一种支持算术张成程序的密文策略属性加密方案

魏 铎,高海英
(战略支援部队信息工程大学,河南郑州 450001)

摘 要: 密文策略属性加密方案适用于云环境中密文数据的访问控制. 已有的支持算术张成程序的属性加密方案多是密钥访问策略的方案,且公开参数规模较大. 本文利用双对偶向量空间(Dual Pair Vector Space,DPVS)技术,提出了一个公开参数长度固定、支持算术张成程序的密文策略属性加密方案. 在新方案中,将密文相关的访问控制向量与随机矩阵结合,密钥相关的属性分量与熵扩张引理中的公开参量结合,设计方法对应了熵扩张引理中给出的密文和密钥分量的形式. 最后,基于素数阶双线性熵扩张引理和 k-Lin 困难假设,证明了该方案具有适应安全性. 新方案与已有支持算术张成程序的属性加密方案相比,实现了密文访问策略、公开参数长度固定且满足适应安全性.

关键词: 算术张成程序; 密文策略属性加密; 双线性熵扩张; k-Lin 假设; 适应安全性
中图分类号: TN918.1 **文献标识码:** A **文章编号:** 0372-2112 (2020)10-1993-10
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.10.017

A Ciphertext-Policy Attribute-Based Encryption Scheme Supporting Arithmetic Span Program

WEI Duo,GAO Hai-ying
(*PLA SSF Information Engineering University*,*Zhengzhou*,*Henan* 450001,*China*)

Abstract: Ciphertext-policy attribute-based encryption scheme is suitable for access control of ciphertext data in cloud environment. Most of the existing attribute-based encryption schemes supporting arithmetic span programs are key-policy schemes with large public parameter scale. By exploiting the double Dual Pair Vector Space (DPVS) technique, a ciphertext-policy attribute-based encryption scheme which has a fixed length of public parameters and supports arithmetic span programs is proposed in this paper. In this scheme, the ciphertext-related access control vector is combined with the random matrix, and the key-related attribute components are combined with the public parameters in the entropy expansion Lemma. The method is designed according to the form of ciphertext and key components given in the entropy expansion Lemma. Finally, the adaptive security of the scheme is proved based on the prime order bilinear entropy extension Lemma and k-Lin difficult assumption. Compared with the existing attribute-based encryption schemes which support arithmetic span programs, the new scheme has the advantages of ciphertext access policy, fixed length of public parameters and adaptive security.

Key words: arithmetic span program; ciphertext-policy attribute-based encryption; bilinear entropy expansion; k-linear assumption; adaptively secure

1 引言

属性加密(Attribute-Based Encryption,ABE)这一概念最早是由 Sahai 和 Waters[1] 等人在 2005 年欧洲密码年会上提出的. 它是一种新型的公钥密码,实现一对多的保密通信,支持加密数据的细粒度访问控制. 根据访问控制策略位置的不同,属性加密可分为两类,一类是密钥策略属性加密方案(Key-Policy Attributed-Based Encryption,KP-ABE),即密文与属性相关联,访问控制策略部署在密钥中;另一类是密文策略属性加密方案[2](Ciphertext-Policy Attributed-Based Encryption, CP-ABE),即访问控制策略部署在密文当中,私钥包含了属性的信息. 在 CP-ABE 方案中,由加密者来制定访问策略,因此,与 KP-ABE 方案相比,CP-ABE 方案更适用于

收稿日期:2019-06-10;修回日期:2020-04-12;责任编辑:孙瑶
基金项目:国家自然科学基金(No.61702548,No.61601515);河南省基础与前沿课题(No.162300410192)

云环境下的密文访问控制.

在属性加密方案中,通过访问结构实现对密文的细粒度访问控制.目前,已提出的属性加密方案中支持的访问结构主要包括:树状访问控制结构[2];基于 LSSS 矩阵的单调访问结构[3]、非单调访问结构[4]、电路访问结构[5]等.不同的应用场景,可选取支持相应访问控制结构的属性加密方案.并且目前设计的属性基加密方案衍生出了许多新的功能,例如具有分层功能的方案[6]、具有可撤销功能的方案[7,8]以及具有多授权机构的方案[9]等.属性加密方案的安全性分为选择安全和适应安全两种.所谓选择安全性,即在方案的安全性证明过程中,攻击者必须首先公开攻击目标;而在适应安全性的证明过程中,攻击者不用在初始阶段公开攻击目标.适应安全模型比选择安全模型的安全强度高,同样,适应安全的属性加密方案的设计要比选择安全的属性加密方案的设计难度要大,并且方案的实现效率普遍较低.

算术张成程序这个概念是由 Karchmer[10]等人首次提出.但在后续研究中,基于算术张成程序的属性加密方案很少被提出,在 2012 年,Ishai[11]等人给出了一个支持算术张成程序的 KP-ABE 方案,该方案的公开参数个数随系统属性个数线性增长,即公开参数长度不固定,该方案只具有选择安全性;2015 年,Attrapadung[12]等人提出了一种一般性转换技术,在不限制分支程序大小但属性集大小受限的情况下,可将支持 span program 访问结构的 KP-ABE 方案转化为基于算术张成程序的 KP-ABE 方案,方案安全性虽然达到了适应安全,但公开参数长度不固定.在 2018 年的欧密会上,Chen[13]等人为了解决 ABE 方案参数规模较大的问题,提出了双线性映射熵扩张引理,该引理为设计参数规模较小的 ABE 方案提供了方法指导,作者在文献[13]中还首次提出了公开参数长度固定的算术张成程序 KP-ABE 方案,该方案具有适应安全性.从研究现状来看,如何设计一个支持算术张成程序、公开参数长度固定且适应安全的 CP-ABE 方案是一个待解决的问题,本文的研究目的就是为了解决这个问题.

为了设计具有适应安全性的 ABE 方案,普遍采用的数学工具一个是合数阶双线性群[14],从安全性方面考虑,基于此技术设计的方案要求群的阶数比较大,导致设计出的方案效率不高.相较于合数阶双线性群,素数阶双线性群在对运算方面有着更高的效率,并且目前衍生出来一些在素数阶双线性群上去模拟合数阶群性质的构造方法[14~17],其中最典型的是文献[16]提到的双对偶向量空间技术(Dual Paring Vector Space, DPVS).DPVS 在理论上被用于替代合数阶双线性群,其可以在保证同等安全性的条件下,在牺牲存储复杂度的条件下使双线性映射运算时间较短.我们从计算效率的角度考虑,决定基于 DPVS 设计支持算术张成程序的适应安全的 ABE 方案.下一步要考虑的是:基于 DPVS 的支持算术张成程序的 CP-ABE 方案该怎么设计?

文献[14]提出了一种利用素数阶双线性群模拟合数阶双线性群的新方法,并且提出了构造 ABE 方案的一般框架结构,具体地说,给出了加密算法中用到的 sE 编码、密钥生成算法中 kE 编码、rE 编码,解密算法中的 rD、sD 编码的抽象描述,这些编码算法必须满足某种性质,这种性质保证了解密正确性,并且保证方案具有适应安全性,文中还给出了一些 ABE 方案实例.基于该框架结构设计得到的 ABE 方案都是基于 k-Lin 假设下适应安全性的,这为我们设计支持算术张成程序的适应安全的 CP-ABE 方案提供了技术基础和方法指导.在该阶段的研究中,我们需要重点解决的问题是设计满足特定性质的 sE、kE、rE 、rD、sD 编码方案,该阶段的技术难点是 sE 编码方案中算术张成程序访问控制结构对应的向量在密文中的嵌入方法,属性向量在 kE 和 rE 编码方案中的嵌入方法.具体解决方法不再赘述.在完成该阶段工作的基础上,我们下一步要考虑的是:如何降低该方案的公开参数规模?

我们通过前期调研了解到,文献[13]对文献[14]中的 DPVS 技术进行了改进,并且给出了双线性映射熵扩张引理,该引理可以用来降低公开参数规模.文中给出的熵扩张引理有两个,分别是针对合数阶双线性群的熵扩张引理和针对素数阶双线性群的熵扩张引理,针对素数阶双线性群的熵扩张引理具体描述见本文 2.3 节,熵扩张引理给出了一系列的公开参数、多个密文分量的具体形式、多个密钥分量的具体形式,但并不是密文和密钥的所有分量,也不涉及到具体的访问控制结构和属性集合.该阶段我们重点研究如何调整在第一阶段已设计出的方案的公开参数、密文和密钥中的编码算法,使得产生的公开参数、密文分量、明文分量符合熵扩张引理中给出的形式.最终我们将访问控制向量与随机矩阵巧妙结合,生成一个可嵌入到双线性熵扩张引理框架中的向量,从而完成密文的生成;在方案的私钥生成算法,通过将用户向量中的每一分量与双线性熵扩张引理框架中的部分向量结合,生成用户的私钥.采用的设计技巧保证了解密正确性,并且基于素数阶双线性熵扩张引理和 k-Lin 假设证明了该方案具有适应安全性.

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.

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

$\mathrm{span}^{k+1}(\boldsymbol{A})$:表示矩阵 $\boldsymbol{A}$ 列向量张成的空间中的 $k+1$ 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.

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

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

2.2　双线性映射

定义 1[18]　合数阶双线性映射. 令$\mathbb{G}=(G_N,H_N,G_T,e)$,$G_N,H_N,G_T$ 都是阶为 $N=p_1p_2p_3$ 的循环群,$e:G_N\times H_N\to G_T$;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. 对于映射 $e:(G_N,H_N)\to G_T$ 称其为合数阶群双线性映射,如果其满足以下三条性质:

(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]中给出的在素数阶群上模拟合数阶群的性质的 DPVS 技术如下所述.

随机选取 $\boldsymbol{A}_1\leftarrow\boldsymbol{Z}_p^{l\times l_1}$,$\boldsymbol{A}_2\leftarrow\boldsymbol{Z}_p^{l\times l_2}$,$\boldsymbol{A}_3\leftarrow\boldsymbol{Z}_p^{l\times l_3}$,并定义$(\boldsymbol{A}_1^{\parallel}\|\boldsymbol{A}_2^{\parallel}\|\boldsymbol{A}_3^{\parallel})^{\mathrm{T}}$,使得 $\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_i^{\parallel}=\boldsymbol{I}$,$(\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_j^{\parallel}=\boldsymbol{0})_{i\neq j}$. 对于合数阶双线性群与素数阶群 G_1 和 G_2 有如下对应关系:

令 $u_i\to[\boldsymbol{A}_i]_1$,$h_i\to[\boldsymbol{A}_i^{\parallel}]_2$,$u_i^s\to[\boldsymbol{A}_i\boldsymbol{s}]_1$,

$w\in\boldsymbol{Z}_N\to\boldsymbol{W}\in\boldsymbol{Z}_p^{l\times l_w}$,$u_i^w\to[\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{W}]_1$.

并定义如下运算法则:

$\forall\boldsymbol{A}\in\boldsymbol{Z}_p^{m\times n}$,$\boldsymbol{B}\in\boldsymbol{Z}_p^{n\times t}$,

令 $e([\boldsymbol{A}]_1,[\boldsymbol{B}]_2)=e(g_1,g_2)^{AB}$.

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

定义 2[10]　算术张成程序(arithmetic span program). 一个算术张成程序$(\boldsymbol{v},\rho)$包含一个向量集合 $\boldsymbol{v}=\{(y_j,z_j);j\in[l],\boldsymbol{y}_j,z_j\in\boldsymbol{Z}_p^{l'}\}$和一个映射 $\rho:[l]\to[n]$. 当向量 $\boldsymbol{x}(\boldsymbol{x}\in\boldsymbol{Z}_p^n)$ 满足$(\boldsymbol{v},\rho)$时,存在常数 $\omega_1,\cdots,\omega_l\in\boldsymbol{Z}_p$ 使得

$$\sum_{j=1}^{l}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_{\rho(j)}z_j)=1$$

这里 $\boldsymbol{1}:=(1,0,\cdots,0)\in\boldsymbol{Z}_p^{l'}$.

与文献[13]一样,在这里需要提出一个限制,即 ρ 是一个恒等映射,并且 $l=n$.

2.3　困难假设

定义 3[14]　k-Lin 假设:定义分布 $\mathcal{G}=(p,G_1,G_2,G_T,g_1,g_2,e)\leftarrow\zeta(1^\lambda)$;$\boldsymbol{B}\leftarrow\boldsymbol{Z}_p^{(k+1)\times k}$;$\boldsymbol{s}\leftarrow\boldsymbol{Z}_p^{k}$;$z\leftarrow\boldsymbol{Z}_p^{k+1}$,$\mathcal{D}=(\mathcal{G},[\boldsymbol{B}])$,对于攻击者 $\mathcal{B}$ 在任意多项式时间内区分 $[\boldsymbol{Bs}]$和$[z]$的优势是可以忽略的,其中$[\boldsymbol{Bs}]$表示$[\boldsymbol{Bs}]_1$或$[\boldsymbol{Bs}]_2$,$\mathrm{Adv}_{\mathcal{B}}^{\text{k-Lin}}=|\Pr[\mathcal{B}(\mathcal{D},[\boldsymbol{Bs}])=1]-\Pr[\mathcal{B}(\mathcal{D},[z])=1]|$,$[z]$表示$[z]_1$或$[z]_2$.

定义 4[13]　素数阶双线性熵扩张引理(prime-order entropy expansion lemma). 随机选取$(\boldsymbol{A}_1,\boldsymbol{a}_2,\boldsymbol{A}_3)\leftarrow\boldsymbol{Z}_p^{3k\times k}\times\boldsymbol{Z}_p^{3k}\times\boldsymbol{Z}_p^{3k\times k}$,并且取$(\boldsymbol{A}_1,\boldsymbol{a}_2,\boldsymbol{A}_3)$的逆$(\boldsymbol{A}_1^{\parallel},\boldsymbol{a}_2^{\parallel},\boldsymbol{A}_3^{\parallel})^{\mathrm{T}}$,选取 $\boldsymbol{B}\leftarrow\boldsymbol{Z}_p^{(k+1)\times k}$,对于攻击者 $\mathcal{B}$ 在任意多项式时间内区分以下两个分布的优势是可以忽略的.

$$\left\{\begin{array}{l}aux:[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1']_1\\ ct:[\boldsymbol{c}_0=\boldsymbol{c}^{\mathrm{T}}]_1,\{[\boldsymbol{c}_{0,j}=\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{c}_{1,j}=\boldsymbol{c}_j^{\mathrm{T}}]_1,[\boldsymbol{c}_{2,j}=\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)]_1,[\boldsymbol{c}'_{0,j}=\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{c}'_{2,j}=\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')]_1\}_{j\in[n]}\\ sk:[\boldsymbol{K}_1=\boldsymbol{D}]_2,\{[\boldsymbol{K}_{1,j}=\boldsymbol{WD}+(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)\boldsymbol{D}_j]_2,[\boldsymbol{K}_{2,j}=\boldsymbol{D}_j]_2,[\boldsymbol{K}'_{1,j}=\boldsymbol{W}'\boldsymbol{D}+(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')\boldsymbol{D}_j']_2,[\boldsymbol{K}'_{2,j}=\boldsymbol{D}_j']_2\}_{j\in[n]}\end{array}\right\}$$

$$\approx_c\left\{\begin{array}{l}aux:[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1']_1\\ ct:[\boldsymbol{c}_0=\boldsymbol{c}^{\mathrm{T}}]_1,\left\{\begin{array}{l}[\boldsymbol{c}_{0,j}=\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}+\boldsymbol{V}_j)]_1,[\boldsymbol{c}_{1,j}=\boldsymbol{c}_j^{\mathrm{T}}],[\boldsymbol{c}_{2,j}=\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_j)]_1,\\ [\boldsymbol{c}'_{0,j}=\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}'+\boldsymbol{V}_j')]_1,[\boldsymbol{c}'_{2,j}=\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1'+\boldsymbol{U}_j')]_1\end{array}\right\}_{j\in[n]}\\ sk:[\boldsymbol{K}_1=\boldsymbol{D}]_2,\left\{\begin{array}{l}[\boldsymbol{K}_{1,j}=(\boldsymbol{W}+\boldsymbol{V}_j)\boldsymbol{D}+(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_j)\boldsymbol{D}_j]_2,[\boldsymbol{K}_{2,j}=\boldsymbol{D}_j]_2,\\ [\boldsymbol{K}'_{1,j}=(\boldsymbol{W}'+\boldsymbol{V}_j')\boldsymbol{D}+(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1'+\boldsymbol{U}_j')\boldsymbol{D}_j']_2,[\boldsymbol{K}'_{2,j}=\boldsymbol{D}_j']_2\end{array}\right\}_{j\in[n]}\end{array}\right\}$$

其中 $\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1,\boldsymbol{W}',\boldsymbol{W}_0',\boldsymbol{W}_1\leftarrow\boldsymbol{Z}_p^{3k\times(k+1)}$,$\boldsymbol{V}_j,\boldsymbol{U}_j,\boldsymbol{V}_j',\boldsymbol{U}_j'\leftarrow\mathrm{span}^{k+1}(\boldsymbol{a}_2^{\parallel})$,$\boldsymbol{D},\boldsymbol{D}_j,\boldsymbol{D}_j'\leftarrow\mathrm{span}^{k+1}(\boldsymbol{B})$,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　支持算术张成程序的 CP-ABE 方案形式化定义

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

Enc(mpk, $\boldsymbol{v}$, m)：该概率加密算法输入公共参数 mpk，访问策略 $\boldsymbol{v}=\{\boldsymbol{y}_j,\boldsymbol{z}_j)_{j\in[n]}$ 和明文 m，输出密文 $ct_{\boldsymbol{v}}$.

KeyGen(mpk, msk, $\boldsymbol{x}$)：该概率密钥生成算法输入主密钥 msk 和向量 $\boldsymbol{x}\in\boldsymbol{Z}_p^n$，输出私钥 sk_x.

Dec(mpk, $sk_{\boldsymbol{x}}$, $ct_{\boldsymbol{v}}$)：该确定性解密算法输入 $sk_{\boldsymbol{x}}$ 和 $ct_{\boldsymbol{v}}$，如果 $\boldsymbol{x}$ 与 $\boldsymbol{v}$ 满足 $\sum_{j\in[n]}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_j\cdot\boldsymbol{z}_j)=\boldsymbol{1}$，则可以解密密文.

2.5 算术张成程序 CP-ABE 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算术张成程序 CP-ABE 方案的适应性安全模型.

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

Phase1：攻击者 $\mathcal{A}$ 自行选择 $\boldsymbol{x}'$ 进行多项式次私钥查询. 挑战者 $\mathcal{B}$ 运行 KeyGen 算法，将生成的私钥发送给 $\mathcal{A}$.

Challenge：攻击者 $\mathcal{A}$ 向挑战者提交两个等长的明文 m_0 和 m_1，以及要挑战的访问结构 $\boldsymbol{v}^*=\{(\boldsymbol{y}_j,\boldsymbol{z}_j)j\in[n],\boldsymbol{y}_j,\boldsymbol{z}_j\in\boldsymbol{Z}_p^{l'}\}$（任何询问向量 $\boldsymbol{x}'$ 与要挑战访问结构 $\boldsymbol{v}^*=\{(\boldsymbol{y}_j,\boldsymbol{z}_j):j\in[n],\boldsymbol{y}_j,\boldsymbol{z}_j\in\boldsymbol{Z}_p^{l'}\}$ 都不满足 $\sum_{j=1}^{n}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_j\cdot\boldsymbol{z}_j)=\boldsymbol{1}$）发送给挑战者 $\mathcal{B}$，$\mathcal{B}$ 随机选取 $b\in\{0,1\}$，计算 $ct_{\boldsymbol{v}^*}=\mathrm{Enc}(mpk,\boldsymbol{v}^*,m_b)$，并将 $ct_{\boldsymbol{v}^*}$ 作为挑战密文返回给攻击者 $\mathcal{A}$.

Phase2：与 Phase1 相同.

Guess：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 $\mathrm{Adv}_A(\lambda)=|\Pr(b'=b)-1/2|$.

定义 5 如果对于任意多项式时间的攻击者，赢得上述游戏的优势都是可忽略的，则称该支持算数分支程序 CP-ABE 加密方案是适应性安全的.

3 方案描述

Setup(1^λ, 1^n)：系统初始化阶段输入安全参数 λ 和系统属性个数 n. 选取

$$\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1,\boldsymbol{W}',\boldsymbol{W}_0',\boldsymbol{W}_1',\boldsymbol{U}_0\leftarrow\boldsymbol{Z}_p^{3k\times(k+1)},$$
$$\boldsymbol{A}_1\leftarrow\boldsymbol{Z}_p^{3k\times k},\boldsymbol{B}\leftarrow\boldsymbol{Z}_p^{(k+1)\times k},\boldsymbol{k}\leftarrow\boldsymbol{Z}_p^{3k}$$

生成公共参数 mpk 和系统主密钥 msk：

$$mpk=\{[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{U}_0]_1,e([\boldsymbol{A}_1^{\mathrm{T}}],[\boldsymbol{k}]_2)\}$$
$$msk=(\boldsymbol{k},\boldsymbol{B},\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1,\boldsymbol{W}',\boldsymbol{W}_0',\boldsymbol{W}_1',\boldsymbol{U}_0)$$

Enc(mpk, $\boldsymbol{v}$, m)：加密阶段输入访问结构 $\boldsymbol{v}=\{\boldsymbol{y}_j,\boldsymbol{z}_j\}_{j\in[n]}$ 和编码为 G_T 上的消息 m，选取 $\boldsymbol{c},\boldsymbol{c}_j\leftarrow\mathrm{span}(\boldsymbol{A}_1)$，$\boldsymbol{U}\leftarrow\boldsymbol{Z}_p^{(l'-1)\times(k+1)}$，计算生成密文 $ct_{\boldsymbol{v}}$ 为

$$ct_{\boldsymbol{v}}=\left\{\begin{array}{l}C_0=[\boldsymbol{c}^{\mathrm{T}}]_1,\\ \left\{\begin{array}{l}C_{0,j}=[\boldsymbol{y}_j\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{W}]_1,\\ C'_{0,j}=[\boldsymbol{z}_j\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{W}']_1\end{array}\right\}_{j\in[n]}\\ \{C_{1,j}=[\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in|n|}\\ \left\{\begin{array}{l}C_{2,j}=[\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)]_1,\\ C'_{2,j}=[\boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')\boldsymbol{d}_j']_1\end{array}\right\}_{j\in[n]},\\ C'=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\cdot m\end{array}\right\}$$

KeyGen(mpk, msk, $\boldsymbol{x}$)：密钥生成阶段输入为主密钥 msk 和向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in\boldsymbol{Z}_p^n$，选取 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}_j'\leftarrow\mathrm{span}(\boldsymbol{B})$，生成 $\boldsymbol{x}$ 对应的私钥为

$$sk_x:\left(\begin{array}{l}K_0=[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{d}]_2\\ \left\{\begin{array}{l}K_{1,j}=[(\boldsymbol{W}+x_j\cdot\boldsymbol{W}')\boldsymbol{d}\\ \quad+(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)\boldsymbol{d}_j\\ \quad+x_j\cdot(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')\boldsymbol{d}_j']_2\end{array}\right\}_{j\in[n]}\\ \{\boldsymbol{K}_{2,j}=[\boldsymbol{d}_j]_2,K'_{2,j}=[\boldsymbol{d}_j']_2\}_{j\in[n]}\end{array}\right)$$

Dec(mpk, $sk_{\boldsymbol{x}}$, $ct_{\boldsymbol{v}}$)：算法以密钥 $sk_{\boldsymbol{x}}$ 和密文 $ct_{\boldsymbol{v}}$ 作为输入. 如果 $\boldsymbol{x}$ 与 $\boldsymbol{v}$ 满足 $\sum_{j\in[n]}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_j\cdot\boldsymbol{z}_j)=\boldsymbol{1}$，那么解密者就可以正确解密. 其解密过程如下：

$$C=\frac{e(C_0,K_0)}{\prod_{j=1}^{n}\left(\begin{array}{l}e(C_{0,j}\cdot(C'_{0,j})^{x_j},K_1)\cdot e(C_{1,j},K_{1,j})^{-1}\\ \cdot e(C_{2,j},K_{2,j})\cdot e((C'_{2,j})^{x_j},K'_{2,j})\end{array}\right)^{\omega_j}}$$
$$m=C'/C$$

下面验证正确性，当 $P(\boldsymbol{x},\boldsymbol{y})=1$，即 $\sum_{j\in[n]}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_j\cdot\boldsymbol{z}_j)=\boldsymbol{1}$ 时，解密者能正确解密.

$$\begin{aligned}&e(C_{0,j}\cdot(C'_{0,j})^{x_j},K_1)\cdot e(C_{1,j},K_{1,j})^{-1}\\&\cdot e(C_{2,j},K_{2,j})\cdot e((C'_{2,j})^{x_j},K'_{2,j})\\&=e([(\boldsymbol{y}_j+\boldsymbol{x}_j\boldsymbol{z}_j\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}+\boldsymbol{x}_j\cdot\boldsymbol{W}')]_1,[\boldsymbol{d}]_2)\\&\quad\cdot e([\boldsymbol{c}_j^{\mathrm{T}}]_1,[(\boldsymbol{W}+\boldsymbol{x}_j\cdot\boldsymbol{W}')\boldsymbol{d}+(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)\boldsymbol{d}_j\\&\quad+\boldsymbol{x}_j(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')\boldsymbol{d}_j']_2)^{-1}\cdot e([\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)]_1,[\boldsymbol{d}_j]_2)\\&\quad\cdot e([\boldsymbol{x}_j\cdot\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')]_1,[\boldsymbol{d}_j']_2\\&=[(\boldsymbol{y}_j+\boldsymbol{x}_j\boldsymbol{z}_j)\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}\boldsymbol{d}+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}+\boldsymbol{x}_j\cdot\boldsymbol{W}')\boldsymbol{d}\\&\quad-\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}+\boldsymbol{x}_j\cdot\boldsymbol{W}')\boldsymbol{d}-\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)\boldsymbol{d}_j\\&\quad-\boldsymbol{x}_j\cdot\boldsymbol{c}_j(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')\boldsymbol{d}_j'+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)\boldsymbol{d}_j\\&\quad+\boldsymbol{x}_j\cdot\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')\boldsymbol{d}_j']_T\\&=[\boldsymbol{y}_j+\boldsymbol{x}_j\boldsymbol{z}_j\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}\boldsymbol{d}]_T\end{aligned}$$

而

$$\prod_{j=1}^{n}(e(C_{0,j}\cdot(C'_{0,j})^{x_j},K_0)\cdot e(C_{1,j},K_{1,j})^{-1}\cdot e(C_{2,j},K_{2,j})\cdot e((C'_{2,j})^{x_j},K'_{2,j}))^{\omega_j}$$
$$=\prod_{j=1}^{n}[\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_j\boldsymbol{z}_j)\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}\boldsymbol{d}]_T$$
$$=[\sum_{j=1}^{n}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_j\boldsymbol{z}_j)\left(\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}\boldsymbol{d}\right)]_T$$
$$=[\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\boldsymbol{d}]_T=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{U}_0\boldsymbol{d}]_2)$$
$$C=e(C_0,K_0)/e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{U}_0\boldsymbol{d}]_2)$$
$$=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2)/e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{U}_0\boldsymbol{d}]_2)$$
$$=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)$$

方案的解密正确性得证.

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. 我们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

$\boldsymbol{c},\boldsymbol{c}_j\leftarrow\mathrm{span}(\boldsymbol{A}_1,\boldsymbol{a}_2)$,

$\boldsymbol{W}\rightarrow\widehat{\boldsymbol{V}}_j=\boldsymbol{W}+\boldsymbol{V}_j$,

$\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1\rightarrow\widehat{\boldsymbol{U}}_j=\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_j$,

$\boldsymbol{W}'\rightarrow\widehat{\boldsymbol{V}}'_j=\boldsymbol{W}+\boldsymbol{V}'_j$,

$\boldsymbol{W}'_0+j\cdot\boldsymbol{W}'_1\rightarrow\widehat{\boldsymbol{U}}'_j=\boldsymbol{W}'_0+j\cdot\boldsymbol{W}'_1+\boldsymbol{U}'_j$,

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

$$ct_v=\begin{Bmatrix}C_0=[\boldsymbol{c}^{\mathrm{T}}]_1,\\ \begin{Bmatrix}C_{0,j}=[\boldsymbol{y}_j\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}\widehat{\boldsymbol{V}}_j]_1,\\ C'_{0,j}=[\boldsymbol{z}_j\begin{pmatrix}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}\widehat{\boldsymbol{V}}_j]_1\end{Bmatrix}_{j\in[n]}\\ \{C_{1,j}=[\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in|n|}\\ \{C_{2,j}=[\boldsymbol{c}_j^{\mathrm{T}}\widehat{\boldsymbol{U}}_j]_1,C'_{2,j}=[\boldsymbol{c}_j^{\mathrm{T}}\widehat{\boldsymbol{U}}'_j]_1\}_{j\in[n]},\\ C'=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\cdot m\end{Bmatrix}$$

(2)密钥分布

标准密文:由密钥生成算法生成.

熵扩张密钥:

$$sk_x:=\begin{pmatrix}K_0=[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{d}]_2\\ \{K_{1,j}=[\widehat{\boldsymbol{V}}_j\boldsymbol{d}+\boldsymbol{x}_j\cdot\widehat{\boldsymbol{V}}'_j\boldsymbol{d}+\widehat{\boldsymbol{U}}_j\boldsymbol{d}_j+\boldsymbol{x}_j\cdot\widehat{\boldsymbol{U}}'_j\boldsymbol{d}'_j]_2\}_{j\in[n]}\\ \{K_{2,j}=[\boldsymbol{d}_j]_2,K'_{2,j}=[\boldsymbol{d}'_j]_2\}_{j\in[n]}\end{pmatrix}$$

其中 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}'_j\leftarrow\mathrm{span}(\boldsymbol{B})$.

伪标准密钥:

$$sk_x:=\begin{pmatrix}K_0=[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{d}]_2\\ \{\boldsymbol{K}_{1,j}=[\widehat{\boldsymbol{V}}_j\boldsymbol{d}+\boldsymbol{x}_j\cdot\widehat{\boldsymbol{V}}'_j\boldsymbol{d}+\widehat{\boldsymbol{U}}_j\boldsymbol{d}_j+\boldsymbol{x}_j\cdot\widehat{\boldsymbol{U}}'_j\boldsymbol{d}'_j]_2\}_{j\in[n]}\\ \{K_{2,j}=[\boldsymbol{d}_j]_2,K'_{2,j}=[\boldsymbol{d}'_j]_2\}_{j\in[n]}\end{pmatrix}$$

其中 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}'_j\leftarrow\boldsymbol{Z}_p^{k+1}$.

伪半功能密钥:

$$sk_x:=\begin{pmatrix}K_0=[\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{d}]_2\\ \{K_{1,j}=[\widehat{\boldsymbol{V}}_j\boldsymbol{d}+\boldsymbol{x}_j\cdot\widehat{\boldsymbol{V}}'_j\boldsymbol{d}+\widehat{\boldsymbol{U}}_j\boldsymbol{d}_j+\boldsymbol{x}_j\cdot\widehat{\boldsymbol{U}}'_j\boldsymbol{d}'_j]_2\}_{j\in[n]}\\ \{K_{2,j}=[\boldsymbol{d}_j]_2,K'_{2,j}=[\boldsymbol{d}'_j]_2\}_{j\in[n]}\end{pmatrix}$$

其中 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}'_j\leftarrow\boldsymbol{Z}_p^{k+1}$

半功能密钥:

$$sk_x:=\begin{pmatrix}K_0=[\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{d}]_2\\ \{K_{1,j}=[\widehat{\boldsymbol{V}}_j\boldsymbol{d}+\boldsymbol{x}_j\cdot\widehat{\boldsymbol{V}}'_j\boldsymbol{d}+\widehat{\boldsymbol{U}}_j\boldsymbol{d}_j+\boldsymbol{x}_j\cdot\widehat{\boldsymbol{U}}'_j\boldsymbol{d}'_j]_2\}_{j\in[n]}\\ \{K_{2,j}=[\boldsymbol{d}_j]_2,K'_{2,j}=[\boldsymbol{d}'_j]_2\}_{j\in[n]}\end{pmatrix}$$

其中 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}'_j\leftarrow\mathrm{span}(\boldsymbol{B})$.

假设攻击者 $\mathcal{A}$ 在一次游戏中最多可进行 Q 次私钥查询,用$\mathrm{Adv}_{xx}(\lambda)$表示 $\mathcal{A}$ 在Game_{xx}的优势. 基于描述的密文、密钥分布,下面我们详细描述 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.

$\mathrm{Game}_{i,2}$: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.

$\mathrm{Game}_{i,3}$: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.

$\mathrm{Game}_{\mathrm{Final}}$: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Game'_0的攻击优势满足$|\mathrm{Adv}_0(\lambda)-\mathrm{Adv}'_0(\lambda)|>\varepsilon$,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中的左右分布,并且 $\mathrm{Time}(\mathcal{B}_0)\approx\mathrm{Time}(\mathcal{A})$.

证明　挑战者 $\mathcal{B}_0$ 得到分布:

$$\begin{Bmatrix} aux:[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,\\ \quad [\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1']_1\\ ct:[\boldsymbol{c}_0]_1,\{[\boldsymbol{c}_{0,j}]_1,[\boldsymbol{c}_{0,j}']_1,[\boldsymbol{c}_{1,j}]_1,[\boldsymbol{c}_{2,j}]_1,[\boldsymbol{c}_{2,j}']_1\}_{j\in[n]}\\ sk:[\boldsymbol{K}_1]_2,\{[\boldsymbol{K}_{1,j}]_2,[\boldsymbol{K}_{1,j}']_2,[\boldsymbol{K}_{2,j}]_2,[\boldsymbol{K}_{2,j}']_2\}_{j\in[n]} \end{Bmatrix}$$

表 1 安全性证明中用到的 Game 序列

Game	CT	SK		
		$\kappa<i$	$\kappa=i$	$\kappa>i$
0 0′	标准 熵扩张	标准 熵扩张		
i	—	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 m	半功能		

$\mathcal{B}_0$ 需区分该分布是熵扩张引理的左分布,还是右分布.

steup:挑战者模拟方案,在 $\boldsymbol{Z}_p^{3k}$ 和 $\boldsymbol{Z}_p^{3k\times(k+1)}$ 分别随机选取向量 $\boldsymbol{k}$ 和矩阵 $\boldsymbol{U}_0$,输出公共参数 mpk:

$$mpk=\{[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{U}_0]_1,e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\}$$

Phase1:攻击者 $\mathcal{A}$ 申请向量 $\boldsymbol{x}'=(x_1',x_2',\cdots,x_n')$ 对应的私钥,挑战者 $\mathcal{B}_0$ 模拟密钥生成算法,在 $\boldsymbol{Z}_p^{k+1}$ 随机选取向量 $\boldsymbol{d}$,生成向量 $\boldsymbol{x}'$ 对应的私钥:

$$sk_{\boldsymbol{x}'}:=\begin{pmatrix} K_0=[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{K}_1\boldsymbol{d}]_2\\ \{K_{1,j}=[\boldsymbol{K}_{1,j}\boldsymbol{d}]_2\cdot[\boldsymbol{x}_j'\cdot\boldsymbol{K}_{1,j}\boldsymbol{d}]_2\}_{j\in[n]}\\ \{K_{2,j}=[\boldsymbol{K}_{2,j}\boldsymbol{d}]_2\cdot\boldsymbol{K}_{2,j}'[\boldsymbol{K}_{2,j}'\boldsymbol{d}]_2\}_{j\in[n]} \end{pmatrix}$$

Challenge:攻击者 $\mathcal{A}$ 选择两个等长的消息 m_0 和 m_1,以及要挑战的访问结构 $\boldsymbol{v}^*=\{(\boldsymbol{y}_j,\boldsymbol{z}_j):j\in[n],\boldsymbol{y}_j,\boldsymbol{z}_j\in\boldsymbol{Z}_p^{l'}\}$(Phase1 中的向量 $\boldsymbol{x}'$ 与挑战访问结构 $\boldsymbol{v}^*=\{(\boldsymbol{y}_j,\boldsymbol{z}_j):j\in[n],\boldsymbol{y}_j,\boldsymbol{z}_j\in\boldsymbol{Z}_p^{l'}\}$ 都不满足 $\sum_{j=1}^{n}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_j'\cdot\boldsymbol{z}_j)=\boldsymbol{1}$)发送给挑战者 $\mathcal{B}_0$,挑战者 $\mathcal{B}_0$ 随机选取 $b\in\{0,1\}$ 及 $\boldsymbol{U}\leftarrow\boldsymbol{Z}_p^{(l'-1)\times(k+1)}$,利用向量 $\boldsymbol{k}$ 以及得到的分布生成挑战密文:

$$ct_{\boldsymbol{v}^*}=\begin{Bmatrix} C_0^{\boldsymbol{v}^*}=[\boldsymbol{c}_0]_1,\\ \{C_{0,j}^{\boldsymbol{v}^*}=[\boldsymbol{y}_j\begin{pmatrix}\boldsymbol{c}_0\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}]_1\cdot[\boldsymbol{c}_{0,j}]_1,\\ C_{0,j}^{\boldsymbol{v}^*\prime}=[\boldsymbol{z}_j\begin{pmatrix}\boldsymbol{c}_0\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}]_1\cdot[\boldsymbol{c}_{0,j}']_1\}_{j\in[n]}\\ \begin{Bmatrix}C_{1,j}^{\boldsymbol{v}^*}=[\boldsymbol{c}_{1,j}]_1,\boldsymbol{c}_{2,j}^{\boldsymbol{v}^*}=[\boldsymbol{c}_{2,j}]_1,\\ C_{2,j}^{\boldsymbol{v}^*\prime}=[\boldsymbol{c}_{2,j}']_1\end{Bmatrix}_{j\in[n]},\\ C^{\boldsymbol{v}^*\prime}=e([\boldsymbol{c}_0]_1,[\boldsymbol{k}]_2)\cdot m \end{Bmatrix}$$

Phase2:与 Phase1 相同.

Guess: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,那么得到标准密钥与标准挑战密文,如果挑战者 $\mathcal{B}_0$ 得到的是右分布,那么得到熵扩张密钥与熵扩张挑战密文.

如果攻击者 $\mathcal{A}$ 的攻击优势满足 $|\mathrm{Adv}_0(\lambda)-\mathrm{Adv}_{0'}(\lambda)|\geqslant\varepsilon$,那么挑战者 $\mathcal{B}_0$ 同样以不可忽略的优势区分熵扩张引理的左右分布. 由于左右分布的不可区分性,因此 Game_0 和 $\mathrm{Game}_{0'}$ 也是不可区分的.

证毕

引理 2 根据表格 1,容易得到 $\mathrm{Game}_{0'}\equiv\mathrm{Game}_1$.

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

证明 挑战者 $\mathcal{B}_1$ 得到分布 $([\boldsymbol{B}]_2,[\boldsymbol{t}]_2,\{[\boldsymbol{t}_j]_2,[\boldsymbol{t}_j']_2\}_{j\in[n]})$. $\mathcal{B}_0$ 需区分分布 $\boldsymbol{t},\boldsymbol{t}_j,\boldsymbol{t}_j'$ 是 $\boldsymbol{t}=\boldsymbol{B}\boldsymbol{s},\boldsymbol{t}_j=\boldsymbol{B}\boldsymbol{s}_j,\boldsymbol{t}_j'=\boldsymbol{B}\boldsymbol{s}_j'$,其中 $\boldsymbol{s},\boldsymbol{s}_j,\boldsymbol{s}_j'\leftarrow\boldsymbol{Z}_p^k$,还是 $\boldsymbol{t}=\boldsymbol{z},\boldsymbol{t}_j=\boldsymbol{z}_j,\boldsymbol{t}_j'=\boldsymbol{z}_j'$,其中 $\boldsymbol{z},\boldsymbol{z}_j,\boldsymbol{z}_j'\leftarrow\boldsymbol{Z}_p^{k+1}$.

steup:挑战者 $\mathcal{B}_1$ 模拟方案,选取 $\boldsymbol{A}_1,\boldsymbol{a}_2\leftarrow\boldsymbol{Z}_p^{3k\times k}\times\boldsymbol{Z}_p^{3k}$,取定 $\boldsymbol{a}_2^{\parallel}$,选取 $\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1,\boldsymbol{W}',\boldsymbol{W}_0',\boldsymbol{W}_1',\boldsymbol{U}_0\leftarrow\boldsymbol{Z}_p^{3k\times(k+1)}$,$\boldsymbol{V}_j,\boldsymbol{U}_j,\boldsymbol{V}_j',\boldsymbol{U}_j'\leftarrow\mathrm{span}^{k+1}(\boldsymbol{a}_2^{\parallel})$,$\alpha\leftarrow\boldsymbol{Z}_p$. 在 $\boldsymbol{Z}_p^{3k}$ 随机选取向量 $\boldsymbol{k}$,输出公共参数 mpk 给 $\mathcal{A}$:

$$mpk=\{[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0'],[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{U}_0]_1e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\}$$

Phase1,2: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 $\boldsymbol{x}'=(x_1',x_2',\cdots,x_n')$,我们分三种情况讨论.

Case1:$\kappa<i$,挑战者 $\mathcal{B}_1$ 利用得到的分布随机选取 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}_j'\leftarrow\mathrm{span}(\boldsymbol{B})$,并且利用向量 $\boldsymbol{k}$、随机数 α 和攻击者 $\mathcal{A}$ 询问的向量生成半功能私钥 $sk_{\boldsymbol{x}'}$:

$$sk_{\boldsymbol{x}'}:=\begin{pmatrix} K_0=[\boldsymbol{k}+\alpha\,\boldsymbol{a}_2^{\parallel}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{d}]_2\\ \{K_{1,j}=[\widehat{\boldsymbol{V}}_j\boldsymbol{d}+\boldsymbol{x}_j'\cdot\widehat{\boldsymbol{V}}_j\boldsymbol{d}+\widehat{\boldsymbol{U}}_j'\boldsymbol{d}_j+\boldsymbol{x}_j'\cdot\widehat{\boldsymbol{U}}_j'\boldsymbol{d}_j']_2\}_{j\in|n|}\\ \{K_{2,j}=[\boldsymbol{d}_j]_2,K_{2,j}'=[\boldsymbol{d}_j']_2\}_{j\in[n]} \end{pmatrix}$$

发送给攻击者 $\mathcal{A}$.

Case2:$\kappa>i$,挑战者 $\mathcal{B}_1$ 利用得到的分布随机选取 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}_j'\leftarrow\mathrm{span}(\boldsymbol{B})$,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 $sk_{\boldsymbol{x}'}$:

$$sk_{\boldsymbol{x}'}:=\begin{pmatrix} K_0=[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{d}]_2\\ \{K_{1,j}=[\widehat{\boldsymbol{V}}_j\boldsymbol{d}+\boldsymbol{x}_j'\cdot\widehat{\boldsymbol{V}}_j'\boldsymbol{d}+\widehat{\boldsymbol{U}}_j\boldsymbol{d}_j+\boldsymbol{x}_j'\cdot\widehat{\boldsymbol{U}}_j'\boldsymbol{d}_j']_2\}_{j\in[n]}\\ \{K_{2,j}=[\boldsymbol{d}_j]_2,K_{2,j}'=[\boldsymbol{d}_j']_2\}_{j\in|n|} \end{pmatrix}$$

发送给攻击者 $\mathcal{A}$.

Case3:$\kappa=i$,挑战者 $\mathcal{B}_2$ 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 $\boldsymbol{k}$、$[\boldsymbol{t}]_2$,$\{[\boldsymbol{t}_j]_2,[\boldsymbol{t}_j']_2\}_{j\in[n]}$

和攻击者 $\mathcal{A}$ 询问的向量生成私钥 $sk_{x'}$:

$$sk_{x'}:\begin{pmatrix} K_0=[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{t}]_2, K_1=[\boldsymbol{t}]_2 \\ \left\{\begin{matrix} K_{1,j}=[(\boldsymbol{W}+\boldsymbol{x}_j'\cdot\boldsymbol{W}')\boldsymbol{t}+(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)\boldsymbol{t}_j \\ \quad +\boldsymbol{x}_j'\cdot(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')\boldsymbol{t}_j']_2 \end{matrix}\right\}_{j\in|n|} \\ \{K_{2,j}=[\boldsymbol{t}_j]_2, K_{2,j}'=[\boldsymbol{t}_j']_2\}_{j\in[n]} \end{pmatrix}$$

发送给攻击者 $\mathcal{A}$.

Challenge:攻击者 $\mathcal{A}$ 选择两个等长的消息 m_0 和 m_1,以及要挑战的访问结构 $\boldsymbol{v}^*=\{(\boldsymbol{y}_j,\boldsymbol{z}_j):j\in[n],\boldsymbol{y}_j,\boldsymbol{z}_j\in\boldsymbol{Z}_p^{l'}\}$(任何询问向量 $\boldsymbol{x}'$ 与挑战访问结构 $\boldsymbol{v}^*=\{(\boldsymbol{y}_j,\boldsymbol{z}_j):j\in[n],\boldsymbol{y}_j,\boldsymbol{z}_j\in\boldsymbol{Z}_p^{l'}\}$ 都不满足 $\sum_{j=1}^{n}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}_j'\cdot\boldsymbol{z}_j)=\boldsymbol{1}$)发送给挑战者 $\mathcal{B}_1$,挑战者 $\mathcal{B}_1$ 随机选取 $b\in\{0,1\}$,随机选取 $\boldsymbol{c},\boldsymbol{c}_j,\boldsymbol{c}_j'\leftarrow\text{span}(\boldsymbol{A}_1,\boldsymbol{a}_2)$,生成熵扩张挑战密文:

$$ct_{\boldsymbol{v}}=\begin{Bmatrix} C_0=[\boldsymbol{c}^{\mathrm{T}}]_1, \\ \left\{\begin{matrix} C_{0,j}=[\boldsymbol{y}_j\begin{pmatrix}\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}+\boldsymbol{V}_j)]_1, \\ C_{0,j}'=[\boldsymbol{z}_j\begin{pmatrix}\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}'+\boldsymbol{V}_j')]_1 \end{matrix}\right\}_{j\in[n]} \\ \{C_{1,j}=[\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in|n|} \\ \left\{\begin{matrix} C_{2,j}=[\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_1)]_1, \\ C_{2,j}'=[\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1'+\boldsymbol{U}_j')]_1 \end{matrix}\right\}_{j\in[n]} \\ C'=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\cdot m \end{Bmatrix}$$

Guess:攻击者 $\mathcal{A}$ 给出 b 的猜测 b'.

如果分布 $\boldsymbol{t},\boldsymbol{t}_j,\boldsymbol{t}_j'$ 是 $\boldsymbol{t}=\boldsymbol{B}\boldsymbol{s},\boldsymbol{t}_j=\boldsymbol{B}\boldsymbol{s}_j,\boldsymbol{t}_j'=\boldsymbol{B}\boldsymbol{s}_j'$,那么第 i 次询问的私钥是熵扩张私钥,上述游戏对应的是 Game_i,如果是 $\boldsymbol{t}=\boldsymbol{z},\boldsymbol{t}_j=\boldsymbol{z}_j,\boldsymbol{t}_j'=\boldsymbol{z}_j'$,那么第 i 次询问的私钥是伪标准私钥,对应的是 $\text{Game}_{i,1}$.

如果存在攻击者 $\mathcal{A}$ 使得 $|\text{Adv}_i(\lambda)-\text{Adv}_{i,1}(\lambda)|>\varepsilon$,那么挑战者以不可忽略的优势区分 $\boldsymbol{t}=\boldsymbol{B}\boldsymbol{s},\boldsymbol{t}_j=\boldsymbol{B}\boldsymbol{s}_j,\boldsymbol{t}_j'=\boldsymbol{B}\boldsymbol{s}_j'$ 和 $\boldsymbol{t}=\boldsymbol{z},\boldsymbol{t}_j=\boldsymbol{z}_j,\boldsymbol{t}_j'=\boldsymbol{z}_j'$,从而解决 k-Lin 问题. 因此 Game_i 和 $\text{Game}_{i,1}$ 是不可区分的.

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. 下面说明这两个游戏无法区分.

Setup:同引理 3 中的 Setup,在这里挑战者选取 $\boldsymbol{B}\leftarrow\boldsymbol{Z}_p^{(k+1)\times k}$,输出公共参数:

$$mpk=\{[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{U}_0]_1,e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\}$$

Phase1,2:$\text{Game}_{i,1}$ 中挑战者 $\mathcal{B}$ 针对攻击者 $\mathcal{A}$ 询问的向量 $\boldsymbol{x}'=(x_1',x_2',\cdots,x_n')$,随机选取向量 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}_j'\leftarrow\text{span}(\boldsymbol{B})$,生成 $\boldsymbol{x}'$ 私钥:

$$sk_{x'}:\begin{pmatrix} K_0=[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2, K_1=[\boldsymbol{d}]_2 \\ \left\{\begin{matrix} K_{1,j}=[(\boldsymbol{W}+\boldsymbol{V}_j+\boldsymbol{x}_j'\cdot(\boldsymbol{W}'+\boldsymbol{V}_j')\boldsymbol{d} \\ \quad +(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_j)\boldsymbol{d}_j \\ \quad +\boldsymbol{x}_j'\cdot(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1'+\boldsymbol{U}_j')\boldsymbol{d}_j']_2 \end{matrix}\right\}_{j\in[n]} \\ \{K_{2,j}=[\boldsymbol{d}_j]_2, K_{2,j}'=[\boldsymbol{d}_j']_2\}_{j\in|n|} \end{pmatrix}$$

在 $\text{Game}_{i,2}$ 中,挑战者随机选取 $\boldsymbol{d}_1,\boldsymbol{d}_{1j},\boldsymbol{d}_{1j}'\leftarrow\boldsymbol{Z}_p^{k+1}$,生成 $\boldsymbol{x}'$ 私钥:

$$sk_{x'}:\begin{pmatrix} K_0=[\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}+\boldsymbol{U}_0\boldsymbol{d}_1]_2, K_1=[\boldsymbol{d}_1]_2 \\ \left\{\begin{matrix} K_{1,j}=[(\boldsymbol{W}+\boldsymbol{V}_j+\boldsymbol{x}_j'\cdot(\boldsymbol{W}'+\boldsymbol{V}_j')\boldsymbol{d}_1 \\ \quad +(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_j)\boldsymbol{d}_{1j} \\ \quad +\boldsymbol{x}_j'\cdot(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1'+\boldsymbol{U}_j')\boldsymbol{d}_{1j}']_2 \end{matrix}\right\}_{j\in[n]} \\ \{K_{2,j}=[\boldsymbol{d}_{1j}]_2, K_{2,j}'=[\boldsymbol{d}_{1j}']_2\}_{j\in[n]} \end{pmatrix}$$

可以观察到两个游戏的第 i 次私钥查询差别仅在于 K_0 这一分量,分析这两个分量:

$$[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2=[\boldsymbol{k}]_2\cdot[\boldsymbol{0}+\boldsymbol{U}_0\boldsymbol{d}]_2,$$

$$[\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}+\boldsymbol{U}_0\boldsymbol{d}_1]_2=[\boldsymbol{k}]_2\cdot[\alpha\boldsymbol{a}_2^{\parallel}+\boldsymbol{U}_0\boldsymbol{d}_1]_2$$

由于随机值 $\boldsymbol{\alpha}$ 以及随机向量 $\boldsymbol{d},\boldsymbol{d}_1$ 的参与,$[\boldsymbol{0}+\boldsymbol{U}_0\boldsymbol{d}]_2$ 与 $[\alpha\boldsymbol{a}_2^{\parallel}+\boldsymbol{U}_0\boldsymbol{d}_1]_2$ 同分布,并且在解密算法中,由于 $e([\boldsymbol{c}^{\mathrm{T}}]_1,[\alpha\boldsymbol{a}_2^{\parallel}]_2)=[\boldsymbol{c}^{\mathrm{T}}\alpha\boldsymbol{a}_2^{\parallel}]_T=1$,尽管两个游戏这一分量不同,但对解密没有影响,故从攻击者角度来看这两种密钥无法区分.

Challenge:由于这两个游戏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 k-Lin 问题,并且 $\text{Time}(\mathcal{B}_2)\approx\text{Time}(\mathcal{A})$.

证明　引理 5 的证明完全相似引理 3,只是在私钥查询的时候: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 $\boldsymbol{x}'=(x_1',x_2',\cdots,x_n')$,挑战者 $\mathcal{B}_2$ 利用向量 $\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}$ 和 $[\boldsymbol{t}]_2,\{[\boldsymbol{t}_j]_2,[\boldsymbol{t}_j']_2\}_{j\in[n]}$ 生成私钥:

$$sk_{x'}:\begin{pmatrix} K_0=[\boldsymbol{k}+\alpha\boldsymbol{a}_2^{\parallel}+\boldsymbol{U}_0\boldsymbol{t}]_2, K_1=[\boldsymbol{t}]_2 \\ \left\{\begin{matrix} K_{1,j}=[(\boldsymbol{W}+\boldsymbol{x}_j'\cdot\boldsymbol{W}')\boldsymbol{t}+(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1)\boldsymbol{t}_j \\ \quad +\boldsymbol{x}_j'\cdot(\boldsymbol{W}_0'+j\cdot\boldsymbol{W}_1')\boldsymbol{t}_j']_2 \end{matrix}\right\}_{j\in[n]} \\ \{K_{2,j}=[\boldsymbol{t}_j]_2, K_{2,j}'=[\boldsymbol{t}_j']_2\}_{j\in[n]} \end{pmatrix}$$

发送给攻击者 $\mathcal{A}$.

如果 $\boldsymbol{t}=\boldsymbol{z},\boldsymbol{t}_j=\boldsymbol{z}_j,\boldsymbol{t}_j'=\boldsymbol{z}_j'$,那么第 i 次询问的私钥是伪半功能私钥,上述游戏对应的是 $\text{Game}_{i,2}$,如果 $\boldsymbol{t}=\boldsymbol{B}\boldsymbol{s},\boldsymbol{t}_j=\boldsymbol{B}\boldsymbol{s}_j,\boldsymbol{t}_j'=\boldsymbol{B}\boldsymbol{s}_j'$,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 $\mathcal{A}$ 使 $|\text{Adv}_{i,2}(\lambda)-\text{Adv}_{i,3}(\lambda)|>\varepsilon$ 不可忽视,那么挑战者同样以不可忽略的优势区分 $\boldsymbol{t}=\boldsymbol{Bs}$,$\boldsymbol{t}_j=\boldsymbol{Bs}_j$,$\boldsymbol{t}'_j=\boldsymbol{Bs}'_j$和 $\boldsymbol{t}=\boldsymbol{z}$,$\boldsymbol{t}_j=\boldsymbol{z}_j$,$\boldsymbol{t}'_j=\boldsymbol{z}'_j$.

证毕

引理 6 从表 1,容易得到$\text{Game}_i\equiv\text{Game}_{i-1,3}$.

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

证明 这两个游戏的差别在于Game_{Q+1}挑战密文是对消息 m 加密的熵扩张密文,$\text{Game}_{\text{Final}}$挑战密文是对随机数加密的熵扩张密文. 下面来说明这两个游戏不可区分.

挑战者 $\mathcal{B}_3$ 选取$\boldsymbol{A}_1,\boldsymbol{a}_2\leftarrow\boldsymbol{Z}_p^{3k\times k}\times\boldsymbol{Z}_p^{3k}$,$\boldsymbol{B}\leftarrow\boldsymbol{Z}_p^{(k+1)\times k}$,取定$\boldsymbol{a}_2^{\parallel}$,选取 $\boldsymbol{W},\boldsymbol{W}_0,\boldsymbol{W}_1,\boldsymbol{W}',\boldsymbol{W}'_0,\boldsymbol{W}'_1,\boldsymbol{U}_0\leftarrow\boldsymbol{Z}_p^{3k\times(k+1)}$,$\boldsymbol{V}_j,\boldsymbol{U}_j,\boldsymbol{V}'_j,\boldsymbol{U}'_j\leftarrow\text{span}^{k+1}(\boldsymbol{a}_2^{\parallel})$,$\alpha\leftarrow\boldsymbol{Z}_p$.

steup:挑战者模拟方案,在 $\boldsymbol{Z}_p^{3k}$ 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 $e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)=e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)$,输出公共参数 mpk 给 $\mathcal{A}$:

$$mpk=\{[\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}']_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}_0]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{W}'_1]_1,[\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{U}'_0]_1,e([\boldsymbol{A}_1^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)\}$$

Phase1:挑战者 $\mathcal{B}_3$ 针对攻击者 $\mathcal{A}$ 申请访问的向量 $\boldsymbol{x}'=(x'_1,x'_2,\cdots,x'_n)$,随机选取 $\boldsymbol{d},\boldsymbol{d}_j,\boldsymbol{d}'_j\leftarrow\text{span}(\boldsymbol{B})$,生成半功能私钥:

$$sk_{\boldsymbol{x}}:\begin{pmatrix}K_0=[\boldsymbol{k}+\boldsymbol{U}_0\boldsymbol{d}]_2,K_1=[\boldsymbol{d}]_2\\ \left\{\begin{array}{l}K_{1,j}=[(\boldsymbol{W}+\boldsymbol{V}_j+\boldsymbol{x}'_j\cdot(\boldsymbol{W}'+\boldsymbol{V}'_j)\boldsymbol{d}\\ \qquad+(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_j)\boldsymbol{d}_j\\ \qquad+\boldsymbol{x}'_j\cdot(\boldsymbol{W}'_0+j\cdot\boldsymbol{W}'_1+\boldsymbol{U}'_j)\boldsymbol{d}'_j]_2\end{array}\right\}_{j\in[n]}\\ \{K_{2,j}=[\boldsymbol{d}_j]_2,K'_{2,j}=[\boldsymbol{d}'_j]_2\}_{j\in[n]}\end{pmatrix}$$

Challenge:攻击者 $\mathcal{A}$ 选择两个等长的消息 m_0 和 m_1,以及要挑战的访问结构 $\boldsymbol{v}^*=\{(\boldsymbol{y}_j,\boldsymbol{z}_j):j\in[n],\boldsymbol{y}_j,\boldsymbol{z}_j\in\boldsymbol{Z}_p^{l'}\}$(Phase1 中的 $\boldsymbol{x}'$与挑战访问结构 $\boldsymbol{v}^*=\{(\boldsymbol{y}_j,\boldsymbol{z}_j):j\in[n],\boldsymbol{y}_j,\boldsymbol{z}_j\in\boldsymbol{Z}_p^{l'}\}$都不满足 $\sum_{j=1}^{n}\omega_j(\boldsymbol{y}_j+\boldsymbol{x}'_j\cdot\boldsymbol{z}_j)=\boldsymbol{1}$)发送给挑战者 $\mathcal{B}_3$,挑战者 $\mathcal{B}_3$ 随机选取 $b\in\{0,1\}$,随机选取向量 $\boldsymbol{c},\boldsymbol{c}_j,\boldsymbol{c}'_j\leftarrow\text{span}(\boldsymbol{A}_1,\boldsymbol{a}_2)$生成熵扩张挑战密文:

$$ct_v=\begin{Bmatrix}C_0=[\boldsymbol{c}^{\mathrm{T}}]_1,\\ \left\{\begin{array}{l}C_{0,j}=[\boldsymbol{y}_j\begin{pmatrix}\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}+\boldsymbol{V}_j)]_1,\\ C'_{0,j}=[\boldsymbol{z}_j\begin{pmatrix}\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{U}_0\\ \boldsymbol{U}\end{pmatrix}+\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}+\boldsymbol{V}'_j)]_1\end{array}\right\}_{j\in[n]}\\ \{C_{1,j}=[\boldsymbol{c}_j^{\mathrm{T}}]_1\}_{j\in[n]}\\ \left\{\begin{array}{l}C_{2,j}=[\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}_0+j\cdot\boldsymbol{W}_1+\boldsymbol{U}_j)]_1,\\ C'_{2,j}=[\boldsymbol{c}_j^{\mathrm{T}}(\boldsymbol{W}'_0+j\cdot\boldsymbol{W}'_1+U'_j)]_1\end{array}\right\}_{j\in[n]}\\ C'=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}]_2)\cdot m\\ \quad=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)\cdot e([\boldsymbol{c}^{\mathrm{T}},[\alpha\boldsymbol{a}_2]_1)^{-1}\cdot m\end{Bmatrix}$$

Phase2:与 Phase1 相同.

Guess:攻击者 $\mathcal{A}$ 给出 b 的猜测 b'.

在熵扩张挑战密文中,

$$e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)\cdot e([\boldsymbol{c}^{\mathrm{T}}]_1,[\alpha\boldsymbol{a}]_2)^{-1}=e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{k}']_2)\cdot e([\boldsymbol{c}^{\mathrm{T}}]_1,[\boldsymbol{a}_2^{\parallel}]_2)^{-\alpha},$$

由于随机数 α 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 m 得到的密文同分布. 所以从攻击者 $\mathcal{A}$ 角度来说加密消息 m 和加密随机数得到的熵扩张密文不可区分.

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

证毕

定理 1 在熵扩张引理和 k-Lin 困难假设成立的条件下,本文提出的算术张成 CP-ABE 方案是适应性安全的,并且

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

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:

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

由引理 1 知$|\text{Adv}_0(\lambda)-\text{Adv}_{0'}(\lambda)|\leqslant\varepsilon$.

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

Game_i 到Game_{i+1}的逼近可理解为

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

由引理 3~6 知:

$$\begin{aligned}|\text{Adv}_i(\lambda)-\text{Adv}_{i+1}(\lambda)|&\leqslant|\text{Adv}_i(\lambda)-\text{Adv}_{i,1}(\lambda)|\\&\quad+|\text{Adv}_{i,1}(\lambda)-\text{Adv}_{i,2}(\lambda)|\\&\quad+|\text{Adv}_{i,2}(\lambda)-\text{Adv}_{i,3}(\lambda)|\\&\quad+|\text{Adv}_{i,3}(\lambda)-\text{Adv}_{i+1}(\lambda)|\\&\leqslant2\varepsilon\end{aligned}$$

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

综上分析,得到攻击者 $\mathcal{A}$ 在 Game_0 中的优势 $\text{Adv}_0(\lambda)\leqslant(2Q+1)\varepsilon$.

在熵扩张引理和 k-Lin 困难假设成立的条件下可知攻击者 $\mathcal{A}$ 在Game_0 中的攻击优势可以忽略,因此本方案是适应性安全的.

证毕

5 效率对比

下面我们在表2中给出本节方案与现有支持算术张成程序的ABE加密方案性能方面的一个对比,从表2中可以看到,与文献[11]中的方案相比,本节方案公开参数长度固定,并且安全性达到了适应安全,但解密复杂度稍高. 与文献[13]相比,本节方案在加密策略方面实现了密文策略,应用范围更加广泛. 其中 n 表示系统属性的个数,$O(n)$表示双线性映射运算时间.

表2　性能对比

方案	公开参数规模	加密策略	困难假设	解密复杂度	安全性
文献[11]	$O(n)$	KP-ABE	DBDH	$2nP$	选择安全
文献[13]	$O(1)$	KP-ABE	$\mathrm{MDDH}^n_{k,k+1}$	$12nP$	适应安全
本节方案	$O(1)$	CP-ABE	$\mathrm{MDDH}^n_{k,k+1}$	$12nP$	适应安全

6 结束语

本文基于素数阶双线性群提出了具有适应安全性的支持算术张成程序的CP-ABE方案,并且基于素数阶双线性群的熵扩张引理,降低了参数规模,最终提出的方案具有参数长度固定和适应安全性的特点. 下一步重点是研究具有适应安全性的、支持其它访问控制策略的、公开参数固定的CP-ABE方案的设计方法.

参考文献

[1] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data [A]. Proceedings of the 13th ACM Conference on Computer and Communications Security [C]. USA: ACM, 2006. 89 – 98.

[2] 宋衍,韩臻,等. 基于访问树的策略隐藏属性加密方案[J]. 通信学报,2015,36(9):1351 – 1358.
SONG Yan, HAN Zhen, et al. Encryption scheme for policy-based hidden attributes based on access tree [J]. Journal on Communications, 2015, 36(9): 1351 – 1358. (in Chinese)

[3] 刘梦君,刘树波,等. 基于LSSS共享矩阵无授权策略的属性密码解密效率提高方案[J]. 电子学报,2015,43(6):1065 – 1072.
LIU Meng-jun, LIU Shu-bo, et al. A scheme for improving the decryption efficiency of attribute passwords based on the LSSS shared matrix unauthorized policy [J]. Acta Electronica Sinica, 2015, 43(6): 1065 – 1072. (in Chinese)

[4] Cheng Y, Zhou H, Ma J, Wang Z. Efficient CP-ABE with non-monotonic access structures [A]. International Conference on Cloud Computing and Security (vol. 10603) [C]. Berlin: Springer, 2017. 315 – 325.

[5] 胡鹏,高海英. 一种实现一般电路的密钥策略的属性加密方案[J]. 软件学报,2016,27(6):1498 – 1510.
HU Peng, GAO Hai-ying. An attribute-based encryption scheme for implementing key strategies for general circuits [J]. Journal of Software, 2016, 27(6): 1498 – 1510. (in Chinese)

[6] Guo Yuyan, Li Jiguo, Zhang Yichen, et al. Hierarchical attribute-based encryption with continuous auxiliary inputs leakage [J]. Security and Communication Networks, 2016, 9(18): 4852 – 4862.

[7] 赵志远,朱智强,等. 属性可撤销且密文长度恒定的属性基加密方案[J]. 电子学报,2018,46(10):2391 – 2399.
ZHAO Zhi-yuan, ZHU Zhi-qiang, et al. Attribute-based encryption scheme with revocable attributes and constant ciphertext length [J]. Acta Electronica Sinica, 2018, 46(10): 2391 – 2399. (in Chinese)

[8] Li Ji-guo, Yao Wei, Zhang Yichen, et al. Flexible and fine-grained attribute-based data storage in cloud computing [J]. IEEE Transactions on Services Computing, 2016, 10(5): 785 – 796.

[9] 李琦,马建峰,等. 一种素数阶群上构造的自适应安全的多授权机构CP-ABE方案[J]. 电子学报,2014,42(4):696 – 702.
LI Qi, MA Jian-feng, et al. A CP-ABE scheme with adaptive security and multi-authorization mechanism constructed on prime order group [J]. Acta Electronica Sinica, 2014, 42(4): 696 – 702. (in Chinese)

[10] Karchmer M, Wigderson A. On span programs [A]. Proceedings of the Eighth Annual Structure in Complexity Theory Conference [C]. USA: IEEE, 1993. 102 – 111.

[11] Ishai Y, Wee H. Partial garbling schemes and their applications [A]. International Colloquium on Automata, Languages, and Programming (Part I, volume 8572) [C]. Berlin, Heidelberg: Springer, 2014. 650 – 662.

[12] Attrapadung N, Hanaoka G, Yamada S. Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs [A]. International Conference on the Theory and Application of Cryptology and Information Security (volume 9452) [C]. Berlin, Heidelberg: Springer, 2015. 575 – 601.

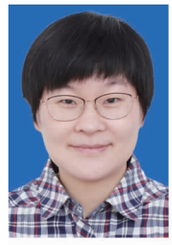
[13] Chen J, Gong J, Kowalczyk L, et al. Unbounded ABE via bilinear entropy expansion, revisited [A]. Annual International Conference on the Theory and Applications of Cryptographic Techniques (volume 10820) [C]. Berlin, Heidelberg: Springer, 2018. 503 – 534.

[14] Chen J, Gay R, Wee H. Improved dual system ABE in prime-order groups via predicate encodings [A]. Annual International Conference on the Theory and Applications of Cryptographic Techniques (volume 9057) [C]. Ber-

lin, Heidelberg: Springer, 2015. 595 – 624.
[15] Lewko A. Tools for simulating features of composite order bilinear groups in the prime order setting[A]. Annual International Conference on the Theory and Applications of Cryptographic Techniques (volume 7237)[C]. Berlin, Heidelberg: Springer, 2012. 318 – 335.
[16] Okamoto T, Takshima K. Fully secure functional encryption with general relations from the decisional linear assumption[A]. Annual Cryptology Conference (volume 6223)[C]. Berlin, Heidelberg: Springer, 2010. 191 – 208.
[17] Chen J, Wee H. Dual systems groups and its applications-compact HIBE and more—Cryptology ePrint Archive [R]. 2014/265.
[18] Lewko A B, Waters B. New techniques for dual system encryption and fully secure HIBE with short ciphertexts [A]. Theory of Cryptography Conference (volume 5978)[C]. Berlin, Heidelberg: Springer, 2010. 455 – 479.

作者简介

魏 铎(通讯作者) 男,1994 年 1 月出生,山西怀仁人. 解放军信息工程大学硕士,主要研究方向公钥密码设计与分析.



高海英 女,1978 年 7 月出生,河南沈丘人. 2006 年毕业于北京邮电大学,获军事学博士学位,现为信息工程大学教授,博士生导师. 主要研究方向为密码算法设计和分析.