基于格的密码学技术专题讲座(二)

第4讲 基于格的全同态加密实现方案与相关技术

韩敬利，杨 明，王兆丽
(解放军理工大学指挥信息系统学院，江苏 南京 210007)

摘 要：文章主要介绍了Gentry突破性的工作：基于格的全同态加密方案的构建蓝本，对Gentry的构造思想、构造过程等方面进行了详细的描述。随后介绍了两种全同态加密方案的实现方法，详细介绍了两种方案的构造过程，并对其安全性和有效性进行了分析讨论。

关键词：格；理想格；初级全同态加密方案；自解性质；全同态加密方案
中图分类号：TP309.2 **文献标识码**：A **文章编号**：CN32-1289(2014)02-0087-06

Implemention and Technology of Lattice-based Fully Homomorphic Encryption Scheme

HAN Jing-li, YANG Ming, WANG Zhao-li
(College of Command Information System, PLAUST, Nanjing 210007, China)

Abstract: The breakthrough work of Gentry: the construction framework of fully homomorphic scheme based on ideal lattices was introduced. The Gentry's framework was specifically explained and two improvements to Gentry's framework and the analysis were also described.

Key words: lattice; ideal-lattice; somewhat homomorphic encryption scheme; bootstrappable; fully homomorphic encryption scheme

随着云计算研究的日益发展，隐私保护等安全问题被称为是云计算能否真正普及的关键问题。用户不愿将数据以不加密的形式存放在云端服务器中，特别是一些敏感、隐私数据，但是将数据加密又丧失了云计算的优势。全同态加密方案则可以有效的将隐私和便利结合在一起，使云端服务器能够在不知道密钥、不解密的情况下对用户的加密数据进行正确处理，而处理过程可以保证用户的隐私安全。

全同态加密最开始被称为隐私同态(privacy homomorphism)，是1978年由Ron Rivest、Leonard Adleman、Dertouzous[1]在RSA方案公布之后提出来的。RSA是一个关于乘法同态的加密方案，同样ElGamal密码系统、Goldwasser-Micali密码系统等都仅对某一种运算保持同态。Ron Rivest等于是提出了这样一个公开的问题：依靠一个全同态的加密方案，用户在没有解密密钥的情况下，可以在加密数据上做任何操作。问题提出后的30年中，虽然出现了很多同态加密方案，但是都不能做到对任意操作保持同态。直到2009年，Craig Gentry[2,3]的突破性工作才解决了这个公开问题，Gentry基于格理论提出了构建全同态加密方案的基本蓝本。此后，根据Gentry提出的全同态方案的构建技术，一些研究者相继提出了一些具体的实现方案[4~10]，并提出了一些有效的算法改进技术。

收稿日期：2014-01-20；**修回日期**：2014-03-26
作者简介：韩敬利(1983－)，女，硕士，讲师.

1 全同态加密方案

一般的公钥加密方案包括三个算法：密钥生成算法 KeyGen、加密算法 Encrypt 和解密算法 Decrypt，方案有两个密钥(sk，pk)，其中公钥 pk(public key)为加密密钥、私钥 sk(secrect key)为解密密钥。全同态加密方案在普通公钥加密方案的基础上又加入了密文计算算法 Evaluate。在全同态加密方案中，一般用布尔线路来描述任意运算，所以在方案中密文计算算法只需给出加法操作 Add 和乘法操作 Mult 即可。什么是全同态加密呢？对于密文 ψ_1，…，ψ_t(ψ_i 为明文 m_i 在公钥 pk 下的加密结果：$\psi_i=\text{Encrypt}(m_i)$)和任意的运算函数 f，全同态加密允许任何人在不知道私钥的情况下求得 $f(m_1, \cdots, m_t)$的加密形式。具体来说，对任意的密钥对(sk，pk)，对任意的布尔线路 C 和任意的密文 $\psi_i=\text{Encrypt}(m_i)$，只需公钥即可计算输出 $\psi=\text{Evaluate}(C, \psi_1, \cdots, \psi_t)$，而利用私钥 sk 解密可得 $\text{Decrypt}(\psi)=C(m_1, \cdots, m_t)$。

Gentry 基于格理论提出的全同态加密方案的构建蓝本中，首先基于理想格构建初级同态加密方案，此方案只可正确同态操作简单的布尔线路，称为 SHE(Somewhat Homomorphic Encryption Scheme)；然后通过压缩(squash)技术降低方案解密线路的运算深度，使方案具有自解的(bootstrappable)性质，从而构建出全同态加密方案 FHE(Fully Homomorphic Encryption Scheme)。

在 Gentry 的方案中密文 ψ 具有 $v+e$ 的形式，v 为格中向量，e 为随机误差向量，当误差在固定的区域内则可保证解密正确 $m=\text{Decrypt}(\psi)$。每当对密文进行同态计算时，误差也会随之增大，而当误差超出界限后将会导致解密错误，这就限定了可以对密文同态操作的次数。如果加密方案 $\varepsilon=$(KeyGen, Encrypt, Decrypt, Evaluate)对一个线路集 $\mathbf{C}_\varepsilon$ 中的任意布尔线路 $C\in\mathbf{C}_\varepsilon$ 都能正确同态解密，则称方案 ε 为同态加密方案。若线路集 $\mathbf{C}_\varepsilon$ 包含所有的线路，方案 ε 即为全同态加密方案。

Gentry 全同态加密方案的核心是密文的重加密操作。重加密操作允许“refresh”密文，对于密文 $\psi=\text{Encrypt}(m)$，计算一个对应于 m 的新密文 ψ'，ψ'的误差比 ψ 的要小。当计算布尔线路 C 时，在计算完 C 的每一个门后都“refresh”结果密文，那么方案就可以保证即使 C 计算次数很多也能解密正确，方案即为全同态加密方案。重加密操作的实现是在每一步运算后同态执行方案的解密线路，所以方案解密线路的复杂性是一个可实现的、有效的全同态加密方案的基础。如果方案 ε 的解密线路 $D_\varepsilon\in\mathbf{C}_\varepsilon$，那么称方案 ε 是自解的“bootstrappable”，在此基础上即可构建全同态加密方案。

Gentry 通过在公钥中加入私钥的提示来降低方案解密线路的复杂性，但带来的问题是导致密钥长度过大。虽然在 Gentry 的构建蓝本提出之后，研究者相继提出了多个全同态加密方案的实现方案，但是这些方案或者不能真正实现全同态或者密钥长度过大，都不是实际有效的方案。

2 Smart 全同态加密方案

Smart 等提出的密钥、密文长度小的全同态加密方案是对 Gentry 的构建蓝本的第一个尝试操作，其构造过程主要是基于素理想理论，从而可借助理想格理论构建方案。Smart 方案的最主要优点就是密钥比较小，其中公钥只包含两个用以描述素理想的整数。但是 Smart 方案的密钥生成算法太复杂，需要在很多候选中挑选出一个满足要求的公钥，对应于 n 维格一般需要生成 $n^{1.5}$ 个候选。选择公钥以后，计算私钥的复杂度为 $\tilde{\Theta}(n^{2.5})$，所以当格的维度大于 2048 时方案就没法生成对应的密钥，这是 Smart 方案的主要缺点。Smart 方案估计压缩后的解密多项式次数大概为几百，如果要能够同态操作此解密线路的话，格的维度至少为 $n=2^{27}$，这已经远远超出了 Samrt 方案的能力范畴。所以 Smart 方案为我们提供了一种实现全同态加密方案的新思路，但并不是实际有效的，还需进一步优化。

2.1 初级同态加密方案 SHE

对于一个次数为 N 的、首一不可约多项式 $F(x)\in\mathbf{Z}[x]$，复数集合中的元素 θ 是 $F(x)$的一个根，在数域

$\mathbf{K}=\mathbf{Q}(\theta)=\{a_0+a_1\theta+\cdots+a_{n-1}\theta^{n-1}\mid a_i\in\mathbf{Q}\}$上定义整环 $\mathbf{O}_\mathbf{K}=\mathbf{Z}[\theta]\subset\mathbf{K}$，则 $\mathbf{O}_\mathbf{K}$ 是一个 Dedakind 整环。对于一个素数 p，Dedekind 整环 $\mathbf{O}_\mathbf{K}$ 中理想 $p\mathbf{O}_\mathbf{K}$ 有唯一分解 $p\mathbf{Z}[\theta]=\boldsymbol{\rho}_1\cdots\boldsymbol{\rho}_t$，其中 $\boldsymbol{\rho}_i=<p^{\deg(f_i)},f_i(\theta)>$是一个范数为 $p^{\deg(f_i)}$的素理想(剩余次数为 $\deg(f_i)$)。当剩余次数为 1 时，理想可表示成 $\boldsymbol{\rho}=<p,\theta-\alpha>$($p$ 为理想的范数，α 是 $F(x)\bmod q$ 的根)，就很容易构造理想对应的 Hermite 正则形式 HNF：

$$\mathbf{H}=\begin{bmatrix} p & & & 0\\ -\alpha & 1 & & \\ -\alpha^2 & & 1 & \\ \cdots & & & \cdots \\ -\alpha^{N-1} & 0 & & 1 \end{bmatrix}。$$

$\boldsymbol{\rho}$ 是素理想，则 $\boldsymbol{\rho}$ 可由一个元素生成 $\boldsymbol{\rho}=<\gamma>=\gamma\cdot\mathbf{Z}[\theta]$，而由 HNF 或者理想的两个参数表示 $\boldsymbol{\rho}=<p,\theta-\alpha>$，找到理想的生成元是个难解问题。

下面介绍基于代数的理想理论构造 SHE(KeyGen, Encrypt, Decrypt, Add, Mult)的方法，此方案为逐位加密方案，明文空间为 $P=\{0,1\}$。

KeyGen(η)。η 为系统安全参数，算法生成公钥 pk 为(p, α)，私钥 sk 为 B。

公钥中 p 的生成过程：首先选择一个 N 次的首一不可约多项式 $F(x)\in\mathbf{Z}[x]$，然后重复以下过程直到 p 为素数：在 $\mathbf{T}_{\infty,N}(\eta/2)$ ($\mathbf{T}_{\infty,N}(r)=\{\sum_{i=0}^{N-1}a_ix^i:-r\leqslant a_i\leqslant r,\ a_i\in\mathbf{Z}\}$)内随机选择多项式 $S(x)$，令 $G(x)=1+2S(x)$，计算多项式 $G(x)$和 $F(x)$的结式 $p=\text{resultant}(G(x),F(x))$。

公钥中 α 的生成过程：在域 $F_p[x]$上计算 $D(x)=\gcd(G(x),F(x))$，$D(x)$在 F_p 中的根记为 α。

私钥中 B 的生成过程：在 $\mathbf{Q}[x]$上应用 XGCD 算法得到满足 $Z(x)\cdot G(x)=p \bmod F(x)$的多项式$Z(x)=\sum_{i=0}^{N-1}z_ix^i\in\mathbf{Z}[x]$，令 $B=z_0(\bmod 2p)$。

Encrypt(m)。在 $\mathbf{T}_{\infty,N}(\eta/2)$内随机选择多项式 $R(x)$，令 $C(x)=m+2R(x)$，利用公钥 pk 加密可得到密文 $\psi=C(\alpha)\bmod p$。

Decrypt(ψ)。利用私钥 sk 解密：$m=(\psi-[\psi B]/p)\bmod 2$。

Add(ψ_1, ψ_2)。只需公钥 pk 即可进行密文加法计算：$\psi_3=(\psi_1+\psi_2)\bmod p$。

Mult(ψ_1, ψ_2)只需公钥 pk 即可进行密文乘法计算：$\psi_3=(\psi_1\cdot\psi_2)\bmod p$。

KeyGen 算法实际上生成了元素 $\gamma=G(\theta)$，对应的素理想为 $\boldsymbol{\rho}=\gamma\cdot\mathbf{Z}[\theta]=p\cdot\mathbf{Z}[\theta]+(\theta-\alpha)\cdot\mathbf{Z}[\theta]$。在上述方案中，根据公钥破解私钥是 SPIP(Small Principal Ideal Problem)难解问题，由密文破解明文问题可规约到理想格上的 BDDP(Bounded Distance Decoding Problem)，是难解问题，所以在恢复密钥、加密算法的单向性和语义安全方面方案均可从理论上证明其安全性。

2.2　全同态加密方案 FHE

同态加密方案之所以不能对任意深度的线路进行同态计算，是因为每一层操作都会加大密文的噪声，超出一定程度就会造成解密错误。为了得到全同态加密方案，需要在线路的每一层计算结束后对密文进行降噪声处理，使其不超出能正确解密的范围。为了达到这一效果，需要在 SHE 方案的公钥中加入私钥的提示，同时构造一个新的算法 Recrypt，算法输入密文 ψ，输出一个噪声较小的新密文 ψ_{new}，且 Decrypt(ψ)=Decrypt(ψ_{new})。下面介绍 Smart 构建全同态加密方案的尝试方法。

KeyGen。随机选择 s_1 个整数 $B_i\in[-p,\cdots,p]$，存在一个子集 $\mathbf{S}$(基数为 s_2)，满足 $\sum_{j\in\mathbf{S}}B_j=B$。当 $i\in\mathbf{S}$ 的时候 $k_i=1$，其余为 0，对每一个 k_i 利用 SHE 加密 $\psi_i\leftarrow\text{Encrypt}(k_i)$。公钥 pk 为($p$, α, s_1, s_2, $\{\psi_i,B_i\}_{i=1}^{s_1}$)。

Recrypt(ψ)。利用公钥 pk 对密文 ψ 进行重加密。令 $r_i=(B_i\cdot\psi)/p$，r_i 精确到 $\log s_2+2$ 位，每一位记作 $r_{i,j}(j=1,\cdots,\log s_2+2)$。逐位加密 $e_{i,j}=\text{Encrypt}(r_{i,j})$，令 $t_{i,j}=\text{Mult}(e_{i,j},\psi_i)$，对所有 r_i 进行同态加法。

FHE 方案中由公钥推算出私钥是 SSSP(Sparse Subset-Sum Problem)难解问题，加密算法同 SHE 方案

是单向函数。若修改方案中的 KeyGen 算法，令 $B=(z_0 \pmod{2p}, z_1 \pmod{2p}, \cdots, z_{N-1} \pmod{2p})$，对加解密算法同时做相应修改，方案即可扩充到对多位明文同时加密。

取 $F(x)=x^{2^n}+1$，$N=2^n$，$\delta_\infty=N$，$\eta=2^{\sqrt{N}}$，$\mu=\sqrt{N}$ 或者 $\mu=2$，在 x86-64 平台、2.4 GHz 英特尔 Core2 处理器、使用 GCC4.3.2 C 编译器的机器上对方案进行测试分析可得到表 1 中的测试结果。

Smart 方案中 KeyGen 算法太复杂，且当 $N=2^{12}$ 时方案就没法生成密钥了，即使 N 的取值小于 2^{12}，KeyGen 算法也要运行数个小时。虽然在实际运行中，方案的解密线路深度小于理论值，但是仍大于方案所能同态计算的线路深度，所以 Smart 方案不具有自解性质，不足以构建全同态加密方案。

表 1 Smart 方案分析测试结果（b 表示方案实际能够同态计算的乘法深度）

n	Encrypt	Decrypt	Mult	b	
				$\mu=2$	$\mu=\sqrt{N}$
8	4.2	0.2	0.2	1.0	0.0
9	38.8	0.3	0.3	1.5	1.0
10	386.4	0.6	0.6	2.0	1.0
11	3717.2	3.0	3.0	2.5	1.5

3 Gentry 全同态加密实现方案

Gentry 的全同态加密方案是对 Smart 方案进行了一系列的改进，最突出的一个改进就是对密钥生成算法的改进，对于 n 维格将复杂度从 $O(n^{2.5})$ 降至 $O(n^{1.5})$（在实际操作中从耗时几小时、几天降至几秒钟）。

Gentry 的初级同态加密方案 SHE 是一个基于理想格的 GGH 型的加密系统，但是 SHE 暂时没有自解性质，通过在公钥中加入私钥的提示来降低方案解密线路的深度，从而使方案具有自解性质，然后得到全同态加密方案 FHE。公钥称为坏基，是格基的 HNF 形式，私钥称为好基，是一组短基且接近正交。

3.1 初级同态加密方案 SHE

KeyGen。令 $f_n(x)=x^n+1$，其中 n 为 2 的幂。首先任意选取 n 维的整数向量 $\boldsymbol{v}=(v_0, v_1, \cdots, v_{n-1})$，其中 v_i 为 t 位整数，由向量 v 可得到对应多项式 $v(x)=\sum_{i=0}^{n-1} v_i x^i$，进而得到理想格的旋转基 $\boldsymbol{V}$，$\boldsymbol{V}$ 的每一行是对应于多项式 $v_i(x)=v(x)\cdot x^i/f_n(x)$ 的参数向量。

$$\boldsymbol{V}=\begin{bmatrix} v_0 & v_1 & v_2 & & v_{n-1} \\ -v_{n-1} & v_0 & v_1 & & v_{n-2} \\ -v_{n-2} & -v_{n-1} & v_0 & & v_{n-3} \\ & & & \cdots & \\ -v_1 & -v_2 & -v_3 & & v_0 \end{bmatrix}$$

计算整系数多项式 $w(x)$（次数最高为 $n-1$），满足 $w(x)\times v(x)=d \pmod{f_n(x)}$（常数 d 为格 $\mathbf{L}(\boldsymbol{V})$ 的行列式），可用快速傅里叶算法计算 $w(x)$，提高算法的有效性。

最后检测看基 $\boldsymbol{V}$ 的 Hermite 正则形式 HNF($\boldsymbol{V}$) 是否具有规定的格式：

$$\boldsymbol{H}=\mathrm{HNF}(\boldsymbol{V})=\begin{bmatrix} d & 0 & 0 & 0 & 0 \\ -r & 1 & 0 & 0 & 0 \\ -[r^2]_d & 0 & 1 & 0 & 0 \\ -[r^3]_d & 0 & 0 & 1 & 0 \\ & & & \cdots & \\ -[r^{n-1}]_d & 0 & 0 & 0 & 1 \end{bmatrix}$$

Gentry 给出了验证旋转基 $\boldsymbol{V}$ 是否具有规定格式的简单方法和正确性证明。如果矩阵 $\boldsymbol{V}$ 符合要求，那么格 $\mathbf{L}(\boldsymbol{V})$ 中存在一个向量 $(-r, 1, 0, \cdots, 0)$，所以令 $r=w_0/w_1 \bmod d$，只需验证 $r^n=-1 \bmod d$ 即可。如果 $\boldsymbol{V}$ 的 HNF 不具有规定格式，则从头开始执行密钥生成算法，直到满足为止。

公钥 pk（坏基）为 $\boldsymbol{V}$ 的 Hermite 正则形式，只需两个整数 d，r 即可表示，私钥 sk（好基）为向量 $\boldsymbol{v}$ 和 $\boldsymbol{w}$（实

际上只需要向量 w 的一个奇数参量即可)。所以 Gentry 的实现方案中密钥生成算法很简单,而且生成的密钥小。

Encrypt(m)。对明文 $m\in\{0,1\}$,首先选择一个随机的 0-1 误差向量 $\boldsymbol{u}=(u_0,u_1,\cdots,u_{n-1})$(向量 $\boldsymbol{u}$ 中,参数选择 0 的概率为 q,选择±1 的概率均为$(1-q)/2$),令 $\boldsymbol{a}=2\boldsymbol{u}+m\cdot\boldsymbol{e}_1=(2u_0+m,2u_1,\cdots,2u_{n-1})$,密文为 $\psi=\boldsymbol{a} \bmod \boldsymbol{H}=\boldsymbol{a}-(\lceil \boldsymbol{a}\times\boldsymbol{H}^{-1}\rfloor\times\boldsymbol{H})=[\boldsymbol{a}\times\boldsymbol{H}^{-1}]\times\boldsymbol{H}$。$\lceil \boldsymbol{a}\times\boldsymbol{H}^{-1}\rfloor$ 表示 $\boldsymbol{a}\times\boldsymbol{H}^{-1}$的整数部分,$\lceil \boldsymbol{a}\times\boldsymbol{H}^{-1}\rfloor\times\boldsymbol{H}$ 为格中距 $\boldsymbol{a}$ 最近的格点,$[\boldsymbol{a}\times\boldsymbol{H}^{-1}]$表示 $\boldsymbol{a}\times\boldsymbol{H}^{-1}$的小数部分,密文 ψ 为 $\boldsymbol{a}$ 与最近格点的距离向量。密文 $\boldsymbol{\psi}=(\psi,0,\cdots,0)$可以直接由一个整数 $\psi=[\boldsymbol{a}(r)]_d=\boldsymbol{a}(r) \bmod d=[m+2\sum_{i=0}^{n-1}u_ir^i]_d$ 表示。

选择误差向量时取 0 的概率 q 尽可能的大(q 必须小于 1,保证从密文 ψ 中恢复出原始的误差向量是困难的),使得误差向量越稀疏越好。在加密的过程中,Gentry 通过降阶手段 $\sum_{i=0}^{n-1}u_ir^i=r^{n/2}(\sum_{i=0}^{n/2-1}u_{i+n/2}r^i)+\sum_{i=0}^{n/2-1}u_ir^i$,采用递归方法加速密文的计算,但是此种方法需要足够的空间存储所有的分块多项式,所以在 Gentry 的方案在实际测试中格的最高维度为 2^{15},所以存储空间、计算时间的取舍和哪些参数选择更有效还要在更多的实验中去证明。

对于高维度的格来说,利用格基规约算法寻找最近格点问题是难解问题,Gentry 的方案中误差向量的非零分量的个数设置在 15 到 20 个之间就可以有效的避免对误差向量的生日攻击。

Decrypt(ψ)。令 $\boldsymbol{a}=\boldsymbol{\psi} \bmod \boldsymbol{V}=[\boldsymbol{c}\times\boldsymbol{W}/d]\times\boldsymbol{V}$,计算 $m=\boldsymbol{a} \bmod 2$ 求得明文 m。

$\boldsymbol{\psi}$ 是 $\boldsymbol{a}$ 与最近格点的距离向量,所以存在一个格点 $\boldsymbol{y}$ 使得 $\boldsymbol{\psi}=\boldsymbol{a}-\boldsymbol{y}\times\boldsymbol{V}$,由密钥生成算法知 $\boldsymbol{y}\times\boldsymbol{V}\times\boldsymbol{W}/d=\boldsymbol{y}\times\boldsymbol{I}$ 是整数向量,则$[\boldsymbol{\psi}\times\boldsymbol{W}/d]\times\boldsymbol{V}=[\boldsymbol{a}\times\boldsymbol{W}/d]\times\boldsymbol{V}$。要满足$[\boldsymbol{a}\times\boldsymbol{W}/d]\times\boldsymbol{V}=\boldsymbol{a}=(\boldsymbol{a}\times\boldsymbol{W}/d)\times\boldsymbol{V}$ 则要求 $\boldsymbol{a}\times\boldsymbol{W}/d$ 每一个元素的绝对值都要小于 1/2,所以$[\boldsymbol{\psi}\times\boldsymbol{W}]_d=[\boldsymbol{a}\times\boldsymbol{W}]_d=\boldsymbol{a}\times\boldsymbol{W}$。由加密算法可知 $\boldsymbol{\psi}=(\psi,0,\cdots,0)$,通过推导可得$([\psi w_0]_d,[\psi w_1]_d,\cdots,[\psi w_{n-1}]_d)=m\cdot(w_0,w_1,\cdots,w_{n-1})(\bmod 2)$,即对任意的 i,$[cw_i]_d=m\cdot w_i(\bmod 2)$成立。因此在解密过程中,只需取其中一个为奇数的 w_i 即可利用 $m=[\psi w_i]_d(\bmod 2)$解密出明文 m 来。由此可知方案的私钥实际上只需要向量 $\boldsymbol{w}$ 的一个奇数参量即可。

Add(ψ_1,ψ_2)。只需公钥 pk 即可进行密文加法计算:$\psi_3=(\psi_1+\psi_2) \bmod \boldsymbol{H}$。

Mult(ψ_1,ψ_2)。只需公钥 pk 即可进行密文乘法计算:$\psi_3=(\psi_1\cdot\psi_2) \bmod \boldsymbol{H}$。

Genty 对构建的初级同态加密方案做了一系列的测试,测试结果显示方案解密正确的线路的最高次数与 b 成线性关系,要处理次数为 r 的 n 元初等多项式,满足 $2^b>c^r\times\sqrt{\binom{n}{r}}$ 则可正确解密,其中 $c\approx 9$ 为一个常数。

3.2　全同态加密方案 FHE

Gentry 通过压缩方法降低解密线路的深度,使方案具有自解性质,则可进一步实现全同态加密。压缩方案的具体做法是在公钥当中加入私钥的提示,但是根据提示不足以获得私钥,也不可能进行解密。

在公钥中加入一个集合 $\mathbf{S}=\{x_i\in\mathbf{Z}_d:i=1,2,\cdots,s\}$,集合中存在一个非常稀疏的子集 $\mathbf{S}'$,$\mathbf{S}'$所有元素和为 $w \bmod d$,且$|\mathbf{S}'|=s'$。稀疏子集的属性向量是一个位向量 $\boldsymbol{\sigma}=<\sigma_1,\cdots,\sigma_s>$,满足 $\sum_{i=1}^{s}\sigma_ix_i=w \bmod d$。对于密文 ψ,令 $y_i=[\psi x_i]_d$,则 $\mathrm{Decrypt}(\psi)=[\psi w_i]_d(\bmod 2)=[\sum_{i=1}^{s}\sigma_iy_i]_d(\bmod 2)$,通过进一步的优化,将向量 $\boldsymbol{\sigma}$ 拆解成多个向量 $\boldsymbol{\sigma}_1,\cdots,\boldsymbol{\sigma}_s$,每个向量最多只有一个参数为 1,其余为 0,满足 $\sum_{i=1}^{s}\boldsymbol{\sigma}_i=\boldsymbol{\sigma}$。同时定义 s 个大集合 $\boldsymbol{S}_1,\cdots,\boldsymbol{S}_s$($\boldsymbol{S}_k=\{x_{k,i}\mid i=1,\cdots,s\}$)使得 $\sum_{k=1}^{s}\boldsymbol{\sigma}_k\cdot\boldsymbol{S}_k=\sum_{k=1}^{s}\sum_{i=1}^{s}\sigma_{k,i}\cdot x_{k,i}=w$,令 $y_{k,i}=[c\cdot x_{k,i}]_d$。令 $p=\lceil\log_2(s+1)\rfloor$,对每一个 k 和 i,定义 $z_{k,i}$ 为所有 $0,\frac{1}{2^p},\frac{2}{2^p},\cdots\cdots,\frac{2^p}{2^p}$中与 $y_{k,i}/d$ 最接近的那个,则最终可以将解密算法简化为 $\mathrm{Decrypt}(\psi)=\oplus_{k,i}\sigma_{k,i}[y_{k,i}]_2\oplus\left[\lceil\sum_{k=1}^{s}\sum_{i=1}^{s}\sigma_{k,i}z_{k,i}\rfloor\right]_2$。解密算法

的前半部分$\oplus_{k,i}\sigma_{k,i}[y_{k,i}]_2$是线性运算，后面部分$\sum_{k=1}^{s}\sum_{i=1}^{s}\sigma_{k,i}z_{k,i}$就是 XOR 运算和 s 个数的加法运算，至此解密算法完全可由初级同态加密方案同态计算，则方案具有自解性质，由此可得到全同态加密方案。

Gentry 对上述初级和全同态加密实现方案也做了一系列的实验测试，测试在 IBM 系统、x3500 服务器、64 位四核英特尔 Xeon E5450 处理器、12 MB 二级缓存、24 GB 随机存储器的机器上进行，分别得到表 2、表 3 中的测试结果。

对格的维度分三个层次进行测试，小维度 2048，中等维度 8192，大维度 32768，相应的公钥的长度从小维度格的 17 MB 增长到大维度格的 2.25 GB，运行一个自解操作的时间从小维度格的 30 s 到大维度格的 30 min，但是从安全性考虑，格的维度又不能太小，所以 Genty 的全同态加密方案在实际应用过程中并不十分有效。

表 2 初级同态加密方案分析测试（$|d|$表示格行列式 d 的位长）

格的维度 n	位参数 t	格的行列式 d	KeyGen	Encrypt	Decrypt
512	380	$\|d\|=195764$	0.32 s	0.19 s	——
2048	380	$\|d\|=785006$	1.2 s	1.8 s	0.02 s
8192	380	$\|d\|=3148249$	10.6 s	19 s	0.13 s
32768	380	$\|d\|=12625500$	3.2 min	3 min	0.66 s

表 3 全同态加密方案分析测试

格的维度 n	位参数 t	大集合 S 的基数 s	稀疏集合 S' 的基数 s'	公钥长度	KeyGen	Recrypt
512	380	512	15	17 MB	2.5 s	6 s
2048	380	512	15	69 MB	41 s	32 s
8192	380	547	15	284 MB	8.4 min	2.8 min
32768	380	2185	15	2.25 GB	2.2 h	31 min

4 结束语

随着 Gentry 基于格的构建蓝本的提出，使得全同态加密方案的实现成为可能。本文主要介绍了两个基于格的全同态加密实现方案的尝试，Smart 等提出的密钥、密文长度小的全同态加密方案作为第一个尝试，因其密钥生成算法过于复杂使得方案并不能达到真正的全同态，文章随后介绍了 Gentry 提出的全同态加密实现方案，此方案是对 Smart 方案进行了一系列的改进，密钥生成算法和解密算法相对简单，但是为了实现自解性质最终造成密钥过长，而且自解操作对于维度较大的格耗时较长，所以方案仍然不是实际有效的。全同态加密方案是云计算是否能够普及的关键一环，构建实际有效的全同态加密方案的重要性不言而喻，还需要研究者的继续努力。

参考文献：

[1] Rivest R, Adleman L, Dertouzos M. On data banks and privacy homomorphisms[C]// In Foundations of Secure Computation. New York: Academic Press, 1978: 169-177.

[2] Gentry C. A fully homomorphic encryption scheme[D]. Stanford: Stanford University, 2009.

[3] Gentry C. Fully homomorphic encryption using ideal lattices[C]// In Proceedings of STOC 2009. Washington D C: ACM Press, 2009: 169-178.

[4] Smart N P, Vercauteren F. Fully homomorphic encryption with relatively small key and ciphertext sizes[C]// In Public Key Cryptography(PKC'10). Paris: Springer, 2010: 420-443.

[5] Gentry C, Halevi S. Implementing gentry's fully-homomorphic encryption scheme [C]// Advances in Cryptology-EUROCRYPT 2011 Lecture Notes in Computer Science. Tallinn: Springer, 2011: 129-148.

[6] Van Dijk M, Gentry C, Halevi S, et al. Fully homomorphic encryption over the integers [C]// Advances in Cryptology-EUROCRYPT 2010 Lecture Notes in Computer Science. French Riviera: Springer, 2010: 24-43.

[7] Stehle D, Steinfeld R. Faster fully homomorphic encryption[C]// ASIACRYPT 2010 Lecture Notes in Computer Science. Singapore: Springer, 2010: 377-394.

[8] Gentry C. Toward basing fully homomorphic encryption on worst-case hardness[C]// CRYPTO 2010. Santa Barbara, CA, USA: Springer, 2010: 116-137.

[9] Gentry C, Halevi S, Vaikuntanathan V. I-hop homomorphic encryption and rerandomizable yao circuits[C]// CRYPTO 2010. Santa Barbara, CA, USA: Springer, 2010: 155-172.

[10] Melchor C A, Gaborit P, Herranz J. Additively homomorphic encryption with d-operand multiplications[C]// CRYPTO 2010. Santa Barbara, CA, USA: Springer, 2010: 138-154.