支持多密文批量审计的解密外包 SM9-HIBE 密钥封装机制

刘宽[1]，宁建廷[1,2,3]，伍玮[3,4]，许胜民[1,2]，林超[1,2]

（1. 福建师范大学计算机与网络空间安全学院，福建 福州 350117；2. 福建省网络安全与密码技术重点实验室，福建 福州 350117；
3. 福建师范大学分析数学及应用教育部重点实验室，福建 福州 350117；4. 福建师范大学数学与统计学院，福建 福州 350117）

摘 要：SM9-HIBE 密钥封装机制的解密操作需要 2 次双线性配对运算，在设备算力受限且需对大规模信息资源进行高频解密时，配对运算的高额计算开销会束缚系统的有效部署。为此，基于 SM9-HIBE 提出了一种支持解密外包和多密文批量审计的新型密钥封装机制 OASM9-HIBE，并利用 Fujisaki-Okamoto 转换技术在随机谕言模型下证明了 OASM9-HIBE 具备 RCCA 安全性。OASM9-HIBE 将计算繁重的双线性配对运算全部安全外包至算力强大的云端，第 k 层用户只需执行一次简单的指数运算即可完成最终解密，有效提升了原 SM9-HIBE 的解密效率，OASM9-HIBE 同时运用密钥盲化技术实现了多份转换密文的高效批量审计功能，从而拓展了 SM9 系列算法的应用领域。

关键词：分层密钥封装机制；解密外包；批量审计；密钥封装

中图分类号：TP309

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2023222

Multi-ciphertext batch auditable decryption outsourcing SM9-HIBE key encapsulation mechanism

LIU Kuan[1]，NING Jianting[1,2,3]，WU Wei[3,4]，XU Shengmin[1,2]，LIN Chao[1,2]

1. College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China
2. Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou 350117, China
3. Key Laboratory of Analytical Mathematics and Applications, Fujian Normal University, Fuzhou 350117, China
4. School of Mathematics and Statistics, Fujian Normal University, Fuzhou 350117, China

Abstract: The decryption operation of SM9-HIBE key encapsulation mechanism required two bilinear pairing operations, for the equipment requiring frequent decryption of massive data and with limited computing resources, such resourcing-consuming pairing operation will become an important bottleneck restricting the system deployment. To address the above issue, a decryption outsourcing key encapsulation mechanism OASM9-HIBE based on SM9-HIBE was proposed, which supported multi-ciphertext batch auditing. The Fujisaki-Okamoto transformation technology was utilized to prove the RCCA security of OASM9-HIBE under the random oracle model. All resourcing-consuming bilinear pairing operations were safely offloaded to the cloud server in OASM9-HIBE, the k-th hierarchical user only need to perform one simple exponentiation operation to complete the final decryption. The decryption efficiency of the original SM9-HIBE was effectively improved under the premise of not changing the downward proxy generation function of the user's private key between hierarchical. OASM9-HIBE additively achieved the property of batch auditing of multi-transformed ciphertexts by employing the key blinding technology. Theoretical analysis and evaluation of experimental data highlight the feasibility and efficiency of OASM9-HIBE, OASM9-HIBE extends the application field of SM9 series algorithms.

Keywords: hierarchical key encapsulation mechanism, decryption outsourcing, batch audit, key encapsulation

收稿日期：2023−07−18；**修回日期**：2023−10−23

通信作者：宁建廷，jtning88@gmail.com

基金项目：国家自然科学基金资助项目（No.61972094, No.62372108, No.62102090, No.62102089, No.U21A20466）

Foundation Item: The National Natural Science Foundation of China (No.61972094, No.62372108, No.62102090, No.62102089, No.U21A20466)

0　引言

Shamir 等[1]开创性地提出了一种灵活且通用的公钥设置方式，通过利用标识（如个人住址、证件编号、出生日期等）作为用户公钥，省略了可信第三方参与公钥发布与认证时所带来的繁重计算校验和更新过程，从而有效提升了系统效率。随后，Boneh 等[2]首次基于双线性对技术，设计出满足可证明安全且实用的标识加密（IBE, identity-based encryption）方案后，一系列基于双线性对的标识密码学开始受到学术界和工业界的广泛关注。然而，传统标识密码体制中系统用户的私钥依赖于密钥生成中心（KGC, key generation center）进行生成和发送，当系统用户的数量增加时，针对大量系统用户的标识校验、私钥生成和传递等工作就会让 KGC 负载过重，从而引发网络拥塞使网络传输性能下降，系统实时性和效率降低。显然，单个 KGC 的标识密码体系不适于在大型网络应用中部署。

为有效摆脱单个 KGC 对系统用户私钥管理的束缚，Horwitz 和 Lynn[3]于 2002 年首次给出了分层标识加密（HIBE, hierarchical identity-based encryption）的定义。在 HIBE 系统中，系统的总层数默认大于 1（总层数为 1 时与 IBE 等价），随着层数的向下递进，用户标识的长度也随之增加，每层用户的私钥均可由上层用户（节点 KGC）通过系统全局参数和本身私钥向下代理生成，并可用于系统用户对密文进行解密，从而与单个 KGC 的私钥生成达到了相同目的。与传统标识加密系统相比，HIBE 系统中，用户标识校验、私钥生成和传递工作由下层节点 KGC 承担，而顶层 KGC 只需向下一层 KGC 传递私钥，以用户私钥逐层向下派生的形式，有效减缓了顶层 KGC 的工作负载，系统整体性能得到了显著改善。HIBE 同时能有效降低传统标识加密系统中因单 KGC 主私钥泄露而存在的系统安全风险，若某层节点 KGC 的私钥被恶意窃取，只会对该节点 KGC 下层用户的私钥生成产生威胁，而其他节点的用户私钥生成依然能够顺利执行，系统的健壮性也因此得到了有效提升，但上述方案中的用户私钥及密文长度、加解密计算开销都会随着层数线性增长。为此，Boneh 和 Boyen[4]构造了标准模型下满足选择身份和选择明文攻击下的不可区分性（IND-sID-CPA）安全的 HIBE 方案，该方案实现了常量级密文长度（3 个群元素组成），解密开销也独立于系统最大层数。Waters [5]则引入了对偶加密技术，基于判定性 BDH（bilinear Diffie-Hellman）假设和判定性线性假设提出了具有完全安全的 HIBE 方案。Lewko 和 Waters[6]基于对偶加密设计了不限制系统最大层数且基于标准模型满足完全安全的分层标识加密算法。文献[7]首次提出具有匿名性的 HIBE 方案，并基于标准模型证明了该方案满足 IND-sID-CPA 安全性，然而其密文长度与系统最大层数相关。Langrehr 等[8]则首次提出了基于标准模型的具有紧归约的分层标识加密方案。随后，文献[9]基于矩阵形式的判定性 Diffie-Hellman 假设给出了在多挑战安全模型下具有紧归约的 HIBE 方案，分层标识加密中的紧归约问题在文献[10-12]中得到了进一步研究。

然而，现有的标识加密方案大多基于国外密码标准构造与扩展，为推动信息安全技术体系实现自主可控，我国先后对基于 SM9 的密钥封装机制、公钥加密、数字签名和密钥协商系列标识密码算法[13]进行研制，有效保障了国家和网络信息安全，并取得一系列重要的研究成果[14-16]。然而 SM9 标识密码算法中的系统用户私钥生成依赖 KGC 完成，当系统用户数目增多时，依旧存在单个 KGC 工作负载过重而使系统性能降低的缺陷。为此，赖建昌等[17]基于标准 SM9 密钥封装机制的核心要素，融合分层标识加密技术[18]，首次提出基于 SM9 的分层密钥封装机制—— SM9-HIBE，并依据随机谕言机证明了该方案具备选择身份下不可区分抵抗选择明文攻击（sID-CPA）的安全性。

然而，SM9-HIBE 密钥封装机制在解密时包含 2 次计算高昂的配对操作，其解密耗时随着待解密文档数（或解密频次）的增加而呈线性增长，在计算设备算力受限且需对海量信息进行高频解密时（如小型终端设备、芯片等），配对运算将会阻遏系统部署的有效应用。

本文提出一种新型分层密钥封装机制（OASM9-HIBE, decryption outsourcing and batch auditable key encapsulation mechanism based on SM9-HIBE），以实现原 SM9-HIBE 的解密外包和多密文批量审计，并基于随机谕言机使其满足可重放选择密文攻击（RCCA, replayable chosen ciphertext

attack）[19]安全性。实验数据表明，OASM9-HIBE 对一份和 100 份密文的审计时间分别约为 10.68 ms 和 11.33 ms，对应终端解密时间分别约为 11.07 ms 和 12.58 s，而原 SM9-HIBE 的终端解密时间分别约为 78.63 ms 和 79.27 s。显然，以 100 份密文为例，OASM9-HIBE 额外增加了 11.33 ms 批量审计开销，使原方案 SM9-HIBE 的最终解密时间降低了约 84.13%。OASM9-HIBE 同时具备以下特性：1) 解密服务的安全外包；2) 多份转换密文的批量审计；3) 密钥泄露防护；4) 更强的 RCCA 安全性。具体如下。

1) 解密服务的安全外包。在不改变 SM9-HIBE 层级间用户密钥向下代理生成性质的条件下，OASM9-HIBE 将其解密中耗时较长的配对运算安全外包至云端处理，并利用指数盲化方法使云端基于外包转换密钥获取不到任何关键内容，算力受限的第 k 层用户主要通过一次指数操作就能完成最终解密，重建明文的成本得到有效降低。

2) 多份转换密文的批量审计。KGC 作为审计实体，在没有扩充附加设置的前提下，对于云端返回的多份转换密文，其只需执行一次简单的指数运算即可完成转换密文的批量审计，从而以较小的代价有效降低审计机制对系统解密效率的影响。

3) 密钥泄露防护。第 k 层用户在与云端关联过程中，即使其解密密钥被恶意实体获取，其对转换密文进行部分解密后仍然被第 k 层用户的私钥所盲化。当第 k 层用户不与云端关联时，恶意实体依据其解密密钥完成最终解密后得到的结果将被云端私钥和终端用户私钥所盲化。

4) 更强的 RCCA 安全性。本文利用 Fujisaki-Okamoto 转换方法[20]，通过在系统初始化阶段增加 2 个密码哈希函数作为随机谕言机对原有满足 CPA 安全性的 OASM9-HIBE 方案进行拓展，使其具备更强的 RCCA 安全性。

OASM9-HIBE 和 SM9-HIBE [17]密钥封装机制的功能性和安全性比较结果如表 1 所示。其中，√表示具备此性质，×表示不具备此性质。从表 1 可知，基于相同的(q, n)-DBDHI 困难问题假设条件，OASM9-HIBE 分别从解密外包和批量审计两方面对 SM9-HIBE 进行功能性拓展，使其更适于算力受限但又涉及计算成本高的强并发读写操作实际场景中。

表 1 功能性和安全性比较

密钥封装机制	解密外包	批量审计	困难问题假设	安全性
SM9-HIBE	×	×	(q,n)-DBDHI	sID-CPA
OASM9-HIBE	√	√	(q,n)-DBDHI	RCCA

1 预备知识

1.1 双线性群

已知安全参数为 λ，给定循环群 $\mathcal{G}_1$、$\mathcal{G}_2$ 和 $\mathcal{G}_T$（群阶均为 p），则双线性映射 $e:\mathcal{G}_1\times\mathcal{G}_2\to\mathcal{G}_T$ 应具备如下特性。

1) 双线性：给定生成元 $u\in\mathcal{G}_1$，$v\in\mathcal{G}_2$ 和 $a,b\in\mathcal{Z}_p$，满足 $e(au,bv)=e(u,v)^{ab}$。

2) 非退化性：至少存在元素 $u\in\mathcal{G}_1$，$v\in\mathcal{G}_2$，满足 $e(u,v)\neq 1$。

3) 可计算性：给定任意的 $u\in\mathcal{G}_1$，$v\in\mathcal{G}_2$，则 $e(u,v)$ 可由有效的多项式时间算法求解得到。

1.2 困难问题假设

(q,n)-DBDHI 问题。已知双线性群 $\mathrm{BP}=(\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,p)$，随机选取 $a,b,c\in\mathbb{Z}_p$，计算 $(c,P_1,bP_1,aP_1,a^2P_1,\cdots,a^{n+1}P_1,P_2,bP_2,aP_2,a^2P_2,\cdots,a^qP_2,\frac{1}{(a+c)^2}P_2,\frac{1}{(a+c)^3}P_2,\cdots,\frac{1}{(a+c)^n}P_2)$ 和选取元素 $T\in\mathcal{G}_T$，判断 $T=e(P_1,P_2)^{\frac{b}{a+c}}$ 是否成立，或者 T 是否为群 $\mathcal{G}_T$ 中的一个随机元素。

1.3 OASM9-HIBE 系统架构、方案定义和安全模型

1.3.1 系统架构

OASM9-HIBE 系统共包含 5 个关键实体，分别是密钥生成中心（KGC）、云存储中心、云端、数据管理者和第 k 层用户，如图 1 所示。系统操作流程分为以下步骤。

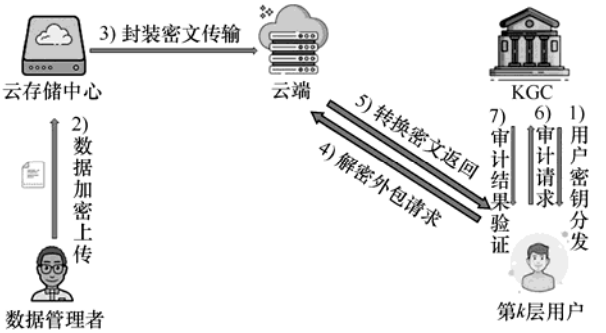


图 1 OASM9-HIBE 系统架构

1) 用户密钥分发阶段。KGC 首先利用其生成的系统全局参数为第 k 层用户生成解密密钥，以确保第 k 层用户能通过自身解密密钥生成外包转换密钥，并传送至云端实现对加密数据的部分解密。

2) 数据加密上传阶段。数据管理者将待上传的明文数据进行加密生成封装密文，随后将封装密文传送至云存储中心。

3) 封装密文传输阶段。云存储中心在接收到加密后的封装密文后，将其传输到算力强大的云端。

4) 解密外包请求阶段。第 k 层用户向云端发起解密外包请求以访问需要解密的封装密文。

5) 转换密文返回阶段。云端在收到第 k 层用户的解密外包请求后，对封装密文进行部分解密，然后将其生成的转换密文结果返回给第 k 层用户。

6) 审计请求阶段。KGC 作为可信实体响应第 k 层用户的审计请求，通过对云端生成的转换密文的正确性进行审计，以确保加密数据在被云端解密后没有被篡改或损坏，审计结果最终返回给第 k 层用户。

7) 审计结果验证阶段。只有在 KGC 审计通过（转换密文无误）后，第 k 层用户才能完成最终解密获取有效的封装密钥，从而以较低的成本完成最终的解密。

1.3.2 OASM9-HIBE 方案定义

OASM9-HIBE方案共包含如下9个多项式时间算法。

1) $\text{Setup}(\lambda, n)\to(\text{pp},\text{msk})$。系统初始化算法，由可信实体 KGC 运行。给定系统安全参数 λ 和分层加密系统中层数的最大值 n，即用户标识长度的最大值，算法输入 λ 和 n，输出系统全局参数 pp 和主私钥 msk，其中，pp 公开，msk 由 KGC 秘密存储。

2) $\text{Setup}_{\text{c}}(\lambda, \text{pp})\to(\text{pk}_{\text{c}},\text{sk}_{\text{c}})$。云端初始化算法，由云端运行。算法输入安全参数 λ 和系统全局参数 pp，输出云端主公私钥对 $(\text{pk}_{\text{c}},\text{sk}_{\text{c}})$，其中，$\text{pk}_{\text{c}}$ 公开，sk_{c} 由云端秘密存储。

3) $\text{Setup}_{\text{u}}(\lambda, \text{pp})\to(\text{pk}_{\text{u}},\text{sk}_{\text{u}})$。第 k 层用户初始化算法，由第 k 层用户运行。算法输入安全参数 λ 和系统全局参数 pp，输出第 k 层用户主公私钥对 $(\text{pk}_{\text{u}},\text{sk}_{\text{u}})$，其中，$\text{pk}_{\text{u}}$ 公开，sk_{u} 由第 k 层用户秘密存储。

4) $\text{KeyGen}(\text{pp}, \text{pk}_{\text{c}},\text{pk}_{\text{u}},\text{msk},\text{ID}|_k, d_{\text{ID}|_{k-1}})\to d_{\text{ID}|_k}$。第 k 层用户解密密钥生成算法，由其上层用户（第 k–1 层用户）运行。算法输入系统全局参数 pp、云端公钥 pk_{c}、第 k 层用户公钥 pk_{u} 及其标识 $\text{ID}|_k=(\text{ID}_1,\cdots,\text{ID}_k)$、第 k–1 层用户解密密钥 $d_{\text{ID}|_k}$（$1<k\leqslant n$），输出 $d_{\text{ID}|_k}$ 为第 k 层用户解密密钥，其中第 k–1 层用户的私钥可通过上层用户派生或者由顶层 KGC 直接生成。

5) $\text{KeyGen}_{\text{out}}(d_{\text{ID}|_k})\to\text{tk}$。外包转换密钥生成算法，由第 k 层用户运行。算法输入第 k 层用户解密密钥 $d_{\text{ID}|_k}$，输出外包转换密钥 tk。

6) $\text{Encap}(\text{pp},\text{pk}_{\text{u}},\text{ID}|_k)\to(K,\text{CT})$。密钥封装算法，由数据管理者运行。算法输入系统全局参数 pp、第 k 层用户公钥和第 k 层用户标识 $\text{ID}|_k$，输出封装密钥 K 和封装密文 CT。

7) $\text{Transform}(\text{CT},\text{tk},\text{sk}_{\text{c}})\to C'$。解密外包算法，由云端运行。算法输入密文 CT、外包转换密钥 tk 和云端私钥 sk_{c}，输出转换密文 C'。

8) $\text{Verify}(\{\text{CT}_i\}_{i\in[m]},\{C_i'\}_{i\in[m]},\text{msk})\to1$ 或 0。转换密文批量审计算法，由可信实体 KGC 运行。算法输入封装密文集合 $\{\text{CT}_i\}_{i\in[m]}$、转换密文集合 $\{C_i'\}_{i\in[m]}$ 和系统主私钥 msk（m 为密文份数），如果云端正确地执行解密外包算法，则算法输出 1；否则输出 0。

9) $\text{Decrypt}(C',\text{CT},\text{sk}_{\text{u}})\to K'$。本地解密算法，由第 k 层用户运行。算法输入转换密文 C'、封装密文 CT 和第 k 层用户主私钥 sk_{u}，若 $\text{Verify}(\text{CT}, C',\text{msk})\to1$，则算法输出为封装密钥 K'；否则，算法输出为 $\perp$。

1.3.3 OASM9-HIBE 安全模型

基于 1.3.1 节的系统架构，对 OASM9-HIBE 的安全模型（由挑战者和敌手之间的交互游戏所得）进行定义，共包含以下两类敌手。

1) 敌手-I。该敌手表示恶意的第 k 层用户。针对敌手-I，该方案必须确保没有未经许可的第 k 层用户能够成功解密云存储中心中的密文，即使多个未经许可的第 k 层用户合谋，依旧得不到有效批准实施解密。

2) 敌手-II。该敌手表示半可信的云端实体。针对敌手-II，其目标是在系统设置的基本流程内，获取关键的秘密信息。规定其安全性是在执行过程中不被许可获取任何关键的秘密信息，即敌手-II 不能确定哪个第 k 层用户正在访问云端的密文数据或提出解密外包服务申请。

针对上述两类敌手，本文提出了 OASM9-HIBE 在可重放选择密文攻击（RCCA）下的安全性。在安全游戏中，挑战者和敌手都以系统层数上限 n 作为输入，安全游戏定义如下。

RCCA。类似于文献[19]，RCCA 安全是选择密文攻击安全性的一种削弱分支，即可以无恶意修订原密文，通过以下两类敌手对系统 RCCA 安全性进行定义。

敌手-I 的 RCCA 安全。由挑战者和敌手之间交互的安全游戏来定义，安全游戏描述如下。

初始化。敌手输出挑战标识 $\mathrm{ID}^* = (\mathrm{ID}_1^*, \mathrm{ID}_2^*, \cdots, \mathrm{ID}_m^*)$，其中 $1 < m \leqslant n$，n 为系统层数上限。

系统建立阶段。给定挑战者系统安全参数 λ，挑战者运行 Setup 算法并输出系统主公私钥对 $(\mathrm{pp}, \mathrm{msk})$，pp 被发送给敌手。

询问阶段 1。敌手适应性地发起关于标识 ID_i 的解密私钥询问（其中 $\mathrm{ID}_i \neq \mathrm{ID}^*$ 且 ID_i 不是 ID^* 的前缀），挑战者执行第 k 层用户私钥生成算法 KeyGen 并输出私钥 d_{ID_i}，随后 d_{ID_i} 被发送给敌手。

1) 创建标识阶段。挑战者设置 $j = j + 1$，执行 $\mathrm{Setup_u}$ 算法以获取第 k 层用户的公私钥对 $(\mathrm{pk_u}, \mathrm{sk_u})$，随后将标识 $\mathrm{ID}|_k$ 作为输入，执行 KeyGen 算法，生成解密密钥 $d_{\mathrm{ID}|_k}$ 并传送至敌手。

2) 解密私钥询问阶段。挑战者对表 T 进行核实以确定第 i 个五元组 $(i, \mathrm{ID}, d_{\mathrm{ID}|_k}, \mathrm{pk_u}, \mathrm{sk_u})$ 是否存在，当其不存在时，返回敌手为 $\perp$。反之挑战者设定 $D = D \cup \{\mathrm{ID}\}$，运行 KeyGen 算法生成私钥 $d_{\mathrm{ID}|_k}$，最终返还敌手为 $(d_{\mathrm{ID}|_k}, \mathrm{pk_u}, \mathrm{sk_u})$。

3) 外包转换密钥询问阶段。挑战者对表 T 进行核实以确定第 i 个三元组 $(i, \mathrm{ID}, \mathrm{tk})$ 是否存在，当其不存在时，返还敌手为 $\perp$，否则将外包转换密钥 tk 返还给敌手。

4) 密文解密询问阶段。挑战者对表 T 进行核实以确定第 i 个三元组 $(i, \mathrm{ID}, d_{\mathrm{ID}|_k})$ 是否存在，当其不存在时，返回敌手为 $\perp$，反之对密文 CT 进行解密并返还给敌手。

挑战阶段。待询问阶段 1 结束后，挑战者运行加密算法 $\mathrm{Encap}(\mathrm{mpk}, \mathrm{ID}^*)$ 并将 (K^*, CT^*) 输出（挑战封装密钥和密文），随后选择随机比特 $b \in \{0,1\}$，令 $K_b = K^*$，从封装密钥空间中随机选取一个会话密钥并将其指定为 K_{1-b}，并返还 $(\mathrm{CT}^*, K_0, K_1)$ 给敌手。

询问阶段 2。敌手针对挑战者标识 ID_i 继续发起解密私钥询问（$\mathrm{ID}_i \neq \mathrm{ID}^*$ 且 ID_i 不属于 ID^* 的前缀），挑战者的回应与询问阶段 1 一致。

猜测阶段。敌手将关于 ξ 的猜测 $\xi' \in \{0,1\}$ 输出，当 $\xi' = \xi$ 时表示敌手获胜。敌手在游戏中获胜的优势由等式 $\mathrm{Adv}_{\mathbb{A}}^{\mathrm{RCCA}}(\lambda) = \left| \Pr[\xi' = \xi] - \frac{1}{2} \right|$ 定义。

定义 1 若 OASM9-HIBE 密钥封装机制关于敌手-I 满足 RCCA 安全性，则不存在一个概率多项式时间的敌手-I 能以不可忽略的优势 $\mathrm{Adv}_{\mathbb{A}}^{\mathrm{RCCA}}(\lambda)$ 赢得上述安全游戏。

敌手-II 的 RCCA 安全性。安全性交互游戏描述与敌手-I 安全性交互游戏基本一致，其中系统建立阶段调整如下。

系统建立阶段。挑战者执行 Setup 算法生成系统全局参数 pp 并返还给敌手，敌手运行 $\mathrm{Setup_c}$ 算法生成云端公钥 $\mathrm{pk_c}$ 并返还给挑战者。

定义 2 若 OASM9-HIBE 密钥封装机制关于敌手-II 满足 RCCA 安全性，则不存在一个概率多项式时间的敌手-II 能以不可忽略的优势 $\mathrm{Adv}_{\mathbb{A}}^{\mathrm{RCCA}}(\lambda)$ 赢得上述安全游戏。

可审计性。云端生成的转换密文需要经过审计以验证其正确性，以防止恶意云端对同一密文解密输出 2 种不同的有效转换密文。可审计性由挑战者和敌手之间的交互游戏来定义。与关于敌手-II 的安全模型相比，猜测阶段调整如下。

输出阶段。敌手输出一个密文元组 $(C_1'^*, C_2'^*)$，$(C_1'^*, C_2'^*)$ 为 C'^* 的 2 个转换密文。假定元组 $(\mathrm{ID}^*, \mathrm{sk}_{\mathrm{ID}^*}, \mathrm{tk}_{\mathrm{ID}^*}, \mathrm{pk_u}, \mathrm{sk_u})$ 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 $\mathrm{Verify}(\mathrm{CT}^*, C_1'^*, \mathrm{msk}) \to 1 \wedge \mathrm{Verify}(\mathrm{CT}^*, C_2'^*, \mathrm{msk}) \to 1 \wedge \mathrm{Dec_u}(C_1'^*, \mathrm{sk_u}) \neq \mathrm{Dec_u}(C_2'^*, \mathrm{sk_u})$ 同时成立时，敌手才能在上述安全游戏中获胜。

定义 3 若 OASM9-HIBE 密钥封装机制具备可审计安全性，则不存在一个概率多项式时间敌手能以不可忽略的优势 $\mathrm{Adv}_{\mathbb{A}}^{\mathrm{RCCA}}(\lambda)$ 赢得上述安全游戏。

2 OASM9-HIBE 密钥封装机制

本节借鉴外包计算技术，将 SM9-HIBE 中的配

对操作统一交付给算力充足的云端处理，并实现对多份转换密文的批量审计，随后分别给出满足 CPA 安全和 RCCA 安全的 OASM9-HIBE 密钥封装机制的完整描述，并对方案的正确性进行验证。

2.1　算法描述

2.1.1　CPA 安全系统构造

$\text{Setup}(\lambda, n)\to(\text{pp},\text{msk})$：系统初始化算法。KGC 输入安全参数 λ 和分层密钥封装机制中系统最大层数 n（用户标识长度上限），选取非对称双线性群 $D=(\mathcal{G}_1,\mathcal{G}_2,\mathcal{G}_T,e,N)$，其中 N 为大素数且满足 $N>2^{\lambda}$，随机选择生成元 $P\in\mathcal{G}_1$，$Q, Q_1, Q_2, \cdots, Q_n\in\mathcal{G}_2$，$\alpha\in\mathcal{Z}_p^*$，定义密码哈希函数 $H:\mathcal{Z}_p^*\to\mathcal{Z}_p^*$ 和密钥派生函数 KDF：$\{0,1\}^*\to\{0,1\}^l$（l 为封装密钥的比特长度）。计算 $P_{\text{pub}}=\alpha P$，$u=e(P,Q)$，$v=e(P_{\text{pub}},Q)$。系统全局参数 pp 和主私钥 msk 为 $\text{pp}=(D,P,P_{\text{pub}},Q,Q_1,Q_2,\cdots,Q_n,u,v,H,\text{KDF},n,l)$，$\text{msk}=\alpha$。

$\text{Setup}_{\text{c}}(\lambda,\text{pp})\to(\text{pk}_{\text{c}},\text{sk}_{\text{c}})$：云端初始化算法。算法选取随机数 $k_{\text{c}}\in[1,N-1]$，设置云端主私钥为 $\text{sk}_{\text{c}}=k_{\text{c}}$，计算群 $\mathcal{G}_2$ 中的元素 $K_{\text{c}}=k_{\text{c}}Q$，令 $\text{pk}_{\text{c}}=K_{\text{c}}$，最后输出云端公私钥对 $(\text{pk}_{\text{c}},\text{sk}_{\text{c}})$。

$\text{Setup}_{\text{u}}(\lambda,\text{pp})\to(\text{pk}_{\text{u}},\text{sk}_{\text{u}})$：第 k 层用户初始化算法。算法输入安全参数 λ 和系统全局参数 pp，选取随机数 $y_{\text{u}}\in[1,N-1]$ 作为第 k 层用户私钥，令 $\text{sk}_{\text{u}}=y_{\text{u}}$，计算群 $\mathcal{G}_2$ 中的元素 $Z_{\text{u}}=y_{\text{u}}Q$，群 $\mathcal{G}_T$ 中的元素 $S_{\text{u}}=g_0^{y_{\text{u}}}$，设置 $\text{pk}_{\text{u}}=(Z_{\text{u}},S_{\text{u}})$ 作为第 k 层用户公钥，输出第 k 层用户公私钥对 $(\text{pk}_{\text{u}},\text{sk}_{\text{u}})$。

$\text{KeyGen}(\text{pp},\text{pk}_{\text{c}},\text{pk}_{\text{u}},\text{msk},\text{ID}|_k,d_{\text{ID}|_{k-1}})\to d_{\text{ID}|_k}$：第 k 层用户解密密钥生成算法。算法输入系统全局参数 pp，云端公钥 $\text{pk}_{\text{c}}=K_{\text{c}}$，第 k 层用户公钥 $\text{pk}_{\text{u}}=(Z_{\text{u}},S_{\text{u}})$ 以及第 k 层用户的标识集合 $\text{ID}|_k=(\text{ID}_1,\text{ID}_2,\cdots,\text{ID}_k)\in(\mathbb{Z}_p^*)^k$（$1<k\leqslant n$），顶层 KGC 选取随机数 $r,\gamma\in[1,N-1]$ 并通过系统主私钥 $\text{msk}=\alpha$ 计算第 k 层用户私钥

$$d_{\text{ID}|_k}=\left(\frac{\alpha}{\alpha+H(\text{ID}_1)}Z_{\text{u}}+\gamma K_{\text{c}}+r\left(Q_1+\sum_{i=2}^{k}H(\text{ID}_k)Q_k\right),\right.$$
$$\left.r(\alpha+H(\text{ID}_1))P,\gamma Q,rQ_{k+1},rQ_{k+2},\cdots,rQ_n\right)=$$
$$(d_1,d_2,d_3,u_{k+1},u_{k+2},\cdots,u_n)$$

第 k 层用户的私钥 $d_{\text{ID}|_k}$ 同样能依据上层（第 k–1 层）用户的私钥派生得到。不妨设第 k–1 层用户 $\text{ID}|_{k-1}=(\text{ID}_1,\text{ID}_2,\cdots,\text{ID}_{k-1})\in(\mathbb{Z}_p^*)^{k-1}$ 的私钥为

$$d_{\text{ID}|_{k-1}}=\left(\frac{\alpha}{\alpha+H(\text{ID}_1)}Z_{\text{u}}+\gamma' K_{\text{c}}+r'\left(Q_1+\sum_{i=2}^{k-1}H(\text{ID}_i)Q_i\right),\right.$$
$$\left.r'(\alpha+H(\text{ID}_1))P,\gamma' Q,r'Q_k,r'Q_{k+1},\cdots,rQ_n\right)=$$
$$(d_1',d_2',d_3',u_k',u_{k+1}',\cdots,u_n')$$

为构建 $d_{\text{ID}|_k}$，选择随机数 $t\in[1,N-1]$，计算

$$d_1=\left(d_1'+H(\text{ID}_k)u_k'+t\left(Q_1+\sum_{i=2}^{k}H(\text{ID}_k)Q_k\right)+K_{\text{c}}\right),$$
$$d_2=d_2'+t(P_{\text{pub}}+H(\text{ID}_1)P),d_3=d_3'+tQ,u_{k+1}=u_{k+1}'+$$
$$tQ_{k+1},\cdots,u_n=u_n'+tQ_n,$$

令 $d_{\text{ID}|_k}=(d_1,d_2,d_3,u_{k+1},u_{k+2},\cdots,u_n)$ 并输出，显然 $d_{\text{ID}|_k}$ 是第 k 层用户 $\text{ID}|_k=(\text{ID}_1,\text{ID}_2,\cdots,\text{ID}_k)\in(\mathbb{Z}_p^*)^k$ 的正确私钥，且生成私钥用的随机数为 $r=r'+t$，$\gamma=\gamma'+t\in[1,N-1]$。容易验证，用户私钥的长度随着层数增加而逐层递减，最后输出 $d_{\text{ID}|_k}$ 作为第 k 层用户解密密钥。

$\text{KeyGen}_{\text{out}}(d_{\text{ID}|_k})\to\text{tk}$：第 k 层用户外包转换密钥生成算法。算法输入第 k 层用户解密密钥 $d_{\text{ID}|_k}$，设置外包转换密钥为 $\text{tk}=d_{\text{ID}|_k}=\{d_1,d_2,d_3\}$ 并将其输出。

$\text{Encap}(\text{pp},\text{pk}_{\text{u}},\text{ID}|_k)\to(K,\text{CT})$：密钥封装算法。算法输入系统全局参数 pp，第 k 层用户公钥 $\text{pk}_{\text{u}}=(Z_{\text{u}},S_{\text{u}})$ 及其标识 $\text{ID}|_k$，为生成第 k 层用户的封装密钥（比特长度为 l），数据管理者选择随机数 $s\in[1,N-1]$，并构造哈希函数 $h=H(\text{ID}\|\text{hid},N)$，计算封装密文 CT 为 $C_1=s(H(\text{ID}_1)P+P_{\text{pub}})$，$C_2=s\left(Q_1+\sum_{i=2}^{k}H(\text{ID}_i)Q_i\right)$，关于 s 的承诺值密文 $C_3=S_{\text{u}}^s$，$w=v^s$，封装密钥 $K=\text{KDF}(C_1\|C_2\|w\|\text{ID}|_k,l)$，若 $K=0$，则算法重新选择随机数 s，最后令封装密文 $\text{CT}=(C_1,C_2,C_3)$，并输出 (K,CT)。

$\text{Transform}(\text{CT},\text{tk},\text{sk}_{\text{c}})\to C'$：解密外包算法。云端收到第 k 层用户的解密外包请求后，执行解密外包算法。算法输入封装密文 $\text{CT}=(C_1,C_2,C_3)$，外包转换密钥 $\text{tk}=\{d_1,d_2,d_3\}$，云端主私钥 $\text{sk}_{\text{c}}=k_{\text{c}}$，对密文进行部分解密生成转换密文 $C'=\dfrac{e(C_1,d_1)}{e(C_1,k_cd_3)e(d_2,C_2)}=e(P,Z_{\text{u}})^{\alpha s}$ 并将

其输出。

$\text{Verify}(\{\text{CT}_i\}_{i\in[m]},\{C'_i\}_{i\in[m]},\text{msk})\to 1$：转换密文批量审计算法。系统指定 KGC 为可信实体实现对 m 份转换密文的批量审计，算法输入系统主私钥 msk，封装密文集合 $\{\text{CT}_i\}_{i\in[m]}=\{C_{i,1},C_{i,2},C_{i,3}\}_{i\in[m]}$，转换密文集合 $\{C'_i\}_{i\in[m]}$，其中 m 为密文份数。对于封装密文集合 $\{C_{i,3}\}_{i\in[m]}$，计算 $E=\prod_{i=1}^{m}S_{\text{u}}^{s_i}=S_{\text{u}}^{\sum_{i=1}^{m}s_i}$；对于云服务器返回的转换密文集合 $\{C'_i\}_{i\in[m]}$，计算 $\text{CK}=\prod_{i=1}^{m}C'_i=e(P,Z_{\text{u}})^{\alpha\sum_{i=1}^{m}s_i}$，判断等式 $E^{\alpha}=\text{CK}$ 是否成立，成立则输出 1 代表云端解密正确，否则输出 0 代表云端解密错误。

$\text{Decrypt}(C',\text{CT},\text{sk}_{\text{u}})\to K'$：本地解密算法。算法输入转换密文 C'，封装密文 $\text{CT}=(C_1,C_2,C_3)$ 和第 k 层用户解密私钥 $\text{sk}_{\text{u}}=y_{\text{u}}$，当且仅当 $\text{Verify}(\text{CT},C',\text{msk})=1$（密文份数 m=1）时，计算 $w'=C'^{\text{sk}_{\text{u}}^{-1}}=e(P,Q)^{\alpha s}$，计算并输出密钥派生函数 $K'=\text{KDF}\left(C_1\|C_2\|w'\|\prod_{i=1}^{k}H(\text{ID}_i),l\right)$，最后第 k 层用户输出封装密钥 K'。

随后本文利用 Fujisaki-Okamoto 转换技术[20]对满足 CPA 安全性的 OASM9-HIBE 方案进行扩展，使其具备更强的 RCCA 安全性。与满足 CPA 安全性的方案相比，云端初始化算法、第 k 层用户初始化算法、第 k 层用户解密密钥生成算法、第 k 层用户外包转换密钥生成算法、转换密文批量审计算法保持不变，系统初始化算法需额外增加 2 个密码哈希函数 $H_1:\{0,1\}^*\to Z_p^*$，$H_2:\{0,1\}^*\to\{0,1\}^k$，剩余算法具体如 2.1.2 节所述。

2.1.2 RCCA 安全系统构造

$\text{Encap}_{\text{rcca}}(\text{pp},\text{pk}_{\text{u}},\text{ID}|_k)\to(K,\text{CT})$：密钥封装算法。算法输入系统全局参数 pp，第 k 层用户公钥 $\text{pk}_{\text{u}}=(Z_{\text{u}},S_{\text{u}})$ 和其标识 $\text{ID}|_k=(\text{ID}_1,\text{ID}_2,\cdots,\text{ID}_k)\in(\mathbb{Z}_p^*)^k$。为生成第 k 层用户的封装密钥（比特长度为 l），数据管理者选取随机数 $R\in G_T$，计算 $s=H_1(R,\text{ID})$，$r=H_2(R)$ 并构造哈希函数 $h=H(\text{ID}\|\text{hid},N)$，$C_1=s(H(\text{ID}_1)P+P_{\text{pub}})$，$C_2=s(Q_1+H(\text{ID}_2)Q_2+\cdots+H(\text{ID}_k)Q_k)$，$C_3=S_{\text{u}}^s$，$C_4=r\oplus\text{ID}$，$C_5=Re(P,Q)^{\alpha s}$，$w=e(P,Q)^{\alpha s}$，输出封装密钥 $K=\text{KDF}(C_1\|C_2\|w\|\text{ID}|_k,l)$，封装密文 $\text{CT}=(C_1,C_2,C_3,C_4,C_5)$。

$\text{Transform}_{\text{rcca}}(\text{CT},\text{tk},\text{sk}_{\text{c}})\to C'$：解密外包算法。云端响应第 k 层用户的解密外包需求，开始执行解密外包算法。算法输入封装密文 $\text{CT}=(C_1,C_2,C_3,C_4,C_5)$，外包转换密钥 $\text{tk}=\{d_1,d_2,d_3\}$，云端主私钥 $\text{sk}_{\text{c}}=k_{\text{c}}$，对 CT 进行部分解密，生成转换以下密文并将其输出。

$$C'=\frac{e(C_1,d_1)}{e(C_1,k_{\text{c}}d_3)e(d_2,C_2)}=e(P,Z_{\text{u}})^{\alpha s}$$

$\text{Decrypt}_{\text{rcca}}(C',\text{CT},\text{sk}_{\text{u}})\to K'$：本地解密算法。算法输入转换密文 C'，封装密文 $\text{CT}=(C_1,C_2,C_3,C_4,C_5)$ 和第 k 层用户解密私钥 $\text{sk}_{\text{u}}=y_{\text{u}}$，当且仅当 $\text{Verify}(\text{CT},C',\text{msk})=1$（$m=1$）时，计算 $w'=C'^{\text{sk}_{\text{u}}^{-1}}=e(P,Q)^{\alpha s}$，$R=\frac{C_5}{C'^{\text{sk}_{\text{u}}^{-1}}}$，$\text{ID}=C_4\oplus H_2(R)$ $s=H_1(R,\text{ID})$，解封装得到封装密钥 $K'=\text{KDF}\left(C_1\|C_2\|w'\|\prod_{i=1}^{k}H(\text{ID}_i),l\right)$ 并将其输出。

2.2 正确性验证

已知封装密文 $\text{CT}=(C_1,C_2,C_3)$ 和转换密文 C'，则 OASM9-HIBE 分层密钥封装机制的正确性可得到验证。

证明 详见附录 1。

3 安全性证明

定理 1 设密码哈希函数 H、H_1 和 H_2 为随机谕言机。若 $(q,n)-\text{DBDHI}$ 困难问题假设成立，则本文提出的 OASM9-HIBE 分层密钥封装机制关于定义 1 和定义 2 是 RCCA 安全的。

证明 不妨设攻击算法 $\mathcal{A}$，其攻破该方案的优势 ε 是不可忽略的，则可设计一个模拟算法 $\mathcal{B}$ 与 $\mathcal{A}$ 交互的安全游戏，使 $\mathcal{B}$ 求解 $(q,n)-\text{DBDHI}$ 难题的优势是不可忽略的。在 RCCA 安全模型中，给定模拟算法 $\mathcal{B}$ 为 $(q,n)-\text{DBDHI}$ 问题实例

$$(c,P_1,bP_1,aP_1,a^2P_1,\cdots,a^{n+1}P_1,P_2,bP_2,aP_2,a^2P_2,\cdots,$$
$$a^qP_2,\frac{1}{(a+c)^2}P_2,\frac{1}{(a+c)^3}P_2,\cdots,\frac{1}{(a+c)^n}P_2,T)$$

其目的是评估 $T=e(P_1,P_2)^{\frac{b}{a+c}}$ 是否成立。同时，

假定 $\mathcal{B}$ 和 $\mathcal{A}$ 均输入 q 和 n（q 为第一层标识数目，n 为系统层数上限）。

初始化阶段。$\mathcal{A}$ 将挑战者的标识 $\mathrm{ID}^* = (\mathrm{ID}_1^*, \mathrm{ID}_2^*, \cdots, \mathrm{ID}_m^*)$ 输出（$1 < m \leqslant n$）。

系统建立阶段。$\mathcal{B}$ 设置 $x^* = c$（设为 $H(\mathrm{ID}^*)$ 的值），选择互不相同的随机数 $x_2^*, x_3^*, \cdots, x_m^* \in \mathcal{Z}_p^*$（设为 $H(\mathrm{ID}_i^*)$ 的值，其中 $i = 2,3,\cdots,m$），选择随机数 $w_1, w_2, \cdots, w_q \in \mathcal{Z}_p^*$（设为第一层除 ID_1^* 外标识的哈希值），随后选择两两不同的随机数 $x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n \in \mathcal{Z}_p^*$，在 a 未知的情况下隐含地设置 $\alpha = a$，$\beta = (a + w_1)(a + w_2)\cdots(a + w_q) \mod p$，依据已知困难问题实例计算 $P = P_1$，$P_{\mathrm{pub}} = \alpha P_1$，$Q = \beta P_2$，$Q_i = x_i P_2 + \dfrac{y_i}{(a + x^*)^{n+2-i}} P_2$（$i \in [2,n]$），$Q_1 = x_1(a + x^*)P_2 - \sum_{i=2}^{m} x_i^* Q_i = x_1(aP_2 + x^* P_2) - \sum_{i=2}^{m} x_i^* Q_i$，$u = e(P,Q)$，$v = e(P_{\mathrm{pub}}, Q)$。最终 $\mathcal{B}$ 选择合适的密钥派生函数 KDF，将计算生成的系统主公钥 $\mathrm{pp} = (P, P_{\mathrm{pub}}, Q, Q_1, \cdots, Q_n, u, v, H, \mathrm{KDF}, n, l)$ 输出，其中密码哈希函数 H 被认定为 $\mathcal{B}$ 掌控的随机谕言机。显然，pp 中的元素均可通过已知困难问题实例运算获得。

哈希询问阶段。设询问的标识为 ID_i，$\mathcal{B}$ 首先建立哈希列表 $\mathcal{L}_H$，列表中元素为二元组 (ID_i, h_i)，并初始化为空。若 ID_i 在列表 $\mathcal{L}_H$ 中，则 $\mathcal{B}$ 返回相应的 h_i。否则，根据以下步骤回应 $\mathcal{A}$。

1) ID_i 是第一层用户的标识

a) 若 $\mathrm{ID}_i = \mathrm{ID}_1^*$，设 $h_i = H(\mathrm{ID}_1^*) = x^* = c$，将 h_i 发送给 $\mathcal{A}$ 并把新的二元组 (ID_i, h_i) 添加到列表 $\mathcal{L}_H$ 中，由困难问题实例中 a、b、c 的随机性可知，该设置满足哈希函数随机性的要求。

b) 若 $\mathrm{ID}_i \neq \mathrm{ID}_1^*$，设 $h_i = H(\mathrm{ID}_1^*) = w_i$，将 h_i 发送给 $\mathcal{A}$ 并把新的二元组 (ID_i, h_i) 添加到列表 $\mathcal{L}_H$ 中。

2) ID_i 不是第一层用户的标识

a) 若 $\mathrm{ID}_i = \mathrm{ID}_j^*$，$j \in [2,m]$，设 $h_i = H(\mathrm{ID}_j^*) = x_j^*$，将 h_i 发送给 $\mathcal{A}$ 并把新的二元组 (ID_i, h_i) 添加到列表 $\mathcal{L}_H$ 中。

b) 若 $\mathrm{ID}_i \neq \mathrm{ID}_j^*$，$j \in [2,m]$，随机选取 $z_i \in \mathcal{Z}_p^*$，设 $h_i = H(\mathrm{ID}_i) = z_i$，将 h_i 发送给 $\mathcal{A}$ 并把新的二元组 (ID_i, h_i) 添加到列表 $\mathcal{L}_H$ 中。

询问阶段 1。在该阶段，$\mathcal{A}$ 允许适应性地询问标识 $\mathrm{ID} = (\mathrm{ID}_1, \mathrm{ID}_2, \cdots, \mathrm{ID}_j)$ 的私钥，其中 $j \in [0,n]$（注意 $\mathrm{ID} \neq \mathrm{ID}^*$，且 ID 不是 ID^* 的前缀）。$\mathcal{B}$ 创建并初始化空表 T、T_1、T_2 和一个整型计数器 $j = 0$，并根据以下步骤进行对 $\mathcal{A}$ 进行回应。

$H_1(R, \mathrm{ID})$ 询问。如果列表 T_1 中存在元组 $(R, \mathrm{ID}|_k, s)$，则将 s 返回 $\mathcal{A}$。否则 $\mathcal{B}$ 选择一个随机数 $s \in \mathbb{Z}_p$，以 $(R, \mathrm{ID}|_k, s)$ 的形式记录在列表 T_1 中并将 s 返回给 $\mathcal{A}$。

$H_2(R)$ 询问。如果列表 T_2 中存在元组 (R, r)，则返回敌手 $\mathcal{A}$ 为 r。否则 $\mathcal{B}$ 选择一个随机数 $r \in \{0,1\}^k$，以 (R, r) 的形式记录在列表 T_2 中并将 r 返回给 $\mathcal{A}$。

创建标识询问。在 $\mathcal{A}$ 发起对标识 $\mathrm{ID}|_k$ 的解密私钥询问请求后，$\mathcal{B}$ 设定 $j = j + 1$ 并通过标识 $\mathrm{ID}|_k$ 计算解密密钥 $d'_{\mathrm{ID}|_k} = (d'_1, d'_2, d'_3, u'_{k+1}, u'_{k+2}, \cdots, u'_n)$。随后 $\mathcal{B}$ 随机选取 z'_{u}，$\gamma' \in \mathbb{Z}_N^*$ 并计算 $Z_{\mathrm{u}} = y'_{\mathrm{u}} Q$，$S_{\mathrm{u}} = u^{y'_{\mathrm{u}}}$，对应第 k 层用户的公钥为 $\mathrm{pk}_{\mathrm{u}} = (Z_{\mathrm{u}}, S_{\mathrm{u}})$，主私钥为 $\mathrm{sk}_{\mathrm{u}} = y'_{\mathrm{u}}$。$\mathcal{B}$ 计算

$$d_{\mathrm{ID}|_k} = \left(\frac{\alpha}{\alpha + H(\mathrm{ID}_1)} Z'_{\mathrm{u}} + \gamma' K_{\mathrm{c}} + r\left(Q_1 + \sum_{i=2}^{k} H(\mathrm{ID}_i) Q_i\right),\right.$$
$$\left. r(\alpha + H(\mathrm{ID}_1))P, \gamma' Q, rQ_{k+1}, rQ_{k+2}, \cdots, rQ_n\right) =$$
$$(d_1, d_2, d_3, u_{k+1}, u_{k+2}, \cdots, u_n)$$

且设置外包转换密钥为 $\mathrm{tk} = (d_1, d_2, d_3)$，并将 $(j, \mathrm{ID}|_k, d_{\mathrm{ID}|_k}, \mathrm{tk})$ 插入表 T 中。

解密私钥询问。如果对应第 i 个元组 $(i, \mathrm{ID}|_k, d_{\mathrm{ID}|_k}, \mathrm{tk})$ 存在于表 T 中，则 $\mathcal{B}$ 得到该元组后将 $\mathrm{ID}|_k$ 添加到集合 D 中，即 $D := D \cup \mathrm{ID}|_k$，并返回 $\mathcal{A}$ 为 $d_{\mathrm{ID}|_k}$，否则返回为空。

1) 当 $\mathrm{ID}_1 \neq \mathrm{ID}_1^*$ 时，$\mathcal{B}$ 设置 $j = j + 1$，并将接收到的标识 ID 发送给挑战者 $\mathcal{B}$，$\mathcal{B}$ 选择 $y'_{\mathrm{u}} \in [1, N-1]$，计算 $Z_{\mathrm{u}} = y'_{\mathrm{u}} Q = y'_{\mathrm{u}} \beta P_2$，$S_{\mathrm{u}} = e(P,Q)^{z'_{\mathrm{u}}} = e(P_1, P_2)^{\beta z'_{\mathrm{u}}}$，第 k 层用户公钥被设置为 $(Z_{\mathrm{u}}, S_{\mathrm{u}})$，第 k 层用户主私钥被设置为 $\mathrm{sk}_{\mathrm{u}} = y'_{\mathrm{u}}$。$\mathcal{B}$ 随后设定 $H(\mathrm{ID}_1)$ 的值为 w_i，根据参数设置 $\dfrac{\alpha}{\alpha + H(\mathrm{ID}_1)} Z_{\mathrm{u}} = \dfrac{a\beta y'_{\mathrm{u}}}{a + w_i} P_2$，而 $\beta = (a + w_1) \cdot (a + w_2)\cdots(a + w_q) \mod q$，则容易得到 $\dfrac{\alpha}{\alpha + H(\mathrm{ID}_1)} Z_{\mathrm{u}}$

的值可通过已知困难问题实例计算得到，最后根据私钥生成算法很容易计算出正确的私钥为 $d_{\mathrm{ID}|_k}=(d_1,d_2,d_3,u_{k+1},u_{k+2},\cdots,u_n)$。

2) 当 $\mathrm{ID}_1=\mathrm{ID}_1^*$ 时，因为 ID 不等于 ID^*，且不是 ID^* 的前缀，则至少存在 $k\in[2,j]$ 使 $\mathrm{ID}_k\neq\mathrm{ID}_k^*$（$k$ 是第一个不相等标识的下标），故有

$$\frac{\alpha}{\alpha+H(\mathrm{ID}_1^*)}Z_{\mathrm{u}}=\frac{a\beta y'_{\mathrm{u}}}{a+x^*}P_2=f(a)P_2+\frac{W}{a+x^*}P_2$$

其中，$f(a)$ 为 q 次多项式，$W\neq 0$ 且可求。又因为 $(\mathrm{ID}_1,\mathrm{ID}_2,\cdots,\mathrm{ID}_{k-1})=(\mathrm{ID}_1^*,\mathrm{ID}_2^*,\cdots,\mathrm{ID}_{k-1}^*)$，对 $i\in[2,n]$，$Q_i=x_iP_2+\dfrac{y_i}{(a+c)^{n+2-i}}P_2$，故

$$Q_1+\sum_{i=2}^{j}H(\mathrm{ID}_i)Q_i=x_1(aP_2+x^*P_2)-\sum_{i=2}^{m}x_i^*Q_i+\sum_{i=2}^{j}z_iQ_i=$$

$$x_1(aP_2+x^*P_2)-\sum_{i=k}^{m}x_i^*Q_i+z_kQ_k+\cdots+z_jQ_j=$$

$$x_1(a+x^*)P_2+XP_2+\frac{Y_k}{(a+c)^{2+n-k}}P_2+\cdots+$$

$$\frac{Y_{\max(m,j)}}{(a+c)^{2+n-\max(m,j)}}P_2$$

其中，X 为可求系数，Y_i 为非 0 的可求系数，$i=k,k+1,\cdots,\max(m,j)$。显然，此值能通过问题实例计算。$\mathcal{B}$ 随机选取 $Z,M\in\mathcal{Z}_p^*$，令 $r=Z-\dfrac{W}{Y_k}(a+x^*)^{(2+n-k)-1}$，计算

$$d_1=\frac{\alpha}{\alpha+H(\mathrm{ID}_1^*)}Z_{\mathrm{u}}+MK_{\mathrm{c}}+r\left(Q_1+\sum_{i=2}^{j}H(\mathrm{ID}_i)Q_i\right)=$$

$$f(a)P_2+Mb\beta P_2+Z\left(x_1(a+x^*)P_2+XP_2+\frac{Y_k}{(a+c)^{2+n-k}}P_2+\cdots+\frac{Y_{\max(m,j)}}{(a+c)^{2+n-\max(m,j)}}P_2\right)-$$

$$\frac{x_1W}{Y_k}(a+x^*)^{(2+n-k)}P_2-\frac{XW}{Y_k}(a+x^*)^{(2+n-k)-1}P_2-\frac{Y_{k+1}W}{Y_k}P_2-$$

$$\frac{Y_{k+2}W}{Y_k}(a+x^*)P_2-\cdots-\frac{Y_{\max(m,j)}W}{Y_k}(a+x^*)^{(\max(m,j)-k-1)}P_2,$$

$$d_2=r(\alpha+H(\mathrm{ID}_1{}^*))P=\left(Z-\frac{W}{Y_k}(a+x^*)(2+n-k)-1\right)(a+x^*)P_1=$$

$$Z(a+x^*)P_1-\frac{W}{Y_k}(a+x^*)^{(2+n-k)},d_3=MQ=M\beta P_2$$

对任意 $i\in[j+1,n]$，计算

$$u_i=rQ_i=\left(Z-\frac{W}{Y_k}(a+x^*)^{(2+n-k)-1}\right)$$

$$\left(x_iP_2+\frac{y_i}{(a+x^*)^{n+2-i}}P_2\right)=$$

$$Z\left(x_iP_2+\frac{y_i}{(a+x^*)^{n+2-i}}P_2\right)-\frac{x_iW}{Y_k}(a+x^*)^{(2+n-k)-1}P_2-$$

$$\frac{y_iW}{Y_k}(a+x^*)^{i-k-1}P_2$$

模拟算法 $\mathcal{B}$ 随后发送私钥 $d_{\mathrm{ID}}=(d_1,d_2,d_3,u_{j+1},u_{j+2},\cdots,u_n)$ 给攻击算法 $\mathcal{A}$。从上面设置可知，d_{ID} 为正确的私钥，d_{ID} 中的元素可通过给定的困难问题计算得到。

外包转换密钥询问。$\mathcal{B}$ 对表 T 进行核实以确定第 i 个三元组 $(i,\mathrm{ID}|_k,\mathrm{tk})$ 是否存在，当其未出现时，返还 $\mathcal{A}$ 为 $\perp$，反之将外包转换密钥 tk 返还给 $\mathcal{A}$。

密文解密询问。不失一般性，假定所有密文在此谕言机下均已完成部分解密，由于模拟算法 $\mathcal{B}$ 和攻击算法 $\mathcal{A}$ 均能得到外包转换密钥 tk，故都能执行解密外包操作。令 $\mathrm{CT}=(C_1,C_2,C_3,C_4,C_5)$ 是标识为 $\mathrm{ID}|_k$ 的第 k 层用户对应的密文，$\mathcal{B}$ 从表 T 中查询所记录的元组 $(i,\mathrm{ID},d_{\mathrm{ID}|_k},\mathrm{tk})$，如果查询的元组不存在，则返回敌手 $\mathcal{A}$ 为空。如果元组索引 i 不满足对应的挑战标识 ID^*，则 $\mathcal{B}$ 进行如下操作。

1) 解析得到第 k 层用户私钥 $\mathrm{sk}_{\mathrm{u}}=y'_{\mathrm{u}}$，计算 $\beta=C'^{\frac{1}{\mathrm{sk}_{\mathrm{u}}}}$。

2) 对于表 T_1 中的每个元组 $(R_i,\mathrm{ID}|_{ki},s_i)$，寻找匹配等式 $\beta=e(P,Q)^{\alpha s_i}$ 是否存在。

3) 如果没有匹配，则 $\mathcal{B}$ 输出为空并返回给 $\mathcal{A}$。

4) 如果至少有一个匹配，则 $\mathcal{B}$ 终止模拟过程。

5) 否则，令 $(R,\mathrm{ID}|_k,s)$ 作为唯一匹配，从表 T_2 中查询得到元组 (R,r)，如果表 T_2 不存在，则 $\mathcal{B}$ 输出为空。

6) 测试等式 $C_1=Re(P,Q)$，$C_4=\mathrm{ID}\oplus r$，$C_5=Re(P,Q)^{\alpha s}$ 是否全部成立，如果全部成立，则输出 $\mathrm{ID}|_k$，否则返回为空。

挑战阶段。待询问阶段 1 结束后，$\mathcal{A}$ 提交一个挑战标识对 $(\mathrm{ID}_0^*|_k,\mathrm{ID}_1^*|_k)\in\{0,1\}^{2\times k}$，$\mathcal{B}$ 做出如下回应。

1) $\mathcal{B}$ 选择一个随机标识 $\mathrm{ID}|_k$ 得到密文 $\mathrm{CT}=(C_1,C_2,C_3,C_5)$。

2) $\mathcal{B}$ 选择一个随机值 $C_4\in\{0,1\}^k$。

3) $\mathcal{B}$ 返回 $\mathcal{A}$ 为挑战密文 $\mathrm{CT}^*=(C_1,C_2,C_3,C_4,C_5)$。

询问阶段 2。$\mathcal{A}$ 允许继续适应性地询问私钥，$\mathcal{B}$ 的回应与询问阶段 1 不同的地方是，如果是对 $\mathrm{ID}_0^*|_k$ 或 $\mathrm{ID}_1^*|_k$ 的解密询问响应，则 $\mathcal{B}$ 用测试结果取代。

猜测阶段。最后 $\mathcal{A}$ 输出对 μ 的猜测 $\mu'\in\{0,1\}$。若 $\mu=\mu'$，$\mathcal{B}$ 输出 1，此时 $\mathcal{B}$ 认定困难问题实例中 $T=e(P_1,P_2)^{\frac{b}{a+c}}$，从而达到与真实攻击环境的不可区分性。由假设可知 $\mathcal{A}$ 有不可忽略的优势 ε 攻破该方案，则有 $\Pr\left[\mu=\mu'|T=e(P_1,P_2)^{\frac{b}{a+c}}\right]=\varepsilon+\frac{1}{2}$。若 $T\neq e(P_1,P_2)^{\frac{b}{a+c}}$，则可认定 $e(bP_1,f(a)P_2)T^W$ 是随机的，可推出 w^* 随机。此时在 $\mathcal{A}$ 视角下 w^* 与 C_1^*、C_2^* 独立无关，则有 $\Pr\left[\mu=\mu'|T\neq e(P_1,P_2)^{\frac{b}{a+c}}\right]=\frac{1}{2}$，故 $\mathcal{B}$ 成功求解 $(q,n)-$DBDHI 困难问题的概率为

$$\mathrm{Adv}_{\mathrm{RCCA}}^{(q,n)-\mathrm{DBDHI}}(\lambda)=\left|\Pr\left[\mu=\mu'|T=e(P_1,P_2)^{\frac{b}{a+c}}\right]-\Pr\left[\mu=\mu'|T\neq e(P_1,P_2)^{\frac{b}{a+c}}\right]\right|=\left|\varepsilon+\frac{1}{2}-\frac{1}{2}\right|=\varepsilon$$

与 OASM9-HIBE 密钥封装机制针对定义 1 的 RCCA 安全性证明相比，其针对定义 2（即敌手-II）的 RCCA 安全性证明做出部分调整，即 $\mathrm{Setup_c}$ 算法由 $\mathcal{A}$ 运行。

定理 2　设密码哈希函数 H、H_1 和 H_2 为随机谕言机，若 $(q,n)-$DBDHI 困难问题假设成立，则不存在一个概率多项式时间敌手能以不可忽略的优势赢得可审计安全游戏（定义 3）。

证明　若存在敌手能在 OASM9-HIBE 的可审计安全游戏中获胜，则敌手与模拟者之间的可审计性安全游戏交互与敌手-II 相比，猜测阶段被修订如下。

输出阶段。敌手输出关于挑战密文 CT^* 的两份转换密文，当下述条件同时满足时，敌手才能赢得上述交互游戏。

1) $\mathrm{Verify}(\mathrm{pk_u},\mathrm{CT}^*,C_1'^*,\mathrm{msk})\to 1$。

2) $\mathrm{Verify}(\mathrm{pk_u},\mathrm{CT}^*,C_2'^*,\mathrm{msk})\to 1$。

3) $\mathrm{Decrypt_{rcca}}(C_1'^*,\mathrm{sk_u})\to K'\neq\mathrm{Decrypt_{rcca}}(C_2'^*,\mathrm{sk_u})\to K'$。

条件 1)表明 $C_1'^*=e(P,Z_{\mathrm{u}})^{\alpha r}$，条件 2)表明 $C_2'^*=e(P,Z_{\mathrm{u}})^{\alpha r}$，根据条件 1)和条件 2)，有 $C_1'^*=C_2'^*=e(P,Z_{\mathrm{u}})^{\alpha r}$。然而，由条件 3)容易得到 $C_1'^*\neq C_2'^*$，显然与条件 1)和条件 2)相悖，故敌手在审计安全游戏中获胜的优势是可忽略的。

4　算法性能分析

本节从存储开销和计算代价两方面将 OASM9-HIBE 密钥封装机制与 SM9-HIBE[17]进行比较。符号说明如下：$|G_1|$、$|G_2|$ 和 $|G_T|$ 分别表示群 G_i（$i=1,2,T$）元素大小，T_p 表示双线性配对操作，$T_{\mathrm{sm}_i}(i=1,2)$ 表示加法群 $G_i(i=1,2)$ 中的倍乘操作，T_{e_t} 表示乘法群 G_T 中的指数操作，$\mathcal{H}_K$ 表示映射到 $\{0,1\}^l$ 的密码哈希函数（l 为封装密钥长度），计算开销比较和存储开销比较如表 2 和表 3 所示。

4.1　算法性能理论分析

本文假定 OASM9-HIBE 密钥封装机制系统的最大层数为 n，顶层 KGC 调用密钥生成算法为第 k 层用户生成密钥。从密钥生成、密钥派生（第 $k-1$ 层到第 k 层）、密文生成和终端解密算法 4 个方面对其计算开销进行比较，结果如表 2 所示。然后以存储开销为基准对方案进行比较，从系统全局参数生成、密钥生成、密钥派生、密文生成和终端解密 5 个方面对其存储开销进行比较，结果如表 3 所示。由表 2 和表 3 可得，以系统全局参数和密钥

表 2 计算开销比较

密钥封装机制	密钥生成	密钥派生	密文生成	终端解密
SM9-HIBE	$2T_{\mathrm{sm}_1}+(n+1)T_{\mathrm{sm}_2}$	$2T_{\mathrm{sm}_1}+(n+1)T_{\mathrm{sm}_2}$	$2T_{\mathrm{sm}_1}+kT_{\mathrm{sm}_2}+1E_t$	$2T_p+\mathcal{H}_K$
OASM9-HIBE	$2T_{\mathrm{sm}_1}+(n+2)T_{\mathrm{sm}_2}$	$2T_{\mathrm{sm}_1}+(n+2)T_{\mathrm{sm}_2}$	$2T_{\mathrm{sm}_1}+kT_{\mathrm{sm}_2}+2E_t$	$T_{e_t}+\mathcal{H}_K$

表 3 存储开销比较

密钥封装机制	全局参数生成	密钥生成	密钥派生	密文生成	终端解密
SM9-HIBE	$\lvert G_T\rvert+(n+1)\lvert G_2\rvert+2\lvert G_1\rvert$	$(n-k+1)\lvert G_2\rvert+\lvert G_1\rvert$	$(n-k+1)\lvert G_2\rvert+\lvert G_1\rvert$	$\lvert G_1\rvert+\lvert G_2\rvert$	$2\lvert G_T\rvert$
OASM9-HIBE	$2\lvert G_T\rvert+(n+1)\lvert G_2\rvert+2\lvert G_1\rvert$	$(n-k+2)\lvert G_2\rvert+\lvert G_1\rvert$	$(n-k+2)\lvert G_2\rvert+\lvert G_1\rvert$	$\lvert G_1\rvert+\lvert G_2\rvert+\lvert G_T\rvert$	$\lvert G_2\rvert$

生成为比较基准，相比于 SM9-HIBE，OASM9-HIBE 均分别微幅度增加了一个群元素，终端解密计算开销由原方案中的 2 次配对运算降低至一次简单的指数运算，从而有效改善解密性能。

4.2 算法性能实验数据分析

本节通过实际编程对方案进行实现，同时与 SM9-HIBE[17]中每个算法的运行时间进行比较。仿真模拟中算力受限的客户端配置是一部 12 GB 内存，Intel(R) Core(TM) i5-4210U CPU@1.70 GHz 2.40 GHz 中央处理器，64 位 Windows 8 操作系统的个人计算机，服务端则设置为一部 32 GB 内存，Intel® Core i7-10700F CPU@2.90GHz×16 中央处理器，64 位 Ubuntu 22.04.1 LTS 操作系统的物理服务器，编程语言是 Go 1.19，密码学库为 Cryptogm，测试系统安全级别为 128 bit（$\lvert G_1\rvert=256$ bit），通过 go 语言基准测试得到在乘法群 $\mathcal{G}_T$ 实行一次双线性配对操作并将群元素输出耗时约为 39.45 ms，实行一次指数操作耗时约为 10.72 ms，在加法群 $G_i(i=1,2)$ 实行一次标量乘运算耗时分别约为 3.48 ms 和 7.07 ms。以系统最大层数 10 为例，对方案每一步算法进行性能评估得到数据如表 4~表 7 所示。由表 4~表 7 可得，以 100 份密文为例，OASM9-HIBE 密钥封装机制在额外耗时约 11.33 ms 的批量审计成本外，最终解密时间约为 12.58 s，而原 SM9-HIBE 密钥封装机制的终端解密时间约为 79.27 s，从而下降了约 84.13%。由图 2 可得，OASM9-HIBE 通过将大量计算高昂的配对操作安全交付给云端处理后，其终端解密时间增长速度随着密文数目增加明显低于文献[17]，从而适配于需即时回应高并发解密请求且算力受限的计算设备（如微控制器、小型芯片等）中，故本文方案是有效可行的，仿真数据与理论对比一致。

表 4 终端一次解密时间比较

密钥封装机制	终端解密/ms
SM9-HIBE	78.63
OASM9-HIBE	11.07

表 5 终端 100 次解密时间比较

密钥封装机制	终端解密/ms
SM9-HIBE	7 927
OASM9-HIBE	1 258

表 6 其他步骤一次仿真时间比较

密钥封装机制	密钥生成/ms	密钥派生/ms	加密/ms	解密外包/ms	批量审计/ms
SM9-HIBE	82.90	82.13	84.79	—	—
OASM9-HIBE	88.11	87.90	95.13	14.25	10.68

表 7 其他步骤 100 次仿真时间比较

密钥封装机制	密钥生成/ms	密钥派生/ms	加密/ms	解密外包/ms	批量审计/ms
SM9-HIBE	82.90	82.13	8 572	—	—
OASM9-HIBE	88.11	87.90	9 737	1 501	11.33

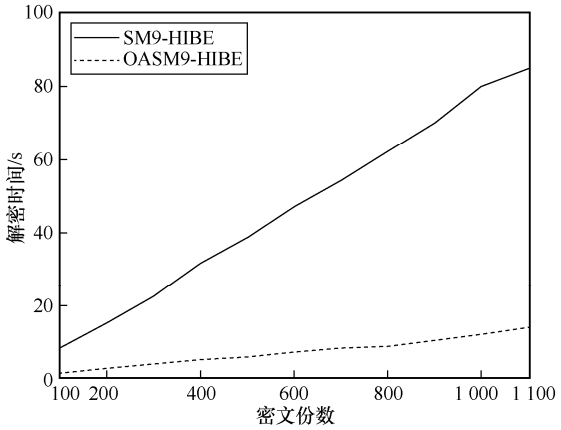


图 2 SM9-HIBE 与 OASM9-HIBE 终端解密时间对比

5 结束语

本文针对 SM9-HIBE 解密过程中需要 2 次耗时

较大的配对运算，不适于算力受限但又需对大规模数据进行高频解密处理的计算设备中的缺陷，设计出了一种具备解密外包和多密文批量审计特性的分层密钥封装机制——OASM9-HIBE，在不改变 SM9-HIBE 层级间用户私钥向下代理生成功能条件下，第 k 层用户完成最终解密只需一次简单的指数运算，并利用 Fujisaki-Okamoto 转换技术，在随机谕言模型下将原来具备 CPA 安全性的 OASM9-HIBE 方案改进为具备 RCCA 安全性。最后，通过仿真实验与 SM9-HIBE 进行比较，实验数据表明 OASM9-HIBE 方案在大规模网络且需进行高并发解密操作环境中是实用的，从而有效拓展了国产商用密码 SM9 的应用范围。

附录 1　正确性证明

云端通过自身私钥 $\mathrm{sk_c}=k_{\mathrm{c}}$ 和外包转换密钥

$$\mathrm{tk}=\{d_1,d_2,d_3\}=\left\{\frac{\alpha}{\alpha+H(\mathrm{ID}_1)}Z_{\mathrm{u}}+\gamma K_{\mathrm{c}}+r\left(Q_1+\sum_{i=2}^{k}H(\mathrm{ID}_k)Q_k\right),r(\alpha+H(\mathrm{ID}_1))P,\ \gamma Q\right\}$$

对密文 CT 中的

$$\{C_1,C_2\}=\left\{s(H(\mathrm{ID}_1)P+P_{\mathrm{pub}}),s\left(Q_1+\sum_{i=2}^{k}H(\mathrm{ID}_i)Q_i\right)\right\}$$

进行部分解密生成转换密文

$$C'=\frac{e(C_1,d_1)}{e(C_1,\mathrm{sk_c}d_3)e(d_2,C_2)}=\frac{e(C_1,d_1)}{e(C_1,k_{\mathrm{c}}\gamma Q)e(d_2,C_2)}=$$

$$\frac{e\left(s\left(H(\mathrm{ID}_1)P+P_{\mathrm{pub}}\right),\alpha\left(H(\mathrm{ID}_1)+\alpha\right)^{-1}Z_{\mathrm{u}}+\gamma K_{\mathrm{c}}+r\left(Q_1+\sum_{i=2}^{k}H(\mathrm{ID}_i)Q_i\right)\right)}{e\left(s\left(H(\mathrm{ID}_1)P+P_{\mathrm{pub}}\right),k_{\mathrm{c}}\gamma Q\right)e\left(r\left(\alpha+H(\mathrm{ID}_1)\right)P,s\left(Q_1+\sum_{i=2}^{k}H(\mathrm{ID}_i)Q_i\right)\right)}=$$

$$\frac{e\left(s\left(H(\mathrm{ID}_1)P+P_{\mathrm{pub}}\right),\alpha\left(H(\mathrm{ID}_1)+\alpha\right)^{-1}Z_{\mathrm{u}}+\gamma k_{\mathrm{c}}Q+r\left(Q_1+\sum_{i=2}^{k}H(\mathrm{ID}_i)Q_i\right)\right)}{e\left(s\left(H(\mathrm{ID}_1)P+P_{\mathrm{pub}}\right),k_{\mathrm{c}}\gamma Q\right)e\left(r\left(\alpha+H(\mathrm{ID}_1)\right)P,s\left(Q_1+\sum_{i=2}^{k}H(\mathrm{ID}_i)Q_i\right)\right)}=$$

$$\frac{e\left(s\left(H(\mathrm{ID}_1)P+P_{\mathrm{pub}}\right),\alpha\left(H(\mathrm{ID}_1)+\alpha\right)^{-1}Z_{\mathrm{u}}+r\left(Q_1+\sum_{i=2}^{k}H(\mathrm{ID}_i)Q_i\right)\right)}{e\left(r\left(\alpha+H(\mathrm{ID}_1)\right)P,s\left(Q_1+\sum_{i=2}^{k}H(\mathrm{ID}_i)Q_i\right)\right)}=e(sP,\alpha Z_{\mathrm{u}})=e(P,Z_{\mathrm{u}})^{\alpha s}$$

KGC 作为可信实体（以系统主私钥 $\mathrm{msk}=\alpha$ 作为审计凭证）依据承诺值密文 $C_3=S_{\mathrm{u}}^{s}$，对同一数据管理者 m 份转换密文的正确性进行批量审计

$$E^{\alpha}=\prod_{i=1}^{m}S_{\mathrm{u}}^{\alpha s_i}=S_u^{\alpha\sum_{i=1}^{m}s_i}=e(P,Q)^{\alpha y_{\mathrm{u}}\sum_{i=1}^{m}s_i}=$$

$$\mathrm{CK}=\prod_{i=1}^{m}C_i'=e(P,Z_{\mathrm{u}})^{\alpha\sum_{i=1}^{m}s_i}$$

证毕。

参考文献：

[1] SHAMIR A. Identity-based cryptosystems and signature schemes[C]// Advances in Cryptology. Berlin: Springer, 2007: 47-53.

[2] BONEH D, FRANKLIN M. Identity-based encryption from the Weil pairing[C]//Advances in Cryptology 2001. Berlin: Springer, 2001: 213-229.

[3] HORWITZ J, LYNN B. Toward hierarchical identity-based encryption[C]//Advances in Cryptology. Berlin: Springer, 2002: 466-481.

[4] BONEH D, BOYEN X. Efficient selective-id secure identity-based

encryption without random oracles[C]//International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2004: 223-238.

[5] WATERS B. Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions[C]//Annual International Cryptology Conference. Berlin: Springer, 2009: 619-636.

[6] LEWKO A, WATERS B. Unbounded HIBE and attribute-based encryption[C]//Advances in Cryptology. Berlin: Springer, 2011: 547-567.

[7] BOYEN X, WATERS B. Anonymous hierarchical identity-based encryption (without random oracles)[C]//Lecture Notes in Computer Science. Berlin: Springer, 2006: 290-307.

[8] LANGREHR R, PAN J X. Tightly secure hierarchical identity-based encryption[C]//International Workshop on Public Key Cryptography. Berlin: Springer, 2019: 436-465.

[9] LANGREHR R, PAN J X. Hierarchical identity-based encryption with tight multi-challenge security[C]//IACR International Conference on Public-Key Cryptography. Cham: Springer, 2020: 153-183.

[10] BLAZY O, KILTZ E, PAN J X. (Hierarchical) identity-based encryption from affine message authentication[C]//Annual Cryptology Conference. Berlin: Springer, 2014: 408-425.

[11] GONG J Q, CAO Z F, TANG S H, et al. Extended dual system group and shorter unbounded hierarchical identity based encryption[J]. Designs, Codes and Cryptography, 2016, 80(3): 525-559.

[12] LANGREHR R, PAN J X. Tightly secure hierarchical identity-based encryption[J]. Journal of Cryptology, 2020, 33: 1787-1821.

[13] Cryptography Standardization Technical Committee. SM9 identity-based cryptographic algorithm: GM/T0044-2016[S]. 2016.

[14] 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.
LAI J C, HUANG X Y, HE D B, et al. Security analysis of SM9 digital signature and key encapsulation[J]. Scientia Sinica (Informationis), 2021, 51(11): 1900-1913.

[15] 赖建昌, 黄欣沂, 何德彪. 一种基于商密 SM9 的高效标识广播加密方案[J]. 计算机学报, 2021, 44(5): 897-907.
LAI J C, HUANG X Y, HE D B. An efficient identity-based broadcast encryption scheme based on SM9[J]. Chinese Journal of Computers, 2021, 44(5): 897-907.

[16] 秦宝东, 张博鑫, 白雪. 基于仲裁的 SM9 标识加密算法[J]. 计算机学报, 2022, 45(2): 412-426.
QIN B D, ZHANG B X, BAI X. Mediated SM9 identity-based encryption algorithm[J]. Chinese Journal of Computers, 2022, 45(2): 412-426.

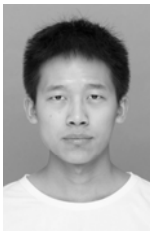
[17] 赖建昌, 黄欣沂, 何德彪, 等. 基于商用密码 SM9 的高效分层标识加密[J]. 中国科学(信息科学), 2023, 53(5): 918-930.
LAI J C, HUANG X Y, HE D B, et al. An efficient hierarchical identity-based encryption based on SM9[J]. Scientia Sinica (Informationis), 2023, 53(5): 918-930.

[18] BONEH D, BOYEN X, GOH E J. Hierarchical identity based encryption with constant size ciphertext[C]//Lecture Notes in Computer Science. Berlin: Springer, 2005: 440-456.

[19] GREEN M, HOHENBERGER S, WATERS B. Outsourcing the decryption of ABE ciphertexts[C]//Proceedings of the 20th USENIX Security Symposium. Berkeley: USENIX Association, 2011: 523-538.

[20] FUJISAKI E, OKAMOTO T. Secure integration of asymmetric and symmetric encryption schemes[J]. Journal of Cryptology, 2013, 26(1): 80-101.

[作者简介]



刘宽（1995− ），男，河南周口人，福建师范大学博士生，主要研究方向为应用密码学、数据安全等。

宁建廷（1988− ），男，浙江衢州人，博士，福建师范大学教授、博士生导师，主要研究方向为公钥密码学、数据安全和区块链安全等。

伍玮（1981− ），女，江苏南京人，博士，福建师范大学教授，主要研究方向为密码学、信息安全等。

许胜民（1989− ），男，山东荣成人，博士，福建师范大学副教授，主要研究方向为应用密码学、区块链等。

林超（1991− ），男，福建平和人，博士，福建师范大学副教授，主要研究方向为应用密码学、区块链隐私保护等。