EK 证书和签名，然后用私钥对 K_C 签名生成身份证书，将 K_C 的摘要值和身份证书加密发送给 C。C 验证摘要值正确后解密得到身份证书。

令牌颁发（token issue）。为保护身份证书隐私性，C 向 IA 申请用于身份证明的 token（如图 5 所示）。C 将身份证书序列号、平台配置证明信息发送给 IA。IA 验证序列号对应证书有效性和证明信息，为 C 创建 token，用 C 的 EK 公钥加密 token 和远程证明的随机数发送给 C。C 解密后，验证随机数，将 token 用当前平台配置封装存储，保证平台配置变化后 token 不能访问。

身份验证（identity verification）。V 收到访问请求后要求 C 证明平台身份，与 IA 协作验证客户端身份证明（如图 6 所示）。C 用令牌密钥 K_{CT} 计算散列值 d，将 d 和 IA 标识 ID_{IA} 发送给 V，并将 (ID_C, n_C, d) 发送给 IA。V 根据 ID_{IA} 查找对应的通信密钥，将 (n_V, d) 发送给 IA。IA 查找来自 C 和 V 的 d 相等的消息得到 (ID_C, n_C, n_V, d)，查找 ID_C 对应的 token，验证 token 未过期后用 K_{CT} 验证 d 计算是否正确，将结果发送给 V。

C 不直接使用身份证书而采用与证书绑定的 token 向 V 进行身份证明，且证明过程中 V 仅收到散列值，每次证明散列值不同，这种证书和令牌相结合的方式能防止 V 追踪客户端行为，保护客户端平台隐私。如果 V 确定 C 有不良行为，可以将散列值提交给 IA，要求其停止认证该值关联的平台，IA 将该平台的 token 从令牌列表中删除就实现了客户端平台身份撤销。

4 协议分析

4.1 安全性分析

本文用 PCL 分析 PIA 方案中平台身份颁发—验证协议的安全性。在简介 PCL 的基础上，首先分析 3 个子协议的不变量和安全属性，然后利用阶段组合证明方法证明整个协议的安全性。由于篇幅原因，本文省略了部分安全属性的证明过程。

4.1.1 协议组合逻辑

PCL 主要用于各种网络协议安全属性的公理化证明，其理论核心是协议可组合的安全性证明，即将复杂的协议看作是由简单协议经过一系列演绎操作得到的，而演绎操作的前提是简单协议的相加性组合与非破坏性组合，这样复杂协议的证明可以通过对简单协议证明的组合得到。

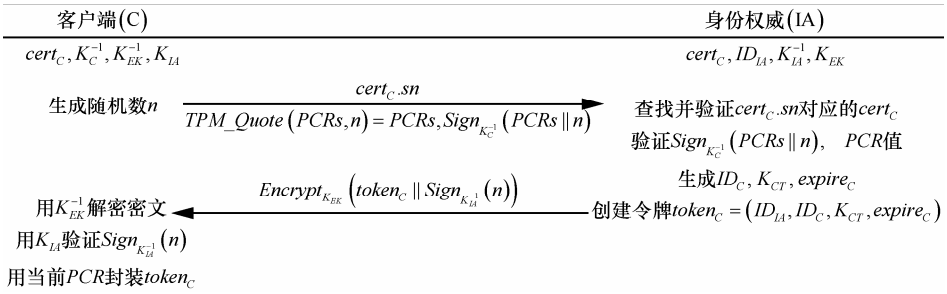

图 5　令牌颁发协议

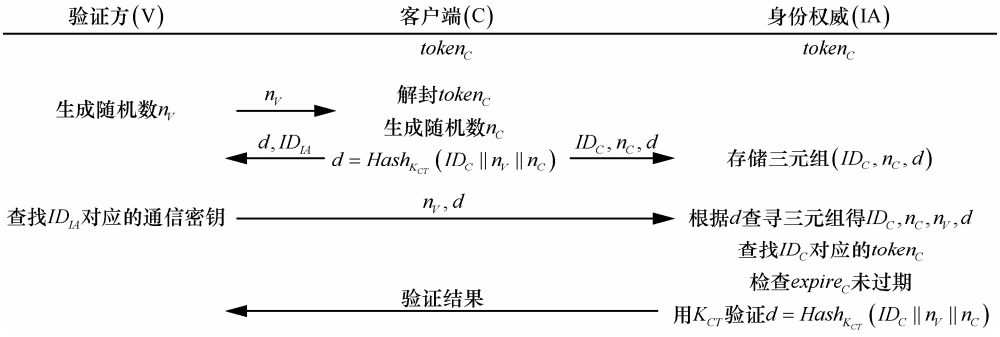

图 6　身份验证协议

1) 协议建模

PCL 将协议定义为有限角色集合，并利用 cord 演算[25]将角色描述为由输入、输出参数和诚实参与方执行的动作序列组成的程序。

2) 语法和语义

PCL 在证明过程中通常使用模态公式 $\theta[P]_X\phi$ 描述程序执行前后状态的变化，其含义是线程 X 执行动作序列 P，如果初始状态中 θ 成立，那么结果状态中 ϕ 一定成立，详细语法和语义描述参见文献[26]。

3) 证明系统

PCL 证明系统[26]在一阶逻辑基础上，扩展了有关协议动作、时序推理、知识等的公理和证明规则，以及一种称为诚实规则的特殊形式的不变性规则。诚实规则用于将某个角色的事实与其他角色的推导动作结合起来。PCL 证明简单协议安全性包含 2 个步骤，首先使用诚实规则证明协议不变量，然后用不变量作为假设证明协议属性。

4) 组合证明方法

PCL 能在分析组件安全性的基础上，将组件以不同方式组合，若组件原有安全性未被破坏，即可证明组合协议安全性。下面以 2 个子协议组合为例介绍本文使用的阶段组合证明方法[27]。①分析子协议 Q_1，Q_2 的不变量 Γ_1，Γ_2 和安全属性 $\theta_1[P_1]\varphi_1$，$\theta_2[P_2]\varphi_2$。②验证子协议满足彼此不变量：$Q_1 \vdash \Gamma_2$，$Q_2 \vdash \Gamma_1$。③验证 Q_1 安全属性的后置条件蕴含 Q_2 安全属性的前提条件：$\varphi_1 \supset \theta_2$。④验证 Q_1，Q_2 的任意协议步骤 B 保持 Q_1 安全属性的前提条件：$\theta_1[B]\theta_1$。上述验证通过，则对于 Q_1，Q_2 的阶段组合协议 Q 有 $Q \vdash \theta_1[P_1P_2]\varphi_2$。

4.1.2　证书颁发协议分析

1) 协议建模

证书颁发协议有客户端和身份权威 2 个参与实体，分别用 $\hat{C}$、$\hat{I}$ 表示，相应的线程分别用 C、I 表示。协议中客户端和身份权威角色程序的形式化描述如图 7 所示，程序语法参见文献[26]。

2) 不变量和安全属性

不变量 Γ_{CI} 描述客户端如果能解密一个加密的消息，那么该实体一定持有对应的解密密钥

$$\begin{aligned}\Gamma_{CI} \equiv\ & Honest(\hat{C}) \wedge Decrypt(C, ENC_K(t)) \\ & \supset Has(\hat{C}, K^{-1})\end{aligned} \tag{1}$$

证书颁发协议的安全目标形式化为证书颁发正确性 $\phi_{CI,correct}$，下面给出协议对身份权威的证书颁发正确性保证，对客户端的保证与之类似。

定理 1　证书颁发正确性。执行身份权威角色，证书颁发协议保证平台身份证书颁发正确性，即 $CI \vdash [IdentityAuthority_{CI}]_I \phi_{CI,correct}$

$$\phi_{CI,correct} \equiv Honest(\hat{C}) \wedge Has(\hat{C}, cert_C)$$

$$\supset Has(\hat{C},K_{EK}^{-1}) \wedge Send(I,msg) \wedge Contains(msg,cert_C) \quad (2)$$

定理 1 表明执行证书颁发协议后，如果客户端获得身份证书，那么该证书是身份权威颁发的且客户端有合法 EK。

3) 安全属性证明

定理 1 的证明过程见附录 A，下面简介证明梗概。执行证书颁发协议时，身份权威 $\hat{I}$ 按以下步骤进行推理：①1~4 行 $\hat{I}$ 根据自身执行的动作得出其发送的消息中包含用 K_{EK} 加密的 $sessionKey_{IAC}$ 和用 $sessionKey_{IAC}$ 加密的 $cert_c$；②5~7 行根据客户端 $\hat{C}$ 持有 $cert_c$ 且 $cert_c$ 包含 $\hat{I}$ 的签名得出 $cert_c$ 由 $\hat{I}$ 生成，并且 $\hat{I}$ 发送了包含 $cert_c$ 的消息；③第 8 行 $\hat{I}$ 根据自己是用 $sessionKey_{IAC}$ 加密发送的 $cert_c$ 得出存在某个实体 $\hat{X}$ 用 $sessionKey_{IAC}$ 解密了 $cert_c$，$\hat{X}$ 是 $\hat{I}$ 或 $\hat{C}$；④9~11 行 $\hat{I}$ 已知自己没有解密过 $cert_c$ 得出 $\hat{X}=\hat{C}$，并根据不变量 Γ_{CI} 得出 $\hat{C}$ 已知 $sessionKey_{IAC}$；⑤12~15 行 $\hat{I}$ 根据自己是用 K_{EK} 加密发送的 $sessionKey_{IAC}$ 得出存在某个实体 $\hat{X}$ 用 K_{EK}^{-1} 解密了 $sessionKey_{IAC}$，$\hat{X}$ 是 $\hat{I}$ 或 $\hat{C}$，而 $\hat{I}$ 已知自己没有解密过 $sessionKey_{IAC}$ 得出 $\hat{X}=\hat{C}$，并根据不变量 Γ_{CI} 得出 $\hat{C}$ 已知 K_{EK}^{-1}；⑥综合第 7 行和第 15 行的证明得出如果 $\hat{C}$ 持有 $cert_c$，那么 $\hat{C}$ 已知 K_{EK}^{-1} 且 $cert_c$ 由 $\hat{I}$ 生成。

4.1.3 令牌颁发协议分析

令牌颁发协议客户端和身份权威角色程序的形式化描述见附录 B。下面给出协议的不变量 Γ_{TI} 和对身份权威的令牌颁发正确性保证。

不变量 Γ_{TI} 描述如果线程签名随机数 n，那么该线程在发送消息 $(\hat{C},\hat{I},cert_C.sn,SIG_{K_C^{-1}}(n))$ 前验证了身份权威对其公钥的签名

$$\Gamma_{TI} \equiv Honest(\hat{C}) \wedge Sign(C,SIG_{K_C^{-1}}(n)) \supset Verify(C,SIG_{K_{IA}^{-1}}(K_C)) < Send(C,\hat{C},\hat{I},cert_C.sn,SIG_{K_C^{-1}}(n)) \quad (3)$$

定理 2 令牌颁发正确性。执行身份权威角色，令牌颁发协议保证身份令牌颁发正确性，即 $TI \vdash [IdentityAuthority_{TI}]_I \phi_{TI,correct}$

$$\phi_{TI,correct} \equiv Honest(\hat{C}) \wedge Has(\hat{C},token_C) \supset Has(\hat{C},K_C^{-1}) \wedge Has(\hat{C},cert_C) \wedge Send(I,msg) \wedge Contains(msg,cert_C) \quad (4)$$

$Client_{CI} \equiv (\hat{I},K_{IA})[$
$new\ \{K_C,K_C^{-1}\};$
$identityBinding := sign(K_{IA},K_C),K_C^{-1};$
$identityProof := (K_C,identityBinding,cert_{EK});$
$new\ sessionKey_{CIA};$
$symBlob := \text{symenc}\ identityProof, sessionKey_{CIA};$
$asymBlob := pkenc\ sessionKey_{CIA},K_{IA};$
$send\ \hat{C},\hat{I},symBlob,asymBlob;$

$receive\ \hat{I},\hat{C},symIaBlob,asymIaBlob;$
$asymIaBlobContents := pkdec\ asymIaBlob,K_{EK}^{-1};$
$match\ asymIaBlobContents/(keyDigest,sessionKey_{IAC});$
$kDigest := hash\ K_C;$
$match\ keyDigest/kDigest;$
$cert_C := symdec\ symIaBlob,sessionKey_{IAC};$
$]_C(cert_C)$

$IdentityAuthority_{CI} \equiv (K_{EK})[$
$receive\ \hat{C},\hat{I},symBlob,asymBlob;$
$sessionKey_{CIA} := pkdec\ asymBlob,K_{IA}^{-1};$
$identityProof := symdec\ symBlob,sessionKey_{CIA};$
$match\ identityProof/(K_C,identityBinding,cert_{EK});$
$verify\ identityBinding,(K_{IA},K_C),K_C;$
$new\ sn;$
$csig_C := sign\ K_C,K_{IA}^{-1};$
$cert_C := (sn,csig_C);$
$new\ sessionKey_{IAC};$
$symIaBlob := \text{symenc}\ cert_C,sessionKey_{IAC};$
$keyDigest := hash\ K_C;$
$asymIaBlob := pkenc(keyDigest,sessionKey_{IAC}),K_{EK};$
$send\ \hat{I},\hat{C},symIaBlob,asymIaBlob;$
$]_I()$

图 7 证书颁发协议角色

定理 2 表明执行令牌颁发协议后，如果客户端持有令牌，那么该令牌是身份权威颁发的并且客户端有合法身份证书和证书对应私钥。

4.1.4 身份验证协议分析

1) 不变量

身份验证协议有验证方、客户端和身份权威 3 个参与实体，其角色程序的形式化描述见附录 C。

不变量 $\Gamma_{IV,1}$ 描述诚实主体不能同时执行验证方和客户端角色，$\Gamma_{IV,2}$ 描述如果线程发送身份验证通过消息，那么它之前一定收到了对应的身份验证信息

$$\Gamma_{IV,1} \equiv Honest(\hat{V}) \wedge Send(V,\hat{V},\hat{I},n_V,HASH_{K_{CT}}(ID_C,n_V,n_C)) \supset$$

$$\neg Compute(V, HASH_{K_{CT}}(ID_C, n_V, n_C)) \quad (5)$$

$$\Gamma_{IV,2} \equiv Honest(\hat{I}) \wedge Send(I, (\hat{I}, \hat{V}, true)) \supset Receive(I, \hat{C}, \hat{I}, ID_C, n_{C,} \exp ire_C), HASH_{token_C.K_{CT}}(token_c ID_C, n_V, n_C) < Send(I, (\hat{I}, \hat{V}, "true")) \quad (6)$$

2) 安全属性

身份验证协议的安全目标形式化为身份验证正确性 $\phi_{IV,correct}$ 和身份匿名性 $\phi_{IV,anonymity}$，下面给出协议对身份权威和验证方的身份验证正确性保证，以及对客户端的身份匿名性保证。

定理 3　身份验证正确性-IA。执行身份权威角色，身份验证协议保证客户端平台身份验证正确性，即 $IV \vdash [IdentityAuthority_{IV}]_I \phi_{IV,correct}$

$$\phi_{IV,correct} \equiv Honest(\hat{V}) \wedge Honest(\hat{I}) \supset Has(\hat{C}, token_C) \quad (7)$$

定理 3 表明 I 执行身份权威角色，执行结束后被验证的客户端一定持有有效身份令牌。

定理 4　identity verification correctness-V. 执行验证方角色，如果 $\theta_{IV,V}$ 公式成立，那么身份验证协议保证客户端平台身份验证正确性，即 $IV \vdash \theta_{IV,V}[Verifier_{IV}]_V \phi_{IV,correct}$

$$\phi_{IV,correct} \equiv Honest(\hat{V}) \wedge Honest(\hat{I}) \supset Has(\hat{C}, token_C) \quad (8)$$

$$\theta_{IV,V} \equiv (\mathrm{Re}ceive(V, (\hat{I}, \hat{V}, m)) \supset \exists I.Send(I, (\hat{I}, \hat{V}, m))) \wedge (\mathrm{Re}ceive(I, (\hat{V}, \hat{I}, m)) \supset \exists V.Send(V, (\hat{V}, \hat{I}, m))) \quad (9)$$

定理 4 表明从 $\theta_{IV,V}$ 成立的状态开始执行验证方角色，在结束状态身份验证正确性得到保证，定理证明见附录 D。前提 $\theta_{IV,V}$ 是线程执行动作的初始状态，描述验证方与身份权威已建立安全信道。

定理 5　身份验证正确性-V。执行客户端角色，身份验证协议保证客户端平台身份匿名性，即 $IV \vdash [Client_{IV}]_C \phi_{IV,anonymity}$

$$\phi_{IV,anonymity} \equiv Honest(\hat{C}) \wedge Honest(\hat{I}) \supset Has(\hat{C}, token_C) \wedge Has(\hat{I}, token_C) \wedge (Has(\hat{X}, token_C) \supset \hat{X} = \hat{C} \vee \hat{X} = \hat{I}) \quad (10)$$

定理 5 说明执行身份验证协议，客户端身份令牌不会被客户端和身份权威以外的主体知晓。直观地，客户端仅使用令牌计算散列值，且诚实主体不会明文发送身份令牌，匿名性可直接由这 2 个事实得出，在此省略证明过程。

4.1.5　协议组合证明

本节利用阶段组合证明方法，组合 3 个子协议的证明得出平台身份颁发—验证协议（IIV，identity issue-verification）的安全属性，下面给出协议向身份权威保证身份验证正确性的定理和证明，省略身份匿名性定理和证明。

1) 子协议的不变量和安全属性如下。

$$\Gamma_{CI} \equiv Honest(\hat{C}) \wedge Decrypt(C, ENC_K(t)) \supset Has(\hat{C}, K^{-1}),$$

$$CI \vdash [IdentityAuthority_{CI}]_I \phi_{CI,correct};$$

$$\Gamma_{TI} \equiv Honest(\hat{C}) \wedge Sign(C, SIG_{K_C^{-1}}(n \supset)) Verify(C, SIG_{K_{IA}^{-1}}(K_C)) < Send(C, \hat{C}, \hat{I}, cert_C.sn, SIG_{K_C^{-1}}(n)),$$

$$TI \vdash [IdentityAuthority_{TI}]_I \phi_{TI,correct};$$

$$\Gamma_{IV,1} \equiv Honest(\hat{V}) \wedge Send(V, \hat{V}, \hat{I}, n_V, HASH_{K_{CT}}(ID_C, n_V, n_C)) \supset \neg Compute(V, HASH_{K_{CT}}(ID_C, n_V, n_C)),$$

$$IV \vdash [IdentityAuthority_{IV}]_I \phi_{IV,correct} \quad (11)$$

2) 可以验证各子协议遵守其他子协议的不变量，即

$$CI \vdash \Gamma_{TI} \wedge \Gamma_{IV,1}$$
$$TI \vdash \Gamma_{CI} \wedge \Gamma_{IV,1}$$
$$IV \vdash \Gamma_{CI} \wedge \Gamma_{TI} \quad (12)$$

3) 定理 2 和定理 3 中安全属性的前提条件 θ_{TI}、θ_{IV} 为空，所以下列公式成立

$$\phi_{CI,correntness} \supset \theta_{TI}$$
$$\phi_{TI,correctness} \supset \theta_{IV} \quad (13)$$

4) θ_{CI} 为空，所以对 3 个子协议的任意协议步骤 B 有 $\theta_{CI}[B]\theta_{CI}$ 成立；θ_{TI} 为空，所以对令牌颁发和身份验证协议的任意协议步骤 B' 有 $\theta_{TI}[B']\theta_{TI}$ 成立。

通过上述分析和验证步骤知，能应用阶段组合定理证明对于 3 个子协议的阶段组合协议 IIV 有

$IIV \vdash [IdentityAuthority_{IIV}]_I \phi_{IV,correct}$。

进一步验证对令牌颁发和身份验证协议的任意协议步骤 B 有 $\phi_{CI,correct}[B]\phi_{CI,correct}$ 成立，对身份验证协议的任意协议步骤 B' 有 $\phi_{TI,correct}[B']\phi_{TI,correct}$ 成立，可得 IIV 协议身份验证正确性安全属性 $\phi_{IIV,correct} \equiv \phi_{CI,correct} \wedge \phi_{TI,correct} \wedge \phi_{IV,correct}$。

定理 6 身份验证正确性。执行身份权威角色，平台身份颁发—验证协议保证身份验证正确性，即 $IIV \vdash [IdentityAuthority_{IIV}]_I \phi_{IIV,correct}$

$$\begin{aligned}\phi_{IIV,correct} \equiv\ & Honest(\hat{V}) \wedge Honest(\hat{C}) \supset \\ & (Has(\hat{C}, K_{EK}^{-1}) \wedge\ Has(\hat{C}, K_C^{-1}) \wedge \\ & Has(\hat{C}, cert_C) \wedge\ Has(\hat{C}, token_C) \\ & Send(I, msg_1) \wedge \\ & Contains(msg_1, cert_C) \wedge \\ & Send(I, msg_2) \wedge \\ & Contains(msg_2, token_C))\end{aligned} \tag{14}$$

定理 6 表明平台身份颁发—验证协议执行后，通过验证的客户端一定有有效 EK、身份证书、证书对应私钥和身份令牌，且证书和令牌都是身份权威颁发的。

验证方仅参与了身份验证协议，在 4.1.4 节证明了该协议为其提供的安全保证，事实上经过分析可知如果 $\theta_{IV,V}$ 成立，即验证方与身份权威建立了安全信道，那么平台身份颁发—验证协议能为验证方提供与身份权威相同的安全属性。

4.2 性能分析

表 1 给出了 PIA 方案与 TPM 标准中的 PCA 方案，文献[6]中的 ePCA 方案在性能方面的详细比较，表中 √、×、— 分别表示方案具备、不具备和不涉及某个特性，HASH 代表散列函数输出数据的长度（1Hash≤512 bit）。从比较结果可以看出，本方案的优势在于以下几个方面：1）减轻了可信第三方的证书颁发负担，客户端可以基于同一证书进行多次证明；2）证明过程计算和通信效率高，2 种对比方案直接使用 AIK 证书作为证明信息，证书中仅 AIK 公钥一项的长度就是 2 048 bit，PIA 方案的证明信息长度仅为 1Hash，通信效率远高于 2 048 bit，并且 PIA 方案只需进行散列运算就可以验证客户端身份，计算效率高于对比方案采用的签名验证操作；3）安全性高，在 PIA 方案中基于相同身份证书的证明不会被验证方链接，并且方案在保证平台身份隐私性的情况下，实现了平台完整性验证，能够确保只有平台状态可信的客户端才能获得 token，而获得 token 的客户端只有在平台状态未发生改变时才能解封 token 进行身份证明。

表 1 方案性能比较

特性	PCA 方案	ePCA 方案	本方案
n 次证明申请证书数	n	n	1
expire 时间内 n 次证明申请令牌数	—	—	1
证明信息长度	>2 048 bit	>2 048 bit	1Hash
身份验证运算	签名验证	签名验证	散列
验证效率	低	低	高
同一 AIK 证书证明的不可链接性	×	×	√
同一令牌证明的不可链接性	—	—	√
平台身份隐私保护	√	√	√
AIK 与 EK 绑定性	×	√	√
平台状态验证	×	×	√
安全级别	低	中	高

5 系统实现

本文开发了 PIA 方案的原型系统并在无线局域网环境内进行了部署和测试。

5.1 系统组件

客户端。客户端使用 wpa_supplicant 开源软件利用无线网络与其他参与方进行通信。客户端平台身份证明组件实现为浏览器插件的形式，访问验证方服务时，该插件分别与 V 和 IA 交互完成平台身份证明，证明时用 HMAC（hash-based message authentication code）算法计算散列值发送给 V 和 IA。

身份权威。身份权威原型实现为一个 Web 服务，等待来自 C 或 V 的服务请求，收到请求后根据请求类型创建服务进程进行处理。该原型使用开源 Java 密码分组 bouncy castle 生成平台身份证书，证书签名算法采用椭圆曲线数字签名算法（ECDSA, elliptic curve digital signature algorithm）。

验证方。验证方原型为实验室内部资源网站，提供邮件、FTP 等功能，在其上增加平台身份验证模块，用户登录时在原有用户身份认证之外，还必须通过平台身份认证，之后才能访问网站服务。验证模块实现为一个 Web 服务，收到验证请求后按照身份验证协议验证平台身份，将验证结果返回给调

用该模块的服务。

5.2　系统部署

原型系统部署方案如图 8 所示。C 和 IA 处在同一局域网内，C 与 IA 和 V 通过 Wi-Fi 通信，IA 和 V 间的可信信道为 TLS。

5.3　评估和分析

本节评估原型系统各组件的性能。组件部署情况如下：C 和 IA 运行在联想台式机上，主机装配有 2.80 GHz Intel Core2 Duo、4 GB RAM 和 ZTEIC TCM1.1，V 运行在配置为 2.4 GHz Intel(R) Pentium(R) 4、1 GB RAM 的台式机上。统计了 100 次证书颁发、令牌颁发和身份证明所需要的时间，取其平均值作为最终结果，并统计了并发客户端数目变化对上述实验的影响。原型系统实验结果以及与 PCA 方案的对比如图 9 所示。

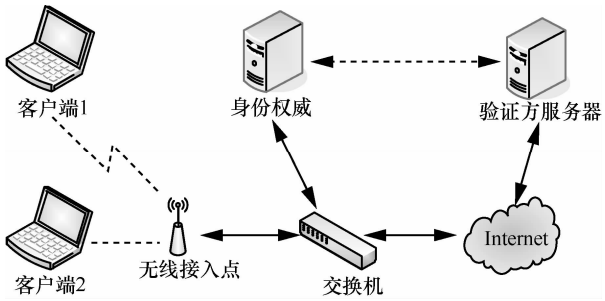

图 8　原型系统部署

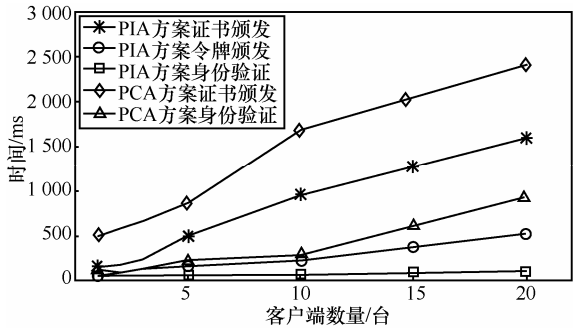

图 9　实验结果

身份颁发时间。在一台客户端的情况下，PCA 方案颁发平台身份证书需要 481 ms；PIA 方案中平台身份颁发分为颁发证书和令牌 2 个阶段，耗时分别为 138 ms 和 38 ms，2 部分时间之和远小于 PCA 方案。事实上，在 PIA 方案中，C 申请证书的操作并不频繁，通常仅在 PIA 创建阶段进行一次，之后只需定期向 IA 申请新的令牌，并不需要重复申请新证书，而其令牌颁发时间仅为 PCA 方案的 8%。在多台客户端并发时，虽然两方案的颁发时间均随客户端数量线性增加，但 PIA 方案的令牌颁发时间增长较缓慢。

身份验证时间。PCA 方案中，一台客户端的平台身份验证需要 101 ms，且验证时间随并发客户端数量线性增加；PIA 方案验证一台客户端的平台身份只需 49 ms，而该方案的身份验证时间不随并发客户端数量而变化，由此可见 PIA 方案的平台身份验证算法效率高，IA 能快速处理验证请求，不会成为系统性能瓶颈。

原型系统的实验结果表明，PIA 方案的平台身份颁发和验证性能均优于 PCA 方案。PIA 方案获得性能提升的主要原因在于其在身份证书的基础上，引入了身份令牌，一方面降低了客户端的证书颁发需求，另一方面使客户端可以通过令牌直接完成身份证明，从而有效地提高了身份颁发和证明效率。但是在获得高性能的同时，与 PCA 方案相比，PIA 方案对可信第三方 IA 提出了更高的要求。首先，在身份颁发阶段，IA 需要额外维护一张已颁发的令牌列表。其次，为保证平台身份证明的隐私性，IA 需参与客户端平台身份证明，因此要求 IA 必须实时在线。

6　结束语

本文提出一种基于可信第三方的平台身份证明方案，使客户端能高效地向验证方证明自身是安全的可信计算平台，方案满足平台身份验证正确性和匿名性 2 个安全属性，既能向验证方保证通过身份验证的客户端是嵌入可信芯片且具有合法身份的平台，又能保证证明过程不会向验证方泄露客户端芯片的真实身份。与现有同类方案相比，本方案的平台身份证明过程的效率和安全性更高。此外，本文设计并实现了原型系统，实验结果表明该方案客户端身份证明速度快、计算和通信效率高。

附录 A　定理 1 证明

定理 1 证明过程如图 10 所示。

附录 B　令牌颁发协议建模

令牌颁发协议中客户端和身份权威角色程序的形式化描述如图 11 所示。

附录 C　身份验证协议建模

身份验证协议中验证方、客户端和身份权威角色程序的形式化描述如图 12 所示。

1) AAI, P1　$\top[IdentityAuthority_{CI}]_I\, Send(I,\hat{I},\hat{C},symIaBlob,asymIaBlob)$

2) 1, ARI　$\top[symIaBlob := symenc\ cert_C, sessionKey_{IAC};\ keyDigest := hash\ K_C;\ asymIaBlob := pkenc(keyDigest, sessionKey_{IAC}), K_{EK}]_I$
$Send(I,\hat{I},\hat{C},ENC_{sessionKey_{IAC}}(cert_C),ENC_{K_{EK}}(HASH(K_C),sessionKey_{IAC}))$

3) 2, ARI　$\top[csig_C := sign\ K_C, K_{IA}^{-1};\ cert_C := (sn, csig_C)]_I$
$Send(I,\hat{I},\hat{C},ENC_{sessionKey_{IAC}}(sn,SIG_{K_{IA}^{-1}}(K_C)),ENC_{K_{EK}}(HASH(K_C),sessionKey_{IAC}))$

4) 3, P1　$\top[IdentityAuthority_{CI}]_I\, Send(I,\hat{I},\hat{C},ENC_{sessionKey_{IAC}}(sn,SIG_{K_{IA}^{-1}}(K_C)),ENC_{K_{EK}}(HASH(K_C),sessionKey_{IAC}))$

5) 3　$\top[IdentityAuthority_{CI}]_I\, Has(\hat{C},cert_C) \supset Has(\hat{C},(sn,SIG_{K_{IA}^{-1}}(K_C)))$

6) 5, VER　$\top[IdentityAuthority_{CI}]_I\, Has(\hat{C},(sn,SIG_{K_{IA}^{-1}}(K_C))) \supset Send(I,msg) \wedge Contains(msg,SIG_{K_{IA}^{-1}}(K_C))$

7) 2, 5, 6　$\top[IdentityAuthority_{CI}]_I\, Has(\hat{C},cert_C) \supset Send(I,msg) \wedge Contains(msg,cert_C)$

8) 2　$\top[IdentityAuthority_{CI}]_I\, Has(\hat{C},cert_C) \supset \exists X.Decrypt(X,ENC_{sessionKey_{IAC}}(cert_C)) \supset \hat{X} = \hat{C} \vee \hat{X} = \hat{I}$

9) 8, AAI, P1　$\top[IdentityAuthority_{CI}]_I \neg Decrypt(I,ENC_{sessionKey_{IAC}}(cert_C)) \supset \hat{X} \neq \hat{I}$

10) 8, 9　$\top[IdentityAuthority_{CI}]_I\, Has(\hat{C},cert_C) \supset Decrypt(C,ENC_{sessionKey_{IAC}}(cert_C))$

11) 10, Γ_{CI}　$\top[IdentityAuthority_{CI}]_I\, Honest(\hat{C}) \wedge Has(\hat{C},cert_C) \supset Has(\hat{C},sessionKey_{IAC})$

12) 2, 11　$\top[IdentityAuthority_{CI}]_I\, Has(\hat{C},sessionKey_{IAC}) \wedge \exists X.Decrypt(X,ENC_{K_{EK}}(HASH(K_C),sessionKey_{IAC})) \supset \hat{X} = \hat{C} \vee \hat{X} = \hat{I}$

13) 12, AAI, P1　$\top[IdentityAuthority_{CI}]_I \neg Decrypt(I,ENC_{K_{EK}}(HASH(K_C),sessionKey_{IAC})) \supset \hat{X} \neq \hat{I}$

14) 11, 12, 13　$\top[IdentityAuthority_{CI}]_I\, Honest(\hat{C}) \wedge Has(\hat{C},cert_C) \supset Decrypt(C,ENC_{K_{EK}}(HASH(K_C),sessionKey_{IAC}))$

15) 14, Γ_{CI}　$\top[IdentityAuthority_{CI}]_I\, Honest(\hat{C}) \wedge Has(\hat{C},cert_C) \supset Has(\hat{C},K_{EK}^{-1})$

16) 7, 15　$\top[IdentityAuthority_{CI}]_I\, Honest(\hat{C}) \wedge Has(\hat{C},cert_C) \supset Has(\hat{C},K_{EK}^{-1}) \wedge Send(I,msg) \wedge Contains(msg,cert_C)$

图 10　定理 1 证明

$Client_{TI} \equiv (\hat{I}, K_{IA}, cert_C)[$
match $cert_C / (sn, csig_C)$;
verify $csig_C, K_C, K_{IA}$;
new n;
$sig_n := sign(PCRs, n), K_C^{-1}$;
send $\hat{C}, \hat{I}, sn, sig_n$;
receive $\hat{I}, \hat{C}, encToken$;
$decToken := pkdec\ encToken, K_{EK}^{-1}$;
match $decToken / (token_C, sig)$;
verify sig, n, K_{IA};
$]_C(token_C)$

$IdentityAuthority_{TI} \equiv (cert_C)[$
receive $\hat{C}, \hat{I}, sn, sig_n$;
match $sn / cert_C.sn$;
verify $cert_C.csig_C, K_C, K_{IA}$;
verify $sig_n, (PCRs, n), K_C$;
new $ID_C, K_{CT}, expire_C$;
$token_C := (ID_{IA}, ID_C, K_{CT}, expire_C)$;
$sig := sign\ n, K_{IA}^{-1}$;
$encToken := pkenc(token_C, sig), K_{EK}$;
send $\hat{I}, \hat{C}, encToken$;
$]_I()$

图 11　令牌颁发协议角色

$Verifier_{IV} \equiv (\hat{C}, \hat{I}, ID_{IA})[$
new n_V;
send $\hat{V}, \hat{C}, n_V$;
receive $\hat{C}, \hat{V}, nonceHash, ID$;
match ID / ID_{IA};
send $\hat{V}, \hat{I}, n_V, nonceHash$;
receive $\hat{I}, \hat{V}$, "true";
$]_V()$

$Client_{IV} \equiv (\hat{I}, token_C)[$
receive $\hat{V}, \hat{C}, n_V$;
match $token_C / (ID_{IA}, ID_C, K_{CT}, expire_C)$;
new n_C;
$nonceHash := hash(ID_C, n_V, n_C), K_{CT}$;
send $\hat{C}, \hat{V}, nonceHash, ID_{IA}$;
send $\hat{C}, \hat{I}, ID_C, n_C, expire_C, nonceHash$;
$]_C()$

$IdentityAuthority_{IV} \equiv (token_C)[$
receive $\hat{V}, \hat{I}, n_V, nonceHash$;
receive $\hat{C}, \hat{I}, ID_C, n_C, expire_C, nonceHash$;
match $ID_C / token_C.ID_C$;
match $expire_C / token_C.expire_C$;
$K_{CT} := token_C.K_{CT}$;
$nHash := hash(ID_C, n_V, n_C), K_{CT}$;
match $nonceHash / nHash$;
new "true";
send $\hat{I}, \hat{V}$, "true";
$]_I()$

图 12　身份验证协议角色

1) AA1　$\top[send\ \hat{I},\hat{V},"true"]_I\ Send(I,\hat{I},\hat{V},"true")$

2) 1,P1　$\top[IdentityAuthority_{IV}]_I Send(I,\hat{I},\hat{V},"true")$

3) AA1　$\top[receive\ \hat{C},\hat{I},ID_C,n_C,expire_C,nonceHash]_I\ Receive(I,\hat{C},\hat{I},ID_C,n_C,expire_C,nonceHash)$

4) 3,P1　$\top[IdentityAuthority_{IV}]_I Receive(I,\hat{C},\hat{I},ID_C,n_C,expire_C,nonceHash)$

5) 2,4,AA 4　$\top[IdentityAuthority_{IV}]_I Receive(I,\hat{C},\hat{I},ID_C,n_C,expire_C,nonceHash) < Send(I,\hat{I},\hat{V},"true")$

6) 5,AR1　$\top[match\ ID_C / token_C.ID_C;\ K_{CT} := token_C.K_{CT};\ nHash := hash(ID_C,n_V,n_C),K_{CT};$

$\top[match\ nonceHash/nHash]_I\ Receive(I,\hat{C},\hat{I},ID_C,n_C,expire_C,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C))$

7) 6,P1　$\top[IdentityAuthority_{IV}]_I Receive\ (I,\hat{C},\hat{I},ID_C,n_C,expire_C,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C))$

8) 5,7　$\top[IdentityAuthority_{IV}]_I\ Receive\ (I,\hat{C},\hat{I},ID_C,n_C,expire_C,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C)) < Send(I,\hat{I},\hat{V}"true")$

图 13　身份验证协议不变量 $\Gamma_{IV,2}$ 证明

1) AA1　$\top[receive\ \hat{C},\ \hat{V},nonceHash,ID;match\ ID/ID_{IA}]_V\ Receive(V,\ \hat{C},\hat{V}\ nonceHash,ID_{IA})$

2) 1,P1　$\top[Verifier_{IV}]_V Has(\hat{C},ID_{IA})$

3) AA1　$\top[receive\ \hat{I},\ \hat{V},"true"]_V\ Receive(V,\ \hat{I},\ \hat{V},"true")$

4) 3,P1　$\top[Verifier_{IV}]_V\ Receive(V,\hat{I},\ \hat{V},"true")$

5) 4,θ_{IV}　$\top[Verifier_{IV}]_V Send(I,\hat{I},\ \hat{V},"true")$

6) 5,$\Gamma_{IV,2}$　$\top[Verifier_{IV}]_V Honest(\hat{I}) \supset$

$Receive(I,\hat{C},\ \hat{I},ID_C,n_C,expire_C,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C)) <\ Send(I,\hat{I},\ \hat{V},"true")$

7) 6,HASH3　$\top[Verifier_{IV}]_V Honest(\hat{I}) \supset \exists X.Computes(X,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C)) \wedge$

$Send(X,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C)) \wedge Has(\hat{X},expire_C)$

8) 7　$\top[Verifier_{IV}]_V Honest(\hat{I}) \supset \exists X.Computes(X,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C)) \supset\ \hat{X} = \hat{C} \vee\ \hat{X} = \hat{V}$

9) 7,AA1　$\top[send\ \hat{V},\hat{I},n_V,nonceHash]_V Send(V,\ \hat{V},\ \hat{I},n_V,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C))$

10) 9,P1　$\top[Verifier_{IV}]_V Send(V,\hat{V},\ \hat{I},n_V,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C))$

11) 8,10,$\Gamma_{IV,1}$　$\top[Verifier_{IV}]_V Honest(\hat{V}) \wedge\ Send(V,\hat{V},\ \hat{I},n_V,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C)) \supset\ \hat{X} \neq \hat{V}$

12) 7,8,11　$\top[Verifier_{IV}]_V Honest(\hat{V}) \wedge\ Honest(\hat{I}) \supset \exists X.Computes(X,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C)) \wedge$

$Send(X,HASH_{Token_C.K_{CT}}(Token_C.ID_C,n_V,n_C)) \wedge Has(\hat{X},expire_C) \wedge\ \hat{X} = \hat{C}$

13) 12　$\top[Verifier_{IV}]_V Honest(\hat{V}) \wedge\ Honest(\hat{I}) \supset Computes(C,HASH_{token_C.K_{CT}}(token_C.ID_C,n_V,n_C)) \wedge Has(\hat{C},exp\ ire_C)$

14) 13,HASH1　$\top[Verifier_{IV}]_V Honest(\hat{V}) \wedge\ Honest(\hat{I}) \supset Has(\hat{C},(token_C.ID_C,n_V,n_C)) \wedge\ Has(\hat{C},token_C.K_{CT})$

15) 14,PROJ　$\top[Verifier_{IV}]_V Honest(\hat{V}) \wedge\ Honest(\hat{I}) \supset Has(\hat{C},token_C.ID_C) \wedge\ Has(\hat{C},token_C.K_{CT})$

16) 15,AR1　$\top[Verifier_{IV}]_V Honest(\hat{V}) \wedge\ Honest(\hat{I}) \supset\ Has(\hat{C},ID_C) \wedge\ Has(\hat{C},K_{CT})$

17) 2,16,12　$\top[Verifier_{IV}]_V Honest(\hat{V}) \wedge\ Honest(\hat{I}) \supset Has(\hat{C},token_C)$

图 14　定理 4 证明

附录 D　定理 4 证明

本附录描述定理 4 的证明细节，首先证明协议不变量，然后利用不变量证明协议属性。下面介绍不变量 $\Gamma_{IV,2}$ 的证明过程，$\Gamma_{IV,1}$ 的证明与之类似。分析身份验证协议序列可知，C 和 V 角色中不包含 $Send(I,\hat{I},\ \hat{R},"true")$ 动作，因此 $\Gamma_{IV,2}$ 对上述角色平凡成立，仅需要对 IA 角色的基本序列进行验证，验证过程如图 13 所示。定理 4 证明过程如图 14 所示。

参考文献：

[1]　冯登国, 秦宇, 汪丹等. 可信计算技术研究[J]. 计算机研究与发展, 2011, 48(8): 1332-1349.
FENG D G, QIN Y, WANG D, *et al*. Research on trusted computing technology[J]. Journal of Computer Research and Development, 2011, 48(8): 1332-1349.

[2]　BRICKELL E, CAMENISCH J, CHEN L Q. Direct anonymous attestation[A]. Proceedings of the 11th ACM Conference on Computer and Communications security[C]. Washington, DC, USA, 2004.

[3]　Trusted Computing Group. TPM main specification version 1.2[EB/OL]. http://www.trustedcomputinggroup.org/, 2011.

[4]　REID J, NIETO J M G, DAWSON E, *et al*. Privacy and trusted computing[A]. Proceedings of the 14th International Workshop on Database and Expert Systems Applications[C]. Prague, Czech Republic, 2003.

[5]　PIRKER M, TOEGEL R, HEIN D, *et al*. A privacy CA for anonymity

and trust[A]. Proceedings of the 2nd International Conference on Trusted Computing[C]. Oxford, UK, 2009.

[6] CHEN L Q, WARINSCHI B. Security of the TCG privacy-CA solution[A]. Proceedings of the 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous[C]. Hong Kong, China, 2010.

[7] CHEN L Q, LEE M F, WARINSCHI B. Security of the enhanced TCG privacy-CA solution[A]. Proceedings of the 6th International Conference on Trustworthy Global Computing[C]. Aachen, Germany, 2011.

[8] 杨力, 马建峰, 朱建明. 可信的匿名无线认证协议[J]. 通信学报, 2009, 30(9): 29-35.
YANG L, MA J F, ZHU J M. Trusted and anonymous authentication scheme for wireless networks[J]. Journal on Communications, 2009, 30(9): 29-35.

[9] 杨力, 马建峰, 裴庆祺等. 直接匿名的无线网络可信接入认证方案[J]. 通信学报, 2010, 31(8): 98-104.
YANG L, MA J F, PEI Q Q, *et al*. Direct anonymous authentication scheme for wireless networks under trusted computing[J]. Journal on Communications, 2010, 31(8): 98-104.

[10] 崔巍, 李益发, 斯雪明. 基于 Eucalyptus 的基础设施即服务云框架协议设计[J]. 电子与信息学报, 2012, 34(7): 1748-1754.
CUI W, LI Y F, SI X M. The protocol design of a Eucalyptus-based infrastructure-as-a-service (IaaS) cloud framework[J]. Journal of Electronics & Information Technology, 2012, 34(7): 1748-1754.

[11] WINKLER T, RINNER B. User-centric privacy awareness in video surveillance[J]. Multimedia Systems, 2012, 18(2): 99-121.

[12] WINKLER T, RINNER B, ESTERLE L, *et al*. Privacy and security in video surveillance[J]. IEEE Signal Processing Magazine, 2013, 30: 190-198.

[13] PIRKER M, WINTER J, TOEGL R. Lightweight Distributed Heterogeneous Attested Android Clouds. Trust and Trustworthy Computing[M]. Springer Berlin Heidelberg, 2012.122-141.

[14] FONGEN A, MANCINI F. Attested genuineness in service oriented environments[A]. Proceedings of the 3rd International Conference on Digital Information Processing and Communications [C]. 2013.8-17.

[15] KRAXBERGER S, TOEGL R, PIRKER M, *et al*. Trusted Identity Management for Overlay Networks[M]. Information Security Practice and Experience, Springer Berlin Heidelberg, 2013.16-30.

[16] 陈小峰, 冯登国. 一种多信任域内的直接匿名证明方案[J]. 计算机学报, 2008, 31(7): 1122-1130.
CHEN X F, FENG D G. A direct anonymous attestation scheme in multi-domain environment[J]. Chinese Journal of Computers, 2008, 31(7): 1122-1130.

[17] BRICKELL E, CHEN L Q, LI J T. A new direct anonymous attestation scheme from bilinear maps[A]. Proceedings of the 1st International Conference on Trusted Computing and Trust in Information Technologies[C]. Villach, Austria, 2008.

[18] CHEN L Q, PAGE D, SMART N P. On the design and implementation of an efficient DAA scheme[A]. Proceedings of the 9th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Application[C]. Passau, Germany, 2010.

[19] CHEN X F, FENG D G. Direct anonymous attestation for next generation TPM[J]. Journal of Computers, 2008, 3(12): 43-50.

[20] BRICKELL E, LI J T. Enhanced privacy ID from bilinear pairing[EB/OL]. http://eprint. iacr.org/2009/095, 2011.

[21] CHEN L Q. A DAA scheme requiring less TPM resources[A]. Proceedings of the 5th International Conference on Information Security and Cryptology[C]. Beijing, China, 2009.

[22] BRICKELL E, LI J T. A pairing-based DAA scheme further reducing TPM resources[A]. Proceedings of the 3rd International Conference on Trust and Trustworthy Computing[C]. Berlin, Germany, 2010.

[23] 陈小峰, 冯登国. 一种基于双线性映射的直接匿名证明方案[J]. 软件学报, 2010, 21(8): 2070-2078.
CHEN X F, FENG D G. Direct anonymous attestation based on bilinear maps[J]. Journal of Software, 2010, 21(8): 2070-2078.

[24] 杨力, 马建峰, 姜奇. 无线移动网络跨可信域的直接匿名证明方案[J]. 软件学报, 2012, 5: 1260-1271.
YANG L, MA J F, JIANG Q. Direct anonymous attestation scheme in cross trusted domain for wireless mobile networks[J]. Journal of Software, 2012, 5:1260-1271.

[25] DURGIN N, MITCHELL J C, PAVOLVIC D. A compositional logic for proving security properties of protocols[J]. Journal of Computer Securtity, 2003, 11(4): 677-721.

[26] DATTA A, DEREK A, MITCHELL J C, *et al*. Protocol composition logic (PCL)[J]. Electronic Notes in Theoretical Computer Science, 2007, 172: 311-358.

[27] HE C H, SUNDARARAJAN M, DATTA A, *et al*. A modular correctness proof of IEEE 802.11i and TLS[A].Proceedings of the 12th ACM Conference on Computer and Communications Security[C]. Alexandria, USA, 2005. 2-15.

作者简介：

张倩颖（1986-），女，河北三河人，中国科学院博士生，主要研究方向为网络与系统安全、可信计算。

冯登国（1965-），男，陕西靖边人，博士，中国科学院研究员、博士生导师，主要研究方向为密码学和信息安全。

赵世军（1985-），男，山东潍坊人，中国科学院博士生，主要研究方向为网络与系统安全、可信计算。