移动医疗系统中的可撤销无证书代理重签名方案

郭 瑞[1,2], 刘颖菲[1,2], 王翊丞[1,2], 蒙 彤[1,2]

[1]西安邮电大学网络空间安全学院 西安 中国 710121
[2]西安邮电大学无线网络安全技术国家工程实验室 西安 中国 710121

摘要 代理重签名在保证委托双方私钥安全的前提下, 通过半可信代理实现了双方签名的转换, 在本文方案中, 通过代理重签名实现了在通信过程中终端用户对于身份的隐私要求。移动医疗服务系统因为其有限的计算和存储能力, 需要借助云服务器来对医疗数据进行计算和存储。然而, 在将医疗数据外包给云服务器后, 数据便脱离了用户的控制, 这给用户隐私带来了极大地安全隐患。现有的无证书代理重签名方案大多都不具有撤销功能, 存在着密钥泄露等安全性问题。为了解决这一问题, 本文提出了一种可撤销的无证书代理重签名方案, 在不相互信任的移动医疗服务系统中, 实现了医疗数据传输过程以及云存储过程中的用户匿名性, 同时, 本文方案具有单向性和非交互性, 更适合在大规模的移动医疗系统中使用。此外, 当用户私钥泄露时, 本文利用 KUNode 算法实现了对用户的高效撤销, 并利用移动边缘计算技术将更新密钥和撤销列表的管理外包给移动边缘计算设备, 降低了第三方的计算成本, 使其具有较低的延迟。最后, 在随机谕言机模型下证明了所构造的方案在自适应选择消息攻击下的不可伪造性, 并利用 JPBC 库与其他方案进行计算与通信开销的对比。其结果表明, 本方案在具备更优越的功能的同时, 具有较小的计算成本、通信成本和撤销成本。

关键词 无证书代理重签名; 随机谕言机模型; 外包撤销; 移动医疗系统; 云计算
中图法分类号 TP309 DOI 号 10.19363/J.cnki.cn10-1380/tn.2024.05.01

Revocable Certificateless Proxy Re-signature Scheme in Mobile Healthcare System

GUO Rui[1,2], LIU Yingfei[1,2], WANG Yicheng[1,2], MENG Tong[1,2]

[1] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[2] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Abstract Proxy re-signature is a semi-trusted proxy that converts the signatures of both parties on the premise of ensuring the security of the private keys of the entrusting parties. In this scheme, proxy re-signing realizes the privacy requirements of the terminal users for identity in the communication process. Due to its limited computing and storage capacity, mobile healthcare service system needs to use cloud server to calculate and store healthy data. However, after outsourcing healthy data to cloud servers, the data will be out of users' control, causing great security risks to users' privacy. Most of the existing certificateless proxy re-signature schemes do not have the revocation function and have security problems such as key leakage. In order to solve this problem, a revocable unidirectional certificateless proxy re-signature scheme was proposed. In a mobile healthcare service system without mutual trust, this scheme realizes user anonymity in the process of healthy data transmission and cloud storage. At the same time, the scheme in this paper is unidirectional and non-interactive, which is more suitable for large-scale mobile healthcare service system. In addition, when the user's private key is leaked, this paper uses KUNode algorithm to realize the efficient revocation of the user, and uses mobile edge computing technology to outsource the management of updating the key and revocation list to mobile edge computing equipment, which reduces the computing cost of the third party and makes it have a lower delay. Finally, the proposed scheme was proved to be existentially unforgeable against chosen-message attacks on a random oracle model, and the computational and communication costs were compared with other schemes using JPBC library. The result shows that the scheme has better function and less computation cost, communication cost and revocation cost.

Key words certificateless proxy re-signature; random oracle model; outsourcing revocation; mobile healthcare system; cloud computing

通讯作者: 刘颖菲, 硕士, Email: hellopanshang@163.com。
本课题得到国家自然科学基金资助项目(No. 62072369, No. 62072371)、陕西省重点研发计划基金资助项目(No. 2020ZDLGY08-04)、陕西省创新能力支持计划基金资助项目(No. 2020KJXX-052)、陕西高校青年创新团队的资助。
收稿日期: 2022-07-07; 修改日期: 2022-10-03; 定稿日期: 2024-01-12

1 引言

随着移动智能终端以及传感器技术的日益普及, 移动医疗服务(Mobile Healthcare Service, MHS)系统的地位显著提高, 成为了智慧电子医疗中的一个重要分支[1]。相较于传统医疗系统, 它可以随时随地为患者提供医疗服务, 主要涉及到远程数据采集、远程监控、跟踪诊断治疗以及医务人员之间的沟通等方面[2]。MHS 系统可以保障医疗数据的实时传输以及大量医疗数据的存储, 具有广阔的应用前景。但其发展面临着很大的挑战, 比如移动设备的计算和存储能力有限、如何实现数据共享和管理等。

云计算以其海量的存储容量和强大的数据处理能力成为智慧医疗环境下的主要助推器[3]。在基于云的存储架构中, 医疗数据需要上传到集中的云中心, 当需要数据时再返回给医疗中心, 以此降低本地医疗信息的管理成本。然而, 当数十亿终端设备连接在一起时, 这种集中式数据存储架构已经变得不切实际, 因为由于网络拥塞而引起的延迟问题将非常严重, 尤其是对时间敏感的应用程序[4]。移动边缘计算(Mobile Edge Computing, MEC)是位于云和终端设备之间的、更接近数据源的一种概念, 它的提出很好地解决了这个问题[5]。

在云计算时代, 一方面, 云计算技术带给人们便捷的同时, 也具有一定的安全风险。一些具有较高隐私要求的患者数据被上传至云服务器后, 一旦泄露, 可能会暴露患者隐私; 另一方面, 云服务器由医疗中心租用, 医疗数据通常不会是患者直接提交给云服务器, 而是通过先向医疗中心提交数据, 再由医疗中心提交给云服务器。此外, 当系统中的用户权限到期或发生了私钥暴露的问题时, 应当对该用户及时进行撤销。

代理重签名方案可以为用户对于同一消息进行签名转换, 其通过半可信代理, 可以实现将被委托人的签名转换为委托人的签名的功能, 且代理不能从这一过程中得知两方的私钥, 根据代理重签名方案具有双向性或者单向性, 代理可以代替这两方中的任何一方或者单独一方进行签名。如果代理只能将这两个签名进行单向转化, 就称该方案为单向的[6], 如果委托人可以用自己的私钥和被委托人的公钥创建重签名密钥, 即委托人不参与委托过程, 就称该方案为非交互性的[6]。因此, 针对智慧医疗中用户隐私暴露、用户权限到期或密钥暴露等问题, 本文提出一种可撤销的无证书代理重签名方案, 具体工作如下:

1) 本方案具有单向性和非交互性, 在医疗中心(Healthcare Authority, HA)生成重签名密钥时不需要终端用户(Terminal User, TU)的参与, 同时也不要求这两方具有信任关系, 增加了系统的执行效率以及安全性;

2) 本方案将 TU 的签名转化为 HA 的签名, 最后将数据上传云端, 实现了患者身份在云端的匿名性;

3) 本方案将用户密钥更新以及撤销列表的维护外包给移动边缘计算设备(Mobile Edge Computing Device, MECD), 这使得密钥生成中心(Key Generation Centrer, KGC)不需要持续在线, 减轻了 KGC 的计算负担, 同时结合 KUNode 算法实现了用户的高效撤销。

4) 本方案基于拓展计算 Diffie-Hellman 问题证明了对于四类敌手在自适应选择消息攻击下是存在不可伪造性的。此外, 本文方案使用了 JPBC 密码库对方案的运行效率进行了仿真, 在满足前向安全性的同时, 需要较少的总计算开销, 具有更高的性能。同时, 在与其他撤销方法的更新密钥效率的对比中, 本方案具有更高的撤销效率。

2 相关工作

2.1 无证书代理重签名

Blaze 等人[7]于 1998 年首次提出了代理重签名的概念, 它允许半可信代理通过重签名密钥将被委托者的签名转换为同一消息的委托者的签名。Ateniese 和 Hohenberger[6]于 2005 年首次总结了关于代理重签名的属性并且对于其安全模型进行了形式化定义, 同时指出文献[7]中存在的安全问题, 第一个问题是攻击者在获得有效的签名或是重签名后可以从中恢复重签名密钥, 另一个问题是被委托者和代理可以串通, 这样会暴露委托者的密钥。此外, 他们还将构建单向多用途代理重签名方案作为开放问题。随后, Libert 和 Vergnaud[8]在随机谕言机模型下设计了第一个单向的多用途代理重签名方案, 解决了文献[6]中存在的问题。

但上述方案是基于证书的代理重签名方案, 其委托者和被委托者的公钥在验证签名之前需要经过证书的认证。为了减轻证书分发和管理的成本, Shao 等人[9]构建了第一个基于身份的代理重签名方案, 该方案以用户的身份信息作为公钥, 避免了对于证书的依赖。随后, 为满足实际需要, 各种基于身份的代理重签名方案先后提出[10-12]。然而, 因为基于身份的代理重签名方案中委托人和被委托人的私钥都是由私钥生成中心(Private Key Generator, PKG)生成, 从而导致基于身份的代理重签名方案无法避免密钥

托管问题[13], 即 PKG 知道任意用户的私钥, 就可以窃听用户的通信或者伪造用户的签名。

为了解决以上问题, 无证书密码学中自然出现了关于代理重签名的研究[14-16]。Guo 等人[17]将无证书密码学与代理重签名相结合, 提出了第一个具有双向性的无证书代理重签名方案, 不仅摆脱了对证书的依赖, 也避免了密钥托管问题, 但是没有给出具体的安全性证明。之后, Xiao 等人[18]提出了一种多用途的双向无证书代理重签名方案, 但也存在一些安全问题[19]。Chen 等人[20]提出了一种双向无证书代理重签名方案, 这些具有双向属性的无证书代理重签名方案要求委托者和被委托者保持信任关系, 这导致双向无证书代理重签名方案在实际的医疗环境中适用范围十分狭窄。

由于代理重签名具有良好的前景, 现在已被应用在各个领域。为了减少物联网移动支付过程中智能设备的资源消耗, Chen 等人[21]基于单向的无证书代理重签名设计了一种适用于移动设备的轻量级认证协议。Rabaninejad 等人[22]通过使用代理重签名, 提出了一种基于云环境下的公共共享数据审计协议。Xiong 等人[23]提出了一种使用代理重签名的 IIOT 异构系统的隐私保护认证协议。Fan 等人[24]提出了一种 NDN 网络下的数据传输协议, 利用代理重签名确保生产者的匿名性。随后, Xiong 等人[25]提出了一种使用代理重签名的 CDN 下的匿名身份验证协议。

2.2 可撤销的代理重签名

在实际应用中, 存在用户密钥被泄露或者是用户授权到期等问题。2014 年, Shen 等人[26]基于无证书公钥密码体制提出了一种新的撤销方法, 即 KGC 为用户生成部分私钥和时间密钥, 并定期更新未撤销用户的时间密钥。一旦需要撤销某一用户, KGC 可以停止更新用户的时间密钥。在整个方案中, KGC 需要持续在线, 更容易受到攻击, 并且如果系统用户数量巨大, 所面临的计算成本也会随之增大。Yang 等人[27]在 2018 年引入了第一个可撤销的双向和多用途的基于身份的代理重签名方案, 方案将私钥生成中心 PKG 的主密钥分为两部分, 其中一部分在服务器辅助下用于更新用户密钥, 但是由于缺乏单向性和一次性使用的特性, 使得它不适合移动医疗服务系统的环境。

3 基础知识

3.1 双线性映射

令 G_1 和 G_2 来表示两个 q 阶的循环加法群, P 表示 G_1 的一个生成元。如果 $\hat{e}: G_1 \times G_1 \to G_2$ 是一个双线性映射, 它应该满足以下条件:

1) 双线性性: 对于任意 $P,Q \in G_1$, $a,b \in Z_q^*$, 有 $\hat{e}(aP,bQ) = \hat{e}(P,Q)^{ab}$;

2) 非退化性: 存在 $P \in G_1$, 使得 $\hat{e}(P,P) \neq 1_{G_2}$;

3) 可计算性: 对任意 $P,Q \in G_1$, 存在有效的多项式时间算法可以计算出 $\hat{e}(P,Q)$。

3.2 困难假设

本文方案的安全性依赖于 eCDH 困难假设[13], 其定义如下:

eCDH 问题 (Extended Computational Diffie-Hellman Problem): 设 G 是阶为素数 q 的群, 且 P 为 G 的生成元。随机选取 $a,b \in Z_q^*$, 给定元组 $(P,aP,bP) \in G$, 则在概率多项式时间内计算出 $(Q,abQ) \in G \times G$ 是困难的, 其中 $Q \neq 1_G$。如果有 $\Pr[\mathcal{A}(P,aP,bP) = (Q,abQ)] \geqslant \varepsilon$, 则存在算法 $\mathcal{A}$ 以概率 ε 成功解决 G 上的 eCDH 问题, 其中概率的计算基于 $a,b \in Z_q^*$ 的随机选取以及算法 $\mathcal{A}$ 的随机选取。

定义 1. eCDH 假设: 如果任意多项式时间算法成功解决 G 上的 eCDH 问题的概率 ε 是可忽略的, 则称 G 上的 eCDH 问题是困难的。

3.3 KUNode 算法

在本文方案中使用了 KUNode 算法[28]来提高用户撤销的效率, 该算法采用了二叉树结构来管理系统用户。

KUNode 算法将二叉树 $\mathcal{BT}$, 撤销列表 RL 和时间周期 t 作为输入, 输出一个包含所有未被撤销节点的最小集合 Y。如果 θ 不是一个叶子节点, 其左孩子节点和右孩子节点分别表示为 θ_l 和 θ_r, 当用户被注册给一个叶子节点 η, 那么 $Path(\eta)$ 表示从根节点 root 到叶子节点 η 的路径上的所有节点的集合。如果被注册给节点 η_i 的用户在时间周期 t_i 被撤销, 那么表示为元组 $(\eta_i,t_i) \in RL$。

例如, 如图 1 所示, 假设 4 个用户被注册给 4 个叶子节点 η_1, η_2, η_3 和 η_4, 如果在时间周期 t 对于 η_2 的用户进行撤销, 则通过 KUNode 算法得到节点集合 X,Y, 其包含的节点如图 1 所示, 此时算法输出 Y, 那么 KGC 就只需更新 Y, 即所有未被撤销节点的最小集合中包含节点的密钥, 而不需要对所有用户节点进行更新。详细算法以伪代码的形式给出。

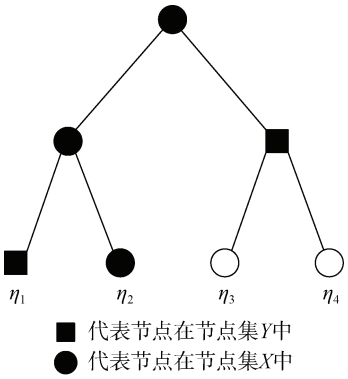


图 1　KUNode 算法的一个实例

Figure 1　An instance of the KUNode algorithm

Algorithm 1 ***KUNode***

Input: $\mathcal{BT}$, RL, t

Output: Y

1: X, $Y \leftarrow \varnothing$
2: for all $(\eta_i, t_i) \in RL$ do
3:　　if $t_i \leqslant t$ then
4:　　Add Path(η_i) to X
5:　　end if
6: end for
7: for all $\theta \in X$ do
8:　　if $\theta_l \notin X$ then
9:　　Add θ_l to Y
10:　　end if
11:　　if $\theta_r \notin X$ then
12:　　Add θ_r to Y
13:　　end if
14: end for
15: if $Y=\varnothing$ then
16:　　Add the root node to Y
17: end if
18: return Y

4　方案模型

4.1　系统模型

系统模型如图 2 所示, 主要由 5 个实体组成, 分别是密钥生成中心(Key Generation Centrer, KGC), 移动边缘计算设备(Mobile Edge Computing Device, MECD), 医疗中心(Healthcare Authority, HA), 终端用户(Terminal User, TU)以及云服务器(Cloud Server, CS), 具体功能如下:

密钥生成中心 KGC: KGC 负责为 TU 和 HA 进行注册, 一般来说, KGC 是一个半可信的实体, 由商业组织担任。它生成系统参数、系统主密钥和时间主密钥。此外, KGC 将时间主密钥秘密发送给 MECD, 并使用系统主密钥生成所有其他实体的部分私钥。

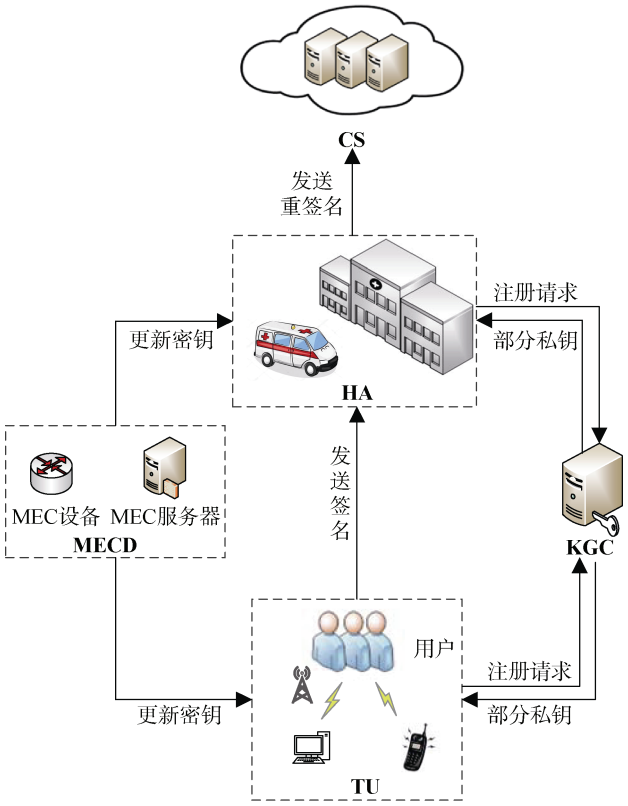


图 2　系统模型

Figure 2　The system model

移动边缘计算设备 MECD: MECD 主要由移动边缘设备、移动边缘服务器和边缘交换机组成, 具备一定的计算和存储能力, 同时相较于集中的云中心还具备较低的延迟性。本文假设 MECD 是诚实但好奇的, MECD 主要负责为所有未撤销的用户生成更新密钥以及维护撤销列表。由于为未撤销用户生成更新密钥时的大部分计算任务是由 MECD 来执行。因此, 大大减少了 KGC 的计算负担。

医疗中心 HA: HA 负责提供重签名服务, 即 HA 将 TU 的签名转换为 HA 的签名, 从而保护用户的个人信息。HA 是一个可信的实体, 而 CS 存储经过 HA 认证的数据。HA 在提供服务之前, 还需要向 KGC 注册以获得系统参数和私钥。

终端用户 TU: TU 的存储、计算数据的能力有限, 需要在注册阶段向 KGC 请求获取部分私钥。TU 要将个人医疗数据, 如电子病历, 进行签名, 并将其发送至 HA。出于安全考虑, 如果 KGC 检测到用户的不当行为或该用户声明其私钥被泄露, 则必须从系统中撤销 TU。

云服务器 CS: CS 具有强大的计算和存储能力。在验证 HA 传输数据的签名有效性后, CS 存储这些医疗数据。当用户需要数据时, CS 将数据和签名再发

送给用户。

本方案的实施步骤大致如下。首先, KGC 进行系统初始化, KGC 将时间主密钥发送给 MECD, 之后 KGC 不必一直在线, 密钥更新操作就由 MECD 完成; 然后, TU 和 HA 通过移动终端向 KGC 发送注册请求, KGC 为其生成部分私钥并发送; 接着, MECD 为未撤销用户生成更新密钥并通过公开信道发送给用户, 可以确保一些权限过期或权限被撤销的用户无法再继续使用服务; 最后, TU 利用前面步骤得到的私钥, 通过个人移动终端对想要提交的医疗数据进行签名, 并将第一级签名发送给 HA, HA 再对其进行重签名, 生成第二级签名, 和医疗数据一起上传云端, 由云服务器验证、保存数据, 这样云服务器就无法分辨数据最初发送者的身份。

在本文系统模型中, 最关键的过程是将终端用户 TU 的签名转换为医疗中心 HA 的签名, 即代理重签名的步骤。与代理重签名的结合, 满足了移动医疗系统中用户对个人医疗数据在云端的隐私要求, 隐藏了 TU 的身份, 并让云服务器相信数据来自 HA。同时, 即使云服务器的数据泄露, 这些医疗数据的最初发送方——TU 的身份也不会被攻击者得知。为了实现这一功能, 本文提出了一种可撤销的无证书代理重签名方案。

4.2 可撤销的无证书代理重签名方案模型

本文提出的可撤销的无证书代理重签名方案由下面 11 个算法组成, 如下所示:

1) **系统初始化算法 SetUp**(λ, T, N): 该算法由 KGC 运行, 输入安全参数 λ、系统最大用户数 N 和时间总周期 T。输出公开参数 *params*、系统主密钥 *msk*、时间主密钥 *mtk*、初始撤销列表 *RL*、状态信息 *st*;

2) **部分私钥提取算法 ExtPartPriKey**$(params, msk, ID)$: 该算法由 KGC 运行, 输入 *params*、*msk* 和用户身份 *ID*。输出用户部分私钥 D_{ID};

3) **密钥更新算法 KeyUp**$(params, mtk, ID, t, RL, st)$: 该算法由 MECD 运行, 输入 *params*、*mtk*、身份 *ID*、更新的时间周期 t、当前的用户撤销列表 *RL* 和状态信息 *st*。若 $t > T$, 输出“⊥”; 否则, 输出更新密钥 $UK_{ID,t}$;

4) **秘密值提取算法 ExtSecValue**(ID): 该算法由用户运行, 输入身份 *ID*。输出秘密值 r_{ID};

5) **密钥生成算法 KeyGen**$(params, ID, r_{ID}, D_{ID}, UK_{ID,t})$: 该算法由用户运行, 输入 *params*、*ID*、秘密值 r_{ID}、部分私钥 D_{ID} 和更新密钥 $UK_{ID,t}$。若用户在时间周期 t 已被撤销, 输出“⊥”; 否则, 输出用户公钥 PK_{ID} 和关于 t 的用户私钥 $SK_{ID,t}$;

6) **签名算法 Sign**$(params, t, M, SK_{A,t})$: 该算法由被委托者 A 运行, 输入 *params*、时间周期 t、消息 M 和被委托者的私钥 $SK_{A,t}$。输出被委托者关于消息 M 的签名 σ_A;

7) **签名验证算法 SignVer**$(params, M, t, ID_A, \sigma_A)$: 该算法由验证者运行, 输入 *params*、时间周期 t、消息 M、身份 ID_A 和被委托者 A 的签名 σ_A。若 σ_A 是合法的, 输出 1; 否则, 输出 0;

8) **重签名密钥生成算法 ReSignKeyGen**$(params, ID_A, SK_{B,t})$: 该算法由委托者 B 运行, 输入 *params*、被委托者身份 ID_A 与委托者的私钥 $SK_{B,t}$。输出关于时间周期 t 的重签名密钥 RK_t;

9) **重签名算法 ReSign**$(params, t, M, ID_A, RK_t, \sigma_A)$: 该算法由委托者 B 运行, 输入 *params*、时间周期 t、消息 M、被委托者的身份 ID_A、关于时间周期 t 的重签名密钥 RK_t 和签名 σ_A。如果运行签名验证算法验证 σ_A 合法, 输出关于委托者 ID_B 和 t 的关于消息 M 的签名 σ_B; 否则, 输出“⊥”;

10) **重签名验证算法 ReSignVer**$(params, M, t, ID_B, \sigma_B)$: 该算法由验证者运行, 输入 *params*、时间周期 t、消息 M、身份 ID_B 和委托者 B 的签名 σ_B。若 σ_B 是合法的, 输出 1; 否则, 输出 0;

11) **用户撤销算法 Revoke**(ID, t, RL, st): 该算法由 MECD 运行, 输入时间周期 t、撤销的用户身份 *ID*、撤销列表 *RL* 和状态信息 *st*。输出更新后的撤销列表 RL'。

4.3 安全模型

本文方案涉及到的敌手分为以下 4 类。

1) 类型Ⅰ: 这类敌手模仿不诚实的用户, $\mathcal{A}_1$ 不能得到系统主密钥或者用户部分私钥, 但可以替换任意合法用户的公钥。

2) 类型Ⅱ: 这类敌手可以看作是诚实但好奇的 KGC, $\mathcal{A}_2$ 可以得到系统主密钥和用户的部分私钥, 但无法获取用户秘密值, 也无法替换用户公钥。

3) 类型Ⅲ: 这类敌手模仿被恶意撤销的用户, 假设将 $\mathcal{A}_3$ 看作身份为 ID' 在时间周期 t' 被撤销的用户。$\mathcal{A}_3$ 可以获得用户 ID' 的部分私钥和秘密值, 但在

t' 之后无法获得用户 ID' 的更新密钥。

4) 类型Ⅳ: 这类敌手可以看作是诚实但好奇的 MECD, $\mathcal{A}_4$ 可以得到时间主密钥和任意用户的更新密钥, 假设 $\mathcal{A}_4$ 可以获得除目标用户之外的其他用户的部分私钥和秘密值。

本文方案的安全模型由挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ 和 $\mathcal{A}_4$ 之间进行交互的安全游戏给出, 具体如下:

游戏 1

初始化 挑战者 $\mathcal{C}$ 运行 $\text{SetUp}(\lambda,T,N)$ 算法并获取主公钥 mpk、系统主密钥 msk 和时间主密钥 mtk, 然后发送生成的主公钥和系统参数 $params$ 给 $\mathcal{A}_1$, 自己秘密保存系统主密钥 msk 和时间主密钥 mtk。

询问 $\mathcal{A}_1$ 可以自适应地询问以下谕言机:

1) 哈希询问: $\mathcal{A}_1$ 输入任意数据, $\mathcal{C}$ 将对应的哈希询问输出的哈希值返回给 $\mathcal{A}_1$。

2) 用户生成询问: 对于 $\mathcal{A}_1$ 请求的关于身份 ID 的用户生成询问, $\mathcal{C}$ 运行算法 $\text{ExtPartPriKey}(params,msk,ID)$ 产生用户部分私钥 D_{ID}, 运行算法 $\text{ExtSecValue}(ID)$ 产生秘密值 r_{ID}。然后计算用户公钥 PK_{ID}, $\mathcal{C}$ 将 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID})$ 存入列表 L_{PK}, 并将 PK_{ID} 返回给 $\mathcal{A}_1$。

3) 部分私钥提取询问: 对于 $\mathcal{A}_1$ 请求的关于身份 ID 的部分私钥提取询问, $\mathcal{C}$ 先在列表 L_{PK} 中搜索, 如果列表中存在 $(ID,D_{ID},r_{ID},PK_{ID})$, 输出用户部分私钥 D_{ID}; 否则, 输出 $\perp$。最后 $\mathcal{C}$ 将输出结果发送给 $\mathcal{A}_1$。

4) 密钥更新询问: 对于 $\mathcal{A}_1$ 请求的关于时间周期 t 的密钥更新询问, 其中 t 不能小于以前所有询问过的时间周期, $\mathcal{C}$ 运行算法 $\text{KeyUp}(params,mtk,ID,t,RL,st)$, 将输出的更新密钥 $UK_{ID,t}$ 发送给 $\mathcal{A}_1$。

5) 秘密值提取询问: 对于 $\mathcal{A}_1$ 请求的关于 ID 的秘密值提取询问, $\mathcal{C}$ 在列表 L_{PK} 中搜索, 如果列表中存在 $(ID,D_{ID},r_{ID},PK_{ID})$, 输出秘密值 r_{ID}, 否则, 输出 $\perp$。最后 $\mathcal{C}$ 将输出的秘密值 r_{ID} 发送给 $\mathcal{A}_1$。

6) 公钥替换询问: 对于 $\mathcal{A}_1$ 请求的关于 ID 的公钥 PK_{ID} 替换为 PK_{ID}^* 的询问, $\mathcal{C}$ 将列表 L_{PK} 中的 PK_{ID} 替换为 PK_{ID}^*。

7) 签名询问: 对于 $\mathcal{A}_1$ 请求的关于 (ID,t,M) 的签名询问, $\mathcal{C}$ 运行算法 $\text{Sign}(params,t,M,SK_{ID,t})$, 将生成的关于消息 M 的签名 σ_{ID} 返回给 $\mathcal{A}_1$。

8) 重签名密钥生成询问: 对于 $\mathcal{A}_1$ 请求的关于两个用户的身份 (ID_A,ID_B) 和时间周期 t 的重签名密钥询问, $\mathcal{C}$ 通过算法 $\text{ReSignKeyGen}(params,ID_A,SK_{B,t})$, 将生成的重签名密钥 RK_t 发送给 $\mathcal{A}_1$。

9) 重签名询问: 对于 $\mathcal{A}_1$ 请求的关于两个身份 (ID_A,ID_B,t) 的重签名询问, $\mathcal{C}$ 运行算法 $\text{ReSign}(params,t,M,ID_A,RK_t,\sigma_A)$, 并将生成的重签名 σ_B 返回给 $\mathcal{A}_1$。

10) 撤销询问: 对于 $\mathcal{A}_1$ 请求的关于 (ID,t) 的撤销询问, $\mathcal{C}$ 运行算法 $\text{Revoke}(ID,t,RL,st)$, 并将输出的用户撤销列表 RL' 返回给 $\mathcal{A}_1$。

伪造 敌手 $\mathcal{A}_1$ 输出身份 ID^*、时间周期 t^*、公钥 PK_{ID^*}、消息 M^* 和签名 σ_{ID^*}。若以下 4 个条件均成立, 则称 $\mathcal{A}_1$ 在以上游戏中获胜。

1) $\text{ReSignVer}(params,M^*,t^*,ID^*,\sigma_{ID^*})=1$。

2) ID^* 未进行过部分私钥提取询问。

3) (ID^*,t^*,M^*) 未进行过签名询问。

4) (ID^*,t^*,M^*,Δ) 未进行过重签名询问, 其中, Δ 表示任意签名。

游戏 2

初始化 挑战者 $\mathcal{C}$ 以与游戏 1 相同的方式运行 SetUp 算法, 并将系统主密钥 msk 和系统参数 $params$ 发送给 $\mathcal{A}_2$。

询问 $\mathcal{A}_2$ 可以自适应地询问与游戏 1 相同的谕言机: 哈希询问、用户生成询问、密钥更新询问、秘密值提取询问、签名询问、重签名密钥生成询问、重签名询问、撤销询问。

伪造 敌手 $\mathcal{A}_2$ 输出身份 ID^*、时间周期 t^*、公钥 PK_{ID^*}、消息 M^* 和签名 σ_{ID^*}。若以下 4 个条件均成立, 则称 $\mathcal{A}_2$ 在以上游戏中获胜。

1) $\text{ReSignVer}(params,M^*,t^*,ID^*,\sigma_{ID^*})=1$。

2) ID^* 未进行过秘密值提取询问。

3) (ID^*,t^*,M^*) 未进行过签名询问。

4) (ID^*,t^*,M^*,Δ) 未进行过重签名询问, 其中, Δ 表示任意签名。

游戏 3

初始化 挑战者 $\mathcal{C}$ 以与游戏 1 相同的方式运行 SetUp 算法, 并将系统参数 *params* 发送给 $\mathcal{A}_3$。

询问 $\mathcal{A}_3$ 可以自适应地询问与游戏 1 相同的谕言机: 哈希询问、用户生成询问、部分私钥提取询问、密钥更新询问、秘密值提取询问、公钥替换询问、签名询问、重签名密钥生成询问、重签名询问、撤销询问。

伪造 敌手 $\mathcal{A}_3$ 输出身份 ID^*、时间周期 t^*、公钥 PK_{ID^*}、消息 M^* 和签名 σ_{ID^*}。若以下 4 个条件均成立, 则称 $\mathcal{A}_3$ 在以上游戏中获胜。

1) $\mathrm{ReSignVer}\left(params, M^*, t^*, ID^*, \sigma_{ID^*}\right)=1$。

2) $\left(ID^*, t^*\right)$ 未进行过密钥更新询问。

3) $\left(ID^*, t^*, M^*\right)$ 未进行过签名询问。

4) $\left(ID^*, t^*, M^*, \Delta\right)$ 未进行过重签名询问, 其中, Δ 表示任意签名。

游戏 4

初始化 挑战者 $\mathcal{C}$ 以与游戏 1 相同的方式运行 SetUp 算法, 并将系统参数 *params* 发送给 $\mathcal{A}_4$。

询问 $\mathcal{A}_4$ 可以自适应地询问与游戏 1 相同的谕言机: 哈希询问、用户生成询问、部分私钥提取询问、秘密值提取询问、签名询问、重签名密钥生成询问、重签名询问、撤销询问。

伪造 敌手 $\mathcal{A}_4$ 输出身份 ID^*、时间周期 t^*、公钥 PK_{ID^*}、消息 M^* 和签名 σ_{ID^*}。若以下 4 个条件均成立, 则称 $\mathcal{A}_4$ 在以上游戏中获胜。

1) $\mathrm{ReSignVer}\left(params, M^*, t^*, ID^*, \sigma_{ID^*}\right)=1$。

2) ID^* 未进行过部分私钥提取询问和秘密值提取询问。

3) $\left(ID^*, t^*, M^*\right)$ 未进行过签名询问。

4) $\left(ID^*, t^*, M^*, \Delta\right)$ 未进行过重签名询问, 其中, Δ 表示任意签名。

定义 2. 如果对于任何一个概率多项式时间敌手 $\mathcal{A}_1$ (敌手 $\mathcal{A}_2$ /敌手 $\mathcal{A}_3$ /敌手 $\mathcal{A}_4$), 在多项式时间内在游戏 1(游戏 2/游戏 3/游戏 4)中获胜的概率可以忽略不计, 则本文提出的方案在自适应选择消息攻击下是不可伪造的。

在以上询问阶段中, 敌手可以自适应地询问上述谕言机, 但整个过程中必须遵守以下限制条件:

1) 密钥更新询问和撤销询问在时间周期 t 上进行查询, 其中 t 不能大于 T, 并且 t 不能小于以前所有询问过的时间周期;

2) 在时间周期 t 时进行密钥更新询问, 则在 t 之前的时间周期内不能进行秘密值提取询问。

3) 敌手未进行关于 t 的密钥更新询问前, 不能发起关于 t 的秘密值提取询问。

4) 如果敌手在时间周期 t 发起过关于 *ID* 的密钥更新询问, 则敌手在时间周期 t 不能发起关于 *ID* 的撤销询问。

5 可撤销的无证书代理重签名方案构造

5.1 系统初始化阶段

输入安全参数 k、素数 q、最大系统用户数量 N 和最大系统时间周期 T, KGC 执行以下步骤。

1) 任意选取两个 q 阶循环加法群 G_1 和 G_2, P 为 G_1 生成元, $e: G_1 \times G_1 \to G_2$ 为双线性对。

2) 随机选取 $s, c \in Z_q^*$, 定义系统主密钥 $msk = s$, 时间主密钥 $mtk = c$, 并计算系统主公钥 $PK_{pub} = sP$, 时间主公钥 $PK_t = cP$。

3) 选取 6 个安全的哈希函数为 $H_1: \{0,1\}^* \times G_1^2 \to Z_q^*$, $H_2: \{0,1\}^* \to Z_q^*$, $H_3: \{0,1\}^* \times G_1 \times \{0,1\}^* \times G_1 \to Z_q^*$, $H_4: \{0,1\}^* \times \{0,1\}^* \to Z_q^*$, $H_5: \{0,1\}^* \times G_1^3 \to Z_q^*$, $H_6: \{0,1\}^* \to G_1$。

4) 选择一棵有 N 个叶子节点的二叉树 $\mathcal{BT}$ 用来管理系统用户, 初始化撤销列表为 $RL = \varnothing$, 状态信息为 $st = \mathcal{BT}$。

5) 公开系统参数 $params = \{G_1, G_2, e, q, P, PK_{pub}, PK_t, H_1, H_2, H_3, H_4, H_5, H_6\}$, 秘密保存系统主密钥 *msk*, 将时间主密钥 *mtk* 通过一个安全信道发送给 MECD。

5.2 部分私钥提取阶段

在进行通信前, TU 和 HA 需要进行用户注册, TU 的注册过程需要先执行下列操作:

1) 当 KGC 收到来自 ID_{TU} 的注册请求后, KGC 随机在 $\mathcal{BT}$ 中选取一个未被注册的叶子节点 η, 并且将它注册给 TU。

2) 对于任意一个节点 $\theta \in Path(\eta)$, 随机选取 $w_\theta \in Z_q^*$, 计算 $W_{\mathrm{TU}} = w_\theta P$, $h_{1,\mathrm{TU}} = H_1(ID_{\mathrm{TU}}, W_{\mathrm{TU}}, PK_{pub})$, $h_{2,\mathrm{TU}} = H_2(ID_{\mathrm{TU}})$ 和 $d_\theta = w_\theta h_{2,\mathrm{TU}} + s h_{1,\mathrm{TU}}$。

然后 KGC 将部分私钥 $D_{\mathrm{TU}}=\{(\theta,d_\theta,W_{\mathrm{TU}})\}_{\theta\in Path(\eta)}$ 通过一个安全信道发送给 TU。

3) 当 TU 收到部分私钥 D_{TU} 后, 执行下列操作:

① 计算 $h_{1,\mathrm{TU}}=H_1(ID_{\mathrm{TU}},W_{\mathrm{TU}},PK_{pub})$ 和 $h_{2,\mathrm{TU}}=H_2(ID_{\mathrm{TU}})$。

② 通过等式 $e(d_\theta,P)=e(W_{\mathrm{TU}},h_{2,\mathrm{TU}})e(PK_{pub},h_{1,\mathrm{TU}})$ 是否成立来验证收到的部分私钥的合法性。若等式成立, 则部分私钥有效; 否则, 输出⊥。

HA 部分私钥提取过程与 TU 的过程相同, 生成 HA 的部分私钥为 D_{HA}。

5.3 密钥更新阶段

1) 当 MECD 收到来自 ID_{TU} 的密钥更新请求后, 对于时间周期 $t<T$、撤销列表 RL 和状态信息 st, 执行下列操作:

① 运行算法 $KUNode(\mathcal{BT},RL,t)$ 得到节点集 Y。

② 对于任意节点 $\theta\in Y$, 随机选取 $v_\theta\in Z_q^*$, 计算 $V_{\mathrm{TU}}=v_\theta P$, $h_{3,\mathrm{TU}}=H_3(ID_{\mathrm{TU}},V_{\mathrm{TU}},t,PK_t)$, $h_{4,\mathrm{TU}}=H_4(ID_{\mathrm{TU}},t)$ 和 $t_\theta=v_\theta h_{4,\mathrm{TU}}+ch_{3,\mathrm{TU}}$。然后 MECD 将更新密钥 $UK_{\mathrm{TU},t}=\{(\theta,t_\theta,V_{\mathrm{TU}})\}_{\theta\in Y}$ 通过一个公开信道发送给 TU。

2) 当 TU 收到更新密钥 $UK_{\mathrm{TU},t}$ 后, 执行下列操作:

① 计算 $h_{3,\mathrm{TU}}=H_3(ID_{\mathrm{TU}},V_{\mathrm{TU}},t,PK_t)$ 和 $h_{4,\mathrm{TU}}=H_4(ID_{\mathrm{TU}},t)$。

② 通过等式 $e(t_\theta,P)=e(V_{\mathrm{TU}},h_{4,\mathrm{TU}})e(PK_t,h_{3,\mathrm{TU}})$ 是否成立来验证收到的更新密钥的合法性。若等式成立, 则部分私钥有效; 否则, 输出⊥。

HA 部分密钥更新过程与 TU 的过程相同, 生成 HA 的更新密钥为 $UK_{\mathrm{HA},t}$。

5.4 秘密值提取阶段

TU 随机选择 $r_{\mathrm{TU}}\in Z_q^*$ 作为秘密值, HA 随机选择 $r_{\mathrm{HA}}\in Z_q^*$ 作为秘密值。

5.5 密钥生成阶段

1) TU 计算公钥 $PK_{\mathrm{TU}}=r_{\mathrm{TU}}P$, 表示私钥为 $SK_{\mathrm{TU},t}=(sk_{\mathrm{TU},1},sk_{\mathrm{TU},2})=(d_\theta+t_\theta,r_{\mathrm{TU}})$。

2) HA 计算其公钥 $PK_{\mathrm{HA}}=r_{\mathrm{HA}}P$, 表示私钥为 $SK_{\mathrm{HA},t}$。

5.6 签名阶段

TU 拥有待签名消息 M, 给定 TU 私钥 $SK_{\mathrm{TU},t}=(sk_{\mathrm{TU},1},sk_{\mathrm{TU},2})=(d_\theta+t_\theta,r_{\mathrm{TU}})$, 公钥 $PK_{\mathrm{TU}}=r_{\mathrm{TU}}P$。计算 $h_{5,\mathrm{TU}}=H_5(ID_{\mathrm{TU}},PK_{\mathrm{TU}},W_{\mathrm{TU}},PK_{pub})$, 可以得到关于消息 M 的签名, 表示为:

$$\sigma_{\mathrm{TU},1}=H_6(M)(sk_{\mathrm{TU},1}+r_{\mathrm{TU}}h_{5,\mathrm{TU}}) \tag{1}$$

$$\sigma_{\mathrm{TU},2}=W_{\mathrm{TU}}h_{2,\mathrm{TU}}+V_{\mathrm{TU}}h_{4,\mathrm{TU}}+PK_{pub}h_{1,\mathrm{TU}}+PK_th_{3,\mathrm{TU}}+PK_{\mathrm{TU}}h_{5,\mathrm{TU}} \tag{2}$$

最后将得到的签名 $\sigma_{\mathrm{TU}}=(\sigma_{\mathrm{TU},1},\sigma_{\mathrm{TU},2})$ 发送给 HA。

5.7 签名验证阶段

当 HA 收到来自 ID_{TU} 关于消息 M 的签名 σ_{TU} 后, 先通过下式验证签名 σ_{TU} 的合法性:

$$e(\sigma_{\mathrm{TU},1},P)=e(\sigma_{\mathrm{TU},2},H_6(M)) \tag{3}$$

若验证通过, 继续进行下面的步骤; 否则, 输出⊥。

5.8 重签名密钥生成阶段

1) HA 计算 $h_{1,\mathrm{HA}}=H_1(ID_{\mathrm{HA}},W_{\mathrm{HA}},PK_{pub})$, $h_{2,\mathrm{HA}}=H_2(ID_{\mathrm{HA}})$, $h_{3,\mathrm{HA}}=H_3(ID_{\mathrm{HA}},V_{\mathrm{HA}},t,PK_t)$, $h_{4,\mathrm{HA}}=H_4(ID_{\mathrm{HA}},t)$, $h_{5,\mathrm{HA}}=H_5(ID_{\mathrm{HA}},PK_{\mathrm{HA}},W_{\mathrm{HA}},PK_{pub})$。

2) 对于重签名密钥进行计算:

$$rk_1=(sk_{\mathrm{HA},1}+r_{\mathrm{HA}}h_{5,\mathrm{HA}})^{-1}(W_{\mathrm{TU}}h_{2,\mathrm{TU}}+V_{\mathrm{TU}}h_{4,\mathrm{TU}}+PK_{pub}h_{1,\mathrm{TU}}+PK_th_{3,\mathrm{TU}}+PK_{\mathrm{TU}}h_{5,\mathrm{TU}})$$

$$rk_2=W_{\mathrm{HA}}h_{2,\mathrm{HA}}+V_{\mathrm{HA}}h_{4,\mathrm{HA}}+PK_{pub}h_{1,\mathrm{HA}}+PK_th_{3,\mathrm{HA}}+PK_{\mathrm{HA}}h_{5,\mathrm{HA}}$$

得到代理重签名密钥 $RK_t=(rk_1,rk_2)$。

5.9 重签名阶段

HA 随机选取 $y\in Z_q^*$, 计算 $x=(sk_{\mathrm{HA},1}+r_{\mathrm{HA}}h_{5,\mathrm{HA}})\cdot(sk_{\mathrm{TU},1}+r_{\mathrm{TU}}h_{5,\mathrm{TU}})^{-1}y$, 然后计算重签名如下:

$$\sigma_{\mathrm{HA},1}=\sigma_{\mathrm{TU},1}x=H_6(M)(sk_{\mathrm{TU},1}+r_{\mathrm{TU}}h_{5,\mathrm{TU}})x=H_6(M)(sk_{\mathrm{HA},1}+r_{\mathrm{HA}}h_{5,\mathrm{HA}})y$$

$$\sigma_{\mathrm{HA},2}=\sigma_{\mathrm{TU},2}x=(W_{\mathrm{TU}}h_{2,\mathrm{TU}}+V_{\mathrm{TU}}h_{4,\mathrm{TU}}+PK_{pub}h_{1,\mathrm{TU}}+PK_th_{3,\mathrm{TU}}+PK_{\mathrm{TU}}h_{5,\mathrm{TU}})x=(W_{\mathrm{HA}}h_{2,\mathrm{HA}}+V_{\mathrm{HA}}h_{4,\mathrm{HA}}+PK_{pub}h_{1,\mathrm{HA}}+PK_th_{3,\mathrm{HA}}+PK_{\mathrm{HA}}h_{5,\mathrm{HA}})y$$

$$\sigma_{\mathrm{HA},3}=rk_1x=Py$$

$$\sigma_{\mathrm{HA},4}=rk_2=W_{\mathrm{HA}}h_{2,\mathrm{HA}}+V_{\mathrm{HA}}h_{4,\mathrm{HA}}+PK_{pub}h_{1,\mathrm{HA}}+PK_th_{3,\mathrm{HA}}+PK_{\mathrm{HA}}h_{5,\mathrm{HA}}$$

将关于消息 M 的重签名 $\sigma_{\mathrm{HA}}=(\sigma_{\mathrm{HA},1},\sigma_{\mathrm{HA},2},\sigma_{\mathrm{HA},3},\sigma_{\mathrm{HA},4})$ 发送给 CS。

5.10 重签名验证阶段

当 CS 收到了关于消息 M 的重签名后, 通过下式验证重签名 σ_{HA} 的合法性。

$$e(\sigma_{\mathrm{HA},1},P)=e(\sigma_{\mathrm{HA},2},H_6(M)) \quad (4)$$

$$e(\sigma_{\mathrm{HA},2},P)=e(\sigma_{\mathrm{HA},4},\sigma_{\mathrm{HA},3}) \quad (5)$$

若以上等式成立, 则 σ_{HA} 有效, CS 将收到的数据存储在云端。

5.11 撤销阶段

当用户的服务到期或者用户私钥遭到泄露时, MECD 需要撤销此用户身份。输入一个在时间周期 t 被撤销的节点 η, 且节点 η 在 $\mathcal{BT}$ 中被用户注册过, 对于当前用户撤销列表 RL 和状态 st, MECD 更新撤销列表为 $RL'\leftarrow RL\cup\{(\eta,t)\}$ 并输出。

6 安全性分析

6.1 正确性分析

本文方案的正确性分析如下:

验证签名 $\sigma_{\mathrm{TU}}=(\sigma_{\mathrm{TU},1},\sigma_{\mathrm{TU},2})$:

$$\begin{aligned}e(\sigma_{\mathrm{TU},1},P)&=e(H_6(M)(sk_{\mathrm{TU},1}+r_{\mathrm{TU}}h_{5,\mathrm{TU}}),P)=\\&e(P(d_\theta+t_\theta+r_{\mathrm{TU}}h_{5,\mathrm{TU}}),H_6(M))=\\&e(P(w_\theta h_{2,\mathrm{TU}}+sh_{1,\mathrm{TU}}+v_\theta h_{4,\mathrm{TU}}+ch_{3,\mathrm{TU}}+\\&r_{\mathrm{TU}}h_{5,\mathrm{TU}}),H_6(M))=\\&e(W_{\mathrm{TU}}h_{2,\mathrm{TU}}+V_{\mathrm{TU}}h_{4,\mathrm{TU}}+PK_{pub}h_{1,\mathrm{TU}}+\\&PK_th_{3,\mathrm{TU}}+PK_{\mathrm{TU}}h_{5,\mathrm{TU}},H_6(M))=\\&e(\sigma_{\mathrm{TU},2},H_6(M))\end{aligned}$$

验证代理重签名 $\sigma_{\mathrm{HA}}=(\sigma_{\mathrm{HA},1},\sigma_{\mathrm{HA},2},\sigma_{\mathrm{HA},3},\sigma_{\mathrm{HA},4})$:

$$\begin{aligned}e(\sigma_{\mathrm{HA},1},P)&=e(H_6(M)(sk_{\mathrm{HA},1}+r_{\mathrm{HA}}h_{5,\mathrm{HA}})y,P)=\\&e((sk_{\mathrm{HA},1}+r_{\mathrm{HA}}h_{5,\mathrm{HA}})yP,H_6(M))=\\&e((w_\theta h_{2,\mathrm{HA}}+sh_{1,\mathrm{HA}}+v_\theta h_{4,\mathrm{HA}}+ch_{3,\mathrm{HA}}+\\&r_{\mathrm{HA}}h_{5,\mathrm{HA}})yP,H_6(M))=\\&e((W_{\mathrm{HA}}h_{2,\mathrm{HA}}+V_{\mathrm{HA}}h_{4,\mathrm{HA}}+PK_{pub}h_{1,\mathrm{HA}}+\\&PK_th_{3,\mathrm{HA}}+PK_{\mathrm{HA}}h_{5,\mathrm{HA}})y,H_6(M))=\\&e(\sigma_{\mathrm{HA},2},H_6(M))\end{aligned}$$

$$\begin{aligned}e(\sigma_{\mathrm{HA},2},P)&=e((W_{\mathrm{HA}}h_{2,\mathrm{HA}}+V_{\mathrm{HA}}h_{4,\mathrm{HA}}+PK_{pub}h_{1,\mathrm{HA}}+\\&PK_th_{3,\mathrm{HA}}+PK_{\mathrm{HA}}h_{5,\mathrm{HA}})y,P)=\\&e((W_{\mathrm{HA}}h_{2,\mathrm{HA}}+V_{\mathrm{HA}}h_{4,\mathrm{HA}}+PK_{pub}h_{1,\mathrm{HA}}+\\&PK_th_{3,\mathrm{HA}}+PK_{\mathrm{HA}}h_{5,\mathrm{HA}}),yP)=\\&e(\sigma_{\mathrm{HA},4},\sigma_{\mathrm{HA},3})\end{aligned}$$

6.2 安全性证明

通过敌手和挑战者之间的安全游戏, 给出可撤销的无证书代理重签名方案的安全性证明过程, 其中定理 1 的证明由以下四个引理的证明给出。

定理 1.在随机谕言机模型中, 本文方案在扩展计算 Diffie-Hellman(eCDH)假设下满足自适应选择消息攻击下的不可伪造性。

引理 1.在随机谕言机模型中, 假设存在敌手 $\mathcal{A}_{\mathrm{I}}$ 以不可忽略的概率 ε' 打破本文方案的不可伪造性, 那么存在算法 $\mathcal{C}$ 在多项式时间 t 内以不可忽略的概率 $\varepsilon\geqslant\frac{\varepsilon'}{q_1}(1-\frac{1}{q_1+q_2})^{q_e}$ 解决 eCDH 问题,其中 q_1, q_2, q_e 分别为 H_1 询问、H_2 询问和部分私钥询问的次数。

证明. 给定一个随机的 eCDH 问题实例 (P,aP,bP), 挑战者 $\mathcal{C}$ 通过与敌手 $\mathcal{A}_{\mathrm{I}}$ 交互, 目的是输出一个 eCDH 的解 (Q,abQ), 其中 $a,b\in{}_RZ_q^*$, $Q\in{}_RG$。在挑战者 $\mathcal{C}$ 与敌手 $\mathcal{A}_{\mathrm{I}}$ 之间建立游戏, 两者的交互过程具体如下所示:

初始化

挑战者 $\mathcal{C}$ 选取随机值 $c\in Z_q^*$, 计算 $PK_t=cP$, 设 $PK_{pub}=aP$, $\mathcal{C}$ 安全保存时间主密钥 $mtp=c$, 然后随机选取一个身份 ID_C 作为被挑战身份, 并发送 ID_C 和系统公共参数 $params=\{G_1,G_2,e,q,P,PK_{pub},PK_t,H_1,H_2,H_3,H_4,H_5,H_6\}$ 给敌手 $\mathcal{A}_{\mathrm{I}}$。

挑战者 $\mathcal{C}$ 保存有七个初始为空的列表 $<L_{PK},L_1,L_2,L_3,L_4,L_5,L_6>$。其中, L_{PK} 用来存放用户生成询问的记录, $<L_1,L_2,L_3,L_4,L_5,L_6>$ 分别存储对应六个哈希函数 $<H_1,H_2,H_3,H_4,H_5,H_6>$ 的询问。

询问

1) H_1 询问: 初始 L_1 为空, $\mathcal{C}$ 在收到敌手 $\mathcal{A}_{\mathrm{I}}$ 对于 (ID,W_{ID},PK_{pub}) 的询问后, 先在列表 L_1 中搜索 (ID,W_{ID},PK_{pub},h_1)。若元组存在, 则 $\mathcal{C}$ 返回 h_1 作为应答; 否则, $\mathcal{C}$ 随机选择 $h_1\in Z_q^*$, 计算 $W_{ID}=w_\theta P$, 将 h_1 返回给 $\mathcal{A}_{\mathrm{I}}$, 并将 (ID,W_{ID},PK_{pub},h_1) 记录到列表 L_1 中。

2) H_2 询问: 初始 L_2 为空, $\mathcal{C}$ 在收到敌手 $\mathcal{A}_{\mathrm{I}}$ 对于 ID 的询问后, 先在列表 L_2 中搜索 (ID,h_2)。若 (ID,h_2) 存在, 则 $\mathcal{C}$ 返回 h_2 作为应答; 否则, $\mathcal{C}$ 随机选择 $h_2\in Z_q^*$ 返回给 $\mathcal{A}_{\mathrm{I}}$, 并将 (ID,h_2) 记录到列表 L_2 中。

3) H_3 询问: 初始 L_3 为空, $\mathcal{C}$ 在收到敌手 $\mathcal{A}_{\mathrm{I}}$ 对于 (ID,V_{ID},t,PK_t) 的询问后, 先在列表 L_3 中搜索 (ID,V_{ID},t,PK_t,h_3)。若元组存在, 则 $\mathcal{C}$ 返回 h_3 作为应答; 否则, $\mathcal{C}$ 随机选择 $h_3\in Z_q^*$ 返回给 $\mathcal{A}_{\mathrm{I}}$, 并将 (ID,V_{ID},t,PK_t,h_3) 记录到列表 L_3 中。

4) H_4 询问: 初始 L_4 为空, $\mathcal{C}$ 在收到敌手 $\mathcal{A}_{\mathrm{I}}$ 对

于 (ID,t) 的询问后, 先在列表 L_4 中搜索 (ID,t,h_4)。若 (ID,t,h_4) 存在, 则 $\mathcal{C}$ 返回 h_4 作为应答; 否则, $\mathcal{C}$ 随机选择 $h_4 \in Z_q^*$ 返回给 $\mathcal{A}_{\mathrm{I}}$, 并将 (ID,t,h_4) 记录到列表 L_4 中。

5) H_5 询问: 初始 L_5 为空, $\mathcal{C}$ 在收到敌手 $\mathcal{A}_{\mathrm{I}}$ 对于 $(ID,PK_{ID},W_{ID},PK_{pub})$ 的询问后, 先在列表 L_5 中搜索 $(ID,PK_{ID},W_{ID},PK_{pub},h_5)$。若元组存在, 则 $\mathcal{C}$ 返回 h_5 作为应答; 否则, $\mathcal{C}$ 询问列表 L_{I}, 随机选择 $h_5 \in Z_q^*$ 返回给 $\mathcal{A}_{\mathrm{I}}$, 并将 $(ID,PK_{ID},W_{ID},PK_{pub},h_5)$ 记录到列表 L_5 中。

6) H_6 询问: 初始 L_6 为空, $\mathcal{C}$ 在收到敌手 $\mathcal{A}_{\mathrm{I}}$ 对于 M 的询问后, 先在列表 L_6 中搜索 (M,α_{ID},h_6)。若元组存在, 则 $\mathcal{C}$ 返回 h_6 作为应答; 否则, 随机选取 $\alpha_{ID} \in Z_q^*$, $\mathcal{C}$ 计算 $h_6=\alpha_{ID}bP$ 返回给 $\mathcal{A}_{\mathrm{I}}$, 并将 (M,α_{ID},h_6) 记录到表 L_6 中。

7) 用户生成询问: $\mathcal{C}$ 在收到 $\mathcal{A}_{\mathrm{I}}$ 关于身份 ID 的用户生成询问后, 选择一个 $\mathcal{BT}$ 中未被注册的节点 η 注册给身份 ID, 对于每一个节点 $\theta \in Path(\eta)$, $\mathcal{C}$ 随机选择 $w_\theta, r_{ID} \in Z_q^*$ 并计算 $PK_{ID}=r_{ID}P$, $W_{ID}=w_\theta P$。进行 H_1 询问、H_2 询问, 得到 h_1,h_2, 计算 $d_\theta=w_\theta h_2+sh_1$。

经过上述操作, 身份 ID 已被创建成功, 身份 ID 的部分私钥是 $D_{ID}=\{(\theta,d_\theta,W_{ID})\}_{\theta\in Path(\eta)}$, 秘密值是 r_{ID}。最后 $\mathcal{C}$ 将 PK_{ID} 发送给 $\mathcal{A}_{\mathrm{I}}$, 并且将元组 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID})$ 插入进列表 L_{PK}。

8) 部分私钥提取询问: $\mathcal{C}$ 在收到 $\mathcal{A}_{\mathrm{I}}$ 关于身份 ID 的用户生成询问后, 若 $ID \neq ID_C$, $\mathcal{C}$ 在列表 L_{PK} 中进行查询, 提取元组 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID})$ 并将 (d_θ,W_{ID}) 返回给 $\mathcal{A}_{\mathrm{I}}$; 否则, $\mathcal{C}$ 停止应答, 游戏结束。

9) 密钥更新询问: $\mathcal{C}$ 在收到 $\mathcal{A}_{\mathrm{I}}$ 关于 (ID,t) 的密钥更新询问后, 运行算法 $KUNode(\mathcal{BT},RL,t)$ 得到集合 Y, 对于 $\theta \in Y$, $\mathcal{C}$ 随机选取 $v_\theta \in Z_q^*$, 计算 $V_{ID}=v_\theta P$。然后 $\mathcal{C}$ 进行 H_3 询问、H_4 询问, 得到 h_3,h_4, 计算 $t_\theta=v_\theta h_4+ch_3$。并将更新密钥 $UK_{ID,t}=\{(\theta,t_\theta,V_{ID})\}_{\theta\in Y}$ 发送给 $\mathcal{A}_{\mathrm{I}}$。

10) 秘密值提取询问: $\mathcal{C}$ 在收到 $\mathcal{A}_{\mathrm{I}}$ 关于身份 ID 的秘密值提取询问后, 在列表 L_{PK} 中进行查询, 提取元组 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID})$ 并将 r_{ID} 返回给 $\mathcal{A}_{\mathrm{I}}$。

11) 公钥替换询问: $\mathcal{C}$ 在收到 $\mathcal{A}_{\mathrm{I}}$ 关于 (ID,PK_{ID}') 的密钥更新询问后, 在列表 L_{PK} 中进行查询, 提取元组 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID})$, 令 $PK_{ID}=PK_{ID}'$, $r_{ID}=\perp$, 并更新这条记录为 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID}')$。

12) 签名询问: $\mathcal{C}$ 在收到 $\mathcal{A}_{\mathrm{I}}$ 关于 (ID,t,M,PK_{ID}) 的签名询问后, $\mathcal{C}$ 执行以下操作:

① 若 $ID \neq ID_C$, 并且公钥 PK_{ID} 没有被替换, $\mathcal{C}$ 首先在列表 L_{PK} 中进行查询, 提取元组 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID})$, 进行 H_5, H_6 询问得到 h_5,h_6, 然后 $\mathcal{C}$ 先进行密钥更新询问获取更新密钥 $UK_{ID,t}$, 计算 $\sigma_{ID,1}=h_6(sk_{ID,1}+r_{ID}h_5)$, $\sigma_{ID,2}=W_{ID}h_2+V_{ID}h_4+PK_{pub}h_1+PK_th_3+PK_{ID}h_5$, 生成关于 (ID,t,M) 的签名 $\sigma_{ID}=(\sigma_{ID,1},\sigma_{ID,2})$, 并将其返回给 $\mathcal{A}_{\mathrm{I}}$。

② 若 $ID=ID_C$ 或公钥 PK_{ID} 已经被替换, $\mathcal{C}$ 停止应答, 游戏结束。

13) 重签名密钥生成询问: $\mathcal{C}$ 在收到 $\mathcal{A}_{\mathrm{I}}$ 的关于两个身份 (ID_A,ID_B,t) 的重签名密钥询问后, 进行下列操作:

① 若 $ID_A \neq ID_C$, 先运行密钥更新询问获得更新密钥 $UK_{A,t}$, $UK_{B,t}$, 获取在列表 L_{PK} 中进行查询, 提取元组 $(ID_A,d_{\theta_A},W_A,r_A,PK_A)$, $(ID_B,d_{\theta_B},W_B,r_B,PK_B)$, 进行 H_5 询问得到 h_5, 将用户私钥表示为 $SK_{A,t}=(sk_{A,1},sk_{A,2})=(d_{\theta_A}+t_{\theta_A},r_A)$, $SK_{B,t}=(sk_{B,1},sk_{B,2})=(d_{\theta_B}+t_{\theta_B},r_B)$, 计算 $rk_1=(sk_{B,1}+r_Bh_{5,B})^{-1}\cdot(W_Ah_{2,A}+V_Ah_{4,A}+PK_{pub}h_{1,A}+PK_th_{3,A}+PK_Ah_{5,A})$, $rk_2=W_Bh_{2,B}+V_Bh_{4,B}+PK_{pub}h_{1,B}+PK_th_{3.B}+PK_{HA}\cdot h_{5,B}$, 将得到的重签名密钥 $RK_t=(rk_1,rk_2)$ 发送给 $\mathcal{A}_{\mathrm{I}}$;

② 若 $ID_A=ID_C$, $\mathcal{C}$ 停止应答, 游戏结束。

14) 重签名询问: $\mathcal{C}$ 在收到 $\mathcal{A}_{\mathrm{I}}$ 的关于两个身份 $(ID_A,ID_B,t,M,PK_A,PK_B,\sigma_A)$ 的重签名询问后, $\mathcal{C}$ 进行重签名密钥生成询问, 获取重签名密钥 $RK_t=(rk_1,\ rk_2)$, 在列表 L_{PK} 中进行查询, 提取 $(ID_A,d_{\theta_A},W_A,\ r_A,PK_A),(ID_B,d_{\theta_B},W_B,r_B,PK_B)$, $\mathcal{C}$ 进行密钥更新询问获取更新密钥 $UK_{A,t}$, $UK_{B,t}$, 然后随机选取 $y \in Z_q^*$, 计算 $\sigma_{B,1}=\sigma_{A,1}x$, $\sigma_{B,2}=\sigma_{A,2}x$, $\sigma_{B,3}=rk_1x$, $\sigma_{B,4}=rk_2$, 其中 $x=(sk_{A,1}+r_Ah_{5,A})(sk_{B,1}+r_Bh_{5,B})^{-1}\cdot\ y$, $\mathcal{C}$ 得到 $\sigma_B=(\sigma_{B,1},\sigma_{B,2},\sigma_{B,3},\sigma_{B,4})$ 并将其发送给敌手 $\mathcal{A}_{\mathrm{I}}$。

15) 撤销询问: $\mathcal{C}$ 在收到 $\mathcal{A}_1$ 关于 (ID,t) 的撤销询问后, 若 $t>T$, 输出 $\perp$; 若 $t<T$, 运行算法 Revoke (ID,t,RL,st) 并将结果返回给 $\mathcal{A}_1$。

伪造

通过上述询问, 根据分叉引理, 挑战者 $\mathcal{C}$ 可以利用敌手 $\mathcal{A}_1$ 在多项式时间内产生两个伪造消息。

签名阶段: 对于身份 ID^*, 如果 $ID^* \neq ID_C$, $\mathcal{C}$ 中止伪造并结束游戏; 否则可以产生两条伪造消息签名对, 分别是 $\left(M^*,ID^*,t^*,\sigma_{ID^*}=(\sigma_1,\sigma_2)\right)$ 和 $\left(M^*,ID^*,t^*,\sigma_{ID^*}{}'=(\sigma_1',\sigma_2')\right)$, 若这两条信息有效, 则有:

$$e(\sigma_1,P)=e(\sigma_2,H_6(M^*)) \tag{6}$$

$$e(\sigma_1',P)=e(\sigma_2',H_6(M^*)) \tag{7}$$

然后可得:

$$e(\sigma_1,P)=e(\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+PK_{pub}h_1^*+PK_th_3^*),P)$$

$$e(\sigma_1',P)=e(\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+PK_{pub}h_1'+PK_th_3^*),P)$$

则挑战者 $\mathcal{C}$ 可以通过下列计算解决 eCDH 问题, 为表示便捷, 在下列计算中将 ID^* 表示为 ID, 则有:

$$\frac{\sigma_1-\sigma_1'}{h_1^*-h_1'}=\frac{\alpha_{ID}b(aPh_1^*-aPh_1')}{h_1^*-h_1'}=\frac{\alpha_{ID}abP(h_1^*-h_1')}{h_1^*-h_1'}=\alpha_{ID}abP \tag{8}$$

$\mathcal{C}$ 将 $(\alpha_{ID}P,\alpha_{ID}abP)$ 作为挑战 eCDH 实例的输出结果, 其中 α_{ID} 是列表 L_6 中与 M^* 对应的值。

重签名阶段: 对于身份 ID^*, 如果 $ID^* \neq ID_C$, $\mathcal{C}$ 中止伪造并结束游戏; 否则可以产生两条伪造信息, 分别是 $\left(M^*,ID^*,t^*,\sigma_{ID^*}=(\sigma_1,\sigma_2,\sigma_3,\sigma_4)\right)$ 和 $\left(M^*,ID^*,t^*,\sigma_{ID^*}{}'=(\sigma_1',\sigma_2',\sigma_3,\sigma_4')\right)$, 若这两条信息合法, 则有:

$$e(\sigma_1,P)=e(\sigma_2,H_6(M^*)) \tag{9}$$

$$e(\sigma_1',P)=e(\sigma_2',H_6(M^*)) \tag{10}$$

然后可得:

$$e(\sigma_1,P)=e(y\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+PK_{pub}h_1^*+PK_th_3^*),P)$$

$$e(\sigma_1',P)=e(y\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+PK_{pub}h_1'+PK_th_3^*),P)$$

则挑战者 $\mathcal{C}$ 可以通过下列计算解决 eCDH 问题, 为表示便捷, 在下列计算中将 ID^* 表示为 ID, 则有:

$$\frac{\sigma_1-\sigma_1'}{h_1^*-h_1'}=\frac{\alpha_{ID}yb(aPh_1^*-aPh_1')}{h_1^*-h_1'}=\frac{\alpha_{ID}yabP(h_1^*-h_1')}{h_1^*-h_1'}=\alpha_{ID}\sigma_3ab \tag{11}$$

$\mathcal{C}$ 将 $(\alpha_{ID}\sigma_3,\alpha_{ID}\sigma_3ab)$ 作为挑战 eCDH 实例的输出结果。

挑战者 $\mathcal{C}$ 成功解决 eCDH 困难问题的概率可以用事件 E_1, E_2, E_3 进行表示, 事件 E_1, E_2, E_3 定义为:

E_1: 敌手 $\mathcal{A}_1$ 对于身份 ID_C 没有进行部分私钥询问。

E_2: 敌手 $\mathcal{A}_1$ 在伪造阶段对于身份 $ID^* \neq ID_C$ 没有返回消息签名对 (M^*,σ_{ID^*})。

E_3: σ_{ID^*} 是一个有效的伪造签名。

当事件 E_1, E_2 不发生, 事件 E_3 发生时, 则称挑战者 $\mathcal{C}$ 赢得了游戏, 其成功的概率为 $\Pr[\neg E_1\wedge\neg E_2\wedge E_3]=\Pr[\neg E_1]\Pr[\neg E_2\mid\neg E_1]\Pr[E_3\mid\neg E_1\wedge\neg E_2]$, 其中 $\Pr[\neg E_1]\geqslant(1-\frac{1}{q_1+q_2})^{q_e}$, $\Pr[\neg E_2\mid\neg E_1]\geqslant\frac{1}{q_1}$, $\Pr[E_3\mid\neg E_1\wedge\neg E_2]\geqslant\varepsilon'$, 可得

$$\varepsilon=\Pr[\neg E_1\wedge\neg E_2\wedge E_3]\geqslant\frac{\varepsilon'}{q_1}(1-\frac{1}{q_1+q_2})^{q_e} \tag{12}$$

因此挑战者 $\mathcal{C}$ 在多项式时间内以优势 ε 成功解决了 eCDH 困难问题, 与理论相矛盾。

引理 2.在随机谕言机模型中, 假设存在敌手 $\mathcal{A}_2$ 以不可忽略的概率 ε' 打破本文方案的不可伪造性, 那么存在算法 $\mathcal{C}$ 在多项式时间 t 内以不可忽略的概率 $\varepsilon\geqslant\frac{\varepsilon'}{q_1}(1-\frac{1}{q_1+q_2})^{q_s}$ 解决 eCDH 问题,其中 q_1, q_2, q_s 分别为 H_1 询问、H_2 询问和秘密值提取询问的次数。

证明. 给定一个随机的 eCDH 问题实例 (P,aP,bP), 挑战者 $\mathcal{C}$ 通过与敌手 $\mathcal{A}_2$ 交互, 目的是输出一个 eCDH 的解 (Q,abQ), 其中 $a,b\in{}_RZ_q^*$, $Q\in{}_RG$。在挑战者 $\mathcal{C}$ 与敌手 $\mathcal{A}_2$ 之间建立游戏, 两者的交互过程具体如下所示:

初始化

挑战者 $\mathcal{C}$ 选取随机值 $s,c\in Z_q^*$, 计算 $PK_{pub}=sP$, $PK_t=cP$, $\mathcal{C}$ 安全保存系统主密钥 $msk=s$, 时间主密钥 $mtp=c$, 然后随机选取一个身份 ID_C 作为被挑

战身份, 并发送 ID_C 和系统公共参数 $params=\{G_1, G_2, e, q, P, PK_{pub}, PK_t, H_1, H_2, H_3, H_4, H_5, H_6\}$给敌手 $\mathcal{A}_2$。

询问

1) $\mathcal{A}_2$ 在进行哈希询问、密钥更新询问、签名询问、重签名密钥生成询问、重签名询问、撤销询问时, 询问过程与引理 1 的证明 1 中的操作相同, 挑战者返回相应的应答结果。

2) 用户生成询问: $\mathcal{C}$ 在收到 $\mathcal{A}_2$ 关于身份 ID 的用户生成询问后, 选择一个 $\mathcal{BT}$ 中未被注册的节点 η 注册给身份 ID, 随机选择 $w_\theta, r_{ID} \in Z_q^*$, 对于每一个节点 $\theta \in Path(\eta)$, $\mathcal{C}$ 计算 $W_{ID}=w_\theta P$。进行 H_1 询问、H_2 询问, 得到 h_1, h_2, 计算 $d_\theta = w_\theta h_2 + s h_1$。

① 若 $ID \neq ID_C$, 计算 $PK_{ID}=r_{ID}P$;

② 若 $ID=ID_C$, $\mathcal{C}$ 令 $PK_{ID}=r_{ID}aP$。

经过上述操作, 身份 ID 已被创建成功, 身份 ID 的部分私钥是 $D_{ID}=\{(\theta, d_\theta, W_{ID})\}_{\theta \in Path(\eta)}$, 秘密值是 r_{ID}。最后 $\mathcal{C}$ 将 PK_{ID} 发送给 $\mathcal{A}_2$, 并且将元组 $(ID, d_\theta, W_{ID}, r_{ID}, PK_{ID})$ 插入进列表 L_{PK}。

3) 秘密值提取询问: $\mathcal{C}$ 在收到 $\mathcal{A}_2$ 关于身份 ID 的秘密值提取询问后, 若 $ID \neq ID_C$, 在列表 L_{PK} 中进行查询, 提取元组 $(ID, d_\theta, W_{ID}, r_{ID}, PK_{ID})$ 并将 r_{ID} 返回给 $\mathcal{A}_2$; 否则, $\mathcal{C}$ 停止应答, 游戏结束。

伪造

通过上述询问, 根据分叉引理, 挑战者 $\mathcal{C}$ 可以利用敌手 $\mathcal{A}_2$ 在多项式时间内产生两个伪造消息。

签名阶段: 对于身份 ID^*, 如果 $ID^* \neq ID_C$, $\mathcal{C}$ 中止伪造并结束游戏; 否则可以产生两条伪造消息签名对, 分别是 $\left(M^*, ID^*, t^*, \sigma_{ID^*}=(\sigma_1, \sigma_2)\right)$ 和 $\left(M^*, ID^*, t^*, \sigma_{ID^*}{}'=(\sigma_1', \sigma_2')\right)$, 若这两条信息有效, 则有:

$$e(\sigma_1, P)=e(\sigma_2, H_6(M^*)) \tag{13}$$

$$e(\sigma_1', P)=e(\sigma_2', H_6(M^*)) \tag{14}$$

然后可得:

$$e(\sigma_1, P)=e(\alpha_{ID^*} b(W_{ID^*} h_2^* + V_{ID^*} h_4^* + PK_{ID^*} h_5^* + PK_{pub} h_1^* + PK_t h_3^*), P)$$

$$e(\sigma_1', P)=e(\alpha_{ID^*} b(W_{ID^*} h_2^* + V_{ID^*} h_4^* + PK_{ID^*} h_5' + PK_{pub} h_1^* + PK_t h_3^*), P)$$

则挑战者 $\mathcal{C}$ 可以通过下列计算解决 eCDH 问题, 为表示便捷, 在下列计算中将 ID^* 表示为 ID, 则有:

$$\frac{\sigma_1-\sigma_1'}{h_5^*-h_5'}=\frac{\alpha_{ID} b(r_{ID} a P h_5^* - r_{ID} a P h_5')}{h_5^*-h_5'}=\frac{\alpha_{ID} r_{ID} abP(h_5^*-h_5')}{h_5^*-h_5'}=\beta_{ID} abP \tag{15}$$

$\mathcal{C}$ 将 $(\beta_{ID}P, \beta_{ID}abP)$ 作为挑战 eCDH 实例的输出结果, 其中 $\beta_{ID}=\alpha_{ID} r_{ID}$, α_{ID} 是列表 L_6 中与 M^* 对应的值。

重签名阶段: 对于身份 ID^*, 如果 $ID^* \neq ID_C$, $\mathcal{C}$ 中止伪造并结束游戏; 否则可以产生两条伪造信息, 分别是 $\left(M^*, ID^*, t^*, \sigma_{ID^*}=(\sigma_1, \sigma_2, \sigma_3, \sigma_4)\right)$ 和 $\left(M^*, ID^*, t^*, \sigma_{ID^*}{}'=(\sigma_1', \sigma_2', \sigma_3, \sigma_4')\right)$, 若这两条信息合法, 则有:

$$e(\sigma_1, P)=e(\sigma_2, H_6(M^*)) \tag{16}$$

$$e(\sigma_1', P)=e(\sigma_2', H_6(M^*)) \tag{17}$$

然后可得:

$$e(\sigma_1, P)=e(y\alpha_{ID^*} b(W_{ID^*} h_2^* + V_{ID^*} h_4^* + PK_{ID^*} h_5^* + PK_{pub} h_1^* + PK_t h_3^*), P)$$

$$e(\sigma_1', P)=e(y\alpha_{ID^*} b(W_{ID^*} h_2^* + V_{ID^*} h_4^* + PK_{ID^*} h_5' + PK_{pub} h_1^* + PK_t h_3^*), P)$$

则挑战者 $\mathcal{C}$ 可以通过下列计算解决 eCDH 问题, 为表示便捷, 在下列计算中将 ID^* 表示为 ID, 则有:

$$\frac{\sigma_1-\sigma_1'}{h_5^*-h_5'}=\frac{\alpha_{ID} yb(r_{ID} a P h_5^* - r_{ID} a P h_5')}{h_5^*-h_5'}=\frac{\alpha_{ID} r_{ID} yabP(h_5^*-h_5')}{h_5^*-h_5'}=\beta_{ID}\sigma_3 ab \tag{18}$$

$\mathcal{C}$ 将 $(\beta_{ID}\sigma_3, \beta_{ID}\sigma_3 ab)$ 作为挑战 eCDH 实例的输出结果。

挑战者 $\mathcal{C}$ 成功解决 eCDH 困难问题的概率为

$$\varepsilon \geqslant \frac{\varepsilon'}{q_1}\left(1-\frac{1}{q_1+q_2}\right)^{q_s} \tag{19}$$

因此挑战者 $\mathcal{C}$ 在多项式时间内以优势 ε 成功解决了 eCDH 困难问题, 与理论相矛盾。

引理 3. 在随机谕言机模型中, 假设存在敌手 $\mathcal{A}_3$ 以不可忽略的概率 ε' 打破本文方案的不可伪造性, 那么存在算法 $\mathcal{C}$ 在多项式时间 t 内以不可忽略的概率 $\varepsilon \geqslant \frac{\varepsilon'}{q_1}$ 解决 eCDH 问题,其中 q_1 是 H_1 询问的次数。

证明. 给定一个随机的 eCDH 问题实例 (P, aP, bP), 挑战者 $\mathcal{C}$ 通过与敌手 $\mathcal{A}_3$ 交互, 目的是

输出一个 eCDH 的解 (Q,abQ)，其中 $a,b\in {}_RZ_q^*$，$Q\in {}_RG$。在挑战者 $\mathcal{C}$ 与敌手 $\mathcal{A}_3$ 之间建立游戏，两者的交互过程具体如下所示:

初始化

挑战者 $\mathcal{C}$ 以与引理 2 中的初始化算法相同的方式运行初始化算法。

询问

1) $\mathcal{A}_3$ 在进行哈希询问、用户生成询问、部分私钥提取询问、秘密值提取询问、公钥替换询问、签名询问、重签名密钥生成询问、重签名询问、撤销询问时，询问过程与引理 1 的证明 1 中的操作相同，挑战者返回相应的应答结果。

2) 密钥更新询问: $\mathcal{C}$ 在收到 $\mathcal{A}_3$ 关于 (ID,t) 的密钥更新询问后，运行算法 $KUNode(\mathcal{BT},RL,t)$ 得到集合 Y，对于 $\theta\in Y$，$\mathcal{C}$ 进行以下操作:

① 若 $ID\neq ID_C$，$\mathcal{C}$ 随机选取 $v_\theta\in Z_q^*$，计算 $V_{ID}=v_\theta P$。然后 $\mathcal{C}$ 进行 H_3 询问、H_4 询问，得到 h_3,h_4，计算 $t_\theta=v_\theta h_4+ch_3$。

② 若 $ID=ID_C$，$\mathcal{C}$ 令 $V_{ID}=aP$，$t_\theta=\perp$。

在以上两种情况下，$\mathcal{C}$ 都将更新密钥 $UK_{ID,t}=\{(\theta,t_\theta,V_{ID})\}_{\theta\in Y}$ 发送给 $\mathcal{A}_3$。

伪造

通过上述询问，根据分叉引理，挑战者 $\mathcal{C}$ 可以利用敌手 $\mathcal{A}_3$ 在多项式时间内产生两个伪造消息。

签名阶段: 对于身份 ID^*，如果 $ID^*\neq ID_C$，$\mathcal{C}$ 中止伪造并结束游戏; 否则可以产生两条伪造消息签名对，分别是 $\left(M^*,ID^*,t^*,\sigma_{ID^*}=(\sigma_1,\sigma_2)\right)$ 和 $\left(M^*,ID^*,t^*,\sigma_{ID^*}{}'=(\sigma_1',\sigma_2')\right)$，若这两条信息有效，则有:

$$e(\sigma_1,P)=e(\sigma_2,H_6(M^*)) \tag{20}$$

$$e(\sigma_1',P)=e(\sigma_2',H_6(M^*)) \tag{21}$$

然后可得:

$$e(\sigma_1,P)=e(\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+PK_{pub}h_1^*+PK_th_3^*),P)$$

$$e(\sigma_1',P)=e(\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4'+PK_{ID^*}h_5^*+PK_{pub}h_1^*+PK_th_3^*),P)$$

则挑战者 $\mathcal{C}$ 可以通过下列计算解决 eCDH 问题，为表示便捷，在下列计算中将 ID^* 表示为 ID，则有:

$$\frac{\sigma_1-\sigma_1'}{h_4^*-h_4'}=\frac{\alpha_{ID}b(aPh_4^*-aPh_4')}{h_4^*-h_4'}=\frac{\alpha_{ID}abP(h_4^*-h_4')}{h_4^*-h_4'}=\alpha_{ID}abP \tag{22}$$

$\mathcal{C}$ 将 $(\alpha_{ID}P,\alpha_{ID}abP)$ 作为挑战 eCDH 实例的输出结果，其中 α_{ID} 是列表 L_6 中与 M^* 对应的值。

重签名阶段: 对于身份 ID^*，如果 $ID^*\neq ID_C$，$\mathcal{C}$ 中止伪造并结束游戏; 否则可以产生两条伪造信息，分别是 $\left(M^*,ID^*,t^*,\sigma_{ID^*}=(\sigma_1,\sigma_2,\sigma_3,\sigma_4)\right)$ 和 $\left(M^*,ID^*,t^*,\sigma_{ID^*}{}'=(\sigma_1',\sigma_2',\sigma_3,\sigma_4')\right)$，若这两条信息合法，则有:

$$e(\sigma_1,P)=e(\sigma_2,H_6(M^*)) \tag{23}$$

$$e(\sigma_1',P)=e(\sigma_2',H_6(M^*)) \tag{24}$$

然后可得:

$$e(\sigma_1,P)=e(y\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+PK_{pub}h_1^*+PK_th_3^*),P)$$

$$e(\sigma_1',P)=e(y\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4'+PK_{ID^*}h_5^*+PK_{pub}h_1^*+PK_th_3^*),P)$$

则挑战者 $\mathcal{C}$ 可以通过下列计算解决 eCDH 问题，为表示便捷，在下列计算中将 ID^* 表示为 ID，则有:

$$\frac{\sigma_1-\sigma_1'}{h_4^*-h_4'}=\frac{\alpha_{ID}yb(aPh_4^*-aPh_4')}{h_4^*-h_4'}=\frac{\alpha_{ID}yabP(h_4^*-h_4')}{h_4^*-h_4'}=\alpha_{ID}\sigma_3ab \tag{25}$$

$\mathcal{C}$ 将 $(\alpha_{ID}\sigma_3,\alpha_{ID}\sigma_3ab)$ 作为挑战 eCDH 实例的输出结果。

挑战者 $\mathcal{C}$ 成功解决 eCDH 困难问题的概率为

$$\varepsilon\geqslant\frac{\varepsilon'}{q_1} \tag{26}$$

因此挑战者 $\mathcal{C}$ 在多项式时间内以优势 ε 成功解决了 eCDH 困难问题，与理论相矛盾。

引理 4.在随机谕言机模型中，假设存在敌手 $\mathcal{A}_4$ 以不可忽略的概率 ε' 打破本文方案的不可伪造性，那么存在算法 $\mathcal{C}$ 在多项式时间 t 内以不可忽略的概率 $\varepsilon\geqslant\frac{\varepsilon'}{q_1}(1-\frac{1}{q_1+q_2})^{q_s}$ 解决 eCDH 问题，其中 q_1，q_2，q_s 分别为 H_1 询问、H_2 询问、秘密值提取询问的次数。

证明. 给定一个随机的 eCDH 问题实例 (P,aP,bP)，挑战者 $\mathcal{C}$ 通过与敌手 $\mathcal{A}_4$ 交互，目的是

输出一个 eCDH 的解 (Q,abQ)，其中 $a,b\in {}_RZ_q^*$，$Q\in {}_RG$。在挑战者 $\mathcal{C}$ 与敌手 $\mathcal{A}_4$ 之间建立游戏, 两者的交互过程具体如下所示:

初始化

挑战者 $\mathcal{C}$ 以与引理 2 中的初始化算法相同的方式运行初始化算法。

询问

1) $\mathcal{A}_4$ 在进行哈希询问、部分私钥提取询问、秘密值提取询问、重签名密钥生成询问、重签名询问、撤销询问时, 询问过程与引理 1 的证明 1 中的操作相同, 挑战者返回相应的应答结果。

2) 用户生成询问: $\mathcal{C}$ 在收到 $\mathcal{A}_4$ 关于身份 ID 的用户生成询问后, 选择一个 $\mathcal{BT}$ 中未被注册的节点 η 注册给身份 ID，随机选择 $w_\theta,r_{ID}\in Z_q^*$，计算 $PK_{ID}=r_{ID}P$。然后对于每一个节点 $\theta\in Path(\eta)$，$\mathcal{C}$ 进行以下操作:

① 若 $ID\neq ID_C$，$\mathcal{C}$ 随机选择 $w_\theta\in Z_q^*$，计算 $W_{ID}=w_\theta P$，然后进行 H_1 询问、H_2 询问, 得到 h_1,h_2，计算 $d_\theta=w_\theta h_2+sh_1$。

② 若 $ID=ID_C$，$\mathcal{C}$ 令 $W_{ID}=aP$，$d_\theta=\perp$。

经过上述操作, 身份 ID 已被创建成功, 身份 ID 的部分私钥是 $D_{ID}=\{(\theta,d_\theta,W_{ID})\}_{\theta\in Path(\eta)}$，秘密值是 r_{ID}。最后 $\mathcal{C}$ 将 PK_{ID} 发送给 $\mathcal{A}_4$，并且将元组 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID})$ 插入进列表 L_{PK}。

3) 签名询问: $\mathcal{C}$ 在收到 $\mathcal{A}_4$ 关于 (ID,t,M,PK_{ID}) 的签名询问后, $\mathcal{C}$ 执行以下操作:

① 若 $ID\neq ID_C$，并且公钥 PK_{ID} 没有被替换, $\mathcal{C}$ 首先在列表 L_{PK} 中进行查询, 提取元组 $(ID,d_\theta,W_{ID},r_{ID},PK_{ID})$，进行 H_5、H_6 询问得到 h_5,h_6，然后 $\mathcal{C}$ 进行密钥更新询问获取更新密钥 $UK_{ID,t}$，并将元组 (ID,V_{ID},t,PK_t,h_3) 记录在列表 L_3 中，将元组 (ID,t) 记录在列表 L_4 中，计算 $\sigma_{ID,1}=h_6(sk_{ID,1}+r_{ID}h_5)$，$\sigma_{ID,2}=W_{ID}h_2+V_{ID}h_4+PK_{pub}h_1+PK_th_3+PK_{ID}h_5$，生成关于 (ID,t,M) 的签名 $\sigma_{ID}=(\sigma_{ID,1},\sigma_{ID,2})$ 并将其返回给 $\mathcal{A}_4$。

② 若 $ID=ID_C$ 或公钥 PK_{ID} 已经被替换, $\mathcal{C}$ 停止应答, 游戏结束。

伪造

通过上述询问, 根据分叉引理, 挑战者 $\mathcal{C}$ 可以利用敌手 $\mathcal{A}_4$ 在多项式时间内产生两个伪造消息。

签名阶段: 对于身份 ID^*，如果 $ID^*\neq ID_C$，$\mathcal{C}$ 中止伪造并结束游戏; 否则可以产生两条伪造消息签名对, 分别是 $\left(M^*,ID^*,t^*,\sigma_{ID^*}=(\sigma_1,\sigma_2)\right)$ 和 $\left(M^*,ID^*,t^*,\sigma_{ID^*}{}'=(\sigma_1',\sigma_2')\right)$，若这两条信息有效, 则有:

$$e(\sigma_1,P)=e(\sigma_2,H_6(M^*)) \tag{27}$$

$$e(\sigma_1',P)=e(\sigma_2',H_6(M^*)) \tag{28}$$

然后可得:

$$\begin{aligned}e(\sigma_1,P)=e(&\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+\\&PK_{pub}h_1^*+PK_th_3^*),P)\end{aligned}$$

$$\begin{aligned}e(\sigma_1',P)=e(&\alpha_{ID^*}b(W_{ID^*}h_2'+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+\\&PK_{pub}h_1^*+PK_th_3^*),P)\end{aligned}$$

则挑战者 $\mathcal{C}$ 可以通过下列计算解决 eCDH 问题, 为表示便捷, 在下列计算中将 ID^* 表示为 ID，则有:

$$\begin{aligned}\frac{\sigma_1-\sigma_1'}{h_2^*-h_2'}=\frac{\alpha_{ID}b(aPh_2^*-aPh_2')}{h_2^*-h_2'}=\\\frac{\alpha_{ID}abP(h_2^*-h_2')}{h_2^*-h_2'}=\alpha_{ID}abP\end{aligned} \tag{29}$$

$\mathcal{C}$ 将 $(\alpha_{ID}P,\alpha_{ID}abP)$ 作为挑战 eCDH 实例的输出结果, 其中 α_{ID} 是列表 L_6 中与 M^* 对应的值。

重签名阶段: 对于身份 ID^*，如果 $ID^*\neq ID_C$，$\mathcal{C}$ 中止伪造并结束游戏; 否则可以产生两条伪造信息, 分别是 $\left(M^*,ID^*,t^*,\sigma_{ID^*}=(\sigma_1,\sigma_2,\sigma_3,\sigma_4)\right)$ 和 $\left(M^*,ID^*,t^*,\sigma_{ID^*}{}'=(\sigma_1',\sigma_2',\sigma_3,\sigma_4')\right)$，若这两条信息合法, 则有:

$$e(\sigma_1,P)=e(\sigma_2,H_6(M^*)) \tag{30}$$

$$e(\sigma_1',P)=e(\sigma_2',H_6(M^*)) \tag{31}$$

然后可得:

$$\begin{aligned}e(\sigma_1,P)=e(&y\alpha_{ID^*}b(W_{ID^*}h_2^*+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+\\&PK_{pub}h_1^*+PK_th_3^*),P)\end{aligned}$$

$$\begin{aligned}e(\sigma_1',P)=e(&y\alpha_{ID^*}b(W_{ID^*}h_2'+V_{ID^*}h_4^*+PK_{ID^*}h_5^*+\\&PK_{pub}h_1^*+PK_th_3^*),P)\end{aligned}$$

则挑战者 $\mathcal{C}$ 可以通过下列计算解决 eCDH 问题, 为表示便捷, 在下列计算中将 ID^* 表示为 ID，则有:

$$\begin{aligned}\frac{\sigma_1-\sigma_1'}{h_2^*-h_2'}=\frac{\alpha_{ID}yb(aPh_2^*-aPh_2')}{h_2^*-h_2'}=\\\frac{\alpha_{ID}yabP(h_2^*-h_2')}{h_2^*-h_2'}=\alpha_{ID}\sigma_3ab\end{aligned} \tag{32}$$

$\mathcal{C}$ 将 $(\alpha_{ID}\sigma_3, \alpha_{ID}\sigma_3 ab)$ 作为挑战 eCDH 实例的输出结果。

挑战者 $\mathcal{C}$ 成功解决 eCDH 困难问题的概率为

$$\varepsilon \geqslant \frac{\varepsilon'}{q_1}(1-\frac{1}{q_1+q_2})^{q_s} \tag{33}$$

因此挑战者 $\mathcal{C}$ 在多项式时间内以优势 ε 成功解决了 eCDH 困难问题, 与理论相矛盾。

由以上四条引理的证明可得, 本文方案在自适应选择消息攻击下具有不可伪造性。

7 性能分析

本章将本文方案与方案[13, 21, 29, 30]的功能进行对比, 进行了计算开销和通信开销的对比分析, 利用了JPBC库进行了算法的仿真。此外, 与方案[30]中所用的撤销方法进行了更新密钥开销的对比。

实验配置为 Intel(R) Core(TM) i5-8500 CPU@ 3.00GHz 3.00 GHz 处理器, 8GB 内存。实验在 Windows10 操作系统下使用 IDEA 编译器, 利用 JPBC 密码库仿真并统计几种基本运算的消耗时间, 如表 1 所示。

表 1 基本运算消耗时间

Table 1 Basic operations consume time

基本运算符号	描述	执行时间/ms
T_{par}	双线性对运算	5.251
T_{sm}	群 G_1 的标量乘运算	8.460
T_h	哈希函数映射到 G_1 上运算	18.933

在表 2 中列出了本文方案与其他方案的功能对比, 其中√表示方案具有该性质, ×表示方案不具有该性质。从表 2 可以看出方案[13]没有采用无证书体制, 方案[30]不具备单向性、非交互性和前向安全性, 方案[13, 21, 29]都不具有撤销功能和前向安全性。所以就整体功能性质而言, 本文方案是具有优越性的。

由于在本方案实施的过程中, 签名验证阶段和重签名阶段由同一实体完成, 所以本文从签名、签名验证及重签名、重签名验证三个过程对比了方案[13, 21, 29, 30]与本文方案的计算开销, 如表 2 所示, 其中, —表示方案不具有该阶段。假设这些方案均选择阶为素数 q 的群 G_1, G_2, 在计算各个阶段的计算开销时, 不考虑计算量较小的点加运算等。如图 3 所示, 其中在重签名阶段, 本文方案耗时明显小于方案[13]与方案[21]; 在验证阶段, 本文方案耗时小于其他四种方案; 在签名阶段本文方案耗时较长, 但是因为本文具备撤销功能, 签名阶段耗时较长是可以接受的。如图 4 所示, 本文方案由于具有重签名的功能, 所以总计算开销大于方案[30]是可以接受的。总体来说, 本文方案的总计算开销还是明显小于方案[13, 21, 29]的。

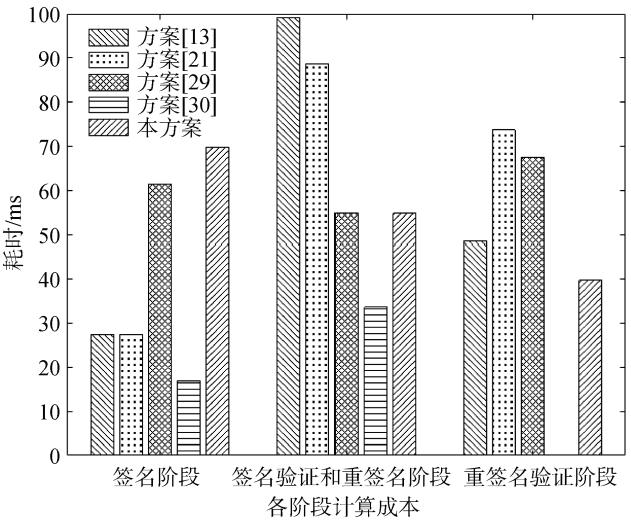


图 3 各阶段计算开销对比

Figure 3 Calculation cost comparison of each stage

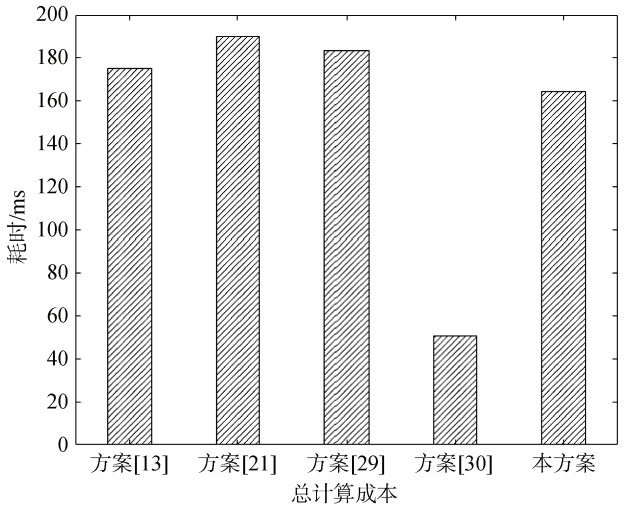


图 4 总计算开销对比

Figure 4 Total computational overhead comparison

如表 2 所示, 我们对比了方案[13, 21, 29, 30]以及本文方案的通信成本, 其中, $|G|$ 表示在群 G_1 中一个元素的平均长度。由于仿真中采用了 JPBC 库中的 Type A 类曲线构造对称质数阶双线性群, 其中群的阶数为 512bits, 所以 $|G|=512\text{bits}$。显然, 可以看出本文方案的签名长度小于方案[29, 30], 重签名长度小于方案[29]。本文方案的签名长度、重签名长度与方案[13, 21]相同。

表 2　方案的计算开销和通信开销

Table 2　The computational cost and communication cost of the scheme

	方案[13]	方案[21]	方案[29]	方案[30]	本方案
不可伪造性	√	√	√	√	√
单向性	√	√	√	×	√
非交互性	√	√	√	×	√
前向安全性	×	×	×	×	√
无证书	×	√	√	√	√
用户撤销	×	×	×	√	√
签名阶段	$T_{sm}+T_h$	$T_{sm}+T_h$	$5T_{sm}+T_h$	$2T_{sm}$	$6T_{sm}+T_h$
签名验证和重签名阶段	$2T_{par}+6T_{sm}+2T_h$	$2T_{par}+7T_{sm}+T_h$	$2T_{par}+3T_{sm}+T_h$	$4T_{sm}$	$2T_{par}+3T_{sm}+T_h$
重签名验证阶段	$4T_{par}+T_{sm}+T_h$	$4T_{par}+4T_{sm}+T_h$	$6T_{par}+2T_{sm}+T_h$	—	$4T_{par}+T_h$
总时间/ms	$6T_{par}+8T_{sm}+4T_h$ $=174.918$	$6T_{par}+12T_{sm}+3T_h$ $=189.825$	$8T_{par}+10T_{sm}+3T_h$ $=183.407$	$6T_{sm}=50.76$	$6T_{par}+9T_{sm}+3T_h$ $=164.445$
签名长度/bits	$2\|G\|=1024$	$2\|G\|=1024$	$4\|G\|=2048$	$3\|G\|=1536$	$2\|G\|=1024$
重签名长度//bits	$4\|G\|=2048$	$4\|G\|=2048$	$6\|G\|=3072$	—	$4\|G\|=2048$

通过计算开销与通信开销的对比分析, 可以看出本文方案在保证安全性的同时, 具有较低的计算成本和通信成本, 更加适用于资源有限的移动医疗环境中。

基于 Shen 等人提出的撤销方法, 方案[30]通过第三方停止更新用户时间密钥的方式来实现用户的撤销。而本文方案中采用了 KUNode 算法来实现用户撤销。所以, 我们将方案[30]与本文方案的撤销开销, 即密钥更新成本, 进行了对比。

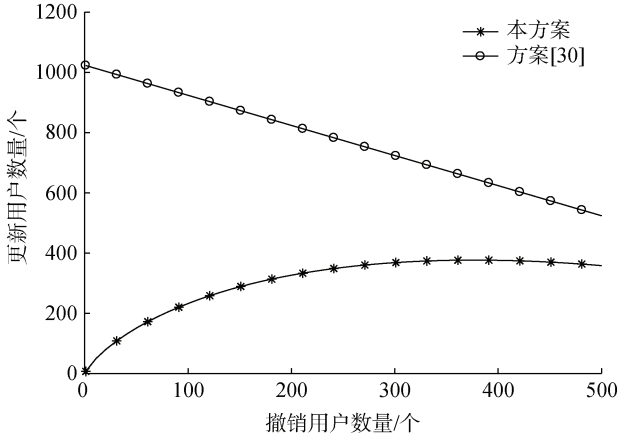


图 5　密钥更新成本对比

Figure 5　Key update cost comparison

在我们的实验中, 将系统的密钥更新代价看做是用户总数量 N 与撤销用户数量 r 之间的函数。我们将用户总数量 N 设置为 1024, 则由图 5 可以明显看出, 当 $r \leqslant \frac{N}{2}$ 时, 本系统密钥更新代价仅与用户撤销数量呈对数关系, 而方案[30]的密钥更新代价与用户撤销数量则呈线性关系, 当 $\frac{N}{2} < r \leqslant N$ 时, 本文方案与方案[30]的密钥更新代价相近, 与用户撤销数量呈线性关系。因此, 本文方案具有更低的撤销开销, 更适合应用在拥有大规模用户的医疗系统中。

8　结束语

为了保证 MHS 系统中用户数据在云端的隐私性以及用户密钥泄露或服务到期等问题, 本文基于 Shen 方案[26]和 KUNode 算法[28]构造了一种可撤销的无证书代理重签名方案, 支持用户撤销功能, 并且具有单向性、非交互性以及前向安全性。在随机谕言机模型下证明了其安全性依赖于 eCDH 困难问题, 说明了在安全性上, 本文方案在 eCDH 假设下满足自适应选择消息攻击下的不可伪造性。同时, 本文方案与其他方案进行了详细的性能分析与成本对比, 说明了本文方案在具有更优越的功能的同时, 在效率性上具备更低的计算成本、通信成本以及撤销成本, 更加适用于资源有限的大规模移动医疗系统中。

参考文献

[1] Park J H, Seol J A, Oh Y H. Design and Implementation of an Effective Mobile Healthcare System Using Mobile and RFID Technology[C]. *Proceedings of 7th International Workshop on Enterprise networking and Computing in Healthcare Industry*, *2005. HEALTHCOM*, 2005: 263-266.

[2] Ma M M, He D B, Khan M K, et al. Certificateless Searchable Public Key Encryption Scheme for Mobile Healthcare System[J]. *Computers & Electrical Engineering*, 2018, 65: 413-424.

[3] Zhang J H, Bai W L, Wang Y H. Non-Interactive ID-Based Proxy

re-Signature Scheme for IoT Based on Mobile Edge Computing[J]. *IEEE Access*, 2019, 7: 37865-37875.

[4] Mao Y Y, Zhang J, Letaief K B. Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices[J]. *IEEE Journal on Selected Areas in Communications*, 2016, 34(12): 3590-3605.

[5] Etsi M. Mobile edge computing (mec); framework and reference architecture[J]. *ETSI, DGS MEC*, 2016, 3: 1-18.

[6] Ateniese G, Hohenberger S. Proxy re-Signatures: New Definitions, Algorithms, and Applications[C]. *The 12th ACM conference on Computer and communications security*, 2005: 310-319.

[7] Blaze M, Bleumer G, Strauss M. Divertible Protocols and Atomic Proxy Cryptography[M]. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998: 127-144.

[8] Libert B, Vergnaud D. Multi-Use Unidirectional Proxy re-Signatures[C]. *The 15th ACM conference on Computer and communications security*, 2008: 511-520.

[9] Shao J, Cao Z F, Wang L C, et al. Proxy re-Signature Schemes without Random Oracles[C]. *International Conference on Cryptology in India*, 2007: 197-209.

[10] Yang X D, Xiao L K, Li Y T, et al. Identity-Based Blind Proxy re-Signature Scheme for Data Security[C]. *2018 IEEE Third International Conference on Data Science in Cyberspace*, 2018: 28-32.

[11] Lee E, Kim S W. Non-Interactive Conditional Proxy re-Signature in the Standard Model[J]. *The Computer Journal*, 2018, 61(12): 1772-1782.

[12] Wang Z W, Xia A D, He M J. ID-Based Proxy re-Signature without Pairing[J]. *Telecommunication Systems*, 2018, 69(2): 217-222.

[13] Shao J, Wei G Y, Ling Y, et al. Unidirectional Identity-Based Proxy re-Signature[C]. *2011 IEEE International Conference on Communications*, 2011: 1-5.

[14] Al-Riyami S S, Paterson K G. Certificateless Public Key Cryptography[C]. *International Conference on the Theory and Application of Cryptology and Information Security*, 2003: 452-473.

[15] Xiong H. Cost-Effective Scalable and Anonymous Certificateless Remote Authentication Protocol[J]. *IEEE Transactions on Information Forensics and Security*, 2014, 9(12): 2327-2339.

[16] Xiong H, Qin Z G. Revocable and Scalable Certificateless Remote Authentication Protocol with Anonymity for Wireless Body Area Networks[J]. *IEEE Transactions on Information Forensics and Security*, 2015, 10(7): 1442-1455.

[17] Guo D T, Wei P, Yu D, et al. A Certificateless Proxy re-Signature Scheme[C]. *2010 3rd International Conference on Computer Science and Information Technology*, 2010: 157-161.

[18] Xiao H Y, Zhang M Q. Provably-Secure Certificateless Proxy re-Signature Scheme[C]. *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, 2013: 591-594.

[19] Hu X M, Liu Y, Xu H J, et al. Analysis and Improvement of Certificateless Signature and Proxy re-Signature Schemes[C]. *2015 IEEE Advanced Information Technology, Electronic and Automa*.

[20] Chen L, Chen X Y, Sun Y, et al. A New Certificateless Proxy re-Signature Scheme in the Standard Model[C]. *2014 Seventh International Symposium on Computational Intelligence and Design*, 2014: 202-206.

[21] Chen Y N, Xu W X, Peng L, et al. Light-Weight and Privacy-Preserving Authentication Protocol for Mobile Payments in the Context of IoT[J]. *IEEE Access*, 1809, 7: 15210-15221.

[22] Rabaninejad R, Attari M A, Asaar M R, et al. A Lightweight Auditing Service for Shared Data with Secure User Revocation in Cloud Storage[J]. *IEEE Transactions on Services Computing*, 2022, 15(1): 1-15.

[23] Xiong H, Wu Y, Jin C J, et al. Efficient and Privacy-Preserving Authentication Protocol for Heterogeneous Systems in IIoT[J]. *IEEE Internet of Things Journal*, 2020, 7(12): 11713-11724.

[24] Fan C N, Karati A, Yang P S. Reliable File Transfer Protocol with Producer Anonymity for Named Data Networking[J]. *Journal of Information Security and Applications*, 2021, 59: 102851.

[25] Xiong H, Zhou Z D, Wang L L, et al. An Anonymous Authentication Protocol with Delegation and Revocation for Content Delivery Networks[J]. *IEEE Systems Journal*, 2022, 16(3): 4118-4129.

[26] Shen L M, Zhang F T, Sun Y X. Efficient Revocable Certificateless Encryption Secure in the Standard Model[J]. *The Computer Journal*, 2014, 57(4): 592-601.

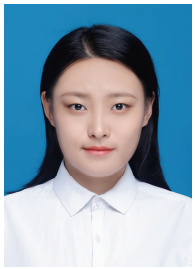
[27] Yang X D, Chen C L, Ma T C, et al. Revocable Identity-Based Proxy re-Signature Against Signing Key Exposure[J]. *PLoS ONE*, 2018, 13(3): e0194783.

[28] Boldyreva A, Goyal V, Kumar V. Identity-Based Encryption with Efficient Revocation[C]. *The 15th ACM conference on Computer and communications security*, 2008: 417-426.

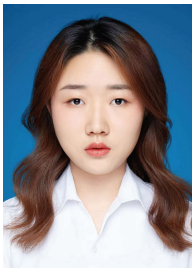
[29] Wu Y, Xiong H, Jin C J. A Multi-Use Unidirectional Certificateless Proxy re-Signature Scheme[J]. *Telecommunication Systems*, 2020, 73(3): 455-467.

[30] Liu Y, Wang D, Wang Z M, et al. Efficient Revocable Certificateless Signature Scheme for Cloud Computing[J]. *Computer Engineering and Design*, 2020, 41(9): 2442-2446.
(刘艳, 王丹, 汪祖民, 等. 云计算中高效可即时撤销的无证书签名方案[J]. *计算机工程与设计*, 2020, 41(9): 2442-2446.)

郭瑞 于 2014 年在北京邮电大学获得信息安全专业博士学位。现在西安邮电大学网络空间安全学院副教授, CCF 会员。研究领域为云计算安全,区块链技术。研究兴趣包括密码学、区块链等。Email: guorui@xupt.edu.cn



刘颖菲 于 2020 年在陕西科技大学物联网专业获得学士学位。现在西安邮电大学网络空间安全专业攻读硕士研究生学位。研究领域为密码学。研究兴趣包括信息安全、区块链技术等。Email: hellopanshang@163.com



王翊丞 于 2020 年在西安邮电大学获得信息对抗技术专业学士学位。现在西安邮电大学网络空间安全专业攻读硕士研究生学位。研究领域为密码学。研究兴趣包括公钥密码学、网络安全等。Email: wyc1522325840@163.com

蒙彤 于 2020 年在宝鸡文理大学信息与计算科学专业获得学士学位。现在西安邮电大学网络空间安全专业攻读硕士研究生学位。研究领域为密码学。研究兴趣包括公钥密码学、云计算等。Email: m18829387595@163.com