具有小规模公开参数的适应安全的非零内积加密方案

高海英 魏 铎*
(信息工程大学 郑州 450001)

摘 要：内积加密是一种支持内积形式的函数加密，已有内积加密方案的公开参数规模较大，为解决该问题，该文基于素数阶熵扩张引理，利用双对偶向量空间(DPVS)技术，提出一个公开参数规模较小的具有适应安全性的内积加密方案。在方案的私钥生成算法中，将用户的属性向量的分量与主私钥向量结合，生成一个可与熵扩张引理中密钥分量结合的向量；在方案的加密算法中，将内积向量的每一分量与熵扩张引理中的部分密文分量结合。在素数阶熵扩张引理和$\mathrm{MDDH}^n_{k,k+1}$困难假设成立条件下，证明了方案具有适应安全性。该文方案公开参数仅有10个群元素，与现有内积加密方案相比，公开参数规模最小。

关键词：内积加密；素数阶熵扩张引理；$\mathrm{MDDH}^n_{k,k+1}$困难假设；适应安全
中图分类号：TN918.1 **文献标识码**：A **文章编号**：1009-5896(2020)11-2698-08
DOI: 10.11999/JEIT190510

Adaptive Secure Non-zero Inner Product Encryption Scheme with Small-scale Public Parameters

GAO Haiying WEI Duo
(*University of Information Engineering*, *Zhengzhou* 450001, *China*)

Abstract: Inner product encryption is a kind of function encryption which supports inner product form. The public parameter scale of the existing inner product encryption schemes are large. In order to solve this problem, based on prime-order bilinear entropy expansion lemma and Double Pairing Vector Space (DPVS), an inner product encryption scheme is proposed in this paper, which has fewer public parameters and adaptive security. In the private key generation algorithm of the scheme, the components of the user's attribute with the main private key are combined to generate a vector that can be combined with the key components in the entropy expansion lemma, and in encryption algorithm of the scheme, each component of the inner product vector is combined with ciphertext component in the entropy expansion lemma. Finally, under the condition of prime order bilinear entropy extension lemma and $\mathrm{MDDH}^n_{k,k+1}$ difficult assumption, the adaptive secure of the scheme is proved. The proposed scheme has only 10 group elements as public parameters, which is the smallest compared with the existing inner product encryption schemes.

Key words: Inner product encryption; Prime-order bilinear entropy expansion; $\mathrm{MDDH}^n_{k,k+1}$ difficult assumption; Adaptive secure

1 引言

函数加密[1,2]作为一种新型的公钥加密形式，它包含身份基加密[3]、隐藏向量加密[4]、内积加密[5]、谓词加密[6]和属性基加密。与传统的公钥加密相比，函数加密在共享数据方面提供了细粒度的访问控制功能。函数加密支持的最广泛的一类关系是内积谓词，在内积函数加密方案中，密文与向量$\boldsymbol{x}$相关联，密钥与向量$\boldsymbol{y}$相关联，当且仅当$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}=z$时才能正确解密，$z=0$时称为0内积加密，$z\neq 0$时称为非零内积加密，本文所研究的是$z=1$的非零内积加密，即$z=1$时正确解密。

目前已提出一系列内积加密方案，在2008年欧密会上，Katz等人[5]基于合数阶群上的双线性映射提出了一个内积加密方案，该方案的公开参数长度不固定，仅实现选择安全；文献[7,8]给出的内积加密方案虽然实现了适应安全，但方案的公开参数个数仍然与系统属性个数呈平方关系；2012年Okamoto

收稿日期：2019-07-08；改回日期：2020-03-28；网络出版：2020-09-02
*通信作者： 魏铎 1500506441@qq.com
基金项目：国家自然科学基金(61702548, 61601515)，河南省基础与前沿技术课题(162300410192)
Foundation Items: The National Natural Science Foundation of China (61702548, 61601515), The Fundamental and Frontier Technology Research of Henan Province (162300410192)

等人[9]利用DPVS技术给出了第1个公开参数长度固定的内积加密方案，该方案实现了适应安全；2018年Tomida等人[10]给出一个适应安全的内积加密方案，公开参数个数比文献[9]有所减少，但规模仍较大。从已有方案可以看出，这些方案的公开参数的个数或者随属性个数增长，或者达到固定但规模仍较大，并且其中一些方案的安全性较弱，仅具有选择安全性。从研究现状来看，如何设计一个公开参数规模较小的适应安全的内积加密方案是一个待解决的问题。

为了设计适应安全的内积加密方案，通过前期研究发现：在2009年，Waters[11]首次提出了双系统加密技术，该技术是在素数阶双线性群上实现的，相较于合数阶双线性群效率更高，这为设计适应安全的函数加密方案提供了方法指导。基于双系统加密的思想，2010年，Okamoto等人[8]提出了双对偶向量空间(Dual Pairing Vector Space, DPVS)技术，该技术可用于设计适应安全加密方案。2015年，Chen等人[12]同样基于双系统加密思想提出了一个素数阶双系统群的框架，素数阶双系统群本质上仍然利用的是DPVS技术，但对正交性进行弱化，进一步提高了方案效率。2018年，文献[13]对素数阶双系统群框架进行扩展，提出了新的DPVS技术，适用范围更广，新的DPVS技术为我们设计适应安全的内积加密方案提供了技术基础。

2014年，Wee[14]给出了谓词编码(predicate encoding)的定义，并给出了一个基于双系统加密思想设计的函数加密框架，方案的构造被简化为设计谓词编码，即不同的谓词编码对应了不同的函数加密方案。一个谓词编码主要涉及两部分：发送者编码sE和接收者编码rE，这两个编码分别与密文生成算法和密钥生成算法相关联，并且文中给出了一些编码的例子以及基本的设计思想。基于Wee提出的编码思想，结合用到的DPVS技术，本文在设计内积加密方案时，主要解决在sE编码和rE编码中分别嵌入向量的问题。

为了解决内积加密方案的公开参数问题，文献[13]从Lewko等人提出的一个KP-ABE方案出发，对方案的公开参数进行了压缩，提出了一个素数阶熵扩张引理，该引理给出了公开参数、密文以及密钥的分量形式，具体见2.3节。素数阶熵扩张引理的提出，为我们解决参数规模问题提供了方法指导。基于熵扩张引理，在设计方案的私钥生成算法时，首先将用户的属性向量的分量与主私钥向量结合，继而将该向量嵌入到熵扩张引理的部分密钥分量中；方案的加密算法，则将内积向量的每一分量与熵扩张引理中的部分密文分量结合，完成方案密文的生成。基于此方法设计出的方案正确性可以得到保证，并在素数阶熵扩张引理和MDDH$^n_{k,k+1}$困难假设成立条件下，证明了方案具有适应安全性。与现有内积加密方案相比，本文方案公开参数规模最小。

本文结构如下：第2节介绍相关符号、定义以及困难假设；第3节详细描述方案的设计并给出正确性证明；第4节给出了方案的安全性证明；第5节给出方案的性能对比；第6节总结全文。

2　预备知识

2.1　符号说明

粗体大写字母$\boldsymbol{A}$：代表矩阵$\boldsymbol{A}$, $\boldsymbol{A}^{\mathrm{T}}$表示$\boldsymbol{A}$的转置；

粗体小写字母$\boldsymbol{a}$：代表向量$\boldsymbol{a}$, $\boldsymbol{a}^{\mathrm{T}}$表示$\boldsymbol{a}$的转置；

span($\boldsymbol{A}$)：表示矩阵$\boldsymbol{A}$列向量张成的线性空间；

span^{k+1}($\boldsymbol{A}$)：表示矩阵$\boldsymbol{A}$列向量张成的线性空间中的k+1个向量；

$\boldsymbol{Z}_p^{m\times n}$：表示模$p$剩余类环上$m\times n$维矩阵集合；

$\boldsymbol{Z}_p^{m}$：表示模p剩余类环上m维向量集合；

$(\boldsymbol{A}_1|\boldsymbol{A}_2|\boldsymbol{A}_3)$：表示$\boldsymbol{Z}_p$上3个矩阵的连接；

$(\boldsymbol{A}_1^{||}|\boldsymbol{A}_2^{||}|\boldsymbol{A}_3^{||})^{\mathrm{T}}$：表示$(\boldsymbol{A}_1|\boldsymbol{A}_2|\boldsymbol{A}_3)$的逆，满足条件：$\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_i^{||}=I, (\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_j^{||}=\boldsymbol{0})_{i\neq j}$。

对于$\boldsymbol{a}=(a_1,a_2,\cdots,a_n), \boldsymbol{b}=(b_1,b_2,\cdots,b_n)\in\boldsymbol{Z}_p^n$：$[\boldsymbol{a}]_1$表示$(g_1{}^{a_1},g_1{}^{a_2},\cdots,g_1{}^{a_n})$，$[\boldsymbol{b}]_2$表示$(g_2{}^{b_1},g_2{}^{b_2},\cdots,g_2{}^{b_n})$，$[\boldsymbol{a}]_T$表示$e(g_1,g_2)^a$，即$[\boldsymbol{a}]_T$=$(e(g_1,g_2)^{a_1},e(g_1,g_2)^{a_2},\cdots,e(g_1,g_2)^{a_n})$。

2.2　双线性映射

定义1[15]　(合数阶双线性映射) 令$G=(G_N,H_N,G_T,e)$, G_N,H_N,G_T都是阶为$N=p_1p_2p_3$的循环群，p_1,p_2,p_3都是大素数，$e:G_N\times H_N\to G_T$，u是G_N的生成元，u_1,u_2,u_3分别为G_{p_1},G_{p_2},G_{p_3}的生成元，h是H_N的生成元，h_1,h_2,h_3分别为H_{p_1},H_{p_2},H_{p_3}的生成元。若映射$e:(G_N,H_N)\to G_T$满足以下3条性质，称e为合数阶群双线性映射。3条性质描述如下：

(1) 双线性性：对于$\forall u\in G_N,h\in H_N,a,b\in Z_p$，有$e(u^a,h^b)=e(u,h)^{ab}$。

(2) 非退化性：$\exists u\in G_N,h\in H_N$，使得$e(u,h)$的阶是$N$。

(3) 正交性：$\forall i,j\in\{1,2,3\},i\neq j$，满足$e(u_i,h_j)=1$。

令(G_1,G_2,G_T)是阶为素数p的双线性群，G_1的生成元是g_1, G_2的生成元是g_2。

文献[13]给出的在素数阶群上模拟合数阶群性质的DPVS技术如下所述：

随机选取矩阵$\boldsymbol{A}_1\leftarrow Z_p^{l\times l_1},\boldsymbol{A}_2\leftarrow Z_p^{l\times l_2},\boldsymbol{A}_3\leftarrow Z_p^{l\times l_3}$，并定义$(\boldsymbol{A}_1^{||}|\boldsymbol{A}_2^{||}|\boldsymbol{A}_3^{||})^{\mathrm{T}}$，满足条件：$\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_i^{||}=I$,

$(\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_j^{||}=0)_{i\neq j}$对于合数阶双线性群与素数阶双线性群$G_1$和$G_2$有如下对应关系：令$u_i\to[\boldsymbol{A}_i]_1$，$h_i\to[\boldsymbol{A}_i^{||}]_2$，$u_i^s\to[\boldsymbol{A}_i s]_1$，$w\in Z_N\to\boldsymbol{W}\in Z_p^{l\times l_w}$，$u_i^w\to[\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{W}]_1$。

并定义如下运算法则：

$\forall\boldsymbol{A}\in Z_p^{m\times n},\boldsymbol{B}\in Z_p^{n\times t}$，令$e([\boldsymbol{A}]_1,[\boldsymbol{B}]_2)=e(g_1,g_2)^{\boldsymbol{AB}}$。

则可以得到：$\forall i,j\in\{1,2,3\},i\neq j$，满足$e([\boldsymbol{A}_i^{\mathrm{T}}]_1,[\boldsymbol{A}_j^{||}]_2)=e(g_1,g_2)^{\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_j^{||}}=[\boldsymbol{0}]_T$。

2.3 困难假设

定义2[13] (判定性矩阵Diffie – Hellman假设($\mathrm{MDDH}_{k,k+1}^n$Assumption))已知分布$\mathbb{G}=(p,G_1,G_2,G_T,e)$，$\boldsymbol{B}\leftarrow Z_p^{(k+1)\times k},\boldsymbol{S}\leftarrow Z_p^{k\times n},\boldsymbol{Z}\leftarrow Z_p^{(k+1)\times n}$，对于任意多项式时间敌手$\mathcal{B}$，区分$[\mathbf{BS}]_1$和$[\boldsymbol{Z}]_1$的优势$\mathrm{Adv}_{\mathcal{B}}^{\mathrm{MDDH}_{k,k+1}^n}=|\Pr[\mathcal{B}(\mathbb{G},[\boldsymbol{B}]_1,[\mathbf{BS}]_1)=1]-\Pr[\mathcal{B}(\mathbb{G},[\boldsymbol{B}]_1,[\boldsymbol{Z}]_1)=1]|$是可以忽略的，同样，区分$[\mathbf{BS}]_2$和$[\boldsymbol{Z}]_2$的优势$\mathrm{Adv}_{\mathcal{B}}^{\mathrm{MDDH}_{k,k+1}^n}=|\Pr[\mathcal{B}(\mathbb{G},[\boldsymbol{B}]_2,[\mathbf{BS}]_2)=1]-\Pr[\mathcal{B}(\mathbb{G},[\boldsymbol{B}]_2,[\boldsymbol{Z}]_2)=1]|$也是可以忽略的。

定义3[13] (素数阶熵扩张引理(prime-order entropy expansion lemma))：随机选取$(\boldsymbol{A}_1,\boldsymbol{a}_2,\boldsymbol{A}_3)\leftarrow Z_p^{(2k+1)\times(k+1)}\times Z_p^{(2k+1)}\times Z_p^{(2k+1)\times(k+1)}$，$\boldsymbol{B}\leftarrow Z_p^{(k+1)\times k}$，并且计算$(\boldsymbol{A}_1,\boldsymbol{a}_2,\boldsymbol{A}_3)$的逆$(\boldsymbol{A}_1^{||},\boldsymbol{a}_2^{||},\boldsymbol{A}_3^{||})^{\mathrm{T}}$，对于任意多项式时间敌手$\mathcal{B}$区分以下两个分布的优势是可以忽略的：

$$\left\{\begin{array}{l}\mathrm{aux}:[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1\\ \mathrm{ct}:[\boldsymbol{c}_0=\boldsymbol{c}^{\mathrm{T}}]_1,\{[\boldsymbol{c}_{1,j}=\boldsymbol{c}^{\mathrm{T}}\boldsymbol{W}\\ \quad+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)]_1,[\boldsymbol{c}_{2,j}=\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in[n]}\\ \mathrm{sk}:\{[\boldsymbol{K}_{0,j}=\mathbf{W}\mathbf{D}_j]_2,[\boldsymbol{K}_{1,j}=\boldsymbol{D}_j]_2,\\ [\boldsymbol{K}_{2,j}=(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)\boldsymbol{D}_j]_2\}_{j\in[n]}\end{array}\right\}\approx_c$$

$$\left\{\begin{array}{l}\mathrm{aux}:[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1\\ \mathrm{ct}:[\boldsymbol{c}_0=\boldsymbol{c}^{\mathrm{T}}]_1,\{[\boldsymbol{c}_{1,j}=\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{W}+\boldsymbol{V}_j)\\ \quad+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_j)]_1,[\boldsymbol{c}_{2,j}=\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in[n]}\\ \mathrm{sk}:\{[\boldsymbol{K}_{0,j}=(\boldsymbol{W}+\boldsymbol{V}_j)\boldsymbol{D}_j]_2,[\boldsymbol{K}_{1,j}=\boldsymbol{D}_j]_2,\\ [\boldsymbol{K}_{2,j}=(\boldsymbol{W}_0+j\cdot\mathbf{W}_1+\boldsymbol{U}_j)\boldsymbol{D}_j]_2\}_{j\in[n]}\end{array}\right\}$$

其中$\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1\leftarrow Z_p^{(2k+1)\times(k+1)}$，$\boldsymbol{V}_j,\boldsymbol{U}_j\leftarrow\mathrm{span}^{k+1}(\boldsymbol{a}_2^{||})$，$\boldsymbol{D}_j\leftarrow\mathrm{span}^{k+1}(\boldsymbol{B})$，左分布$\boldsymbol{c},\boldsymbol{c}_j\leftarrow\mathrm{span}(\boldsymbol{A}_1)$，右分布$\boldsymbol{c},\boldsymbol{c}_j\leftarrow\mathrm{span}(\boldsymbol{A}_1,\boldsymbol{a}_2)$。

2.4 非零内积加密方案形式化定义

一个非零内积加密方案由以下4个多项式时间算法构成：

$\mathrm{Setup}(1^\lambda,1^n)$：该概率初始化算法输入安全参数$(1^\lambda,1^n)$，输出系统公开参数mpk和主密钥msk。

$\mathrm{Enc}(\mathrm{mpk},\boldsymbol{x},m)$：该概率加密算法输入公开参数mpk，向量$\boldsymbol{x}\in Z_p^n$和明文$m$，输出密文$\mathrm{ct}_{\boldsymbol{x}}$。

$\mathrm{KeyGen}(\mathrm{mpk},\mathrm{msk},\boldsymbol{y})$：该概率密钥生成算法输入主密钥msk和向量$\boldsymbol{y}\in Z_p^n$，输出私钥$\mathrm{sk}_{\boldsymbol{y}}$。

$\mathrm{Dec}(\mathrm{mpk},\mathrm{sk}_{\boldsymbol{y}},\mathrm{ct}_{\boldsymbol{x}})$：该确定性解密算法输入$\mathrm{sk}_{\boldsymbol{y}}$和$\mathrm{ct}_{\boldsymbol{x}}$，若$P(\boldsymbol{x},\boldsymbol{y})=1$(即$\boldsymbol{x}^{\mathbf{T}}\boldsymbol{y}=1$)，则可以解密密文。

2.5 非零内积加密方案的适应性安全模型

下面通过挑战者$\mathcal{B}$和攻击者$\mathcal{A}$之间的交互游戏给出非零内积加密方案的适应性安全模型。

初始化阶段：挑战者$\mathcal{B}$运行方案的初始化算法，将生成的公开参数mpk发送给攻击者$\mathcal{A}$，保留主私钥msk。

查询阶段1：攻击者$\mathcal{A}$选择向量$\boldsymbol{y}'$进行多项式次数私钥查询。挑战者$\mathcal{B}$运行KeyGen算法，将向量$\boldsymbol{y}'$对应的私钥发送给$\mathcal{A}$。

挑战阶段：攻击者$\mathcal{A}$向挑战者提交两个等长的明文m_0和m_1，以及挑战向量$\boldsymbol{x}^*$，(私钥查询中用到的向量$\boldsymbol{y}'$与挑战向量$\boldsymbol{x}^*$都不满足$(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{y}'=1$)，挑战者$\mathcal{B}$随机选取$b\in\{0,1\}$，计算$\mathrm{ct}_{\boldsymbol{x}^*}=\mathrm{Enc}(\mathrm{mpk},\boldsymbol{x}^*,m_b)$，并将$\mathrm{ct}_{\boldsymbol{x}^*}$作为挑战密文返回给攻击者$\mathcal{A}$。

查询阶段2：与查询阶段1相同。

猜测阶段：攻击者$\mathcal{A}$给出关于b的猜测b'。

如果$b'=b$，称攻击者$\mathcal{A}$赢得了此游戏，定义攻击者$\mathcal{A}$在此游戏中的优势为$\mathrm{Adv}_{\mathcal{A}}(\lambda)=|\Pr(b'=b)-1/2|$。

定义4 如果对于所有多项式时间的攻击者，赢得上述游戏的优势都是可忽略的，则称该内积加密方案是适应性安全的。

3 方案描述

$\mathrm{Setup}(1^\lambda,1^n)$：系统输入安全参数$\lambda$和系统属性个数$n$。随机选取

$$\boldsymbol{A}_1\leftarrow Z_p^{(2k+1)\times k},\boldsymbol{B}\leftarrow Z_p^{(k+1)\times k},$$
$$\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1\leftarrow Z_p^{(2k+1)\times(k+1)},\boldsymbol{k}\leftarrow Z_p^{2k+1}。$$

生成公开参数mpk和系统主密钥msk：

$$\mathrm{mpk}=\{[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,$$
$$e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\},\ \mathrm{msk}=(\boldsymbol{k},\boldsymbol{B},\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1)。$$

$\mathrm{Enc}(\mathrm{mpk},\boldsymbol{x},m)$：输入向量$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$和编码为$G_T$上的消息$m$，选取$\boldsymbol{c},\boldsymbol{c}_j\leftarrow\mathrm{span}(\boldsymbol{A}_1)$，计算生成密文$\mathrm{ct}_{\boldsymbol{x}}$为

$$\mathrm{ct}_{\boldsymbol{x}}=\Big\{C_0=[\boldsymbol{c}^{\mathrm{T}}]_1,$$
$$\Big\{C_{1,j}=[x_j\boldsymbol{c}^{\mathrm{T}}\boldsymbol{W}+x_j\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)]_1,$$
$$C_{2,j}=[x_j\boldsymbol{c}_j^{\mathrm{T}}]_1\Big\}_{j\in[\mathrm{n}]},C'=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\cdot m\Big\}。$$

$\mathrm{KeyGen}(\mathrm{mpk},\mathrm{msk},\boldsymbol{y})$：输入主密钥msk和$\boldsymbol{y}=(y_1,y_2,\cdots,y_n)$，选取$\boldsymbol{d}_j\leftarrow\mathrm{span}(\boldsymbol{B})$，生成$\boldsymbol{y}$对应的私钥为

$\mathrm{sk}_{\boldsymbol{y}} := (\{K_{0,j} = [y_j\boldsymbol{k} + \boldsymbol{W}\boldsymbol{d}_j]_2, K_{1,j} = [\boldsymbol{d}_j]_2,$
$K_{2,j} = [(\boldsymbol{W}_0 + j \cdot \boldsymbol{W}_1)\boldsymbol{d}_j]_2\}_{j\in[n]})$。

$\mathrm{Dec}(\mathrm{mpk}, \mathrm{sk}_{\boldsymbol{y}}, \mathrm{ct}_{\boldsymbol{x}})$：输入密钥$\mathrm{sk}_{\boldsymbol{y}}$和密文$\mathrm{ct}_{\boldsymbol{x}}$。若$\boldsymbol{x}$与$\boldsymbol{y}$满足$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 1$，那么解密者就可以正确解密。其解密过程如下：

$$m = C' \Big/ \prod_{j=1}^{n} (e((C_0)^{x_j}, K_{0,j}) \cdot e(C_{1,j}, K_{1,j})^{-1} \cdot e(C_{2,j}, K_{2,j}))。$$

下面验证，当$P(\boldsymbol{x}, \boldsymbol{y}) = 1$，即$\boldsymbol{x}^{\mathrm{T}}y = \sum_{i=1}^{n} x_i y_i = 1$时，解密者能正确解密。

$$\begin{aligned}
&e((C_0)^{x_j}, K_{0,j}) \cdot e(C_{1,j}, K_{1,j})^{-1} \cdot e(C_{2,j}, K_{2,j}) \\
&= e([x_j\boldsymbol{c}^{\mathrm{T}}]_1, [y_j\boldsymbol{k} + \boldsymbol{W}\boldsymbol{d}_j]_2) \\
&\quad \cdot e([x_j\boldsymbol{c}^{\mathrm{T}}\boldsymbol{W} + x_j\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0 + j \cdot \boldsymbol{W}_1)]_1, [\boldsymbol{d}_j]_2)^{-1} \\
&\quad \cdot e([x_j\boldsymbol{c}_j^{\mathrm{T}}]_1, [(\boldsymbol{W}_0 + j \cdot \boldsymbol{W}_0)\boldsymbol{d}_j]_2) \\
&= [x_j\boldsymbol{c}^{\mathrm{T}}y_j\boldsymbol{k} + x_j\boldsymbol{c}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{d}_j - x_j\boldsymbol{c}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{d}_j \\
&\quad - x_j\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0 + j \cdot \boldsymbol{W}_1)\boldsymbol{d}_j + x_j\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0 + j \cdot \boldsymbol{W}_1)\boldsymbol{d}_j]_T \\
&= [x_j y_j \boldsymbol{c}^{\mathrm{T}}\boldsymbol{k}]_T。
\end{aligned}$$

$$\begin{aligned}
&\prod_{j=1}^{n} (e((C_0)^{x_j}, K_{0,j}) \cdot e(C_{1,j}, K_{1,j})^{-1} \cdot e(C_{2,j}, K_{2,j})) \\
&= \prod_{j=1}^{n} [x_j y_j \boldsymbol{c}^{\mathrm{T}}\boldsymbol{k}]_T = [\boldsymbol{c}^{\mathrm{T}}\boldsymbol{k} \sum_{j=1}^{n} x_j y_j]_T \\
&= [\boldsymbol{c}^{\mathrm{T}}\boldsymbol{k}]_T = e([\boldsymbol{c}^{\mathrm{T}}]_1, [\boldsymbol{k}]_2)。
\end{aligned}$$

因此：

$$\frac{C'}{\prod_{j=1}^{n} (e((C_0)^{x_j}, K_{0,j}) \cdot e(C_{1,j}, K_{1,j})^{-1} \cdot e(C_{2,j}, K_{2,j}))} = \frac{e([\boldsymbol{c}^{\mathrm{T}}]_1, [\boldsymbol{k}]_2) \cdot m}{e([\boldsymbol{c}^{\mathrm{T}}]_1, [\boldsymbol{k}]_2)} = m。$$

4　安全性证明

方案的安全性证明是基于一系列Game序列之间的不可区分。首先给出证明过程中需要用到的密文分布和密钥分布，如下所述：

(1) 密文分布

标准密文：由加密算法生成，其中$\boldsymbol{c}, \boldsymbol{c}_j \leftarrow \mathrm{span}(\boldsymbol{A}_1)$。

熵扩张密文：与标准密文不同的是$\boldsymbol{c}, \boldsymbol{c}_j \leftarrow \mathrm{span}(\boldsymbol{A}_1, \boldsymbol{a}_2)$，

$$\begin{aligned}
\boldsymbol{W} \to \boldsymbol{V}_j' &= \boldsymbol{W} + \boldsymbol{V}_j,\ \boldsymbol{W}_0 + j \cdot \boldsymbol{W}_1 \to \boldsymbol{U}_j' \\
&= \boldsymbol{W}_0 + j \cdot \boldsymbol{W}_1 + \boldsymbol{U}_j。
\end{aligned}$$

其中$\boldsymbol{V}_j, \boldsymbol{U}_j \leftarrow \mathrm{span}^{k+1}(\boldsymbol{a}_2^{\|})$，熵扩张密文形式为

$$\mathrm{ct}_{\boldsymbol{x}} = \Big\{C_0 = [\boldsymbol{c}^{\mathrm{T}}]_1, \{C_{1,j} = [x_j\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}_j' + x_j\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{U}_j']_1, C_{2,j} = [x_j\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in[n]}, C' = e([\boldsymbol{c}^{\mathrm{T}}]_1, [\boldsymbol{k}]_2) \cdot m\Big\}。$$

(2) 密钥分布：

标准密钥：由密钥生成算法生成。

熵扩张密钥：

$\mathrm{sk}_{\boldsymbol{y}} := (\{K_{0,j} = [y_j\boldsymbol{k} + \boldsymbol{V}_j'\boldsymbol{d}_j]_2, K_{1,j} = [\boldsymbol{d}_j]_2,$
$K_{2,j} = [\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]})$。

其中$\boldsymbol{d}_j \leftarrow \mathrm{span}(\boldsymbol{B})$。

伪标准密钥：

$\mathrm{sk}_{\boldsymbol{y}} := (\{K_{0,j} = [y_j\boldsymbol{k} + \boldsymbol{V}_j'\boldsymbol{d}_j]_2, K_{1,j} = [\boldsymbol{d}_j]_2,$
$K_{2,j} = [\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]})$。

其中$\boldsymbol{d}_j \leftarrow Z_p^{k+1}$。

伪半功能密钥：

$\mathrm{sk}_{\boldsymbol{y}} := (\{K_{0,j} = \left[y_j\left(\boldsymbol{k} + a\boldsymbol{a}_2^{\|}\right) + \boldsymbol{V}_j'\boldsymbol{d}_j\right]_2, K_{1,j} = [\boldsymbol{d}_j]_2,$
$K_{2,j} = [\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]})$。

其中$\boldsymbol{d}_j \leftarrow Z_p^{k+1}$。

半功能密钥：

$\mathrm{sk}_{\boldsymbol{y}} := (\{K_{0,j} = \left[y_j\left(\boldsymbol{k} + a\boldsymbol{a}_2^{\|}\right) + \boldsymbol{V}_j'\boldsymbol{d}_j\right]_2, K_{1,j} = [\boldsymbol{d}_j]_2,$
$K_{2,j} = [\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]})$。

其中$\boldsymbol{d}_j \leftarrow \mathrm{span}(\boldsymbol{B})$。

方案的安全性证明思想是Game序列之间的不可区分，假设攻击者$\mathcal{A}$在一次Game中最多可进行Q次私钥查询，用$\mathrm{Adv}_{xx}(\lambda)$表示攻击者$\mathcal{A}$在$\mathrm{Game}_{xx}$的优势。基于上述密文、密钥分布，下面详细描述Game序列，并在表1给出了Game序列的对比。

Game_0：查询得到标准私钥，挑战密文是标准密文。

$\mathrm{Game}_{0'}$：查询得到熵扩张私钥，挑战密文是熵扩张密文。

Game_i：查询前$i-1$次是半功能私钥、最后$Q-i+1$次是熵扩张私钥，挑战密文是熵扩张密文。

$\mathrm{Game}_{i,1}$：查询前$i-1$次是半功能私钥、最后$Q-i$次是熵扩张私钥、第i次是伪标准私钥，挑战密文是熵扩张密文。

表 1　Game序列

Game	ct	sk		
		$\kappa < i$	$\kappa = i$	$\kappa > i$
0	标准		标准	
0'	熵扩张		熵扩张	
i	熵扩张	半功能	熵扩张	熵扩张
$i,1$	–	–	伪标准	–
$i,2$	–	–	伪半功能	–
$i,3$	–	–	半功能	–
Final	随机消息		半功能	

$\text{Game}_{i,2}$：查询前$i-1$次是半功能私钥、最后$Q-i$次是熵扩张私钥、第i次是伪半功能私钥，挑战密文是熵扩张密文。

$\text{Game}_{i,3}$：查询前$i-1$次是半功能私钥、最后$Q-i$次是熵扩张私钥、第i次是半功能私钥，挑战密文是熵扩张密文。

$\text{Game}_{\text{Final}}$：查询得到半功能私钥，挑战密文是对随机数加密的熵扩张密文。

引理1 如果存在一个攻击者$\mathcal{A}$在Game_0和$\text{Game}_{0'}$的攻击优势满足$|\text{Adv}_0(\lambda)-\text{Adv}_{0'}(\lambda)|>\varepsilon$，那么可以构造一个算法$\mathcal{B}_0$以不可忽略的优势区分熵扩张引理左右分布，并且$\text{Time}(\mathcal{B}_0)\approx\text{Time}(\mathcal{A})$。

证明 挑战者$\mathcal{B}_0$得到分布：

$$\left\{\begin{array}{l}\text{aux}:[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1\\ \text{ct}:[\boldsymbol{c}_0]_1,\{[\boldsymbol{c}_{1,j}]_1,[\boldsymbol{c}_{2,j}]_1\}_{j\in[n]}\\ \text{sk}:[\boldsymbol{K}_{0,j}]_2,[\boldsymbol{K}_{1,j}]_2,[\boldsymbol{K}_{2,j}]_2\end{array}\right\}$$

$\mathcal{B}_0$需要判断此分布是熵扩张引理的左分布或是右分布。

初始化阶段：挑战者$\mathcal{B}_0$模拟内积加密方案，在Z_p^{2k+1}随机选取向量$\boldsymbol{k}$，输出公开参数mpk给$\mathcal{A}$。

$$\begin{aligned}\text{mpk}=\{&[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,\\ &e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\}。\end{aligned}$$

查询阶段1：攻击者$\mathcal{A}$申请向量$\boldsymbol{y}'=(y_1',y_2',\cdots,y_n')$对应的私钥，挑战者$\mathcal{B}_0$模拟$\text{KeyGen}(\text{mpk},\text{msk},\boldsymbol{y})$算法，在$Z_p^{k+1}$上随机选取向量$\boldsymbol{d}_j'$，生成向量$\boldsymbol{y}'$对应的私钥：

$$\begin{aligned}\text{sk}_{\boldsymbol{y}'}:=(\{&K_{0,j}=[y_j'\boldsymbol{k}+\boldsymbol{K}_{0,j}\boldsymbol{d}_j']_2,\\ &K_{1,j}=[\boldsymbol{K}_{1,j}\boldsymbol{d}_j']_2,K_{2,j}=[\boldsymbol{K}_{2,j}\boldsymbol{d}_j']_2\}_{j\in[n]})。\end{aligned}$$

挑战阶段：攻击者$\mathcal{A}$选择两个等长的消息m_0和m_1，以及挑战向量$\boldsymbol{x}^*=(x_1^*,x_2^*,\cdots,x_n^*)$(查询阶段1中的向量$\boldsymbol{y}'$与挑战向量$\boldsymbol{x}^*$都不满足$(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{y}'=1$)发送给挑战者$\mathcal{B}_0$，挑战者$\mathcal{B}_0$随机选取$b\in\{0,1\}$，利用向量$\boldsymbol{k}$以及得到的分布生成挑战密文：

$$\begin{aligned}\text{ct}_{\boldsymbol{x}^*}=\Big\{&C_0{}^{\boldsymbol{x}^*}=[\boldsymbol{c}_0]_1,\{C_{1,j}{}^{\boldsymbol{x}^*}=[\boldsymbol{c}_{1,j}]_1{}^{x_j{}^*},\\ &C_{2,j}{}^{\boldsymbol{x}^*}=[\boldsymbol{c}_{2,j}]_1{}^{x_j{}^*}\}_{j\in[n]},C^{\boldsymbol{x}^*}=e([\boldsymbol{c}_0]_1,[\boldsymbol{k}]_2)\cdot m_b\Big\}。\end{aligned}$$

查询阶段2：与查询阶段1相同。

猜测阶段：攻击者$\mathcal{A}$给出b的猜测b'。

注：如果挑战者$\mathcal{B}_0$得到的是左分布，那么$\mathcal{A}$询问得到标准密钥，挑战密文是标准密文，如果挑战者$\mathcal{B}_0$得到的是右分布，那么$\mathcal{A}$询问得到熵扩张密钥，挑战密文是熵扩张密文。

如果攻击者$\mathcal{A}$的攻击优势满足$|\text{Adv}_0(\lambda)-\text{Adv}_{0'}(\lambda)|>\varepsilon$，那么挑战者$\mathcal{B}_0$同样以不可忽略的优势区分素数阶熵扩张引理的左右分布。

证毕

引理2 由表1，可以得到$\text{Game}_{0'}\equiv\text{Game}_1$。

引理3 如果存在一个攻击者$\mathcal{A}$在Game_i和$\text{Game}_{i,1}$的优势满足$|\text{Adv}_i(\lambda)-\text{Adv}_{i,1}(\lambda)|>\varepsilon$，那么可以构造一个算法$\mathcal{B}_1$以不可忽略的优势解决$\text{MDDH}_{k,k+1}^n$问题，并且$\text{Time}(\mathcal{B}_1)\approx\text{Time}(\mathcal{A})$。

证明 挑战者$\mathcal{B}_1$得到分布$\Big([\boldsymbol{B}]_2,\{[\boldsymbol{t}_j]_2\}_{j\in[n]}\Big)$，其中$[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n]=\boldsymbol{\text{BS}},\boldsymbol{B}\leftarrow Z_p^{(k+1)\times k},\boldsymbol{S}\leftarrow Z_p^{k\times n}$或者$[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n]=\boldsymbol{Z},\boldsymbol{Z}\leftarrow Z_p^{(k+1)\times n}$。

初始化算法：挑战者$\mathcal{B}_1$模拟方案，随机选取$\boldsymbol{A}_1,\boldsymbol{a}_2\leftarrow Z_p^{(2k+1)\times(k+1)}\times Z_p^{2k+1}$，计算出$\boldsymbol{a}_2^{\parallel}$，随机选取$\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1\leftarrow Z_p^{(2k+1)\times(k+1)}$，$\boldsymbol{V}_j,\boldsymbol{U}_j\leftarrow\text{span}^{k+1}(\boldsymbol{a}_2^{\parallel})$，$\alpha\leftarrow Z_p$。在$Z_p^{2k+1}$上随机选取向量$\boldsymbol{k}$，输出公开参数mpk给$\mathcal{A}$：

$$\begin{aligned}\text{mpk}=\{&[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,\\ &e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\}。\end{aligned}$$

查询阶段1,2：设将攻击者$\mathcal{A}$对挑战者$\mathcal{B}_1$进行第κ次私钥询问，对应的向量是$\boldsymbol{y}'=(y_1',y_2',\cdots,y_n')$，分3种情况讨论：

情况1：$\kappa<i$，挑战者$\mathcal{B}_1$利用$[\boldsymbol{B}]_2$，随机选取$[\boldsymbol{d}_j]_2\leftarrow\text{span}([\boldsymbol{B}]_2)$，并且利用向量$\boldsymbol{k}$、随机数$\alpha$和攻击者$\mathcal{A}$询问的向量$\boldsymbol{y}'$生成半功能私钥$\text{sk}_{\boldsymbol{y}'}$：

$$\begin{aligned}\text{sk}_{\boldsymbol{y}'}:=(\{&K_{0,j}=[y_j'(\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel})+\boldsymbol{V}_j'\boldsymbol{d}_j]_2,\\ &K_{1,j}=[\boldsymbol{d}_j]_2,K_{2,j}=[\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]}),\end{aligned}$$

发送给攻击者$\mathcal{A}$。

情况2：$\kappa>i$，挑战者$\mathcal{B}_1$利用$[\boldsymbol{B}]_2$，随机选取$[\boldsymbol{d}_j]_2\leftarrow\text{span}([\boldsymbol{B}]_2)$，并且利用向量$\boldsymbol{k}$和攻击者$\mathcal{A}$询问的向量生成熵扩张私钥$\text{sk}_{\boldsymbol{y}'}$：

$$\begin{aligned}\text{sk}_{\boldsymbol{y}'}:=(\{&K_{0,j}=[y_j'\boldsymbol{k}+\boldsymbol{V}_j'\boldsymbol{d}_j]_2,K_{1,j}=[\boldsymbol{d}_j]_2,\\ &K_{2,j}=[\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]}),\end{aligned}$$

发送给攻击者$\mathcal{A}$。

情况3：$\kappa=i$，挑战者$\mathcal{B}_1$针对攻击者$\mathcal{A}$询问的向量$\boldsymbol{y}'=(y_1',y_2',\cdots,y_n')$，并且利用向量$\boldsymbol{k}$、$\{[\boldsymbol{t}_j]_2\}_{j\in[n]}$和攻击者$\mathcal{A}$询问的向量生成私钥 $\text{sk}_{\boldsymbol{y}'}$：

$$\begin{aligned}\text{sk}_{\boldsymbol{y}'}:=(\{&K_{0,j}=[y_j'\boldsymbol{k}+\boldsymbol{V}_j'\boldsymbol{t}_j]_2,K_{1,j}=[\boldsymbol{t}_j]_2,\\ &K_{2,j}=[\boldsymbol{U}_j'\boldsymbol{t}_j]_2\}_{j\in[n]}),\end{aligned}$$

发送给攻击者$\mathcal{A}$。

挑战阶段：攻击者$\mathcal{A}$选择两个等长的消息m_0和m_1，以及挑战向量$\boldsymbol{x}^*=(x_1^*,x_2^*,\cdots,x_n^*)$ (任何询问向量$\boldsymbol{y}'$与要挑战向量$\boldsymbol{x}^*$都不满足$(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{y}'=1$)发送给挑战者$\mathcal{B}_1$，挑战者$\mathcal{B}_1$随机选取$b\in\{0,1\}$，$\boldsymbol{c},\boldsymbol{c}_j\leftarrow\text{span}(\boldsymbol{A}_1,\boldsymbol{a}_2)$，生成熵扩张挑战密文：

$$\begin{aligned}\text{ct}_{\boldsymbol{x}^*}=\Big\{&C_0=[\boldsymbol{c}^{\mathrm{T}}]_1,\{C_{1,j}=[x_j{}^*\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}_j'+x_j{}^*\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{U}_j']_1,\\ &C_{2,j}=[x_j{}^*\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in[n]},C'=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\cdot m_b\Big\}。\end{aligned}$$

猜想阶段：攻击者$\mathcal{A}$给出b的猜测b'。

如果$[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n]=\mathbf{BS}$，那么第$i$次询问的私钥是熵扩张私钥，上述Game对应的是$\text{Game}_i$，如果$[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n]=\boldsymbol{Z}$，那么第$i$次询问的私钥是伪标准私钥,对应的是$\text{Game}_{i,1}$。

如果攻击者$\mathcal{A}$使$|\text{Adv}_i(\lambda)-\text{Adv}_{i,1}(\lambda)|>\varepsilon$不可忽视，那么挑战者$\mathcal{B}_1$同样以不可忽略的优势区分$[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n]=\mathbf{BS}$和$[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n]=\boldsymbol{Z}$。　证毕

引理4　对于任意攻击者$\mathcal{A}$，在$\text{Game}_{i,1}$和$\text{Game}_{i,2}$的优势满足$|\text{Adv}_{i,1}(\lambda)-\text{Adv}_{i,2}(\lambda)|=0$。

证明　$\text{Game}_{i,1}$和$\text{Game}_{i,2}$不同之处仅在于第i次私钥询问，挑战者在$\text{Game}_{i,1}$中用向量$\boldsymbol{k}$生成伪标准私钥，在$\text{Game}_{i,2}$中用$\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}$生成伪半功能私钥。下面说明这两个Game无法区分。

初始化阶段：同引理3中的初始化阶段，在这里挑战者随机选取$\boldsymbol{B}\leftarrow Z_p^{(k+1)\times k}$，输出公开参数：

$$\begin{aligned}\text{mpk}=\{&[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,\\&e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\}。\end{aligned}$$

初始化阶段1,2：$\text{Game}_{i,1}$中挑战者$\mathcal{B}$针对攻击者$\mathcal{A}$的第i次私钥询问的向量$\boldsymbol{y}'=(y_1',y_2',\cdots,y_n')$，随机选取向量$\boldsymbol{d}_j\leftarrow Z_p^{k+1}$，生成$\boldsymbol{y}'$对应的私钥：

$$\begin{aligned}\text{sk}_{\boldsymbol{y}'}:=(&\{K_{0,j}=[y_j'\boldsymbol{k}+\boldsymbol{V}_j'\boldsymbol{d}_j]_2,K_{1,j}=[\boldsymbol{d}_j]_2,\\&K_{2,j}=[\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]})。\end{aligned}$$

在$\text{Game}_{i,2}$中，挑战者随机选取$\boldsymbol{d}_j'\leftarrow Z_p^{k+1}$，生成$\boldsymbol{y}'$对应的私钥：

$$\begin{aligned}\text{sk}_{\boldsymbol{y}'}:=(&\{K_{0,j}=[y_j'(\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel})+\boldsymbol{V}_j'\boldsymbol{d}_j]_2,\\&K_{1,j}=[\boldsymbol{d}_j]_2,K_{2,j}=[\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]})。\end{aligned}$$

可以观察到两个Game的第i次私钥查询差别仅在于$\boldsymbol{k}_{0,j}'$这一分量，分析这两个分量：

$$\begin{aligned}[y_j'\boldsymbol{k}+\widetilde{\boldsymbol{V}_j}'\boldsymbol{d}_j]_2=[y_j'\boldsymbol{k}]_2\cdot[\boldsymbol{0}+\widetilde{\boldsymbol{V}_j}'\boldsymbol{d}_j]_2,\ [y_j'(\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel})\\+\widetilde{\boldsymbol{V}_j}'\boldsymbol{d}_j']_2=[y_j'\boldsymbol{k}]_2\cdot[y_j'\alpha\boldsymbol{a}_2^{\parallel}+\widetilde{\boldsymbol{V}_j}'\boldsymbol{d}_j']_2。\end{aligned}$$

由于随机值α以及随机向量$\boldsymbol{d}_j$,$\boldsymbol{d}_j'$的参与，$\boldsymbol{0}+\widetilde{\boldsymbol{V}_j}'\boldsymbol{d}_j$与$y_j'\alpha\boldsymbol{a}_2^{\parallel}+\widetilde{\boldsymbol{V}_j}'\boldsymbol{d}_j'$同分布，故从攻击者角度来看这两种私钥无法区分。

挑战阶段：由于这两个Game输出的都是熵扩张挑战密文，同分布。

由上分析得到$|\text{Adv}_{i,1}(\lambda)-\text{Adv}_{i,2}(\lambda)|=0$。　证毕

引理5　如果存在一个攻击者$\mathcal{A}$在$\text{Game}_{i,2}$和$\text{Game}_{i,3}$的优势满足$|\text{Adv}_{i,2}(\lambda)-\text{Adv}_{i,3}(\lambda)|>\varepsilon$，那么可以构造一个算法$\mathcal{B}_2$以不可忽略的优势解决$\text{MDDH}_{k,k+1}^n$问题，并且$\text{Time}(\mathcal{B}_2)\approx\text{Time}(\mathcal{A})$。

证明　引理5的证明与引理3相似，只是在私钥查询的时候：将第i次查询用的向量$\boldsymbol{k}$换成向量$\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}$。针对攻击者$\mathcal{A}$询问的向量$\boldsymbol{y}'=(y_1',y_2',\cdots,y_n')$，挑战者$\mathcal{B}_2$利用向量$\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}$和$\{[\boldsymbol{t}_j]_2\}_{j\in[n]}$生成私钥：

$$\begin{aligned}\text{sk}_{\boldsymbol{y}'}:=(&\{K_{0,j}=[y_j'(\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel})+\boldsymbol{V}_j'\boldsymbol{t}_j]_2,\\&K_{1,j}=[\boldsymbol{t}_j]_2,K_{2,j}=[\boldsymbol{U}_j'\boldsymbol{t}_j]_2\}_{j\in[n]}),\end{aligned}$$

发送给攻击者$\mathcal{A}$。

如果$[\boldsymbol{t}_1,\boldsymbol{t}_1,\cdots,\boldsymbol{t}_n]=\boldsymbol{Z}$，那么第$i$次询问的私钥是伪半功能私钥，上述Game对应的是$\text{Game}_{i,2}$，如果$[\boldsymbol{t}_1,\boldsymbol{t}_1,\cdots,\boldsymbol{t}_n]=\mathbf{BS}$,那么第$i$次询问的私钥是半功能私钥,对应的是$\text{Game}_{i,3}$。所以如果攻击者$\mathcal{A}$以不可忽视的优势区分$\text{Game}_{i,2}$和$\text{Game}_{i,3}$，即$|\text{Adv}_{i,2}(\lambda)-\text{Adv}_{i,3}(\lambda)|>\varepsilon$，那么挑战者同样以不可忽略的优势区分$[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n]=\mathbf{BS}$和$[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n]=\boldsymbol{Z}$。　证毕

引理6　由表1，可以得到$\text{Game}_i\equiv\text{Game}_{i-1,3}$。

引理7　对于任意一个攻击者$\mathcal{A}$，在Game_{Q+1}和$\text{Game}_{\text{Final}}$的攻击优势满足$|\text{Adv}_{Q+1}(\lambda)-\text{Adv}_{\text{Final}}(\lambda)|=0$。

证明　这两个Game的差别在于Game_{Q+1}挑战密文是熵扩张密文，$\text{Game}_{\text{Final}}$挑战密文是熵扩张加密的随机数。下面来说明这两个Game不可区分。

挑战者$\mathcal{B}_3$随机选取$\boldsymbol{A}_1,\boldsymbol{a}_2\leftarrow Z_p^{(2k+1)\times(k+1)}\times Z_p^{2k+1}$，$\boldsymbol{B}\leftarrow Z_p^{(k+1)\times k}$，确定$\boldsymbol{a}_2^{\parallel}$，随机选取$\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1\leftarrow Z_p^{(2k+1)\times(k+1)}$，$\boldsymbol{V}_j,\boldsymbol{U}_j\leftarrow\text{span}^{k+1}(\boldsymbol{a}_2^{\parallel})$，$\alpha\leftarrow Z_p$。

初始化阶段：挑战者模拟方案，在Z_p^{2k+1}随机选取向量$\boldsymbol{k}'$(在这里$\boldsymbol{k}'$为生成半功能密钥时所用到的向量)，设$\boldsymbol{k}=\boldsymbol{k}'-\alpha\boldsymbol{a}_2^{\parallel}$，由于$\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{a}_2^{\parallel}=\boldsymbol{0}$，那么$e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)=e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)$，输出公开参数mpk给$\mathcal{A}$：

$$\begin{aligned}\text{mpk}=\{&[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,\\&e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)\}。\end{aligned}$$

查询阶段1：挑战者$\mathcal{B}_3$针对攻击者$\mathcal{A}$申请访问的向量$\boldsymbol{y}'=(y_1',y_2',\cdots,y_n')$，随机选取$\boldsymbol{d}_j\leftarrow\text{span}(\boldsymbol{B})$，生成半功能私钥：

$$\begin{aligned}\text{sk}_{\boldsymbol{y}'}:=(&\{K_{0,j}=[y_j'\boldsymbol{k}+\boldsymbol{V}_j'\boldsymbol{d}_j]_2,K_{1,j}=[\boldsymbol{d}_j]_2,\\&K_{2,j}=[\boldsymbol{U}_j'\boldsymbol{d}_j]_2\}_{j\in[n]})。\end{aligned}$$

挑战阶段：攻击者$\mathcal{A}$选择两个等长的消息m_0和m_1，以及挑战向量$\boldsymbol{x}^*=(x_1^*\cdots x_n^*)$ (任何询问向量$\boldsymbol{y}'$与挑战向量$\boldsymbol{x}^*$都不满足$(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{y}'=\boldsymbol{1}$)发送给挑战者$\mathcal{B}_3$，挑战者$\mathcal{B}_3$随机选取$b\in\{0,1\}$，向量$\boldsymbol{c},\boldsymbol{c}_j\leftarrow\text{span}(\boldsymbol{A}_1,\boldsymbol{a}_2)$，生成熵扩张挑战密文：

$$\text{ct}_{\boldsymbol{x}^*}=\begin{Bmatrix}C_0=[\boldsymbol{c}^{\mathrm{T}}]_1,\{C_{1,j}=[x_j{}^*\boldsymbol{c}^{\mathrm{T}}\boldsymbol{V}_j'+x_j{}^*\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{U}_j']_1,\\C_{2,j}=[x_j{}^*\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in[n]}\\C'=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\cdot m_b=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)\\\cdot e([\boldsymbol{c}^{\mathrm{T}}]_1,[\alpha\boldsymbol{a}_2]_2)^{-1}\cdot m_b\end{Bmatrix}。$$

查询阶段2：与查询阶段1相同。

猜测阶段：攻击者$\mathcal{A}$给出b的猜测b'。

在熵扩张挑战密文中，$e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)\cdot e([\boldsymbol{c}^{\mathrm{T}}]_1,[\alpha\boldsymbol{a}_2]_2)^{-1}=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)\cdot e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{a}_2^{\parallel}]_2)^{-\alpha}$，由于随机数$\alpha$的参与，$e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{a}_2^{\parallel}]_2)^{-\alpha}$在$G_T$中的值是均匀分布，这表明在加密随机数得到的密文与加密消息得到的密文同分布。所以从攻击者$\mathcal{A}$角度来说熵扩张密文与熵扩张加密随机数不可区分。

由以上分析得到$|\mathrm{Adv}_{Q+1}(\lambda)-\mathrm{Adv}_{\mathrm{Final}}(\lambda)|=0$。

证毕

定理1 在素数阶熵扩张引理和$\mathrm{MDDH}^n_{k,k+1}$困难假设成立的条件下，本文提出的非零内积加密方案是适应性安全的，并且

$$\max\{\mathrm{Time}(\mathcal{B}_0),\mathrm{Time}(\mathcal{B}_1),\mathrm{Time}(\mathcal{B}_2)\}\approx\mathrm{Time}(\mathcal{A})。$$

证明 在适应性安全模型下，攻击者$\mathcal{A}$对本文给出的非零内积方案的攻击优势就是对Game_0的攻击优势，由安全性证明中给出的Game序列之间的关系，可得

$$\begin{aligned}\mathrm{Adv}_0(\lambda)=&\mathrm{Adv}_0(\lambda)-\mathrm{Adv}_{0'}(\lambda)+\mathrm{Adv}_{0'}(\lambda)\\&-\mathrm{Adv}_1(\lambda)+\cdots+\mathrm{Adv}_Q(\lambda)-\mathrm{Adv}_{Q+1}(\lambda)\\&+\mathrm{Adv}_{Q+1}(\lambda)-\mathrm{Adv}_{\mathrm{Final}}(\lambda)+\mathrm{Adv}_{\mathrm{Final}}(\lambda)\\\le&|\mathrm{Adv}_0(\lambda)-\mathrm{Adv}_{0'}(\lambda)|+|\mathrm{Adv}_{0'}(\lambda)\\&-\mathrm{Adv}_1(\lambda)|+\cdots+|\mathrm{Adv}_{Q+1}(\lambda)\\&-\mathrm{Adv}_{\mathrm{Final}}(\lambda)|+\mathrm{Adv}_{\mathrm{Final}}(\lambda)\end{aligned}$$

由引理1知$|\mathrm{Adv}_0(\lambda)-\mathrm{Adv}_{0'}(\lambda)|\le\varepsilon$。

由引理2知$|\mathrm{Adv}_{0'}(\lambda)-\mathrm{Adv}_1(\lambda)|=0$。

Game_i到Game_{i+1}的逼近可表示为

$$\begin{aligned}\mathrm{Game}_i-\mathrm{Game}_{i+1}=&(\mathrm{Game}_i-\mathrm{Game}_{i,1})\\&+(\mathrm{Game}_{i,1}-\mathrm{Game}_{i,2})\\&+(\mathrm{Game}_{i,2}-\mathrm{Game}_{i,3})\\&+(\mathrm{Game}_{i,3}-\mathrm{Game}_{i+1})。\end{aligned}$$

由引理3, 4, 5, 6知

$$\begin{aligned}|\mathrm{Adv}_i(\lambda)-\mathrm{Adv}_{i+1}(\lambda)|\le&|\mathrm{Adv}_i(\lambda)-\mathrm{Adv}_{i,1}(\lambda)|\\&+|\mathrm{Adv}_{i,1}(\lambda)-\mathrm{Adv}_{i,2}(\lambda)|\\&+|\mathrm{Adv}_{i,2}(\lambda)-\mathrm{Adv}_{i,3}(\lambda)|\\&+|\mathrm{Adv}_{i,3}(\lambda)-\mathrm{Adv}_{i+1}(\lambda)|\\\le&2\varepsilon\end{aligned}$$

由引理7知$|\mathrm{Adv}_{Q+1}(\lambda)-\mathrm{Adv}_{\mathrm{Final}}(\lambda)|=0$，攻击者$\mathcal{A}$在$\mathrm{Game}_{\mathrm{Final}}$中的优势$\mathrm{Adv}_{\mathrm{Final}}(\lambda)=0$。

综上分析，得到攻击者$\mathcal{A}$在Game_0中的攻击优势：

$$\mathrm{Adv}_0(\lambda)\le(2Q+1)\varepsilon$$

在素数阶熵扩张引理和$\mathrm{MDDH}^n_{k,k+1}$困难假设成立的条件下可知ε可以忽略，所以攻击者$\mathcal{A}$在Game_0中的攻击优势可以忽略，则本方案是适应性安全的。

证毕

5 性能对比

本文设计了一个公开参数规模较小且适应安全的内积加密方案，本文与文献[5,7–10]的方案相比，公开参数长度固定，且规模最小，但付出的代价是增加了私钥的长度。与文献[10]相比，降低了密文长度。下面给出当$k=1$时，本方案与现有内积加密方案的数据长度对比，见表2。

表 2 与现有内积加密方案的数据长度比较

方案	公开参数长度	私钥长度	密文长度	安全性假设	安全性
文献[5]	$(4n^2+3)\|G_1\|$	$(2n+1)\|G_1\|$	$(2n+1)\|G_1\|$	2 variants of GSD	选择安全
文献[7]	$(4n^2+2n)\|G_1\|$	$(2n+3)\|G_1\|$	$(2n+3)\|G_1\|+\|G_T\|$	n-eDDH	适应安全
文献[8]	$(4n^2+3)\|G_1\|$	$(3n+2)\|G_1\|$	$(3n+2)\|G_1\|+\|G_T\|$	DLIN	适应安全
文献[9](type1)	$105\|G_1\|$	$(3n+2)\|G_1\|$	$(3n+2)\|G_1\|+\|G_T\|$	DLIN	适应安全
文献[10]	$28\|G_1\|$	$7n\|G_2\|+\alpha$	$7n\|G_1\|$	SXDH	适应安全
本方案	$9\|G_1\|+G_T$	$8n\|G_2\|$	$(5n+3)\|G_1\|+\|G_T\|$	$\mathrm{MDDH}^n_{k,k+1}$	适应安全

注：其中n表示系统属性的个数，$|G_1|,|G_2|,|G_T|$分别表示G_1,G_2,G_T中群元素的长度。

6 结束语

本文利用DPVS技术提出了一个内积加密方案，该方案的公开参数规模较小，并且在素数阶熵扩张引理和$\mathrm{MDDH}^n_{k,k+1}$困难假设下证明了方案是适应安全的。如何设计公开参数规模小，并且密钥和密文长度都较短的适应安全的内积加密方案是我们下一步的研究目标。

参 考 文 献

[1] BALTICO C E Z, CATALANO D, and FIORE D. Practical functional encryption for quadratic functions with applications to predicate encryption[C]. The 37th Annual International Cryptology Conference on Advances in Cryptology, Santa Barbara, USA, 2017: 67–100.

[2] BONEH D, SAHAI A, and WATERS B. Functional encryption: Definitions and challenges[C]. The 8th conference on Theory of Cryptography, Providence, USA, 2011: 253–273.

[3] 曹丹, 王小峰, 王飞, 等. SA-IBE: 一种安全可追责的基于身份加密方案[J]. 电子与信息学报, 2011, 33(12): 2922–2928.
CAO Dan, WANG Xiaofeng, WANG Fei, *et al*. SA-IBE: A secure and accountable identity-based encryption scheme[J].

Journal of Electronics & Information Technology, 2011, 33(12): 2922–2928.

[4] BONEH D and WATERS B. Conjunctive, subset, and range queries on encrypted data[C]. The 4th conference on Theory of Cryptography. Amsterdam, Netherlands, 2007: 535–554.

[5] KATZ J, SAHAI A, and WATERS B. Predicate encryption supporting disjunctions, polynomial equations, and inner products[C]. The 27th Annual International Conference on Advances in Cryptology, Istanbul, Turkey: 2008: 146–162.

[6] DATTA P, OKAMOTO T, and TAKASHIMA K. Adaptively simulation-secure attribute-hiding predicate encryption[C]. The 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, Australia, 2018: 640–672.

[7] LEWKO A, OKAMOTO T, and SAHAI A. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption[C]. The 29th Annual International Conference on Theory and Applications of Cryptographic Techniques, French Riviera, 2010: 62–91.

[8] OKAMOTO T and TAKASHIMA K. Fully secure functional encryption with general relations from the decisional linear assumption[C]. The 30th Annual Conference on Advances in Cryptology, Santa Barbara, USA, 2010: 191–208.

[9] OKAMOTO T and TAKASHIMA K. Fully secure unbounded inner-product and attribute-based encryption[C]. The 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, 2012: 349–366.

[10] TOMIDA J and TAKASHIMA K. Unbounded inner product functional encryption from bilinear maps[C]. The 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, Australia, 2018: 609–639.

[11] WATERS B. Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions[C]. The 29th Annual International Cryptology Conference on Advances in Cryptology, Santa Barbara, USA, 2009: 619–636.

[12] CHEN Jie, GAY R, and WEE H. Improved dual system ABE in prime-order groups via predicate encodings[C]. The 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 2015: 595–624.

[13] CHEN Jie, GONG Junqing, KOWALCZYK L, *et al*. Unbounded ABE via bilinear entropy expansion, revisited[C]. The 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, 2018: 503–534.

[14] WEE H. Dual system encryption via predicate encodings[C]. The 11th Theory of Cryptography Conference, San Diego, USA, 2014: 616–637.

[15] LEWKO A B and WATERS B. New techniques for dual system encryption and fully secure HIBE with short ciphertexts[C]. The 7th International Conference on Theory of Cryptography, Zurich, Switzerland, 2010: 455–479.

高海英：女，1978年生，教授，主要研究方向是密码算法设计与分析.

魏　铎：男，1994年生，硕士生，主要研究方向是公钥密码.

责任编辑：陈　倩