

支持批量审计的解密外包 Twin-SM9 密钥封装机制

刘 宽¹, 宁建廷^{1,3}, 伍 玮², 陈海霞²

¹ 福建师范大学 计算机与网络空间安全学院 福州 中国 350117

² 福建师范大学 数学与统计学院 福州 中国 350117

³ 中国科学院信息工程研究所 信息安全国家重点实验室 北京 中国 100093

摘要 自 SM9 标识密码相关算法先后被纳入 ISO/IEC 国际标准以来, 为推动密码技术实现安全先进、自主可控, 一系列关于 SM9 标识密码算法的功能性拓展和安全性(拓展)证明被提出。Cheng 依据 Gap- q -BCAA1 困难问题假设对 SM9 密钥封装、公钥加密和密钥协商系列算法进行了安全性分析。为有效消除 SM9 系列算法对 Gap 困难问题的依附, Lai 等人随后利用 Twin-Hash-ElGamal 技术构造出了 Twin-SM9 密钥封装机制。然而, Twin-SM9 密钥封装机制的解密操作需要 2 次双线性配对运算, 在需要对海量数据进行频繁解密操作且算力资源受限的环境中(如无线传感设备、密码芯片等), 计算代价高昂的配对运算将会成为制约系统效率的重要瓶颈。针对上述问题, 本文基于 Twin-SM9 提出了支持多密文批量审计的解密外包新型密钥封装机制 BAOC-Twin-SM9, 并在随机谰言模型下证明了 BAOC-Twin-SM9 具备 Replayable Chosen Ciphertext Attacks (RCCA) 安全性。BAOC-Twin-SM9 利用云服务中心的强大算力有效消除了双线性配对运算对原 Twin-SM9 密钥封装机制解密效率的影响, 计算资源有限的终端数据使用者最终只需进行两次简单的指数运算就能对外包计算结果解密。相比于 Twin-SM9, 其更适用于解密操作频繁且算力资源受限的环境中。另外, 针对半可信云服务中心解密外包计算结果的高效审计问题, BAOC-Twin-SM9 运用随机盲化技术实现了多密文外包解密结果的批量审计功能, 从而保证了外包计算结果的正确性。理论分析和仿真实验数据论证了 BAOC-Twin-SM9 的可行性与高效性, BAOC-Twin-SM9 拓展了 SM9 系列算法的应用领域。

关键词 Twin-SM9; 解密外包; 批量审计; SM9; 密钥封装

中图法分类号 TP309 DOI 号 10.19363/J.cnki.cn10-1380/tn.2023.07.07

Multi-Ciphertext Batch Auditable Outsourced Twin-SM9 Key Encapsulation Mechanism

LIU Kuan¹, NING Jianting^{1,3}, WU Wei², CHEN Haixia¹

¹ College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China

² School of Mathematics and Statistics, Fujian Normal University, Fuzhou 350117, China

³ State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Abstract A series of functional extensions and security proofs (extensions) of identity-based cryptographic algorithms have been proposed in order to promote the cryptographic techniques to achieve the goal of safety and advancement, independent control since identity-based cryptographic algorithm SM9 was incorporated into ISO/IEC international standards. Based on Gap- q -BCAA1 assumption, Cheng gave security analysis of SM9 key encapsulation and encryption algorithm, key exchange protocol. Later, Lai et al. proposed Twin-SM9 key encapsulation mechanism to effectively eliminate the dependence of SM9 series algorithm on Gap assumption with Twin-Hash-ElGamal. However, the decryption operation of Twin-SM9 key encapsulation mechanism requires two bilinear pairing operations. In resource-constrained environment where frequent decryptions of massive data are required (wireless sensing equipment, cryptographic chip etc), the expensive pairing cost will become an important bottleneck which restricts the efficiency of the system. To solve the problem, we propose a new key encapsulation mechanism named BAOC-Twin-SM9 based on Twin-SM9, with purpose of supporting multi-ciphertexts batch auditing and decryption outsourcing. The security of our BAOC-Twin-SM9 is secure against Replayable Chosen Ciphertext Attacks (RCCA) under random oracle model. BAOC-Twin-SM9 eliminates the influence of bilinear pairing operations on the decryption efficiency of the Twin-SM9 key encapsulation mechanism powerfully using the federic computing power of cloud service center, terminal data user with limited computing resources can finally decrypt the outsourced computing results with only two simple exponential operations. Compared to the Twin-SM9, it is more suitable for resource-constrained environment when frequent decryption operations are required. To solve the issue of efficient audit for decryption of outsourced computing results in semi-trusted cloud service center, BAOC-Twin-SM9

通讯作者: 伍玮, 博士, 教授, Email: weiwu81@gmail.com。

本课题得到国家自然科学基金项目(No. 62032005, No. 61972094, No. 61872087, No. 61902070)、福建省自然科学基金项目(No. 2020J02016)和福建省科协第二届青年人才托举工程资助。

收稿日期: 2022-03-14; 修改日期: 2022-06-22; 定稿日期: 2023-04-18

implements batch auditing for multi-ciphertexts outsourcing decryption by using random blinding technology, thus ensuring the correctness of outsourcing computing results. Theoretical analysis and simulation data demonstrate the feasibility and efficiency of our BAOC-Twin-SM9. Our BAOC-Twin-SM9 extends the application scope of SM9 series algorithms.

Key words Twin-SM9; outsourced decryption; batch audit; SM9; key encapsulation

1 引言

为解决传统公钥密码基础设施(Public Key Infrastructure, PKI)体制下证书管理系统随着数据使用者的增多而变得复杂难以管理的问题, Shamir^[1]于 1984 年创造性地提出了基于标识的密码体制, 通过使用数据使用者容易获得的公开信息(如电话号码、邮箱账号、生日等)作为公钥, 消除了公钥对数字证书的依赖, 给数据使用者带来灵活的公钥管理空间。2001 年, Boneh 和 Franklin^[2]构造出了第一个基于双线性对的可证明安全标识加密方案。随后, 具备各种性能的标识密码方案被陆续提出^[3-13]。

尽管标识密码已经取得了长足发展, 但大多是基于国外密码标准设计。为实现构建自主可控的信息安全技术体系, 改变标识密码算法大多由国外设计为主的形势, 我国自主构建了包括密钥封装机制、公钥加密算法、数字签名算法和密钥协商协议的 SM9 标识系列密码算法^[14], 并被我国商用密码行业标准和国家标准所采纳, 最终被纳入为 ISO/IEC 国际标准, 有效保障了国家和网络信息安全。

SM9 标识系列密码算法在被纳入国家密码行业标准后, 愈来愈受到国内学者的广泛关注, 并涌现出一系列显著性的研究成果。Zhang 等人^[15]将 SM3 杂凑算法对明文进行盲化处理, 再通过 SM9 数字签名技术构造出了 SM9 盲签名方案。文献[16]通过构造预处理矩阵的方式将 SM9 数字签名中签名和验证算法的计算复杂度得到了显著降低。Xu 等人^[17]针对 Wang 等人^[18]和 Gesiler 等人^[19]的私钥分发安全性问题, 构造出基于 SM9 的可分离分布式匿名密钥分发方案。文献[20]为实现区块链交易过程中的隐私保护, 通过改进的 SM9 标识密码算法构造出一种满足签名不可伪造性、保障节点匿名性和前向安全性等特性的群签名方案。文献[21]建立了基于 SM9 标识加密算法的用户撤销机制, 并提出了具备鲁棒性的服务器辅助撤销方案。随后, 文献[22-23]针对 SM9 中的 R-ate 双线性对运算做出进一步优化。2018 年, Cheng^[24]基于 Gap- q -BCAA1^[25]困难问题假设对 SM9 密钥封装、公钥加密、密钥协商算法进行了安全性分析。最近, Lai 等人^[26]在基于 q -Strong Diffie-Hellman (q -SDH)假设条件下证明了 SM9 数字签名具

备存在性不可伪造性(existentially unforgeable against adaptive chosen message and identity attacks, EUF-CMIA), 并采用 Twin-Hash-ElGamal^[27]技术给出了基于 q -Bilinear Diffie-Hellman Inversion (q -BDHI) 困难问题且满足 Indistinguishability against Adaptive Chosen Ciphertext Attacks (IND-CCA) 安全的 Twin-SM9 密钥封装机制。

然而, Twin-SM9 标识密钥封装机制在解密过程中需要两次配对运算, 且解密配对次数与解密文件数(或解密次数)呈线性相关, 在需要对海量数据进行频繁解密操作且资源受限的环境中(如无线传感设备、密码芯片等)存在一定的挑战, 计算代价高昂的配对运算所带来的计算开销将是系统部署的一大瓶颈。

1.1 本文贡献

针对上述问题, 本文提出了一种基于 Twin-SM9 的支持多密文批量审计的解密外包新型密钥封装机制 BAOC-Twin-SM9, 并在随机谰言模型下证明了其具备 Replayable Chosen Ciphertext Attacks (RCCA)^[28]安全性。本文对 BAOC-Twin-SM9 进行仿真实验, 针对 1 份和 100 份密文, BAOC-Twin-SM9 分别额外增加了 375.68ms 和 492.75ms 的审计时间, 终端解密时间分别约为 397.27ms 和 39215ms, 而 Twin-SM9 的终端解密时间分别约为 1085.99ms 和 107792ms。实验结果表明, 针对 100 份密文, 相比 Twin-SM9, BAOC-Twin-SM9 在牺牲 492.75ms 额外批量审计开销的情况下, 将终端解密时间减少了约 63.62%。BAOC-Twin-SM9 同时支持了如下特性: (1)安全的解密外包; (2)外包结果的批量可审计性; (3)抗密钥泄露攻击。具体如下:

(1) 安全的解密外包。BAOC-Twin-SM9 将解密过程中耗时较大的配对运算外包至云服务中心, 并采用外包密钥盲化技术使云服务中心得不到任何有效信息, 使得计算资源有限的数据使用者只需执行 2 次指数运算就能对外包结果进行解密, 从而以较小的代价恢复明文。

(2) 外包结果的批量可审计性。为保证外包结果的正确性, 同时尽量降低审计机制对系统解密效率的影响, 本文将密钥分发中心(Key Generate Center, KGC)作为审计机构, 在不需要增加额外参数的情

况下, 对云服务中心返回的多份外包解密结果, KGC 只需进行 2 次指数操作就能对外包结果实现批量审计。

(3) 抗密钥泄露攻击. 在与云服务中心交互的情况下, 恶意第三方即使拿到数据使用者解密密钥, 其对云服务中心返回的外包结果执行解密操作后的结果依旧被数据使用者的私钥所盲化, 在不与云服务中心交互的情况下, 恶意第三方拿到数据使用者解密密钥, 此时恶意第三方本地执行解密操作后的结果将被云服务中心私钥和数据使用者私钥所盲化。

表 1 列举了 BAOC-Twin-SM9 密钥封装机制与其他 SM9 密钥封装机制的功能性和安全性对比。由表 1 与实验结果可得, BAOC-Twin-SM9 更适用于计算资源有限但又需要进行耗时较大的高并发读写操作环境中。

表 1 与其他工作相比较
Table 1 Comparison with other related work

功能对比	解密外包计	批量审	困难假设	安全性
SM9 ^[24]	×	×	Gap- q -BCAA1	IND-CCA
Twin-SM9 ^[26]	×	×	q -BDHI	IND-CCA
BAOC-Twin-SM9	✓	✓	q -BDHI	RCCA

(注: ✓表示具备此性质, ×表示不具备此性质)

1.2 本文组织结构

本文的组织结构安排如下: 第 2 节将阐述双线性群、困难问题假设和支持多密文批量审计的解密外包 Twin-SM9 密钥封装机制系统模型、定义和安全模型; 第 3 节给出 BAOC-Twin-SM9 算法的具体构造并给出算法正确性分析; 第 4 节依据安全模型对算法安全性进行证明; 第 5 节对 BAOC-Twin-SM9 性能进行理论和实验数据分析; 第 6 节对全文工作进行归纳。

2 预备知识

本节简要介绍双线性群、困难问题假设、支持多密文批量审计的解密外包标识密钥封装机制的算法基本描述、安全模型等预备知识。有关预备知识的符号说明为: 设 D 为双线性群, P 和 g 分别为加法群和乘法群中的元素, 加法群中的倍数运算用 mP 所示, 乘法群中的指数运算用 g^m 表示。字符串或比特串的拼接用 $m||n$ 表示。 $R(m) \rightarrow n$ 表示一个输入为 m , 输出为 n 的概率算法。定义自然数集合为 \mathbb{N} , 如

果对每个 $a \in \mathbb{N}$, 存在 $\sigma_a \in \mathbb{N}$ 使得对所有 $\sigma > \sigma_a$, 都存在 $\varepsilon(\sigma) \leq \sigma^{-a}$, 则函数 $\varepsilon: \mathbb{N} \rightarrow [0, 1]$ 是可忽略的。

2.1 双线性群

给定安全参数为 λ , p 设为与 λ 相关的大素数, 定义阶均为 p 的循环群 \mathbb{G}_1 , \mathbb{G}_2 和 \mathbb{G}_T , 则双线性映射 $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ 应满足以下三个性质:

(1) 双线性性: 对所有的生成元 $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ 和 $a, b \in \mathbb{Z}_p$, 有 $\hat{e}(au, bv) = \hat{e}(u, v)^{ab}$;

(2) 非退化性: 至少存在元素 $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ 使得 $\hat{e}(u, v) \neq 1$;

(3) 可计算性: 存在有效的多项式时间算法对任意的 $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, 求解 $\hat{e}(u, v)$ 的值。

双线性群 D 由上述五元组 $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, p)$ 组成。其中非对称双线性群满足 $\mathbb{G}_1 \neq \mathbb{G}_2$, 否则是对称双线性群, 非对称双线性群是 SM9 标识系列密码算法的构造基础。令群 \mathbb{G}_1 , \mathbb{G}_2 的生成元分别为 Q_1 , Q_2 , 则存在公开可高效计算的同构映射 $\varphi: \mathbb{G}_2 \rightarrow \mathbb{G}_1$, 满足 $\varphi(Q_2) = Q_1$ 。

2.2 困难问题假设

设群 \mathbb{G}_1 , \mathbb{G}_2 的生成元分别为 Q_1 , Q_2 , 则在非对称双线性群中的 q -BDHI 问题如下所示:

定义 1. q -BDHI 问题. 给定 $q+2$ 个元素 $(Q_1, Q_2, rQ_2, r^2Q_2, \dots, r^qQ_2) \in \mathbb{G}_1 \times \mathbb{G}_2^{q+1}$, 其中 r 未知, 求解 $e(Q_1, Q_2)^{\frac{1}{r}}$ 的值。

q -BDHI 假设成立的条件是对任意概率多项式时间算法 \mathcal{A} , 其求解 q -BDHI 问题的优势是可忽略的。

2.3 支持多密文批量审计的解密外包标识密钥封装的系统模型、定义和安全模型

2.3.1 支持多密文批量审计的解密外包标识密钥封装的系统模型

支持多密文批量审计的解密外包标识密钥封装的系统模型整体架构图如图 1 表示, 系统共涉及 5 个实体, 即 KGC、云存储中心、云服务中心、数据拥有者、数据使用者。首先, KGC 生成系统公开参数并为数据使用者分发密钥。其次, 数据拥有者将密文上传到云存储中心, 云存储中心再将封装密文传送至云服务中心。然后, 数据使用者向云服务中心发起解密外包请求, 在收到解密外包请求后, 云服务中心对封装密文进行部分解密并将外包结果返还给数据使用者。最后 KGC 作

为权威机构执行外包结果审计算法并返回审计结果给数据使用者。当且仅当 KGC 审计通过, 即外

包结果正确时, 数据使用者才能(通过解密)获得正确的封装密钥。

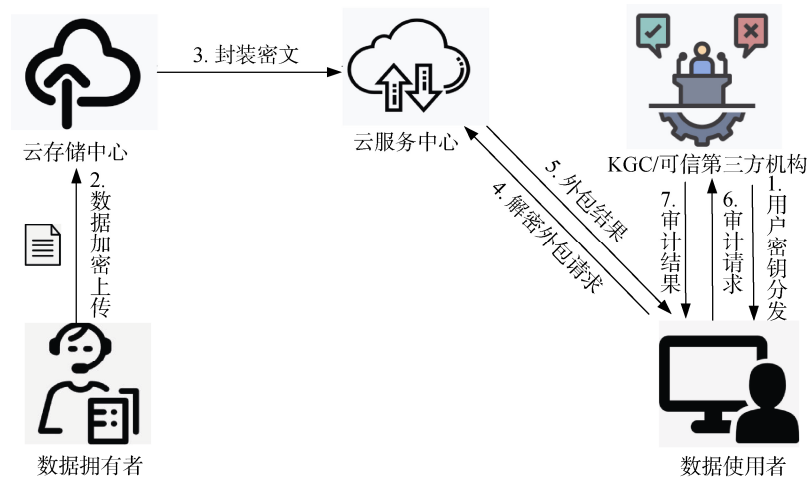


图 1 系统模型图

Figure 1 The diagram of system model

2.3.2 支持多密文批量审计的解密外包标识密钥封装定义

支持多密文批量审计的解密外包标识密钥封装机制由以下九个多项式时间算法描述。

- $\text{Setup}(\lambda) \rightarrow (pp, msk)$: 系统初始化算法。已知系统安全参数 λ , 算法以 λ 为输入, 输出系统公开参数 pp 和主私钥 msk , 其中 pp 是公开的, msk 由 KGC 秘密保存, 该算法由 KGC 运行。
- $\text{Setup}_c(\lambda, pp) \rightarrow (pk_c, sk_c)$: 云服务中心初始化算法。已知系统安全参数 λ 和系统公开参数 pp , 算法以 λ 和 pp 为输入, 输出云服务中心主公私钥对 (pk_c, sk_c) , 其中 pk_c 公开, sk_c 由云服务中心秘密保存, 该算法由云服务中心运行。
- $\text{Setup}_u(\lambda, pp) \rightarrow (pk_u, sk_u)$: 数据使用者初始化算法。已知系统安全参数 λ 和系统公开参数 pp , 算法以 λ 和 pp 为输入, 输出数据使用者主公私钥对 (pk_u, sk_u) , 其中 pk_u 公开, sk_u 由数据使用者秘密保存, 该算法由数据使用者运行。
- $\text{KeyGen}(pp, pk_c, pk_u, msk, ID) \rightarrow sk_{ID}$: 数据使用者解密密钥生成算法。已知数据使用者标识 ID , 算法以 (pp, pk_c, pk_u, msk) 和 ID 为输入, 输出 sk_{ID} 为数据使用者解密密钥, 并将 sk_{ID} 发送给数据使用者, 该算法由 KGC 运行。
- $\text{KeyGen}_{out}(sk_{ID}) \rightarrow tk_{ID}$: 外包密钥生成算法。已知数据使用者解密密钥 sk_{ID} , 算法以 sk_{ID} 为输入, 输出外包密钥 tk_{ID} , 该算法由数据使用

者运行。

- $\text{Encap}(pp, pk_u, ID) \rightarrow (CT, K)$: 密钥封装算法。算法以系统公开参数 pp 、数据使用者公钥 pk_u 和数据使用者标识 ID 为输入, 输出密文 CT 和封装密钥 K , 该算法由数据所有者运行。
- $\text{Dec}_{out}(CT, tk_{ID}, sk_c) \rightarrow C'$: 解密外包算法。算法以密文 CT 、外包密钥 tk_{ID} 和云服务中心私钥 sk_c 为输入, 输出外包结果 C' , 该算法由云服务中心运行。
- $\text{Audit}(\{CT_i\}_{i \in [n]}, \{C'_i\}_{i \in [n]}, msk) \rightarrow 1 \text{ 或 } 0$: 外包结果批量审计算法。算法以封装密文集合 $\{CT_i\}_{i \in [n]}$ 、外包结果集合 $\{C'_i\}_{i \in [n]}$ 和系统主私钥 msk 为输入 (n 为密文份数), 输出审计结果 1 或者 0, 其中 1 代表云服务中心解密正确, 0 代表解密错误, 该算法由 KGC 运行。
- $\text{Dec}_u(C', CT, sk_u) \rightarrow K'$: 数据使用者解密算法。算法以外包结果 C' 、封装密文 CT 和数据使用者私钥 sk_u 为输入, 输出封装密钥 K' , 该算法由数据使用者运行。

2.3.3 支持多密文批量审计的解密外包标识密钥封装安全模型

支持多密文批量审计的解密外包标识密钥封装机制安全模型由挑战者和攻击者之间交互的安全游戏定义。基于以上系统模型, 定义如下两种类型的攻击者:

(1) 类型-1 攻击者: 该攻击者为恶意的数据使用者。对于类型-1 攻击者, 本方案应保证任何未经授权

的数据使用者都不能成功解密云存储中心中的密文, 即使任何一组非授权的数据使用者共谋也不能获得有效的解密权限。

(2) 类型-2 攻击者: 该攻击者为半可信的云服务中心。对于类型-2 攻击者, 在遵循系统规定的基本步骤前提下尽可能地收集有用的秘密信息, 其安全性要求是在运行过程中不被允许得到任何有用的秘密信息, 即类型-2 攻击者不能判断出是哪个数据使用者在访问云服务中心的密文数据或者发起解密外包服务请求。

对于以上两种类型的攻击者, 为了定义满足以上安全要求的适用于 BAOC-Twin-SM9 密钥封装机制的安全性, 定义安全游戏如下:

1. RCCA: 与文献[28]类似, 定义 RCCA 安全, 即是选择密文攻击的一种弱化版本, 其允许对密文进行“无恶意修改”, 为此根据如下所示的两种类型的攻击者来定义系统的 RCCA 安全。

(1) 类型-1 攻击者的 RCCA 安全: 由挑战者和攻击者之间交互的安全游戏来定义, 安全游戏如下所示:

初始化. 给定系统安全参数 λ 和公开参数 pp , 挑战者执行 $\text{Setup}(\lambda)$ 和 $\text{Setup}_c(\lambda, pp)$ 算法, 输出系统公开参数和云服务中心公私钥对并返回攻击者为 (pp, pk_c, sk_c) 。

询问阶段(一). 挑战者创建一个空的列表 T , 整型计数器 $j = 0$ 和空集 D , 攻击者适应性地进行如下询问:

1) **创建标识阶段:** 挑战者设置 $j = j + 1$, 运行 Setup_u 算法获得数据使用者的公私钥 pk_u, sk_u , 运行 KeyGen (以标识 ID 为输入) 算法, 输出解密密钥 sk_{ID} , 运行 KeyGen_{out} (以 sk_{ID} 为输入) 算法输出外包密钥 tk_{ID} , 最后挑战者存储元组 $(j, ID, sk_{ID}, tk_{ID}, pk_u, sk_u)$ 到表 T 中。

2) **解密私钥询问阶段:** 挑战者检查第 i 个元组 $(i, ID, sk_{ID}, pk_u, sk_u)$ 是否存在于表 T 中。如果不存在, 则返回为 \perp 。否则, 挑战者设置 $D = D \cup \{ID\}$, 执行 KeyGen 算法并将私钥 sk_{ID} 输出, 最终返回攻击者为 (sk_{ID}, pk_u, sk_u) 。

3) **外包密钥询问阶段:** 挑战者检查第 i 个元组 (i, ID, tk_{ID}) 是否存在于表 T 中, 如果不存在, 则返回为 \perp , 否则返回 tk_{ID} 。

4) **密文解密询问阶段:** 挑战者检查第 i 个元组 (i, ID, sk_{ID}) 是否存在于表 T 中, 如果不存在, 则返回

\perp , 否则返回对密文 CT 的解密结果。

挑战阶段. 待询问阶段(一)终止后, 挑战标识 ID^* 由攻击者输出。安全模型规定攻击者没有对 ID^* 进行过私钥询问, 挑战者执行 $\text{Encap}(pp, pk_u, ID^*)$ 算法并将挑战密文和封装密钥 (CT^*, K^*) 输出。随后随机生成一个比特 $\eta \in \{0, 1\}$, 令 $K_\eta = K^*$, 同时在封装密钥空间中随机设置一个会话密钥为 $K_{1-\eta}$, 最后返回攻击者为 (CT^*, K_0, K_1) 。

询问阶段(二). 与询问阶段(一)相比, 此阶段有如下限制: (1) 攻击者不能简单获得关于挑战密文的解密密钥和对应数据使用者的私钥; (2) 攻击者不能对挑战密文进行解密询问, 即模型要求攻击者不能对 ID^* 进行私钥询问, 同时不能对密文 (CT^*, ID^*) 进行解密询问, 挑战者根据询问阶段(一)的结果对攻击者回复。

上述安全游戏赋予了类型-1 攻击者更强的能力, 因为在与挑战者交互过程中, 类型-1 攻击者可以获得云服务中心的主私钥 sk_c 。

猜测阶段. 关于 η 的猜测 $\eta' \in \{0, 1\}$ 由攻击者输出, $\eta' = \eta$ 时表明攻击者赢得游戏。攻击者 \mathcal{A} 赢得游戏的优势由如下等式定义:

$$\text{Adv}_{\mathcal{A}}^{\text{RCCA}}(\lambda) = \left| \Pr[\eta' = \eta] - \frac{1}{2} \right|.$$

定义 2. BAOC-Twin-SM9 密钥封装机制关于类型-1 攻击者具备 RCCA 安全性满足对任意概率多项式时间的类型-1 攻击者 \mathcal{A} , 其在上述安全游戏中获胜的优势 $\text{Adv}_{\mathcal{A}}^{\text{RCCA}}(\lambda)$ 都是可忽略的。

(2) 类型-2 攻击者的 RCCA 安全: 安全性游戏描述与类型-1 攻击者安全性游戏几乎相同, 但是初始化被作如下修订:

初始化. 挑战者运行 Setup 算法并发送公开参数 pp 给攻击者, 攻击者运行 Setup_c 算法并发送云服务中心公钥 pk_c 给挑战者。

定义 3. BAOC-Twin-SM9 密钥封装机制关于类型-2 攻击者具备 RCCA 安全性满足对任意概率多项式时间的类型-2 攻击者 \mathcal{A} , 其在上述游戏中获胜的优势 $\text{Adv}_{\mathcal{A}}^{\text{RCCA}}(\lambda)$ 都是可忽略的。

(3) 选择明文安全: 如果攻击者在上述交互游戏中不能进行任何密文解密询问, 则 BAOC-Twin-SM9 密钥封装机制是选择明文安全的。

(4) 选择性安全: 如果在初始化前增加一个攻击

者声明挑战标识 ID^* 的预备阶段, 则 BAOC-Twin-SM9 密钥封装机制是选择性安全的。

2. 可审计性. 可审计性确保权威机构能对云服务中心生成的外包结果的正确性进行审计。恶意云服务中心对同一密文解密不能生成两种不同的有效外包结果。当且仅当外包结果通过审计时, 外包结果才是有效的, 其安全性通过挑战者和攻击者之间运行的一系列交互游戏来定义。安全模型如下所示:

初始化. 挑战者执行 Setup 算法将系统公私钥对 (pp, msk) 输出, 并将 pp 发送给攻击者, 攻击者运行 $Setup_c$ 生成 pk_c , 并发送 pk_c 到挑战者。

询问阶段(一). 攻击者适应性地发起类型-2 攻击者在 RCCA 安全游戏模型中的所有询问。

1) **创建标识阶段:** 挑战者设置 $j = j + 1$, 运行 $Setup_u$ 算法获得数据使用者的公私钥 pk_u, sk_u , 运行 KeyGen (以标识 ID 为输入) 算法, 输出解密密钥 sk_{ID} , 运行 $KeyGen_{out}$ (以 sk_{ID} 为输入) 算法输出外包密钥 tk_{ID} , 最后挑战者存储元组 $(j, ID, sk_{ID}, tk_{ID}, pk_u, sk_u)$ 到表 T 中。

2) **解密私钥询问阶段:** 挑战者检查第 i 个元组 $(i, ID, sk_{ID}, pk_u, sk_u)$ 是否存在于表 T 中。如果不存在, 则返回为 \perp 。否则, 挑战者设置 $D = D \cup \{ID\}$, 执行 KeyGen 算法并输出私钥 sk_{ID} , 以 (sk_{ID}, pk_u, sk_u) 的形式返回给攻击者。

3) **外包密钥询问阶段:** 挑战者检查第 i 个元组 (i, ID, tk_{ID}) 是否存在于表 T 中, 如果不存在, 则返回为 \perp , 否则返回 tk_{ID} 。

4) **密文解密询问阶段:** 挑战者检查第 i 个元组 (i, ID, sk_{ID}) 是否存在于表 T 中, 如果不存在, 则返回 \perp , 否则返回对密文 CT 的解密结果。

挑战阶段. 待询问阶段(一)终止后, 挑战标识 ID^* 由攻击者输出, 安全模型规定攻击者没有对 ID^* 的私钥进行过询问。挑战者执行 $Encap(pp, ID^*)$ 算法并将挑战密文和封装密钥 (CT^*, K^*) 输出, 随后随机生成一个比特 $\eta \in \{0, 1\}$, 令 $K_\eta = K^*$, 同时在封装密钥空间中随机设置一个会话密钥为 $K_{1-\eta}$, 最后返回攻击者为 (CT^*, K_0, K_1) 。

询问阶段(二). 与询问阶段(一)保持一致。

输出阶段. 攻击者输出标识 ID^* 和一个密文元组 (C_1^*, C_2^*) , 其中 (C_1^*, C_2^*) 是 C^* 的两个外包结

果。假定元组 $(ID^*, sk_{ID^*}, tk_{ID^*}, pk_u, sk_u)$ 存在于表 T 中 (如果不存在, 则挑战者可以生成一个元组), 只有 $Audit(CT^*, C_1^*, msk) \rightarrow 1 \wedge Audit(CT^*, C_2^*, msk) \rightarrow 1 \wedge Dec_u(C_1^*, sk_u) \neq Dec_u(C_2^*, sk_u)$ 同时成立时, 攻击者才能赢得上述交互游戏。

定义 4. BAOC-Twin-SM9 密钥封装机制具备可审计性满足对任意概率多项式时间的攻击者 \mathcal{A} , 其在上述游戏中获胜的优势 $Adv_{\mathcal{A}}^{RCCA}(\lambda)$ 都是可忽略的。

3 BAOC-Twin-SM9 密钥封装机制

本节将外包计算与 Twin-SM9 密钥封装机制结合, 将 Twin-SM9 解密过程中涉及到的配对运算全部安全外包至云服务中心执行, 并对云服务中心返回过来的多份外包结果进行批量审计, 最后给出 BAOC-Twin-SM9 密钥封装机制的完整描述。

3.1 技术概述

在本节中, 将给出构造 BAOC-Twin-SM9 密钥封装机制所涉及到的相关技术概述。

在安全解密外包方面, 利用外包密钥盲化技术实现安全的解密外包计算。本文将 $Setup_u$ 阶段生成的数据使用者的主公钥之一 Z_u 作为外包密钥盲化因子, 使得云服务中心得不到任何有效信息, 数据使用者利用其主密钥只需进行 2 次指数运算就能完成最终的解密操作。

在外包结果的可审计性方面, 本文的审计方法无需添加任何额外参数, KGC 只需进行 2 次指数运算就能对云服务中心返回的多份外包结果进行批量审计, 使得审计机制在确保外包结果正确性的前提下对系统解密效率影响较低, 从而 BAOC-Twin-SM9 更适用于资源受限的小型无线设备中。

在抗密钥泄露方面, 本文提出的 BAOC-Twin-SM9 密钥封装机制在算法 KeyGen 的运行过程中, 数据使用者解密密钥的生成需要云服务中心公钥和数据使用者公钥的共同参与。故即使一个有效的密钥 sk_{ID} 被泄露出去, 其他任意恶意的第三方用户依旧无法利用 sk_{ID} 成功解密密文。因为算法 Dec_{out} 的生成在 sk_{ID} 被暴露出来后依旧被原始数据使用者私钥所盲化。

3.2 算法描述

- Setup: KGC 输入安全参数 λ , 选择一个非对称双线性群 $D = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, N)$, 其中 N 为大

素数且 $N > 2^\lambda$, 令群 $\mathbb{G}_1, \mathbb{G}_2$ 的生成元分别为 P_1 和 P_2 , 其中 $P_1 = \varphi(P_2)$ 。随机选取 $k_1, k_2 \in [1, N-1]$, 定义密码哈希函数 $H_1: \{0,1\}^* \times \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*, H_2: \{0,1\}^* \times \mathbb{Z}_N^* \rightarrow \{0,1\}^{klen}$ ($klen$ 为封装密钥长度), 接着分别计算群 \mathbb{G}_1 中的元素 $P_{pub} = k_1 P_1$, 群 \mathbb{G}_T 中的元素 $g_0 = e(P_1, P_2)$, $g_1 = e(P_1, P_2)^{k_1}$, $g_2 = e(P_1, P_2)^{k_2}$, 密钥生成函数识别符用 hid 表示, 输出系统公开参数 pp 和系统主私钥 msk 分别为:

$$pp = (D, P_1, P_2, P_{pub}, g_0, g_1, g_2, H_1, H_2, hid),$$

$$msk = (k_1, k_2)。$$

- **Setup_c**: 云服务中心输入安全参数 λ 和系统公开参数 pp , 选择随机数 $k_c \in [1, N-1]$, 令云服务中心主私钥为 $sk_c = k_c$, 计算群 \mathbb{G}_2 中的元素 $K_c = k_c P_2$, 令 $pk_c = K_c$, 输出云服务中心公私钥对 (pk_c, sk_c) 。
- **Setup_u**: 数据使用者输入安全参数 λ 和系统公开参数 pp , 选择随机数 $z_u \in [1, N-1]$ 作为数据使用者主私钥, 令 $sk_u = z_u$, 计算群 \mathbb{G}_2 中的元素 $Z_u = z_u P_2$, 群 \mathbb{G}_T 中的元素 $S_u = g_0^{z_u}$, 令 $pk_u = (Z_u, S_u)$ 作为数据使用者主公钥, 输出数据使用者公私钥对 (pk_u, sk_u) 。
- **KeyGen**: 数据使用者向 KGC 提交密钥申请, 并发送自己标识相关信息。KGC 输入系统公开参数 pp , 系统主私钥 $msk = (k_1, k_2)$, 云服务中心公钥 $pk_c = K_c$, 数据使用者公钥 $pk_u = (Z_u, S_u)$, 数据使用者标识 $ID \in \{0,1\}^*$, KGC 首先在有限域 F_N 上计算非零元素 $s_1 = H_1(ID \parallel hid, N) + k_1$, 选择随机数 $\gamma \in [1, N-1]$, 计算 $K_0 = \gamma \cdot K_c + k_1 \cdot s_1^{-1} \cdot Z_u$, $K_1 = \gamma \cdot K_c + k_2 \cdot s_1^{-1} \cdot Z_u$, $K_2 = \gamma \cdot P_2$, 令 $sk_{ID} = (K_0, K_1, K_2)$, 输出 sk_{ID} 作为数据使用者解密密钥。
- **KeyGen_{out}**: 数据使用者调用外包密钥生成算法, 输入数据使用者解密密钥, 令 $tk_{ID} = sk_{ID} = (K_0, K_1, K_2)$, 输出外包密钥 tk_{ID} 。
- **Encap**: 数据拥有者调用加密算法, 输入系统公开参数 pp , 数据使用者公钥 $pk_u = (Z_u, S_u)$, 为了封装比特长度为 $klen$ 的密钥给标识为 ID 的数据使用者, 数据拥有者选择随机数 $r \in [1, N-1]$,

计算 $C_0 = S_u^r$, $C_1 = r(H_1(ID \parallel hid, N)P_1 + P_{pub})$, $w_1 = g_1^r$, $w_2 = g_2^r$, $K = H_2(C_1 \parallel w_1 \parallel w_2 \parallel ID, klen)$, 输出 (K, CT) , 其中 K 是封装的密钥, $CT = (C_0, C_1)$ 是封装密文。

- **Dec_{out}**: 云服务中心收到数据使用者的解密外包请求后, 调用解密外包算法。算法输入为封装密文 $CT = (C_0, C_1)$, 外包密钥 $tk = (K_0, K_1, K_2)$, 云服务中心主私钥 $sk_c = k_c$, 执行解密外包算法生成外包结果 $C'_1 = \frac{e(C_1, K_0)}{e(C_1, sk_c \cdot K_2)} = e(P_1, Z_u)^{k_1 r}$, $C'_2 = \frac{e(C_1, K_1)}{e(C_1, sk_c \cdot K_2)} = e(P_1, Z_u)^{k_2 r}$, 输出外包结果 $C' = (C'_1, C'_2)$ 。

- **Audit**: KGC 作为可信第三方机构对同一数据拥有者 n 份外包结果的正确性进行批量审计。算法输入系统主私钥 msk , 封装密文集合 $\{CT_i\}_{i \in [n]} = \{C_{i,0}, C_{i,1}\}_{i \in [n]}$, 外包结果集合 $\{C'_i\}_{i \in [n]} = \{C'_{i,1}, C'_{i,2}\}_{i \in [n]}$, 其中 n 为密文份数。对于封装密文集合 $\{C_{i,0}\}_{i \in [n]}$, 计算

$$E = \prod_{i=1}^n S_u^{r_i} = S_u^{\sum_{i=1}^n r_i}, \text{ 对于外包结果集合 } \{C'_{i,1}\}_{i \in [n]},$$

$$\{C'_{i,2}\}_{i \in [n]}. \text{ 计算 } CK_1 = \prod_{i=1}^n C'_{i,1} = e(P_1, Z_u)^{k_1 \sum_{i=1}^n r_i},$$

$$CK_2 = \prod_{i=1}^n C'_{i,2} = e(P_1, Z_u)^{k_2 \sum_{i=1}^n r_i}, \text{ 判断等式}$$

$E^{k_1} = CK_1$, $E^{k_2} = CK_2$ 是否同时成立, 输出 1 代表云服务中心解密正确, 输出 0 代表云服务中心解密错误。

- **Dec_u**: 数据使用者运行在线解密算法。输入外包结果 $C' = (C'_1, C'_2)$, 封装密文 $CT = (C_0, C_1)$, 数据使用者解密私钥 $sk_u = z_u$, 当且仅当 $\text{Audit}(CT, C', msk) = 1$ ($n=1$) 时, 计算 $w'_1 = C'_1{}^{sk_u^{-1}} = e(P_1, P_2)^{k_1 r}$, $w'_2 = C'_2{}^{sk_u^{-1}} = e(P_1, P_2)^{k_2 r}$, 计算封装的密钥 $K' = H_2(C_1 \parallel w'_1 \parallel w'_2 \parallel ID, klen)$, 数据使用者最后输出封装密钥 K' 。

3.3 正确性分析

假设 $CT = (C_0, C_1)$ 为封装密文, $C' = (C'_1, C'_2)$ 为外包结果, 则可依据以下等式对 BAOC-Twin-SM9 标识密钥封装机制的正确性进行证明。

$$\begin{aligned}
C'_1 &= \frac{e(C_1, K_0)}{e(C_1, sk_c \cdot K_2)} \\
&= \frac{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + P_{pub}), \gamma \cdot K_c + k_1(H_1(ID \| hid, N) + k_1)^{-1} \cdot Z_u)}{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + P_{pub}), \gamma \cdot k_c \cdot P_2)} \\
&= \frac{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + k_1 \cdot P_1), \gamma \cdot K_c + k_1(H_1(ID \| hid, N) + k_1)^{-1} \cdot Z_u)}{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + k_1 \cdot P_1), \gamma \cdot k_c \cdot P_2)} \\
&= \frac{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + k_1 \cdot P_1), \gamma \cdot k_c \cdot P_2 + k_1(H_1(ID \| hid, N) + k_1)^{-1} \cdot Z_u)}{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + k_1 \cdot P_1), \gamma \cdot k_c \cdot P_2)} \\
&= e(r \cdot (H_1(ID \| hid, N) + k_1) \cdot P_1, k_1(H_1(ID \| hid, N) + k_1)^{-1} \cdot Z_u) \\
&= e(P_1, Z_u)^{r \cdot (H_1(ID \| hid, N) + k_1) \cdot k_1(H_1(ID \| hid, N) + k_1)^{-1}} \\
&= e(P_1, Z_u)^{k_1 r} \\
C'_2 &= \frac{e(C_1, K_1)}{e(C_1, sk_c \cdot K_2)} \\
&= \frac{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + P_{pub}), \gamma \cdot K_c + k_2(H_1(ID \| hid, N) + k_1)^{-1} \cdot Z_u)}{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + P_{pub}), \gamma \cdot k_c \cdot P_2)} \\
&= \frac{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + k_1 \cdot P_1), \gamma \cdot K_c + k_2(H_1(ID \| hid, N) + k_1)^{-1} \cdot Z_u)}{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + k_1 \cdot P_1), \gamma \cdot k_c \cdot P_2)} \\
&= \frac{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + k_1 \cdot P_1), \gamma \cdot k_c \cdot P_2 + k_2(H_1(ID \| hid, N) + k_1)^{-1} \cdot Z_u)}{e(r \cdot (H_1(ID \| hid, N) \cdot P_1 + k_1 \cdot P_1), \gamma \cdot k_c \cdot P_2)} \\
&= e(r \cdot (H_1(ID \| hid, N) + k_1) \cdot P_1, k_2(H_1(ID \| hid, N) + k_1)^{-1} \cdot Z_u) \\
&= e(P_1, Z_u)^{r \cdot (H_1(ID \| hid, N) + k_1) \cdot k_2(H_1(ID \| hid, N) + k_1)^{-1}} \\
&= e(P_1, Z_u)^{k_2 r}
\end{aligned}$$

$$\begin{aligned}
E &= \prod_{i=1}^n C_{i,0} = S_u^{r_1 + r_2 + \dots + r_n} = S_u^{\sum_{i=1}^n r_i} = e(P_1, P_2)^{z_u \sum_{i=1}^n r_i} \\
E^{k_1} &= e(P_1, P_2)^{k_1 z_u \sum_{i=1}^n r_i} = CK_1 = \prod_{i=1}^n C'_{i,1} = e(P_1, z_u \cdot P_2)^{k_1 \sum_{i=1}^n r_i} \\
E^{k_2} &= e(P_1, P_2)^{k_2 z_u \sum_{i=1}^n r_i} = CK_2 = \prod_{i=1}^n C'_{i,2} = e(P_1, z_u \cdot P_2)^{k_2 \sum_{i=1}^n r_i} \\
w'_1 &= C_1^{sk_u^{-1}} = e(P_1, Z_u)^{k_1 r z_u^{-1}} = e(P_1, P_2)^{k_1 r} \\
w'_2 &= C_2^{sk_u^{-1}} = e(P_1, Z_u)^{k_2 r z_u^{-1}} = e(P_1, P_2)^{k_2 r}
\end{aligned}$$

4 安全性分析

本节给出 BAOC-Twin-SM9 密钥封装机制的安全性证明。

定理 1. 令密码杂凑函数 H_1 、 H_2 为随机预言器,

若 q -BDHI 困难问题假设成立, 则本文提出的 BAOC-Twin-SM9 标识密钥封装机制关于定义 2 和定义 3 是 RCCA 安全的。

证明. 假设存在多项式时间概率攻击算法 \mathcal{A} , 其攻破 BAOC-Twin-SM9 密钥封装机制的优势 ε 是不可忽略的, 则可设计一个多项式时间的概率模拟算法 \mathcal{B} , 其成功求解 q -BDHI 问题的概率为 $\frac{\varepsilon}{q_{H_1}}$ 。

输入一个既定的 q -BDHI 问题实例 $(Q_1, Q_2, rQ_2, r^2Q_2, \dots, r^qQ_2)$, $e(Q_1, Q_2)^{\frac{1}{r}}$ 是其求解目标 (r 未知), $q = q_{H_1}$ 为对随机预言器 H_1 的询问次数。

初始化. \mathcal{B} 选取不同的随机数 $z^*, x, y, k_c, \gamma \in \mathbb{Z}_{\mathbb{N}}^*$, 在 r 未知的前提下隐含地设置 $k_1 = r - z^*$,

$k_2 = x + yk_1$. \mathcal{B} 随后随机选择 $i^* \in [1, q]$, 并从 \mathbb{Z}_N^* 选取 $q-1$ 个两两不同于 z^* 的随机数 $z_1, z_2, \dots, z_{i^*-1}, z_{i^*+1}, \dots, z_q$. 定义

$$f(e) = \prod_{i=1, i \neq i^*}^q (e - z^* + z_i) = \sum_{i=0}^{q-1} c_i e^i \mod N,$$

$$f_i(e) = \frac{f(e)}{e - z^* + z_i} = \sum_{i=0}^{q-2} b_i e^i \mod N,$$

$$\text{则 } f(r)Q_2 = \sum_{i=0}^{q-1} c_i (r^i Q_2), \quad rf(r)Q_2 = \sum_{i=0}^{q-1} c_i (r^{i+1} Q_2),$$

$$f_i(r)Q_2 = \sum_{i=0}^{q-2} b_i (r^i Q_2), \quad rf_i(r)Q_2 = \sum_{i=0}^{q-2} b_i (r^{i+1} Q_2)$$

均可根据已知问题实例计算得到。然后计算

$$P_2 = \sum_{i=0}^q c_i (r^i Q_2) \mod N = f(r)Q_2,$$

$$P_1 = \varphi(P_2) = f(r)Q_1,$$

$$P_{pub} = \varphi(rf(r)Q_2) - z^* \varphi(P_2) = k_1 P_1, \quad K_c = k_c P_2, \quad K_2 = \gamma P_2,$$

$$g_0 = e(P_1, P_2), \quad g_1 = e(P_{pub}, P_2) = e(P_1, P_2)^{k_1},$$

$$g_2 = e(P_1, P_2)^x \cdot e(P_{pub}, P_2)^y = e(P_1, P_2)^{x+yk_1} = e(P_1, P_2)^{k_2}.$$

令 $pp = (P_1, P_2, P_{pub}, g_0, g_1, g_2)$, 输出 (pp, K_c, k_c)

并返回给攻击算法 \mathcal{A} . \mathcal{A} 选择一个随机数 $k_c \in \mathbb{Z}_N^*$, 其中 k_c 为云服务中心私钥, 并公开云服务中心公钥 $pk_c = K_c = k_c P_2$, 模拟算法 \mathcal{B} 在证明过程中掌控 H_1 和 H_2 为随机谕言器。系统主公钥的构造表明, pp 中的元素均可由已知问题实例计算得出。

哈希询问阶段. \mathcal{A} 允许对随机谕言器 H_1 和 H_2 进行如下询问。

(1) H_1 -询问. \mathcal{B} 通过给定标识 ID_i 首先创建列表 \mathcal{L}_1 记录 H_1 询问的输入和输出, 元素以 (ID, z) 形式存储到 \mathcal{L}_1 中。令第 i 个 H_1 询问为 ID_i , 如果 \mathcal{L}_1 中存在 ID_i , 则将对应的 z_i 输出。否则设

$$H_1(ID_i) = \begin{cases} z^*, & i = i^* \\ z_i, & i \neq i^* \end{cases}$$

最终 \mathcal{B} 返回给 \mathcal{A} 为 z_i 或 z^* , 同时 \mathcal{L}_1 以 (ID_i, z_i) 或者 (ID_{i^*}, z_{i^*}) 形式更新。

(2) H_2 -询问. \mathcal{B} 首先创建列表 \mathcal{L}_2 记录 H_2 询问的输入和输出, 元素以 $(C_0, C_1, w_1, w_2, ID, K)$ 形式存储到列表中。令第 i 个 H_2 询问为 $(C_{0,i}, C_{1,i}, w_{1,i}, w_{2,i}, ID_i)$, 若 \mathcal{L}_2 中存在 $(C_{0,i}, C_{1,i}, w_{1,i}, w_{2,i}, ID_i)$, 则返

回对应的 K_i . 否则根据以下条件对 \mathcal{A} 进行回复。

- 若 $ID_i = ID_{i^*}$. \mathcal{B} 先从 \mathcal{L}_1 中获取 z^* ($z^* = H_1(ID_{i^*})$), (若不存在, 输入 ID_i 并询问 H_1 获取 z^*), 判断如下等式是否成立:

$$w_{2,i} = \left(\frac{e(C_{1,i}, P_2)}{w_{1,i}} \right)^{\frac{x}{z^*}} \cdot w_{1,i}^y. \quad (1)$$

- 若等式(1)成立, 则检查元素 $(C_{1,i}, ID_i)$ 是否存在于解密询问所创建的列表 \mathcal{L}_D 中。若元素存在, 则输出列表 \mathcal{L}_D 中对应的 K_i . 若不存在, 则输入 $(C_{1,i}, ID_i)$ 进行密文解密询问并输出 K_i , 随后令 $H_2(C_{1,i} \| w_{1,i} \| w_{2,i} \| ID_i) = K_i$.

- 若等式(1)不成立, 则 \mathcal{B} 从 $\{0, 1\}^{klen}$ 中随机选取一个元素 K_i , 随后令 $H_2(C_{1,i} \| w_{1,i} \| w_{2,i} \| ID_i) = K_i$.

最后 \mathcal{B} 发送 K_i 给 \mathcal{A} 并更新 \mathcal{L}_2 为 $(C_{0,i}, C_{1,i}, w_{1,i}, w_{2,i}, ID_i, K_i)$.

询问阶段(一). 在这一阶段, \mathcal{B} 创建一个空表 T 和整型计数器 $j = 0$, \mathcal{A} 适应性地进行如下询问。

(1) 创建标识询问. 当从攻击者 \mathcal{A} 获得带有标识为 ID 的解密私钥询问请求时, \mathcal{B} 设置 $j = j + 1$, 将接收到的标识 ID 发送给挑战者, 并获得解密密钥 $sk_{ID}' = (K_0', K_1', K_2')$. 接下来 \mathcal{B} 随机选择 z_u' , $\gamma' \in \mathbb{Z}_N^*$ 并计算 $Z_u = z_u' P_2$, $S_u = g_0^{z_u'}$. 对应数据使用者的公钥为 $pk_u = (Z_u, S_u)$, 数据使用者主私钥为 $sk_u = z_u'$. \mathcal{B} 计算 $K_0' = \gamma' \cdot K_c + k_1 \cdot s_1^{-1} \cdot Z_u$, $K_1' = \gamma' \cdot K_c + k_2 \cdot s_1^{-1} \cdot Z_u$, $K_2' = \gamma' \cdot P_2$. 令 $sk_{ID} = (K_0', K_1', K_2')$, \mathcal{B} 设置外包密钥为 $tk_{ID} = sk_{ID}$, 并将 $(j, ID, sk_{ID}, tk_{ID}, pk_u, sk_u)$ 插入到表 T 中。

(2) 解密私钥询问. 已知标识 ID_i , 如果 $ID_i = ID_{i^*}$, \mathcal{B} 停止模拟并输出失败, 否则计算

$$K_{0,i} = \gamma k_c (r - z^* + z_i) f_i(r) Q_2 + z_u (r - z^*) f_i(r) Q_2$$

$$= \gamma k_c f(r) Q_2 + \frac{z_u (r - z^*) f(r)}{r - z^* + z_i} Q_2$$

$$= \gamma K_c + \frac{k_1}{H_1(ID_i) + k_1} Z_u,$$

$$K_{1,i} = \gamma k_c(r - z^* + z_i)f_i(r)Q_2 + z_u(x + y(r - z^*))f_i(r)Q_2$$

$$= \gamma k_c f(r)Q_2 + \frac{z_u(x + y(r - z^*))f(r)}{r - z^* + z_i}Q_2$$

$$= \gamma K_c + \frac{k_2}{H_1(ID_i) + k_1}Z_u,$$

$$K_{2,i} = \gamma(r - z^* + z_i)f_i(r)Q_2 = \gamma f(r)Q_2 = \gamma P_2.$$

令 $sk_{ID_i} = (K_{0,i}, K_{1,i}, K_{2,i})$, \mathcal{B} 检查第 i 个元组 $(i, ID, sk_{ID_i}, tk_{ID_i}, pk_u, sk_u)$ 是否存在于表 T 中, 如果不存在则返回为 \perp , 否则将 (sk_{ID_i}, pk_u, sk_u) 返回攻击者 \mathcal{A} , 显然 sk_{ID_i} 满足正确私钥的条件。

(3) 外包密钥询问. \mathcal{B} 检查第 i 个元组 (i, ID, tk_{ID_i}) 是否存在于表 T 中, 如果不存在, 则 \mathcal{B} 返回为 \perp , 否则返回外包密钥 tk_{ID_i} 。

(4) 密文解密询问. 已知密文 $(C_{1,i}, ID_i)$, 模拟算法 \mathcal{B} 创建列表 \mathcal{L}_D 并对密文解密询问的输入和输出进行记录, 列表 \mathcal{L}_D 中元素存储形式为 (C_1, ID, K) 。若 \mathcal{L}_D 中存在 $(C_{1,i}, ID_i)$, 则将对应的 K_i 输出, 否则根据以下条件对 \mathcal{A} 进行回复。

- $ID_i \neq ID_{i^*}$. \mathcal{B} 首先计算标识 ID_i 的私钥 sk_{ID_i} , 然后输入 $(sk_{ID_i}, C_{1,i})$ 并执行解密算法输出 $(w_{1,i}, w_{2,i})$, 随后输入 $(C_{1,i}, w_{1,i}, w_{2,i}, ID_i)$ 对 H_2 随机预言器进行询问得到 K_i 并返回 \mathcal{A} 为 K_i 。
 - $ID_i = ID_{i^*}$. \mathcal{B} 从 $\{0, 1\}^{klen}$ 中随机选取元素 K_i 作为解密结果, 并返回给 \mathcal{A} 。
- 最后 \mathcal{L}_D 以 $(C_{1,i}, ID_i, K_i)$ 形式更新。

挑战阶段. 待挑战标识 ID^* 被 \mathcal{A} 输出后, 若 $ID^* \neq ID_i$, 则 \mathcal{B} 模拟失败终止, 否则有 $H_1(ID^*) = z^*$. \mathcal{B} 接下来随机选取 $s \in \mathbb{Z}_N^*$, $K^* \in \{0, 1\}^{klen}$, 并计算挑战密文 $C_0^* = S_u^s$, $C_1^* = sP_1$, 安全模型规定 (C_0^*, C_1^*, K^*) 没有进行过解密询问操作, 否则 \mathcal{B} 重新选取 s , 最终返回 \mathcal{A} 为 (C_0^*, C_1^*, K^*) 。令输出的挑战密文的随机数为 $s^* = \frac{s}{r}$, 有

$$C_0^* = (S_u^r)^{s^*} = S_u^{\frac{rs}{r}} = S_u^s,$$

$$C_1^* = s^*(H_1(ID^*)P_1 + P_{pub}) = \frac{s}{r}(z^*P_1 + (r - z^*)P_1) = sP_1.$$

故由随机数 s^* 生成的挑战密文 C_0^* , C_1^* 是有效的。

询问阶段(二). \mathcal{A} 允许继续进行私钥询问和密文解密询问, 但不能进行挑战标识 ID^* 的私钥询问和挑战密文 (C_0^*, C_1^*, ID^*) 的解密询问, \mathcal{B} 根据询问阶段(一)的结果对 \mathcal{A} 进行回复。

猜测阶段. \mathcal{A} 最后输出其猜测结果。此时, \mathcal{B} 忽略 \mathcal{A} 的猜测结果, 并定义 H_2 的挑战询问为 $(C_0^*, C_1^*, w_1^*, w_2^*, ID^*)$, 其中 $w_1^* = e(P_1, P_2)^{s^*k_1}$, $w_2^* = e(P_1, P_2)^{s^*k_2}$ 。接着, \mathcal{B} 从列表 \mathcal{L}_2 中找到满足等式的 w_1^* 和 w_2^*

$$w_2^* = \left(\frac{e(sP_1, P_2)}{w_1^*}\right)^{\frac{x}{z^*}} \cdot (w_1^*)^y \quad (2)$$

则有

$$w_1^* = e(P_1, P_2)^{s^*k_1} = e(f(r)Q_1, f(r)Q_2)^{\frac{s^*}{r}(r-z^*)}$$

$$= e(f(r)Q_1, f(r)Q_2)^s \cdot e(Q_1, Q_2)^{\frac{-sz^*f^2(r)}{r}}.$$

$$\text{又 } e \square f^2(e), \frac{-sz^*f^2(e)}{e} = T(e) + \frac{d}{e}, \text{ 其中 } T(e)$$

是一个 $2q-3$ 次多项式, d 是一个可求的非零整数。而 $\varphi(Q_2) = Q_1$, $f(r)Q_1 = \varphi(f(r)Q_2)$, 故 $e(Q_1, Q_2)^{T(r)}$ 是可求的。模拟算法 \mathcal{B} 最终计算

$$\left(\frac{w_1^*}{e(P_1, P_2)^s \cdot e(Q_1, Q_2)^{T(r)}}\right)^{\frac{1}{d}}$$

$$= \left(\frac{e(f(r)Q_1, f(r)Q_2)^s \cdot e(Q_1, Q_2)^{T(r) + \frac{d}{r}}}{e(P_1, P_2)^s \cdot e(Q_1, Q_2)^{T(r)}}\right)^{\frac{1}{d}}$$

$$= (e(Q_1, Q_2)^{\frac{d}{r}})^{\frac{1}{d}} = e(Q_1, Q_2)^{\frac{1}{r}}$$

作为 q -BDHI 困难问题实例的解。

注意以上证明过程中元素都是随机选取, 模拟环境和实际攻击是不可区分的。设 ID^* 是第 i^* 个询问 H_1 的挑战标识, 则攻击算法 \mathcal{A} 没有对 ID^* 发起过私钥询问, 此时模拟成功的概率为 $\frac{1}{q_{H_1}}$ 。

如果 \mathcal{A} 没有对 $(C_0^*, C_1^*, w_1^*, w_2^*, ID^*)$ 发起过 H_2 询问, 则 \mathcal{A} 攻破 BAOC-Twin-SM9 密钥封装机制的优势是可忽略的。根据假设, 若 \mathcal{A} 能以不可忽略的优势 ε 攻破算法, 则 \mathcal{A} 能以 ε 概率对 $(C_0^*, C_1^*, w_1^*, w_2^*, ID^*)$ 发起询问得到相应的 H_2 。根据等式(2), \mathcal{B} 可成功找到 (w_1^*, w_2^*) 并求出给定 q -BDHI 问题的解。综上所述, \mathcal{B} 求解 q -BDHI 困难问题成功

的概率为 $\frac{\varepsilon}{q_{H_1}}$ 。

本节中提出的 BAOC-Twin-SM9 密钥封装机制关于定义 3(即类型-2 攻击者)的安全性证明与 BAOC-Twin-SM9 密钥封装机制关于定义 2 的安全性证明基本相同, 除了 Setup_c 算法由攻击算法 \mathcal{A} 运行。

定理 2. 令密码杂凑函数 H_1 、 H_2 为随机谕言器, 若 q -BDHI 困难问题假设成立, 则攻击者在 BAOC-Twin-SM9 密钥封装机制关于定义 4 是可审计安全的。

证明. 假定存在一个攻击算法 \mathcal{A} 能攻破 BAOC-Twin-SM9 密钥封装系统中的可审计性, 则攻击算法 \mathcal{A} 与模拟算法 \mathcal{B} 之间的可审计性安全游戏交互如下。

初始化. \mathcal{B} 选取不同的随机数 $z^*, x, y, k_c, \gamma \in \mathbb{Z}_N^*$, 在 r 未知的前提下隐含地设置 $k_1 = r - z^*$, $k_2 = x + yk_1$ 。 \mathcal{B} 随后随机选择 $i^* \in [1, q]$, 并从 \mathbb{Z}_N^* 选取 $q-1$ 个两两不同于 z^* 的随机数 $z_1, z_2, \dots, z_{i^*-1}, z_{i^*+1}, \dots, z_q$ 。定义

$$f(e) = \prod_{i=1, i \neq i^*}^q (e - z^* + z_i) = \sum_{i=0}^{q-1} c_i e^i \mod N,$$

$$f_i(e) = \frac{f(e)}{e - z^* + z_i} = \sum_{i=0}^{q-2} b_i e^i \mod N,$$

$$\text{则 } f(r)Q_2 = \sum_{i=0}^{q-1} c_i (r^i Q_2), \quad rf(r)Q_2 = \sum_{i=0}^{q-1} c_i (r^{i+1} Q_2),$$

$$f_i(r)Q_2 = \sum_{i=0}^{q-2} b_i (r^i Q_2), \quad rf_i(r)Q_2 = \sum_{i=0}^{q-2} b_i (r^{i+1} Q_2)$$

均可根据已知问题实例计算得到。然后计算

$$P_2 = \sum_{i=0}^q c_i (r^i Q_2) \mod N = f(r)Q_2,$$

$$P_1 = \varphi(P_2) = f(r)Q_1,$$

$$P_{pub} = \varphi(rf(r)Q_2) - z^* \varphi(P_2) = k_1 P_1, \quad K_c = k_c P_2, \quad K_2 = \gamma P_2,$$

$$g_0 = e(P_1, P_2), \quad g_1 = e(P_{pub}, P_2) = e(P_1, P_2)^{k_1},$$

$$g_2 = e(P_1, P_2)^x \cdot e(P_{pub}, P_2)^y = e(P_1, P_2)^{x+yk_1} = e(P_1, P_2)^{k_2}.$$

令 $pp = (P_1, P_2, P_{pub}, g_0, g_1, g_2)$, 输出 (pp, K_c, k_c)

并返回给攻击算法 \mathcal{A} 。 \mathcal{A} 选择一个随机数 $k_c \in \mathbb{Z}_N^*$, 其中 k_c 为云服务中心私钥, 并公开云服务中心公钥 $pk_c = K_c = k_c P_2$, 模拟算法 \mathcal{B} 在证明过程中掌控 H_1 和 H_2 为随机谕言器。系统主公钥的构造表明, pp 中的元素均可由已知问题实例计算得出。

询问阶段(一). \mathcal{A} 适应性地发起解密私钥询问和外包密钥询问。由于 \mathcal{B} 知道系统主私钥 msk , 故能正确回答攻击者 \mathcal{A} 的询问。

挑战阶段. 待挑战标识 ID^* 被 \mathcal{A} 输出后, 若 $ID^* \neq ID_i$, 则 \mathcal{B} 模拟失败终止, 否则有 $H_1(ID^*) = z^*$ 。 \mathcal{B} 接下来随机选取 $s \in \mathbb{Z}_N^*$, $K^* \in \{0, 1\}^{klen}$, 并计算挑战密文 $C_0^* = S_u^s$, $C_1^* = sP_1$, 安全模型规定 (C_0^*, C_1^*, K^*) 没有进行过解密询问操作, 否则 \mathcal{B} 重新选取 s , 最终返回 \mathcal{A} 为 (C_0^*, C_1^*, K^*) 。令输出的挑战密文的随机数为 $s^* = \frac{s}{r}$, 有

$$C_0^* = (S_u^r)^{s^*} = S_u^{\frac{rs}{r}} = S_u^s,$$

$$C_1^* = s^* (H_1(ID^*)P_1 + P_{pub}) = \frac{s}{r} (z^* P_1 + (r - z^*)P_1) = sP_1.$$

故由随机数 s^* 生成的挑战密文 C_0^* , C_1^* 是有效的。

询问阶段(二). \mathcal{A} 允许继续对私钥和密文发起解密询问, 但不能对挑战标识 ID^* 和挑战密文 (C_0^*, C_1^*, ID^*) 分别进行私钥和解密询问, \mathcal{B} 根据询问阶段(一)的结果对 \mathcal{A} 进行回复。

输出阶段. 攻击算法 \mathcal{A} 输出 C_1^* 两个外包密文 C_1^* 和 C_2^* , \mathcal{A} 只有在以下条件都满足的情况下才能赢得游戏。

- (1) $\text{Audit}(CT^*, C_1^*, msk) \rightarrow 1$;
- (2) $\text{Audit}(CT^*, C_2^*, msk) \rightarrow 1$;
- (3) $\text{Dec}_u(C_1^*, CT^*, sk_u) \neq \text{Dec}_u(C_2^*, CT^*, sk_u)$ 。

条件 (1) 是指 $C_1^* = (C_{1,1}', C_{1,2}')$, $E^{k_1} = C_{1,1}'$, $E^{k_2} = C_{1,2}'$, 条件 (2) 是指 $C_2^* = (C_{2,1}', C_{2,2}')$, $E^{k_1} = C_{2,1}'$, $E^{k_2} = C_{2,2}'$, 根据条件 1 和条件 2, 有 $E^{k_1} = C_{1,1}' = C_{2,1}'$, $E^{k_2} = C_{1,2}' = C_{2,2}'$ 。然而, 条件(3)是指 $C_1^* \neq C_2^*$, 即 $(C_{1,1}', C_{1,2}') \neq (C_{2,1}', C_{2,2}')$, 与条件(1)和条件(2)矛盾。因此, 攻击算法 \mathcal{A} 赢得审计安全游戏的优势是可忽略的。

5 算法性能分析

本节对 BAOC-Twin-SM9 密钥封装机制从通信和计算开销两个方面进行性能分析, 并与文献[24], [26]进行对比。符号解释: 加法群 \mathbb{G}_1 、 \mathbb{G}_2 和乘法群 \mathbb{G}_T 中元素的大小分别用 $|\mathbb{G}_1|$ 、 $|\mathbb{G}_2|$ 和 $|\mathbb{G}_T|$ 表示, T_p 双线

性配对运算, 加法群 \mathbb{G}_1 、 \mathbb{G}_2 中的倍乘运算分别用 T_{sm_1} 、 T_{sm_2} 表示, 乘法群 \mathbb{G}_T 中的指数运算用 T_{e_i} 表示, \mathcal{H}_Z 表示映射到 \mathbb{Z}_N^* 的密码哈希函数, \mathcal{H}_K 表示映射到 $\{0,1\}^{klen}$ 的密码哈希函数, 表 2 和表 3 即为对比结果。

5.1 算法性能理论分析

由表 2, 3 所示, 本文提出的 BAOC-Twin-SM9 密钥封装算法在系统公钥和密钥生成方面, 相比于 Twin-SM9 密钥封装算法^[26]均分别小幅度增加 1 个群元素, 通过将耗时较大的配对运算全部安全外包至算力强大的云端处理, 使得终端解密计算开销由原先 Twin-SM9 中 2 个耗时较大的配对运算降低至 2 个简单的指数运算, 解密效率得到有效提升。

表 2 通信代价比较

Table 2 Comparison of communication costs

算法	系统公钥	密钥生成	密文生成
SM9 密钥封装机制 ^[24]	$2 \mathbb{G}_1 + 1 \mathbb{G}_2 + 1 \mathbb{G}_T $	$1 \mathbb{G}_2 $	$1 \mathbb{G}_1 $
Twin-SM9 ^[26]	$2 \mathbb{G}_1 + 1 \mathbb{G}_2 + 2 \mathbb{G}_T $	$2 \mathbb{G}_2 $	$1 \mathbb{G}_1 $
BAOC-Twin-SM9	$2 \mathbb{G}_1 + 1 \mathbb{G}_2 + 3 \mathbb{G}_T $	$3 \mathbb{G}_2 $	$1 \mathbb{G}_1 + 1 \mathbb{G}_T $

表 3 计算开销比较

Table 3 Comparison of computational costs

算法	密钥生成	密文生成	终端解密
SM9 密钥封装机制 ^[24]	$1T_{sm_2} + 1\mathcal{H}_Z$	$2T_{sm_1} + 1T_{e_i} + 1\mathcal{H}_Z + 1\mathcal{H}_K$	$1T_p + 1\mathcal{H}_K$
Twin-SM9 ^[26]	$2T_{sm_2} + 1\mathcal{H}_Z$	$2T_{sm_1} + 2T_{e_i} + 1\mathcal{H}_Z + 1\mathcal{H}_K$	$2T_p + 1\mathcal{H}_K$
BAOC-Twin-SM9	$3T_{sm_2} + 1\mathcal{H}_Z$	$2T_{sm_1} + 3T_{e_i} + 1\mathcal{H}_Z + 1\mathcal{H}_K$	$2T_{e_i} + 1\mathcal{H}_K$

5.2 算法性能实验数据分析

本节通过编程仿真对方案进行实现, 并与文献[24]和文献[26]中每个算法的运行时间进行对比。本文使用 Miracl 大数库, 采用的椭圆曲线为 256 位的 BN 曲线, 选取 R-ate 双线性对, 经过 100 次测试取平均值得到实验中在乘法群 \mathbb{G}_T 上执行一次双线性配

对运算并输出群元素所需时间约为 524.71ms, 在乘法群 \mathbb{G}_T 上执行一次指数运算所需时间约为 192.25ms, 在加法群 \mathbb{G}_1 、 \mathbb{G}_2 上执行一次标量乘运算所需时间各分别约为 54.06ms 和 128.17ms。仿真实验中客户端所用的设备是一台 12GB 内存, CPU 为 Intel(R) Core(TM) i5-4210U CPU@1.70GHz 2.40GHz, 64 位 Windows 8.0 操作系统的笔记本电脑, 服务端所用的设备是一台 64 位 Ubuntu 20.04 LTS 操作系统, CPU 为 Intel® Xeon(R) Gold 6428 CPU@2.50GHz×80, 251.5GiB 内存, 25.0TB 硬盘的应用服务器, 使用 C 语言编程。通过分别生成 1 份和 100 份密文, 并先后对 1 份和 100 份外包结果进行批量审计测试得到的结果如表 4、5、6、7 所示。从表 4、5、6、7 可以看出, 本方案在牺牲额外 492.75ms 的批量审计开销外, 终端解密时间约为 39215ms, 相比于 SM9 和 Twin-SM9 密钥封装机制的 52716ms 和 107792ms 的终端解密时间, 分别降低了 25.61%和 63.62%。图 2 表明, 随着密文量的增长, 本文提出的方案在通过算力强大的云服务中心完成大量耗时配对操作后, 本地计算时间增长速度明显低于文献[26], 从而更适用于计算资源有限但又需要快速响应大量解密请求的环境中(如密码、传感器芯片等), 故本文方案是实用有效的, 实验结果与理论分析一致。

表 4 算法客户端 1 次解密时间比较

Table 4 Comparison of client 1 time decryption time

算法	终端解密/ms
SM9 密钥封装机制 ^[24]	534.38
Twin-SM9 ^[26]	1085.99
BAOC-Twin-SM9	397.27

表 5 算法客户端 100 次解密时间比较

Table 5 Comparison of client 100 times decryption time

算法	终端解密/ms
SM9 密钥封装机制 ^[24]	52716
Twin-SM9 ^[26]	107792
BAOC-Twin-SM9	39215

表 6 算法其他 1 次仿真时间比较

Table 6 Comparison of other 1 time simulation time

算法	数据使用者初始化/ms	密钥生成/ms	加密/ms	解密外包/ms	批量审计/ms
SM9 密钥封装机制 ^[24]	—	121.86	303.23	—	—
Twin-SM9 ^[26]	—	253.26	487.90	—	—
BAOC-Twin-SM9	342.07	373.41	664.75	33.86	375.68

表 7 算法其他 100 次仿真时间比较

Table 7 Comparison of other 100 times simulation time

算法	数据使用者初始化/ms	私钥生成/ms	加密/ms	解密外包/ms	批量审计/ms
SM9 密钥封装机制 ^[24]	—	121.86	31763	—	—
Twin-SM9 ^[26]	—	253.26	48573	—	—
BAOC-Twin-SM9	342.07	373.41	67127	3418.47	492.75

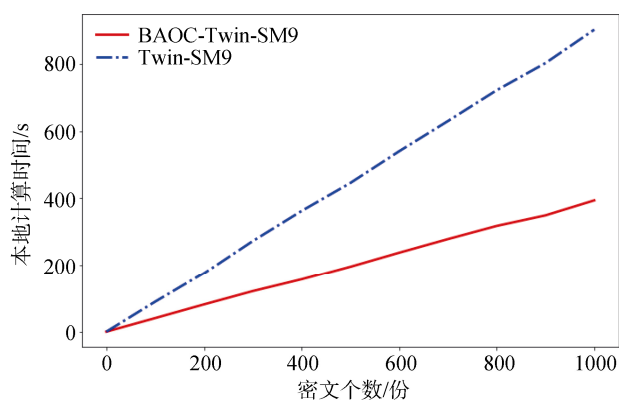


图 2 本地计算时间对比

Figure 2 Comparison of local computation time

6 结论

本文提出了一种适用于对海量数据进行高并发解密操作并支持多密文批量审计的 Twin-SM9 密钥封装机制。该方案可将解密过程当中耗时较大的配对操作安全外包至算力强大的云服务器中心计算,并以 KGC 作为审计机构对云服务中心端返回的多份外包结果进行批量审计。本文在随机谕言模型下基于 q -BDHI 困难假设证明了方案具备 RCCA 安全性,并通过实验仿真与现有 SM9 密钥封装机制相关方案的解密效率进行对比。由对比实验数据结果可知,本文算法中数据使用者的解密耗时明显低于其他现有方案,因此,本文提出的算法是切实可行的。

参考文献

- [1] Shamir A. Identity-based cryptosystems and signature schemes[C]. In: *Proceedings of Workshop on the Theory and Application of Cryptographic Techniques*, 1984: 47-53.
- [2] Boneh D, Franklin M K. Identity-Based Encryption from the Weil Pairing[C]. *The 21st Annual International Cryptology Conference on Advances in Cryptology*, 2001: 213-229.
- [3] BOYEN X. Multipurpose identity-based signcryption (A Swissarmy knife for identity-based cryptography)[C]. *Advances in Cryptology—CRYPTO 2003*, 2003:383-399.
- [4] Boneh D, Boyen X. Efficient selective-id secure identity-based encryption without random oracles[C]. In: *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, 2004: 223-238.
- [5] BONEH D, BOYEN X, GOH E J. Hierarchical identity based encryption with constant size ciphertext[C]. *Advances in Cryptology—EUROCRYPT 2005*, 2005: 440-456.
- [6] WATERS B. Efficient identity-based encryption without random oracles[C]. *Advances in Cryptology—EUROCRYPT 2005*, 2005: 114-127.
- [7] Gentry C. Practical Identity-Based Encryption without Random Oracles[C]. *The 24th annual international conference on The Theory and Applications of Cryptographic Techniques*, 2006: 445-464.
- [8] DELERABLÉE C. Identity-based broadcast encryption with constant size ciphertexts and private keys[C]. *Advances in Cryptology—ASIACRYPT 2007*, 2007: 200-215.
- [9] YAMADA S. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters[C]. *Advances in Cryptology—EUROCRYPT 2016, Part II*, 2016: 32-62.
- [10] Hofheinz D, Jia D D, Pan J X. Identity-Based Encryption Tightly Secure under Chosen-Ciphertext Attacks[C]. *Advances in Cryptology - ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security*, 2018: 190-220.
- [11] Ge A J, Wei P W. Identity-Based Broadcast Encryption with Efficient Revocation[C]. *IACR International Workshop on Public Key Cryptography*, 2019: 405-435.
- [12] Xue J T, Xu C X, Zhao J N, et al. Identity-Based Public Auditing for Cloud Storage Systems Against Malicious Auditors via Blockchain[J]. *Science China Information Sciences*, 2019, 62(3): 32104.
- [13] Qin B D, Liu X M, Wei Z, et al. Space Efficient Revocable IBE for Mobile Devices in Cloud Computing[J]. *Science China Information Sciences*, 2020, 63(3): 139110.
- [14] Cryptography Standardization Technical Committee. SM9 identity-based cryptographic algorithm. GM/T0044-2016. <http://www-w.gmbz.org.cn/main/postDetail.html?id=20180322410400>.
- [15] Zhang X F, Peng H. Blind Signature Scheme Based on SM9 Algorithm[J]. *Netinfo Security*, 2019(8): 61-67.
(张雪锋, 彭华. 一种基于 SM9 算法的盲签名方案研究[J]. *信息安全*, 2019(8): 61-67.)
- [16] Wang S, Fang L G, Han L B, et al. Fast Implementation of SM9 Digital Signature and Verification Algorithms[J]. *Communications Technology*, 2019, 52(10): 2524-2527.
(王松, 房利国, 韩炼冰, 等. 一种 SM9 数字签名及验证算法的快速实现方法[J]. *通信技术*, 2019, 52(10): 2524-2527.)
- [17] Xu S W, Ren X P, Yuan F, et al. A Secure Key Issuing Scheme of SM9[J]. *Computer Applications and Software*, 2020, 37(1): 314-319.
(许盛伟, 任雄鹏, 袁峰, 等. 一种关于 SM9 的安全密钥分发方案[J]. *计算机应用与软件*, 2020, 37(1): 314-319.)

- [18] Wang C J, Lin W L, Lin H T. Design of an Instant Messaging System Using Identity Based Cryptosystems[C]. *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, 2013: 277-281.
- [19] Geisler M, Smart N P. Distributing the Key Distribution Centre in Sakai—Kasahara Based Systems[C]. *The 12th IMA International Conference on Cryptography and Coding*, 2009: 252-262.
- [20] Yang Y T, Cai J L, Zhang X W, et al. Privacy Preserving Scheme in Block Chain with Provably Secure Based on SM9 Algorithm[J]. *Journal of Software*, 2019, 30(6): 1692-1704.
(杨亚涛, 蔡居良, 张筱薇, 等. 基于 SM9 算法可证明安全的区块链隐私保护方案[J]. *软件学报*, 2019, 30(6): 1692-1704.)
- [21] Sun S Z, Ma H, Zhang R, et al. Server-Aided Immediate and Robust User Revocation Mechanism for SM9[J]. *Cybersecurity*, 2020, 3(1): 1-13.
- [22] Gan Z W, Liao F Y. Rapid Calculation of R-Ate Bilinear Pairing in China State Cryptography Standard SM9[J]. *Computer Engineering*, 2019, 45(6): 171-174.
(甘植旺, 廖方圆. 国密 SM9 中 R-ate 双线性对快速计算[J]. *计算机工程*, 2019, 45(6): 171-174.)
- [23] Wang M D, He W G, Li J, et al. Optimal design of R-ate pair in SM9 algorithm[J]. *Commun Technol*, 2020, 53:2241-2244.
(王明东, 何卫国, 李军, 等. 国密 SM9 算法 R-ate 对计算的优化设计[J]. *通信技术*, 2020, 53: 2241-2244.)
- [24] Cheng Z H. Security analysis of SM9 key agreement and encryption[C]. In: *Proceedings of the 14th International Conference Information Security and Cryptology*, 2018: 3-25.
- [25] Chen L Q, Cheng Z H. Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme[C]. *The 10th international conference on Cryptography and Coding*, 2005: 442-459.
- [26] Lai J C, Huang X Y, He D B, et al. Security Analysis of SM9 Digital Signature and Key Encapsulation Algorithm for State Secret[J]. *Science China Information Science*, 2021, 51(11): 1900-1913.
(赖建昌, 黄欣沂, 何德彪, 等. 国密 SM9 数字签名和密钥封装算法的安全性分析[J]. *中国科学信息科学*, 2021, 51(11): 1900-1913.)
- [27] Cash D, Kiltz E, Shoup V. The twin Diffie-Hellman problem and applications[C]. In: *Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2008: 127-145.
- [28] Mathew Green, Susan Hohenberger, Brent Waters. Outsourcing the decryption of ABE ciphertexts[C]. In *20th USENIX Security Symposium*, 2011: 1-16.



刘宽 于 2018 年在天津工业大学获得工学学士学位。现在福建师范大学攻读网络空间安全硕士学位。研究领域为密码学与信息安全。研究兴趣包括: 公钥密码学、数据安全。Email: dawn_lk@126.com



宁建廷 于 2016 年在上海交通大学计算机科学与技术专业获得博士学位。现任福建师范大学计算机与网络空间安全学院教授、博士生导师。研究领域为密码学、信息安全。研究兴趣包括: 应用密码学、数据安全。Email: jtning88@gmail.com



伍玮 于 2011 年在澳大利亚卧龙岗大学信息安全专业获得博士学位。现任福建师范大学数学与统计学院教授。研究领域为密码学、信息安全。研究兴趣包括: 密码学、信息安全。Email: weiwu81@gmail.com



陈海霞 于 2022 年在福建师范大学网络空间安全专业获得博士学位。现任福建师范大学数学与统计学院副教授。研究领域为密码学与信息安全。研究兴趣包括: 图像信息隐藏、图像数据认证。Email: tummy7882@hotmail.com