软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(6):1523−1537 [doi: 10.13328/j.cnki.jos.004992]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

云环境下基于 PTPM 和无证书公钥的身份认证方案*

王中华[1], 韩 臻[1], 刘吉强[1], 张大伟[1], 常 亮[2]

[1](北京交通大学 计算机与信息技术学院,北京 100044)

[2](广西可信软件重点实验室(桂林电子科技大学),广西 桂林 541004)

通信作者: 王中华, E-mail: wangzhonghua@bjtu.edu.cn

摘 要: 为了解决目前云环境下用户与云端之间进行身份认证时所存在的安全问题和不足,将 PTPM(portable TPM)和无证书公钥密码体制应用到云环境中,提出一种用于实现用户与云端之间双向身份认证的方案.与现有方案相比,新方案具有以下特点:在通过建立身份管理机制实现用户和云端身份唯一性的基础上,首先利用 PTPM 不仅确保了终端平台的安全可信和云端与用户之间认证结果的真实正确,而且支持用户利用任意终端设备来完成与云端的身份认证过程;其次,新方案基于无证书公钥签名算法实现了"口令+密钥"的双因子认证过程;最后,通过安全性理论证明和性能分析,证明所提方案在保证 EUF-CMA 安全性的同时,显著提高了用户和云端之间身份认证的计算效率.

关键词: 云计算;身份认证;便携式 TPM;无证书公钥密码

中图法分类号: TP309

中文引用格式: 王中华,韩臻,刘吉强,张大伟,常亮.云环境下基于 PTPM 和无证书公钥的身份认证方案.软件学报,2016,27(6):1523−1537. http://www.jos.org.cn/1000-9825/4992.htm

英文引用格式: Wang ZH, Han Z, Liu JQ, Zhang DW, Chang L. ID authentication scheme based on PTPM and certificateless public key cryptography in cloud environment. Ruan Jian Xue Bao/Journal of Software, 2016,27(6):1523−1537 (in Chinese). http://www.jos.org.cn/1000-9825/4992.htm

ID Authentication Scheme Based on PTPM and Certificateless Public Key Cryptography in Cloud Environment

WANG Zhong-Hua[1], HAN Zhen[1], LIU Ji-Qiang[1], ZHANG Da-Wei[1], CHANG Liang[2]

[1](School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

[2](Guangxi Key Laboratory of Trusted Software (Guilin University of Electronic Technology), Guilin 541004, China)

Abstract: To tackle the problems of security threat and the shortcomings in the process of ID authentication between user and cloud, this paper applies Portable TPM chip and certificateless public key cryptography for the first time to solve the issues in the cloud environment, and proposes a scheme for bidirectional ID authentication between user and cloud. Compared with previous authentication schemes, the proposed scheme has the several advantages. First, based on the unique identity of user and cloud by the identity management mechanism, portable TPM can not only achieves secure and trusted terminal platform, which ensures the authentication result between user and cloud is correct and valid, but also supports the objectives of ID authentication between user and cloud in user's any terminal device. Furthermore, Dual-factor ID authentication (password + key) is implemented with certificateless public key signature algorithm provided by the new scheme. Finally, security proof and performance analysis show that this proposed scheme has the security level of EUF-CMA, and the computation overhead of ID authentication between user and cloud is significantly improved.

Key words: cloud computing; ID authentication; PTPM; certificateless public key cryptography

* 基金项目: 国家自然科学基金(61502486)

Foundation item: National Natural Science Foundation of China (61502486)

收稿时间: 2015-08-10; 修改时间: 2015-10-09; 采用时间: 2015-12-05; jos 在线出版时间: 2016-01-21

CNKI 网络优先出版: 2016-01-22 10:14:36, http://www.cnki.net/kcms/detail/11.2560.TP.20160122.1014.002.html

1 引 言

云计算是一种基于因特网提供存储和计算等资源的新兴服务模式.借助于云服务,企业、组织和个人用户能够方便快捷地进行海量数据计算和数据存储共享等操作.但是,云服务提供商 CSP(cloud service provider)首先需要对使用云服务的企业、组织和个人用户的身份进行认证,确定其正确性和合法性.否则,未申请注册或购买云服务的用户均可以使用云服务,从而一方面给 CSP 带来巨大的服务响应负担和严重的经济损失,同时合法用户可能会因没有得到及时的服务响应而造成计算结果和存储信息的丢失.同时,申请使用云服务的用户也需要对 CSP 的身份进行认证,否则黑客或恶意组织可以通过假冒 CSP 获取用户账号和隐私等重要信息,给用户带来严重的经济损失和信息泄露的威胁.因此,需要对 CSP 和使用云服务的用户的身份进行安全认证,确保二者身份的合法性和正确性.同时,云计算基于多种部署模式和服务模式为海量用户能够提供多种不同类型的服务,而这些服务可能来自不同的管理域,如果采用基于服务的身份认证机制,势必会造成认证过程的繁琐[1];此外,用户也会在不同的工作域(比如企业内部工作域和外部云工作域)中随时切换身份,如果每个工作域各自建立云用户身份管理机制,用户身份就会出现多重性,从而使用户认证和访问变得异常复杂[2−4].因此,与传统计算模式相比,云环境下的身份认证还需要考虑云用户身份管理的问题,通过建立身份管理机制来实现不同域内用户身份信息的唯一性,从而提高用户的使用体验和解决不同域内用户身份同步的问题.

在云环境中,由于企业、组织和个人用户可以利用包括 PC(personal computer)、PDA(personal digital assistant)、Laptop 和手机在内的终端设备来访问使用云服务,因此身份认证不仅涉及云端和终端设备之间的安全连接,还需要考虑用户与云端之间的安全连接.这是因为用户才是 CSP 的最终服务对象,终端设备只是用户的使用工具和服务平台.如图 1 所示,云端用于认证用户身份的结点服务器和用户终端设备均嵌入 TPM(trusted platform module)安全芯片来完成远程认证过程.虽然利用 TPM 芯片可以在服务器与终端设备之间建立可信连接[5],但如果用于实现用户认证过程,就会出现安全问题.这是因为如果用户使用的终端设备存在恶意软件,那么攻击者就可以通过篡改认证结果而欺骗用户,即不能将可信路径连接从终端设备安全地延伸到用户.此外,云环境下的用户可以使用任意终端设备来访问和使用云服务,如果用户利用 TPM 加密存储密钥或其他数据在某台终端设备,当其试图在其他终端设备上使用时,就需要进行数据迁移操作,而这会给用户带来复杂的操作过程甚至造成用户的隐私泄露.因此,实现云端与用户之间的身份认证一方面需要保证认证结果的真实性,另一方面需要支持用户可以利用任意终端设备来完成身份认证过程.

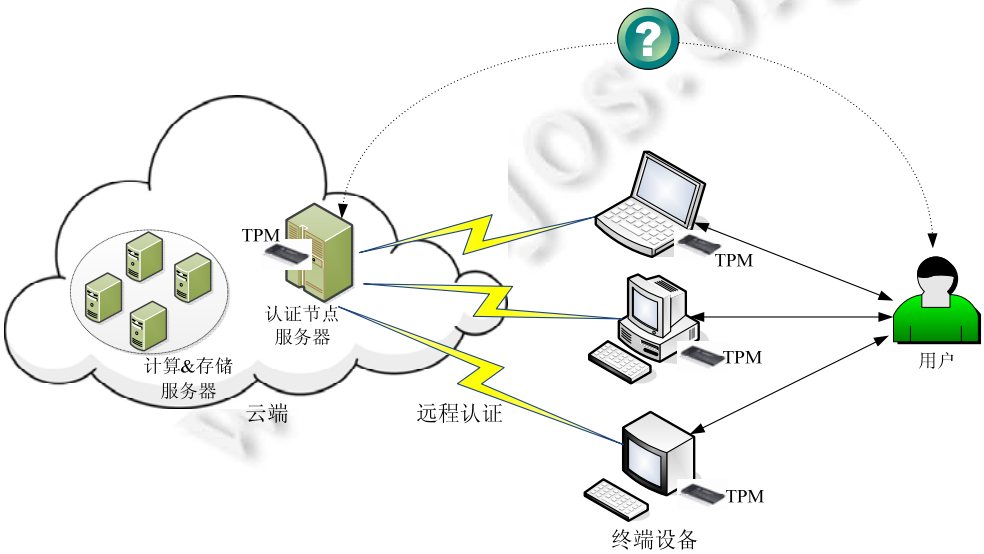


Fig.1 ID authentication based on TPM in cloud environment

图 1 云环境下基于 TPM 的身份认证

1.1 相关工作

近年来,国内外学者针对云环境下的身份认证问题已作了大量研究.文献[2,6–13]均采用基于证书的公钥密码体制[14]来解决用户与云端之间的身份认证问题.虽然采用公钥证书能够正确地实现云环境下的身份认证过程,但公钥证书的管理和维护会消耗巨大的计算资源[15];其次,用户使用的终端设备的安全性没有得到保障;另外云用户的身份管理问题也没有得到有效解决.文献[16–25]基于身份 ID 的密码体制[26]提出了用户和云端之间的身份认证方案.与公钥密码体制相比,基于身份 ID 的密码体制无需公钥证书的存在,从而解决了证书管理的问题.同时,文献[16,17,19,20,23]分别通过建立身份管理机制解决了云环境下用户身份唯一性的问题.但是由于第三方 PKG(private key generator)的引入,上述方案产生了密钥托管问题[15];同时如果 PKG 存在恶意行为,那么它就可以利用任何用户的私钥而伪造签名,达到欺骗验证方的目的;此外,与文献[2,6–13]相同,文献[16–25]也存在无法保证终端设备安全可信的问题.

Al-Riyamal 和 Perterson 于 2003 年提出了无证书公钥密码体制[27].它不仅避免了传统公钥密码体制的证书管理问题,而且解决了基于身份的密码体制的密钥托管问题.因此,与传统公钥密码体制和基于身份的密码体制相比,无证书公钥密码体制具有效率高和安全性强的优点.考虑到云计算具有资源共享和支持用户多种接入方式等特点,同时使用云服务的用户数量巨大,因此无证书公钥密码体制更加适合解决用户和云端之间的身份认证问题.文献[28,29]基于无证书密码体制提出了用户和云端之间的匿名身份认证方案.方案提出的身份认证过程均是通过验证由通信方身份 ID 和公钥值等信息生成的哈希值是否正确来实现的,攻击者通过中间人攻击就可以轻易攻破认证过程.此外,方案采用的是基于密钥的单因子认证过程;同时也没有考虑终端设备平台的安全问题,因此也就无法保证认证结果的真实性.

此外,根据富士通研究所作出的调查[30],88%的用户担心自己存储在云端的数据会被非授权访问.为了保证访问数据的用户的身份合法性,需要建立更加安全的用户认证机制,即文献[31]提出的多因子认证模式.因此,综上所述,针对现有工作在实现用户和云端之间的身份认证时所存在的问题和不足,本文基于 PTPM(portable TPM)[32,33]和无证书公钥签名算法,提出了一种支持云端与用户之间的双向身份认证方案.具体贡献如下:

(1) 首次将 PTPM 和无证书公钥签名算法相结合来解决云环境下用户和云端之间的身份认证问题;

(2) 基于分层 ID 树结构建立了包括用户和云端在内的身份管理机制,实现了任意通信实体身份唯一性的目标;

(3) 利用 PTPM 保证了终端平台的安全可信和云端与用户之间认证结果的真实正确;

(4) 实现了云端与用户之间“口令+密钥”的双因子认证过程;

(5) 支持用户利用任意终端设备来完成与云端的双向身份认证过程.

1.2 本文结构

本文第 2 节介绍方案所用到的相关基础知识.第 3 节详细描述本文提出的身份认证方案.第 4 节给出方案的安全性证明.第 5 节针对现有工作和本文方案进行对比分析.第 6 节给出结论.

2 相关基础知识

2.1 安全性理论假设

本方案的安全性基于 CDH(computational Diffie-Hellman)问题的困难性,相关定义如下[34].

定义 1. CDH 问题.已知 $a,b \xleftarrow{R} \mathbb{Z}_q^*, g$ 是生成元,给定(g,g^a,g^b),计算 g^{ab}.

这里,$a,b \xleftarrow{R} \mathbb{Z}_q^*$ 表示从符合均匀分布的 $\mathbb{Z}_q^*$ 中选取元素 a 和 b.

定义 2. CDH 假设.在概率多项式时间内算法 B 解决 CDH 问题的概率为

$$Adv_{\mathrm{CDH}}(B)=Pr[\ g^{ab} \leftarrow B(g,g^a,g^b)\ ],$$

若 $Adv_{\mathrm{CDH}}(B)$ 可忽略,则称 CDH 问题是困难的.

2.2 安全模型

本文设计提出的身份认证方案借鉴了无证书公钥签名算法的思想,因此根据文献[27]所定义的安全攻击模型,本文方案的安全性需要考虑如下两类敌手.

外部敌手 A_{I}: A_{I} 代表普通第三方攻击者,A_{I} 不具有系统主密钥,但可以使用任意值替换用户公钥;

内部敌手 A_{II}: A_{II} 代表恶意 KGC(key generating centre),A_{II} 具有系统主密钥,但不允许替换用户公钥.

2.3 PTPM

文献[35]提到 Intel 公司于 2002 年首次提出了便携式 TPM(portable TPM)的概念.与 TPM 相同,PTPM 也具有安全存储、密钥生成及数据签名等功能.根据文献[32,33]中的描述,由于 PTPM 通过 USB 接口或 PC 卡接口与终端设备通信,因此可以将可信计算平台的信任基础从平台本身转移到用户本身.每个用户都可以拥有标识自己身份的 PTPM,并可用于一台或多台终端设备上.此外,文献[32]实现的 PTPM 硬件模块还具有微型液晶窗口,这样就可以保证操作过程中计算结果的真实性和正确性.因此,用户利用 PTPM 一方面可以构建终端平台的可信链,实现对终端平台的完整性度量[36];另一方面,用户可以将密钥等重要数据安全存储在 PTPM 内,实现用户利用任意终端设备来完成身份认证的目的.

需要说明的是,由于本文研究的重点是云端与用户之间的身份认证问题,因此对于 TPM 和 PTPM 如何保证云端认证节点服务器和用户终端平台的安全可信就不再进行讨论研究.在后面认证方案的介绍中,可以认为云端与用户在进行身份认证时,已经利用文献[32,33]分别实现了基于 TPM 和 PTPM 的终端平台完整性度量过程.

3 身份认证方案设计

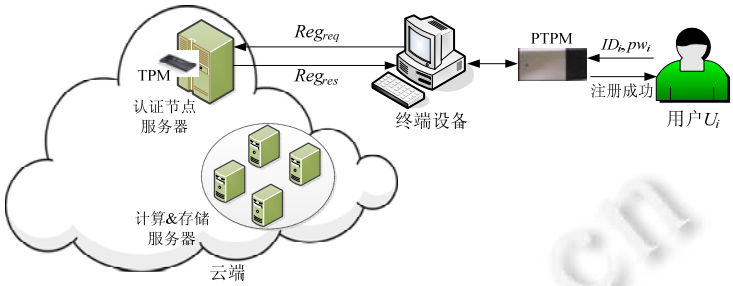
3.1 总体架构

如图 2 所示,用户持有 PTPM 硬件模块,云端认证节点服务器嵌入 TPM 安全芯片.用户与云端之间的双向身份认证过程包括图 2(a)所示的用户注册和图 2(b)所示的登录认证两个阶段.

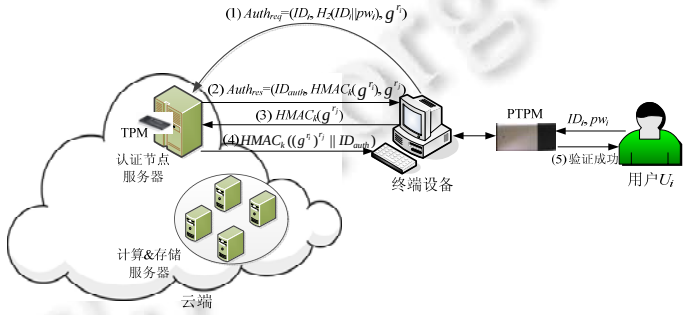
在注册阶段,用户 u_i 首先输入口令 pw_i 和身份 ID_i 等信息,然后利用 PTPM 计算得到注册请求信息 Reg_{req};认证节点服务器收到用户的注册请求信息 Reg_{req} 后,首先根据身份 ID_i 查询用户 u_i 是否已注册,然后输入 KGC 和 u_i 的公钥并利用 TPM 计算验证由 pw_i 和 ID_i 等信息生成的签名值是否正确,待验证正确后,认证节点服务器存储用户 u_i 的注册信息,并发送相应的注册响应信息 Reg_{res} 给 u_i;在收到注册响应信息 Reg_{res} 后,u_i 首先输入 KGC 和认证节点服务器的公钥并利用 PTPM 来验证认证节点服务器的签名值是否正确,如果正确则输出注册成功标志并存储认证节点服务器的身份 ID_{auth} 和秘密值等信息.

在认证阶段,用户 u_i 首先发送包括 ID_i、$H_2(ID_i||pw_i)$和 g^{r_i} 等认证请求信息 $Auth_{req}$ 给认证节点服务器;在验证收到的 $H_2(ID_i||pw_i)$值正确后,认证节点服务器首先计算 $HMAC_k(g^{r_i})$,其中 HMAC 运算所使用的密钥 k 取决于用户和认证节点服务器在注册阶段生成的秘密信息值,然后认证节点服务器发送 ID_{auth}、$HMAC_k(g^{r_i})$和 g^{r_j} 等认证响应信息 $Auth_{res}$ 给 u_i;u_i 在计算验证所收到的 $HMAC_k(g^{r_i})$值的正确性后,就完成了对认证节点服务器身份的认证,同时还需要计算 $HMAC_k(g^{r_j})$作为响应信息;而认证节点服务器通过验证 u_i 发送的 $HMAC_k(g^{r_j})$值是否正确来完成对用户 u_i 身份的认证,同时为了让 u_i 确认已通过认证,还需要再次发送 $HMAC_k((g^{r_i})^{r_j}\|ID_{auth})$ 值给 u_i,该 HMAC 值基于之前双方生成的随机数和 ID_{auth} 而计算得到;最终用户 u_i 在利用 PTPM 验证 $HMAC_k((g^{r_i})^{r_j}\|ID_{auth})$ 值的正确性后,输出验证成功标志到 PTPM 的显示窗口.

由于云环境中的用户可以使用任意终端设备来访问使用云服务,因此就出现了图3(a)所示的单用户利用多个终端设备和图 3(b)所示的多用户利用一台终端设备来完成用户与云端之间的身份认证过程.



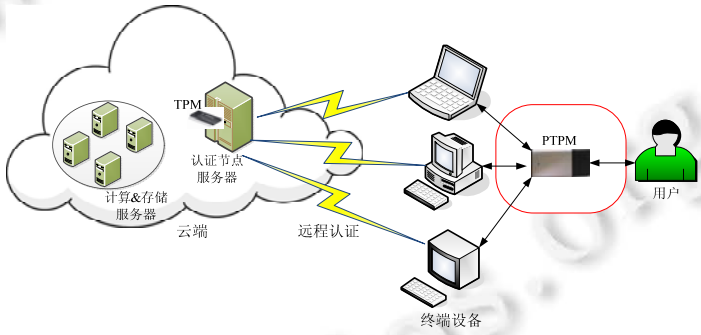
(a) 用户注册阶段



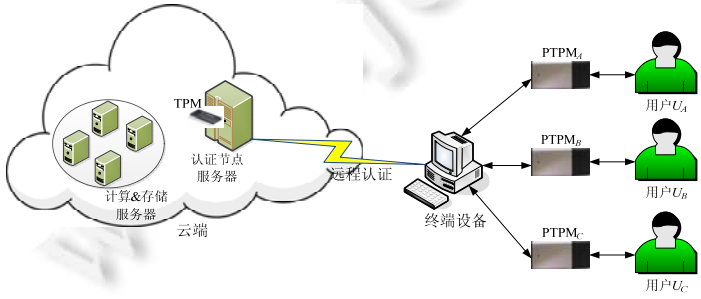
(b) 登录认证阶段

Fig.2 Bidirectional ID authentication between user and cloud

图 2 用户与云端之间的双向身份认证



(a) 单用户利用多个终端设备完成认证



(b) 多用户利用一台终端设备完成认证

Fig.3 ID authentication between user and cloud in any terminal device

图 3 用户利用任意终端完成身份认证

3.2 算法描述

3.2.1 系统建立

给定安全参数 K,选取 K 比特长的大素数 p.假设 G_1 和 G_2 均是阶为 p 的乘法循环群,g 是 G_1 的生成元.双线性映射 $e: G_1 \times G_1 \to G_2$.选择抗碰撞哈希函数 $H_1,H_2,H_1: \{0,1\}^* \to G_1,H_2: \{0,1\}^* \to G_1$.系统公开全局参数 $params$ 为(G_1,G_2,e,p,g,H_1,H_2).

3.2.2 身份 ID 生成

本文基于文献[23]提出的分层 ID 树结构来定义云环境中用户、云服务器等角色的身份 ID 值.整个分层结构由 2 层构成,根结点是 KGC,即生成用户部分私钥的第三方密钥生成中心;叶子结点表示在云端注册的终端用户和云端认证结点服务器.显然,分层 ID 树结构中的所有结点都有唯一的名称,从而实现了用户和云端服务器身份唯一性的目标.假设用户 u_i 的身份 $ID_i=DN_0||DN_i$,云端认证结点服务器 $server_{\text{auth}}$ 的身份 $ID_{\text{auth}}=DN_0||DN_{\text{server}}$,其中,$DN_0,DN_i,DN_{\text{server}}$ 分别表示 KGC、u_i 和 $server_{\text{auth}}$ 在分层 ID 树结构中所定义的名称,“||”表示字符串的拼接操作.

3.2.3 密钥生成

根据无证书公钥密码体制的思想,方案中叶子结点 $node_i$ 的密钥生成过程如下所述:

(1) $node_i$ 选取 $x_i \xleftarrow{R} \mathbb{Z}_p^*$ 作为秘密值,计算并公开公钥 $pk_i=g^{x_i}$.

(2) KGC 选取 $S_0 \xleftarrow{R} \mathbb{Z}_p^*$,$S_0$ 为 KGC 的主密钥,计算并公开公钥 $pk_{\text{KGC}}=g^{s_0}$.给定分层 ID 结构中的每个叶子结点 $node_i$,KGC 首先获取 $node_i$ 对应的身份 ID_i 和 pk_i 值,接着计算 $Q_i=H_1(ID_i)$,最后 KGC 利用 Q_i 计算并发送 $(g^{x_i})^{s_0}Q_i^{s_0}$ 给 $node_i$.

(3) $node_i$ 首先通过计算 $\frac{(g^{x_i})^{s_0}Q_i^{s_0}}{(g^{s_0})^{x_i}}$ 获得 KGC 生成的部分私钥 $Q_i^{s_0}$,然后通过查询分层 ID 树结构获取身份 ID_i,计算 $Q_i=H_1(ID_i)$并验证 $Q_i^{s_0}$ 值的正确性: $e(Q_i^{s_0},g)\overset{?}{=}e(Q_i,pk_{\text{KGC}})$,如果相等则生成私钥 $sk_i=(Q_i^{s_0},x_i)$.

在后面的叙述过程中,用户 u_i 的公私钥对分别表示为 $sk_i=(Q_i^{s_0},x_i),pk_i=g^{x_i}$;而云端认证结点服务器 $server_{\text{auth}}$ 的公私钥对分别表示为 $sk_{\text{server}}=(Q_j^{s_0},x_j),pk_{\text{server}}=g^{x_j}$.

3.2.4 用户注册

(1) 用户 u_i 输入 ID_i 和口令值 pw_i,利用 PTPM 首先计算 $H_2(ID_i||pw_i)$;然后选取 $S_i \xleftarrow{R} \mathbb{Z}_p^*$ 计算 g^{S_i} 和 $V=H_2(ID_i||pw_i)\,g^{S_i}$;最后 u_i 发送注册请求 $Reg_{req}=(ID_i,g^{S_i},V^{x_i}Q_i^{s_0},H_2(ID_i||pw_i))$给云端认证结点服务器 $server_{\text{auth}}$.

(2) $server_{\text{auth}}$ 收到 Reg_{req} 后,首先根据已注册用户信息表 $T_{register}$ 来查询 ID_i 是否存在,如果没有,则计算 $Q_i'=H_1(ID_i)$,然后验证 $V^{x_i}Q_i^{s_0}$ 值的正确性: $e(V^{x_i}Q_i^{s_0},g)\overset{?}{=}e(H_2(ID_i\,\|\,pw_i)g^{S_i},g^{x_i})\cdot e(Q_i',pk_{\text{KGC}})$,如果相等 $server_{\text{auth}}$ 选取 $S_j \xleftarrow{R} \mathbb{Z}_p^*$ 并利用 TPM 计算 g^{S_j},将 ID_i,S_j,g^{S_i} 和 $H_2(ID_i||pw_i)$存储到 $T_{register}$,然后利用 TPM 计算 $W=H_2(ID_{\text{auth}}\,\|\,pk_j)g^{S_j}$ 并发送注册响应信息 $Reg_{res}=(ID_{\text{auth}},g^{S_j},W^{x_j}Q_j^{s_0})$ 给 u_i;否则返回注册失败标志给 u_i.如果 $T_{register}$ 已存储该 ID_i 值,则返回已注册标志给 u_i.

(3) u_i 收到注册响应信息 Reg_{res} 后,首先利用 PTPM 计算 $Q_j'=H_1(ID_{\text{auth}})$和 $H_2(ID_{\text{auth}}\,\|\,pk_j)$,然后计算 $H_2(ID_{\text{auth}}\,\|\,pk_j)g^{S_j}$ 并通过判断等式 $e(W^{x_j}Q_j^{s_0},g)\overset{?}{=}e(H_2(ID_{\text{auth}}\,\|\,pk_j)g^{S_j},pk_j)\cdot e(Q_j',pk_{\text{KGC}})$ 是否成立来验证 $W^{x_j}Q_j^{s_0}$ 值的正确性.如果相等表示 u_i 在 $server_{\text{auth}}$ 处成功注册,PTPM 输出注册成功标志到显示窗口,同时 u_i 存储 ID_{auth},S_i 和 g^{S_j};否则输出注册失败标志.

3.2.5 登录认证

(1) 用户 u_i 首先输入 ID_i 和 pw_i,并利用 PTPM 计算 $H_2(ID_i||pw_i)$,同时选取 $r_i \xleftarrow{R} \mathbb{Z}_p^*$ 并计算 g^{r_i},然后 PTPM 发送登录认证请求 $Auth_{req}=(ID_i,H_2(ID_i||pw_i),g^{r_i})$给 $server_{\text{auth}}$.

(2) $server_{\text{auth}}$ 收到信息 $Auth_{req}$ 后,首先根据 ID_i 查询 $T_{register}$ 存储的 $H_2(ID_i||pw_i)$值与收到的是否相等,如果不等,$server_{\text{auth}}$ 返回口令错误信息给 u_i;否则 $server_{\text{auth}}$ 首先获取 ID_i 对应的 S_j 和 g^{S_i} 值,利用 TPM 计算 $(g^{S_i})^{S_j}$;然后选取 $r_j \xleftarrow{R} \mathbb{Z}_p^*$ 并利用 TPM 计算 $D_1=HMAC_k(g^{r_i})$和 g^{r_j},其中 $k=g^{S_iS_j}$;最后发送 $Auth_{res}=(ID_{\text{auth}}, g^{r_j}, D_1)$给 u_i.

(3) u_i 收到 $server_{\text{auth}}$ 返回的信息 $Auth_{res}$ 后,首先根据收到的 ID_{auth} 值查询获得对应的 S_i 和 g^{S_j},然后利用 PTPM 分别计算 $k'=(g^{S_j})^{S_i}$ 和 $D_1'=HMAC_{k'}(g^{r_i})$,验证 D_1 与 D_1' 是否相等.如果相等表示 u_i 完成了对 $server_{\text{auth}}$ 身份的认证,然后利用 PTPM 计算 $D_2=HMAC_{k'}(g^{r_j})$ 并发送给 $server_{\text{auth}}$;否则验证 $server_{\text{auth}}$ 身份失败,u_i 终止验证过程.

(4) $server_{\text{auth}}$ 利用 TPM 计算 $D_2'=HMAC_k(g^{r_j})$ 并与 D_2 进行比较.如果相等表示 $server_{\text{auth}}$ 完成了对 u_i 身份的认证,然后 $server_{\text{auth}}$ 利用 TPM 计算 $D_3=HMAC_k((g^{r_i})^{r_j}\|ID_{\text{auth}})$ 并发送给 u_i;否则验证 u_i 身份失败,$server_{\text{auth}}$ 终止验证过程.

(5) u_i 利用 PTPM 计算 $D_3'=HMAC_{k'}((g^{r_j})^{r_i}\|ID_{\text{auth}})$ 并与收到的 D_3 进行比较.如果相等,PTPM 输出验证成功标志到显示窗口;否则输出验证失败标志.

完成上述身份认证过程后,u_i 和 $server_{\text{auth}}$ 就可以利用会话密钥 $sk_{auth\leftrightarrow i}=g^{S_iS_j}g^{r_ir_j}$ 来进行后续信息交互过程.

3.2.6 口令更新

假设 u_i 需要将原有口令 pw_i 更新为 pw_i',那么 u_i 首先利用 PTPM 分别计算 $H_2(ID_i\|pw_i')$、$New_{pw}=sk_{\text{auth}\leftrightarrow i}\times H_2(ID_i\|pw_i')$ 和 $(g^{r_j})^{S_i}$,然后发送口令更新请求 $Update_{pw}=(ID_i,New_{pw},(g^{r_j})^{S_i})$给 $server_{\text{auth}}$,其中,×表示群 G_1 上的乘法运算.待 $server_{\text{auth}}$ 收到 $update_{pw}$ 后,首先根据 ID_i 查询存储的 g^{S_i} 值并利用 TPM 计算 $(g^{S_i})^{r_j}$,判断 $(g^{r_j})^{S_i}$ 与 $(g^{S_i})^{r_j}$ 是否相等.如果不等终止口令更新过程;否则利用 TPM 计算 $\frac{New_{pw}}{sk_{\text{auth}\leftrightarrow i}}$ 得到 $H_2(ID_i\|pw_i')$,通过查询 ID_i 将 $H_2(ID_i||pw_i)$替换为 $H_2(ID_i\|pw_i')$.

3.3 方案特点

本文提出的身份认证方案解决了云环境下云端与用户之间身份认证的问题.方案特点主要体现在:

(1) 在密钥生成阶段,用户获取 KGC 生成的部分私钥不需要用户和 KGC 之间建立安全信道,更符合云环境下公开通信的实际应用要求;

(2) 所有数据计算过程均在 TPM 或 PTPM 内完成,二者硬件的安全性确保了计算结果的正确和存储安全;

(3) 用户利用 PTPM 存储密钥等信息,由于 PTPM 具有携带方便的特点,用户就可以利用任意终端设备来完成注册和登录认证过程;

(4) 基于 HMAC 算法实现了身份认证过程,在保证认证结果正确性的同时,显著提高了认证双方的计算效率.

4 安全性证明

本文设计提出的身份认证方案是基于无证书公钥签名算法,同时根据第 3.2 节对方案算法的介绍,由于登录认证阶段的安全取决于 HMAC 算法的密钥值 $k=g^{S_iS_j}$,而 HMAC 算法的安全性已在文献[37]中得到证明.因此,只要攻击者在之前的用户注册阶段能够计算获得 k,那么就可以认为能够攻破本文提出的身份认证方案.而 k 值的安全性依赖于用户和云端之间利用无证书公钥签名算法完成用户注册的过程.如第 2.2 节所述,无证书公钥签名算法的攻击模型包括两种类型敌手,因此需要分别针对这两类敌手的攻击能力来给出方案的安全性证明过程.

定理 1. 假设 CDH 在群 G_1 上成立,对于攻击敌手 A_{I} 来说,本文方案基于 Random Oracle 模型在适应性选择消息攻击下具有不可伪造性(EUF-CMA).即如果任何外部敌手 A_{I} 在时间 t_{I} 内,以优势 ε_{I} 对散列函数 H_1、H_2、秘密值生成、公钥生成、KGC 生成部分私钥、公钥替换和签名生成等 Oracle 最多进行 q_{H_1} 次,q_{H_2} 次,q_{sv} 次,q_{pk}

次,q_{part} 次,q_{pkp} 次和 q_S 次问询后能够伪造签名,则存在算法 B_I,能够在时间 t_1 内以优势 ε_1 攻破群 G_1 上的 CDH 问题.其中,

$$\varepsilon_1 \geqslant \frac{(q_{part}+q_S)^{q_{part}+q_S}\varepsilon_I}{q_{H_1}(q_{part}+q_S+1)^{q_{part}+q_S+1}},$$

$$t_1 \leqslant t_I + (q_{H_1}+q_{part}+q_S)T_{G_1}+q_{H_2}+q_{sv}+q_{pk},$$

T_{G_1} 表示群 G_1 上的 1 次指数运算时间.

证明:以算法 B_I 为挑战者,选取 $a,b \xleftarrow{R} \mathbb{Z}_p^*$.给定$(g,g^a,g^b)$,$B_I$ 与敌手 A_I 进行如下 EUF-CMA 攻击游戏来获得 g^{ab},这里 a,b 均对 B_I 未知.

(1) 系统建立.B_I 发送公开的系统参数$(G_1,G_2,e,p,g,H_1,H_2,pk_{KGC})$给 A_I,其中,$pk_{KGC}=g^a$.B_I 控制 Random Oracle H_1 和 H_2,同时维护初始状态为空的散列值列表 H_1^{list} 和 H_2^{list},对于 A_I 的散列 Oracle 问询响应过程如下:

H_1 问询. A_I 请求身份 ID_i 的 H_1 值问询.假设 $Y_i \in \{0,1\}$,其中,$Pr[Y_i=1]=\alpha$.对于每次问询(ID_i,Y_i),B_I 选取 $r \xleftarrow{R} \mathbb{Z}_p^*$,如果 $Y_i=1$ 定义 $H_1(ID_i)=g^r$;否则定义 $H_1(ID_i)=(g^b)^r$.最后将$(ID_i,Y_i,H_1(ID_i))$添加到列表 H_1^{list} 中,并以 $H_1(ID_i)$为结果响应.

H_2 问询. A_I 请求身份 ID_i 和公钥 pk_i 的 H_2 值问询.如果列表 H_2^{list} 中存在元组(ID_i,pk_i,β),则返回预定义的输出值作为问询结果.否则,选取$\gamma \xleftarrow{R} G_1$,将$(ID_i,\gamma)$添加到列表 H_2^{list} 中,并以γ为结果响应.

(2) 阶段 1. A_I 发起一系列问询.B_I 维护初始状态为空的公钥列表 PK^{list},响应如下:

① 公钥问询 i.A_I 选取 ID_i,如果 PK^{list} 存在元组(ID_i,pk_i),返回 pk_i 作为结果响应;否则选取 $x_i \xleftarrow{R} \mathbb{Z}_p^*$,计算公钥 $pk_i=g^{x_i}$ 并返回给 A_I,并将(ID_i,x_i,pk_i)添加到列表 PK^{list} 中.

② 秘密值生成问询 i.A_I 选取 ID_i,B_I 提交 ID_i 给公钥问询 Oracle,并返回 x_i 作为结果响应.

③ 部分私钥生成问询 i.A_I 请求身份 ID_i 的部分私钥值问询.A_I 选取 ID_i,如果在列表 H_1^{list} 中查询 ID_i 对应的 $Y_i=1$,则计算 $(g^a)^r$ 作为结果响应;否则返回$\perp$.

④ 公钥替换问询 i.根据第 1.2 节敌手 A_I 的能力描述,A_I 可以替换任何实体的公钥值.假设 A_I 替换 ID_s 的公钥值.首先 A_I 选取 $x_s \xleftarrow{R} \mathbb{Z}_p^*$ 并计算 $pk_s=g^{x_s}$,然后发送(ID_s,x_s,pk_s)给 B_I.B_I 保存该元组值.

⑤ 签名生成问询 i.A_I 选取 ID_i,如果在列表 PK^{list} 存在 ID_i,通过秘密值生成问询得到 ID_i 的秘密值 x_i,同时在列表 H_1^{list} 中查询 ID_i 对应的 $Y_i=1$,则返回$U^{x_i}(g^a)^r$,这里 U 为 ID_i 在 H_2^{list} 中的输出值;否则返回$\perp$.

(3) 伪造. A_I 结束阶段 1 的问询,输出目标 ID_s 和伪造签名δ_s.B_I 做出如下响应过程:

① 从 H_1 问询中得到 ID_s 的 H_1 值 $(g^b)^r$;

② 输出伪造签名$\delta_s=U^{x_s}(g^{ab})^r$.

从而 B_I 能够通过计算 $g^{ab}=(\delta_s/U^{x_s})^{r^{-1}}$ 得到 g^{ab},因为δ_s、U、x_s 和 r 对于 B_I 来说是已知的,如果 A_I 赢得 EUF-CMA 攻击游戏,则有: $e(U^{x_s}(g^{ab})^r,g)=e(U,g^{x_s})\cdot e(g^{br},g^a)$,那么 B_I 能够攻破群 G_1 上的 CDH 问题.

现对 B_I 攻破群 G_1 上的 CDH 问题的概率进行分析:事件$\neg ppk_{abort}$ 表示 B_I 没有停止 A_I 对部分私钥生成的问询,事件$\neg sign_{abort}$ 表示 B_I 没有停止 A_I 对签名生成的问询,事件 $signErr$ 表示生成目标 ID_s 的伪造签名δ_s,事件 Err_{id} 表示列表 H_1^{list} 存储 ID_s,事件 Err_{yi} 表示 $Y_i=0$,*Succeed 表示* B_I 攻破群 G_1 上的 CDH 问题.根据模拟过程描述,事件 *Succeed* 可以表示为$\neg ppk_{abort} \wedge \neg sign_{abort} \wedge signErr \wedge Err_{id} \wedge Err_{yi}$.

当 $Y_i=0$ 时,B_I 会停止 A_I 对部分私钥生成的问询.由于 A_I 最多进行 q_{part} 次部分私钥生成问询,因此 $Pr[\neg ppk_{abort}] \geqslant (1-\alpha)^{q_{part}}$.

当 $Y_i=0$ 时,B_I 会停止 A_I 对签名生成的问询.由于 A_I 最多进行 q_S 次签名生成问询,因此 $Pr[\neg sign_{abort}] \geqslant (1-\alpha)^{q_S}$.

如果 $Err=\neg ppk_{abort} \wedge \neg sign_{abort} \wedge Err_{yi}$ 发生,那么 A_I 就认为模拟攻击和真实环境不可区分.由于 A_I 攻破方案的优势为ε_I,因此 $Pr[signErr|Err] \geqslant \varepsilon_I$.

由于 A_{I} 最多进行 q_{H_1} 次 H_1 问询,因此 $Pr[Err_{id}]\geqslant 1/q_{H_1}$.即:

$$\begin{aligned}Pr[Succeed]&=Pr[\neg ppk_{\mathrm{abort}}\wedge\neg sign_{\mathrm{abort}}\wedge signErr\wedge Err_{id}\wedge Err_{yi}]\\&=Pr[\neg ppk_{\mathrm{abort}}\wedge\neg sign_{\mathrm{abort}}\wedge Err_{yi}\wedge signErr]Pr[Err_{id}]\\&=Pr[Err]Pr[signErr|Err]Pr[Err_{id}]\\&=Pr[\neg ppk_{\mathrm{abort}}]Pr[\neg sign_{\mathrm{abort}}]Pr[Err_{yi}]Pr[signErr|Err]Pr[Err_{id}]\\&\geqslant\frac{(1-\alpha)^{q_{part}}(1-\alpha)^{q_S}\alpha\varepsilon_I}{q_{H_1}}\\&\geqslant\frac{(1-\alpha)^{q_{\mathrm{part}}+q_S}\alpha\varepsilon_I}{q_{H_1}}.\end{aligned}$$

当 $\alpha=\dfrac{1}{q_{\mathrm{part}}+q_S+1}$ 时,$\dfrac{(1-\alpha)^{q_{\mathrm{part}}+q_S}\alpha\varepsilon_I}{q_{H_1}}$ 存在最大值.因此 B_{I} 攻破群 G_1 上的 CDH 问题的概率为 $\varepsilon_1\geqslant\dfrac{(q_{\mathrm{part}}+q_S)^{q_{\mathrm{part}}+q_S}\varepsilon_I}{q_{H_1}(q_{\mathrm{part}}+q_S+1)^{q_{\mathrm{part}}+q_S+1}}$.

根据模拟过程描述,针对每次 H_1、KGC 生成部分私钥和签名生成等问询,B_{I} 需要分别额外进行 1 次群 G_1 上的指数运算,因此算法 B_{I} 的运行时间为 $t_1+(q_{H_1}+q_{\mathrm{part}}+q_S)T_{G_1}+q_{H_2}+q_{sv}+q_{pk}$. □

定理 2. 假设 CDH 在群 G_1 上成立,对于攻击敌手 A_{II} 来说,本文方案基于 Random Oracle 模型在适应性选择消息攻击下具有不可伪造性(EUF-CMA).即如果任何敌手 A_{II} 在时间 t_{II} 内,以优势 $\varepsilon_{\mathrm{II}}$ 对散列函数 H_1、H_2、秘密值生成、公钥生成、KGC 生成部分私钥和签名生成等 Oracle 最多进行 q_{H_1} 次,q_{H_2} 次,q_{sv} 次,q_{pk} 次,q_{part} 次和 q_S 次问询后能够伪造签名,则存在算法 B_{II},能够在时间 t_2 内以优势 ε_2 攻破群 G_1 上的 CDH 问题.其中,

$$\varepsilon_2\geqslant\frac{(q_{sv}+q_S)^{q_{sv}+q_S}\varepsilon_{\mathrm{II}}}{q_{H_2}(q_{sv}+q_S+1)^{q_{sv}+q_S+1}},$$

$$t_2\leqslant t_{\mathrm{II}}+(q_{H_2}+q_S)T_{G_1}+q_{H_1}+q_{sv}+q_{pk}.$$

证明:以算法 B_{II} 为挑战者,选取 $a,b\xleftarrow{R}\mathbb{Z}_p^*$.给定 (g,g^a,g^b),B_{II} 与敌手 A_{II} 进行如下 EUF-CMA 攻击游戏来获得 g^{ab},这里 a,b 均对 B_{II} 未知.

(1) 系统建立. B_{II} 发送公开的系统参数 $(G_1,G_2,e,p,g,H_1,H_2,pk_{\mathrm{KGC}})$ 给 A_{II},其中 $pk_{\mathrm{KGC}}=g^{s_0}$. B_{II} 控制 Random Oracle H_1 和 H_2,同时维护初始状态为空的散列值列表 H_1^{list} 和 H_2^{list},对于 A_{II} 的散列 Oracle 问询响应过程如下:

H_1 问询. A_{II} 请求身份 ID_i 的 H_1 值问询.B_{II} 计算 $Q_i=H_1(ID_i)$ 并返回给 A_{II}.

H_2 问询. A_{II} 请求身份 ID_i 和公钥 pk_i 的 H_2 值问询.假设 $Y_i\in\{0,1\}$,其中,$Pr[Y_i=1]=\alpha$.对于每次问询 (ID_i,pk_i,Y_i),B_{II} 选取 $r\xleftarrow{R}\mathbb{Z}_p^*$, 如果 $Y_i=1$ 定义 $H_2(ID_i\|pk_i)=g^r$;否则定义 $H_2(ID_i\|pk_i)=(g^b)^r$. 最后将 $(ID_i,pk_i,Y_i,H_2(ID_i\|pk_i))$ 添加到列表 H_2^{list} 中,并以 $H_2(ID_i\|pk_i)$ 为结果响应.

(2) 阶段 1. A_{II} 发起一系列问询.B_{II} 维护初始状态为空的公钥列表 PK^{list},响应如下:

① 公钥问询 $i.A_{\mathrm{II}}$ 选取 ID_i,如果 PK^{list} 存在元组 (ID_i,pk_i),返回 pk_i 作为结果响应;否则选取 $x_i\xleftarrow{R}\mathbb{Z}_p^*$, 计算公钥 $pk_i=g^{x_i}$ 并返回给 A_{II},并将 (ID_i,x_i,pk_i) 添加到列表 PK^{list} 中.

② 秘密值生成问询 $i.A_{\mathrm{II}}$ 选取 ID_i,B_{II} 提交 ID_i 给公钥问询 Oracle,并返回 x_i 作为结果响应.

③ 部分私钥生成问询 $i.A_{\mathrm{II}}$ 请求身份 ID_i 的部分私钥值问询.A_{II} 选取 ID_i,B_{II} 计算 $Q_i^{s_0}$ 并返回给 A_{II}.

④ 签名生成问询 $i.A_{\mathrm{II}}$ 选取 ID_i,如果在列表 H_2^{list} 中查询 ID_i 对应的 $Y_i=1$,则返回 $(g^a)^r Q_i^{s_0}$, 这里,a 表示用户的私钥值;否则返回 $\perp$.

(3) 伪造. A_{II} 结束阶段 1 的问询,输出目标 ID_s 和伪造签名 δ_s.B_{II} 做出如下响应过程:

① 从公钥生成问询中获得 ID_s 的公钥值 g^a;

② 从秘密值生成问询中获得 ID_s 的秘密值 a;

③ 从 H_2 问询中得到 ID_s 的 H_2 值 $(g^b)^r$;

④ 输出伪造签名 $\delta_s=(g^{ab})^r Q_i^{s_0}$.

从而 B_{II} 能够通过计算 $g^{ab}=(\delta_s/Q_i^{s_0})^{r^{-1}}$ 得到 g^{ab},因为 δ_s、$Q_i^{s_0}$ 和 r 对于 B_{II} 来说是已知的,如果 A_{II} 赢得 EUF-CMA 攻击游戏,则有: $e((g^{ab})^r Q_i^{s_0}, g)=e(Q_i, g^{s_0})\cdot e(g^{br}, g^a)$, 那么 B_{II} 能够攻破群 G_1 上的 CDH 问题.

现对 B_{II} 攻破群 G_1 上的 CDH 问题的概率进行分析:事件 $\neg SV_{\mathrm{abort}}$ 表示 B_{II} 没有停止 A_{II} 对秘密值生成的问询,事件 $\neg sign_{\mathrm{abort}}$ 表示 B_{II} 没有停止 A_{II} 对签名生成的问询,事件 $signErr$ 表示生成目标 ID_s 的伪造签名 δ_s,事件 Err_{id} 表示列表 H_2^{list} 存储 ID_s,事件 Err_{yi} 表示 Y_i=0,$Succeed$ 表示 B_{II} 攻破群 G_1 上的 CDH 问题.根据模拟过程描述,事件 $Succeed$ 可以表示为 $\neg SV_{\mathrm{abort}}\wedge\neg sign_{\mathrm{abort}}\wedge signErr\wedge Err_{id}\wedge Err_{yi}$.

当 Y_i=0 时,B_{II} 会停止 A_{II} 对秘密值生成的问询.由于 A_{II} 最多进行 q_{sv} 次秘密值生成问询,因此 $Pr[\neg SV_{\mathrm{abort}}]\geqslant(1-\alpha)^{q_{sv}}$.

当 Y_i=0 时,B_{II} 会停止 A_{II} 对签名生成的问询.由于 A_{II} 最多进行 q_S 次签名生成问询,因此 $Pr[\neg sign_{\mathrm{abort}}]\geqslant(1-\alpha)^{q_S}$.

如果 $Err=\neg SV_{\mathrm{abort}}\wedge\neg sign_{\mathrm{abort}}\wedge Err_{yi}$ 发生,那么 A_{II} 就认为模拟攻击和真实环境不可区分.由于 A_{II} 攻破方案的优势为 $\varepsilon_{\mathrm{II}}$,因此 $Pr[signErr|Err]\geqslant\varepsilon_{\mathrm{II}}$.

由于 A_{II} 最多进行 q_{H_2} 次 H_2 问询,因此 $Pr[Err_{id}]\geqslant 1/q_{H_2}$. 即:

$$\begin{aligned}Pr[Succeed]&=Pr[\neg SV_{\mathrm{abort}}\wedge\neg sign_{\mathrm{abort}}\wedge signErr\wedge Err_{id}\wedge Err_{yi}]\\&=Pr[\neg SV_{\mathrm{abort}}\wedge\neg sign_{\mathrm{abort}}\wedge Err_{yi}\wedge signErr]Pr[Err_{id}]\\&=Pr[Err]Pr[signErr|Err]Pr[Err_{id}]\\&=Pr[\neg SV_{\mathrm{abort}}]Pr[\neg sign_{\mathrm{abort}}]Pr[Err_{yi}]Pr[signErr|Err]Pr[Err_{id}]\\&\geqslant\frac{(1-\alpha)^{q_{sv}}(1-\alpha)^{q_S}\alpha\varepsilon_{\mathrm{II}}}{q_{H_2}}\\&\geqslant\frac{(1-\alpha)^{q_{sv}+q_S}\alpha\varepsilon_{\mathrm{II}}}{q_{H_2}}.\end{aligned}$$

当 $\alpha=\dfrac{1}{q_{sv}+q_S+1}$ 时, $\dfrac{(1-\alpha)^{q_{sv}+q_S}\alpha\varepsilon_{\mathrm{II}}}{q_{H_2}}$ 存在最大值.因此 B_{II} 攻破群 G_1 上的 CDH 问题的概率为

$$\varepsilon_2\geqslant\frac{(q_{sv}+q_S)^{q_{sv}+q_S}\varepsilon_{\mathrm{II}}}{q_{H_2}(q_{sv}+q_S+1)^{q_{sv}+q_S+1}}.$$

根据模拟过程描述, 针对每次 H_2 和签名生成等问询,B_{II} 需要分别额外进行 1 次群 G_1 上的指数运算,因此算法 B_{II} 的运行时间为 $t_{\mathrm{II}}+(q_{H_2}+q_S)T_{G_1}+q_{H_1}+q_{sv}+q_{pk}$. □

5 方案分析

5.1 效率分析

由于文献[28,29]和本文提出的方案均采用无证书公钥密码体制的思想来解决云端与用户之间的身份认证问题,因此本节给出这 3 种方案中用户和云端在计算和通信开销方面的性能分析.为了叙述方便,这里定义 EXP_{G_1} 表示群 G_1 上的指数运算, EXP_{G_2} 表示群 G_2 上的指数运算,$Pairing$ 表示双线性对运算, H_{G_1} 表示群 G_1 上的哈希运算,H 表示文献[28,29]中哈希值空间不是群 G_1 的哈希运算, M_{G_1} 表示群 G_1 上的乘(除)法运算, M_{G_2} 表示群 G_2 上的乘法运算,H_{mac} 表示 HMAC 运算.这里需要补充说明的是,文献[28,29]方案还涉及到异或操作,由于其运算代价非常小,因此在计算开销时忽略不计.

5.1.1 计算开销

根据文献[28]所述方案,在密钥生成阶段,用户和云端生成公私钥对均需要进行 3 次 M_{G_1} 运算,同时为了验

证 PKG 生成的部分私钥值的正确性,需要进行 2 次 *Pairing* 运算;在注册阶段,云端根据用户发送的身份 ID 来判断是否为授权用户,从而决定是否注册该用户 ID,不涉及任何计算操作过程(虽然高效,但存在严重的安全漏洞);在认证阶段,用户进行 $4Pairing+M_{G_1}+M_{G_2}+2EXP_{G_2}+6H$ 次运算,云端进行 $4Pairing+M_{G_1}+M_{G_2}+2EXP_{G_2}+6H+H_{G_1}$ 次运算.

对于文献[29]所提出的方案,在密钥生成阶段,用户和云端生成公私钥对均需要进行 1 次 M_{G_1} 运算,同时为了验证 PKG 生成的部分私钥值的正确性,需要进行 2 次 *Pairing* 运算;在注册阶段,与文献[14]相同,不涉及任何计算操作过程;在认证阶段,用户进行 $4Pairing+6M_{G_1}+2EXP_{G_2}+6H$ 次运算,云端进行 $3Pairing+5M_{G_1}+EXP_{G_2}+6H+2H_{G_1}$ 次运算.

下面重点分析文本方案的计算开销.在密钥生成阶段,用户和云端首先需要 1 次 EXP_{G_1} 运算来生成公钥,然后进行 $EXP_{G_1}+M_{G_1}$ 次运算来获得 *KGC* 发送的部分私钥,最后通过 $H_{G_1}+2Pairing$ 次运算来验证收到的部分私钥的正确性.在用户注册阶段,用户首先进行 $H_{G_1}+2(M_{G_1}+EXP_{G_1})$ 次运算来生成注册信息;然后云端进行 $3Pairing+M_{G_1}+M_{G_2}+H_{G_1}$ 次运算对注册信息完成验证,同时进行 $H_{G_1}+2(M_{G_1}+EXP_{G_1})$ 次运算生成返回信息;最后用户进行 $3Pairing+M_{G_1}+M_{G_2}+2H_{G_1}$ 次运算对云端的返回信息进行验证.在登录认证阶段,用户首先进行 $EXP_{G_1}+H_{G_1}$ 次运算来生成认证请求信息;然后云端进行 $2EXP_{G_1}+H_{\mathrm{mac}}$ 次运算生成认证信息;接着用户进行 $EXP_{G_1}+2H_{\mathrm{mac}}$ 次运算来对云端服务器的身份进行验证;云端再次进行 $EXP_{G_1}+2H_{\mathrm{mac}}$ 次运算来验证用户的身份;最终用户进行 $EXP_{G_1}+H_{\mathrm{mac}}$ 次运算来确定云端验证用户身份是否成功.当用户需要更新口令时,只需要进行 $EXP_{G_1}+M_{G_1}+H_{G_1}$ 次运算就可以向云端发送口令更新信息,而云端也只需进行 $EXP_{G_1}+M_{G_1}$ 次运算就可以完成口令更新过程.

下面给出文献[28,29]所述方案和本文提出的方案在用户和云端方面的计算开销对比,见表 1.通过表 1 可知,在密钥生成阶段,与文献[28,29]方案相比,本文方案中用户和云端需要进行额外的 $2EXP_{G_1}+H_{G_1}$ 次运算来获得 KGC 发送的部分私钥.虽然增加了计算过程,但不需要用户和 KGC 之间建立安全信道,因此更加符合云环境下公开通信的实际应用要求.其次,在注册阶段,虽然文献[28,29]方案不需要进行任何计算操作,但会带来巨大的安全漏洞,因为攻击者可以利用任意合法用户的 ID 来完成注册;同时由于用户注册是一次性的,因此本文方案注册过程中产生的计算开销对于用户和云端来说是可以接受的.在认证阶段,与文献[28,29]方案相比,本文方案由于没有涉及双线性对运算,计算效率得到了显著提高;同时考虑到用户和云端之间的认证过程可以进行多次,从而极大地减轻了用户和云端在认证过程中的计算负担.

Table 1 Comparison between Ref. [28,29] and our scheme in computation overhead

表 1 文献[28,29]方案和本文方案的计算开销对比

方案	密钥生成阶段		用户注册阶段		登录认证阶段	
	用户	云端	用户	云端	用户	云端
文献[28]	$2Pairing+3M_{G_1}$	$2Pairing+3M_{G_1}$	0	0	$4Pairing+M_{G_1}+M_{G_2}+2EXP_{G_2}+6\mathrm{H}$	$4Pairing+M_{G_1}+M_{G_2}+2EXP_{G_2}+6H+H_{G_1}$
文献[29]	$2Pairing+M_{G_1}$	$2Pairing+M_{G_1}$	0	0	$4Pairing+6M_{G_1}+2EXP_{G_2}+6H$	$3Pairing+5M_{G_1}+EXP_{G_2}+6H+2H_{G_1}$
本文	$2Pairing+M_{G_1}+2EXP_{G_1}+H_{G_1}$	$2Pairing+M_{G_1}+2EXP_{G_1}+H_{G_1}$	$3(Pairing+M_{G_1}+H_{G_1})+2EXP_{G_1}+M_{G_2}$	$3(Pairing+M_{G_1})+(2EXP_{G_1}+H_{G_1}+M_{G_2})$	$3(EXP_{G_1}+H_{\mathrm{mac}})+H_{G_1}$	$3(EXP_{G_1}+H_{\mathrm{mac}})$

5.1.2 通信开销

定义|*K*|表示安全参数 *K* 的长度,|*p*|表示 $\mathbb{Z}_p$ 中元素的长度,|*id*|表示用户身份 ID 的长度,|*hmac*|表示 HMAC 算法的信息长度.

根据文献[28]所述方案,在注册阶段,用户只需要发送身份 ID 值,因此通信开销为|*id*|,而云端不需要返回任何消息给用户,因此没有通信开销;在认证阶段,用户发送给云端的信息长度为 3|*p*|+|*id*|+2|*K*|,云端返回给用户的响应信息长度为 2|*p*|+|*id*|+|*K*|.

对于文献[29]所提出的方案,注册阶段的通信开销与文献[28]方案相同;在认证阶段,用户发送给云端的信息长度为 2|*p*|+|*id*|+|*K*|,云端返回给用户的响应信息长度为 2|*p*|+|*id*|.

对于本文方案而言,在用户注册阶段,用户发送的注册信息包括 ID_i, g^{S_i}, $V^{x_i}Q_i^{s_0}$ 和 $H_2(ID_i||pw_i)$总长度为 3|*p*|+|*id*|;云端返回的信息包括 ID_{auth}, g^{S_j} 和 $W^{x_j}Q_j^{s_0}$, 长度为 2|*p*|+|*id*|.在登录认证阶段,用户发送的登录认证请求包括 ID_i,$H_2(ID_i||pw_i)$和 g^{r_i},其长度为 2|*p*|+|*id*|;服务器端返回的信息包括 ID_{auth}, g^{r_j} 和 D_1,长度为|*p*|+|*id*|+|*hmac*|;然后用户发送|*hmac*|长度的信息给服务器端;最终服务器端发送|*hmac*|长度的信息给用户,用于确定云端验证用户身份是否成功.当用户需要更新口令时,用户发送的口令更新请求信息长度为 2|*p*|+|*id*|.

下面给出文献[28,29]所述方案和本文提出的方案在用户和云端方面的通信开销对比,见表 2.由表 2 可知,在用户注册阶段,与文献[28,29]方案相比,本文方案中的用户和云端需要给对方发送额外的信息来完成用户注册过程,但由于用户注册是一次性的,因此注册过程中的通信开销是可以接受的.在认证阶段,由于|*hmac*|值不可能大于|*K*|,因此与文献[28,29]方案相比,本文方案在认证过程中的通信开销至少保持不变.综上所述,对于用户和云端来说,本文方案所产生的通信开销代价是可以接受的.

Table 2 Comparison between Ref. [28,29] and our scheme in communication overhead

表 2 文献[28,29]方案和本文方案的通信开销对比

方案	用户注册阶段		登录认证阶段	
	用户	云端	用户	云端
文献[28]	|*id*|	0	3|*p*|+|*id*|+2|*K*|	2|*p*|+|*id*|+|*K*|
文献[29]	|*id*|	0	2|*p*|+|*id*|+|*K*|	2|*p*|+|*id*|
本文	3|*p*|+|*id*|	2|*p*|+|*id*|	2|*p*|+|*id*|+|*hmac*|	|*p*|+|*id*|+2|*hmac*|

5.2 其他性能分析

将文献[28,29]所述方案和本文方案在安全性能和灵活性上进行对比分析,其结果见表 3.

Table 3 Comparison between Refs. [28,29] and our scheme in security

表 3 文献[28,29]方案和本文方案的安全性能对比

方案	安全信道	双因子认证	终端平台安全	安全性证明
文献[28]	Yes	No	No	No
文献[29]	Yes	No	No	No
本文	No	Yes	Yes(PTPM)	Yes(EUF-CMA)

由表 3 可知,与文献[28,29]方案相比,首先本文方案中用户和云端在获取 *KGC* 发送的部分私钥过程中不需要安全信道的建立;其次方案支持云端与用户之间“口令+密钥”的双因子认证过程;第三,利用 PTPM 保证了终端平台的安全可信和云端与用户之间认证结果的真实正确;最后本文方案基于 Random Oracle 模型在适应性选择消息攻击下具有不可伪造性.因此,本文方案的安全性能全面优于文献[28,29]方案.这里我们需要指出,文献[28,29]方案实现了用户和云端之间匿名认证的目标,同时,文献[4,38,39]在解决云环境下的用户身份管理和访问控制等问题时也要求支持用户身份的匿名性.但本文在云环境下建立身份管理机制的基础上,主要解决如何安全有效地实现用户与云端之间身份认证的问题.对于用户身份匿名性的问题,可以作为后续工作来进行研究.

6 结束语

本文提出了基于 PTPM 和无证书公钥签名算法的身份认证方案.在实现用户和云端身份唯一性的基础上,

一方面利用 PTPM 实现了终端平台的安全可信和云端与用户之间认证结果的真实正确的目标,同时支持用户利用任意终端设备来完成与云端的双向身份认证过程;另一方面基于无证书公钥签名算法解决了传统公钥密码体制的证书管理问题和基于身份的密码体制的密钥托管问题.最后对方案的安全性进行了理论证明;同时通过与现有方案的效率和其他性能对比分析,本文提出的方案不仅显著提高了用户和云端之间身份认证的计算效率,而且更能适应云环境下公开通信的实际应用要求.下一步工作是考虑跨域间的身份认证和防止用户身份隐私泄露等问题.

References:

[1] Lin C, Su WB, Meng K, Liu Q, Liu WD. Cloud computing security: Architecture, mechanism and modeling. Chinese Journal of Computers, 2013,36(9):1765−1784 (in Chinese with English abstract).

[2] Luo DJ. Research on key issues in cloud computing security based on trusted computing [Ph.D. Thesis]. Guangzhou: South China University of Technology, 2014 (in Chinese with English abstract).

[3] Cloud Security Alliance. Domain 12: Guidance for Identity & Access Management V2.1. http://cloudsecurityalliance.org/guidance/csaguide-dom12-v2.10.pdf

[4] Feng CS, Qin ZG, Yuan D, Qing Y. Key techniques of access control for cloud computing. Acta Electronica Sinica, 2015,43(2): 312−319 (in Chinese with English abstract).

[5] Trusted Computing Group. Trusted platform modules strengthen user and platform authenticity. http://www.trustedcomputinggroup.org/files/resource_files/8D46621F-1D09-3519-ADB205692DBBE135/Whitepaper_TPMs_Stren-gthen_User_and_Platform_Authenticity_Final_1_0.pdf

[6] Binu S, Misbahuddin M, Raj P. A mobile based remote user authentication scheme without verifier table for cloud based services. In: Proc. of the 3rd Int'l Symp. on Women in Computing and Informatics (WCI 2015). New York: ACM Press, 2015. 502−509. [doi: 10.1145/2791405.2791487]

[7] Yassin AA, Jin H, Ibrahim A, Qiang WZ, Zou DQ. A practical privacy-preserving password authentication scheme for cloud computing. In: Proc. of the 26th Int'l Conf. on Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW). IEEE Computer Society Press, 2012. 1210−1217. [doi: 10.1109/IPDPSW.2012.148]

[8] Han W. Research on remote user authentication in the clouds [MS. Thesis]. Lanzhou: Lanzhou University of Technology, 2014 (in Chinese with English abstract).

[9] Hu Y. Research on identity authentication technologies in cloud [MS. Thesis]. Beijing: Beijing University of Technology, 2014 (in Chinese with English abstract).

[10] Yassin AA, Jin H, Ibrahim A, Qiang WZ, Zou DQ. Cloud authentication based on anonymous one-time password. In: Han YH, Park DS, Jia WJ, Yeo SS, eds. Proc. of the 7th Int'l Conf. on Ubiquitous Information Technologies & Applications (CUTE 2012). Netherlands: Springer Science+Business Media Dordrecht, 2013. 423−431. [doi: 10.1007/978-94-007-5857-5_46]

[11] Hong S. Two-Channel user authentication by using USB on cloud. Journal of Computer Virology and Hacking Techniques, 2015, 1−7. [doi: 10.1007/s11416-015-0254-y]

[12] Soares LFB. Secure authentication mechanisms for the management interface in cloud computing environments [MS. Thesis]. Covilhã: University of Beira Interior Engineering, 2013.

[13] Urien P, Marie E, Kiennert C. An innovative solution for cloud computing authentication: Grids of EAP-TLS smart cards. In: Proc. of the 5th Int'l Conf. on Digital Telecommunications (ICDT 2010). IEEE Computer Society Press, 2010. 22−27. [doi: 10.1109/ICDT.2010.12]

[14] Diffie W, Hellman ME. New directions in cryptography. IEEE Trans. on Information Theory, 1976,22(6):644−654. [doi: 10.1109/TIT.1976.1055638]

[15] Sang YX. Study on some topics of certificateless public-key cryptography [Ph.D. Thesis]. Xiamen: Xiamen University, 2009 (in Chinese with English abstract).

[16] Elbaz HA, Abdelaziz MH, Nazmy T. Trusting identity based authentication on hybrid cloud computing. In: Leung VCM, Chen M, eds. Proc. of Cloud Computing 2013. Springer-Verlag, 2014. 179−188. [doi: 10.1007/978-3-319-05506-0_17]

[17] Yan L, Rong CM, Zhao GS. Strengthen cloud computing security with federal identity management using hierarchical identity-based cryptography. In: Jaatun MG, Zhao GS, Rong CM, eds. Proc. of the Cloud Computing 2009. Berlin, Heidelberg: Springer-Verlag, 2009. 167−177. [doi: 10.1007/978-3-642-10665-1_15]

[18] Li XH, Yang B. Efficient identity-based signature authentication scheme in cloud service. Int'l Journal of Advancements in Computing Technology, 2013,5(5):867−876.

[19] Cao CL, Zhang R, Zhang MY, Yang YX. IBC-Based entity authentication protocols for federated cloud systems. KSII Trans. on Internet & Information Systems, 2013,7(5):1291−1312.

[20] Tian JF, Sun KH. Trust-Distributed-Based authentication mechanism using hierarchical identity-based cryptography. Journal of Computer Research and Development, 2015,52(7):1660−1671 (in Chinese with English abstract).

[21] Kang LS, Zhang XJ. Identity-βased authentication in cloud storage sharing. In: Proc. of the Int'l Conf. on Multimedia Information Networking and Security (MINES 2010). IEEE Computer Society Press, 2010. 851−855. [doi: 10.1109/MINES.2010.180]

[22] Mishra D, Kumar V, Mukhopadhyay S. A pairing-free identity based authentication framework for cloud computing. In: Lopez J, Huang XY, Sandhu R, eds. Proc. of the 7th Int'l Conf. on Network and System Security (NSS 2013). Berlin, Heidelberg: Springer-Verlag, 2013. 721−727. [doi: 10.1007/978-3-642-38631-2_62]

[23] Li HW, Dai YS, Tian L, Yang HM. Identity-Based authentication for cloud computing. In: Jaatun MG, Zhao GS, Rong CM, eds. Proc. of the 1st Int'l Conf. on Cloud Computing. Berlin, Heidelberg: Springer-Verlag, 2009. 157−166. [doi: 10.1007/978-3-642-10665-1_14]

[24] Chen TH, Yeh HL, Shih WK. An advanced ECC dynamic ID-based remote mutual authentication scheme for cloud computing. In: Proc. of the 5th Int'l Conf. on Multimedia and Ubiquitous Engineering (MUE 2011). IEEE Computer Society Press, 2011. 155−159. [doi: 10.1109/MUE.2011.69]

[25] Chen PL, Yang JH, Lin CI. ID-Based user authentication scheme for cloud computing. Journal of Electronic Science and Technology, 2013,11(2):221−224.

[26] Shamir A. Identity based cryptosystems and signature schemes. In: Blakley GR, Chaum D, eds. Proc. of the Advances in Cryptology: CRYPTO'84. Berlin, Heidelberg: Springer-Verlag, 1985. 47−53. [doi: 10.1007/3-540-39568-7_5]

[27] Al-Riyami SS, Paterson KG. Certificateless public key cryptography. In: Laih CS, ed. Proc. of the Advances in Cryptology: ASIACRYPT 2003. Berlin, Heidelberg: Springer-Verlag, 2003. 452−473. [doi: 10.1007/978-3-540-40061-5_29]

[28] Mishra R. Anonymous remote user authentication and key agreement for cloud computing. In: Pant M, Deep K, Nagar A, Bansal JC, eds. Proc of the 3rd Int'l Conf. on Soft Computing for Problem Solving. Springer-Verlag, 2014. 899−913. [doi: 10.1007/978-81-322-1771-8_78]

[29] Dong ZM, Zhang L, Li JT. Security enhanced anonymous remote user authentication and key agreement for cloud computing. In: Proc. of the 17th Int'l Conf. on Computational Science and Engineering (CSE 2014). IEEE Computer Society Press, 2014. 1746−1751. [doi: 10.1109/CSE.2014.320]

[30] Fujitsu Research Institute. Personal data in the cloud: A global survey of consumer attitudes. http://www.fujitsu.com/downloads/SOL/fai/reports/fujitsu_personal-data-in-the-cloud.pdf

[31] Choudhury AJ, Kumar P, Sain M, Lim H, Lee HJ. A strong user authentication framework for cloud computing. In: Proc. of the Asia-Pacific Services Computing Conf. (APSCC 2011). IEEE Computer Society Press, 2011. 110−115. [doi: 10.1109/APSCC.2011.14]

[32] Zhang DW, Han Z, Yan GW. A portable TPM based on USB key. In: Proc. of the 17th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2010. 750−752. [doi: 10.1145/1866307.1866419]

[33] Wu XW. Research and implementation of key technology on portable TPM [MS. Thesis]. Beijing: Beijing Jiaotong University, 2010 (in Chinese with English abstract).

[34] Bao F, Deng RH, Zhu HF. Variations of Diffie-Hellman problem. In: Qing SH, Gollmann D, Zhou JY, eds. Proc. of the 5th Int'l Conf. on Information and Communications Security. Berlin, Heidelberg: Springer-Verlag, 2003. 301−312. [doi: 10.1007/978-3-540-39927-8_28]

[35] Feng W, Feng DG, Wei G, Qin Y, Zhang QY, Chang DX. TEEM: A user-oriented trusted mobile device for multi-platform security applications. In: Proc. of the 6th Int'l Conf. on Trust and Trustworthy Computing. Berlin, Heidelberg: Springer-Verlag, 2013. 133−141. [doi: 10.1007/978-3-642-38908-5_10]

[36] Han L, Liu JQ, Zhang DW, Han Z, Wei XY. A portable TPM scheme for general-purpose trusted computing based on EFI. In: Proc of Int'l Conf. on Multimedia Information Networking and Security (MINES 2009). IEEE Computer Society Press, 2009. 140−143. [doi: 10.1109/MINES.2009.37]

[37] Bellare M. New proofs for NMAC and HMAC: Security without collision-resistance. In: Dwork C, ed. Proc. of the 26th Int'l Conf. on Advances in Cryptology-CRYPTO 2006. Berlin, Heidelberg: Springer-Verlag, 2006. 602−619. [doi: 10.1007/11818175_36]

[38] Nuñez D, Agudo I, Lopez J. Privacy preserving identity management as a service. In: Felici M, Gago CF, eds. Accountability and Security in the Cloud. Springer-Verlag, 2015. 114−125. [doi: 10.1007/978-3-319-17199-9_5]

[39] Nuñez D, Agudo I. BlindIdM: A privacy-preserving approach for identity management as a service. Int'l Journal of Information Security, 2014,13(2):199−215. [doi: 10.1007/s10207-014-0230-4]

附中文参考文献:

[1] 林闯,苏文博,孟坤,刘渠,刘卫东.云计算安全:架构、机制与模型评价.计算机学报,2013,36(9):1765−1784.

[2] 罗东俊.基于可信计算的云计算安全若干关键问题研究[博士学位论文].广州:华南理工大学,2014.

[4] 冯朝胜,秦志光,袁丁,卿昱.云计算环境下访问控制关键技术.电子学报,2015,43(2):312−319.

[8] 韩薇.云环境下远程用户身份认证技术研究[硕士学位论文].兰州:兰州理工大学,2014.

[9] 扈莹.云计算环境的身份认证的研究[硕士学位论文].北京:北京工业大学,2014.

[15] 桑永宣.无证书的公钥密码体制的若干问题的研究[博士学位论文].厦门:厦门大学,2009.

[20] 田俊峰,孙可辉.基于 HIBC 的云信任分散统一认证机制.计算机研究与发展,2015,52(7):1660−1671.

[33] 吴晓武.便携式 TPM 关键技术研究与实现[硕士学位论文].北京:北京交通大学,2010.

王中华(1983−),男,山东菏泽人,博士生,主要研究领域为可信计算,云计算安全.

张大伟(1974−),男,副教授,主要研究领域为可信计算,智能卡安全技术.

韩臻(1962−),男,教授,博士生导师,CCF 会员,主要研究领域为信息安全,可信计算,计算机图形学.

常亮(1980−),男,博士,教授,CCF 高级会员,主要研究领域为可信计算,知识表示与推理.

刘吉强(1973−),男,教授,博士生导师,CCF 会员,主要研究领域为可信计算,安全协议,隐私保护.