文章编号:1671-4229(2023)04-0042-14

内积函数加密技术研究与进展

栗亚敏[1]，李　进[2]，陈晓峰[1*]
(1. 西安电子科技大学 网络与信息安全学院，陕西 西安　710071；
2. 广州大学 计算机科学与网络工程学院，广东 广州　510006)

摘　要：内积函数加密技术由 Abdalla 等在 2015 年引入,它允许解密者使用某个向量对应的密钥去解密另一个向量对应的密文,从而得到这两个向量的内积,但又不泄露密文所对应向量的任何其他信息。该技术可以在保证数据机密性的同时,保留对数据的内积计算能力。因此,在涉及内积计算的场景中,相比于传统的加密技术,内积函数加密技术提供了更加灵活的访问控制。内积函数加密技术在统计分析、外包计算、机器学习等场景中有着实际的应用价值。目前,内积函数加密技术的研究仍处于理论研究阶段,主要表现在大部分方案限制所支持的明文空间大小,方案所能达到的安全性较弱,或是方案本身难以实现。文章给出了内积函数加密技术的研究与进展,分别介绍了不同类型内积函数加密技术的形式化定义和安全性模型,系统地介绍了公钥内积函数加密和私钥内积函数加密的研究进展,阐述了内积函数加密的应用研究和相似工作,并对关键的研究工作和技术进行了总结和展望。

关键词：内积函数加密；访问控制；函数隐藏性；多客户端函数加密
中图分类号：TP 309　　**文献标志码**：A

A survey on inner product functional encryption

LI Ya-min[1], *LI Jin*[2], *CHEN Xiao-feng*[1*]
(1. School of Cyber Engineering, Xidian University, Xi'an 710071, China;
2. School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China)

Abstract: Inner product functional encryption was introduced in 2015 by Abdalla et al. It allows the decryptor, having the key corresponding to a vector, to decrypt the ciphertext corresponding to another vector and obtain the inner product of these two vectors without revealing any other information of the vector in the ciphertext. In this way, the data confidentiality can be guaranteed while the inner product calculation ability of the data is preserved. Therefore, in the scenario involving inner product, inner product functional encryption provides more flexible access control than traditional encryption. Inner product functional encryption technology has practical application value in statistical analysis, outsourcing computing, machine learning, and other scenarios. At present, the research on inner product functional encryption is still at the theoretical stage, which is mainly manifested in that most of the schemes limit the size of the supported plaintext space, the security that the scheme can achieve is weak, or the scheme itself is difficult to achieve. This paper gives a survey on inner product functional encryption. First, the formal definitions and security models of different types of inner product functional encryption schemes are introduced. Then the research progress of public-key inner product

收稿日期：2023-02-10；**修回日期**：2023-04-24
基金项目：国家自然科学基金资助项目(61960206014)
作者简介：栗亚敏(1997—),女,博士研究生. E-mail:liyamin97@126.com
* 通信作者. E-mail:xfchen@xidian.edu.cn
引文格式：栗亚敏，李进，陈晓峰. 内积函数加密技术研究与进展[J]. 广州大学学报(自然科学版),2023,22(4):42－55.

functional encryption and private-key inner product functional encryption are systematically discussed. The applications and other similar works of inner product functional encryption are described. Finally, the key research works and technologies of inner product functional encryption are summarized with some prospects.

Key words: inner product functional encryption; access control; function privacy; multi-client functional encryption

加密技术是保障数据机密性的重要工具。在传统加密中,解密是一个全有或全无的事件(即解密者要么能完全恢复出整个明文,要么什么都得不到)。函数加密(Functional Encryption, FE)[1-2]又称为功能加密,是一种现代的加密原语,它在保证数据机密性的同时,保留了数据的可用性。在 FE 中,密钥 sk_f 对应函数 f,密文 Ct 对应 f 定义域中的某个输入 x。给定密钥 sk_f 和密文 Ct,用户解密可得函数值 $f(x)$。函数加密最基本的安全性要求是,解密者仅能获得 $f(x)$ 而无法得到 x 的任何其他信息。自 FE 出现以来,许多学者研究构建支持通用功能的 FE 方案[3-8],例如任意多项式大小的电路或图灵机。然而,所有这样的构造都依赖于强大的密码原语,如不可区分混淆或多线性映射,目前都不能使用有效的构建块,或在经过充分研究的密码假设下实例化。在 PKC 2015 上,Abdalla 等[9]提出了内积函数加密(inner product functional encryption)的原语。内积函数加密是函数加密的一种特殊情况,它支持向量内积的计算。在内积函数加密中,消息以向量的形式表示,密文 $Ct_{\vec{x}}$ 对应消息向量 $\vec{x}$(向量长度为 m),密钥 $sk_{\vec{y}}$ 对应向量 $\vec{y}$(向量长度为 m),解密者使用密钥 $sk_{\vec{y}}$ 对密文 $Ct_{\vec{x}}$ 解密可得内积值 $\langle\vec{x},\vec{y}\rangle$,而无法得到 $\vec{x}$ 的任何其他信息。因此,内积函数加密对涉及隐私保护内积计算的应用具有重要的意义。具体来说,内积函数加密可以用于隐私保护的统计分析、外包计算、加密生物特征认证和机器学习等场景中。目前,已有许多工作研究了内积函数加密方案的构造方法,但现有的内积函数加密方案在效率、安全性、灵活性和实用性方面还存在不足。①大部分方案在解密时需要求解离散对数问题,当加密数据较大时,往往解密效率较低;②目前能实现紧不可区分性安全或模拟安全的方案还较少;③大多数方案都是静态的,即方案所支持的用户数量是固定的,不支持用户的动态加入与退出,这使得内积函数加密的灵活性较差;④传统的内积函数加密存在固有的缺陷,即通过 m 个线性无关的向量所对应的解密密钥可以将明文 $\vec{x}$ 完全恢复出来,这使得内积函数加密的实用性较差。如何设计更高效、更安全、更灵活、更实用的内积函数加密方案仍是一个难题。

本文主要概述了内积函数加密技术的研究进展。第 1 节介绍了内积函数加密的不同分类方法,给出了不同类型内积函数加密的形式化定义;第 2 节详细论述了内积函数加密技术的研究进展;第 3 节阐述了内积函数加密的应用;第 4 节介绍了与内积函数加密技术类似的工作;第 5 节对关键的研究工作和技术进行了总结和展望。

1　内积函数加密的分类及形式化定义

本节将对内积函数加密的分类和不同类型内积函数加密的定义进行概述。

1.1　内积函数加密的分类

随着内积函数加密技术的不断发展,内积函数加密方案按照不同分类标准有着不同的分类方法,如图 1 所示。

根据加密时所使用密钥的不同,内积函数加密可以分为两大类,即公钥内积函数加密(public-key inner product functional encryption)和私钥内积函数加密(private-key inner product functional encryption)。在公钥内积函数加密中,数据拥有者使用主公钥 mpk 将消息向量 $\vec{x}$ 加密为密文 $Ct_{\vec{x}}$,而在私钥内积函数加密中,主公钥 mpk 不足以用来加密(它实际上是一些不需要保密的公共参数),密文的生成还需要使用主私钥 msk。根据数据源个数划分,内积函数加密可以分为单数据源内积函数加密和多数据源内积函数加密。一般地,在没有特别说明时,内积函数加密往往指的是单数据源内积函数加密,又可以称为单输入内积函数加密,所实现的是单输入内积功能,即数据拥有者对自己的数据 $\vec{x}$ 加密,解密者使用解密密钥 $sk_{\vec{y}}$ 解密得到内积值 $\langle\vec{x},\vec{y}\rangle$。多数据源内积函数加密所实现的是多输入内积功能,即系统中存在多个数据拥有者(假设数据拥有者个数为 n),数据拥有者 i 加密自己的数据 $\vec{x}_i$,解密者使用向量 $\vec{y}=(\vec{y}_1,\cdots,\vec{y}_n)$(表示 n 个向量的级联)对应的解密密钥 $sk_{\vec{y}}$ 同时解密 n 个密文得到多输入内积值 $\sum_{i=1}^{n}\langle\vec{x}_i,\vec{y}_i\rangle$。多数据源

内积函数加密又包括多输入内积函数加密(multi-input functional encryption for inner product)、多客户端内积函数加密(multi-client functional encryption for inner product)和去中心化多客户端内积函数加密(decentralized multi-client functional encryption for inner product)。根据所实现的隐私性划分,内积函数加密又可以分为两类,即消息隐藏的内积函数加密和全隐藏的内积函数加密(full-hiding inner product functional encryption)。一般地,在没有特别说明时,内积函数加密往往指的是消息隐藏的内积函数加密,即方案只能保护消息的机密性。全隐藏的内积函数加密,又称为函数隐藏的内积函数加密(function-hiding inner product functional encryption),它不仅可以保证消息的机密性,同时还可以保护函数的隐私性。另外,根据所具有性质的不同,内积函数加密可以粗略地分为普通的内积函数加密和具有特殊性质的内积函数加密。一般地,在没有特别说明时,内积函数加密指的是普通的内积函数加密。目前,还存在一部分工作研究具有特殊性质的内积函数加密,如可追踪的内积函数加密、盲内积函数加密、基于层次身份的内积函数加密、支持细粒度访问控制的内积函数加密等。

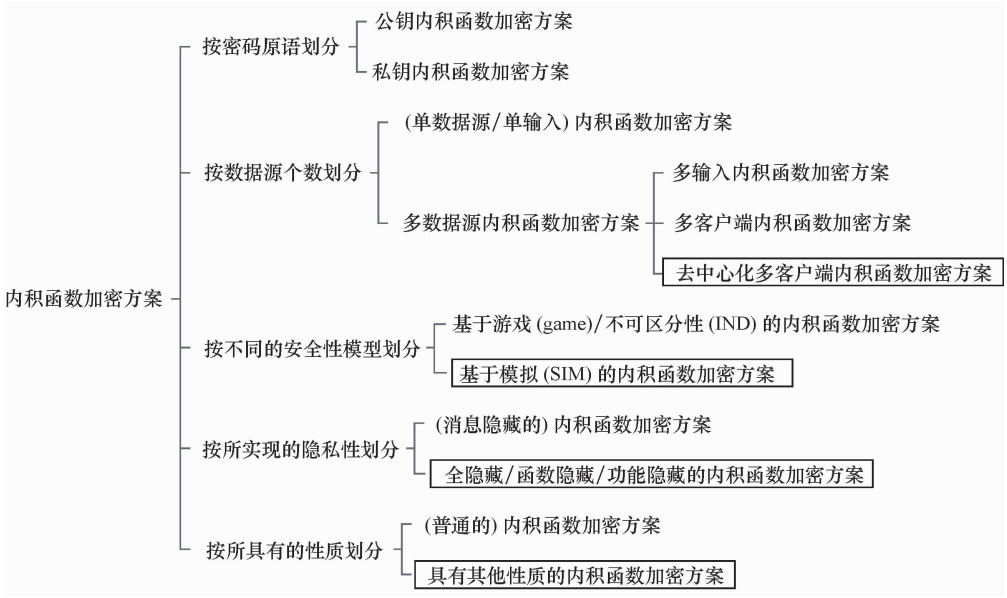

图1 内积函数加密方案分类

Fig. 1 Taxonomy of inner product functional encryption schemes

图1中方框部分表示同一种分类方法下的最优类别,也就是说一个最优的内积函数加密方案应是基于模拟安全的、全隐藏的去中心化多客户端内积函数加密方案。然而,目前不存在同时满足以上几种性质且安全的内积函数加密方案。

1.2 单输入内积函数加密的定义

1.2.1 单输入内积函数加密的形式化定义

针对内积函数簇 F^m(其中,$f_{\vec{y}} \in F^m : \mathbb{Z}^m \to \mathbb{Z}$)的公钥单输入内积函数加密方案包含以下4个算法:

- $\mathrm{Setup}(1^\lambda, m) \to (mpk, msk)$:参数设置算法。输入安全参数 λ,向量长度 m,输出主公钥 mpk 和主私钥 msk。
- $\mathrm{KeyGen}(msk, \vec{y}) \to sk_{\vec{y}}$:密钥生成算法。输入主私钥 msk 和函数向量 $\vec{y} \in \mathbb{Z}^m$,输出解密密钥 $sk_{\vec{y}}$。
- $\mathrm{Enc}(mpk, \vec{x}) \to Ct_{\vec{x}}$:加密算法。输入主私钥 msk,消息向量 $\vec{x} \in \mathbb{Z}^m$,输出密文 $Ct_{\vec{x}}$。
- $\mathrm{Dec}(sk_{\vec{y}}, Ct_{\vec{x}}) \to \langle \vec{x}, \vec{y} \rangle$ 或 $\perp$:解密算法。输入解密密钥 $sk_{\vec{y}}$ 和密文 $Ct_{\vec{x}}$,输出 $\langle \vec{x}, \vec{y} \rangle$ 或 $\perp$。

正确性:针对内积函数簇 F^m(其中,$f_{\vec{y}} \in F^m : \mathbb{Z}^m \to \mathbb{Z}$)的公钥单输入内积函数加密方案是正确的,如果对于任意的安全参数 λ,存在可忽略函数 ε,使得

$$\Pr\left[\mathrm{Dec}(sk_{\vec{y}}, Ct_{\vec{x}}) \to \langle \vec{x}, \vec{y} \rangle \;\middle|\; \begin{array}{c} (mpk, msk) \leftarrow \mathrm{Setup}(1^\lambda, m), \\ sk_{\vec{y}} \leftarrow \mathrm{KeyGen}(msk, \vec{y}), \\ Ct_{\vec{x}} \leftarrow \mathrm{Enc}(mpk, \vec{x}) \end{array}\right] < \varepsilon(\lambda)。$$

在私钥单输入内积函数加密中,主公钥 mpk 不足以用来加密,它实际上指的是一些不需要保密的公共参数,加密算法被替换为 $\mathrm{Enc}(msk, \vec{x}) \to Ct_{\vec{x}}$,其他算法保持不变。

1.2.2 单输入内积函数加密的安全性

(1)消息隐藏性

函数加密的标准安全性定义是基于游戏(game)的

安全性,又称为不可区分性(Indistinguishability,IND)安全性。非正式地说,它要求对密钥生成预言机具有访问权限的敌手在一定的约束条件下无法区分两个消息 $\vec{x}^0$,$\vec{x}^1$ 中哪一个被加密。Boneh 等[1] 和 O'Neill[2] 表明 IND 安全性定义很弱,因为存在不安全的方案无论如何都可以被证明是 IND 安全的。所以,他们提出了一种更强的基于模拟(Simulation,SIM)的安全性定义。

针对公钥单输入内积函数加密,这里给出消息隐藏性的 4 种安全性定义 $yy-zz$,其中,$yy \in \{SEL, AD\}$ 表示选择性或自适应,$zz \in \{IND, SIM\}$ 是指 IND 安全性或 SIM 安全性。

公钥单输入内积函数加密方案的 $yy-IND$ 安全性定义可以通过表 1 中的实验 $yy-IND_\beta^{FE}(1^\lambda, A)$ 来描述。具体地,当 $yy=AD$ 时,

- 设置:挑战者运行算法 $\text{Setup}(1^\lambda, m)\rightarrow(mpk, msk)$,将主公钥 mpk 发送给敌手 A;
- 询问:敌手 A 向预言机 $QKeyG(\cdot)$ 自适应地提交 $\vec{y}$,返回 $sk_{\vec{y}}=\text{KeyGen}(msk,\vec{y})$;
- 挑战:敌手 A 提交一对挑战向量 $(\vec{x}^0,\vec{x}^1)$(满足 $\langle\vec{x}^0,\vec{y}\rangle=\langle\vec{x}^1,\vec{y}\rangle$),挑战者选择 $\beta\leftarrow\{0,1\}$,将挑战密文 $Ct_{\vec{x}^\beta}\leftarrow\text{Enc}(mpk,\vec{x}^\beta)$ 返回给敌手 A;
- 询问:敌手 A 继续向预言机 $QKeyG(\cdot)$ 自适应地提交 $\vec{y}$(满足 $\langle\vec{x}^0,\vec{y}\rangle=\langle\vec{x}^1,\vec{y}\rangle$),返回 $sk_{\vec{y}}=\text{KeyGen}(msk,\vec{y})$;
- 猜测:敌手 A 猜测 $\beta'\leftarrow\{0,1\}$。

该实验中,要求敌手 A 所提交的向量必须满足 $\langle\vec{x}^0,\vec{y}\rangle=\langle\vec{x}^1,\vec{y}\rangle$,否则敌手运行算法 $\text{Dec}(sk_{\vec{y}},Ct_{\vec{x}^\beta})\rightarrow\langle\vec{x}^\beta,\vec{y}\rangle$,并将解密结果分别与内积值 $\langle\vec{x}^0,\vec{y}\rangle$、$\langle\vec{x}^1,\vec{y}\rangle$ 比较,就可以容易地猜出 β。当 $yy=SEL$ 时,要求敌手在游戏开始前提交挑战明文 $(\vec{x}^0,\vec{x}^1)$。

表 1　$yy-zz$ 安全性定义中的实验描述

Table 1　Experiments in the security definition of $yy-zz$

$yy-IND_\beta^{FE}(1^\lambda,A)$:	$yy-REAL^{FE}(1^\lambda,A)$:	$yy-IDEAL^{FE}(1^\lambda,A)$:
当 $yy=SEL$ 时,	当 $yy=SEL$ 时,	当 $yy=SEL$ 时,
$(\vec{x}^0,\vec{x}^1)\leftarrow A(1^\lambda,F^m)$	$\vec{x}\leftarrow A(1^\lambda,F^m)$	$\vec{x}\leftarrow A(1^\lambda,F^m)$
$(mpk,msk)\leftarrow\text{Setup}(1^\lambda,m)$	$(mpk,msk)\leftarrow\text{Setup}(1^\lambda,m)$	$(mpk',msk')\leftarrow\text{Setup}'(1^\lambda,m)$
当 $yy=AD$ 时,	当 $yy=AD$ 时,	当 $yy=AD$ 时,
$(\vec{x}^0,\vec{x}^1)\leftarrow A^{QKeyG(\cdot)}(1^\lambda,F^m,mpk)$	$\vec{x}\leftarrow A^{QKeyG(\cdot)}(1^\lambda,F^m,mpk)$	$\vec{x}\leftarrow A^{O(\cdot)}(1^\lambda,F^m,mpk')$
$\beta\leftarrow\{0,1\}$	$Ct_{\vec{x}}\leftarrow\text{Enc}(mpk,\vec{x})$	$Ct'\leftarrow\text{Enc}'(mpk',msk')$
$Ct_{\vec{x}\beta}\leftarrow\text{Enc}(mpk,\vec{x}^\beta)$	$\alpha\leftarrow A^{QKeyG(\cdot)}(mpk,Ct_{\vec{x}})$	$\alpha\leftarrow A^{O(\cdot)}(mpk',Ct')$
$\beta'\leftarrow A^{QKeyG(\cdot)}(mpk,Ct_{\vec{x}\beta})$	输出:α	输出:α
输出:β'		

$yy-IND$ 安全性:一个公钥单输入内积函数加密方案实现了 $yy-IND$ 安全性,如果对于任意的 PPT 敌手 A,对于任意的安全参数 λ,存在可忽略函数 ε,使得

$$Adv_{FE,A}^{yy-IND}(\lambda)=|\Pr[yy-IND_0^{FE}(1^\lambda,A)=1]-\Pr[yy-IND_1^{FE}(1^\lambda,A)=1]|<\varepsilon(\lambda)。$$

公钥单输入内积函数加密的 $yy-SIM$ 安全性定义可以通过表 1 中的真实实验 $yy-REAL^{FE}(1^\lambda,A)$ 和理想实验 $yy-IDEAL^{FE}(1^\lambda,A)$ 来描述,其中,(Setup′,KeyGen′,Enc′)是模拟器算法。具体地,当 $yy=AD$ 时,在真实实验 $yy-REAL^{FE}(1^\lambda,A)$ 中,

- 挑战者运行算法 $\text{Setup}(1^\lambda,m)\rightarrow(mpk,msk)$,将主公钥 mpk 发送给敌手 A;
- 敌手 A 向预言机 $QKeyG(\cdot)$ 自适应地提交 $\vec{y}$,返回 $sk_{\vec{y}}=\text{KeyGen}(msk,\vec{y})$;
- 敌手 A 发送消息 $\vec{x}$ 给挑战者,挑战者将密文 $Ct_{\vec{x}}\leftarrow\text{Enc}(mpk,\vec{x})$ 返回给敌手 A;
- 敌手 A 继续向预言机 $QKeyG(\cdot)$ 自适应地提交 $\vec{y}$,返回 $sk_{\vec{y}}=\text{KeyGen}(msk,\vec{y})$;
- 实验输出敌手 A 的视图。

在理想实验 $yy-IDEAL^{FE}(1^\lambda,A)$ 中,

- 模拟器运行算法 $\text{Setup}'(1^\lambda,m)\rightarrow(mpk',msk')$,将模拟的主公钥 mpk' 发送给敌手 A;
- 敌手 A 向预言机 $O(\cdot)$ 自适应地提交 $\vec{y}$(预言机 $O(\cdot)$ 可以访问另一个预言机,输入 $\vec{y}$ 返回 $\langle\vec{x},\vec{y}\rangle$),然后预言机 $O(\cdot)$ 返回 $sk_{\vec{y}}=\text{KeyGen}'(msk',\vec{y},\langle\vec{x},\vec{y}\rangle)$;
- 敌手 A 选择消息 $\vec{x}$,模拟器将密文 $Ct'\leftarrow\text{Enc}'(mpk',msk')$ 返回给敌手 A;
- 敌手 A 继续向预言机 $O(\cdot)$ 自适应地提交 $\vec{y}$,预

言机 $O(\cdot)$返回 $sk_{\vec{y}}=\mathrm{KeyGen}'(msk',\vec{y},\langle\vec{x},\vec{y}\rangle)$;

- 实验输出敌手 A 的视图。

同样地,当 $yy=SEL$ 时,要求敌手在游戏开始前选择明文 $\vec{x}$。

$yy-SIM$ 安全性:一个公钥单输入内积函数加密方案具有 $yy-SIM$ 安全性,如果存在一个模拟器(Setup′,KeyGen′,Enc′),对于任意的敌手 A 和任意的安全参数 λ,存在可忽略函数 ε,使得

$$Adv^{yy-SIM}_{FE,A}(\lambda)=|\Pr[yy-REAL^{FE}(1^{\lambda},A)=1]-\Pr[yy-IDEAL^{FE}(1^{\lambda},A)=1]|<\varepsilon(\lambda)。$$

SIM 安全性的精髓在于,模拟器可以模拟敌手的"视图",它既不需要密文也不需要密钥,只需要底层明文相应函数的输出,因此,SIM 安全性更容易捕获函数加密最基本的安全性要求。Boneh 等[1]证明了具有公开索引谓词加密(函数加密的特例)的 IND 安全性与随机预言机(random oracle)模型中的 SIM 安全性等价。

(2)函数隐藏性

函数隐藏安全性定义与消息隐藏的 $yy-IND$ 安全性定义很相似,唯一不同的是,在实验 $IND^{FH-FE}_{\beta}(1^{\lambda},A)$ 中,敌手在密钥询问时,向密钥生成预言机提交一对向量$(\vec{y}^0,\vec{y}^1)$,预言机返回 $\mathrm{KeyGen}(msk,\vec{y}^{\beta})$($\beta\leftarrow\{0,1\}$)。另外,所提交的向量需满足$\langle\vec{x}^0,\vec{y}^0\rangle=\langle\vec{x}^1,\vec{y}^1\rangle$。

强函数隐藏性:一个单输入内积函数加密方案实现了(基于不可区分的)强函数隐藏安全性,如果对于任意的 PPT 敌手 A,对于任意的安全参数 λ,存在可忽略函数 ε,使得

$$Adv^{IND}_{FH-FE,A}(\lambda)=|\Pr[IND^{FH-FE}_{0}(1^{\lambda},A)=1]-\Pr[IND^{FH-FE}_{1}(1^{\lambda},A)=1]|<\varepsilon(\lambda)。$$

弱函数隐藏性:单输入内积函数加密方案的(基于不可区分的)弱函数隐藏安全性的定义与强函数隐藏安全性的定义类似,唯一的区别在于敌手所提交的向量需满足$\langle\vec{x}^0,\vec{y}^0\rangle=\langle\vec{x}^0,\vec{y}^1\rangle=\langle\vec{x}^1,\vec{y}^0\rangle=\langle\vec{x}^1,\vec{y}^1\rangle$。

1.3 多输入内积函数加密的定义

1.3.1 多输入内积函数加密的形式化定义

多输入函数加密考虑了所支持函数接受多个输入且每个输入对应不同密文的情况,其中,第 i 个输入所对应的密文 Ct_i 通常被认为对应于位置或加密槽 i。多输入内积函数加密是单输入内积函数加密的多用户扩展,支持多输入内积功能。

针对多输入内积函数簇 F^m_n(其中,$f_{\{\vec{y}_i\}_{i\in[n]}}\in F^m_n:(\mathbb{Z}^m)^n\to\mathbb{Z}$)的私钥多输入内积函数加密方案包含以下 4 个算法:

- $\mathrm{Setup}(1^{\lambda},m,n)\to(mpk,msk)$,
- $\mathrm{KeyGen}(msk,\vec{y}=(\vec{y}_1,\cdots,\vec{y}_n))\to sk_{\vec{y}}$,
- $\mathrm{Enc}(msk,\vec{x}_i)\to Ct_i$,
- $\mathrm{Dec}(sk_{\vec{y}},Ct_1,\cdots,Ct_n)\to\sum_{i=1}^{n}\langle\vec{x}_i,\vec{y}_i\rangle$ 或 $\perp$。

正确性:针对多输入内积函数簇 F^m_n(其中,$f_{\{\vec{y}_i\}_{i\in[n]}}\in F^m_n:(\mathbb{Z}^m)^n\to\mathbb{Z}$)的私钥多输入内积函数加密方案是正确的,如果对于任意的安全参数 λ,存在可忽略函数 ε,使得

$$\Pr\left[\begin{array}{l}\mathrm{Dec}(sk_{\vec{y}},Ct_1,\cdots,Ct_n)\to\sum_{i=1}^{n}\langle\vec{x}_i,\vec{y}_i\rangle\\ (mpk,msk)\leftarrow\mathrm{Setup}(1^{\lambda},m,n),\\ sk_{\vec{y}}\leftarrow\mathrm{KeyGen}(msk,\vec{y}),\\ \forall i\in[n],Ct_i\leftarrow\mathrm{Enc}(msk,\vec{x}_i)\end{array}\right|<\varepsilon(\lambda)。$$

在公钥多输入内积函数加密中,加密算法被替换为 $\mathrm{Enc}(mpk,\vec{x}_i)\to Ct_i$,其他算法保持不变。

1.3.2 多输入内积函数加密的安全性

(1)消息隐藏性

针对私钥多输入内积函数加密,这里给出消息隐藏性的 8 种安全性定义 $xx-yy-zz$,其中,$xx\in\{one,many\}$ 表示加密槽 i 上挑战密文的个数是一个或多个;$yy\in\{SEL,AD\}$ 表示选择性或自适应;$zz\in\{IND,SIM\}$ 是指不可区分性安全性或基于模拟的安全性。这里有以下的简单关系(⇐表示后者的安全性比前者强):$one\Leftarrow many$,$SEL\Leftarrow AD$,以及以下的标准关系:$IND\Leftarrow SIM$,$many-yy-IND\Leftarrow one-yy-IND$(仅在公钥设置中)。

私钥多输入内积函数加密的 $xx-yy-zz$ 安全性定义与 1.2.2 节类似,这里只描述对应的实验,见表 2。在实验 $xx-yy-IND^{MIFE}_{\beta}(1^{\lambda},A)$ 中,敌手 A 针对第 i($i\in[n]$)个加密槽在第 j_i($j_i\in[Q_i]$)次挑战时提交一对向量$(\vec{x}_i^{(j_i,0)},\vec{x}_i^{(j_i,1)})$,其中,$Q_i$ 表示敌手 A 在第 i 个加密槽上挑战的次数。另外,敌手所提交的向量需满足以下限制:

$$\forall i\in[n],j_i\in[Q_i],\text{有}\ \sum_{i=1}^{n}\langle\vec{x}_i^{(j_i,0)},\vec{y}_i\rangle=\sum_{i=1}^{n}\langle\vec{x}_i^{(j_i,1)},\vec{y}_i\rangle。$$

在理想实验 $xx-yy-IDEAL^{MIFE}(1^{\lambda},A)$ 中,对于询问 $\vec{y}=(\vec{y}_1,\cdots,\vec{y}_n)$,预言机 $O(\cdot)$ 返回 $\mathrm{KeyGen}'(msk',\vec{y},\sum_{i=1}^{n}\langle\vec{x}_i^{j_i},\vec{y}_i\rangle)$。当 $xx=one$ 时,要求敌手 A 在每个加密槽上只发送一个挑战,即对于所有 $i\in[n]$,$Q_i=1$。

表 2　$xx-yy-zz$ 安全性定义中的实验描述

Table 2　Experiments in the security definition of $xx-yy-zz$

$xx-yy-IND_{\beta}^{MIFE}(1^{\lambda},A)$:	$xx-yy-REAL^{MIFE}(1^{\lambda},A)$:	$xx-yy-IDEAL^{MIFE}(1^{\lambda},A)$:
当 $yy=SEL$ 时,	当 $yy=SEL$ 时,	当 $yy=SEL$ 时,
$\{(\vec{x}_i^{(j_i,0)},\vec{x}_i^{(j_i,1)})\}_{i\in[n],j_i\in[Q_i]}\leftarrow A(1^{\lambda},F_n^m)$	$\{\vec{x}_i^{j_i}\}_{i\in[n],j_i\in[Q_i]}\leftarrow A(1^{\lambda},F_n^m)$	$\{\vec{x}_i^{j_i}\}_{i\in[n],j_i\in[Q_i]}\leftarrow A(1^{\lambda},F_n^m)$
$(mpk,msk)\leftarrow\text{Setup}(1^{\lambda},m,n)$	$(mpk,msk)\leftarrow\text{Setup}(1^{\lambda},m,n)$	$(mpk',msk')\leftarrow\text{Setup}'(1^{\lambda},m,n)$
$\beta\leftarrow\{0,1\}$	当 $yy=AD$ 时,	当 $yy=AD$ 时,
当 $yy=AD$ 时,	$\{\vec{x}_i^{j_i}\}_{i\in[n],j_i\in[Q_i]}\leftarrow A^{QKeyG(\cdot)}(1^{\lambda},F_n^m,$	$\{\vec{x}_i^{j_i}\}_{i\in[n],j_i\in[Q_i]}\leftarrow A^{O(\cdot)}(1^{\lambda},F_n^m,mpk')$
$\{(\vec{x}_i^{(j_i,0)},\vec{x}_i^{(j_i,1)})\}_{i\in[n],j_i\in[Q_i]}\leftarrow A^{QKeyG(\cdot)}$	$mpk)$	$\forall i\in[n],j_i\in[Q_i]$,
$(1^{\lambda},F_n^m,mpk)$	$\forall i\in[n],j_i\in[Q_i]$,	$Ct_i^{j_i}\leftarrow\text{Enc}'(msk',i,j_i)$
$\forall i\in[n],j_i\in[Q_i]$, $Ct_i^{j_i}\leftarrow\text{Enc}(msk,$	$Ct_i^{j_i}\leftarrow\text{Enc}(msk,\vec{x}_i^{j_i})$	$\alpha\leftarrow A^{O(\cdot)}(mpk',\{Ct_i^{j_i}\}_{i\in[n],j_i\in[Q_i]})$
$\vec{x}_i^{(j_i,\beta)})$	$\alpha\leftarrow A^{QKeyG(\cdot)}(mpk,\{Ct_i^{j_i}\}_{i\in[n],j_i\in[Q_i]})$	输出:α
$\beta'\leftarrow A^{QKeyG(\cdot)}(mpk,\{Ct_i^{j_i}\}_{i\in[n],j_i\in[Q_i]})$	输出:α	
输出:β'		

(2)函数隐藏性

同样地,私钥多输入内积函数加密的函数隐藏安全性定义与其消息隐藏的 $xx-yy-IND$ 安全性定义类似,所不同的是,在实验 $IND_{\beta}^{FH-MIFE}(1^{\lambda},A)$ 中,敌手在第 $l\in[q_k]$ 次密钥询问时,向密钥生成预言机提交一对向量 $(\vec{y}^{(l,0)},\vec{y}^{(l,1)})$,其中,$\vec{y}^{(l,\beta)}=(\vec{y}_1^{(l,\beta)},\cdots,\vec{y}_n^{(l,\beta)})(\beta\in\{0,1\})$,预言机返回 $\text{KeyGen}(msk,\vec{y}^{(l,\beta)})$。另外,所提交的向量需满足以下限制:

$\forall l\in[q_k],i\in[n],j_i\in[Q_i]$,有

$$\sum_{i=1}^{n}\langle\vec{x}_i^{(j_i,0)},\vec{y}_i^{(l,0)}\rangle=\sum_{i=1}^{n}\langle\vec{x}_i^{(j_i,1)},\vec{y}_i^{(l,1)}\rangle$$

或

$$\sum_{i=1}^{n}\langle\vec{x}_i^{(j_i,0)},\vec{y}_i^{(l,0)}\rangle=\sum_{i=1}^{n}\langle\vec{x}_i^{(j_i,0)},\vec{y}_i^{(l,1)}\rangle=\sum_{i=1}^{n}\langle\vec{x}_i^{(j_i,1)},\vec{y}_i^{(l,0)}\rangle=\sum_{i=1}^{n}\langle\vec{x}_i^{(j_i,1)},\vec{y}_i^{(l,1)}\rangle,$$

前者对应私钥多输入内积函数加密的强函数隐藏性,后者对应私钥多输入内积函数加密的弱函数隐藏性。

1.4　多客户端内积函数加密的定义

1.4.1　多客户端内积函数加密的形式化定义

多客户端内积函数加密同样支持多输入内积功能,它是单输入内积函数加密更自然的扩展,它不仅考虑数据来自不同数据源/客户端的情况,还考虑了这些数据源/客户端可能彼此不信任且能被敌手自适应腐化的问题。

针对多输入内积函数簇 F_n^m(其中,$f_{\{\vec{y}_i\}_{i\in[n]}}\in F_n^m:(\mathbb{Z}^m)^n\rightarrow\mathbb{Z}$)的多客户端内积函数加密方案包含以下 4 个算法:

- $\text{Setup}(1^{\lambda},m,n)\rightarrow(mpk,msk,\{sk_i\}_{i\in[n]})$,
- $\text{KeyGen}(msk,\vec{y})\rightarrow dk_{\vec{y}}$,
- $\text{Enc}(sk_i,\vec{x}_i,t)\rightarrow Ct_{i,t}$,
- $\text{Dec}(dk_{\vec{y}},Ct_{1,t},\cdots,Ct_{n,t})\rightarrow\sum_{i=1}^{n}\langle\vec{x}_i,\vec{y}_i\rangle$ 或 $\perp$。

正确性:针对多输入内积函数簇 F_n^m(其中,$f_{\{\vec{y}_i\}_{i\in[n]}}\in F_n^m:(\mathbb{Z}^m)^n\rightarrow\mathbb{Z}$)的多客户端内积函数加密方案是正确的,如果对于任意的安全参数 λ,任意的标签 t,存在可忽略函数 ε,使得

$$\Pr\left[\text{Dec}(dk_{\vec{y}},Ct_{1,t},\cdots,Ct_{n,t})\rightarrow\sum_{i=1}^{n}\langle\vec{x}_i,\vec{y}_i\rangle\ \middle|\ \begin{array}{l}(mpk,msk,\{sk_i\}_{i\in[n]})\leftarrow\text{Setup}(1^{\lambda},m,n),\\ dk_{\vec{y}}\leftarrow\text{KeyGen}(msk,\vec{y}),\\ \forall i\in[n],Ct_{i,t}\leftarrow\text{Enc}(sk_i,\vec{x}_i,t)\end{array}\right]\langle\varepsilon(\lambda)。$$

多客户端内积函数加密往往含有标签检查机制,即只有 n 个密文对应相同的标签 t,才能正确解密,该机制可以抵抗混合匹配攻击,减少信息泄漏量。

1.4.2　多客户端内积函数加密的安全性

多客户端内积函数加密的安全模型与多输入内积函数加密的安全模型非常相似,唯一的区别在于多客户端内积函数加密必须考虑腐化,因为发送方相互不信任,所以它们可以合谋并将密钥交给敌手。这里给出多客户端内积函数加密消息隐藏性的 4 个安全性定义 $xxx-yyy-IND$,其中,$xxx\in\{sta,adt\}$ 表示静态腐化或自适应腐化;$yyy\in\{any,pos^{+}\}$ 与敌手对每个加密槽发起的挑战次数相关。

多客户端内积函数加密方案的 $xxx-yyy-IND$ 安全

性可以通过表3中的实验 $xxx-yyy-IND_{\beta}^{MCFE}(1^{\lambda},A)$ 来描述。

表3 $xxx-yyy-IND$ 安全性定义中的实验描述

Table 3 Experiment in the security definition of $xxx-yyy-IND$

$xxx-yyy-IND_{\beta}^{MCFE}(1^{\lambda},A)$:

$(mpk,msk,\{sk_i\}_{i\in[n]})\leftarrow\mathrm{Setup}(1^{\lambda},m,n)$

$\beta\leftarrow\{0,1\}$

$\forall i\in[n],t\in L,Ct_{i,t}=\mathrm{Enc}(sk_i,\vec{x}_i,t)\leftarrow Q\mathrm{Enc}(i,\vec{x}_i,t)$

$\forall i\in[n],t\in L,\widehat{Ct}_{i,t}=\mathrm{Enc}(sk_i,\vec{x}_i^{\beta},t)\leftarrow QLR(i,\vec{x}_i^{0},\vec{x}_i^{1},t)$

$\beta'\leftarrow A^{QKeyG(\cdot)}(mpk,\{Ct_{i,t}\}_{i\in[n],t\in L},\{\widehat{Ct}_{i,t}\}_{i\in[n],t\in L},\{sk_i\}_{i\in CS})$

输出:β'

在表3中,L 是标签空间,$QEnc$ 是加密预言机,QLR 是左右预言机,CS 代表腐化用户集合。当 $xxx=sta$ 时,要求敌手 A 在游戏开始前确定腐化用户集合 CS;当 $xxx=adt$ 时,敌手 A 可以自适应地进行腐化询问。另外,敌手所发起的所有询问需同时满足以下条件:

(1) 如果对腐化用户 $i\in CS$ 发起了左右询问 $QLR(i,\vec{x}_i^{0},\vec{x}_i^{1},t)$,则要求 $\vec{x}_i^{0}=\vec{x}_i^{1}$。

(2)对于腐化用户 $i\in CS$ 和询问过 $QEnc(i,\vec{x}_i,t)$ 的用户 i,定义 $\vec{x}_i^{0}:=\vec{x}_i,\vec{x}_i^{1}:=\vec{x}_i$,对于任意的密钥询问,要求 $\sum_{i=1}^{n}[\vec{x}_i^{0},\vec{y}_i]=\sum_{i=1}^{n}\langle\vec{x}_i^{1},\vec{y}_i\rangle$。

(3)当 $yyy=pos^{+}$ 时,对于任意的标签 t,敌手要么不询问左右预言机 QLR,要么对每个诚实用户 $i\in[n]\setminus CS$ 至少询问左右预言机 QLR 一次。

图2总结了多客户端内积函数加密6个安全性定义之间的关系(⇐表示后者的安全性比前者强),其中,$xxx-pos-IND$ 安全性取自文献[10],它与 $xxx-pos^{+}-IND$ 安全性的唯一区别在于其实验中不存在加密预言机 $QEnc$。

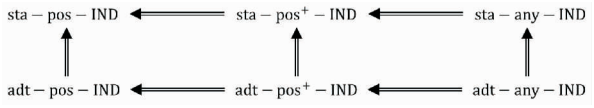

图2 多客户端函数加密安全性定义之间的关系

Fig. 2 Relationship between security of multi-client functional encryption

1.5 去中心化多客户端内积函数加密的定义

去中心化多客户端内积函数加密的定义与多客户端内积函数加密的定义很相似,区别在于:

(1)密钥生成算法 $\mathrm{KeyGen}(msk,\vec{y})\rightarrow dk_{\vec{y}}$ 被拆分为两部分:密钥份额生成算法和密钥组合算法。密钥份额生成算法 $\mathrm{DKeyGenShare}(sk_i,\vec{y})\rightarrow dk_{\vec{y},i}$ 输入密钥 sk_i 和函数向量 $\vec{y}$,输出部分解密密钥 $dk_{\vec{y},i}$。密钥组合算法 $\mathrm{DKeyComb}(\{dk_{\vec{y},i}\}_{i\in[n]},\vec{y})\rightarrow dk_{\vec{y}}$ 输入 n 个部分解密密钥 $\{dk_{\vec{y},i}\}_{i\in[n]}$ 和函数向量 $\vec{y}$,输出解密密钥 $dk_{\vec{y}}$。

(2)在去中心化多客户端内积函数加密的安全性定义中,不存在主私钥 msk,密钥询问被替换为:敌手 A 自适应地提交向量 $\vec{y}=(\vec{y}_1,\cdots,\vec{y}_n)$,挑战者运行 n 次密钥份额生成算法 $\mathrm{DKeyGenShare}(sk_i,\vec{y})\rightarrow dk_{\vec{y},i}$ 生成部分解密密钥 $\{dk_{\vec{y},i}\}_{i\in[n]}$ 并将其返回给敌手 A。

2 内积函数加密技术

2.1 公钥内积函数加密技术

2.1.1 Abdalla 等的内积函数加密方案

在 PKC 2015 上,Abdalla 等[9]基于判定性 Diffie-Hellman(Decision Diffie-Hellman,DDH)假设提出了内积函数加密方案。具体方案如下:

- 参数设置:生成一个阶为素数 p 的循环群 $\mathbb{G}$,g 是该群的生成元。选择向量 $\vec{s}=(s_1,\cdots,s_m)\leftarrow\mathbb{Z}_p^m$,对于 $i\in[m]$,计算 $h_i=g^{s_i}$。设置主公钥为 $mpk=(h_1,\cdots,h_m)$,主私钥为 $msk=\vec{s}$。
- 加密:对于消息 $\vec{x}=(x_1,\cdots,x_m)\leftarrow\mathbb{Z}_p^m$,选择随机数 $r\leftarrow\mathbb{Z}_p$,计算 $ct_0=g^r$,对于 $i\in[m]$,计算 $ct_i=h_i^r\cdot g^{x_i}$,得到密文 $Ct=(ct_0,ct_1,\cdots,ct_m)$。
- 密钥生成:对于向量 $\vec{y}=(y_1,\cdots,y_m)\in\mathbb{Z}_p^m$,计算解密密钥为 $sk_{\vec{y}}=\langle\vec{y},\vec{s}\rangle$。
- 解密:计算 $g^{\langle\vec{x},\vec{y}\rangle}=\prod_{i\in[m]}ct_i^{y_i}/ct_0^{sk_{\vec{y}}}$,然后对 $g^{\langle\vec{x},\vec{y}\rangle}$ 计算离散对数得到 $\langle\vec{x},\vec{y}\rangle$。

该方案被证明是选择性不可区分(SEL-IND)安全的。值得注意的是,该方案对所支持的消息空间施加了严格的限制,以保证 $\langle\vec{x},\vec{y}\rangle$ 的值较小,从而可以有效求解离散对数。另外,对该方案进行推广,得到通用方案,并给出了通用方案在容错学习(Learning-with-Errors,LWE)假设下的实例化。尽管该方案非常简单,安全性不强,但其思想仍被后续很多工作采用。可以说,Abdalla 等[9]的工作开创了内积函数加密研究的先河。在此之后,出现了关于内积函数加密研究的两大主流:一个是公钥内积函数加密,目的是获得更强的消息机密性;另一个是私钥内积函数加密,旨在进行多用户扩展、获得函数隐藏性和更高的效率。

2.1.2　其他的公钥内积函数加密方案

选择性安全性是一个比较弱的概念,通常被视为证明自适应安全性的基础。Boneh 等[11]观察到,复杂性泄露论证技术(complexity leveraging argument)可以用来证明一个选择性安全的系统也是自适应安全的。然而,这种说法一般不尽人意,因为安全性归约会产生消息空间大小的指数级安全损失。为此,Agrawal 等[12]分别基于 DDH 假设、LWE 假设和判定性合数剩余(Decision Composite Residuosity,DCR)假设提出了自适应(AD-IND)安全的内积函数加密方案。其中,基于 DDH 假设和基于 LWE 假设的整数内积函数加密方案的效率可与 Abdalla 等[9]的方案相媲美。此外,他们分别基于 DCR 假设和 LWE 假设构造了模 p 和模 N($N=p\cdot q$,其中,p 和 q 均为素数)意义下的内积函数加密方案,而 Abdalla 等的方案仅限于求解短整数向量的整数内积。之后,Castagnos 等[13]也提出了自适应(AD-IND)安全的内积函数加密方案,该方案允许计算在模 p 意义下任意大小的内积。方案依赖于循环群中新的密码学假设,该循环群包含一个容易求解离散对数(Discrete Logarithm,DL)问题的子群,而内积被编码在该子群的指数中,因此,无论内积值的大小如何,都可以有效地恢复出来。为了追求更强的安全性,Agrawal 等[14]又分别基于 DDH、DCR 和 LWE 假设提出了自适应模拟(AD-SIM)安全的内积函数加密方案。

Abdalla 等[9]的方案是有界方案,即方案可处理的向量长度是在设置阶段固定下来的,这使得方案不具有灵活性。为此,Tomida 等[15]首次提出了无界内积函数加密方案,该方案可以处理无界多项式长度的向量,且在标准模型中被证明是自适应(AD-IND)安全的。之后,Dufour-Sans 等[16]在随机预言机模型中基于标准假设也提出了无界内积函数加密方案,该方案具有固定大小的公钥和主密钥,并且每个解密密钥仅由一个群元素组成。然而,该方案只能达到选择性(SEL-IND)安全。

评估攻破方案的难易程度与解决问题的难易程度具有重要的理论意义和实践意义。更正式地说,当归约算法中敌手在时间 t 内以概率 ε 攻破方案和在时间 t 内以概率 ε/L 解决底层问题时,评估安全损失 L 是很重要的。因为方案的参数需要设置得足够大,以抵消 L 对安全性的影响。安全损失 L 越小,安全性归约越理想。当安全损失 L 为常数,即 $L=O(1)$ 时,安全归约被称为紧归约。Tomida[17]将紧安全密码学扩展到内积函数加密领域,提出了紧安全的内积函数加密方案,该方案基于矩阵判定 Diffie-Hellman(Matrix Decisional Diffie-Hellman,MDDH)假设是 AD-IND 安全的。

鉴于传统的内积函数加密方案仅能达到选择明文攻击(CPA)安全,Benhamouda 等[18]通过具有同态性质的投影哈希函数(projective Hash functions)提出一种选择密文攻击(CCA)安全的内积函数加密通用构造,并给出了基于 DCR 假设、DDH 假设以及 MDDH 假设的实例化。然而,该方案的安全性归约具有指数级损失。为此,Castagnos 等[19]提出一种紧 CCA 安全的内积函数加密方案,方案的安全性证明放松了对底层投影哈希函数的要求从而获得了紧归约性。同时,Liu 等[20]基于 MDDH 假设也构建了紧 CCA 安全的内积函数加密方案,该方案的主公钥和密文大小与 Tomida[17]的紧 CPA 安全的内积函数加密方案相当。

针对内积函数加密方案的性能问题,Kim 等[21]提出了一种高效的基于双线性对的内积函数加密方案,它只需要 m(消息向量的长度)次幂运算和两次双线性对运算,就可以用更小的公共参数、密钥和密文进行解密。虽然该方案可以完全抗合谋(攻击者即使获得多项式个密钥,也无法创建新的有效密钥),但仅能达到选择性(SEL-IND)安全。由于基于环容错学习(Ring Learning with Errors,RLWE)假设所构造的方案要比基于 LWE 假设所构造的方案效率高且密钥短,Mera 等[22]基于 RLWE 假设提出了两个内积函数加密方案,这两个方案分别达到了选择性(SEL-IND)安全和适应性(AD-IND)安全。

传统的内积函数加密允许多人访问相同的函数,这其中可能存在恶意用户,而解密密钥是由函数和主密钥生成的,与用户的身份无关,如果这些用户中的一个(称为叛徒)泄露解密密钥,那么就没有办法追溯其身份。为此,Do 等[23]引入了可追踪的函数加密原语,使得解密密钥不仅对应于某个函数,而且也对应于某个用户。通过结合 Abdalla 等[9]的内积函数加密方案与可追踪的 Boneh-Franklin 方案[24],构造了一个具有黑盒确认的可追踪内积函数加密方案,该方案是选择性(SEL-IND)安全的且支持单目标黑盒可追溯性。作为叛徒跟踪系统的扩展,跟踪 - 撤销系统存在一个额外的用户撤销机制,使得内容分发者可以使用该机制禁用被破坏的密钥解密功能。Luo 等[25]研究了函数加密的追踪 - 撤销机制,并从标准假设中提出了追踪 - 撤销内积函数加密方案。该方案支持公开的黑盒可追溯性,在标准模型中具有自适应(AD-IND)安全性。

在传统密码学模型中,安全性通常依赖于密钥和随机数等秘密值的完全隐私性。对于这种模型下的许多加密系统来说,如果这些秘密值被泄露了一个比特,安全性就完全丧失了。抗泄漏密码学可以在敌手获得秘

密值的一些信息时依然提供正式的安全保证。为此,Zhang 等[26]在有界检索模型(bounded-retrieval model)中提出了抗泄露的内积函数加密方案,该方案仅通过增加密钥的大小就可以灵活地容忍任意泄漏边界。

在许多应用场景中,数据所有者总是不同于函数所有者,因此,函数加密的经典实现自然意味着在拥有函数f的实体和管理主密钥的实体之间存在交互式密钥生成协议。针对这个特定的阶段,考虑到函数需要保密的情况,Canard 等[27]引入了盲函数加密的原语,使用同态加密和零知识证明得到了非盲函数加密到盲函数加密的转换方法,并针对内积函数给出了一个有效的实例。

研究具有更复杂功能的内积函数加密原语可以限制传统内积函数加密方案固有的信息泄漏。Dufour-Sans 等[16]提出了一种基于身份的内积函数加密方案,该方案允许用户在自己的密文中指定一个身份。然而,该方案仅在随机预言机模型中达到选择性(SEL-IND)安全。之后,Abdalla 等[28]将属性基加密(attribute-based encryption)与内积函数加密相结合,构造了支持细粒度访问控制的内积函数加密方案。另外,他们还基于 LWE 假设构造了身份基内积函数加密方案,与 Dufour-Sans 等[16]的方案相比,该方案所支持的功能更丰富,并且在标准模型中被证明是 SEL-IND 安全的。为了方便密钥的管理,Song 等[29]提出了基于层次身份的内积函数加密,它具有委托功能(因为接收者的身份具有层次结构,上层用户可以生成下层用户的私钥),基于层次身份的内积函数加密方案在标准模型中是选择性(SEL-IND)安全的。

表4对比分析了上述典型公钥内积函数加密方案的主公钥大小、密文大小、解密密钥大小、加解密开销及安全性。这里主要考虑计算开销比较大的双线性对运算(P)、模指数运算(E)和模逆运算(I)。$|\mathbb{G}|$、$|\mathbb{G}_1|$、$|\mathbb{G}_2|$和$|\mathbb{Z}_p|$分别表示群$\mathbb{G}$、$\mathbb{G}_1$、$\mathbb{G}_2$和$\mathbb{Z}_p$中单个元素的大小。m代表向量的长度,k与方案所基于的困难性假设中矩阵的大小有关。

表4 典型公钥内积函数加密方案的比较

Table 4 Comparison of typical public-key inner product functional encryption schemes

方案	$\lvert mpk\rvert$	$\lvert Ct\rvert$	$\lvert sk_{\hat{y}}\rvert$	加密开销	解密开销	困难性假设	安全性
文献[9]	$m\lvert\mathbb{G}\rvert$	$(m+1)\lvert\mathbb{G}\rvert$	$\lvert\mathbb{Z}_p\rvert$	$(2m+1)E$	$(m+1)E+1I$	DDH	SEL-IND
文献[12]	$m\lvert\mathbb{G}\rvert$	$(m+2)\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{Z}_p\rvert$	$(2m+2)E$	$(m+2)E+1I$	DDH	AD-IND
文献[13]	$(m+1)\lvert\mathbb{G}\rvert$	$(m+2)\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{Z}_p\rvert$	$(2m+2)E$	$(m+2)E+1I$	DDH-f	AD-IND
文献[14]	$m\lvert\mathbb{G}\rvert$	$(m+2)\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{Z}_p\rvert$	$(2m+2)E$	$(m+1)E+1I$	DDH	AD-SIM
文献[15]	$28\lvert\mathbb{G}\rvert$	$7m\lvert\mathbb{G}_1\rvert$	$7m\lvert\mathbb{G}_2\rvert$	$7mE$	$7mP$	SXDH	AD-IND
文献[17]	$(k^2m^2+k^2+k)\lvert\mathbb{G}\rvert$	$(k^2+k+1)m\lvert\mathbb{G}\rvert$	$(k^2+k)m\lvert\mathbb{Z}_p\rvert$	$(k^2+k+1)mE$	$(k^2+k+1)mE$	D_k-MDDH	AD-IND

从表4可以看出,大部分方案的主公钥及密文大小、加解密开销,都和向量长度m直接相关,随着向量长度的增加呈线性增长的趋势。在文献[15]的方案中,虽然主公钥的大小和向量长度无关,但其解密密钥的大小仍然和向量长度m线性相关,而其他方案解密密钥的大小大部分都和向量长度无关。另外,大部分方案都实现了 AD-IND 安全性,而文献[9]的方案仅实现了 SEL-IND 安全性,文献[14]的方案实现了 AD-SIM 安全性。

2.2 私钥内积函数加密技术

2.2.1 函数隐藏的内积函数加密方案

当函数本身包含敏感信息时,内积函数加密还需要保证函数的隐私性。然而,在公钥设置和私钥设置之间,实现函数隐藏性的程度有很大不同。具体地说,在公钥设置中,只能获得有限形式的函数隐藏性,为了制定有意义的安全性定义,必须假设函数来自某个具有足够熵的分布[30-31]。相反,函数隐藏性在私钥设置中无论是作为一个独立的特性,还是作为一个非常有用的构建块,都比在公钥设置中具有更大的潜力[12]。

在 ASIACRYPT 2015 上,Bishop 等[32]提出了函数隐藏的内积函数加密方案,尽管该方案仅能实现弱函数隐藏性。为了实现强函数隐藏性,在此基础上,Datta 等[33]和 Tomida 等[34]利用对偶对向量空间(dual pairing vector spaces)分别基于外判定线性假设(eXternal Decisional Linear Assumption, XDLIN)和对称外 Diffie-Hellman(Symmetric eXternal Diffie-Hellman, SXDH)假设提出了强函数隐藏的内积函数加密方案。之后,Kim 等[35]又将密钥和密文的大小减小了一半,但所构造的内积函数加密方案只在通用群(generic group)模型中是 AD-SIM 安全的。

Lin 等[36]也注意到了 Bishop 方案[32]的安全性缺陷,提出了一种由弱函数隐藏的内积函数加密到强函数隐藏的内积函数加密的转换方法。然而,转换后的方案所支持的向量长度仅为原方案的一半。之后,Lin[37]又利用双层加密的思想提出了非函数隐藏的内积函数加

密方案到弱函数隐藏的内积函数加密方案的转换方法。Kim 等[38]利用双重加密的思想提出了非函数隐藏的函数加密到强函数隐藏的函数加密的转换方法。这与 Lin 的双层加密思想很相似,所不同的是,Lin 使用的是同一个函数加密方案的两个实例,而这里使用的是两个不同功能的函数加密。在 Agrawal 等[12]基于 DDH 假设的内积函数加密方案上使用上述转换方法,Kim 等得到了强函数隐藏的内积函数加密方案。

Tomida 等[15]提出了无界强函数隐藏的内积函数加密方案,该方案可以处理无界多项式长度的向量,在标准模型中基于标准假设被证明是自适应(AD-IND)安全的。Tomida[17]将 Lin[37]的转换和 Abdalla 等[39]的转换应用到所设计的紧安全内积函数加密方案中,分别得到了第一个紧安全函数隐藏的内积函数加密方案和紧安全的多输入内积函数加密方案。另外,他还提出了从函数隐藏的内积函数加密到函数隐藏的多输入内积函数加密的通用转换。

Liu 等[20]在所设计的紧 CCA 安全的内积函数加密方案基础上,构造了函数隐藏的内积函数加密方案,该方案具有紧 CCA 安全性。

表 5 对比分析了上述典型函数隐藏的内积函数加密方案的主私钥、密文、解密密钥大小、加解密和密钥生成开销及安全性。这里主要考虑运算开销比较大的双线性对运算(P)和模指数运算(E)。$|\mathbb{G}_1|$、$|\mathbb{G}_2|$和$|\mathbb{Z}_p|$分别表示群$\mathbb{G}_1$、$\mathbb{G}_2$和$\mathbb{Z}_p$中单个元素的大小,m 代表向量的长度。

表 5　典型函数隐藏的内积函数加密方案的比较

Table 5　Comparison of typical function-hiding inner product functional encryption schemes

方 案	$\lvert msk\rvert$	$\lvert Ct\rvert$	$\lvert sk_{\vec{y}}\rvert$	加密/密钥生成开销	解密开销	困难性假设	安全性
文献[32]	$(8m^2+8)\lvert\mathbb{Z}_p\rvert$	$(m+2)\lvert\mathbb{G}_1\rvert$	$(m+2)\lvert\mathbb{G}_2\rvert$	$(m+2)E$	$(m+2)P$	SXDH	弱函数隐藏性
文献[33]	$(8m^2+12m+28)\lvert\mathbb{Z}_p\rvert$	$(4m+8)\lvert\mathbb{G}_1\rvert$	$(4m+8)\lvert\mathbb{G}_2\rvert$	$(4m+8)E$	$(4m+8)P$	SXDH	强函数隐藏性
文献[34]	$(4m^2+18m+20)\lvert\mathbb{Z}_p\rvert$	$(2m+5)\lvert\mathbb{G}_1\rvert$	$(2m+5)\lvert\mathbb{G}_2\rvert$	$(2m+5)E$	$(2m+5)P$	XDLIN	强函数隐藏性
文献[15]	$16\lvert\mathbb{Z}_p\rvert$	$4m\lvert\mathbb{G}_1\rvert$	$4m\lvert\mathbb{G}_2\rvert$	$4mE$	$4mP$	SXDH	强函数隐藏性

从表 5 可以看出,大部分方案的主私钥、密文和解密密钥的大小、加解密和密钥生成的开销,都和向量长度 m 直接相关,随着向量长度的增加呈线性增长的趋势。只有在文献[15]的方案中,主私钥的大小和向量长度无关。另外,大部分方案都实现了强函数隐藏安全性,而文献[32]的方案仅实现了弱函数隐藏安全性。

2.2.2　多输入内积函数加密方案

在多输入函数加密中,公钥设置和私钥设置的安全保证也有很大不同[40]。更详细地说,在公钥设置中,解密者可以冒充大部分用户对所选择的数据加密,以确定其他用户的私有输入。由于多输入函数加密在公钥设置中存在这种固有的泄露,对多输入内积函数加密的研究主要集中在私钥设置中。在 EUROCRYPT 2017 上,Abdalla 等[41]在双线性群中基于标准假设提出了多输入内积函数加密方案,该方案适用于任何多项式个数的加密槽,并实现了针对无界共谋的自适应(many-AD-IND)安全性。

基于 Tomida 等[34]函数隐藏的内积函数加密方案,Datta 等[42]在素数阶双线性群中构造了两个函数隐藏的多输入内积函数加密方案。其中,第一个方案仅支持先验固定数量的加密槽,而第二个方案支持无界数量的加密槽。同年,Abdalla 等[39]也考虑了在标准假设下支持多项式个加密槽的多输入内积函数方案的构造问题,提出了基于任何单输入内积函数加密的多输入内积函数加密的通用构造。另外,他们从 Abdalla 等[41]的多输入内积函数加密方案出发,使用单输入内积函数加密作为额外的构建块,构建了函数隐藏的多输入内积函数加密方案。然而,该方案并不是完全通用的,因为它对底层的构建块施加了限制,无法兼容他们在该工作中所构建的不使用双线性对的多输入内积函数加密方案。

2.2.3　(去中心化)多客户端内积函数加密方案

在(去中心化)多客户端函数加密中,每个客户端使用不同的加密密钥,这些密钥是每个客户端私有的,因此,(去中心化)多客户端函数加密也属于私钥函数加密的范畴。

多客户端函数加密和多输入函数加密一样,仍然假设存在一个可信的第三方运行参数设置算法,生成并分发解密密钥。如果这个第三方是恶意的或被腐化的,则任何客户端的隐私都会很容易地被破坏。Chotard 等[43]基于标准假设给出了多客户端内积函数加密的第一个构造,并引入了去中心化多客户端函数加密的原语,该原语要求参数设置算法和密钥生成算法由生成密文的同一组客户端分布式运行。所提出的去中心化多客户端内积函数加密方案在随机预言机模型中基于标准假

设具有 sta-pos$^+$-IND 安全性。针对 Chotard 等[43]的方案需要依赖随机预言机的问题,Libert 等[44]在标准模型中基于 LWE 假设分别设计了多客户端内积函数加密方案和去中心化多客户端内积函数加密方案,然而它仍然受到与 Chotard 方案[43]相同的安全限制。

Abdalla 等[10]为多客户端内积函数加密提供了两个编译器:第一个编译器可以将任何具有特殊密钥生成属性的多客户端内积函数加密方案转换为去中心化多客户端内积函数加密方案;第二个编译器允许解除现有(去中心化)多客户端内积函数加密方案中存在的非自然限制,即该编译器可以将 pos$^+$安全的方案转变为 any 安全的方案。之后,Abdalla 等[45]首次在标准模型中基于内积函数加密和标准伪随机函数提出了一种多客户端内积函数加密的通用构造,并分别给出了基于 DDH、LWE 和 DCR 假设的实例化。

鉴于 Chotard 等[43]的方案密文长度随客户端数量的增加呈二次增长的趋势,而 Abdalla 等[45]的方案仅支持较小的内积空间,Abdalla 等[46]在随机预言机模型中分别提出了基于 MDDH、DCR 及 LWE 假设的多客户端内积函数加密方案,这些方案都具有线性的密文长度。

由于多输入函数加密要求可信第三方的参与且不支持用户的动态加入,Agrawal 等[47]引入了自组织多输入函数加密(Ad Hoc multi-input functional encryption)的原语,即用户可以动态地加入系统,并且解密密钥可以由每个用户以分布式方式生成。他们给出了通用功能的自组织多输入函数加密的可行性结果,并基于 LWE 假设给出了自组织多输入内积函数加密的实用结构。同时,Chotard 等[48]引入了动态去中心化函数加密(dynamic decentralized functional encryption)的原语,它同样允许用户动态地加入系统,而不依赖于可信第三方或昂贵的交互式多方计算协议。另外,所提出的动态去中心化内积函数加密方案在随机预言机模型中基于 DDH 假设是选择性安全的。与自组织多输入函数加密相比,动态去中心化函数加密更灵活,因为其密钥生成算法不需要指定客户端集合。自组织多输入函数加密不能处理标签,这意味着每个客户端单独计算的密文可以任意混合和匹配,这会泄露明文的大量信息。Agrawal 等[49]提出了多方函数加密(multi-party functional encryption)的原语,该原语可以涵盖目前所有的函数加密原语。此外,他们还构造了一个函数隐藏的多客户端内积函数加密方案,实例化该方案得到了函数隐藏的动态去中心化内积函数加密方案。

3 内积函数加密的应用

内积是描述性统计中的重要工具,内积函数加密对涉及隐私保护内积计算的应用具有重要的意义。具体来说,内积函数加密不仅可以作为底层工具构造其他功能的函数加密方案,还可以用于隐私保护的统计分析、外包计算、加密生物特征认证和机器学习等场景中。

内积函数加密可作为底层工具构造其他功能的函数加密方案。Barbosa 等[50]基于已有的内积函数加密方案[35, 37]为正交关系构造了两个函数加密方案。Zhang 等[51]利用内积函数加密方案[9, 12]提出了针对三次多项式的函数加密方案,并进一步得到了针对所有电路的函数加密方案。Agrawal 等[52]证明了在公钥设置下,公钥内积函数加密方案可以很容易地转化为二次函数的多输入函数加密方案。另外,通过为内积函数加密引入了两个新的原语,并构造出针对二次函数的多输入函数加密方案。在文献[53]中,又对方案进行了改进,得到了针对二次函数的多客户端函数加密方案。

内积函数加密用于构造其他加密方案。Liang 等[54]通过巧妙地利用多输入内积函数加密提出了一个安全高效的多关键词可搜索加密方案。其中,多输入内积函数加密的运用使得云服务器能够使用仅包含两个条目的搜索令牌完成搜索过程,从而降低了通信开销。

针对多用户选择性数据共享设置中的内积计算问题,Yang 等[55]在基于 DCR 假设的内积函数加密方案[12]的基础上提出了一个可验证的外包内积计算方案。之后,在文献[56]中,改进了基于 DCR 假设的内积函数加密方案[12],提出了新的外包计算模型,并基于改进的内积函数加密方案设计出外包内积计算方案。

联邦学习之所以受到广泛关注,是因为它能够在不收集用户原始数据的情况下在云服务器上协同训练模型。函数加密作为一种能对加密数据执行数据聚合的强大方法,可以被用于构建隐私保护的联邦学习方案。在联邦学习的隐私保护方面,函数加密要比同态加密更加轻便灵活。Qian 等[57]使用 Abdalla 等[10]提出的编译器得到了一个去中心化多客户端内积函数加密方案,并将该方案作为底层工具构造了一种基于云的隐私保护联邦学习聚合方案。

Kim 等[35]不仅改进了函数隐藏的内积函数加密方案以提高效率,还阐述了函数隐藏的内积函数加密的3个应用场景,包括加密数据上的线性回归、加密数据上

的最近邻检索以及加密生物特征认证。这些工作使得函数隐藏的内积函数加密算法的实际应用成为可能。

4　内积函数加密的相关工作

通用功能的函数加密方案支持任意的多项式时间内可计算的函数,自然也支持内积运算。在函数加密的早期研究中,很多学者致力于提出通用功能的函数加密方案[3-8]。然而,这些方案的安全性往往基于不可区分混淆及其变体或多线性映射,其实用性存疑。

在内积函数加密提出之前,Katz 等[58]提出了内积加密(inner product encryption)的原语,在文献[59]中称为内积谓词加密,在文献[60]中同样称为内积函数加密。需要注意的是,这与本文所阐述的内积函数加密有很大不同:本文所阐述的内积函数加密可以计算出内积的实际值,而 Katz 等所提出的内积加密只测试内积是否为零(即两个向量是否正交)。为了避免混淆,Abdalla 等[9]在提出内积函数加密时,将 Katz 等的内积加密改称为正交性函数加密(orthogonality functional encryption)。

5　总结与展望

虽然从 2015 年开始已经存在大量的内积函数加密的研究,但目前还没有真正投入实际使用的内积函数加密方案,可以说对于内积函数加密的研究仍停留在起步阶段。对内积函数加密技术研究的展望如下:

- 基于模拟安全的内积函数加密方案的构造。基于不可区分性的安全性在某些情况下对于函数加密来说太弱了,基于模拟的安全性保证了除函数值之外任何关于明文的信息都不会被泄露,并且排除了一些被证明 IND 安全而实际不安全的方案[1]。目前已有的内积函数加密方案,尤其是多输入内积函数加密方案[41]和多客户端内积函数加密方案[43-44],除了仅有的几个方案[14,35]是 SIM 安全外,其他方案[9,12-13,15-17,21-23,25,28-29]都只能实现 IND 安全性。因此,如何构造 SIM 安全的内积函数加密方案仍是一个挑战。
- 不使用双线性对的内积函数加密方案的构造。目前已有的函数隐藏的内积函数加密方案基本都要使用双线性对,这会导致在解密过程中不可避免地需要求解离散对数才能恢复出内积值。为了有效地解密,方案往往对所支持的明文空间的大小施加严格的限制[39]。因此,研究不使用双线性对的函数隐藏的内积函数加密是有意义的。
- 探索内积函数加密新的性质。目前已有可追踪的内积函数加密[23,25]、支持细粒度访问控制的内积函数加密[16,28-29]、支持用户动态加入的内积函数加密[47-49]的研究,这些方案通常更接近实际的应用需求。因此,探索内积函数加密新的性质也是一个重要且有趣的研究方向,如可验证的内积函数加密、可否认的内积函数加密等。
- 内积函数加密应用的研究。目前内积函数加密应用的研究涉及其他功能的函数加密[50-53]、可搜索加密[35,54]、加密数据线性回归[35]、加密生物特征认证[35]、外包计算[55-56]、隐私保护机器学习[57]等,然而,每个领域的应用研究都较少。另外,将内积函数加密应用于其他领域也是一个非常有意义的研究方向。

参考文献:

[1] Bone D, Sahai A, Waters B. Proceedings of the 8th Theory of Cryptography Conference, March 28-30, 2011[C]. Berlin: Springer, 2011.

[2] O'neill A. Definitional issues in functional encryption[EB/OL]. (2011-03-19)[2023-01-27]. https://eprint.iacr.org/2010/556.

[3] Garg S, Gentry C, Halevi S, et al. Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, October 27-29, 2013[C]. Los Alamitos: IEEE Computer Society, 2013.

[4] Boyle E, Chung K M, Pass R. Proceedings of the 11th Theory of Cryptography Conference on Theory of Cryptography, February 24-26, 2014[C]. Berlin: Springer, 2014.

[5] Waters B. Proceedings of the 35th Annual Cryptology Conference, August 16-20, 2015[C]. Berlin: Springer, 2015.

[6] Garg S, Gentry C, Halevi S, et al. Fully secure attribute-based encryption from multilinear maps [EB/OL]. (2014-08-13)[2023-01-27]. https://eprint.iacr.org/2014/622.

[7] Garg S, Gentry C, Halevi S, et al. Proceedings of the 13th International Conference on Theory of Cryptography, January 10-13, 2016[C]. Berlin: Springer, 2016.

[8] Brakerski Z, Segev G. Proceedings of the 12th Theory of Cryptography Conference, March 23-25, 2015[C]. Berlin: Springer, 2015.

[9] Abdalla M, Bourse F, de Caro A, et al. Proceedings of the 18th IACR International Conference on Practice and Theory of

Public-key Cryptography, March 30-April 1, 2015[C]. Berlin: Springer, 2015.
[10] Abdalla M, Benhamouda F, Kohlweiss M, et al. Proceedings of the 22nd IACR International Conference on Practice and Theory of Public-key Cryptography, April 14-17, 2019[C]. Berlin: Springer, 2019.
[11] Boneh D, Boyen X. Efficient selective identity-based encryption without random oracles[J]. Journal of Cryptology, 2011, 24(4): 659-693.
[12] Agrawal S, Libert B, Stehle D. Proceedings of the 36th Annual International Cryptology Conference, August 14-18, 2016[C]. Berlin: Springer, 2016.
[13] Castagnos G, Laguillaumie F, Tucker I. Proceedings of the 24th Annual International Conference on Theory and Application of Cryptology and Information Security, December 2-6, 2018[C]. Berlin: Springer, 2018.
[14] Agrawal S, Libert B, Maitra M, et al. Proceedings of the 23rd IACR International Conference on the Practice and Theory of Public-key Cryptography, May 4-7, 2020[C]. Berlin: Springer, 2020.
[15] Tomida J, Takashima K. Proceedings of the 24th Annual International Conference on Theory and Application of Cryptology and Information Security, December 2-6, 2018[C]. Berlin: Springer, 2018.
[16] Dufour-Sans E, Pointcheval D. Proceedings of the 17th International Conference on Applied Cryptography and Network Security, June 5-7, 2019[C]. Berlin: Springer, 2019.
[17] Tomida J. Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptology and Information Security, December 8-12, 2019[C]. Berlin: Springer, 2019.
[18] Benhamouda F, Bourse F, Lipmaa H. Proceedings of the 20th IACR International Conference on Practice and Theory of Public-key Cryptography, March 28-31, 2017[C]. Berlin: Springer, 2017.
[19] Castagnos G, Laguillaumie F, Tucker I. A tighter proof for CCA secure inner product functional encryption: Genericity meets efficiency[J]. Theoretical Computer Science, 2022, 914: 84-113.
[20] Liu X, Liu S, Han S, et al. Tightly CCA-secure inner product functional encryption scheme[J]. Theoretical Computer Science, 2022, 898: 1-19.
[21] Kim I, Park J H, Hwang S O. An efficient public key functional encryption for inner product evaluations[J]. Neural Computing and Applications, 2020, 32(17): 13117-13128.
[22] Mera J M B, Karmakar A, Marc T, et al. Proceedings of the 25th IACR International Conference on Practice and Theory of Public-key Cryptography, March 8-11, 2022[C]. Berlin: Springer, 2022.
[23] Do X T, Phan D H, Pointcheval D. Proceedings of the Cryptographers Track at the RSA Conference, February 24-28, 2020[C]. Berlin: Springer, 2020.
[24] Boneh D, Franklin M. Proceedings of the 19th Annual International Cryptology Conference, August 15-19, 1999[C]. Berlin: Springer, 1999.
[25] Luo F, Al-Kuwari S, Wang H, et al. Proceedings of the 27th European Symposium on Research in Computer Security, September 26-30, 2022[C]. Berlin: Springer, 2022.
[26] Zhang L, Wang X, Chen Y, et al. Proceedings of the 22nd International Conference on Information and Communications Security, August 24-26, 2020[C]. Berlin: Springer, 2020.
[27] Canard S, Hamdi A, Laguillaumie F. Proceedings of the 22nd International Conference on Information and Communications Security, August 24-26, 2020[C]. Berlin: Springer, 2020.
[28] Abdalla M, Catalano D, Gay R, et al. Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security, December 7-11, 2020[C]. Berlin: Springer, 2020.
[29] Song G, Deng Y, Huang Q, et al. Hierarchical identity-based inner product functional encryption[J]. Information Sciences, 2021, 573: 332-344.
[30] Boneh D, Raghunathan A, Segev G. Proceedings of the 33rd Annual International Cryptology Conference, August 18-22, 2013[C]. Berlin: Springer, 2013.
[31] Boneh D, Raghunathan A, Segev G. Proceedings of the 19th International Conference on the Theory and Application of Cryptology and Information Security, December 1-5, 2013[C]. Berlin: Springer, 2013.
[32] Bishop A, Jain A, Kowalczyk L. Proceedings of the 21st International Conference on the Theory and Application of Cryptology and Information Security, November 29-December 3, 2015[C]. Berlin: Springer, 2015.
[33] Datta P, Dutta R, Mukhopadhyay S. Proceedings of the 19th IACR International Conference on Practice and Theory in Public-key Cryptography, March 6-9, 2016[C]. Berlin: Springer, 2016.
[34] Tomida J, Abe M, Okamoto T. Proceedings of the 19th Annual International Conference on Information Security, September 3-6, 2016[C]. Berlin: Springer, 2016.
[35] Kim S, Lewi K, Mandal A, et al. Function-hiding inner product encryption is practical [EB/OL]. (2018-06-13) [2023-

01-27]. https://eprint.iacr.org/2016/440.
[36] Lin H, Vaikuntanathan V. Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science, October 9-11, 2016[C]. Piscataway: IEEE, 2016.
[37] Lin H. Proceedings of the 37th Annual International Cryptology Conference, August 20-24, 2017[C]. Berlin: Springer, 2017.
[38] Kim S, Kim J, Seo J H. A new approach to practical function-private inner product encryption[J]. Theoretical Computer Science, 2019, 783: 22-40.
[39] Abdalla M, Catalano D, Fiore D, et al. Proceedings of the 38th Annual International Cryptology Conference, August 19-23, 2018[C]. Berlin: Springer, 2018.
[40] Goldwasser S, Gordon S D, Goyal V, et al. Proceedings of the 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, May 11-15, 2014[C]. Berlin: Springer, 2014.
[41] Abdalla M, Gay R, Raykova M, et al. Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, April 30 - May 4, 2017[C]. Berlin: Springer, 2017.
[42] Datta P, Okamoto T, Tomid J. Proceedings of the 21st IACR International Conference on Practice and Theory of Public-key Cryptography, March 25-29, 2018[C]. Berlin: Springer, 2018.
[43] Chotard J, Dufour-Sans E, Gay R, et al. Proceedings of the 24th Annual International Conference on Theory and Application of Cryptology and Information Security, December 2-6, 2018[C]. Berlin: Springer, 2018.
[44] Libert B, Liu R. Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptology and Information Security, December 8-12, 2019[C]. Berlin: Springer, 2019.
[45] Abdalla M, Benhamouda F, Gay R. Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptology and Information Security, December 8-12, 2019[C]. Berlin: Springer, 2019.
[46] Abdalla M, Bourse F, Marival H, et al. Proceedings of the 12th International Conference on Security and Cryptography for Networks, September 14-16, 2020[C]. Berlin: Springer, 2020.
[47] Agrawal S, Clear M, Frieder O, et al. Proceedings of the Conference on Innovations in Theoretical Computer Science, January 12-14, 2020[C]. Dagstuhl: Dagstuhl Publishing, 2020.
[48] Chotard J, Dufour-Sans E, Gay R, et al. Proceedings of the 40th Annual International Cryptology Conference, August 17-21, 2020[C]. Berlin: Springer, 2020.
[49] Agrawal S, Goyal R, Tomida J. Proceedings of the 19th International Conference on Theory of Cryptography, November 8-11, 2021[C]. Berlin: Springer, 2021.
[50] Barbosa M, Catalano D, Soleimanian A, et al. Proceedings of the Cryptographers Track at the RSA Conference 2019, March 4-8, 2019[C]. Berlin: Springer, 2019.
[51] Zhang Z, Zhang F. Functional encryption for cubic polynomials and implementation[J]. Theoretical Computer Science, 2021, 885: 41-54.
[52] Agrawal S, Goyal R, Tomida J. Proceedings of the 41st Annual International Cryptology Conference, August 16-20, 2021[C]. Berlin: Springer, 2021.
[53] Agrawal S, Goyal R, Tomida J. Proceedings of the 20th International Conference on Theory of Cryptography, November 7-10, 2022[C]. Berlin: Springer, 2022.
[54] Liang Y, Cao Z, Dong X, et al. Proceedings of the 20th International Conference on Information and Communications Security, October 29-31, 2018[C]. Berlin: Springer, 2018.
[55] Yang H, Su Y, Qin J, et al. Verifiable inner product computation on outsourced database for authenticated multi-user data sharing[J]. Information Sciences, 2020, 539: 295-311.
[56] Yang H, Su Y, Qin J, et al. Privacy-preserving outsourced inner product computation on encrypted database[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 1320-1337.
[57] Qian X, Li H, Hao M, et al. Proceedings of the GLOBECOM 2022-2022 IEEE Global Communications Conference, December 4-8, 2022[C]. Piscataway: IEEE, 2022.
[58] Katz J, Sahai A, Waters B. Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, April 13-17, 2008[C]. Berlin: Springer, 2008.
[59] Okamoto T, Takashima K. Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security, December 6-10, 2009[C]. Berlin: Springer, 2009.
[60] Agrawal S, Agrawal S, Badrinarayanan S, et al. Proceedings of the 18th IACR International Conference on Practice and Theory of Public-key Cryptography, March 30-April 1, 2015[C]. Berlin: Springer, 2015.

【责任编辑: 卓祯雨】