基于身份的可验证密钥的公钥内积函数加密算法

邓宇乔[1),2)] 宋 歌[3)] 杨 波[4)] 彭长根[5)] 唐春明[6)] 温雅敏[1),2)]

[1)](广东财经大学大数据与教育统计应用实验室 广州 510120)

[2)](广东财经大学统计与数学学院 广州 510120)

[3)](华南农业大学数学与信息学院 广州 510120)

[4)](陕西师范大学计算机科学学院 西安 710062)

[5)](贵州省公共大数据重点实验室(贵州大学) 贵阳 550025)

[6)](广州大学数学与信息科学学院 广州 510006)

摘 要 函数加密(Functional Encryption,FE)是一种多功能的加密原语,最早由 Boneh 等人正式提出.自从 FE 出现以来,许多研究者考虑如何实现通用的 FE 的构造.但是,这些工作使用了较为复杂的理论工具:例如,不可区分性的混淆和多线性映射等,实用性存疑.因此,构造特殊的、高效的 FE 以满足特定应用场合的需要成为了许多学者探索的热点.本文对近来较为热门的一种 FE:内积函数加密方案(Inner Product Functional Encryption,IPFE)进行研究,以解决目前的 IPFE 无法指定接收者身份,以及无法认证密钥颁发者身份的问题.内积函数加密(Inner Product Functional Encryption,IPFE)作为一种新颖的加密原语,可以分为公钥 IPFE(PK-IPFE)和私钥 IPFE(SK-IPFE).目前提出的 PK-IPFE 有两点可改进之处:一方面,不能为密文指定接收者的身份,这将可能在一些应用场景下外泄密文的敏感信息;另一方面,它不能抵抗以下密钥的修改攻击:持有向量密钥的恶意敌手可以将此向量进行修改.因为现存的 PK-IPFE 方案无法提供密钥的验证功能,因此,该攻击也将可能导致安全性的危害.提出一种标准模型下的基于身份的可验证密钥的 PK-IPFE 方案 ID-PK-IPFE,形式化地给出针对该方案的三种攻击模型 s-CPA、s-IMA 和 s-VMA,其中 s-CPA 模型展示选择性的密文不可区分性;s-IMA 模型展示密钥中身份的不可修改性;s-VMA 模型展示密钥中向量的不可修改性.提出了两个新的困难性假设:CBDH 和 DBDH-v,其中 CBDH 假设的安全性可归约到 CDH 假设上,DBDH-v 的安全性可归约到 DBDH 假设上.把 ID-PK-IPFE 的 s-CPA、s-IMA 和 s-VMA 安全性归约到 CBDH 和 DBDH-v 这两个假设中.把 ID-PK-IPFE 的理论效率与 Abdalla 和 Agrawal 等人提出的两个 PK-IPFE 方案进行了对比,得出了 ID-PK-IPFE 的效率稍低,但在权限控制和抵御密钥修改攻击方面存在优势的结论.为进一步检验方案的实用性,使用 JPBC 库在一台 CPU 为 i7-6700 3.40 GHz,内存为 8.00 GB,操作系统为 Windows 7 64-bit 的个人 PC 机上实现了本文的方案.在 Setup 算法中添加了预处理阶段:在该阶段,程序将预先计算多个消息的值,并将预先计算的结果存放到 hash 表中,待解密消息时可供查询.分别进行了两组实验,在第一组实验中,消息的范围为(0,1000),而在第二组实验中,消息的范围为(0,10000).在(0,10 000)范围内时,大多数数据统计应用程序的需求都可以满足.设定实验中向量的长度均从 10 增加到 15,实验证明,ID-PK-IPFE 方案是实用的.

关键词 公钥内积函数加密;基于身份的加密;标准模型;可验证的密钥

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2021.00209

Identity-Based Inner Product Functional Encryption with Verified Secret Key

DENG Yu-Qiao[1),2)] SONG Ge[3)] YANG Bo[4)] PENG Chang-Gen[5)] TANG Chun-Ming[6)] WEN Ya-Min[1),2)]

[1)](*Guangdong University of Finance and Economics, Big Data and Education Statistics Application Laboratory, Guangzhou* 510120)

[2)](*Guangdong University of Finance and Economics, School of Mathematics and Statistics, Guangzhou* 510120)

[3)](*College of Mathematics and Informatics, South China Agricultural University, Guangzhou* 510120)

[4)](*School of Computer Science, Shanxi Normal University, Xi'an* 710062)

[5)](*Guizhou Provincial Key Laboratory of Public Big Data (Guizhou University), Guiyang* 550025)

[6)](*School of Mathematics and Computer Science, Guangzhou University, Guangzhou* 510006)

Abstract Functional encryption (FE) is a multifunctional encryption primitive that was first

收稿日期:2019-10-18;在线发布日期:2020-07-10.本课题得到国家重点研发计划(2017YFB0802000)、国家自然科学基金(61772147,61300204,62002122)、广东省自然科学基金(2017A030313373,2019A1515011797)、国家密码发展基金(MMJJ20170117)、广州市教育局协同创新重大项目(1201610005)、广东省普通高校特色创新类项目(2019KTSCX014)资助.**邓宇乔**,博士,副教授,主要研究方向为密码学、云计算.E-mail:425478541@qq.com.**宋 歌**(通信作者),博士,讲师,主要研究方向为密码学.**杨 波**,博士,教授,主要研究领域为信息安全和密码学.**彭长根**,博士,教授,主要研究领域为密码学.**唐春明**,博士,教授,主要研究领域为信息安全与密码学.**温雅敏**,博士,副教授,主要研究方向为密码学、云计算.

formally proposed by Boneh et al. Since the advent of FE, many researchers have considered how to implement a general FE structure. However, these works use more complicated theoretical tools. For example, indistinguishable confusion and multilinear map, etc., and their practicality is doubtful. Therefore, the construction of special and efficient FE to meet the needs of specific applications has become a hot topic explored by many scholars. In this paper, a popular FE: Inner Product Functional Encryption (IPFE) is studied to solve the problem that the current IPFE cannot specify the identity of the receiver and cannot authenticate the identity of the sender. IPFE can be classified into public-key IPFE(PK-IPFE) and secret-key IPFE(SK-IPFE). However, the proposed PK-IPFEs demonstrate two drawbacks. On the one hand, recipient cannot be designed for a ciphertext; this leads to leaking sensitive information in several applications. On the other hand, it cannot resist the following secret key-modified attack: a malicious adversary that holds a private key of a certain vector can modify this vector to be another. This attack threatens the security of PK-IPFE because it makes the secret key authentication infeasible. An identity-based PK-IPFE scheme, called ID-PK-IPFE, supporting verifiable key under the standard model is proposed. Three attack models, namely, s-CPA, s-IMA, and s-VMA for this scheme are formally given. The s-CPA model shows selective ciphertext indistinguishability; the s-IMA model shows the unmodifiability of identities in secret keys; and the s-VMA model shows the unmodifiability of vectors in secret keys. Two new difficult assumptions are proposed: CBDH and DBDH-v. The security of the CBDH assumption can be reduced to the CDH assumption, and the security of DBDH-v can be reduced to the DBDH assumption. We reduce the security of ID-PK-IPFE's s-CPA, s-IMA and s-VMA to the CBDH and DBDH-v assumptions. We compare the theoretical efficiency of ID-PK-IPFE with the two PK-IPFE schemes proposed by Abdalla and Agrawal. It is concluded that the efficiency of ID-PK-IPFE is lower, however, ID-PK-IPFE has advantages in terms of permission control and resistance to key modification attacks. To further verify the practicality of our scheme, the JPBC library is used to implement the solution in this article on a personal PC with a CPU of i7-6700 3.40 GHz, a memory of 8.00 GB, and an operating system of Windows 7 64-bit. A pre-processing stage is added to the Setup algorithm: at this stage, the program will pre-calculate the values of multiple messages and store the pre-calculated results in a hash table, which can be queried when the messages are decrypted. Two groups of experiments are conducted separately. In the first group of experiments, the range of messages is (0,1000), while in the second group of experiments, the range of messages is (0,10000). In the range of (0,10000), the needs of most data statistics applications can be met. In the experiment, the length of the vector is increased from 10 to 15, and the experiment proves that the ID-PK-IPFE scheme is practical.

Keywords public key inner product functional encryption; identity-based encryption; standard model; verified secret key

1 引 言

函数加密(Functional Encryption,FE)是一种多功能的加密原语,最早由 Boneh 等人正式提出[1].自从 FE 出现以来,许多研究者考虑如何实现通用的 FE 的构造[2-5].但是,这些工作使用了较为复杂的理论工具.例如,不可区分性的混淆(Indistinguishability Obfuscation,IO)和多线性映射(Multilineal Map)等,实用性存疑.因此,构造特殊的、高效的 FE 以满足特定应用场合的需要成为了许多学者探索的热点.本文对近来较为热门的一种 FE:内积函数加密方案(Inner Product Functional Encryption,IPFE)进行研究,以解决目前的 IPFE 无法指定接收者身

份，以及无法认证密钥颁发者身份的问题.

1.1 相关工作

在 PKC2015 中，Abdalla 等人[6]提出了一种新的加密原语，即 IPFE. 此外，Abdalla 等人构造了具体的 PK-IPFE 方案（即 ABCP 方案），并证明了其基于不可区分性的安全性. 但是，ABCP 方案仅被证明具有选择性的安全性，因此不能抵抗自适应敌手. Abdalla 等人[7]进一步提出了一种可抗自适应敌手的 PK-IPFE. 最近，Agrawal 等人[8]在 CRYPTO 16 中提出了一种完全安全的 PK-IPFE 方案. Benhamouda 等人[9]提出了一种使用投影哈希函数构造的、CCA 安全的 PK-IPFE.

在另一个研究方向 SK-IPFE 上，学者们也做了很多的工作. Bishop 等人[10]提出了具有函数隐藏功能的 IPFE. 与 PK-IPFE 不同，SK-IPFE 的方案要求在加密期间引入一个秘密向量. 该方案被证明是完全安全的，但是，文献[10]指出，这种完全安全性存在缺陷，因为它为攻击者引入了特殊的限制. Datta 等人[11]提出了一种新的 SK-IPFE，以提高安全性并消除上述限制. 他们采用了文献[12]中的技术，并更新了对攻击者的限制. 最近，Zhao 等人[13]提出了具有模拟安全性的 SK-IPFE.

1.2 本文的贡献

本文主要有三个贡献，分述如下：

(1) 本文提出了 ID-PK-IPFE. 加密者可以指定密文接收者的身份，并且解密者的密钥可以由任何人公开地验证（即密钥不可改）.

(2) 本文提出了 ID-PK-IPFE 的安全模型，包括选择性 CPA 安全模型和可抗修改的安全模型.

(3) 本文基于标准模型，严格地证明了 ID-PK-IPFE 的安全性.

2 背景知识

在本节中，我们首先介绍将使用到的技术基础，随后，约定一些方便文章描述的记号.

2.1 双线性映射

假定 $\mathbb{G}$ 和 $\mathbb{G}_T$ 是两个阶为 p 的乘法素阶群，e 是具有以下属性的双线性映射：

(1) 双线性. 以下等式对 $\mathbb{G}$ 和 a,b 中的所有的 $u,v\in\mathbb{G}$ 和 $a,b\in\mathbb{Z}_p$ 均成立.

(2) 非退化性. $e(g,g)\neq 1$.

2.2 困难性假设

本节将提出两个新的假设，分别是计算性双线性 Diffie-Hellman 假设（Computational Bilinear Diffie-Hellman Assumption，CBDH）和判定性双线性 Diffie-Hellman 假设的变体（Variant of Decisional Bilinear Diffie-Hellman Assumption，DBDH-v）. 详细描述请参见附录 1.

2.3 术　语

本文给定以下的术语. 令 $SK_{\boldsymbol{y}}^{ID}$ 表示身份为 ID 的接收者持有的与向量 $\boldsymbol{y}$ 相关联的密钥. 令 $CT_{\boldsymbol{x}}^{ID}$ 表示接收者身份为 ID，与向量 $\boldsymbol{x}$ 关联的密文. 令 $[n]$ 表示 $\{1,2,\cdots,n\}$ 的集合.

3 ID-PK-IPFE 的构造动机

本文将提出的 ID-PK-IPFE 主要具有以下的两个新功能：可指定密文接收者的身份以及密钥的可验证性. 本节将首先展示 PK-IPFE 的一个应用场景，随后，说明这两个功能在该应用场景中的作用.

3.1 PK-IPFE 的一个应用场景

考虑在云环境中应用安全的 k-近邻算法（Secure kNN，SkNN）对数据进行挖掘和查询[14-16]. kNN 是数据挖掘领域的经典分类算法. kNN 查询的原理简要描述如下：数据拥有者（Data Owner，DO）拥有一个数据库 D，而 D 由 m 个点（即 m 个向量）$\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_m$ 组成. 查询用户（Queried User，QU）拥有一个需要查询的点（即向量）$\boldsymbol{q}$，它需要检索 D 中最接近点 $\boldsymbol{q}$ 的 k 个点. 本质上，如文献[14]中所述，上述 kNN 查询可以转换为两个向量的内积计算.

如果 DO 将其数据库 D 上传到云服务提供商（Cloud Service Provider，CSP），而 CSP 将利用这些数据为大量的 QU 提供查询服务. 在传统的 kNN 模型中，DO 的数据隐私无法得到保护. 因此，通常的做法是，DO 逐一加密其向量 $\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_m$，然后将其上传到 CSP 以保护数据隐私. QU 将其需要查询的向量 $\boldsymbol{q}$ 发送到 CSP. CSP 通过执行内积计算获得 kNN 结果，并将该结果发送到 QU.

综上，可以采用 PK-IPFE 来实现 SkNN：DO 使用加密算法对向量进行加密，并将其上传到 CSP；QU 将查询密钥（密钥内包含有 QU 的查询向量）发送给 CSP；PK-IPFE 的安全性将可保证 DO 的数据隐私不会外泄（在此模型中我们不考虑 QU 的查询隐私）.

3.2 以上方案中存在的问题

上述方案存在以下两个问题：

(1) DO 无法指定其密文的接收者. 这将在一些应用中带来不便. 例如，假设 DO 仅向已付费的 QU 提供查询服务，如果 DO 不能指定密文的接收者，它就不能阻止未付费的 QU 查询数据库.

(2) 恶意攻击者可能会修改查询密钥. 在下节将指出，在一般的 PK-IPFE 方案中，密钥内嵌入的向量可以被修改，这种修改会对 SkNN 模型造成威胁. 例如，如图 1 所示，假设 QU 向 CSP 发送了与向量关联的查询密钥，并且该密钥被敌手捕获，则敌手可将其修改为 $k \cdot \boldsymbol{q}$（其中 k 为敌手选定的整数），然后将此修改的查询密钥发送到 CSP. CSP 将向 QU 返回错误的 kNN 查询结果. 以下的例 1 将更加清楚地表明，CSP 返回的错误结果将会导致严重的后果.

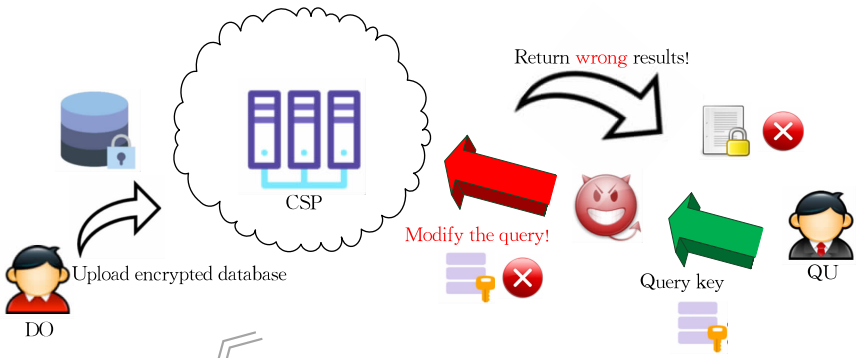


图 1 SkNN 中的密钥修改攻击

例 1. 假设 DO 上传加密点 $\{\boldsymbol{p}_1=(-1,-1), \boldsymbol{p}_2=(0,0), \boldsymbol{p}_3=(4,4), \boldsymbol{p}_4=(5,5)\}$ 到 CSP. QU 的查询点是 $\boldsymbol{q}=(1,1)$. QU 需要得到 2NN 的结果，即查询在 DO 的数据库中，与点(1,1)的欧几里得距离最近的两个点. 实际上，距离点 $\boldsymbol{q}$ 最近的 2 个点应为 $\boldsymbol{p}_1, \boldsymbol{p}_2$. 然而，如果敌手截获了 QU 的查询密钥，并将密钥中的点 $\boldsymbol{q}=(1,1)$ 修改为 $\boldsymbol{q}'=3*\boldsymbol{q}=(3,3)$，则 CSP 返回的 2 个最近的点将变成 $\boldsymbol{p}_3, \boldsymbol{p}_4$，而不再是 $\boldsymbol{p}_1, \boldsymbol{p}_2$. 由此可见，一旦敌手可以修改密钥中嵌入的向量，将有可能导致严重的后果.

3.3 密钥修改攻击

本文先简要回顾 ABCP 方案的 KeyDer 算法(有关整个方案的详细说明，请参见文献[6]的第 3 节). 该算法使用 MSK，即向量 $\boldsymbol{s}=(s_1,\cdots,s_n)$，将向量 $\boldsymbol{y}=(y_1,\cdots,y_n)$ 编码进密钥中：$SK_{\boldsymbol{y}}=(SK_1, SK_2)=(\boldsymbol{y}, \langle \boldsymbol{y}, \boldsymbol{s}\rangle)$.

现在介绍如何修改密钥. 假设 Alice(即敌手)获得了 ℓ 个密钥

$$(SK_{\boldsymbol{y}_i})_{i\in\{1,\cdots,\ell\}}=(SK_{i,1}, SK_{i,2})_{i\in\{1,\cdots,\ell\}}=(\boldsymbol{y}_i, \langle \boldsymbol{y}_i, \boldsymbol{s}\rangle)_{i\in\{1,\cdots,\ell\}},$$

则她可以通过以下计算生成一个新的、关于向量 $\boldsymbol{y}'=k_1\boldsymbol{y}_1+\cdots+k_\ell\boldsymbol{y}_\ell$ 的密钥，其中 $\{k_1,\cdots,k_n\}\in Z^n$ 为 Alice 自行选择的整数：

$$SK_{\boldsymbol{y}'}=(SK_1', SK_2')=\left(\sum_{i=1}^{\ell} k_i\cdot SK_{i,1}, \sum_{i=1}^{\ell} k_i\cdot SK_{i,2}\right)=(\boldsymbol{y}', \langle \boldsymbol{y}', \boldsymbol{s}\rangle).$$

在 SkNN 场景中，假设 QU 拥有关于向量的密钥，并期望查询向量 $\boldsymbol{y}$ 与保存在 CSP 上的秘密向量 $\boldsymbol{x}$ 的内积，但是，QU 向 CSP 传输密钥的过程被敌手所截获. 敌手将 QU 的密钥改为与向量 $\boldsymbol{y}'$ 相关($\boldsymbol{y}'=\boldsymbol{y}$)，随后发送到 CSP，则将导致如下后果：CSP 将向 QU 返回解密得到内积 $\langle \boldsymbol{y}'\cdot\boldsymbol{x}\rangle$(而 QU 要查询的内积实际为 $\langle \boldsymbol{y}\cdot\boldsymbol{x}\rangle$).

此种密钥中向量的修改技术可以被类似地应用于 ALS 方案[7]中，由于篇幅所限，我们省略了对其的描述.

综上所述，一般的 PK-IPFE 方案中，密钥里嵌入的向量可被敌手修改，这将给一些应用带来安全威胁.

4 ID-PK-IPFE 的定义和安全性模型

本节介绍 ID-PK-IPFE 的定义和安全性模型.

4.1 ID-PK-IPFE 的定义

定义 1(ID-PK-IPFE 的定义). ID-PK-IPFE 方案可以形式化地表示为 ID-PK-IPFE=(Setup, Encrypt, KeyGen, Verify, Decrypt)，该方案各算法的描述如下：

Setup$(1^\lambda, n)$：该算法的输入为安全参数 λ 和系统中向量的长度 n. 该算法的输出是公钥 PK 和 MSK.

Encrypt($PK, ID, \boldsymbol{x}$)：该算法的输入为公钥 PK，接收者的身份 ID 和待加密向量 $\boldsymbol{x}$. 该算法的输出为密文 $CT_{\boldsymbol{x}}^{ID}$.

KeyGen($MSK, ID, \boldsymbol{y}$)：该算法的输入为主密钥 MSK，密钥持有者的身份 ID 和向量 $\boldsymbol{y}$. 该算法输出密钥 $SK_{\boldsymbol{y}}^{ID}$.

Verify(PK, SK)：该算法的输入为公钥 PK 和密钥 SK. 算法输出 1 表示密钥验证通过，即密钥未被改动；否则输出 0 表示验证不通过，即密钥已被改动.

Decrypt($PK, CT_{\boldsymbol{x}}^{ID}, SK_{\boldsymbol{y}}^{ID'}$)：解密算法将 PK，$CT_{\boldsymbol{x}}^{ID}$ 和 $SK_{\boldsymbol{y}}^{ID'}$ 作为输入. 如果 $ID \neq ID'$，则解密以极大的概率失败；否则，解密算法输出内积值 $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

在 ID-PK-IPFE 方案中，一个接收者 ID 仅允许与一个向量进行绑定. 例如，如果已形成密钥 $SK_{\boldsymbol{y}}^{ID}$，则不允许形成密钥 $SK_{\boldsymbol{y}'}^{ID}$，其中 $\boldsymbol{y} \neq \boldsymbol{y}'$，因为 ID 已经和向量 $\boldsymbol{y}$ 绑定，则不再允许与另一个向量 $\boldsymbol{y}'$ 绑定. 我们将在第 4 节中说明原因. 如果接收者 u 需要同时绑定多个(例如 n)个不同的向量，从技术上讲，我们可以为接收者 u 创建 n 个不同的接收者 ID(例如，$ID_1, ID_2, \cdots, ID_n$)，并为 ID_i 绑定一个唯一的向量 $\boldsymbol{y}_i$ 即可.

本文以下提出关于 ID-PK-IPFE 的三个安全性模型：包括选择性明文攻击模型(selective Chosen Plaintext Attack，s-CPA)，选择性 ID 修改攻击模型(selective ID Modified Attack，s-IMA)和选择性向量修改攻击模型(selective Vector Modified Attack，s-VMA).

4.2　ID-PK-IPFE 的安全性模型

s-CPA 安全性模型展示密文的不可区分性，并且这些密文与选择性向量和选择性接收者身份相关联.

定义 2(ID-PK-IPFE 的 s-CPA 安全性模型). ID-PK-IPFE 的 s-CPA 安全性模型主要包括如下的 Init、Setup、Queries1、Queries2、Challenge 和 Guess 六个阶段，是挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 之间进行的游戏. 该模型的描述如下：

Init：$\mathcal{A}$ 选择并向 $\mathcal{C}$ 发布一个挑战身份 ID^* 和两个挑战向量 $\boldsymbol{x}_0^*, \boldsymbol{x}_1^*$ ($\boldsymbol{x}_0^* \neq \boldsymbol{x}_1^*$).

Setup：$\mathcal{C}$ 生成并发布公钥 PK.

Queries 1：$\mathcal{A}$ 可以向 $\mathcal{C}$ 提出任何密钥查询 $SK_{\boldsymbol{y}}^{ID}$，但有以下限制：

(1) 对同一个 ID 不能重复查询.

(2) 如果 $ID = ID^*$，即查询的身份 ID 等于挑战的身份 ID^*，则需满足：$\langle \boldsymbol{x}_0^* - \boldsymbol{x}_1^*, \boldsymbol{y} \rangle = 0$.

Challenge：$\mathcal{C}$ 掷硬币 $\beta \in \{0,1\}$，生成挑战密文 $CT_{\boldsymbol{x}_\beta^*}^{ID^*}$，然后将其发送给 $\mathcal{A}$.

Queries 2：与“Queries1”阶段操作相同.

Guess：$\mathcal{A}$ 输出对 β 的猜测 β'.

以上的限制 1 确保一个身份 ID 仅与一个向量相关联；而若无限制 2，则 $\mathcal{A}$ 可能知道密钥 $SK_{\boldsymbol{y}}^{ID^*}$ 满足以下式子：$\langle \boldsymbol{x}_0^*, \boldsymbol{y} \rangle \neq \langle \boldsymbol{x}_1^*, \boldsymbol{y} \rangle$.

因此，$\mathcal{A}$ 可以使用密钥 $SK_{\boldsymbol{y}}^{ID^*}$ 来解密挑战密文 $CT_{\boldsymbol{x}_\beta^*}^{ID^*}$，从而得到 $\langle \boldsymbol{x}_\beta^*, \boldsymbol{y} \rangle$. 若 $\langle \boldsymbol{x}_\beta^*, \boldsymbol{y} \rangle = \langle \boldsymbol{x}_0^*, \boldsymbol{y} \rangle$，$\mathcal{A}$ 输出 0，否则，$\mathcal{A}$ 输出 1. 因此，若无限制 2，$\mathcal{A}$ 可以频繁地赢得该游戏.

当 $ID \neq ID^*$ 时，$\mathcal{A}$ 可以查询密钥 $SK_{\boldsymbol{y}}^{ID}$，其中 $\boldsymbol{y}$ 可以是任意向量. 这是因为，$\mathcal{A}$ 无法使用此密钥来解密挑战密文 $CT_{\boldsymbol{x}_\beta^*}^{ID^*}$.

定义 $\mathcal{A}$ 赢得 s-CPA 游戏的优势为

$$Adv_{\text{s-CPA}} = \left| Pr[\beta' = \beta] - \frac{1}{2} \right|.$$

定义 3(ID-PK-IPFE 的 s-CPA 的安全性定义). 如果对于任何的 PPT 算法 $\mathcal{A}$ 而言，优势 $Adv_{\text{s-CPA}}$ 都是可以忽略的，则 ID-PK-IPF 是 s-CPA 安全的.

以下提出的两种安全性模型，是用来确保密钥的不可修改性. 具体而言，如果密钥 $SK_{\boldsymbol{y}}^{ID}$ 被生成，则密钥中的身份 ID 和向量 $\boldsymbol{y}$ 都不能被任何人修改.

如果敌手试图修改密钥 $SK_{\boldsymbol{y}}^{ID}$，则他可以从以下两种方式中进行选择：首先，他可以修改密钥中的身份 ID，例如，将 ID 修改为 ID'；其次，他可以修改密钥中的向量，例如，将 $\boldsymbol{y}$ 修改成 $\boldsymbol{y}'$. 以下定义的安全模型 s-IMA 用以捕获敌手修改密钥中的身份 ID 的行为，而 s-VMA 模型用以捕获敌手修改密钥中的向量的行为.

定义 4(ID-PK-IPFE 的 s-IMA 安全性模型). s-IMA 模型是挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 之间进行的游戏，它包括 Init、Setup、Queries 和 Forge 四个阶段.

Init：$\mathcal{A}$ 选择并发布挑战的身份 ID^*.

Setup：$\mathcal{C}$ 生成并发布公钥 PK，然后将其发送给 $\mathcal{A}$.

Queries：$\mathcal{A}$ 可以进行 Q 个密钥查询 $SK_{\boldsymbol{y}_i}^{ID_i}$：($i \in [Q]$)，但限制为 $ID_i \neq ID_j$ 和 $ID_i \neq ID^*$，其中 $i, j \in [Q]$.

Forge：$\mathcal{A}$ 选择挑战向量 $\boldsymbol{y}^*$，并生成密钥 $SK_{\boldsymbol{y}^*}^{ID^*}$，然后将其提交给 $\mathcal{C}$.

定义$\mathcal{A}$赢得 s-IMA 游戏的优势为

$$Adv_{\text{s-IMA}}=Pr[\mathcal{A}(PK,\forall SK_{y_i}^{ID_i}:ID_i\neq ID^*)=SK_{y^*}^{ID^*}:(\forall y^*)].$$

定义 5(ID-PK-IPFE 的 s-IMA 安全性定义). 如果对于任意的 PPT 算法$\mathcal{A}$而言,优势$Adv_{\text{s-IMA}}$均可以忽略,则 ID-PK-IPFE 对 s-IMA 是安全的.

s-IMA 模型是一种选择性的安全模型,即敌手$\mathcal{A}$需要在收到公钥之前公开挑战身份ID^*.$\mathcal{A}$可以查询Q个秘密密钥$\{SK_{y_i}^{ID_i}\}_{i\in[Q]}$,唯一的限制是$ID_i\neq ID^*$.$\mathcal{A}$如果可以根据其已知的密钥集合,推导出新的密钥$SK_{y^*}^{ID^*}$,则可以赢得游戏.

定义 6(ID-PK-IPFE 的 s-VMA 安全性模型). ID-PK-IPFE 的 s-VMA 安全性模型包括 Init、Setup、Queries 和 Forge 四个阶段. 该模型是在挑战者$\mathcal{C}$和敌手$\mathcal{A}$之间进行的游戏. s-VMA 的详细说明如下:

Init:$\mathcal{A}$选择挑战身份ID^*和挑战向量$\boldsymbol{y}^*$,并将其发送给$\mathcal{C}$.$\mathcal{A}$的目标是生成新的密钥$SK_{y'^*}^{ID^*}$,其中$\boldsymbol{y}'^*\neq\boldsymbol{y}^*$.

Setup:$\mathcal{C}$生成公钥PK,然后将其发送给$\mathcal{A}$.

Queries 1:$\mathcal{A}$可以向$\mathcal{C}$询问Q个密钥$SK_{y_i}^{ID_i}:(i\in[Q])$,其唯一的限制是$ID_i\neq ID_j$,其中$i,j\in[Q]$,并且挑战身份$ID^*$必须与挑战向量$\boldsymbol{y}^*$绑定.

Forge:$\mathcal{A}$伪造密钥$SK_{y'^*}^{ID^*}$,其中$\boldsymbol{y}'^*\neq\boldsymbol{y}^*$.

定义$\mathcal{A}$赢得 s-VMA 游戏的优势如下:

$$Adv_{\text{s-VMA}}=Pr[\mathcal{A}(PK,\forall SK_{\boldsymbol{y}}^{ID}:(ID\neq ID^*),SK_{y^*}^{ID^*})=SK_{y'^*}^{ID^*}:\boldsymbol{y}'^*\neq\boldsymbol{y}^*].$$

定义 7(ID-PK-IPFE 的 s-VMA 的安全性定义). 如果对于任意的 PPT 算法而言,优势$Adv_{\text{s-VMA}}$都可以忽略不计,则 ID-PK-IPFE 是 s-VMA 安全的.

可以从 s-VMA 模型中观察到以下特点. 在 Init 阶段,敌手应首先声明挑战身份ID^*和挑战向量$\boldsymbol{y}^*$. 其次,在询问过程中,当敌手询问关于挑战身份ID^*的密钥时,该密钥必须与挑战向量$\boldsymbol{y}^*$绑定. 再次,$\mathcal{A}$的目标是将密钥$SK_{y^*}^{ID^*}$修改为$SK_{y'^*}^{ID^*}$,其中$\boldsymbol{y}'^*\neq\boldsymbol{y}^*$. 因此,如果 ID-PK-IPFE 方案在 s-VMA 模型中是安全的,则敌手无法选择性地修改密钥中的向量.

5 ID-PK-IPFE 的构造

5.1 ID-PK-IPFE 的构造

Setup$(1^\lambda,n)$:该算法以安全参数λ,向量的长度n为输入. 选择素数阶为p的乘法群G和G_T,双线性映射$e:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$,以及防碰撞的哈希函数$\mathcal{H}$. 选择$u_1,u_2,v_1,v_2\in G$和整数$s,s_1,\cdots,s_n\in Z_p$,计算$h=g^s$和$\{h_i=g^{s_i}\}_{i\in[n]}$. 该算法公布以下公钥:

$$PK=(\mathbb{G},\mathbb{G}_T,g,p,e,u_1,u_2,v_1,v_2,h,\{h_i\}_{i\in[n]},\mathcal{H}).$$

主密钥为$MSK=(s,\{s_i\}_{i\in[n]})$.

Encrypt$(PK,ID,\boldsymbol{x})$:该算法将PK,接收者的身份 ID 和向量$\boldsymbol{x}$作为输入. 选择$r\in Z_p$,并生成以下密文:

$$C_{x,i}=e(g,g)^{x_i}e(u_1^{ID}u_2,h_i)^r:(i\in[n]),$$

$$C_r=g^r,$$

$$C_v=(v_1^{\mathcal{H}(ID)}v_2)^r,$$

$$C_h=e(u_1^{ID}u_2,h)^r.$$

该算法生成的密文为

$$CT=\{ID,C_r,C_v,C_h,(C_{x,1},\cdots,C_{x,n})\}.$$

KeyGen$(MSK,ID,\boldsymbol{y})$:该算法将身份ID,$MSK=(s,s_1,\cdots,s_n)$和向量$\boldsymbol{y}=\{y_1,y_2,\cdots,y_n\}$作为输入. 选择$t\in Z_p$并计算:

$$K_h=(u_1^{ID}u_2)^{\sum_{j=1}^{n}s_jy_j+s}(v_1^{\mathcal{H}(ID)}v_2)^{-t},$$

$$K_t=g^t.$$

密钥为$SK=\{\boldsymbol{y},K_h,K_t\}$.

Verify(PK,SK):该算法以$PK=(\mathbb{G},\mathbb{G}_T,g,p,e,u_1,u_2,v_1,v_2,h,\{h_i\}_{i\in[n]},\mathcal{H})$和$SK=\{\boldsymbol{y},K_h,K_t\}$作为输入. 如果下式成立则输出 1:

$$e(g,K_h)\cdot e(v_1^{\mathcal{H}(ID)}v_2,K_t)=e(u_1^{ID}u_2,h_1^{y_1}\cdots h_n^{y_n}\cdot h),$$

否则,输出 0.

上述验证算法的正确性如下所示:

$$\begin{aligned}&e(g,K_h)\cdot e(v_1^{\mathcal{H}(ID)}v_2,K_t)\\&=e(g,(u_1^{ID}u_2)^{\sum_{j=1}^{n}s_jy_j+s}(v_1^{\mathcal{H}(ID)}v_2)^{-t})\cdot e(v_1^{\mathcal{H}(ID)}v_2,g^t)\\&=e\Big(u_1^{ID}u_2,\sum_{j=1}^{n}g^{s_jy_j}\cdot g^s\Big)\cdot e(v_1^{\mathcal{H}(ID)}v_2,g^{-t})\cdot e(v_1^{\mathcal{H}(ID)}v_2,g^t)\\&=e(u_1^{ID}u_2,h_1^{y_1}\cdots h_n^{y_n}\cdot h).\end{aligned}$$

Decrypt(PK,CT,SK):该算法以PK、CT和SK作为输入. 如果包含在CT和SK中的身份不同,则算法输出$\perp$并中止.

如果包含在CT和SK中的身份相同,设该接收者身份为ID,并设与CT和SK关联的向量分别为$\boldsymbol{x}=(x_1,\cdots,x_n)$和$\boldsymbol{y}=(y_1,\cdots,y_n)$,计算:

$$\prod_{i=1}^{n}C_{x,i}^{y_i}=e(g,g)^{\langle x,y\rangle}\prod_{i=1}^{n}e(u_1^{ID}u_2,h_i)^{ry_i}$$

和

$$e(C_r,K_h)=e(g^r,(u_1^{ID}u_2)^{\sum_{j=1}^{n}s_jy_j+s})\cdot e(g^r,(v_1^{\mathcal{H}(ID)}v_2)^{-t})$$

$$=e(g^{\sum_{j=1}^{n}s_jy_j+s},u_1^{ID}u_2)^r\cdot e(g,v_1^{\mathcal{H}(ID)}v_2)^{-rt}$$

$$=\prod_{i=1}^{n}e(u_1^{ID}u_2,g^{s_i})^{y_ir}\cdot e(u_1^{ID}u_2,g^s)^r\cdot e(g,v_1^{\mathcal{H}(ID)}v_2)^{-rt}$$

$$=\prod_{i=1}^{n}e(u_1^{ID}u_2,h_i)^{y_ir}\cdot e(u_1^{ID}u_2,h)^r\cdot e(g,v_1^{\mathcal{H}(ID)}v_2)^{-rt}.$$

该算法通过以下方式得到 $e(g,g)^{\langle x,y\rangle}$：

$$\prod_{i=1}^{n}C_{x,i}^{y_i}\cdot e(C_r,K_h)^{-1}\cdot C_h\cdot e(C_v,K_t)^{-1}$$

$$=\left(e(g,g)^{\langle x,y\rangle}\prod_{i=1}^{n}e(u_1^{ID}u_2,h_i)^{ry_i}\right)\cdot$$

$$\left(\prod_{i=1}^{n}e(u_1^{ID}u_2,h_i)^{-y_ir}\cdot e(u_1^{ID}u_2,h)^{-r}\cdot e(g,v_1^{\mathcal{H}(ID)}v_2)^{rt}\right)\cdot$$

$$e(u_1^{ID}u_2,h)^r\cdot e((v_1^{\mathcal{H}(ID)}v_2)^r,g^t)^{-1}$$

$$=e(g,g)^{\langle x,y\rangle}.$$

该算法寻找 $\mathcal{M}$ 满足以下方程式：

$$e(g,g)^{\langle x,y\rangle}=(e(g,g))^{\mathcal{M}},$$

算法输出 $\mathcal{M}$ 作为恢复的明文. 最后这步解密过程涉及到求离散对数问题，但是，当把 $\mathcal{M}$ 可能的取值限制在固定的多项式大小范围内时，解密算法能以多项式的时间对密文解密，目前提出的所有的 IPFE 方案均采用这种解密方式[6-9].

本文在第 3 节中讨论了 ID-PK-IPFE 方案必须设置以下的限制：一个身份 ID 只能与一个向量绑定. 假定存在两个向量，设为 $\boldsymbol{y}_1,\boldsymbol{y}_2$ 都与一个身份 ID 相关联. 这两个密钥设为

$$K_{h,1}=(u_1^{ID}u_2)^{\sum_{j=1}^{n}s_jy_{1,j}+s}(v_1^{\mathcal{H}(ID)}v_2)^{-t},$$
$$K_{t,1}=g^t,$$

$$K_{h,2}=(u_1^{ID}u_2)^{\sum_{j=1}^{n}s_jy_{2,j}+s}(v_1^{\mathcal{H}(ID)}v_2)^{-t'},$$
$$K_{t,2}=g^{t'},$$

其中，$t,t'\in Z_p$ 是两个整数. 则此时敌手可对密钥进行修改，因为以上两个密钥可用于生成关于同一个身份 ID，且关联一个新的向量的密钥. 例如，可以通过以下计算生成关于身份 ID 和向量 $\boldsymbol{y}=2\boldsymbol{y}_1-\boldsymbol{y}_2$ 的新密钥：

$$K_t'=K_{t,1}^2\cdot(K_{t,2})^{-1}=g^{2t-t'}$$

$$K_h'=(K_{h,1})^2\cdot(K_{h,2})^{-1}$$

$$=(u_1^{ID}u_2)^{2\left(\sum_{j=1}^{n}s_jy_{1,j}+s\right)}(v_1^{\mathcal{H}(ID)}v_2)^{-2t}\cdot$$

$$(u_1^{ID}u_2)^{-\left(\sum_{j=1}^{n}s_jy_{2,j}+s\right)}(v_1^{\mathcal{H}(ID)}v_2)^{t'}$$

$$=(u_1^{ID}u_2)^{\sum_{j=1}^{n}s_j(2y_{1,j}-y_{2,j})+s}(v_1^{\mathcal{H}(ID)}v_2)^{-(2t-t')}.$$

因此，本文的 ID-PK-IPFE 方案要求，一个身份 ID 必须仅能与一个向量进行绑定，以防止敌手对密钥的修改攻击.

5.2　安全性分析

在本节中将证明 ID-PK-IPFE 的 s-CPA、s-IMA 和 s-VMA 的安全性. 具体证明过程请参见附录.

5.3　效率分析

具体的理论和实验效率分析请参见附录.

6　结　论

本文通过使用双线性映射技术，提出了 ID-PK-IPFE 方案，该方案主要解决了一般的 PK-IPFE 方案中无法指定接收者，并且密钥可修改的问题. 我们定义了 s-CPA、s-IMA 和 s-VMA 模型，并在这些模型下证明了 ID-PK-IPFE 的安全性. 最后，我们对 ID-PK-IPFE 方案进行了理论和实验的效率分析.

致　谢　我们诚挚感谢编辑老师和匿名审稿专家对稿件提出的中肯意见！

参　考　文　献

[1] Boneh D, Sahai A, Waters B. Functional encryption: Definitions and challenges//Proceedings of the Theory of Cryptography-8th Theory of Cryptography Conference. RI, USA, 2011: 253-273

[2] Garg S, Gentry C, Halevi S, et al. Candidate indistinguishability obfuscation and functional encryption for all circuits// Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. Berkeley, USA, 2013: 40-49

[3] Attrapadung N. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more//Proceedings of the EUROCRYPT 2014. Copenhagen, Denmark, 2014: 557-577

[4] Waters B. A punctured programming approach to adaptively secure functional encryption//Proceedings of the CRYPTO 2015. CA, USA, 2015: 678-697

[5] Asharov G, Segev G. Limits on the power of indistinguishability obfuscation and functional encryption// Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science. CA, USA, 2015: 191-209

[6] Abdalla M, Bourse F, Caro A D, et al. Simple functional encryption schemes for inner products//Proceedings of the Public Key Cryptography 2015. MD, USA, 2015: 733-751

[7] Abdalla M, Bourse F, Caro A D, et al. Better security for functional encryption for inner product evaluations. Cryptology ePrint Archive, 2016, http://eprint.iacr.org/2016/011

[8] Agrawal S, Libert B, Stehl D. Fully secure functional encryption for inner products, from standard assumptions//Proceedings of the CRYPTO 2016. CA, USA, 2016: 333-362

[9] Benhamouda F, Bourse F, Lipmaa H. CCA-secure inner-product functional encryption from projective hash functions//Proceedings of the 20th IACR International Conference on Practice and Theory in Public-Key Cryptography. Amsterdam, The Netherlands, 2017: 36-66

[10] Bishop A, Jain A, Kowalczyk L. Function-hiding inner product encryption//Proceedings of the ASIACRYPT 2015. Auckland, New Zealand, 2015: 470-491

[11] Datta P, Dutta R, Mukhopadhyay S. Functional encryption for inner product with full function privacy//Proceedings of the Public-Key Cryptography 2016. Taiwan, China, 2016: 164-195

[12] Okamoto T, Takashima K. Fully secure functional encryption with general relations from the decisional linear assumption//Proceedings of the CRYPTO 2010. CA, USA, 2010: 191-208

[13] Zhao Qingsong, Zeng Qingkai, Liu Ximeng, Xu Huanliang. Simulation-based security of function-hiding inner product encryption. SCIENCE CHINA Information Sciences, 2018, 61(4): 048102

[14] Wong Wai Kit, Cheung David Wai-Lok, Kao Ben, et al. Secure kNN computation on encrypted databases//Proceedings of the ACM SIGMOD International Conference on Management of Data. RI, USA, 2009: 139-152

[15] Yao Bin, Li Feifei, Xiao Xiaokui. Secure nearest neighbor revisited//Proceedings of the 29th IEEE International Conference on Data Engineering. Brisbane, Australia, 2013: 733-744

[16] Gu Chunsheng, Gu Jixing. Known-plaintext attack on secure kNN computation on encrypted databases. Security and Communication Networks, 2014, 7(12): 2432-2441

[17] Diffie W, Hellman M E. New directions in cryptography. IEEE Transactions on Information Theory, 1976, 22(5): 644-654

附 录.

1. 困难性假设

ID-PK-IPFE 方案的安全性将基于以下所提的两个新的假设:分别为 CBDH 假设和 DBDH-v 假设.这两个假设的安全性分别可归约到两个经典假设 CDH 和 DBDH 的安全性上.即若 CDH 假设是难解的,则 CBDH 假设也难解;若 DBDH 假设是难解的,则 DBDH-v 假设也难解.

以下的安全性归约均遵守同样的原理:使用反证法对结论进行证明,具体证明思路如下.若要证明假设 A 难解,则假设 B 也难解,可设定一个算法 $\mathcal{A}$ 来解决假设 A,设定一个算法 $\mathcal{B}$ 来解决假设 B.利用反证法,首先假定假设 B 可解,若可以推导出假设 A 也可解,由于我们已知假设 A 是难解的(前提条件),根据逆反定理,可知假设 B 是难解的.

以下为 CBDH 和 DBDH-v 的归约过程.

定义 1(CBDH 的定义). 令 $\mathbb{G}$ 和 $\mathbb{G}_T$ 均为阶为素数 p 的乘法群,e 是双线性映射.令 $x,y,z\in Z_p$ 为随机选择的整数,g 为 $\mathbb{G}$ 的生成元.定义可解决 CBDH 假设的敌手 $\mathcal{A}$ 的优势为

$$Adv_{\mathrm{CBDH}}=Pr|\mathcal{A}(\mathbb{G},\mathbb{G}_T,e,g,g^x,g^y,g^z,g^{yz})=g^{xy}|.$$

如果对于任何概率多项式时间(Probability Polynomial Time,PPT)算法 $\mathcal{A}$ 而言,Adv_{CBDH} 都可忽略,则认为 CBDH 假设是难解的.

以下证明,若经典的 CDH[17] 问题是难解的,则本文提出的 CBDH 问题也是难解的.

定理 1. 如果 CDH 是难解的,则 CBDH 也是难解的.

证明. 假设 PPT 算法 $\mathcal{B}$ 可以以不可忽略的概率求解 CBDH 假设,构造一个 PPT 算法 $\mathcal{A}$ 来解决 CDH 假设.为 $\mathcal{A}$ 创建一个 CDH 实例 $CDH=(\mathbb{G},p,g,g^x,g^y)$,$\mathcal{A}$ 的目标是计算 g^{xy}.$\mathcal{A}$ 选择一个阶为素数 p 的群 $\mathbb{G}_T$,一个双线性映射 $e:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ 和整数 $z\in Z_p$,生成元组 $CBDH=(\mathbb{G},\mathbb{G}_T,e,g,g^x,g^y,g^z,(g^y)^z)$,将 $CBDH$ 作为输入提交给 $\mathcal{B}$.$\mathcal{B}$ 将 $CBDH$ 的解 g^{xy} 发送给 $\mathcal{A}$,此结果也是 CDH 实例的解.因此,如果 CBDH 是可解的,则 CDH 是可解的;反过来,如果 CDH 不可解,则 CBDH 不可解.

定义 2(DBDH-v 的定义). DBDH-v 假设是 DBDH 假设[17]的变体.令 $\mathbb{G}$ 和 $\mathbb{G}_T$ 是阶为素数 p 的两个乘法群,e 为双线性映射.令 $x,y,z,k\in Z_p$ 为随机选择的元素,g 为 $\mathbb{G}$ 的生成元,$T\in\mathbb{G}_T$.定义敌手 $\mathcal{A}$ 解决 DBDH-v 假设的优势为

$$Adv_{\mathrm{DBDH}}=Pr|\mathcal{A}(\mathbb{G},\mathbb{G}_T,e,g,g^x,g^y,g^z,g^k,g^{yk},T=e(g,g)^{xyz})=1|-Pr|\mathcal{A}(\mathbb{G},\mathbb{G}_T,e,g,g^x,g^y,g^z,g^k,g^{yk},T=R)=1|,$$

其中,R 是 $\mathbb{G}_T$ 中的随机元素.

我们通过以下定理证明,假设 DBDH 难解,则 DBDH-v 亦难解.

定理 2. 如果 DBDH 难解,则 DBDH-v 也难解.

证明. 假设 PPT 算法 $\mathcal{B}$ 可以以不可忽略的概率求解 DBDH-v 假设,我们构造一个 PPT 算法 $\mathcal{A}$ 来解决 DBDH 假设.首先,给 $\mathcal{A}$ 一个 DBDH 实例 $DBDH=(\mathbb{G},\mathbb{G}_T,p,g,g^x,g^y,g^z,T)$,其目标是确定 $T=e(g,g)^{xyz}$ 或 T 是 $\mathbb{G}_T$ 中的随机元素.$\mathcal{A}$ 选择一个整数 $k\in Z_p$,并生成 DBDH-v 实例:$DBDHV=(\mathbb{G},\mathbb{G}_T,e,g,g^x,g^y,g^z,g^k,(g^y)^k)$,将该 DBDH-v 实例提交给 $\mathcal{B}$.$\mathcal{B}$ 向 $\mathcal{A}$ 发送 $\beta=1$ 表示 $T=g^{xy}$ 或 $\beta=0$ 表示 $T=R$,$\mathcal{A}$ 把 β 作为它对 DBDH 假设的输出.如果 DBDH-v 是可解的,则 DBDH 是可解的;因此,如果 DBDH 难解,则 DBDH-v 也难解.

2. 安全性分析

2.1 s-CPA 的安全性分析

本文通过以下定理证明 ID-PK-IPFE 的 s-CPA 安全性.

定理 3. 如果 DBDH-v 的假设是困难的,则 ID-PK-IPFE 是 s-CPA 安全的.

证明. 如果敌手 $\mathcal{A}$ 以不可忽略的优势 ε 攻破 ID-PK-IPFE 的 s-CPA 安全性,则可以构造挑战者 $\mathcal{C}$ 以不可忽略的

优势 $\varepsilon/2$ 解决 DBDH-v 假设，以下为 $\mathcal{A}$ 和 $\mathcal{C}$ 之间的交互过程.

Init：$\mathcal{A}$ 选择一个挑战身份 ID^* 和两个挑战向量：

$\boldsymbol{x}_0^*=(x_{0,1}^*,x_{0,2}^*,\cdots,x_{0,n}^*)$，$\boldsymbol{x}_1^*=(x_{1,1}^*,x_{1,2}^*,\cdots,x_{1,n}^*)$，

其中，$\boldsymbol{x}_0^*\neq\boldsymbol{x}_1^*$.

Setup：$\mathcal{C}$ 以 DBDH-v 假设的挑战项（$\mathbb{G},\mathbb{G}_T,e,g,p,g^x,g^y,g^z,g^k,g^{yk},T$）作为输入，其任务是判定是否有 $T=e(g,g)^{xyz}$ 成立.

$\mathcal{C}$ 选择向量的长度 n，安全参数 λ 和抗碰撞哈希函数 $\mathcal{H}$. 此外，$\mathcal{C}$ 随机选取 $p_1,p_2,l_1,l_2\in Z_p$，并设置

$$u_1=g^xg^kg^{p_1},\ u_2=(g^k)^{-ID^*}g^{p_2},$$

$$v_1=g^xg^{l_1},\ v_2=(g^x)^{-\mathcal{H}(ID^*)}g^{l_2}.$$

$\mathcal{C}$ 选择 $n+1$ 个随机数 $h',h_1',h_2',\cdots,h_n'\in Z_p$ 并设置：

$$h=g^{h'},$$

$$h_i=g^{h_i'}(g^y)^{(x_{0,i}^*-x_{1,i}^*)}:(i\in[n]).$$

$\mathcal{C}$ 实际上隐含地设置了 $as=h'$ 和 $\{s_i=h_i'+(x_{0,i}^*-x_{1,i}^*)y\}_{i\in[n]}$.

最后，$\mathcal{C}$ 将以下 PK 发送给 $\mathcal{A}$：

$$PK=(\mathbb{G},\mathbb{G}_T,g,p,e,u_1,u_2,v_1,v_2,h,h_1,\cdots,h_n).$$

Queries 1：在该阶段中，$\mathcal{A}$ 向 $\mathcal{C}$ 询问密钥. 对于查询 $SK_{\boldsymbol{y}}^{ID}$，$\mathcal{C}$ 首先执行以下有效性检查：

1. 如果 $\mathcal{A}$ 已查询过身份 ID，则检查返回 -1.

2. 如果 $ID=ID^*$ 和 $\langle\boldsymbol{x}_0^*-\boldsymbol{x}_1^*,\boldsymbol{y}\rangle\neq0$ 同时成立，则检查返回 0.

3. 如果通过以上检查未返回结果，则检查返回 1.

第一步和第二步检查可确保满足 s-CPA 定义中的限制. 其中第一步检查要求敌手不能询问一个身份 ID 两次，因为一个身份 ID 仅能与一个向量进行绑定；第二个检查本文在此进行简单解释：s-CPA 的安全性要求，如果敌手查询密钥 $SK_{\boldsymbol{y}}^{ID^*}$，则向量 $\boldsymbol{y}$ 必须满足以下等式：$\langle\boldsymbol{x}_0^*,\boldsymbol{y}\rangle=\langle\boldsymbol{x}_1^*,\boldsymbol{y}\rangle$. 因此，如果 $ID=ID^*$ 和 $\langle\boldsymbol{x}_0^*-\boldsymbol{x}_1^*,\boldsymbol{y}\rangle\neq0$ 同时成立，则该查询不满足 s-CPA 的安全性模型的限制要求.

$\mathcal{C}$ 执行以下操作以响应来自 $\mathcal{A}$ 的密钥查询 $SK_{\boldsymbol{y}}^{ID}$.

1. 如果有效性检查返回 -1 或 0，则 $\mathcal{C}$ 拒绝生成密钥，并继续监听来自 $\mathcal{A}$ 的下一个密钥查询.

2. 如果有效性检查返回 1 并且 $ID\neq ID^*$ 成立，则以下式子将以极大的概率成立：$\mathcal{H}(ID)\neq\mathcal{H}(ID^*)$，因为 $\mathcal{H}$ 具有抗碰撞性. 在这个情况下，敌手 $\mathcal{A}$ 可以查询与任何向量 $\boldsymbol{y}$ 相关的秘密密钥. $\mathcal{C}$ 随机选择 $\pi\in Z_p$，并隐含地设置 $t=\frac{ID\cdot\sum_{i\in[n]}(x_{0,i}^*-x_{1,i}^*)y_i}{\mathcal{H}(ID)-\mathcal{H}(ID^*)}y+\pi$，随后，$\mathcal{C}$ 推导如下的密钥：

$$K_t=g^t=(g^y)^{\frac{ID\cdot\sum_{i\in[n]}(x_{0,i}^*-x_{1,i}^*)y_i}{\mathcal{H}(ID)-\mathcal{H}(ID^*)}}\cdot g^{\pi},$$

$$K_h=(u_1^{ID}u_2)^{\sum_{j=1}^{n}s_jy_j+s}(v_1^{\mathcal{H}(ID)}v_2)^{-t}$$

$$=((g^x)^{ID}\cdot(g^k)^{ID-ID^*}\cdot g^{p_1ID+p_2})^{y\cdot\sum_{i\in[n]}(x_{0,i}^*-x_{1,i}^*)y_i+\sum_{i\in[n]}h_i'y_i+h'}\cdot$$

$$((g^x)^{\mathcal{H}(ID)-\mathcal{H}(ID^*)}\cdot g^{\mathcal{H}(ID)l_1+l_2})^{-\frac{ID\cdot\sum_{i\in[n]}(x_{0,i}^*-x_{1,i}^*)y_i}{\mathcal{H}(ID)-\mathcal{H}(ID^*)}y+\pi}$$

$$=(g^{xy})^{ID\cdot\sum_{i\in[n]}(x_{0,i}^*-x_{1,i}^*)y_i}(g^{xy})^{-ID\cdot\sum_{i\in[n]}(x_{0,i}^*-x_{1,i}^*)y_i}\cdot$$

$$Ag^x\cdot Bg^{yk}\cdot Cg^y\cdot Dg^k\cdot E$$

$$=Ag^x\cdot Bg^{yk}\cdot Cg^y\cdot Dg^k\cdot E,$$

其中，A,B,C,D,E 是可由 $\mathcal{C}$ 计算得出的整数，而项 g^x,g^{yk},g^y,g^k 为已知项. 挑战者 $\mathcal{C}$ 不知道项 g^{xy}，而它分别出现在 $(u_1^{ID}u_2)^{\sum_{j=1}^{n}s_jy_j+s}$ 和 $(v_1^{\mathcal{H}(ID)}v_2)^{-t}$ 这两项中，但是，如上所示，该项在计算中被抵消了.

3. 如果有效性检查返回 1，且 $ID=ID^*$ 和 $\langle\boldsymbol{x}_0^*-\boldsymbol{x}_1^*,\boldsymbol{y}\rangle=0$ 同时成立，$\mathcal{C}$ 随机选择 $\pi\in Z_p$ 并设置 $t=\pi$，并如下地推导出密钥：

$$K_t=g^t=g^{\pi},$$

$$K_h=(u_1^{ID^*}u_2)^{\sum_{j=1}^{n}s_jy_j+s}(v_1^{\mathcal{H}(ID^*)}v_2)^{-t}$$

$$=((g^x)^{ID^*}\cdot(g^k)^{ID^*-ID^*}\cdot g^{p_1ID+p_2})^{y\cdot\sum_{i\in[n]}(x_{0,i}^*-x_{1,i}^*)y_i+\sum_{i\in[n]}h_i'y_i+h'}\cdot$$

$$((g^x)^{\mathcal{H}(ID^*)-\mathcal{H}(ID^*)}\cdot g^{\mathcal{H}(ID^*)l_1+l_2})^{\pi}$$

$$=((g^x)^{ID^*}\cdot g^{p_1ID+p_2})^{\sum_{i\in[n]}h_i'y_i+h'}\cdot(g^{\mathcal{H}(ID^*)l_1+l_2})^{\pi}$$

$$=A'g^x\cdot B'.$$

在上式中，A',B' 是可以由 $\mathcal{C}$ 计算出来的整数，而 $\mathcal{C}$ 未知的项 g^{xy} 被抵消掉了，因为 $\sum_{i\in[n]}(x_{0,i}^*-x_{1,i}^*)y_i=\langle\boldsymbol{x}_0^*-\boldsymbol{x}_1^*,\boldsymbol{y}\rangle=0$ 成立.

Challenge：$\mathcal{C}$ 抛掷硬币 $\beta\in\{0,1\}$，并如下地加密向量 $\boldsymbol{x}_\beta^*=(x_{\beta,1}^*,x_{\beta,2}^*,\cdots,x_{\beta,n}^*)$ 和挑战的接收者 ID^*.

$\mathcal{C}$ 隐含地设置：

$$r=z.$$

生成：

$$C_r=g^r=g^z,$$

$$C_v=(v_1^{\mathcal{H}(ID^*)}v_2)^r=((g^x)^{\mathcal{H}(ID^*)}g^{\mathcal{H}(ID^*)l_1}(g^x)^{-\mathcal{H}(ID^*)}g^{l_2})^z$$

$$=(g^z)^{\mathcal{H}(ID^*)l_1+l_2},$$

$$C_h=e(u_1^{ID^*}u_2,h)^r=e(g^{xID^*+kID^*+p_1ID^*}g^{-kID^*+p_2},g^{h'})^z$$

$$=e(g^x,g^z)^{ID^*h'}e(g,g^z)^{(p_1ID^*+p_2)h'}.$$

对于 $i\in[n]$，$\mathcal{C}$ 设置：

$$C_{x,i}=e(g,g)^{x_{\beta,i}^*}\cdot(T)^{ID^*(x_{0,i}^*-x_{1,i}^*)}\cdot e(g^x,g^z)^{ID^*h_i'}\cdot$$

$$e(g^y,g^z)^{(p_1ID^*+p_2)(x_{0,i}^*-x_{1,i}^*)}\cdot e(g^z,g)^{p_2h_i'}\qquad(7.1)$$

1. 如果 $T=e(g,g)^{xyz}$，则上式中设置的 $C_{x,i}:(i\in[n])$ 均为合法的密文分布，因为有以下式子成立：

$$C_{x,i}=e(g,g)^{x_{\beta,i}^*}e(u_1^{ID^*}u_2,h_i)^r$$

$$=e(g,g)^{x_{\beta,i}^*}\cdot e(g^{xID^*+p_1ID^*+p_2},g^{(x_{0,i}^*-x_{1,i}^*)y+h_i'})^z$$

$$=e(g,g)^{x_{\beta,i}^*}\cdot(e(g,g)^{xyz})^{ID^*(x_{0,i}^*-x_{1,i}^*)}\cdot e(g^x,g^z)^{ID^*h_i'}\cdot$$

$$e(g^y,g^z)^{(p_1ID^*+p_2)(x_{0,i}^*-x_{1,i}^*)}\cdot e(g^z,g)^{p_2h_i'}$$

$$=e(g,g)^{x_{\beta,i}^*}\cdot(T)^{ID^*(x_{0,i}^*-x_{1,i}^*)}\cdot e(g^x,g^z)^{ID^*h_i'}\cdot$$

$$e(g^y,g^z)^{(p_1ID^*+p_2)(x_{0,i}^*-x_{1,i}^*)}\cdot e(g^z,g)^{p_2h_i'}\qquad(7.2)$$

2. 如果 $T=e(g,g)^{xyz}\cdot e(g,g)^{r'}$，其中 $r'\in Z_p^*$ 是一个随机数（表明 T 是 $\mathbb{G}_T$ 中的随机元素），则有

$$C_{x,i}=e(g,g)^{x^*_{\beta,i}}\cdot(T)^{ID^*(x^*_{0,i}-x^*_{1,i})}\cdot e(g^x,g^z)^{ID^*h'_i}\cdot e(g^y,g^z)^{(p_1ID^*+p_2)(x^*_{0,i}-x^*_{1,i})}\cdot e(g^z,g)^{p_2h'_i}$$
$$=e(g,g)^{x^*_{\beta,i}}\cdot(e(g,g)^{xyz}\cdot e(g,g)^{r'})^{ID^*(x^*_{0,i}-x^*_{1,i})}\cdot e(g^x,g^z)^{ID^*h'_i}\cdot e(g^y,g^z)^{(p_1ID^*+p_2)(x^*_{0,i}-x^*_{1,i})}\cdot e(g^z,g)^{p_2h'_i}$$
$$=e(g,g)^{x^*_{\beta,i}+r'ID^*(x^*_{0,i}-x^*_{1,i})}\cdot e(u_1^{ID^*}u_2,h_i)^r.$$

上式表示被加密的向量为

$$\boldsymbol{x}'=(x^*_{\beta,1}+r'ID^*(x^*_{0,1}-x^*_{1,1}),x^*_{\beta,2}+r'ID^*(x^*_{0,2}-x^*_{1,2}),\cdots,x^*_{\beta,n}+r'ID^*(x^*_{0,n}-x^*_{1,n})).$$

因为 r' 是随机整数，因此，向量 $\boldsymbol{x}'$ 以极大概率不等于挑战向量 $\boldsymbol{x}^*_\beta$. 因此，$\mathcal{C}$ 无法成功地模拟 s-CPA 的游戏；从而，敌手 $\mathcal{A}$ 也无法从该游戏获得任何有用信息，敌手 $\mathcal{A}$ 只能随机地猜测 $\mathcal{C}$ 抛掷硬币的结果.

Queries 2：与 Queries 1 阶段相同.

Guess：最后，$\mathcal{A}$ 输出他对 β 的猜测 β'. 如果 $\beta'=\beta$，则 $\mathcal{C}$ 输出 1 表示 $T=e(g,g)^{xyz}$，否则 $\mathcal{C}$ 输出 0 表示 $T=R$.

以下分析 $\mathcal{C}$ 成功解决 DBDH-v 假设的概率. 假设 *success* 事件表示 $\mathcal{C}$ 成功解决 DBDH-v 假设，$\gamma=0$ 表示 $T=e(g,g)^{xyz}$ 成立，$\gamma=1$ 表示 $T=R$ 成立. 则有

$$\begin{aligned}Pr[success]&=Pr[success|\gamma=0]Pr[\gamma=0]+Pr[success|\gamma=1]Pr[\gamma=1]\\&=\frac{1}{2}\cdot\left(\frac{1}{2}+\varepsilon\right)+\frac{1}{2}\cdot\frac{1}{2}\\&=\frac{1}{2}+\frac{\varepsilon}{2}.\end{aligned}$$

在上式中，如果 $T=e(g,g)^{xyz}$（此事件的发生概率为 $\frac{1}{2}$），则 $\mathcal{C}$ 完美地模拟游戏，$\mathcal{A}$ 以 $\frac{1}{2}+\varepsilon$ 的概率赢得本游戏；如果 $T=R$，则 $\mathcal{C}$ 无法模拟游戏，$\mathcal{A}$ 仅以 $\frac{1}{2}$ 的概率赢得游戏.

综上所述，$\mathcal{C}$ 解决 DBDH-v 假设的优势为

$$Adv_{\text{DBDH-v}}=Pr[success]-\frac{1}{2}=\frac{\varepsilon}{2}.$$

因为在我们的假设中 ε 是不可忽略的，因此 $Adv_{\text{DBDH-v}}$ 是不可忽略的. 因此，如果敌手 $\mathcal{A}$ 以不可忽略的优势攻破 ID-PK-IPFE 的 s-CPA 安全性，则挑战者 $\mathcal{C}$ 将以不可忽略的优势解决 DBDH-v 假设.

2.2 s-IMA 的安全性分析

定理 4. 如果 CBDH 假设是困难的，则 ID-PK-IPFE 是 s-IMA 安全的.

证明. 若敌手 $\mathcal{A}$ 可以以不可忽略的优势攻破 ID-PK-IPFE 的 s-IMA 安全性，则可构造一个挑战者 $\mathcal{C}$ 解决 CBDH 假设. 以下为 $\mathcal{A}$ 和 $\mathcal{C}$ 之间的交互过程.

Init：$\mathcal{A}$ 选择并公布挑战身份 ID^*.

Setup：$\mathcal{C}$ 以 CBDH 假设的实例 $(\mathbb{G},\mathbb{G}_T,e,g,g^x,g^y,g^z,g^{yz})$ 作为输入，其任务是输出 g^{xy}.

$\mathcal{C}$ 选择向量的长度 n，安全参数 λ 和抗碰撞哈希函数 $\mathcal{H}$. 此外，$\mathcal{C}$ 随机选择 $p_1,p_2,l_1,l_2\in Z_p$，并设置

$$u_1=g^xg^zg^{p_1},\ u_2=(g^z)^{-ID^*}g^{p_2},$$
$$v_1=(g^x)^{l_1},\ v_2=(g^x)^{-\mathcal{H}(ID^*)}g^{l_2}.$$

$\mathcal{C}$ 选择随机数 $h',h'',h'_1,h''_1,h'_2,h''_2,\cdots,h'_n,h''_n\in Z_p$ 中，并设置

$$h=(g^y)^{h'}\cdot g^{h''},$$
$$h_i=(g^y)^{h'_i}\cdot g^{h''_i}:(i\in[n]).$$

$\mathcal{C}$ 实际上设置了 $s=h'y+h''$ 和 $\{s_i=h'_iy+h''_i\}_{i\in[n]}$.

最后，$\mathcal{C}$ 将以下 PK 发送给 $\mathcal{A}$：

$$PK=(\mathbb{G},\mathbb{G}_T,g,p,e,u_1,u_2,v_1,v_2,h,h_1,\cdots,h_n).$$

Queries：$\mathcal{A}$ 可以对 $\mathcal{C}$ 进行 Q 次密钥查询，但是，存在以下限制：$ID_i\neq ID_j$ 和 $ID_i\neq ID^*$，其中 $i,j\in[Q]$. $\mathcal{C}$ 执行以下操作以响应来自 $\mathcal{A}$ 的密钥查询 $SK^{ID}_{\boldsymbol{y}}$.

$\mathcal{C}$ 选择 $t'\in Z_p$，隐含地设置 $t=\dfrac{ID\cdot\left(\sum\limits_{i\in[n]}y_ih'_i+h'\right)}{\mathcal{H}(ID)-\mathcal{H}(ID^*)}y+t'$，并生成

$$K_t=g^t=(g^y)^{\frac{ID\cdot\left(\sum\limits_{i\in[n]}y_ih'_i+h'\right)}{\mathcal{H}(ID)-\mathcal{H}(ID^*)}}g^{t'},$$

$$K_h=(u_1^{ID}u_2)^{\sum\limits_{j=1}^{n}s_jy_j+s}(v_1^{\mathcal{H}(ID)}v_2)^{-t}$$
$$=((g^x)^{ID}\cdot(g^z)^{ID-ID^*}\cdot g^{p_1ID+p_2})^{y\cdot\left(\sum\limits_{i\in[n]}h'_iy_i+h'\right)+\left(\sum\limits_{i\in[n]}h''_iy_i+h''\right)}\cdot((g^x)^{\mathcal{H}(ID)-\mathcal{H}(ID^*)}\cdot g^{\mathcal{H}(ID)l_1+l_2})^{-\frac{ID\cdot\left(\sum\limits_{i\in[n]}y_ih'_i+h'\right)}{\mathcal{H}(ID)-\mathcal{H}(ID^*)}y+t'}$$
$$=(g^{xy})^{ID\cdot\left(\sum\limits_{i\in[n]}y_ih'_i+h'\right)}(g^{xy})^{-ID\cdot\left(\sum\limits_{i\in[n]}y_ih'_i+h'\right)}\cdot Ag^{zy}\cdot Bg^z\cdot Cg^x\cdot Dg^y\cdot E,$$

其中，A,B,C,D,E 是 $\mathcal{C}$ 能计算得到的整数，而 g^x,g^y,g^z,g^{zy} 为实例中给出的已知数. $\mathcal{C}$ 所未知的项 g^{xy} 在上面的计算中被抵消了.

Forge：$\mathcal{A}$ 向 $\mathcal{C}$ 提交伪造的密钥 $SK^{ID^*}_{\boldsymbol{y}^*}=(\boldsymbol{y}^*,K_t^*,K_h^*)$，其中 $\boldsymbol{y}^*=(y_1^*,y_2^*,\cdots,y_n^*)$ 是嵌入到密钥中的挑战向量. 若 $\sum\limits_{i\in[n]}y_i^*h'_i+h'=0\bmod p$，则 $\mathcal{C}$ 无法解决 CBDH 假设，并且中止游戏；否则，我们有

$$K_t^*=g^{t^*},$$

$$K_h^*=(u_1^{ID^*}u_2)^{\sum\limits_{j=1}^{n}s_jy_j^*+s}(v_1^{\mathcal{H}(ID^*)}v_2)^{-t}$$
$$=((g^x)^{ID^*}\cdot(g^z)^{ID^*-ID^*}\cdot g^{p_1ID^*+p_2})^{y\cdot\left(\sum\limits_{i\in[n]}h'_iy_i^*+h'\right)+\left(\sum\limits_{i\in[n]}h''_iy_i^*+h''\right)}\cdot((g^x)^{\mathcal{H}(ID^*)-\mathcal{H}(ID^*)}\cdot g^{\mathcal{H}(ID^*)l_1+l_2})^{-t^*}$$
$$=((g^x)^{ID^*}\cdot g^{p_1ID^*+p_2})^{y\cdot\left(\sum\limits_{i\in[n]}h'_iy_i^*+h'\right)+(h'_iy+h'')}\cdot(g^{\mathcal{H}(ID^*)l_1+l_2})^{-t^*}$$
$$=(g^{xy})^{ID^*\cdot\left(\sum\limits_{i\in[n]}y_i^*h'_i+h'\right)}\cdot(g^{t^*})^{-\mathcal{H}(ID^*)l_1-l_2}\cdot A'g^{x}\cdot B'g^{y}\cdot C',$$

其中，A',B',C' 是可由 $\mathcal{C}$ 计算得到的整数. $\mathcal{C}$ 输出：

$$g^{xy}=(K_h^*\cdot(K_t^*)^{\mathcal{H}(ID^*)l_1+l_2}\cdot(A'g^x\cdot B'g^y\cdot C')^{-1})^{\frac{1}{ID^*\cdot\left(\sum\limits_{i\in[n]}y_i^*h'_i+h'\right)}}.$$

现在分析 $\mathcal{C}$ 成功解决 CBDH 假设的概率. 显然，游戏唯一的终止条件是 $\sum\limits_{i\in[n]}y_i^*h'_i+h'=0\bmod p$，但是，此事件可以忽略不计. 考虑以下情况：即使敌手 $\mathcal{A}$ 可以解决 DL 假设，并知

道值 y，也无法以不可忽略的概率生成满足 $\sum_{i\in[n]} y_i^* h_i' + h' = 0 \bmod p$ 的向量 $\boldsymbol{y}^*$. 如果 $\mathcal{A}$ 可以解决 DL 假设，则其可以从 PK 知道 $s = h'y + h''$ 和 $\{s_i = h_i'y + h_i''\}_{i\in[n]}$，即使 $\mathcal{A}$ 还知道 y，其也无法从上述等式导出 $h', h'', \{h_i', h_i''\}_{i\in[n]}$，因为在 $n+1$ 个两两独立的方程中有 $2n+2$ 个未知数. 给定 s(或 s_i)，就有满足上面等式的 p 对 $\langle h', h''\rangle$(或 $\langle h_i', h_i''\rangle$). $\mathcal{A}$ 仅能以 $1/p$ 的概率"猜测"出正确的对，这个概率是可以忽略不计的. 因此，$\mathcal{A}$ 生成满足 $\sum_{i\in[n]} y_i^* h_i' + h' = 0 \bmod p$ 的向量 $\boldsymbol{y}^*$ 的概率是可忽略的.

因此，如果对手 $\mathcal{A}$ 以不可忽略的优势 ε 攻破了 ID-PK-IPFE 的 s-IMA 安全性，则可以构造算法以不可忽略的优势 ε 解决 CBDH 假设.

2.3 s-VMA 的安全性分析

定理 5.　如果 CBDH 假设是困难的，则 ID-PK-IPFE 是 s-VMA 安全的.

证明.　如果对手 $\mathcal{A}$ 可以以不可忽略的优势 ε 攻破 ID-PK-IPFE 的 s-VMA 安全性，则可以构造挑战者 $\mathcal{C}$ 以不可忽略的优势解决 CBDH 假设. 以下为 $\mathcal{A}$ 和 $\mathcal{C}$ 之间的交互过程.

Init：$\mathcal{A}$ 选择并公布挑战身份 ID^* 和挑战向量 $\boldsymbol{y}^* = (y_1^*, \cdots, y_n^*)$.

Setup：$\mathcal{C}$ 以 CBDH 假设的实例 $(\mathbb{G}, \mathbb{G}_T, e, g, g^x, g^y, g^z, g^{yz})$ 作为输入，其任务是输出 g^{xy}.

$\mathcal{C}$ 选择一个安全参数 λ 和一个防碰撞哈希函数 $\mathcal{H}$. 此外，$\mathcal{C}$ 随机选择 $p_1, p_2, l_1, l_2 \in Z_p$，并设置

$$u_1 = g^x g^z g^{p_1},\ u_2 = (g^z)^{-ID^*} g^{p_2},$$

$$v_1 = (g^x) g^{l_1},\ v_2 = (g^x)^{-\mathcal{H}(ID^*)} g^{l_2}.$$

$\mathcal{C}$ 选择 $h'', h_1', h_1'', h_2', h_2'', \cdots, h_n', h_n'' \in Z_p$，并设置

$$h = (g^y)^{-\sum_{i\in[n]} h_i' y_i^*} \cdot g^{h''},$$

$$h_i = (g^y)^{h_i'} \cdot g^{h_i''} : (i \in [n]).$$

在上式中，$\mathcal{C}$ 实际上隐含地设置了 $s = h'' - y \cdot \sum_{i\in[n]} h_i' y_i^*$ 和 $\{s_i = h_i'y + h_i''\}_{i\in[n]}$.

最后，$\mathcal{C}$ 将以下 PK 发送给 $\mathcal{A}$：

$$PK = (\mathbb{G}, \mathbb{G}_T, g, p, e, u_1, u_2, v_1, v_2, h, h_1, \cdots, h_n).$$

Queries：$\mathcal{A}$ 可以向 $\mathcal{C}$ 查询 Q 个密钥 $SK_{\boldsymbol{y}_i}^{ID_i} : (i \in [Q])$，其限制为：$ID_i \neq ID_j$，其中 $i, j \in [Q]$，以及当查询挑战身份 ID^* 时，该身份必须与挑战矢量 $\boldsymbol{y}^*$ 绑定. $\mathcal{C}$ 执行以下操作以响应来自 $\mathcal{A}$ 的密钥查询 $SK_{\boldsymbol{y}}^{ID}$.

1. 若 $ID \neq ID^*$，$\mathcal{C}$ 选择 $t' \in Z_p$，并隐含地设置 $t = \frac{ID \cdot \sum_{i\in[n]} (y_i - y_i^*) h_i'}{\mathcal{H}(ID) - \mathcal{H}(ID^*)} y + t'$，然后生成

$$K_t = g^t = (g^y)^{\frac{ID \cdot \sum_{i\in[n]} (y_i - y_i^*) h_i'}{\mathcal{H}(ID) - \mathcal{H}(ID^*)}} g^{t'},$$

$$K_h = (u_1^{ID} u_2)^{\sum_{j=1}^{n} s_j y_j + s} (v_1^{\mathcal{H}(ID)} v_2)^{-t}$$

$$= ((g^x)^{ID} \cdot (g^z)^{ID - ID^*} \cdot g^{p_1 ID + p_2})^{y \cdot \sum_{i\in[n]} h_i' (y_i - y_i^*) + \left(\sum_{i\in[n]} h_i'' y_i^* + h''\right)} \cdot$$

$$((g^x)^{\mathcal{H}(ID) - \mathcal{H}(ID^*)} \cdot g^{\mathcal{H}(ID) l_1 + l_2})^{-\frac{ID \cdot \left(\sum_{i\in[n]} (y_i - y_i^*) h_i'\right)}{\mathcal{H}(ID) - \mathcal{H}(ID^*)} y + t'}$$

$$= (g^{xy})^{ID \cdot \left(\sum_{i\in[n]} h_i' (y_i - y_i^*)\right)} (g^{xy})^{-ID \cdot \left(\sum_{i\in[n]} h_i' (y_i - y_i^*)\right)} \cdot$$

$$A g^{z y} \cdot B g^z \cdot C g^x \cdot D g^y \cdot E,$$

其中，A, B, C, D, E 是 $\mathcal{C}$ 可以通过计算得到的整数，而 g^x，g^y，g^z，g^{zy} 则是假设实例中给出的已知项. 注意到，$\mathcal{C}$ 未知的项 g^{xy} 在上式中被抵消了.

2. 若 $ID = ID^*$ 且 $\boldsymbol{y} = \boldsymbol{y}^*$ 成立，$\mathcal{C}$ 选择 $t' \in Z_p$ 并设置 $t = t'$，然后生成

$$K_t = g^t = g^{t'},$$

$$K_h = (u_1^{ID^*} u_2)^{\sum_{j=1}^{n} s_j y_j^* + s} (v_1^{\mathcal{H}(ID^*)} v_2)^{-t}$$

$$= ((g^x)^{ID^*} \cdot (g^z)^{ID^* - ID^*} \cdot g^{p_1 ID^* + p_2})^{y \cdot \sum_{i\in[n]} h_i' (y_i^* - y_i^*) + \left(\sum_{i\in[n]} h_i'' y_i^* + h''\right)} \cdot$$

$$((g^x)^{\mathcal{H}(ID^*) - \mathcal{H}(ID^*)} \cdot g^{\mathcal{H}(ID^*) l_1 + l_2})^{-t'}$$

$$= ((g^x)^{ID^*} \cdot g^{p_1 ID^* + p_2})^{\sum_{i\in[n]} h_i'' y_i^* + h''} \cdot (g^{\mathcal{H}(ID^*) l_1 + l_2})^{-t'}$$

$$= A' g^x \cdot B',$$

其中，A', B' 是 $\mathcal{C}$ 可以通过计算得到的整数.

Forge：设 $\mathcal{A}$ 提交的伪造密钥为 $SK_{\boldsymbol{y}'^*}^{ID^*} = (\boldsymbol{y}'^*, K_t^*, K_h^*)$，其中 $\boldsymbol{y}'^* \neq \boldsymbol{y}^*$. 若 $\sum_{i\in[n]} (y_i'^* - y_i^*) h_i' = 0 \bmod p$，则 $\mathcal{C}$ 无法解决 CBDH 假设，并且中止游戏；否则，我们有

$$K_t^* = g^{t^*},$$

$$K_h^* = (u_1^{ID^*} u_2)^{\sum_{j=1}^{n} s_j y_j'^* + s} (v_1^{\mathcal{H}(ID^*)} v_2)^{-t^*}$$

$$= ((g^x)^{ID^*} \cdot (g^z)^{ID^* - ID^*} \cdot g^{p_1 ID^* + p_2})^{y \cdot \sum_{i\in[n]} h_i' (y_i'^* - y_i^*) + \left(\sum_{i\in[n]} h_i'' y_i'^* + h''\right)} \cdot$$

$$((g^x)^{\mathcal{H}(ID^*) - \mathcal{H}(ID^*)} \cdot g^{\mathcal{H}(ID^*) l_1 + l_2})^{-t^*}$$

$$= (g^{xy})^{ID^* \cdot \left(\sum_{i\in[n]} h_i' (y_i'^* - y_i^*)\right)} \cdot (g^{t^*})^{-\mathcal{H}(ID^*) l_1 - l_2} \cdot A'' g^{x} \cdot B'' g^{y} \cdot C'',$$

其中，A'', B'', C'' 可由 $\mathcal{C}$ 计算得到. $\mathcal{C}$ 输出：

$$g^{xy} = \left(K_h^* \cdot (K_t^*)^{\mathcal{H}(ID^*) l_1 + l_2} \cdot (A'' g^x \cdot B'' g^y \cdot C'')^{-1}\right)^{\frac{1}{ID^* \cdot \left(\sum_{i\in[n]} h_i' (y_i'^* - y_i^*)\right)}}.$$

以下分析 $\mathcal{C}$ 成功解决 CBDH 假设的概率. 显然，该游戏唯一可能终止的条件是 $\sum_{i\in[n]} (y_i'^* - y_i^*) h_i' = 0 \bmod p$，与上一部分的 s-IMA 安全证明的分析类似，此事件发生的概率可以忽略不计. 因此，如果敌手 $\mathcal{A}$ 以不可忽略的优势 ε 攻破了 ID-PK-IPFE 的 S-VMA 安全性，则可以构造算法以不可忽略的优势 ε 解决 CBDH 假设.

3. 效率分析

本节将对 ID-PK-IPFE 方案效率进行理论和实验分析.

3.1 理论效率分析

令 M_p, M_g, M_t 分别表示群 Z_p, G, G_T 上的模乘运算消耗的时间；E_g, E_t 分别表示在群 G, G_T 中的幂运算所消耗的时间；BM 表示双线性映射运算所消耗的时间；M 表示普通乘法计算所消耗的时间；n 表示系统中设定的向量的长度，DL 表示解离散对数所消耗的时间. 以下的附表 1 列举出本文方案与文献[6]和文献[7]方案的效率对比.

附表 1 本文方案与文献[6]和文献[7]方案的效率对比表

算法	本文方案	文献[6]方案	文献[7]方案
Setup	$(n+1)E_g$	$(n+1)E_g$	$(n+2)E_g$
Encrypt	$2nBM+nE_t+(n+4)E_g+(n+2)M_g$	$(2n+1)E_g+M_g$	$(2n+1)E_g+M_g$
KeyGen	$3M_g+4E_g$	nM	$2nM$
Verify	$3BM+(n+2)E_g+(n+2)M_g$	—	—
Decrypt	$(n+2)E_t+4M_t+2BM+DL$	$(n+1)E_t+M_t+DL$	$(n+2)E_t+M_t+DL$
指定接收者	Y	N	N
密钥可更改	N	Y	Y

从附表 1 可以看出，文献[6]和文献[7]方案的效率几乎相同，但文献[7]方案的效率稍低，这是由于文献[7]方案对文献[6]方案进行了改进，把其安全性从选择性的安全性改成了适应性的安全性.

本文方案与文献[6]和文献[7]方案相比，效率偏低. 然而，本文的方案在牺牲效率的前提下增加了两个功能：其一，可以指定密文的接收者；其二，本文方案的密钥不可更改，而文献[6]和文献[7]方案的密钥均可被改变. 因此，与文献[6]和文献[7]方案相比，本文方案具有更灵活的授权访问机制，且在一些应用场景下，可抵抗敌手的密钥修改攻击.

3.2 实验效率分析

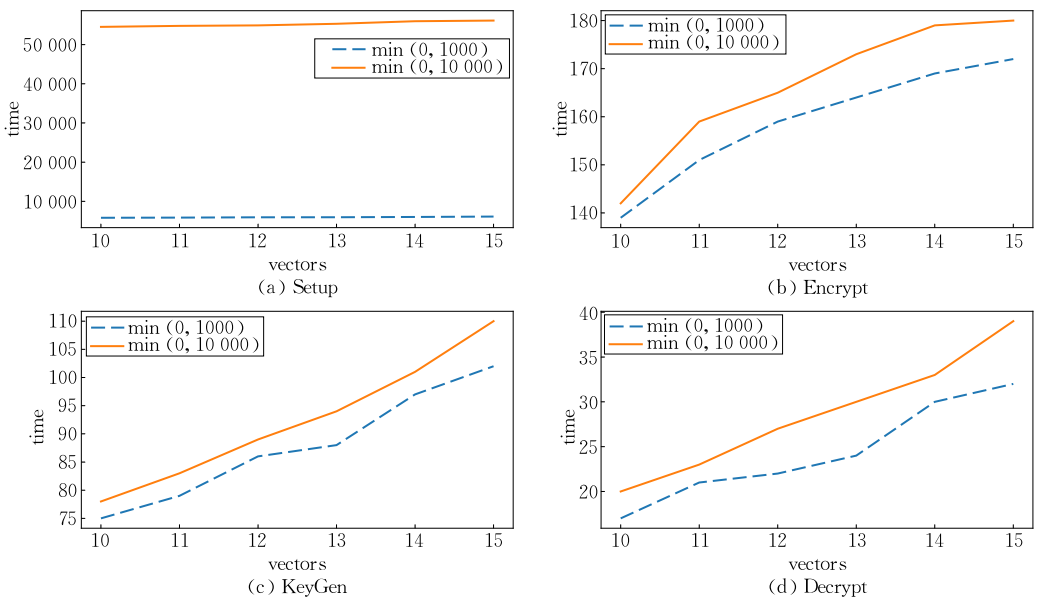
尽管本文方案和文献[6]和文献[7]方案相比，效率稍低，但本文方案在应用场景下是实用的. 本文使用 JPBC 库(http://gas.dia.unisa.it/projects/jpbc/) 在一台 CPU 为 i7-6700 3.40 GHz，内存为 8.00 GB，操作系统为 Windows 7 64-bit 的个人 PC 机上实现了本文的方案.

本文在 Setup 算法中添加了预处理阶段：在该阶段，程序将预先计算 $e(g,g)^m$ 的值，并将预先计算的结果存放到 hash 表中，待解密消息时可供查询. 我们分别进行了两组实验，在第一组实验中，m 的范围为(0,1000)，而在第二组实验中，m 的范围为(0,10000). 在(0,10000)范围内时，大多数数据统计应用程序的需求都可以满足. 本文设定实验中向量的长度均从 10 增加到 15. 在以下的附表 2 中列出了各个算法在这两组实验中花费的时间. 在附图 1 中分别比较了 $m\in(0,1000)$ 和 $m\in(0,10000)$ 时算法所花费的时间.

从附表 2 可以看出，Encrypt、KeyGen、Verify 和 Decrypt 这四种算法所花费的时间是可以接受的. 在附图 1 中展示了 Setup、KeyGen、Encrypt 和 Decrypt 这四个算法在两个实验里的效率对比图. 根据提供的数据可知，仅 Setup 算法需要

附表 2 $m\in(0,1000)$ 和 $m\in(0,10000)$ 时各个算法消耗的时间 (单位：ms)

算法	$m\in(0,1000)$						$m\in(0,10000)$					
	10	11	12	13	14	15	10	11	12	13	14	15
Setup	5804	5847	5925	5939	6012	6113	54533	54788	54912	55321	55977	56143
Encrypt	139	151	159	164	169	172	142	159	165	173	179	180
KeyGen	75	79	86	88	97	102	78	83	89	94	101	110
Verify	86	89	93	95	99	101	89	94	97	103	108	115
Decrypt	17	21	22	24	30	32	20	23	27	30	33	39



附图 1 本文方案 Setup、Encrypt、KeyGen 和 Decrypt 算法两组实验效率比较

较长时间，因为在该算法中的预处理计算并将计算结果保存到 hash 表的过程将消耗一定的时间成本. 然而，在解密时，算法并不需要解决离散对数问题，而只需要在哈希表中搜索数据，解密时间被大大缩短了. 另一方面，尽管 Setup 算法花费的时间较长，但系统只需要运行一次该算法，在以后的处理过程中无需再运行该算法，因此这种时间消耗是可以接受的.

DENG Yu-Qiao, Ph. D., associate professor. His research interests include cryptography and cloud computing.

SONG Ge, Ph. D., lecturer. Her research interests focus on cryptography.

YANG Bo, Ph. D., professor. His research interests include information security and cryptography.

PENG Chang-Gen, Ph. D., professor. His research interest is cryptography.

TANG Chun-Ming, Ph. D., professor. His research interests include information security and cryptography.

WEN Ya-Min, Ph. D., associate professor. Her research interests include cryptography and cloud computing.

Background

Functional encryption (FE) is a versatile cryptographic primitive first formalized by Boneh et al.. An increasing number of studies have considered the construction of generic FE that implements universal circuits since the emergence of FE. However, these works only introduced the requirement for heavy-duty tools, such as indistinguishable obfuscation and multilinear maps; therefore, the practicality of these works remains uncertain. In PKC 2015, Abdalla et al. proposed a new primitive, namely, inner product encryption (IPE). IPE is a special case of FE that executes the computation for the inner product of vectors; IPE is remarkably useful in applications, such as privacy-preserving statistical analysis and conjunctive/disjunctive normal-form formulas.

IPE can be classified into two categories, namely, public-key IPE (PK-IPE) and secret-key IPE (SK-IPE). In the PK-IPE setting, one vector is encoded by the ciphertext, whereas the other vector is encoded by the private key. A user can derive the inner product, if he holds the ciphertext and secret key. In the SK-IPE setting, the situation is similar, except that a master secret key (MSK) (i. e., a secret key maintained by the authority) should be used in generating ciphertext and private key. Thus, the ciphertext cannot be independently formed by an encryptor without an interaction with an authority in SK-IPE. Therefore, SK-IPE is impractical for applications where many users should generate ciphertexts simultaneously, because it inevitably influences the performance of the authority.

This study mainly focuses on the inability to design the recipient, and secret key cannot be verified in the presented PK-IPE. We first propose a PK-IPE-DRVS scheme by using the bilinear maps technique to address the abovementioned problems. We refine the sv-CPA security models of PK-IPE and initially propose the RIMA and SVMA models to capture the secret key modification behaviors of adversaries. Finally, we prove the security of PK-IPE-DRVS under these models.

The management and high cost of many public keys may become a problem, although the recipient can be designed in PK-IPE-DRVS. Thus, to propose a PK-IPE-DRVS with constant-length public parameters may be an interesting and challenging problem. This work is supported by the National Key R&D Program of China (No. 2017YFB0802000), the National Natural Science Foundation of China (61772147, 61300204, 62002122), the Guangdong Province Natural Science Foundation of Major Basic Research and Cultivation Project (2015A030308016), the Natural Science Foundation of Guangdong Province (2015A030313630, 2019A1515011797), the Basic Research Project of Guangdong Provincial Department of Education (2014KZDXM044), the Colleges and Universities Innovation Team Construction Project Guangdong Province (2015KCXTD014), the National Cryptography Development Fund (MMJJ20170117), the Guangzhou City Bureau of Cooperative Innovation Project (1201610005) and the Project of Guangdong Science and Technology Plan (2016A020210103, 2017A020208054).