一种支持并行密钥隔离的无证书聚合签名方案

杨忆欧[1,2],彭长根[1,2,3],丁红发[4],许德权[1,2]
(1. 贵州大学 计算机科学与技术学院,贵州 贵阳 550025;
2. 贵州大学 公共大数据国家重点实验室,贵州 贵阳 550025;
3. 贵州大学 密码学与数据安全研究所,贵州 贵阳 550025;
4. 贵州财经大学 信息学院,贵州 贵阳 550025)

摘 要:聚合签名已成为数据安全认证领域重要的密码原语,其证书及密钥管理开销可通过引入无证书密码体制加以缩减。然而,密钥泄漏仍是聚合签名体制所面临的主要安全挑战,对于面向多用户的聚合签名,敌手可通过某个用户泄露的密钥破坏有此用户参与生成的聚合签名。鉴于此问题,提出一种支持并行密钥隔离的无证书聚合签名方案。首先,采用并行密钥隔离机制分时间片段更新密钥的思想,从而定时更新签名参与用户的密钥,在确保密钥前向安全及后向安全的前提下可支持较频繁的临时签名密钥更新操作;其次,利用无证书椭圆曲线密码技术实现方案构造,在降低了密码运算复杂度的同时,聚合签名长度也维持在常数量级;最后,基于随机预言模型给出该方案的形式化安全模型,证明其可以达到抵御适应性选择消息存在性伪造的安全目标。性能分析表明,该方案相较于其他方案在运算开销及签名传输通信成本方面占优。

关键词:聚合签名;并行密钥隔离;无证书密码体制;随机预言模型;可证安全
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2022)11-0106-09
doi:10.3969/j.issn.1673-629X.2022.11.016

A Certificateless Aggregate Signature Scheme Supporting Parallel Key-isolated

YANG Yi-ou[1,2],PENG Chang-gen[1,2,3],DING Hong-fa[4],XU De-quan[1,2]
(1. School of Computer Science and Technology,Guizhou University,Guiyang 550025,China;
2. State Laboratory of Public Big Data,Guizhou University,Guiyang 550025,China;
3. Institute of Cryptography and Data Security,Guizhou University,Guiyang 550025,China;
4. School of Information,Guizhou University of Finance and Economics,Guiyang 550025,China)

Abstract:Aggregate signature has become an important cryptographic primitive in the field of data security authentication,and its certificate and key management overhead can be reduced by introducing a certificateless cryptosystem. However,key leakage is still the main security challenge faced by the aggregated signature system. For multi-user-oriented aggregated signatures,an adversary can destroy the aggregated signature generated by a user through the key leaked by this user. In view of this problem,a certificateless aggregated signature scheme that supports parallel key isolation is proposed. Firstly,the parallel key-isolated mechanism is used to update the key in time segments,so as to regularly update the keys of the users participating in the signature,which can support more frequent temporary signature key updates under the premise of ensuring the forward security and backward security of the keys. Secondly,the scheme construction is realized by using certificateless elliptic curve cryptography,which reduces the complexity of cryptographic operations while maintaining the length of the aggregate signature at a constant level. Finally,a formalized security model of the scheme is given based on the random oracle model,which proves that it can achieve the security goal of resisting adaptive selection message existence forgery. The performance analysis shows that the scheme is superior in computing overhead and communication cost of signature compared with other schemes.

收稿日期:2021-11-27 **修回日期**:2022-03-29
基金项目:国家自然科学基金项目(1836205);贵州省科技计划基金项目(黔科合平台人才[2020]5017);贵州省教育厅自然科学项目(黔教合KY[2021]140)
作者简介:杨忆欧(1996-),男,硕士,研究方向为数字签名及签密;通讯作者:彭长根(1963-),男(侗),博导,教授,CCF 高级会员(48309S),研究方向为密码学、隐私保护。

Key words:aggregate signature;parallel key-isolated;certificateless cryptography;random oracle model;provably security

0 引　言

云计算、大数据和 5G 等信息技术迅速普及,面向群组的通信需求随之与日俱增,身份认证是实现群组中多用户安全通信面临的首要问题。基于非对称密码体制设计的数字签名因具备认证性、不可抵赖性等性质而成为关键的密码原语。群体的认证需求向数字签名体制提出了新要求,促使其由原本的双方参与逐步扩展到多用户参与。

聚合签名作为一种面向群体认证的特殊数字签名技术,在保持传统数字签名性质的前提下,兼具签名压缩及批量验证特性,有效缩减了签名总长度和验证成本,其在带宽和资源受限的群组通信场景中优势突出。2003 年,由 Boneh 等人[1]正式定义了聚合签名的基本原语,并利用 BLS 短签名方案[2]基本构造给出了具体方案。同年,Al-Riyami 等人[3]提出无证书公钥密码学(certificateless public key cryptography,CL-PKC)的基本思想,用户密钥的生成不完全由密钥生成中心(key generation center,KGC)决定,而是由用户和 KGC 共同产生,旨在消除经典公钥机制中繁琐的公钥证书管理及身份基密码机制(identity-based public key cryptography,ID-PKC)中固有的密钥托管。

无证书密码体制的出现使多数公钥密码系统在运行效率上有了进一步提升的可能,适用于不同场景的特殊无证书密码方案[4-6]陆续被提出。用无证书密码体制构造的聚合签名(certificateless aggregate signature,CLAS)作为设计面向群体的安全认证协议及技术所需的关键密码部件,吸引学术界开展深入研究。2015 年,陈虎等[7]以双线性映射构造了一个安全性较高的 CLAS,在生成聚合签名之前将提前协商的状态信息公布给合法签名者,使任意合法用户能动态地参与聚合签名,避免了聚合过程中用户交互开销,提升了方案的灵活性与通信效率。2018 年,Kumar 等[8]针对资源受限的无线医疗传感网络设计了一种高效的 CLAS,在聚合签名验证阶段只用了常数量级的三个双线性配对操作,方案的能量消耗及计算开销优于同类方案。次年,Kumar 等[9]又考虑了车载网中通信带宽有限的问题,基于自己提出的无证书签名方案构造了 CLAS,利用伪身份的方式实现了车辆用户的条件隐私保护。虽然使用双线性映射构造的方案在确保安全性的基础上能选取长度更短的密钥,但是此类方案大多有实用性、适用性及效率方面的问题。因此,许多研究者开始考虑非配对的无证书聚合签名方案的设计。2020 年,Zhao 等[10]设计了一种适用于计算及能耗有限的轻量级设备的 CLAS,并对方案做了详尽的安全性分析及实验仿真,表明了方案的实用性。Liu 等[11]在同年也提出了一种不含双线性映射的 CLAS,并同时聚合了随机数及签名,存储及计算开销有较大改善,但该方案并未满足其所定义的安全要求。

综上可知,目前的研究主要聚焦于聚合签名的功能、安全性及效率方面,密钥的安全性并未受到足够的重视。某个密钥如果泄露,敌手便能使用该密钥执行特定攻击,由其参与产生的任意签名都不再被信任。针对这个风险最朴素的解决思路是直接删除该密钥及其相关的签名,然而这对签名系统资源消耗和浪费非常大。密钥演化技术为解决密钥泄漏问题提供了新思路,其基本思想是将密码系统的生命周期划分为若干时间片段,分别对每一时间片段的密钥进行更新,在不同时间片段内使用对应密钥,即便攻击者获取到某一时间片段的密钥,系统在其他时间片段的运转依然不受影响。2018 年,韦性佳等[12]基于强 RSA 问题设计了一个前向安全的身份基聚合签名,在签名产生时加入了前向安全的思想,进而提高了签名系统安全性。2019 年,Jihye 等[13]构造了一种有序聚合签名方案,其满足前向安全,并将之用于计算机系统中的审计日志完整性检测。前向安全理论可解决密钥泄露的部分问题,但大多数前向安全的签名方案均存在一个缺陷,即无法保障方案的后向安全性。

前向安全技术的缺陷在 Dodis 等[14]提出密钥隔离机制得以弥补。2019 年,Xiong 等[15]给出了一个无证书并行密钥隔离签名方案,为解决智能设备中数据传输及共享的认证问题提供了帮助,但并未考虑签名聚合的可行性。目前针对密钥隔离机制下聚合签名方案的研究[16-17]依然不多,如何设计抗密钥泄露、运行效率较高的聚合签名方案仍然是当下的难点。

基于上述分析,该文提出了一种支持并行密钥隔离的无证书聚合签名方案,记为 CL-PKIAS。主要工作分述如下:

首先,通过引入并行密钥隔离这一安全密钥更新机制,给出了它在密钥更新阶段的动态更新流程。签名者以轮替的方式与两个协助者交互实现部分签名密钥更新。并且签名用户与 KGC、协助者在传输部分私钥及密钥更新消息时无需安全信道。

其次,基于椭圆曲线密码体制构造了 CL-PKIAS 方案,并形式化地给出了方案架构及其安全模型,将其安全性归结到椭圆曲线群上的离散对数困难问题,基于随机预言机模型证明了所提方案在适应性选择消息攻击下可以抵抗存在性伪造。

最后,性能分析显示,该方案生成的聚合签名长度

和签名参与人数不相关。由于避免了复杂的双线性映射操作,签名及验证均基于椭圆曲线循环群进行,故方案在运算效率及通信开销上具有优势。

1 基础知识

1.1 椭圆曲线群

有限素域上满足 $y^2 = x^3 + ax + b(\mathrm{mod}\ p)$, a , $b \in F_p$ 且 $4a^3 + 27b^2(\mathrm{mod}\ p) \neq 0$ 的点集构成了椭圆曲线 $E(F_p)$ 。定义 $G = \{P \mid P \in E(F_p)\} \cup \{O\}$ 为 $E(F_p)$ 的点集同无穷远点 O 构成的椭圆曲线加法循环群,满足以下性质:

(1)加法运算:设 Q,$P \in G$, R 表示由 Q , P 确定的直线同椭圆曲线的交点,过 R 作垂线与椭圆曲线交于 R' ,则 $Q + P = R'$ 。

(2)标量乘法:若 $P \in G$, q 为 G 的阶, $k \in Z_q^*$,则 $kP = P + P + \cdots + P$ (共 k 次)。

1.2 ECDLP 问题

对生成元为 P 的素数 q 阶群 G ,令 a 为 Z_q^* 中未知随机元素,定义任意多项式时间算法 $\mathcal{A}$ 解决 ECDLP 问题的优势 $\mathrm{Adv}_{\mathrm{ECDLP}} = \Pr[\mathcal{A}(P, aP) = a]$,若 $\mathrm{Adv}_{\mathrm{ECDLP}}$ 均可忽略,则 ECDLP 问题难解。

2 方案定义及安全模型

2.1 CL-PKIAS 方案的形式化定义

方案形式化地记为多项式概率算法 CL-PKIAS = (Setup, Key-Generate, Key-Update, Sign-Generate, Sign-Aggregate, Aggregate-Verify),其中各算法的工作流程如下:

Setup:输入安全参数 λ ,KGC 产生并公布公开参数列表 params ,生成系统主密钥 msk 秘密留存。

Key-Generate:由 KGC 与用户交互执行该算法,输入 params ,用户 C_i 的身份信息 ID_i 及时间片段 t 。

(1) C_i 选择秘密值生成公/私钥对 $(\mathrm{upk}_i, \mathrm{usk}_i)$;

(2)KGC 利用给定信息及系统主密钥为 C_i 生成初始部分公私钥对 $(\mathrm{ppk}_i, \mathrm{sk}_{i,0})$;

(3) $(\mathrm{ppk}_i, \mathrm{upk}_i)$ 及 $(\mathrm{sk}_{i,0}, \mathrm{usk}_i)$ 为 C_i 的初始完整公/私钥对。

Key-Update:该算法由用户 C_i 依次与两协助者交互执行,输入 params , C_i 身份信息 ID_i ,时间片段 t 。

(1) C_i 根据 t 指定协助者 $j(j \equiv t(\mathrm{mod}\ 2))$,利用其私钥 hsk_j 产生更新信息 $U_{i,t,j}$ 。

(2) C_i 利用 $U_{i,t,j}$ 生成临时部分私钥 $\mathrm{sk}_{i,t}$ 。

Sign-Generate:用户 C_i 执行此算法,输入 params ,待签名消息 m_i ,时间片段 t 的完整临时签名密钥对 $(\mathrm{usk}_i, \mathrm{sk}_{i,t})$,签名者 C_i 生成签名 S_i 。

Sign-Aggregate:输入 params , $(C_i)_{i=1,2,\cdots,n}$ 对消息 $(m_i)_{i=1,2,\cdots,n}$ 的签名 $(S_i)_{i=1,2,\cdots,n}$,聚合者进行签名聚合获得 S 并输出。

Aggregate-Verify:输入 params ,聚合签名 S ,用户 $(C_i)_{i=1,2,\cdots,n}$ 对应的完整公钥,验证 S 的合法性,验证者经判定后输出 true 或 false。

2.2 CL-PKIAS 方案的安全模型

在无证书签名体制中,用户的签名密钥由两方产生,伪造一个有效签名必须获取完整用户密钥。因此,若下述两类敌手通过任意的选择消息均无法进行存在性伪造,则表明 CL-PKIAS 方案是安全的。

游戏 1:该模型通过 $\mathcal{A}_1$ 与 $\mathcal{C}_1$ 的博弈流程刻画。

系统准备阶段:由挑战者 $\mathcal{C}_1$ 执行此算法,将安全参数 λ 作为输入,运行 Setup 算法生成系统参数 params 和系统主密钥 msk ,同时 $\mathcal{C}_1$ 记录各预言机询问所产生的数据到相应的列表并对列表进行维护。 $\mathcal{C}_1$ 秘密留存 msk 并发送 params 至敌手 $\mathcal{A}_1$ 。

查询攻击阶段:敌手 $\mathcal{A}_1$ 通过自适应的方式向挑战者 $\mathcal{C}_1$ 发起多项式受限次查询。

(1)秘密值查询:当 $\mathcal{A}_1$ 选择用户身份 ID_i 问询其秘密值时, $\mathcal{C}_1$ 将 ID_i 对应的秘密值 x_i 返回给 $\mathcal{A}_1$ 。

(2)部分私钥查询: $\mathcal{A}_1$ 输入 ID_i 查询其部分私钥, $\mathcal{C}_1$ 模拟运行 Key-Generate 算法产生对应的部分私钥 psk_i 返回给 $\mathcal{A}_1$ 。

(3)公钥查询:敌手 $\mathcal{A}_1$ 根据用户 ID_i 向挑战者 $\mathcal{C}_1$ 请求公钥, $\mathcal{C}_1$ 运行 Key-Generate 算法生成 ID_i 的公钥 pk_i 并返回。

(4)公钥替换查询: $\mathcal{A}_1$ 选取用户 ID_i 和公钥 $(\mathrm{ppk}_i', \mathrm{upk}_i')$, $\mathcal{C}_1$ 用 $(\mathrm{ppk}_i', \mathrm{upk}_i')$ 更新 $(\mathrm{ppk}_i, \mathrm{upk}_i)$ 。

(5)协助者密钥查询: $\mathcal{A}_1$ 选择用户 ID_i 并请求其时间片段 t 的临时签名密钥, $\mathcal{C}_1$ 返回 sk_i 给 $\mathcal{A}_1$ 。

(6)签名查询:当 $\mathcal{A}_1$ 查询用户 ID_i 对消息 m_i 在时间片段 t 及当前时间戳 T_i 的签名时, $\mathcal{C}_1$ 运行签名生成算法产生 S_i 返回给 $\mathcal{A}_1$ 。

伪造阶段:完成查询后, $\mathcal{A}_1$ 伪造 $\{C_1^*, C_2^*, \cdots, C_n^*\}$ 及 $\{m_1^*, m_2^*, \cdots, m_n^*\}$ 所对应的一个聚合签名 S^* 。

$\mathcal{A}_1$ 赢得游戏 1 当且仅当满足下述前提:

(1)基于用户组及消息组伪造的聚合签名 S^* 是一个有效聚合签名。

(2) $\mathcal{A}_1$ 对用户组中的至少一个用户未执行过部分私钥查询,临时签名密钥查询和签名查询。

游戏 2:该模型通过 $\mathcal{A}_2$ 与 $\mathcal{C}_2$ 之间的博弈流程刻画。

系统准备阶段:同游戏 1 基本一致,区别在于 $\mathcal{C}_2$ 需将系统主密钥 msk 与系统参数 params 主密钥均发送至敌手 $\mathcal{A}_2$ 。

查询攻击阶段:敌手 $\mathcal{A}_2$ 通过自适应的方式向挑战者 $\mathcal{C}_2$ 发起多项式受限次查询。

(1)公钥查询:敌手 $\mathcal{A}_2$ 根据用户身份 ID_i 向挑战者 $\mathcal{C}_2$ 请求公钥,$\mathcal{C}_2$ 运行 Key-Generate 算法生成 ID_i 的公钥 pk_i 并返回给 $\mathcal{A}_2$。

(2)秘密值查询:当 $\mathcal{A}_2$ 选择用户 ID_i 问询其秘密值时,$\mathcal{C}_2$ 将用户 ID_i 对应的秘密值 x_i 返回给 $\mathcal{A}_2$。

(3)临时签名密钥查询:当 $\mathcal{A}_2$ 请求查询协助者密钥时,$\mathcal{C}_2$ 运行 Key-Generate 算法生成 hsk_0, hpk_0, hsk_1, hpk_1 返回给 $\mathcal{A}_2$。

(4)签名查询:当 $\mathcal{A}_2$ 查询用户 ID_i 对消息 m_i 在时间片段 t 及当前时间戳 T_i 的签名时,$\mathcal{C}_2$ 运行签名生成算法产生 S_i 返回给 $\mathcal{A}_2$。

伪造阶段:完成查询后,$\mathcal{A}_2$ 伪造 $\{C_1^*, C_2^*, \cdots, C_n^*\}$ 及 $\{m_1^*, m_2^*, \cdots, m_n^*\}$ 所对应的一个聚合签名 S^*。

$\mathcal{A}_2$ 赢得游戏 2 当且仅当满足下述前提:

(1)基于用户组及消息组伪造的聚合签名 S^* 是一个有效聚合签名。

(2) $\mathcal{A}_2$ 对用户组中的至少一个用户未执行过秘密值查询,协助者密钥查询和签名查询。

定义 1:如果任意的多项式有界敌手 $\mathcal{A}_1$、$\mathcal{A}_2$ 分别在游戏 1、游戏 2 中取胜的优势都是可忽略的,则称 CL-PKIAS 方案在适应性选择消息攻击下具有存在性不可伪造性(EUF-CMA)。

3　CL-PKIAS 方案构造

3.1　系统参数初始化(Setup)

输入 λ 作为系统默认的安全参数,KGC 根据安全参数 λ 选择一个大素数 p,同时选定一条构造于有限域 F_p 上的椭圆曲线 $E(F_p)$,然后在 $E(F_p)$ 上选择一个阶为 q 的基点 P,其中 q 为一个 160 bit 以上的大素数且 $q \mid p-1$, G 为基于椭圆曲线的满足加法运算规则的加法循环群, P 为群 G 的生成元,无穷远点 O 包含于 G。KGC 确定系统主密钥 $\mathrm{msk}: s \leftarrow_R Z_q^*$,从而系统公开密钥为 $\mathrm{mpk} = sP$,秘密留存系统主密钥 msk 并确保不被泄露。遵照需求选取哈希函数: $H_0: \{0,1\}^* \times G \rightarrow Z_q^*$, $H_0': G \rightarrow Z_q^*$, $H_1: \{0,1\}^* \times G^2 \rightarrow Z_q^*$, $H_2: \{0,1\}^* \times G \times Z_q \rightarrow Z_q^*$, $H_3: \{0,1\}^{*3} \times G^2 \rightarrow Z_q^*$, $H_4: \{0,1\}^{*2} \times G \times \{0,1\}^* \rightarrow Z_q^*$, $H_5: \{0,1\}^{*3} \times G \rightarrow Z_q^*$。最终,由 KGC 公布 $\mathrm{pramas} = \{P, p, q, \mathrm{mpk}, H_0, H_0', H_1, H_2, H_3, H_4, H_5\}$。

3.2　密钥生成(Key-Generate)

获取系统公开参数列表 params,将当前加入系统的用户实体记作 C_i,身份信息记为 ID_i。C_i 随机均匀选取秘密值 $x_i \in Z_q^*$,并计算 $X_i = x_i P$。于是,用户密钥分别为 $\mathrm{usk}_i = x_i$, $\mathrm{upk}_i = X_i$。

C_i 将其身份信息 ID_i, upk_i 进行公布,KGC 取随机数 $r_i \leftarrow_R Z_q^*$,计算 $R_i = r_i P$, $h_1^i = H_1(\mathrm{ID}_i, R_i, X_i)$ 及 $\mathrm{psk}_i^* = R_i + sh_1^i + H_0(\mathrm{ID}_i, sX_i)$。然后,通过公开信道将 $\mathrm{ppk}_i = R_i$, psk_i^* 发送至 C_i。

C_i 获取 KGC 产生的部分公私钥后,首先对 $H_1(\mathrm{ID}_i, R_i, X_i)$ 进行计算,然后判断下述等式 $\mathrm{psk}_i^* P = R_i + h_1^i \mathrm{mpk} + H_0(\mathrm{ID}_i, x_i \mathrm{mpk}) P$ 成立与否,如果成立,则计算出 $\mathrm{psk}_i = \mathrm{psk}_i^* - H_0'(\mathrm{ID}_i, x_i \mathrm{mpk})$,从而获得用户的完整公钥 $\mathrm{pk}_i = (\mathrm{upk}_i, \mathrm{ppk}_i)$。

KGC 随机均匀选取 $d, d_0 \in Z_q^*$,计算 $\mathrm{hpk} = dP$, $d_1 = d - d_0 (\mathrm{mod}\ q)$。令 d_0, d_1 分别为协助者 a_0, a_1 的私钥,记作 hsk_0, hsk_1,则两协助者的公钥各自为 $\mathrm{hpk}_0 = d_0 P$, $\mathrm{hpk}_1 = d_1 P$。同时,令 $f_i' = H_2(\mathrm{ID}_i, X_i, t)$ (t 表示时间片段), $\mathrm{sk}_{i,0}^* = f_i^0 d_0 + f_i^0 d_1 + H_0'(dX_i)$ 并发送给用户 C_i。由 C_i 自行计算初始临时签名密钥 $\mathrm{sk}_{i,0} = \mathrm{psk}_i + \mathrm{sk}_{i,0}^* - H_0'(x_i \mathrm{hpk})$。

3.3　密钥更新(Key-Update)

获取 params,用户 C_i 选择值 $y_i \leftarrow_R Z_q^*$,计算 $Y_i = y_i P$,并根据时间片段 t 选择对应的协助者 $j \in \{0,1\}$,计算 $U_{i,t,j}^* = \mathrm{hsk}_j (f_i' - f_i^{-1}) + H_0'(d_j Y_i)$ 作为更新信息,利用公开信道发送至 C_i,用户根据 $U_{i,t,j}^*$ 计算出更新信息 $U_{i,t,j} = U_{i,t,j}^* - H_0'(y_i \mathrm{hpk}_j)$,随后更新临时私钥 $\mathrm{sk}_{i,t} = \mathrm{sk}_{i,t-1} + U_{i,t,j} + U_{i,t,j'}$,其中 $(j \equiv t (\mathrm{mod}\ 2), j' \equiv t - 1 (\mathrm{mod}\ 2))$,从而输出 C_i 在 t 的完整临时签名密钥 $(\mathrm{usk}_i, \mathrm{sk}_{i,t})$。

3.4　签名生成(Sign-Generate)

输入 params,用户 C_i 使用时间片段 t 所对应的临时签名密钥对消息 $m_i \in \{0,1\}^*$ 签名。设有效时间戳为 T_i,生成随机数 $k_i \in Z_q^*$,计算 $V_i = k_i P$, $h_{1i} = H_3(m_i, \mathrm{ID}_i, T_i, X_i, V_i)$, $h_{2i} = H_4(m_i, \mathrm{ID}_i, X_i, T_i)$, $h_{3i} = H_5(m_i, \mathrm{ID}_i, T_i, \mathrm{mpk})$。于是,可以进一步计算 $U_i = k_i h_{1i} + x_i h_{2i} + \mathrm{sk}_{i,t} h_{3i}$,然后输出 C_i 对于 m_i 的签名 $S_i = (V_i, U_i)$。

3.5　签名聚合(Sign-Aggregate)

输入 params,任意用户均能充当聚合签名生成者对一组签名进行聚合。对于时间片段 t 内 $\{(m_i, T_i)\}_{i=1,2,\cdots,n}$ 对应的一组签名 $\{S_1, S_2, \cdots, S_n\}$,参与聚合的用户集合 $C = \{C_1, C_2, \cdots, C_n\}$,用户 C_i 的身份信息与公钥分别为 ID_i, pk_i,首先计算 h_{1i},进一步可以计算 $U = \sum_{i=1}^{n} U_i$, $V = \sum_{i=1}^{n} h_{1i} V_i$,最终生成聚合签名 $S = (U, V)$。

3.6 聚合验证(Aggregate-Verify)

给定 $\{(m_i,T_i)\}_{i=1,2,\cdots,n}$ 在时间片段 t 内产生的一个聚合签名 S,每个参与聚合签名用户的身份信息集合 $\{\mathrm{ID}_1,\mathrm{ID}_2,\cdots,\mathrm{ID}_n\}$ 及其对应的公钥集合 $\{\mathrm{pk}_1,\mathrm{pk}_2,\cdots,\mathrm{pk}_n\}$。确定在有效时间窗口内,进而计算 $h_1^i=H_1(\mathrm{ID}_i,R_i,X_i)$,$h_{2i}=H_4(m_i,\mathrm{ID}_i,X_i,T_i)$,$h_{3i}=H_5(m_i,ID_i,T_i,\mathrm{mpk})$,$f_i^t=H_2(\mathrm{ID}_i,X_i,t)$,通过 $UP=V+\sum_{i=1}^{n}h_{2i}X_i+\sum_{i=1}^{n}h_{3i}(R_i+h_1^i\mathrm{mpk}+f_i^t\mathrm{hpk})$ 成立与否判断签名的有效性。以下推导过程表明了该方案的正确性:

$$\begin{aligned}UP&=\sum_{i=1}^{n}U_iP=\sum_{i=1}^{n}(k_ih_{1i}+x_ih_{2i}+sk_{i,t}h_{3i})P=\\&\sum_{i=1}^{n}(k_ih_{1i}P+x_ih_{2i}P+sk_{i,t}h_{3i}P)=\\&\sum_{i=1}^{n}h_{1i}V_i+h_{2i}X_i+h_{3i}(r_iP+sh_1^iP+\\&f_i^td_0P+f_i^td_1P)=V+\sum_{i=1}^{n}h_{2i}X_i+\\&\sum_{i=1}^{n}h_{3i}(R_i+h_1^i\mathrm{mpk}+f_i^t\mathrm{hpk})\end{aligned}$$

4 安全性分析

在本节中将给出 CL-PKIAS 方案的 EUF-CMA 的安全性证明,利用反证法证明其结论,假定挑战者可给出 ECDLP 问题的有效解,则敌手能攻破该方案,然而已知实际情况下 ECDLP 问题是难解的,由逆否定理可知,该方案是安全的,具体证明过程如下所叙:

定理 1:CL-PKIAS 方案的 EUF-CMA 安全性如果被任何 PPT 敌手以不可忽略的优势 ε 攻破,$\mathcal{A}_1$ 至多可以执行 q_{H_i} 次(其中 $i=0,1,\cdots,5$)H_i 预言查询,q_P 次部分私钥查询,q_{PK} 次公钥查询,q_V 次秘密值查询,q_{RP} 次公钥替换查询,q_{SK} 次协助者密钥查询,q_S 次签名查询,则总有挑战者 $\mathcal{C}_1$ 在多项式时间内以至少的概率 ε' 解决 ECDLP 问题。

证明:假设 ECDLP 问题的挑战者为 $\mathcal{C}_1$,群上的一个 ECDLP 问题实例 $(P,\mathrm{mpk}=aP)$ 是其已知信息。$\mathcal{C}_1$ 将敌手 $\mathcal{A}_1$ 作为子程序,目标是利用敌手 $\mathcal{A}_1$ 的攻击能力计算出 a。

系统准备阶段:$\mathcal{C}_1$ 运行 Setup 算法生成系统公开参数列表 params 发送给 $\mathcal{A}_1$,同时建立各个列表并初始化为空,$\mathcal{C}_1$ 维护及更新对应列表来应答 $\mathcal{A}_1$ 的适性多项式有界次查询。

查询攻击阶段:由 $\mathcal{A}_1$ 执行各散列查询与其他预言机查询。

H_0 查询:若 $\mathcal{A}_1$ 可以随时向 H_0 预言机请求关于 (ID_i,x_isP) 的查询至多 q_{H_0} 次,$\mathcal{C}_1$ 维护元素为 $(\mathrm{ID}_i,x_isP,j_i)$ 的列表 L_0 以应答 $\mathcal{A}_1$。针对每次查询,$\mathcal{C}_1$ 如果从 L_0 中查找到元素 (ID_i,x_isP,j_i),直接将已有的 j_i 返回给 $\mathcal{A}_1$;否则,随机选取 $j_i\in Z_q^*$ 响应 $\mathcal{A}_1$,并将新项更新至 L_0。

H_0' 查询:如果 $\mathcal{A}_1$ 能够随时通过输入 $Q_i\in(x_idP,y_id_jP)$ 对 H_0' 预言机作至多 $q_{H_0'}$ 次查询,且 $\mathcal{C}_1$ 可通过维护列表 L_0' 对 $\mathcal{A}_1$ 的查询给出应答。$\mathcal{C}_1$ 若能从 L_0' 中查找到对应元素 Q_i,则直接将预先存在的 l_i 返回给 $\mathcal{A}_1$;否则,随机选取 $l_i\in Z_q^*$,以其响应 $\mathcal{A}_1$,并将新元组 (Q_i,l_i) 记录至 L_0'。

H_1 查询:若 $\mathcal{A}_1$ 能随时向 H_1 预言机请求关于 (ID_i,X_i,R_i) 的查询至多 q_{H_1} 次,$\mathcal{C}_1$ 维护元素为 $(\mathrm{ID}_i,X_i,R_i,h_1^i,c_i)$ 的列表 L_1 以应答 $\mathcal{A}_1$。针对每次查询 (ID_i,X_i,R_i),$\mathcal{C}_1$ 如果从 L_1 查找到对应元素,用已定义的 h_1^i 作为对 $\mathcal{A}_1$ 的应答;否则,任选 $n^*\in Z_q^*$。掷硬币选取 $c_i\in\{0,1\}$,设 $\Pr[c_i=0]=\delta$。若 $c_i=0$,则 $h_1^i=n^*$;反之,选择随机数 $h_1^i\in_R Z_q^*$,最终以 h_1^i 应答 $\mathcal{A}_1$,更新并保存新项至 L_1。

H_2 查询:若 $\mathcal{A}_1$ 能随时向预言机请求关于 (ID_i,X_i,t) 的查询至多 q_{H_2} 次,$\mathcal{C}_1$ 维护元素形为 $(\mathrm{ID}_i,X_i,t,f_i)$ 的列表 L_1 以应答 $\mathcal{A}_1$。针对每次查询,$\mathcal{C}_1$ 如果能从 L_2 中查找到元素 $(\mathrm{ID}_i,X_i,t,f_i)$,则将预先定义的 f_i 返回给 $\mathcal{A}_1$;否则,将 $f_i\in Z_q^*$ 作为对 $\mathcal{A}_1$ 的应答,并保存新项 (ID_i,X_i,t,f_i) 至 L_2。

H_3 查询:若 $\mathcal{A}_1$ 可随时发出查询请求从而查询 $H_3(m_i,\mathrm{ID}_i,T_i,X_i,V_i)$ 至多 q_{H_3} 次,$\mathcal{C}_1$ 负责维护元素为 $(m_i,\mathrm{ID}_i,T_i,X_i,V_i,h_{1i})$ 的列表 L_3。针对每次查询 $(m_i,\mathrm{ID}_i,T_i,X_i,V_i)$,$\mathcal{C}_1$ 如果能从 L_3 中发现相应元素,则将已定义的 h_{1i} 返回给 $\mathcal{A}_1$;否则,将 $h_{1i}\in Z_q^*$ 应答 $\mathcal{A}_1$,并添加新项 $(m_i,\mathrm{ID}_i,T_i,X_i,V_i,h_{1i})$ 至 L_3。

H_4 查询:若 $\mathcal{A}_1$ 能随时向 H_4 预言机请求关于 (m_i,ID_i,X_i,T_i) 的查询至多 q_{H_4} 次,$\mathcal{C}_1$ 维护元素为 $(m_i,\mathrm{ID}_i,X_i,T_i,h_{2i})$ 的列表 L_4 以应答 $\mathcal{A}_1$。针对每次查询 $(m_i,\mathrm{ID}_i,X_i,T_i)$,$\mathcal{C}_1$ 若从 L_4 中查找到相应元素,则将已定义的 h_{2i} 返回给 $\mathcal{A}_1$;否则,将 $h_{2i}\in Z_q^*$ 作为对 $\mathcal{A}_1$ 的应答,并添加新项 $(m_i,\mathrm{ID}_i,X_i,T_i,h_{2i})$ 至 L_4。

H_5 查询:若 $\mathcal{A}_1$ 能随时向 H_5 预言机请求关于 $(m_i,\mathrm{ID}_i,T_i,\mathrm{mpk})$ 的查询至多 q_{H_5} 次,$\mathcal{C}_1$ 维护元素为 $(m_i,\mathrm{ID}_i,T_i,\mathrm{mpk},h_{3i})$ 的列表 L_5 以应答 $\mathcal{A}_1$。针对每次查询 $(m_i,\mathrm{ID}_i,X_i,T_i)$,$\mathcal{C}_1$ 若从 L_5 中查找到相应元素,则以已定义的 h_{3i} 应答 $\mathcal{A}_1$;否则,将 $h_{3i}\in Z_q^*$ 作为对 $\mathcal{A}_1$ 的应答并添加新项 $(m_i,\mathrm{ID}_i,T_i,\mathrm{mpk},h_{3i})$ 至 L_5。

秘密值查询:若 $\mathcal{A}_1$ 可随时执行该查询至多 q_V 次,$\mathcal{C}_1$ 维护 L_V 应答关于 ID_i 的秘密值询问,若在 L_V 上搜寻到相应元组 (ID_i,x_i,X_i),则返回给 $\mathcal{A}_1$;否则,执行 H_1 查询,若 $c_i=0$,选取 $x_i\leftarrow_R Z_q^*$,计算 $X_i=x_iP$;反之,$\mathcal{C}_1$

中断,在列表 L_1,L_V 中分别更新元素,并以 x_i 应答 $\mathcal{A}_1$。

部分私钥查询:若 $\mathcal{A}_1$ 可随时向 $\mathcal{C}_1$ 请求关于身份 ID_i 的部分私钥查询至多 q_P 次,对于每次查询,如果能在形如 $(\mathrm{ID}_i, X_i, R_i, \mathrm{psk}_i, c_i)$ 的 L_P 上搜寻到相关项,则返回。否则,执行秘密值查询获取相应的 X_i,同时从 L_1 上搜寻 ID_i 对应元组,若 $c_i = 1$,选取随机数 $\mathrm{psk}_i \in Z_q^*$,$R_i = \mathrm{psk}_i P - h_1^i \mathrm{mpk}$;若 $c_i = 0$,$\mathrm{psk}_i = \perp$,取 $r^* \in_R Z_q^*$,并计算 $R_i = r^* P$,以 psk_i 应答 $\mathcal{A}_1$,并在 L_P,L_1 中插入新元素。

公钥查询:$\mathcal{C}_1$ 构建元组为 $(\mathrm{ID}_i, X_i, R_i)$ 的列表 L_{PK}。$\mathcal{A}_1$ 向 $\mathcal{C}_1$ 提出关于 ID_i 的公钥查询,若 L_{PK} 有对应项,则应答 $\mathcal{A}_1$;否则,查询对应 (X_i, R_i) 应答 $\mathcal{A}_1$ 并保存至 L_{PK}。

公钥替换查询:$\mathcal{A}_1$ 能选择 (X_i', R_i') 替换 ID_i 对应的公钥 (X_i, R_i),$\mathcal{C}_1$ 若发现 L_P 中有 ID_i 对应元组 $(\mathrm{ID}_i, X_i, R_i, \mathrm{psk}_i, c_i)$,首先令 $x_i = \perp$,将 L_P 上的元组替换为 $(\mathrm{ID}_i, X_i', R_i', \perp, c_i)$,并更新列表 L_V。

协助者密钥查询:若 $\mathcal{A}_1$ 输入 $(\mathrm{ID}_i, t, x_i, \mathrm{sk}_{i,t})$ 向 $\mathcal{C}_1$ 提出临时签名密钥查询请求,$\mathcal{C}_1$ 首先检查列表 L_{SK} 上是否已有对应元素,若未找到,$\mathcal{C}_1$ 随机选取 $d, d_0 \in Z_q^*$,则 $\mathrm{hpk} = dP$,$d_1 = d - d_0 (\mathrm{mod}\ q)$,并令 $\mathrm{hsk}_0 = d_0$,$\mathrm{hsk}_1 = d_1$,从而 $\mathrm{hpk}_0 = d_0 P$,$\mathrm{hpk}_1 = d_1 P$,并将其添加至 L_{SK}。

签名查询:若 $\mathcal{A}_1$ 输入 $(\mathrm{ID}_i, X_i, R_i, t)$ 作签名查询,$\mathcal{C}_1$ 在 L_P 上搜寻 ID_i 对应的项。如果 $c_i = 1$,直接执行 Sign-Generate 算法正常生成签名并返回给 $\mathcal{A}_1$;否则,$\mathcal{C}_1$ 随机均匀地选取 $U_i \in Z_q^*$,可进一步计算出 $V_i = h_{1i}^{-1}(U_i P - h_{2i} X_i - h_{3i}(R_i + n^* \mathrm{mpk} + f_i \mathrm{hpk}))$。进而以 (V_i, U_i) 响应 $\mathcal{A}_1$。

伪造阶段:$\mathcal{A}_1$ 的有界次查询请求得到应答后,可基于 $(m_i^*, \mathrm{ID}_i^*)_{i=1,2,\cdots,n}$ 和 ID_i^* 对应的公钥 (X_i^*, R_i^*) 伪造出一个聚合签名 (V^*, U^*)。若满足下述前提,首先签名有效;其次,存在一个 $i \in [1,n]$ 使得 $c_i = 0$,不妨假设 $i = 1$,且 ID_1^* 未对部分私钥预言机及签名预言机发起过查询请求,则 $\mathcal{C}_1$ 可根据验证等式 $U^* P = V^* + \sum_{i=1}^{n} h_{2i}^* X_i^* + \sum_{i=1}^{n} h_{3i}^* (R_i^* + (h_1^i)^* \mathrm{mpk} + f_i^* \mathrm{hpk})$ 输出 ECDLP 问题的有效解 $aP = (h_{3i}^* n^*)^{-1} \{U^* P - (V^* + \sum_{i=1}^{n} (h_{2i}^* X_i^* + h_{3i}^* f_i^* \mathrm{hpk}) + \sum_{i=2}^{n} h_{3i}^* (R_i^* + (h_1^i)^* \mathrm{mpk})) - h_{31}^* r^* P\}$;否则,$\mathcal{C}_1$ 未能解决 ECDLP 问题。

如果事件 E_1 表示 $\mathcal{A}_1$ 未针对 ID_1^* 提出过部分私钥查询,则 $\Pr[E_1] = \frac{1}{n}(1 - \frac{q_p}{2^k})$;

事件 E_2 表示 $\mathcal{C}_1$ 不中断 $\mathcal{A}_1$ 提出的任意签名查询,则 $\Pr[E_2] = (1-\delta)^{q_V + q_{RP} + q_S}$;

事件 E_3 指即使 $\mathcal{A}_1$ 伪造阶段能伪造出含 ID_1^* 有效聚合签名,$\mathcal{C}_1$ 仍不中断,则 $\Pr[E_3] = 1/q_s + n$。

若 $\mathcal{C}_1$ 在查询及伪造阶段均没有中断,则 $\mathcal{A}_1$ 无法区分 $\mathcal{C}_1$ 在模拟环境与实际攻击环境中的差异。换言之,如果敌手 $\mathcal{A}_1$ 能够以不可忽略的优势 ε 攻破 CL-PKIAS 方案,且 $\mathcal{C}_1$ 的模拟在 $\mathcal{A}_1$ 视图下是完备的,那么 $\mathcal{C}_1$ 就能以概率 ε' 输出 ECDLP 问题的有效解。

$$\varepsilon' = \frac{\varepsilon}{n}(1 - \frac{q_p}{2^k})(1-\delta)^{q_V + q_{RP} + q_S} \frac{1}{q_s + n} =$$

$$(1 - \frac{q_p}{2^k})(1-\delta)^{q_V + q_{RP} + q_S}(\frac{\varepsilon}{n(q_s + n)})$$

定理 2:CL-PKIAS 方案的 EUF-CMA 安全性如果被任何 PPT 敌手 $\mathcal{A}_2$ 以不可忽略的优势 ε 攻破,$\mathcal{A}_2$ 至多可以执行 q_{H_i} 次(其中 $i = 0, 1, \cdots, 5$)H_i 预言查询,q_{PK} 次公钥查询,q_V 次秘密值查询,q_{SK} 次临时签名密钥查询,q_S 次签名查询,则总有挑战者 $\mathcal{C}_2$ 在多项式时间内以至少 ε' 的概率解决 ECDLP 问题。

证明:假设 ECDLP 问题的挑战者为 $\mathcal{C}_2$,群上的一个 ECDLP 问题实例 (P, bP) 是其已知信息。$\mathcal{C}_2$ 将敌手 $\mathcal{A}_2$ 作为子程序,目标是利用敌手 $\mathcal{A}_2$ 的攻击能力计算出 b。

系统准备阶段:$\mathcal{C}_2$ 运行 Setup 算法生成 params 及 $\mathrm{msk} = s$ 发送给 $\mathcal{A}_2$,其中 $s \in_R Z_q^*$ 且 $\mathrm{mpk} = sP$。将 L_0,L_0',L_1,L_2,L_3,L_4,L_5,L_{PK},L_V,L_{SK} 初始化为空,$\mathcal{C}_2$ 维护对应列表应答 $\mathcal{A}_2$ 的适性多项式有界次查询。

查询攻击阶段:除 H_1 查询,公钥查询,临时签名密钥查询外,$\mathcal{A}_2$ 其他预言查询的执行过程与定理 1 一致。

H_1 查询:若 $\mathcal{A}_2$ 能随时向 H_1 预言机请求关于 $(\mathrm{ID}_i, X_i, R_i)$ 的查询至多 q_{H_1} 次,$\mathcal{C}_2$ 维护元素为 $(\mathrm{ID}_i, X_i, R_i, h_1^i)$ 的列表 L_1 以应答 $\mathcal{A}_2$。针对每次查询 $(\mathrm{ID}_i, X_i, R_i)$,$\mathcal{C}_2$ 如果从 L_1 查找到已定义的对应元素,则直接以其回应 $\mathcal{A}_2$;否则,随机选取 $h_1^i \in_R Z_q^*$ 响应 $\mathcal{A}_2$,并把新项 $(\mathrm{ID}_i, X_i, R_i, h_1^i)$ 保存至 L_1。

公钥查询:$\mathcal{C}_2$ 构建元组为 $(\mathrm{ID}_i, X_i, R_i, c_i)$ 的列表 L_{PK} 并进行维护。$\mathcal{A}_2$ 可随时通过输入 ID_i 向 $\mathcal{C}_2$ 进行公钥查询,若 L_{PK} 或 L_1 有对应项,则应答 $\mathcal{A}_2$;否则,选取 $\mathrm{psk}_i, x_i \in_R Z_q^*$,计算 $R_i = \mathrm{psk}_i P - h_1^i \mathrm{mpk}$。掷硬币随机选取 $c_i \leftarrow \{0,1\}$,设 $\Pr[c_i = 0] = \delta$。若 $c_i = 0$,则 $X_i = bP$;反之,$X_i = x_i P$。以 (X_i, R_i) 作为对 $\mathcal{A}_2$ 的应答,在 L_1,L_{PK} 中更新对应元素。

秘密值查询:若 $\mathcal{A}_2$ 可随时执行该查询至多 q_V 次,$\mathcal{C}_2$ 维护 L_V 应答关于 ID_i 的秘密值询问;若在 L_V 上搜寻到相应元组 $(\mathrm{ID}_i, x_i, X_i, c_i)$,则返回给 $\mathcal{A}_2$;否则,从 L_{PK} 搜寻 ID_i 对应项,若 $c_i = 0$,则 $x_i = \perp$。若 $c_i = 1$,返回对

应元素。在列表 L_1,L_V 中分别存储新元素,并以 x_i 应答 $\mathcal{A}_2$。

临时签名密钥查询:若 $\mathcal{A}_2$ 输入 $(\mathrm{ID}_i,t,x_i,\mathrm{sk}_{i,t})$ 向 $\mathcal{C}_2$ 发起临时签名密钥查询,$\mathcal{C}_2$ 首先搜寻列表 L_{SK} 若已存在对应元素,直接以其作为响应,否则,$\mathcal{C}_2$ 选取 $d,d_0 \in Z_q^*$,则 $\mathrm{hpk}=dP$,$d_1=d-d_0(\mathrm{mod}\ q)$,并令 $\mathrm{hsk}_0=d_0$,$\mathrm{hsk}_1=d_1$,从而 $\mathrm{hpk}_0=d_0P$,$\mathrm{hpk}_1=d_1P$,并将其记录至 L_{SK}。然后,在列表 L_{PK},L_2 中分别查询对应的 psk_i,x_i,f_i。计算 $\mathrm{sk}_{i,t}=\mathrm{psk}_i+f_i\mathrm{hsk}_0+f_i\mathrm{hsk}_1$,如果 $c_i=0$,则 $\mathcal{C}_2$ 中断模拟;否则,返回 $(\mathrm{sk}_{i,t},x_i)$。

签名查询:若 $\mathcal{A}_2$ 输入 $(\mathrm{ID}_i,X_i,R_i,t)$ 作签名查询,$\mathcal{C}_2$ 在 L_P 上搜寻 ID_i 对应的项。如果 $c_i=1$,直接执行 Sign-Generate 算法正常生成签名并返回给 $\mathcal{A}_2$;否则,$\mathcal{C}_2$ 随机均匀地选取 $U_i \in Z_q^*$,可进一步计算出 $V_i=h_{1i}^{-1}(U_iP-h_{2i}(bP)-h_{3i}(R_i+h_1^i\mathrm{mpk}+f_ihpk)$。进而以 (V_i,U_i) 响应 $\mathcal{A}_2$。

伪造阶段:$\mathcal{A}_2$ 的有界次查询请求得到应答后,可基于 $(m_i^*,\mathrm{ID}_i^*)_{i=1,2,\cdots,n}$ 和 ID_i^* 对应的公钥 (X_i^*,R_i^*) 伪造出一个聚合签名 (V^*,U^*)。若满足下述前提,首先签名有效;其次,存在一个 $i \in [1,n]$ 使得 $c_i=0$,不妨假设 $i=1$,并且 ID_1^* 未对秘密值预言机及签名预言机发起过查询请求,则 $\mathcal{C}_2$ 通过验证等式 $U^*P=V^*+\sum_{i=1}^{n}h_{2i}^*X_i^*+\sum_{i=1}^{n}h_{3i}^*(R_i^*+(h_1^i)^*\mathrm{mpk}+f_i^*\mathrm{hpk})$ 输出 ECDLP 问题的有效解 $bP=(h_{21}^*)^{-1}\{U^*P-(V^*+\sum_{i=2}^{n}h_{2i}^*X_i^*+\sum_{i=1}^{n}h_{3i}^*(R_i^*+(h_1^i)^*\mathrm{mpk}+f_i^*\mathrm{hpk}))$。否则,$\mathcal{C}_2$ 未能解决 ECDLP 问题。

如果事件 E_1 表示 $\mathcal{A}_2$ 未针对 ID_1^* 提出过部分私钥查询,则 $\Pr[E_1]=\frac{1}{n}(1-\frac{q_V}{2^k})$;

事件 E_2 表示 $\mathcal{C}_2$ 不中断 $\mathcal{A}_2$ 提出的任意临时签名密钥查询,则 $\Pr[E_2]=(1-\delta)^{q_{\mathrm{PK}}+q_{\mathrm{SK}}}$;

事件 E_3 表示 $\mathcal{C}_2$ 不中断 $\mathcal{A}_2$ 提出的任意签名查询,则 $\Pr[E_3]=(1-\delta)^{q_S}$;

事件 E_4 指即使 $\mathcal{A}_2$ 伪造阶段能伪造出含 ID_1^* 有效聚合签名,$\mathcal{C}_2$ 仍不中断,则 $\Pr[E_4]=1/q_s+n$。

同定理 1,$\mathcal{C}_2$ 解决 ECDLP 问题的优势 ε' 为:

$$\varepsilon'=\frac{\varepsilon}{n}(1-\frac{q_p}{2^k})(1-\delta)^{q_{\mathrm{PK}}+q_{\mathrm{SK}}+q_S}\frac{1}{q_s+1}=(1-\frac{q_p}{2^k})(1-\delta)^{q_{\mathrm{PK}}+q_{\mathrm{SK}}+q_S}(\frac{\varepsilon}{n(q_s+n)})$$

证毕。

综上可得,无法构造出以不可忽略的优势解决 ECDLP 困难问题的算法,故该方案的 EUF-CMA 安全性不能为 $\mathcal{A}_1$,$\mathcal{A}_2$ 所攻破。

5 性能分析

为对该方案的性能进行准确评估,在配置为 Intel(R)Core(TM)i5-6200 CPU @ 2.40 GHz,内存 4 GB,Windows10 64 位操作系统的实验环境下利用 JPBC 库为基础进行实验对照分析。该文选取的密码参数均与密钥长度为 1 024 比特的 RSA 体制保持同等安全,以实现相同指标下的性能比对分析。基于双线性映射的密码操作所使用的曲线是嵌入度为 2 的超奇异椭圆曲线 $E(F_p):y^2=x^3+x$,其中 q 是不少于 512 比特的素数,$p=2^{159}+2^{17}+1$ 是 160 比特的 Solinas 素数,p,q 满足 $q+1=12pr$,G_1 表示基于该曲线的 q 阶双线性群,群元素长度为 128 字节,记作 $|G_1|$。选取 Koblitz 曲线实现椭圆曲线加法循环群上密码运算操作,其在有限域 $F_{2^{163}}$ 上形为 $y^2=x^3+ax^2+b$,其中 p,q 为 160 比特的素数,b 为 163 比特的随机数且 $a=1$,G 表示基于该曲线的 q 阶加法循环群,群元素长度为 40 字节,记作 $|G|$。表 1 定义了密码操作符号,并通过运行 1 000 次实验取其平均值的方式给出了单次密码运算的所用耗时。

表 1 单次密码操作的执行时间

符号	含义	运行时间/ms
T_{bp}	双线性对运算	4.411 7
T_{bm}	基于双线性对的乘法运算	2.703 9
T_{ba}	基于双线性对的加法运算	0.017 1
T_{em}	椭圆曲线标量乘法运算	0.732 5
T_{ea}	椭圆曲线加法运算	0.004 8
T_H	映射到点的散列函数运算	2.403 2

表 2 从理论上分析了各方案的计算性能。文献[8]方案中,聚合签名开销为 $2n$ 个基于双线性群的乘法运算 T_{bm},$4n$ 个基于双线性群的加法运算 T_{ba},一个映射到点的散列函数运算 T_H;聚合验证成本为三个双线性运算,n 个基于双线性群的乘法运算 T_{bm},$2n-1$ 个基于双线性群的加法运算 T_{ba},两个映射到点的散列函数运算 T_H。文献[11]方案中,聚合签名开销为 $2n$ 个基于 ECC 的倍点运算 T_{em},$3n$ 个基于 ECC 的加

法运算 T_{ea};聚合验证成本为 $2n$ 个基于 ECC 的倍点运算 T_{em},$3(n-1)$ 个基于 ECC 的加法运算 T_{ea}。文献[16]方案中,聚合签名开销为 $3n$ 个基于双线性群的乘法运算 T_{bm},$2n$ 个基于双线性群的加法运算 T_{ba},一个映射到点的散列函数运算 T_H;聚合验证成本为四个双线性运算,$2n$ 个基于双线性群的乘法运算 T_{bm} 及加法运算 T_{ba},一个映射到点的散列函数运算 T_H。文中方案聚合签名开销为 n 个基于 ECC 的倍点运算 T_{em},$2n$ 个基于 ECC 的加法运算 T_{ea};聚合验证代价为 $3n+1$ 个基于 ECC 的倍点运算 T_{em},$2n-1$ 个基于 ECC 的加法运算 T_{ea}。

表 2　各方案的计算开销对比

方案	聚合签名	聚合验证
文献[8]	$2nT_{bm}+4nT_{ba}+T_H$	$3T_{bp}+nT_{bm}+(2n-1)T_{ba}+2T_H$
文献[11]	$2nT_{em}+3nT_{ea}$	$2nT_{em}+3(n-1)T_{ea}$
文献[16]	$3nT_{bm}+2nT_{ba}+T_H$	$4T_{bp}+2n(T_{bm}+T_{ba})+T_H$
文中方案	$nT_{em}+2nT_{ea}$	$(3n+1)T_{em}+(2n-1)T_{ea}$

利用表 1 中的实验数据,针对文中方案及文献[8,11,16]方案,采用消息数目递增的方式进行实验,实验性能对比如图 1、2 所示。文中方案在聚合签名阶段的计算性能优于其他三个方案,与它耗时最接近的文献[11]方案相比,其效率提升了大约 20%;在聚合验证阶段,文中方案的运行效率优于有双线性映射运算的方案,相较于文献[11]方案效率偏低,然而,文中方案是在牺牲效率的前提下增加了并行密钥隔离功能,在一些应用场景中能够有效降低密钥泄露的威胁,因此这种时间消耗是能够被接受的。

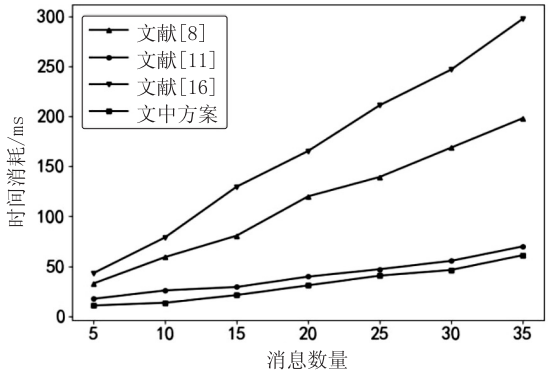

图 1　聚合签名算法执行时间对比

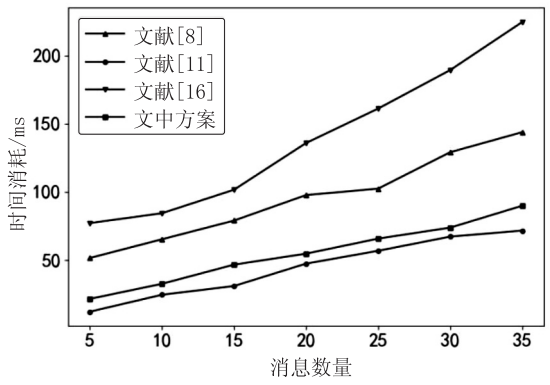

图 2　聚合验证算法执行时间对比

表 3 列出了各方案的功能对比及通信开销,文献[8]方案中,聚合签名所占字节长度为 $128(n+1)$ byte,其签名长度与参与签名人数相关,文献[11]方案中,聚合签名所占字节长度保持在 80 byte,但以上两个方案均未考虑密钥泄露问题。文献[16]方案中,聚合签名长度固定为 256 byte,并且引入密钥隔离机制以保障签名方案的前向及后向安全。文中方案的聚合签名长度为固定的 60 byte,进一步加入并行密钥隔离机制确保密钥频繁更新的安全性,同时在生成部分私钥时,不需要构建安全信道就能实现与协助者及 KGC 的交互,增强了方案的实用性。

表 3　各方案的通信开销及功能对比

方案	签名长度	密钥隔离	实现方式	困难问题
文献[8]	$(n+1)\lvert G_1\rvert$	否	双线性对	CDH
文献[11]	$2\lvert G\rvert$	否	椭圆曲线	CDH
文献[16]	$2\lvert G_1\rvert$	是	双线性对	CDH
文中方案	$\lvert G\rvert+\lvert Z_q^*\rvert$	是	椭圆曲线	ECDLP

6　结束语

该文探讨了密钥泄露对无证书聚合签名安全性的威胁,并针对性地将并行密钥隔离机制引入到方案的密钥更新阶段,以便能够动态地对临时签名密钥进行更新,并利用椭圆曲线密码体制构造了 CL-PKIAS,利用 ECDLP 问题的困难性在随机预言模型下给出了该方案的安全性分析。由性能分析可知,该方案在满足安全要求的前提下具有更低的通信及运算成本。

值得注意的是,该方案实现了数据的认证性,数据的机密性并未考虑。下一步需要考虑的是如何将其扩展到无证书聚合签密,以提高在分布式群组安全通信场景中的适用性。

参考文献:

[1] BONEH D, GENTRY C, LYNN B, et al. Aggregate and verifiably encrypted signatures from bilinear maps[C]// Proceeding of the EUROCRYPT 2003. Warsaw: Springer,

2003:416-432.
[2] BONEH D, LYNN B, SHACHAM H. Short signatures from the weil pairing[C]//Proceeding of the ASIACRYPT 2001. Gold Coas: Springer, 2001:514-532.
[3] AL-RIYAMI S S, PATERSON K G. Certificateless public key cryptography [C]//Proceeding of the ASIACRYPT 2003. Taiwan, China: Springer, 2003:452-473.
[4] HUANG L X, ZHOU J L, ZHANG G X, et al. Certificateless public verification for the outsourced data integrity in cloud storage [J]. Journal of Circuits, Systems and Computers, 2018, 27(11):1850181.
[5] DU H Z, WEN Q Y, ZHANG S S, et al. A new provably secure certificateless signature scheme for Internet of Things [J]. Ad Hoc Networks, 2020, 100:102074.
[6] ELKHALIL A, ELHABOB R, ELTAYIEB N. An efficient signcryption of heterogeneous systems for internet of vehicles [J]. Journal of Systems Architecture, 2021, 113:101885.
[7] 陈 虎,魏仕民,朱昌杰,等. 安全的无证书聚合签名方案[J]. 软件学报, 2015, 26(5):1173-1180.
[8] KUMAR P, KUMARI S, SHARMA V, et al. A certificateless aggregate signature scheme for healthcare wireless sensor network[J]. Sustainable Computing: Informatics and Systems, 2018, 18:80-89.
[9] KUMAR P, KUMARI S, SHARMA V, et al. Secure CLS and CLAS schemes designed for VANETs[J]. The Journal of Supercomputing, 2019, 75(6):3076-3098.
[10] ZHAO Y, HOU Y, WANG L, et al. An efficient certificateless aggregate signature scheme for the Internet of Vehicles [J]. Transactions on Emerging Telecommunica-tions Technologies, 2020, 31(5):1-20.
[11] LIU J, WANG L, YU Y. Improved security of a pairing-free certificateless aggregate signature in healthcare wireless medical sensor networks[J]. IEEE Internet of Things Journal, 2020, 7(6):5256-5266.
[12] 韦性佳,张京花,刘增芳,等. 具有前向安全性质的基于身份的聚合签名方案[J]. 计算机科学, 2018, 45(6):387-391.
[13] KIM J, OH H. FAS: forward secure sequential aggregate signatures for secure logging[J]. Information Sciences, 2019, 471:115-131.
[14] DODIS Y, KATZ J, XU S, et al. Key-insulated public key cryptosystems[C]//International conference on the theory and applications of cryptographic techniques. Amsterdam: Springer, 2002:65-82.
[15] XIONG H, MEI Q, ZHAO Y. Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments[J]. IEEE Systems Journal, 2019, 14(1):310-320.
[16] 寻甜甜,于 佳,杨光洋,等. 密钥隔离的无证书聚合签名[J]. 电子学报, 2016, 44(5):1111-1116.
[17] XIONG H, HOU Y Z, HUANG X, et al. Certificate-based parallel key-insulated aggregate signature against fully chosen key attacks for industrial internet of things[J]. IEEE Internet of Things Journal, 2021, 8(11):8935-8948.

(上接第 99 页)

[8] 陈金广,李 洁,高新波. 一种迭代收缩非线性状态约束滤波算法[J]. 西安电子科技大学学报, 2011, 38(1):104-109.
[9] WASIM M, ALI A. Airship aerodynamic model estimation using unscented Kalman filter[J]. Systems Engineering and Electronics, 2020, 31(6):1318-1329.
[10] WANG Hongwei, ZHANG Wei, ZUO Junyi, et al. Gen-eralized cubature quadrature Kalman filters: derivations and extensions[J]. Systems Engineering and Electronics, 2017, 28(3):556-562.
[11] 张小利,王玥童,夏金松,等. 基于改进的 CDKF 锂电池 SOC 估计方法[J]. 储能科学与技术, 2021, 10(4):1454-1462.
[12] GUO Shiluo, SUN Yingjie, CHANG Limin, et al. Robust cubature Kalman filter method for the nonlinear alignment of SINS[J]. Defence Technology, 2021, 17(4):593-598.
[13] SARKKA S. Bayesian filtering and smoothing[M]. London: Cambridge University Press, 2013.
[14] SARKKA S, NUMMENMAA A. Recursive noise adaptive Kalman filtering by variational Bayesian approximations[J]. IEEE Transactions on Automatic Control, 2009, 54(3):596-600.
[15] ALOUANI A T, BLAIR W D. Use of a kinematic constraint in tracking constant speed, maneuvering targets [J]. IEEE Transactions on Automatic Control, 1993, 38(7): 1107-1111.
[16] WANG L, CHIANG Y, CHANG F. Filtering method for nonlinear systems with constraints[J]. IEEE Proceeding of Control Theory Application, 2002, 149(6):525-531.
[17] MASSIGNAN J A D, LONDON JR J O B, MIRANDA V. Tracking power system state evolution with maximum-correntropy-based extended Kalman filter[J]. Journal of Modern Power Systems and Clean Energy, 2020, 8(4):616-625.