doi:10.11959/j.issn.1000-436x.2015019

支持同态算术运算的数据加密方案算法研究

杨攀[1,2]，桂小林[1,2]，姚婧[1,2]，林建财[1,2]，田丰[1,2]，张学军[1,2]
(1. 西安交通大学 电子与信息工程学院，陕西 西安 710049；
2. 西安交通大学 陕西省计算机网络重点实验室，陕西 西安 710049)

摘 要：针对在计算服务中，对用户信息加密以保护隐私时，无法对密文进行计算的问题，提出一种高效的支持密文四则算术运算的同态加密方案CESIL，包括密钥生成、加密、解密及密文运算4个算法。该方案首先借助多项式环重新定义向量的加法和乘法运算，构建多项式系数向量环；然后利用理想格在向量环上划分剩余类，建立商环及其代表元集合；最后，将整数明文映射为代表元，并用代表元所在剩余类的其他元素替换该代表元，以对明文进行加密。商环的运算特性保证CESIL方案支持对密文的加法和乘法运算。在实现CESIL方案时，利用快速傅里叶变换(FFT)算法进一步提高运算效率、减少密钥长度。理论分析及实验结果表明，CESIL是语义安全的，且相比已有的一些同态加密方案，CESIL支持更多的运算类型，拥有较高的运行效率和较小的密钥及密文长度，能更好地满足实际应用需求。

关键词：同态加密；隐私保护；理想格；代表元；计算服务
中图分类号：TP309 **文献标识码**：A

Research on algorithms of data encryption scheme that supports homomorphic arithmetical operations

YANG Pan[1,2], GUI Xiao-lin[1,2], YAO Jing[1,2], LIN Jian-cai[1,2], TIAN Feng[1,2], ZHANG Xue-jun[1,2]
(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;
2. Shaanxi Province Key Laboratory of Computer Network, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: An efficient homomorphic encryption scheme called CESIL was proposed to meet the requirements of operating on encrypted data when protecting users' privacy in computing services. CESIL included key generation algorithm, encryption algorithm, decryption algorithm and calculation algorithm. In CESIL, a polynomial coefficient vector ring was established by defining addition and multiplication using polynomial ring; by using ideal lattice, the vector ring was partitioned into many residue classes to produce a quotient ring and its representative set; the plaintext was encrypted by mapping it to a representative and replacing the representative with another element in the same residue class. The features of operations in quotient ring ensured CESIL operate on encrypted data. Furthermore, the fast Fourier transform (FFT) algorithm was used to increase the efficiency and decrease the length of key. Theoretical analysis and experimental results show that CESIL is semantically secure, and can do addition and multiplication operations on encrypted data homomorphically in a specific scope. Comparing to some existing homomorphic encryption schemes, the CESIL runs efficiently, and has shorter length in key and ciphertext. Thus, the CESIL fits the practical applications better.

Key words: homomorphic encryption; privacy-preserving; ideal lattice; representative; computing service

收稿日期：2013-09-17；**修回日期**：2013-12-15
基金项目：国家科技重大专项基金资助项目(2012ZX03002001)；高等学校博士学科点专项科研基金资助项目(20120201110013)；陕西省科技攻关基金资助项目(2012K06-30)；国家自然科学基金资助项目(61172090, 61472316)；陕西省科技统筹创新工程基金资助项目(2013SZS16-Z01/P01/K01)
Foundation Items: The National Science and Technology Major Project (2012ZX03002001); Research Fund for the Doctoral Program of Higher Education of China (20120201110013); Scientific and Technological Project in Shaanxi Province (2012K06-30); The National Natural Science Foundation of China (61172090, 61472316); Science and Technology Co-ordinating Innovative Engineering Project in Shaanxi Province (2013SZS16-Z01/P01/K01)

1 引言

加密技术是信息安全的核心技术。信息经过加密后，可读的明文信息转变为无法识别的密文信息，即使密文被他人窃取，也无法获得有效信息。因此，加密是保护隐私信息的重要手段。

近年来，一些计算服务渐渐兴起，如云计算服务、移动服务等。这些服务需要用户将数据提交给服务提供者，利用后者的资源对信息进行处理，为用户提供服务。另外，在一些存储服务中，用户将信息交由服务提供者存储，但存储的数据需要根据计算进行更新（如账户余额根据收入支出变动）。在本文中，这类服务也属于计算服务。在上述过程中，服务提供者可得到用户明文数据，其中一些恶意者可能通过分析明文数据窃取用户的隐私信息。若将数据加密后再提交给服务提供者，服务提供者在没有解密密钥的情况下无法通过密文获取用户的明文信息，从而保证了用户数据的隐私安全。但由于无法对加密后的数据进行正确的计算（如加法、乘法等），服务提供者不能利用密文数据为用户提供有效的服务。因此，需要一种实用的加密方案，在保证计算服务顺利进行的前提下保护用户的隐私安全。

针对密文的计算问题，Rivest 等人[1]提出同态加密的思想。同态加密方案支持对密文的计算，并能由密文的计算结果解密得到正确的明文计算结果。在后来的同态加密方案中，有些只支持加法同态，如 Paillier 方案[2]和 Goldwasser-Micali[3]方案；有些只支持乘法同态，如 Unpadded-RSA 方案[4]和 ElGamal 方案[5]。而能同时支持加法和乘法运算的加密方案则较少。Boneh 等人[6]提出了 Boneh-Goh- Nissim 加密方案，能够支持任意次的加法操作，但只能支持一次乘法操作。Gentry[7,8]提出了一种基于理想格的加密方案，并利用其提出的“Bootstrappable”技术，通过对密文的重加密使密文支持任意次数的同态运算，是真正意义上的全同态加密，也为全同态加密的研究奠定重要基础。后续的很多同态加密方案[9~13]也都基于 Gentry 的“Bootstrappable”技术，这些方案对文献[7]都有不同程度的改进。如 Smart 等人[9]改用整数和多项式实现全同态加密，减少了密钥和密文长度，Dijk 等人[10]使用整数进行加密，更利于理解。但由于“Bootstrappable”技术本身的复杂性，即使较低安全性时，一次“Bootstrappable”操作也需要大约 30 s 的时间[13]，因此，这些加密方案都较难应用到实际中。后来，Gentry 等人[14,15]从加强全同态算法中的自展技术、同态加密算法的解密循环的分解技术、实现方法等方面展开了研究，虽然降低了同态加密的复杂性，但仍较难用于实际应用。黄汝维等人[16]针对云计算环境的隐私保护问题提出了基于向量和矩阵运算的加密可计算方案(CESVMC)，算法运行效率较高，但其算术运算方案不支持加法与乘法的混合运算，且仅支持一次乘法或除法运算，运算后的密文长度会增大，而且需要记录密文经过的运算类型以完成解密。

通过以上分析可以发现，目前支持密文计算的同态加密方案或不支持多种密文混合计算，或不支持多次密文计算，或复杂度高，难以实现，总之还较难应用于实际。针对上述问题，本文利用理想格[17,18,19]的性质，设计一种在一定范围内支持多次密文加法和乘法混合运算，并保证语义安全的高效加密方案 CESIL(computable encryption scheme based on ideal lattice)。CESIL 方案支持多种密文操作类型，具有较高的密钥生成及加解密效率，且密钥长度和密文长度都较小，更有利于实际应用。

本文的贡献有：1) 基于理想格的性质，通过在定义的向量环上划分剩余类，提出一种在向量环上支持密文加法和乘法的初始同态加密方案；2) 基于多项式、整数及向量之间的映射关系，建立整数集与初始方案中向量集的映射函数，将明文空间扩展到整数集；3) 分析密钥与明文的关系，给出在指定明文空间内支持同态运算的密钥选取方法；4) 利用 FFT 算法改进 CESIL 方案，进一步提高方案运行效率，并减少方案的密钥长度。

2 问题描述

2.1 同态加密方案

同态加密是一种支持对密文进行计算的加密方案，并且通过密文的计算结果可以得到相应的明文结果。在本文中，一个同态加密方案ε包括密钥生成算法 Gen_ε、加密算法 Enc_ε、解密算法 Dec_ε 和密文计算算法 Cal_ε。

1) 密钥生成算法 Gen_ε：$U\rightarrow k_{ey}$。Gen_ε 利用用户的输入参数 U 生成密钥 k_{ey}。

2) 加密算法 Enc_ε :(k_{ey}，$\mathcal{P}_\varepsilon$)$\rightarrow\mathcal{C}_\varepsilon$。$\mathcal{P}_\varepsilon$ 表示明文空间，$\mathcal{C}_\varepsilon$ 表示密文空间，Enc_ε利用密钥 k_{ey} 加密明文，返回密文。

3) 解密算法 $Dec_\varepsilon:(k_{ey},C_\varepsilon)\rightarrow\mathcal{P}_\varepsilon$。$Dec_\varepsilon$ 利用密钥 k_{ey} 解密密文，返回明文。

4) 计算算法 $Cal_\varepsilon:(C_\varepsilon^t,F_p)\rightarrow C_\varepsilon$。$F_p$ 表示 $\mathcal{P}_\varepsilon$ 中的一个运算集合，如{+,×}。对于输入 $(c_1,c_2,\cdots,c_t)\in C_\varepsilon^t$ 及 $o_p\in F_p$，Cal_ε 首先将 o_p 转换为 C_ε 中相应的运算，然后对 $c_1,c_2,\cdots,c_t$ 进行该运算，得到密文运算结果。例如，用 $\oplus$ 表示密文空间的加法，则 $Cal((c_1,c_2),+)=c_1\oplus c_2$。

密文计算算法是同态加密方案不同于普通加密方案所特有的算法，通过对密文的计算，并对结果进行解密，可以得到相应的明文计算结果，也称该计算满足同态性。

定义 1　同态性。对于加密方案 ε 以及明文空间 $\mathcal{P}_\varepsilon$ 上的运算 o_p，如果 $\forall\ p_1,p_2,\cdots,p_t\in\mathcal{P}_\varepsilon$，满足

$$Dec_\varepsilon\left(k_{ey},Cal_\varepsilon\left((c_1,\cdots,c_t),o_p\right)\right)=o_p\left(p_1,\cdots,p_t\right)\quad(1)$$

则 ε 是 o_p 上的同态加密，称 ε 支持 o_p 同态运算，或 ε 满足 o_p 运算的同态性。其中，$p_i=Dec_\varepsilon\left(k_{ey},c_i\right)$，$o_p\left(p_1,p_2,\cdots,p_t\right)$ 表示对明文进行 o_p 运算。

2.2　基于同态加密的隐私保护模型

在传统的服务模式下，用户通过终端提交服务请求及服务所需参数，由服务提供者在服务端对数据进行处理。这个过程中，数据以明文形式在服务端运行，存在隐私泄露的风险。然而，若使用传统加密技术，服务端又无法对密文进行有效计算，导致服务不能正常进行或返回错误的服务结果。针对该问题，可以利用同态加密技术保护服务中的用户隐私，如图 1 所示。

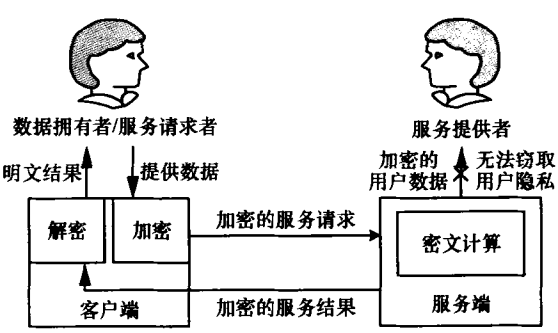


图 1　基于同态加密的隐私保护服务模型

在图 1 模型中，首先利用同态加密技术加密数据，再将密文数据提交给服务端，以保证隐私信息不被恶意服务提供者窃取。服务端无需解密数据，可利用 Cal_ε 算法将原来对明文数据的计算转换为对密文的计算，并得出密文的计算结果。或者，当用户需要更新其存储在服务端的密文信息时，可提交加密的参数，服务器利用 Cal_ε 算法直接对密文进行计算，并用新的计算结果更新原密文数据。最终，当结果以密文形式返回给用户时，经客户端解密可得明文结果，加密方案的同态性保证结果的正确性。上述的整个过程，数据在服务端始终为密文形式，因此，该模型既保证计算服务的顺利进行，又能保护用户隐私的安全。

由此可见，如何实现同态加密方案是要解决的重点问题。同态加密方案对密文运算类型的支持程度决定了其适用范围。若方案能支持任意类型的密文运算，那么所有服务均可对密文进行操作，则该方案可适用于所有计算外包服务中的用户隐私保护问题。然而，支持任意类型密文的同态加密方案目前还较难实现，本文针对最基本的四则算术运算，研究支持密文加减乘除同态运算的高效加密方案 CESIL。特别地，CESIL 能够满足任意次数的加法和乘法混合同态运算，此外，CESIL 也支持有限次数的减法和除法同态运算。因此，将 CESIL 应用于图 1 模型中，只要计算服务由这些运算构成（如更新用户余额、计算合同清单总价等），就可以在保护用户隐私的情况下为用户提供高效服务。

2.3　符号及相关定义

在介绍 CESIL 方案前，为表述方便，对文中符号进行定义，如表 1 所示。其中，本文所用到的关于交换环、理想、商环、剩余类及代表元的概念可参考文献[17]。

表 1　符号定义

符号	意义
$A^{m\times n}$	由集合 A 中元素构成的 $m\times n$ 矩阵
A^m	A 上的 m 维列矩阵或列向量
$x\leftarrow y$	将 y 的值赋给 x
$x\leftarrow_R A$	从集合 A 中随机选取一个值赋给 x
$\lfloor\cdot\rceil$	四舍五入取整，若·代表一个矩阵，则 $\lfloor\cdot\rceil$ 是对矩阵中的每个元素四舍五入取整
(a)	交换环中元素 a 生成的理想，如 A 为交换环，$a\in A$，则 $(a)=\{a\cdot r\mid\forall\ r\in A\}$
$[b]$	元素 b 所在的剩余类，如对于商环 $A/(a)$，$b\in A$，则 $[b]=\{b+a\cdot r\mid\forall\ r\in A\}\in A/(a)$
$[b]_*$	$[b]$的代表元，所有商环 $A/(a)$中元素的代表元构成 $A/(a)$的代表元集合

3　CESIL 的设计

3.1　整数环上基于理想的加密

在整数集 $\mathbb{Z}$ 范围内，任取 $q\in\mathbb{Z}$，可以得到

$\boldsymbol{Z}/(q)$，$\forall [c]\in \boldsymbol{Z}/(q)$，$[c]=\{c+qr|\forall r\in \boldsymbol{Z}\}$。取$\mathcal{P}_{\varepsilon 1}=[-q/2,q/2)\cap \boldsymbol{Z}$作为$\boldsymbol{Z}/(q)$的代表元集合，则$[c]_q=a-q\lfloor a/q\rceil$。由此设计加密方案$\varepsilon_1$，其中，密钥为$q$，明文空间为$\mathcal{P}_{\varepsilon 1}$，密文空间$\mathcal{C}_{\varepsilon 1}=\boldsymbol{Z}$，加解密算法分别为

$$Enc_{\varepsilon 1}(q,p): c\leftarrow_R [p]$$

$$Dec_{\varepsilon 1}(q,\mathrm{c}): p\leftarrow [c]_q$$

其中，p，c分别代表明密文。若明文$p_1,p_2\in\mathcal{P}_{\varepsilon 1}$，对应密文分别为$c_1,c_2$，即$c_1\in[p_1]$，$c_2\in[p_2]$，则

$$Cal_{\varepsilon 1}((c_1,c_2),+)=c_1+c_2$$

并且

$$Dec_{\varepsilon 1}(q,c_1+c_2)=[c_1+c_2]_q=[[c_1]+[c_2]]_q$$
$$=[[p_1]+[p_2]]_q=[p_1+p_2]_q$$

若$p_1+p_2\in\mathcal{P}_{\varepsilon 1}$，则$Dec_{\varepsilon 1}(q,c_1+c_2)=p_1+p_2$。即$\varepsilon_1$支持加法同态运算，同理，若$p_1p_2\in\mathcal{P}_{\varepsilon 1}$，则$\varepsilon_1$支持乘法同态运算。

ε_1方案利用理想构造剩余类，以代表元集合作为明文空间，加密是将代表元随机表示为其所在剩余类中的其他元素，以隐藏明文信息；解密则是计算密文所属剩余类的代表元。根据商环的运算定义可知，只要明文运算结果在明文空间$\mathcal{P}_{\varepsilon 1}$内，$\varepsilon_1$就是{+,×}上的同态加密方案。

但是该方案很容易通过分析有限个数的明/密文对破解出密钥，无法保证安全性。为增强安全性，本文在ε_1方案的加密思想上，利用向量集合来构建加密方案。

3.2 向量集的可计算加密方案

3.2.1 多项式系数向量环的构建

$\boldsymbol{Z}[x]$表示整系数多项式环，多项式$f(x)\in\boldsymbol{Z}[x]$可表示为

$$f(x)=a_0+a_1x+a_2x^2+\ldots+a_nx^n+\ldots=\sum\nolimits_i a_ix^i$$

其中，系数a_i均为整数。对于多项式$g(x)=x^n-1$，$g(x)\in\boldsymbol{Z}[x]$，$\boldsymbol{Z}[x]/(g(x))$代表由所有最高次数小于n的整系数多项式（形如$\sum_{i=0}^{n-1}a_ix^i$）构成的集合[18]。$\boldsymbol{Z}[x]/(g(x))$中多项式的运算为模$g(x)$运算，加法为两多项式同次项系数相加，乘法如下

$$\sum_{i=0}^{n-1}a_ix^i\sum_{i=0}^{n-1}b_ix^i \bmod g(x)=\sum_{i=0}^{n-1}d_ix^i \qquad (2)$$

其中，$d_s=\sum_{i+j\equiv s \bmod n}a_ib_j$。

由于$\boldsymbol{Z}[x]/(g(x))$中的多项式只有系数不同，因此可将其表示成系数的列向量，即$a_0+a_1x+\cdots+a_{n-1}x^{n-1}$可表示成$(a_0,a_1,\cdots,a_{n-1})^{\mathrm{T}}$，称为系数向量。

系数向量是$\boldsymbol{Z}^n$上的列向量，由此可得$\boldsymbol{Z}^n$到$\boldsymbol{Z}[x]/(g(x))$的一一映射，根据$\boldsymbol{Z}[x]/(g(x))$中多项式的加法和乘法重新定义$\boldsymbol{Z}^n$上的加法和乘法。

定义2 $\boldsymbol{Z}^n$的加法。$\boldsymbol{Z}^n$中向量相加(+)定义为

$$(a_0,\cdots,a_{n-1})^{\mathrm{T}}+(b_0,\cdots,b_{n-1})^{\mathrm{T}}=(d_0,\cdots,d_{n-1})^{\mathrm{T}} \quad (3)$$

其中，$d_s=a_s+b_s$，$0\leqslant s\leqslant n-1$。$\boldsymbol{Z}^n$的加法与通常的向量加法一致。

定义3 $\boldsymbol{Z}^n$的乘法。$\boldsymbol{Z}^n$中向量相乘(⊗)定义为

$$(a_0,\cdots,a_{n-1})^{\mathrm{T}}\otimes(b_0,\cdots,b_{n-1})^{\mathrm{T}}=(d_0,\cdots,d_{n-1})^{\mathrm{T}} \quad (4)$$

其中，$d_s=\sum_{i+j\equiv s \bmod n}a_ib_j$。

多项式环$\boldsymbol{Z}[x]$是交换环，其加法和乘法是满足交换率、结合律和分配律的，因此+和⊗也满足这些规律。此外，用$\boldsymbol{0}$表示零向量，$\forall \boldsymbol{a}=(a_0,\cdots,a_{n-1})^{\mathrm{T}}\in\boldsymbol{Z}^n$，$\boldsymbol{a}+\boldsymbol{0}=\boldsymbol{a}$，且$\exists -\boldsymbol{a}=(-a_0,\cdots,-a_{n-1})^{\mathrm{T}}\in\boldsymbol{Z}^n$，使得$\boldsymbol{a}+(-\boldsymbol{a})=\boldsymbol{0}$。

综上所述，$\boldsymbol{Z}^n$是一个交换环，称为系数向量环，其中零元为$\boldsymbol{0}$，$\forall \boldsymbol{a}\in\boldsymbol{Z}^n$，$\boldsymbol{a}$的逆元为$-\boldsymbol{a}$。

3.2.2 商环及代表元集合的构建

本节在构建的系数向量环上利用理想生成商环，并找出商环的代表元。本文借助理想格的性质实现该过程。

定义4 格。集合$\{\boldsymbol{Br}|\boldsymbol{B}\in\mathbb{Z}^{m\times n},\ \forall \boldsymbol{r}\in\mathbb{Z}^n\}$称为以$\boldsymbol{B}$为基的格，记为$\mathcal{L}(\boldsymbol{B})$。其中，$\boldsymbol{Br}$表示$m\times n$维矩阵$\boldsymbol{B}$与$n$维列矩阵$\boldsymbol{r}$相乘，$\boldsymbol{b}_i$是$\boldsymbol{B}$的第$i$列，$\boldsymbol{b}_i\in\boldsymbol{Z}^m$且$\boldsymbol{b}_0,\boldsymbol{b}_1,\cdots,\boldsymbol{b}_{n-1}$线性独立。显然，$\mathcal{L}(\boldsymbol{B})\subseteq\boldsymbol{Z}^m$。

定义5 循环矩阵。循环矩阵的首列（第0列）记为$\boldsymbol{b}_0$，第i+1列($\boldsymbol{b}_{i+1}$)是由第i列($\boldsymbol{b}_i$)的元素经循环下移得到（最后一行元素移到第0行）。因此，循环矩阵可由首列$\boldsymbol{b}_0$生成，记为$cycle(\boldsymbol{b}_0)$，$cycle(\boldsymbol{b}_0)$的第i行第j列元素为$b_{i-j \bmod n}$。

若格$\mathcal{L}(\boldsymbol{B})$的基$\boldsymbol{B}$是$n$维循环矩阵，则$\mathcal{L}(\boldsymbol{B})$是$\boldsymbol{Z}^n$中一个理想（证明见附录A），称为理想格，兼有理想和格的性质。如无特别说明，后文中的$\boldsymbol{B}$

均指循环矩阵。

作为 $\boldsymbol{Z}^n$ 中的理想，$\mathcal{L}(\boldsymbol{B})$可将 $\boldsymbol{Z}^n$ 划分为若干剩余类，构成商环 $\boldsymbol{Z}^n/\mathcal{L}(\boldsymbol{B})$。$\forall \boldsymbol{u}\in \boldsymbol{Z}^n$，$\boldsymbol{u}$ 所属剩余类$[\boldsymbol{u}]=\{\boldsymbol{u}+\boldsymbol{Br}|\forall \boldsymbol{r}\in \boldsymbol{Z}^n\}\in \boldsymbol{Z}^n/\mathcal{L}(\boldsymbol{B})$，$\boldsymbol{Z}^n/\mathcal{L}(\boldsymbol{B})$ 上的加法和乘法运算可分别表示为

$$[\boldsymbol{u}_1]+[\boldsymbol{u}_2]=[\boldsymbol{u}_1+\boldsymbol{u}_2]$$

$$[\boldsymbol{u}_1]\otimes[\boldsymbol{u}_2]=[\boldsymbol{u}_1\otimes\boldsymbol{u}_2]$$

接下来，需要为 $\boldsymbol{Z}^n/\mathcal{L}(\boldsymbol{B})$ 寻找代表元集合。$\boldsymbol{Z}^n/\mathcal{L}(\boldsymbol{B})$ 代表元集合并不唯一，下述定理给出了其中的一个。

定理 1　记集合 $\mathrm{P}(\boldsymbol{B})=\{\boldsymbol{Bx}|\forall\ x_i\in[-0.5,0.5)\}$，则对 $\forall\ \boldsymbol{u}\in \boldsymbol{Z}^n$，有如下性质。

1) 存在性。$\exists \boldsymbol{t}\in \mathrm{P}(\boldsymbol{B})\cap \boldsymbol{Z}^n$，$\exists \boldsymbol{r}\in \boldsymbol{Z}^n$ 使 $\boldsymbol{u}=\boldsymbol{t}+\boldsymbol{Br}$。

2) 唯一性。若 $\exists\ \boldsymbol{t}_1\in \mathrm{P}(\boldsymbol{B})\cap \boldsymbol{Z}^n$，$\exists\ \boldsymbol{r}_1\in \boldsymbol{Z}^n$ 使 $\boldsymbol{u}=\boldsymbol{t}_1+\boldsymbol{Br}_1$，则 $\boldsymbol{t}=\boldsymbol{t}_1$。

证明　首先证明存在性。文献[7]给出了计算 $\boldsymbol{t}$ 的方法，即 $\boldsymbol{t}=\boldsymbol{u}-\boldsymbol{B}\lfloor \boldsymbol{B}^{-1}\boldsymbol{u}\rceil$，显然 $\boldsymbol{t}\in \boldsymbol{Z}^n$，因此只需证 $\boldsymbol{t}\in \mathrm{P}(\boldsymbol{B})$。

令 $\boldsymbol{x}=\boldsymbol{B}^{-1}\boldsymbol{u}-\lfloor \boldsymbol{B}^{-1}\boldsymbol{u}\rceil$，$\boldsymbol{x}$ 中元素均在[−0.5, 0.5)区间内，则

$$\boldsymbol{t}=\boldsymbol{u}-\boldsymbol{B}\lfloor \boldsymbol{B}^{-1}\boldsymbol{u}\rceil=\boldsymbol{u}-\boldsymbol{B}(\boldsymbol{B}^{-1}\boldsymbol{u}-\boldsymbol{x})=\boldsymbol{Bx}$$

故 $\boldsymbol{t}\in \mathrm{P}(\boldsymbol{B})$。

然后证明唯一性。

令 $\boldsymbol{t}_1=\boldsymbol{Bx}_1$，则 $\boldsymbol{Bx}_1+\boldsymbol{Br}_1=\boldsymbol{Bx}+\boldsymbol{Br}$。

两边同乘 $\boldsymbol{B}^{-1}$，整理得 $\boldsymbol{x}_1-\boldsymbol{x}=\boldsymbol{r}-\boldsymbol{r}_1$。

$\boldsymbol{r}-\boldsymbol{r}_1$ 的元素为整数，而 $\boldsymbol{x},\boldsymbol{x}_1$ 中元素均在[−0.5, 0.5)区间内，则 $\boldsymbol{x}_1-\boldsymbol{x}$ 的元素均在(−1,1)区间内，因此，当且仅当 $\boldsymbol{x}_1-\boldsymbol{x}=\boldsymbol{0}$ 时，上式成立，此时 $\boldsymbol{t}=\boldsymbol{t}_1$。

证毕。

$\mathrm{P}(\boldsymbol{B})\cap \boldsymbol{Z}^n$ 是 $\boldsymbol{Z}^n/\mathcal{L}(\boldsymbol{B})$ 的代表元集合，且 $\forall\ \boldsymbol{u}\in \boldsymbol{Z}^n$，$[\boldsymbol{u}]_{\boldsymbol{B}}=\boldsymbol{u}-\boldsymbol{B}\lfloor \boldsymbol{B}^{-1}\boldsymbol{u}\rceil$。

3.2.3　向量集上的加密方案

通过前面的描述，已经得到交换环 $\boldsymbol{Z}^n$，商环 $\boldsymbol{Z}^n/\mathcal{L}(\boldsymbol{B})$ 及其代表元集合 $\mathrm{P}(\boldsymbol{B})\cap \boldsymbol{Z}^n$，从而可以得到 $\boldsymbol{Z}^n$ 上的加密方案ε_2：密钥为 $\boldsymbol{B}$，明文空间 $\mathcal{P}_{\varepsilon 2}=\mathrm{P}(\boldsymbol{B})\cap \boldsymbol{Z}^n$，密文空间 $C_{\varepsilon 2}=\boldsymbol{Z}^n$，加解密算法分别为

$$Enc_{\varepsilon 2}(\boldsymbol{B},\boldsymbol{t}):\boldsymbol{u}\leftarrow_{\mathrm{R}}[\boldsymbol{t}]$$

$$Dec_{\varepsilon 2}(\boldsymbol{B},\boldsymbol{u}):\boldsymbol{t}\leftarrow[\boldsymbol{u}]_{\boldsymbol{B}}$$

其中，$\boldsymbol{t}$ 和 $\boldsymbol{u}$ 分别表示明文和密文。若 $\boldsymbol{t}_1,\boldsymbol{t}_2\in \mathcal{P}_{\varepsilon 2}$，对应密文分别为 $\boldsymbol{u}_1,\boldsymbol{u}_2$，即 $\boldsymbol{u}_1\in[\boldsymbol{t}_1]$，$\boldsymbol{u}_2\in[\boldsymbol{t}_2]$，则密文加法为

$$Cal_{\varepsilon 2}((\boldsymbol{u}_1,\boldsymbol{u}_2),+)=\boldsymbol{u}_1+\boldsymbol{u}_2$$

密文乘法为

$$Cal_{\varepsilon 2}((\boldsymbol{u}_1,\boldsymbol{u}_2),\otimes)=\boldsymbol{u}_1\otimes\boldsymbol{u}_2$$

分析密文加法和乘法运算的同态性。对于明文混合运算 $\boldsymbol{t}_1\overset{1}{\circ}\boldsymbol{t}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{t}_r$，其中 $\overset{i}{\circ}\in\{+,\otimes\}$，其对应的密文运算结果为 $\boldsymbol{u}_1\overset{1}{\circ}\boldsymbol{u}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{u}_r$，则

$$Dec_{\varepsilon 2}(\boldsymbol{B},\boldsymbol{u}_1\overset{1}{\circ}\boldsymbol{u}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{u}_r)=\left[\boldsymbol{t}_1\overset{1}{\circ}\boldsymbol{t}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{t}_r\right]_{\boldsymbol{B}}\quad(5)$$

式(5)的推导见附录 B。若 $\boldsymbol{t}_1\overset{1}{\circ}\boldsymbol{t}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{t}_r\in\mathcal{P}_{\varepsilon 2}$，则

$$Dec_{\varepsilon 2}(\boldsymbol{B},\boldsymbol{u}_1\overset{1}{\circ}\boldsymbol{u}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{u}_r)=\boldsymbol{t}_1\overset{1}{\circ}\boldsymbol{t}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{t}_r$$

即ε_2支持加法和乘法的混合同态运算。

综上所述，只要向量的运算范围不超过 $\mathcal{P}_{\varepsilon 2}$，$\varepsilon_2$ 就是{+,⊗}上的混合同态加密方案，相比大多方案只支持单一运算类型，或不支持混合运算的缺陷，ε_2 方案有了较大改善。

3.3　正整数集上的可计算加密方案

虽然ε_2是{+,⊗}上的同态加密方案，但明文空间 $\mathrm{P}(\boldsymbol{B})\cap \boldsymbol{Z}^n$ 与通常使用的整数明文空间不一致，并不实用，需要进一步将其映射到整数明文空间。

任意整数 p 可以表示为多项式形式，如

$$p=f(x)=a_0+a_1x+a_2x^2\ldots+a_{n-1}x^{n-1}+\ldots$$

设 x 最高次数为 n−1，则 $f(x)\in \boldsymbol{Z}[x]/(g(x))$。给定 x 的值为 2，则任意 $\boldsymbol{Z}^n$ 中的系数向量通过 $\boldsymbol{Z}[x]/(g(x))$ 对应一个整数，记为映射 $\varphi:\boldsymbol{Z}^n\rightarrow[0,2^n)$。为使整数在 $\boldsymbol{Z}^n$ 中的映射唯一，限定映射时 a_i 取值只能为 0 或 1，从而可将[0, 2^n)区间的整数唯一地映射为$\{0,1\}^n$中的向量，记为 δ: $[0,2^n)\rightarrow\{0,1\}^n$。显然，$\varphi(\delta(p))=p$。

对于整数 $p_1,p_2,\cdots,p_r\in[0,\ 2^n)$，映射到$\{0,1\}^n$中的向量分别为 $\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_r$，即 $\boldsymbol{t}_i=\delta(p_i)$。其混合运算 $p_1\overset{1}{\Diamond}p_2\overset{2}{\Diamond}\cdots\overset{r-1}{\Diamond}p_r$，$\overset{i}{\Diamond}\in\{+,\times\}$ 可表示为多项式的混合运算，显然，p_i 表示成的多项式最高次不超过 n−1。此外，若 $p_1\overset{1}{\Diamond}p_2\overset{2}{\Diamond}\cdots\overset{r-1}{\Diamond}p_r\in[0,2^n)$，则结果多项式最高次也不超过 n−1（若最高次超过 n，则所能表示的最小整数值为 2^n，已经超出$[0,2^n)$的范围）。此时，多项式的混合运算又可表示为系数向量的混合运算形式，即 $\boldsymbol{t}_1\overset{1}{\circ}\boldsymbol{t}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{t}_r=\boldsymbol{t}$，其中，若

$\overset{i}{\Diamond}=+$，则$\overset{i}{\circ}=+$；若$\overset{i}{\Diamond}=\times$，则$\overset{i}{\circ}=\otimes$。虽然可能系数向量的混合运算结果$\boldsymbol{t}\notin\{0,1\}^n$，但

$$\varphi\left(\boldsymbol{t}_1\overset{1}{\circ}\boldsymbol{t}_2\overset{2}{\circ}\cdots\overset{r-1}{\circ}\boldsymbol{t}_r\right)=\varphi\left(\delta(p_1)\overset{1}{\Diamond}\delta(p_2)\overset{2}{\Diamond}\cdots\overset{r-1}{\Diamond}\delta(p_r)\right)$$
$$=p_1\overset{1}{\Diamond}p_2\overset{2}{\Diamond}\cdots\overset{r-1}{\Diamond}p_r$$

因此，整数的混合运算也可用系数向量的混合运算来表示。至此，得到了CESIL加密方案的原型：首先将整数映射为$\{0,1\}^n$中的向量，然后利用ε_2方案加密，并进行密文运算。只要$\{0,1\}^n$以及运算结果$\boldsymbol{t}$均在P($\boldsymbol{B}$)内，则可保证CESIL方案在加法和乘法运算上的同态性。

4 CESIL方案的实现

4.1 明文空间与密钥的选择

CESIL方案要求$\{0,1\}^n$，$\boldsymbol{t}\in$P($\boldsymbol{B}$)，因此密钥的选择十分重要，借助文献[8]的定理，可以给出$\{0,1\}^n$，$\boldsymbol{t}\in$P($\boldsymbol{B}$)的一个条件。

定理2 $\boldsymbol{B}\in\mathbb{Z}^{n\times n}$，$\boldsymbol{t}\in\mathbb{Z}^n$，记$\boldsymbol{B}^*=(\boldsymbol{B}^{-1})^{\mathrm{T}}$，若$\|\boldsymbol{t}\|<1/\left(2\|\boldsymbol{B}^*\|\right)$，则$\lfloor\boldsymbol{B}^{-1}\boldsymbol{t}\rceil=\boldsymbol{0}$，即$\boldsymbol{t}\in$P($\boldsymbol{B}$)。其中，$\|\boldsymbol{t}\|$是$\boldsymbol{t}$的欧几里得长度，$\|\boldsymbol{B}^*\|=\max\{\|\boldsymbol{b}_i^*\|\}$，$\boldsymbol{b}_i^*$为$\boldsymbol{B}^*$的第$i$列。

证明 见文献[8]。

据定理2可知，若$\|(1,1,\cdots,1)^{\mathrm{T}}\|=\sqrt{n}<1/\left(2\|\boldsymbol{B}^*\|\right)$，即$\|\boldsymbol{B}^*\|<1/\left(2\sqrt{n}\right)$，则$\{0,1\}^n\subseteq$P($\boldsymbol{B}$)$\cap\mathbb{Z}^n$。

此外，假设$\boldsymbol{t}$所对应的明文为N，$N\leqslant 2^n$，则$\max\{\|\boldsymbol{t}\|\}=\|(N,0,\cdots,0)^{\mathrm{T}}\|=N$，要保证运算在明文空间$[0,N)$内的同态性，则根据定理2，需要保证

$$\|\boldsymbol{B}^*\|<1/(2N) \tag{6}$$

如果运算结果超出$[0,N)$，则方案可能不再满足同态性，N称为失效点。

综上所述，在CESIL方案中，若给定明文空间$[0,N)\cap[0,2^n)\cap\mathbb{Z}$，密钥$\boldsymbol{B}$需满足式(6)。密文空间是$\mathbb{Z}^n$的一个子集，记为$C_{\mathrm{Bin}}$，且$C_{\mathrm{Bin}}=\bigcup_{\boldsymbol{t}\in\{0,1\}^n}[\boldsymbol{t}]$。

4.2 利用快速傅里叶变换改进CESIL

CESIL方案涉及到许多矩阵和向量的运算，有些复杂度为$O(n^2)$，有些甚至为$O(n^3)$。为提高运算效率，可以利用快速傅里叶变换(FFT)来改进CESIL方案。

4.2.1 循环矩阵的逆

利用高斯消元法计算矩阵的逆矩阵，时间复杂度为$O(n^3)$，而利用FFT算法可以在$O(n\log n)$时间复杂度内计算出循环矩阵的逆矩阵[20]。

对于循环矩阵$\boldsymbol{B}=cycle(\boldsymbol{b})$，其逆矩阵$\boldsymbol{B}^{-1}$也为循环矩阵，记为$\boldsymbol{B}^{-1}=cycle(\boldsymbol{b}^{-1})$，则

$$\boldsymbol{b}^{-1}=ifft\left(\left(fft(\boldsymbol{b})\right)^{-1}\right) \tag{7}$$

其中，$\boldsymbol{v}^{-1}$表示分别对向量$\boldsymbol{v}$的每个元素求−1次方，$ifft$为fft的逆变换，时间复杂度也为$O(n\log n)$。

在生成循环矩阵作为密钥时，只需生成其首列$\boldsymbol{b}$，然后根据式(7)计算逆元，若$fft(\boldsymbol{b})$的某个元素为0，则矩阵的逆矩阵不存在，需要重新生成新的向量作为首列。

4.2.2 系数向量的乘法

由附录A可知，$\boldsymbol{b}\otimes\boldsymbol{r}=cycle(\boldsymbol{b})\boldsymbol{r}$，而无论根据此式或式(4)计算系数向量的乘法，复杂度都为$O(n^2)$。容易证明，$cycle(\boldsymbol{b})\boldsymbol{r}$是$cycle(\boldsymbol{b})cycle(\boldsymbol{r})$的首列，而后者正是2个循环矩阵的乘法，可以利用FFT算法在$O(n\log n)$时间复杂度内计算[20]，因此，$\boldsymbol{b}\otimes\boldsymbol{r}$可表示如下

$$\boldsymbol{b}\otimes\boldsymbol{r}=ifft\left(\left(fft(\boldsymbol{b})\right)\left(fft(\boldsymbol{r})\right)\right) \tag{8}$$

其中，$\boldsymbol{u}\times\boldsymbol{v}$表示分别将向量$\boldsymbol{u}$和$\boldsymbol{v}$相同位置的元素相乘。

可以利用式(8)进行密文的乘法运算，此外，加密和解密时密钥与列向量的乘法也可利用式(8)进行计算。更进一步地，由于密钥生成，加密及解密过程都只需密钥的首列$\boldsymbol{b}$参与，因此，可以用首列$\boldsymbol{b}$代替密钥$\boldsymbol{B}$，从而使密钥空间大大减少，只需$O(n)$的空间复杂度。

4.3 算法描述

基于以上的分析，本节给出CESIL加密方案的算法描述。

4.3.1 密钥生成算法 *Gen*

密钥生成算法$Gen(n, N)$返回密钥$\boldsymbol{b}$和$\boldsymbol{b}^{-1}$。n指定了向量的维度，n越大，计算复杂度越高，算法也越安全。此外，n也限定了明文的最大范围为$[0,2^n)$，超出此范围的明文在加密时无法映射为n维向量。N为失效点，满足$\sqrt{n}\leqslant N\leqslant 2^n$，$[0,N)$内的运算能保证同态性，超出$[0,N)$的运算则可能失效。

算法1 密钥生成算法$Gen(n,N)$

输入：整数n,N

输出：n维向量$\boldsymbol{b}$和$\boldsymbol{b}^{-1}$

1) DO{

2)　随机生成正整数范围内的 n 个元素构成数组 $\boldsymbol{b}$

3)　$\boldsymbol{b\_fft}=fft(\boldsymbol{b})$;

4)　IF　$\boldsymbol{b\_fft}$ 含 0 元素

5)　　CONTINUE;

6)　$\boldsymbol{b}^{-1}=iffi(\boldsymbol{b\_fft}^{-1})$;

7)　$len \leftarrow \left\|\boldsymbol{b}^{-1}\right\|$;

8)　}WHILE(len >= 1 / (2N))

第 7)步本应为计算 $\left\|\boldsymbol{B}^{*}\right\|$，而当 $\boldsymbol{B}$ 为循环矩阵时，$\boldsymbol{B}^{*}$ 也为循环矩阵，且各列的长度相等，因此 $\left\|\boldsymbol{B}^{*}\right\|=\left\|\boldsymbol{b}^{-1}\right\|$。

4.3.2　加密算法 *Enc*

加密算法 $Enc(\boldsymbol{b}, p)$用密钥 $\boldsymbol{b}$ 加密明文 p，返回的密文为 n 维列向量。

算法 2　加密算法 $Enc(\boldsymbol{b}, p)$

输入：密钥 $\boldsymbol{b}$, 正整数明文 p

输出：n 维列向量 $\boldsymbol{c}$

1)　FOR (i = 0; i < n; i++){

2)　　$\boldsymbol{c}[i] \leftarrow ((q >> i)$ & 0x01);

3)　} //$\boldsymbol{c}[i]$存储 p 二进制表示的第 i 位

4)　随机生成整数范围内的 n 维列向量 $\boldsymbol{r}$，且 $\boldsymbol{r}!=\boldsymbol{0}$;

5)　$\boldsymbol{c} \leftarrow \boldsymbol{c}+\boldsymbol{b} \otimes \boldsymbol{r}$

整数转化为向量的过程实际上是用数组来记录二进制的每一位，利用位运算可以提高效率。第 5)步可根据式(8)计算。

4.3.3　解密算法 *Dec*

解密算法 $Dec(\boldsymbol{b}, \boldsymbol{b}^{-1}, \boldsymbol{c})$用密钥 $\boldsymbol{b}$ 和 $\boldsymbol{b}^{-1}$ 解密密文 $\boldsymbol{c}$，返回明文。

算法 3　解密算法 $Dec(\boldsymbol{b}, \boldsymbol{b}^{-1}, \boldsymbol{c})$

输入：密钥 $\boldsymbol{b}, \boldsymbol{b}^{-1}$, 密文 $\boldsymbol{c}$

输出：正整数 p

1)　$\boldsymbol{c} \leftarrow \boldsymbol{c}-\boldsymbol{b} \otimes \lfloor \boldsymbol{b}^{-1} \otimes \boldsymbol{c} \rceil$;

2)　$p \leftarrow 0$;

3)　FOR (i = 0; i < n; i++){

4)　　$p \leftarrow p+\boldsymbol{c}[i]2^{i}$;

5)　}//利用 $\boldsymbol{c}[i]$还原 p

经过运算的密文解密出的向量不一定在 $\{0,1\}^{n}$ 中，因此解密不能使用位运算，但可以通过映射 φ 将其转化为明文。

4.3.4　运算算法 *Cal*

运算算法 $Cal(\boldsymbol{c}_1, \boldsymbol{c}_2, o_p)$将利用所给操作符 o_p 对密文直接进行运算，$o_p \in \{+, -, \times, /\}$，减法是加法的逆运算，减 $\boldsymbol{c}_2$ 相当于加$-\boldsymbol{c}_2$, 但由于减法使向量元素可能为负，当运算次数增加时，会使 4.1 节所述 $\max\{\|\boldsymbol{t}\|\} > \left\|(N,0,\cdots,0)^{\mathrm{T}}\right\|$，从而使运算结果在[0, N)内时也不一定满足同态性。因此，方案支持有限次的减法。此外，目前方案不支持任意密文的除法，特殊情况下，对于密文 $\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$，若 $\exists \boldsymbol{c} \in \boldsymbol{Z}^{n}$ 使 $\boldsymbol{c}_1 \otimes \boldsymbol{c}=\boldsymbol{c}_2$，由附录 A 可知，$\boldsymbol{c}=(cycle(\boldsymbol{c}_1))^{-1}\boldsymbol{c}_2=\boldsymbol{c}_1^{-1} \otimes \boldsymbol{c}_2$，并称此情况为可除。

算法 4　运算算法 $Cal(\boldsymbol{c}_1, \boldsymbol{c}_2, o_p)$

输入：密文操作数 $\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$, 操作符 o_p

输出：密文结果 $\boldsymbol{c}$

1)　IF o_p == "+"

2)　　$\boldsymbol{c} \leftarrow \boldsymbol{c}_1+\boldsymbol{c}_2$

3)　IF o_p == "−"

4)　　$\boldsymbol{c} \leftarrow \boldsymbol{c}_1+(-\boldsymbol{c}_2)$

5)　IF o_p == "×"

6)　　$\boldsymbol{c} \leftarrow \boldsymbol{c}_1 \otimes \boldsymbol{c}_2$

7)　IF o_p == "/" {

8)　　$\boldsymbol{c} \leftarrow \boldsymbol{c}_2^{-1} \otimes \boldsymbol{c}_1$

9)　　IF $\boldsymbol{c} \notin \boldsymbol{Z}^{n}$ //不可除

10)　　　$\boldsymbol{c} \leftarrow$ NULL

11)　}

由 4.1 节描述可知，当原明文运算结果超出失效点时，方案可能失效。因此方案适合变量（包括中间值）在有限空间内的服务。失效点 N 的值需要大于变量的最大值，以防止解密结果出错。该最大值可由服务提供者根据其服务特点给出，也可由用户根据实际情况设定。设定好合适的失效点后，服务者可以对密文进行任意的加法或乘法运算，无论密文变为怎样的形式，解密都可得正确的明文结果。

假设服务 S 由算术运算构成，在对密文进行运算时，可调用 *Cal* 算法将原明文的运算转换为相应的对密文的运算，并计算出密文结果。例如，S 为记录用户余额的服务，初始余额为 a, 用户收入 b, 并 c 次支出 d。在明文情况下，服务 S 返回给用户的当前余额应为 $a+b-c \times d$。

为保护隐私信息，用户传给 S 的数据均为密文，假设 a,b,c,d 的密文分别为 $\boldsymbol{c}_a,\boldsymbol{c}_b,\boldsymbol{c}_c,\boldsymbol{c}_d$，则

收入 b 后，余额 $\boldsymbol{c}_1 \leftarrow Cal((\boldsymbol{c}_a, \boldsymbol{c}_b), "+")$;

c 个 d 的总支出为 $\boldsymbol{c}_2 \leftarrow Cal((\boldsymbol{c}_c, \boldsymbol{c}_d), "\times")$;

返回给用户的余额为 $\boldsymbol{c}_3 \leftarrow Cal((\boldsymbol{c}_1, \boldsymbol{c}_2), "-")$。

则 c_3 为最终 S 的结果，经解密，用户可得到正确余额，即 $a+b-c\times d$。而在整个过程中，S 并不知道真实的数值，因而既完成了服务，又有效地保护了用户的隐私。

如果服务包含除法运算，则只有在可除情况下才能得到正确的解。

5 CESIL 方案的理论分析

本节将从安全性、复杂性等方面对 CESIL 方案进行理论分析。

5.1 安全性

首先，维度 n 和失效点 N 越大，攻击者通过穷举攻击获得密钥的可能性越低。密钥元素的值一般与 N 在数量级上接近（具体见 6.1 节），给定 n 和 N，密钥大约有 N^n 种选取可能。因此，如果 n 和 N 的值足够大，攻击者想要通过穷举密钥来攻击 CESIL 方案的成功率就可以忽略。事实上，n 不需要取很大的值，如 $N=2^{20}$，$n=32$，则 $N^n>2^{512}$，已经能够满足一般的安全性要求。

其次，攻击者无法通过获取有限个数的明/密文对解方程来获得密钥。对于一对明/密文，可以得到 n 个方程，其中的密钥 $\boldsymbol{B}$ 和随机向量 $\boldsymbol{r}$ 是未知数，即有 $2n$ 个未知数。假设攻击者 A 获得 m 对明/密文，则可以得到 mn 个方程，但仍然有 $n+mn$ 个未知数。对于此类解方程问题，通常可以将其转化为求格的最小向量问题，并利用格规约的方式求解。但这需要事先知道格的一个基，对于非对称加密算法，其公钥即为一个已知的基，因此在公钥维度较低时可能被破解。而 CESIL 为对称加密，因此，攻击者事先无法知道格中的任意一个基，无法利用格规约的方式破解 CESIL 的安全性。

最后，证明 CESIL 方案是语义安全的。CESIL 方案的安全性是基于理想陪集问题(ICP)[8]的难解性的，根据文献[8]，并结合本文实际情况，ICP(ideal closet problem)可表示为定义 6。

定义 6 给定 $\mathcal{Z}^n$ 和 $\boldsymbol{B}$，可得 $\mathcal{C}_{\text{Bin}}\subseteq\mathcal{Z}^n$（$\mathcal{C}_{\text{Bin}}$ 相当于文献[8]中 ICP 定义的 R，而 $\boldsymbol{B}$ 相当于 $\boldsymbol{B}_J^{\text{pk}}$），令 $\beta\leftarrow_{\text{R}}\{0,1\}$，如果 $\beta=0$，则 $\boldsymbol{c}\leftarrow_{\text{R}}\mathcal{C}_{\text{Bin}}$，$\boldsymbol{t}\leftarrow[\boldsymbol{c}]_{\boldsymbol{B}}$，否则，$\boldsymbol{t}\leftarrow_{\text{R}}\{0,1\}^n$。问题：已知 $\boldsymbol{t}$ 和 $\boldsymbol{B}$，猜测 β 的值。

$\mathcal{C}_{\text{Bin}}$ 中的元素是 $\{0,1\}^n$ 中向量与理想陪集($\boldsymbol{B}$)中向量的和，因此 $\{0,1\}^n$ 可以看作是 $\mathcal{C}_{\text{Bin}}$ 模 $\boldsymbol{B}$ 后的集合。ICP 的困难性是指在已知 $\boldsymbol{t}$ 和 $\boldsymbol{B}$ 后，无法确定 $\boldsymbol{t}$ 是先选择 $\mathcal{C}_{\text{Bin}}$ 中任意元素后再模 $\boldsymbol{B}$ 得到的，还是从 $\{0,1\}^n$ 中随机选择的。

若攻击者 $\mathcal{A}$ 能够以概率 ρ 破坏 CESIL 的语义安全性，则存在算法 $\mathcal{H}$ 能以概率 $\rho'=\rho/2+1/4$ 解决 ICP 问题。

证明 将 $\boldsymbol{t}$ 和 $\boldsymbol{B}$ 给 $\mathcal{H}$，$\mathcal{H}$ 可为如下算法。

令 $\beta'\leftarrow_{\text{R}}\{0,1\}$，$\boldsymbol{t}_{\beta'}\leftarrow\boldsymbol{t}$，$\boldsymbol{t}_{1-\beta'}\leftarrow_R\{0,1\}^n$。令 $\boldsymbol{c}'\leftarrow Enc(\boldsymbol{B},\boldsymbol{t}_{\beta'})$，将 $\boldsymbol{c}'$给 $\mathcal{A}$，由 $\mathcal{A}$ 返回猜测 β''，则 $\mathcal{H}$ 猜测 $\beta=\beta'\odot\beta''$，$\odot$ 为同或运算。

分析上述算法。当 $\beta=1$ 时，$\boldsymbol{t}$ 是 $\{0,1\}^n$ 上的随机元素，无论 β 值为多少，$\boldsymbol{t}_{\beta'}$ 均为 $\{0,1\}^n$ 上的随机元素。而 CESIL 的明文与 $\{0,1\}^n$ 上的向量一一对应，若 $\boldsymbol{t}_{\beta'}$ 对应的明文为 $p_{\beta'}$，则 $\boldsymbol{c}'$是 $p_{\beta'}$ 的密文。此时，$\mathcal{A}$ 能以概率 ρ 猜对 β'，即 $\beta'=\beta''$的概率为 ρ，则 $\mathcal{H}$ 猜对 $\beta=\beta'\odot\beta''=1$ 的概率为 ρ。

当 $\beta=0$ 时，$\boldsymbol{t}=[\boldsymbol{c}]_{\boldsymbol{B}}$，$\boldsymbol{c}'$是$[\boldsymbol{c}]$上的均匀分布，$\boldsymbol{c}$ 为 $\mathcal{C}_{\text{Bin}}$ 上的均匀分布，则 $\boldsymbol{c}'$也是 $\mathcal{C}_{\text{Bin}}$ 上的均匀分布，与明文无关。此时，$\mathcal{A}$ 猜对 β'的概率无异于随机猜测(1/2)，则 $\beta'\neq\beta''$的概率为 1/2，$\mathcal{H}$ 猜对 $\beta=\beta'\odot\beta''=0$ 的概率为 1/2。

因此，$\mathcal{H}$ 解决 ICP 问题的概率 ρ'为

$$\rho\mathrm{P}(\beta=1)+1/2\,\mathrm{P}(\beta=0)=\rho/2+1/4$$

证毕。

若 $\mathcal{A}$ 破坏 CESIL 语义安全性的优势概率 $\rho-1/2$ 是不可忽略的，则 $\mathcal{H}$ 解决 ICP 问题的优势概率 $\rho'-1/2$ 也不可忽略，这与 ICP 问题难解矛盾，因此，$\rho-1/2$ 是可忽略，即 CESIL 方案是语义安全的。

5.2 复杂性

5.2.1 时间复杂度

1) 密钥生成算法。在生成密钥过程中，生成 n 维向量 $\boldsymbol{b}$ 的时间复杂度为 $O(n)$，求 $\boldsymbol{b}^{-1}$ 的时间复杂度为 $O(n\log n)$，计算 $\|\boldsymbol{b}^{-1}\|$ 的复杂度为 $O(n)$。综上所述，密钥生成算法的时间复杂度为 $O(n\log n)$。

2) 加密算法。在加密过程中，整数转换为 n 维向量 $\boldsymbol{t}$ 的时间复杂度为 $O(n)$，生成 n 维随机矩阵 $\boldsymbol{r}$ 的时间复杂度为 $O(n)$，计算 $\boldsymbol{t}+\boldsymbol{b}\otimes\boldsymbol{r}$ 的时间复杂度为 $O(n\log n)$。综上所述，加密算法的时间复杂度为 $O(n\log n)$。

3) 解密算法。在解密过程中，计算 $\boldsymbol{c}-\boldsymbol{b}\otimes\lfloor\boldsymbol{b}^{-1}\otimes\boldsymbol{c}\rceil$ 的时间复杂度为 $O(n\log n)$，向量转换为整数的时间复杂度为 $O(n)$，则解密算法的时间复杂度为 $O(n\log n)$。

4) 运算算法。运算主要包括加法（+）和乘法

（⊗），对于 2 个 n 维密文向量 $\boldsymbol{c}_1$、$\boldsymbol{c}_2$、$\boldsymbol{c}_1+\boldsymbol{c}_2$ 的时间复杂度为 $O(n)$；$\boldsymbol{c}_1 \otimes \boldsymbol{c}_2$ 时间复杂度为 $O(n\log n)$。

5.2.2　密钥和密文长度

密钥 $\boldsymbol{b}$ 为 n 维列向量，因此所占空间为 $O(n)$。整数的明文经加密后，成为 n 维密文向量，因此密文的长度大于明文的长度。假设明文长度为 l，密文向量的每个元素长度为 l'，则密文向量所占空间为 $O(nl')$，密文比明文增长了 nl'/l 倍。

6　实验

本节将通过实验测试 CESIL 方案的性质，并通过与 CESVMC、Unpadded-RSA（简称 URSA）和 Paillier 方案的对比测试 CESIL 方案的性能。实验环境为 Linux 平台，主机配置双核 3.0 GHz 主频的 CPU 和 4 GB 内存。

6.1　密钥取值与失效点的关系

在给定 N 后，为满足式(6)，可能需要多次生成 $\boldsymbol{b}$，本实验测试 $\boldsymbol{b}$ 中整数的最佳取值范围，以减少生成 $\boldsymbol{b}$ 的次数，提高密钥生成算法的效率。

$\boldsymbol{b}$ 由 n 个整数组成，一般地，整数越大，$\left\|\boldsymbol{b}^{-1}\right\|$ 越小，相应地，失效点 N 可取的值也越大。用 $b_{\max}$ 表示密钥中元素取值的最大值，计算 $1/\left(2\left\|\boldsymbol{b}^{-1}\right\|\right)$，该值为当前密钥所能满足的最大 N 值，记为 $N_{\max}$。相比 $b_{\max}$ 和 $N_{\max}$ 的具体值，则更关心它们的二进制位数，分别取位数为 8~64 bit 的 $b_{\max}$，计算 $\log N_{\max}$，多次实验后取平均值，结果如图 2 所示。

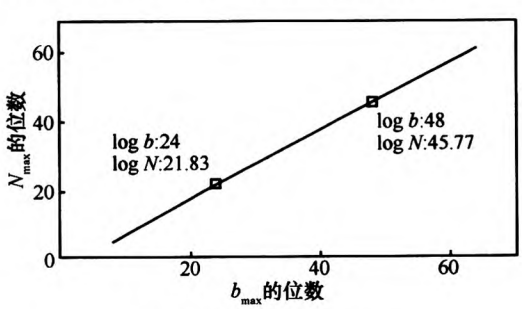


图 2　密钥与 N 位数的关系

由图 2 可知，$b_{\max}$ 与 $N_{\max}$ 的位数成正比，$b_{\max}$ 越大，$N_{\max}$ 越大，即在密钥生成时，输入的 N 越大，所需 $b_{\max}$ 也越大。此外，由图中的数值可知，$b_{\max}$ 的位数比 $N_{\max}$ 多 2 到 3 位，因此在密钥生成时，给定 N 值，密钥中整数的取值范围应至少为区间(0, $8N$)。此时，平均 1.07 次即可生成满足式(6)的可逆循环矩阵 $\boldsymbol{B}$。

6.2　效率

本节测试 CESIL 方案的运行效率，并与 CESVMC、URSA 和 Paillier 进行对比。CESVMC 的密钥也为方阵；URSA 的公钥记为(p_{ke}, p_{kn})，私钥记为 s_{kd}；Paillier 的公钥记为(p_{kg}, p_{kn})，私钥记为($s_{k\lambda}$, $s_{k\mu}$)。n 在 CESIL 和 CESVMC 中表示密钥的维度，在 URSA 和 Paillier 中表示公钥 p_{kn} 的长度（二进制位数）。虽然 n 的表示不同，但 n 越大，方案的安全性越高，因此可统称 n 为安全参数。需要注意的是，在相同的 n 值下，URSA 和 Paillier 的安全性（针对穷举攻击）远远低于 CESIL 和 CESVMC，本文只是将其作为对比实验的一个参照。

6.2.1　密钥生成算法

取不同的 n 值，测试 4 个方案密钥生成算法的运行时间，多次实验取平均值，如图 3 所示。

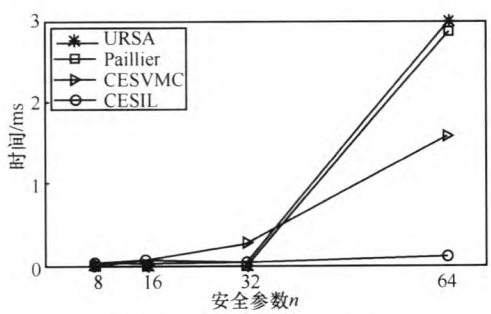


图 3　密钥生成算法的时间效率对比

由图 3 可知，在 n 大于 32 以后，URSA 和 Paillier 的密钥生成算法运行时间骤增，CESVMC 方案也有较大增加，而 CESIL 方案密钥生成算法因采用 FFT 求逆，其效率较高，且随着 n 的增长，算法的时间增幅也较小。

6.2.2　加密算法

取不同的 n 值，测试 CESIL 和 CESVMC 方案加密算法的运行时间，实验结果如图 4 所示。USRA 的加密时间与公钥 p_{ke} 有关($O(p_{ke})$)，一般 p_{ke} 取值较小，本实验取临近 $2n$ 的值作为参考。Paillier 的加密时间与要加密的明文以及公钥 p_{kn} 有关($O(p_{kn})$)，即使 n=16 时的加密时间已超过 0.6 ms，远远大于其他方案，因此不参与比较。

由图 4 可知，3 个方案加密算法效率都很高，相比而言，USRA 方案具有更高的加密效率。随着 n 的增大，CESIL 和 CESVMC 的加密时间都在增加，CESIL 方案在明文转化为向量时使用位运算，因此效率略高于 CESVMC 方案。

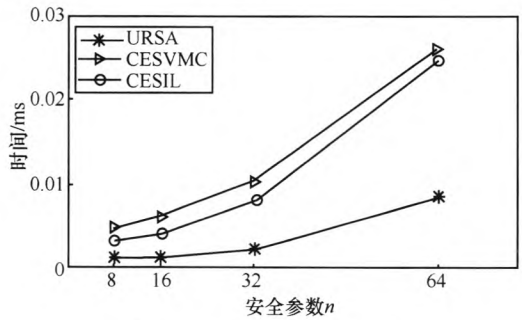


图 4 加密算法的时间效率对比

6.2.3 运算算法

密文的运算算法包括加法运算和乘法运算。取不同的 n 值，随机选择明文进行运算，并分别利用 CESIL、CESVMC、URSA 和 Paillier 方案加密明文，对密文进行相同运算（URSA 只进行乘法运算，Paillier 只进行加法运算），测试运算时间如图 5 所示。

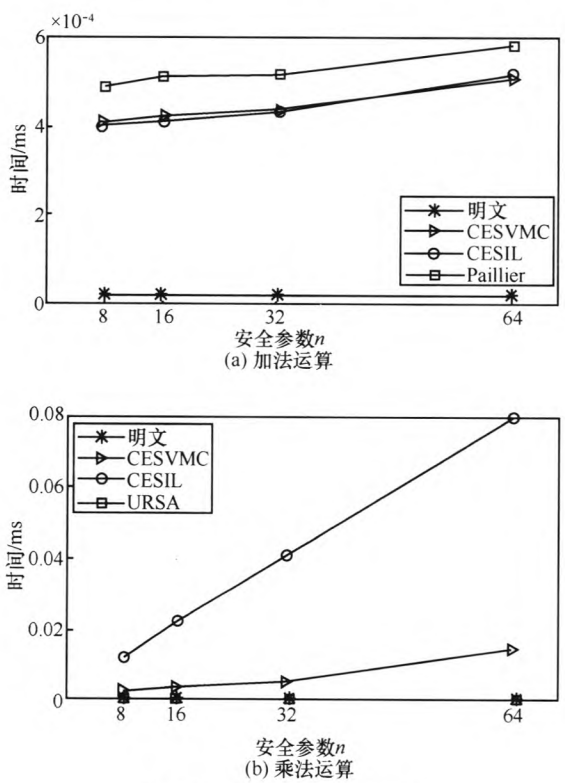


图 5 运算算法的时间效率对比

由图 5 可知，密文加法运算效率都很高，基本与明文的加法相同（小于 1 μs），相较而言，CESIL 加法效率要高于 Paillier。CESIL 方案的密文乘法效率虽然低于其他方案，但也保持较高的效率(小于 1 ms)，对于服务器来说是可以接受的。

6.2.4 解密算法

在 CESVMC 方案中，经过不同运算密文的解密方法不同。根据密文经过的不同运算，可将密文分为原密文（未经过运算或经过加减运算的密文）、乘法密文和除法密文 3 类。不同的密文解密方法不同，解密时需要指定密文类别，而 CESIL 不用关心密文的类别，所有密文解密方法一致。USRA 的解密时间与私钥 s_{kd} 有关($O(s_{kd})$)，Paillier 的解密时间与私钥 $s_{k\lambda}$ 有关($O(s_{k\lambda})$)，s_{kd} 和 $s_{k\lambda}$ 取值都接近 p_{kn}，即使 n=16 时 2 个方案的解密时间已超过 0.5 ms，n=32 时更是超过 1 s，都远远高于 CESIL 方案，因此不参与比较。图 6 是 CESIL 与 CESVMC 2 个方案解密算法的效率对比，其中，CESVMC 解密包括解密原密文和乘法密文 2 种情况。

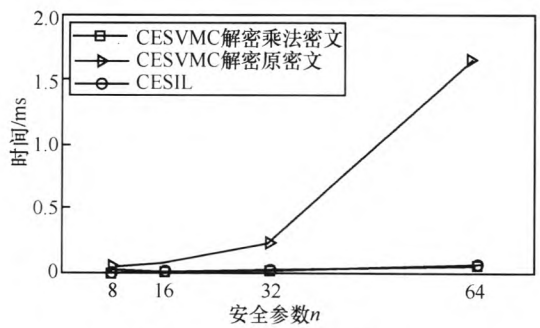


图 6 解密算法的时间效率对比

由图 6 可知，CESIL 方案解密算法效率与 CESVMC 方案解密原密文的效率几乎相同，但比 CESVMC 方案解密乘法密文的效率高很多，且随着 n 的增加，解密时间增幅较小。当 n=64 时，解密时间依然小于 0.05 ms。

对于一般性的安全要求，USRA 和 Paillier 所需 n 值为 512、1 024 或更大，这时的算法运行时间会很长，甚至对用户来说无法接受。而总体来说，CESIL 方案的各算法运行效率都较高，因而有效减少因加密而带来的额外时间开销，从而更适用于计算服务中的隐私保护。

6.3 密钥与密文长度

除了效率，密钥及密文的长度（所占空间）也是加密方案需要考虑的重要性能，密钥和密文长度越小，所需存储空间及通信开销也越小。CESVMC 方案密钥以及乘法密文都为 n×n 的矩阵，密文长度会随着 n 的增长以几何级数增长。CESIL 方案的密钥及密文均为 n 维列向量，因此所占空间更小。取不同的 n 值，并随机生成 1 024 个 n 位（二进制长度）的明文，分别测试在不同的 n 下，CESIL 和 CESVMC 2 个方案密钥及密文所占空间（密钥及密

文均使用 64 位数据类型存储，即 l=64），结果如表 2 所示。其中“CVO”表示 CESVMC 方案的原密文及加法密文，“CVM”表示 CESVMC 方案的乘法密文。

表 2　　密钥及密文长度对比

n	密钥长度/kbyte			明文或密文长度/kbyte		
	CESIL	CESVMC	明文	CESIL	CVO	CVM
8	0.125	1	1	64	64	512
16	0.25	4	2	128	128	2 048
32	0.5	16	4	256	256	8 192
64	1	64	8	512	512	32 768

由表 2 可知，CESIL 以及 CESVMC 密钥和密文的长度都会随 n 的增加而增加。此外，USRA 和 Paillier 方案在 n=512 时(与 CESIL 方案 n=8 时安全性相同)的密钥长度分别为 0.187 5 kbyte 和 0.25 kbyte，1 024 个密文的总长度分别为 64 kbyte 和 128 kbyte。因此，相比而言，CESIL 方案密钥长度更小。CESIL 密文长度也最小，且比 CESVMC 方案更加稳定，即使运算后，密文长度仍然不变，不会增加用户接收密文结果数据时的通信开销，因此更适用于计算服务。

6.4　方案适用情况

现有的加密方案还很难实现全同态，会存在方案不适用的情况，如运算类型或运算次数的限定。随机选取明文，利用不同方案加密并进行运算，若结果满足式(1)，则继续选取明文加密并计算，直至结果不满足式(1)或运算结果超出明文空间。选择不同的运算，可以得到各方案对这些运算的适用情况，如表 3 所示。

表 3　　各方案对不同运算的适用性对比

o_p	+	×	−	/	+×
URSA	×	√	×	×	×
Paillier	√	×	×	×	×
CESVMC	√	1 次	√	1 次	×
CESIL	√	√	>1 次	可除	√

注：√ 表示支持无限次操作　× 表示不支持此操作

由表 3 可知，相比其他方案，CESIL 方案适用的运算类型更多，不但支持任意次数的加法和乘法及加乘混合运算，也能支持多次减法运算和可除情况下的除法运算，因此能够更多地满足实际应用的需要。虽然 CESIL 不支持任意次数的减法，但在测试实验中，CESIL 方案仍可支持平均 72.23 次的加减乘混合运算。因此，对于含有少量减法的应用，CESIL 方案也可以适用。

综合所有实验结果可知，CESIL 方案解决了已有加密方案不支持多次乘法及加、乘混合运算的问题。此外，相比已有同态加密方案而言，CESIL 方案的总体运行效率更高，密钥和密文长度更小，因此更适应于实际的应用。

7　结束语

本文针对在计算服务中既要保护用户隐私，又需要对隐私信息进行计算的问题，利用理想格提出了一种支持密文数据算术运算的同态加密方案 CESIL。CESIL 方案具有语义安全性，能在指定的 $[0, N)$范围内支持加乘法的混合同态运算。其生成密钥、加解密及密文运算效率都较高，且密钥和密文长度较小，能够在用户请求计算服务时保护用户数据的隐私安全。

在以后的研究工作中，将从以下 2 个方面改进 CESIL 方案：1) 改进密文的减法和除法运算算法，使方案真正支持四则混合运算；2) 扩展明文空间，使方案支持对负数及浮点数的同态加密。

附录 A　$\boldsymbol{b}\in\mathbb{Z}^n$，$(\boldsymbol{b})$为 $\mathbb{Z}^n$ 中的理想，证明$(\boldsymbol{b})=\mathcal{L}(cycle(\boldsymbol{b}))$

证明　以下 i,j,s 均为[0,n−1]区间的整数

两向量相乘，如

$$(a_0,a_1,\cdots,a_{n-1})^{\mathrm{T}}\otimes(b_0,b_1,\cdots,b_{n-1})^{\mathrm{T}}=(d_0,d_1\cdots,d_{n-1})^{\mathrm{T}}$$

$$d_s=\sum_{i+j\equiv s \bmod n} a_i b_j$$

$$=a_s b_{n-s \bmod n}+a_{s-1 \bmod n} b_{n-(s-1) \bmod n}+\cdots+$$

$$a_{s-(n-1) \bmod n} b_{n-s-1 \bmod n}$$

将其表示为矩阵形式，如下

$$\begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix}\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}=\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \end{bmatrix}$$

则 $cycle((a_0,a_1,\cdots,a_{n-1})^{\mathrm{T}})(b_0,b_1,\cdots,b_{n-1})^{\mathrm{T}}=(d_0,d_1,\cdots,d_{n-1})^{\mathrm{T}}$

若 $\boldsymbol{B}=cycle(\boldsymbol{b})$，显然 $\boldsymbol{B}\in\boldsymbol{Z}^{n\times n}$，则有

$$(\boldsymbol{b})=\{\boldsymbol{b}\otimes\boldsymbol{r}\mid\boldsymbol{r}\in\boldsymbol{Z}^n\}=\{\boldsymbol{B}\boldsymbol{r}\mid\boldsymbol{r}\in\boldsymbol{Z}^n\}=\mathcal{L}(\boldsymbol{B})$$

故，$(\boldsymbol{b})=\mathcal{L}(cycle(\boldsymbol{b}))$

附录B 式(5)的推导

证明 以下 i,j,k,s 均为[0,n−1]区间的整数

$\exists r_1,r_2\in Z^n$，使 $u_1=t_1+Br_1$，$u_2=t_2+Br_2$。

1) $u_1+u_2=t_1+t_2+B(r_1+r_2)$或表示为 $u_1+u_2=t_1+t_2+Br$。

2) $\forall u,r\in Z^n$，$u\otimes Br=u\otimes cycle(b)\otimes r=B\otimes(u\otimes r)$

或表示为 $u\otimes Br=Br'$。

3) $u_1\otimes u_2=t_1\otimes t_2+t_2\otimes Br_1+t_1\otimes Br_2+Br_1\otimes Br_2$。

由2)可知，$\exists r''\in Z^n$，使 $u_1\otimes u_2=t_1\otimes t_2+Br''$。

4) $\exists r''\in Z^n$，使 $u_1^1\circ u_2^2\circ\cdots\circ u_r^{r-1}=t_1^1\circ t_2^2\circ\cdots\circ t_r^{r-1}+Br''$，即

$$Dec_{\varepsilon 2}(B,u_1^1\circ u_2^2\circ\cdots\circ u_r^{r-1})=\left[u_1^1\circ u_2^2\circ\cdots\circ u_r^{r-1}\right]\cdots=\left[t_1^1\circ t_2^2\circ\cdots\circ t_r^{r-1}\right]_B$$

参考文献：

[1] RIVEST R L, ADLEMAN L, DERTOUZOS M L. On data banks and privacy homomorphisms[A]. DeMillo RA Foundations of Secure Computation[C]. NY, USA: Academic Press, 1978.169-180.

[2] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[A]. Proc of the Advances in Cryptology (EUROCRYPT'99)[C]. Prague, Czech Republic, 1999.223-238.

[3] GOLDWASSER S, MICALI S. Probabilistic encryption[J]. Journal of Computer and System Sciences, 1984, 28(2): 270-299.

[4] RIVEST R L, SHAMIR A, ADLEMAN L. A method for obtaining digital signatures and public-key cryptosystems[J]. Communications of the ACM, 1978, 21(2): 120-126.

[5] ELGAMAL T. A public-key cryptosystem and a signature scheme based on discrete logarithms[J]. IEEE Transactions on Information Theory, 1985, 31(4): 469-472.

[6] BONEH D, GOH E J, NISSIM K. Evaluating 2-DNF formulas on ciphertexts[A]. Second Theory of Cryptography Conference (TTC 2005)[C]. Cambridge, MA, USA, 2005.325-341.

[7] GENTRY C. A Fully Homomorphic Encryption Scheme[D]. California, USA: Stanford University, 2009.

[8] GENTRY C. Fully homomorphic encryption using ideal lattices[A]. Proc of the 41st ACM Symposium on Theory of Computing(STOC' 09)[C]. Bethesda, Maryland, USA, 2009.169-178.

[9] SMART P N, VERCAUTEREN F. Fully homomorphic encryption with relatively small key and ciphertext sizes[A]. Proc of the Public Key Cryptography (PKC 2010)[C]. Paris, France, 2010.420-443.

[10] DIJK V M, GENTRY C, HALEVI S, *et al*. Fully homomorphic encryption over the integers[A]. Proc of the Advances in Cryptology (EUROCRYPT 2010)[C]. Riviera, France, 2010.24-43.

[11] CORON J, MANDAL A, NACCACHE D, *et al*. Fully homomorphic encryption over the integers with shorter public keys[A]. Proc of the Advances in Cryptology (CRYPTO 2011)[C]. Santa Barbara, California, USA, 2011.487-504.

[12] BRAKERSKI Z, VAIKUNTANATHAN V. Fully homomorphic encryption from ring-lwe and security for key dependent messages[A]. Proc of the Advances in Cryptology (CRYPTO 2011)[C]. Santa Barbara, California, USA, 2011.505-524.

[13] GENTRY C, HALEVI S. Implementing Gentry's fully-homomorphic encryption scheme[A]. Proc of the Advances in Cryptology (EUROCRYPT 2011)[C]. Tallinn, Estonia, 2011.129-148.

[14] BRAKERSKI Z, GENTRY C, VAIKUNTANATHAN V. Fully homomorphic encryption without bootstrapping[J]. Computer and Information Science, 2011, 111(111): 1-12.

[15] GENTRY C, HALEVI S. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits[A]. Proc of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science[C]. Palm Springs, CA, USA, 2011.107-109.

[16] 黄汝维, 桂小林, 余思等. 云环境中支持隐私保护的可计算加密方法[J]. 计算机学报, 2011, 34(12): 2391-2402.
HUANG R W, GUI X L, YU S, *et al*. Privacy-preserving computable encryption scheme of cloud computing[J]. Chinese Journal of Computers, 2011, 34(12): 2391-2402.

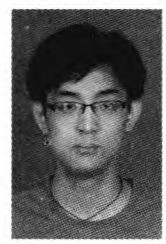
[17] 冯登国. 信息安全中的数学方法与技术[M]. 北京: 清华大学出版社, 2009.
FENG D G. Mathematical Methods and Techniques for Information Security[M]. Beijing: Tsinghua University Press, 2009.

[18] HOFFSTEIN J, PIPHER J, SILVERMAN J H. NTRU: a ring-based public key cryptosystem[A]. Proceedings of the Third International Symposium on Algorithmic Number Theory[C]. Portland, Orgeon, USA, 1998.267-288.

[19] MICCIANCIO D. Improving lattice based cryptosystems using the hermite normal form[A]. Proc of the Cryptography and Lattices Conference[C]. Rhode Island, USA, 2001.126-145.

[20] DAVIS P J. Circulant Matrices[M]. New York: Jonh Wiley & Sons, 1979.85-91.

作者简介：



杨攀（1987-），男，陕西岐山人，西安交通大学博士生，主要研究方向为云计算的用户数据隐私保护及同态加密技术等。

桂小林（1966-），男，江西新余人，博士，西安交通大学教授、博士生导师，主要研究方向为云计算、网络与信息安全、物联网与社会网络理论等。

姚婧（1987-），女，陕西西安人，西安交通大学博士生，主要研究方向为云计算安全及网络信息安全。

林建财（1988-），男，福建泉州人，西安交通大学硕士生，主要研究方向为云计算安全及同态加密技术。

田丰（1987-），男，陕西紫阳人，西安交通大学博士生，主要研究方向为云计算安全、基于位置的服务及位置隐私保护问题。

张学军（1977-），男，宁夏中宁人，西安交通大学博士生，主要研究方向为服务计算及隐私保护。