基于格的密码学技术专题讲座(三)

第 5 讲　基于格的全同态加密的优化技术

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：文章主要介绍了两个区别于传统全同态加密方案构建蓝本的优化方案，介绍了两种优化方案的构造过程和实现方法，对其安全性和有效性进行了分析讨论，并详细介绍了方案中使用的模数改变、密钥改变、维度归约等优化技术。

关键词：全同态加密；误差学习问题；模数改变技术；密钥改变技术；维度归约技术
中图分类号：TP309.2　　**文献标识码**：A　　**文章编号**：CN32-1289(2014)03-0094-06

Optimization Technique 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: Two optimized schemes different from Gentry's construction framework of fully homomorphic scheme were introduced. In the speech, the key switching, module switching and dimension reduction techniques which were used in the two optimized schemes were specifically explained.

Key words: fully homomorphic encryption scheme; learning with error; module switching; key switching; dimension reduction

公开密钥密码的诞生在现代密码学中具有里程碑式的意义，现有的两类有代表性公钥密码体制分别是 RSA 公钥密码方案和 ElGamal 公钥密码方案，分别基于大数因子分解问题和离散对数问题。然而，随着计算机技术发展和计算能力的提高，为确保安全性，现有公钥密码的密码长度一直在增加，密钥长度的增加意味着现有的公钥密码方案需要在更大的有限域或群内进行指数运算操作，使得这些传统公钥密码在加解密效率方面的问题更加突出。为提高公钥密码的效率，目前有两类不同的方法：一是寻找具有更高复杂性的难解问题，以此来降低密钥的长度，有代表性的方案是椭圆曲线公钥方案；另一方法是寻找具有更简单运算的数学难题，以此来降低加解密操作的复杂性，有代表性的是基于格的密码系统，其基本运算为矢量的加法和乘法。和 RSA 相比，椭圆曲线密码尽管有较短的密钥和密文，但在椭圆曲线加法群上所进行的操作还是十分复杂，在加密、解密、签名和验证等方面的效率改善不大，从而基于格构建高效的密码系统的需求更为迫切。

同时，云计算正以颠覆性的技术和突出的赢利模式成为新的技术浪潮，云计算的低成本、高利用率、灵活

* 收稿日期：2014-05-09；修回日期：2014-06-19
作者简介：韩敬利(1983－)，女，硕士，讲师.

性及良好的扩展性等诸多优点使得它迅速为学术界和 IT 界所认同，有关云计算的应用和系统层出不穷，云计算正展现出勃勃生机和光明的前景。然而，对云计算的担忧也依然存在，其中最大的担心是云计算安全吗？值得信任吗？任何公司或机构，无论大小，都不敢置数据安全风险于不顾。事实上，正是这些对云计算可靠、安全和隐私方面的担忧限制和制约了云计算的使用和普及。

云计算的实质是数据中心或计算的外包，数据存放、处理由第三方负责，除了对数据控制的担忧之外，人们还对云计算可能导致的隐私问题忧心忡忡，而隐私问题涉及面广、影响巨大。随着云计算普及，未来云系统会聚集海量的数据，加之云系统具有超强的计算能力，能进行廉价的数据挖掘，隐私问题将变得更为严重和棘手。用户不愿意将数据以不加密的形式存放在云服务器中，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，除非云计算能够在不知道密钥、不解密的情况下来处理数据，这为公钥密码系统提出了更高的挑战。

2009 年，IBM 公司的克雷格. 金特里(Craig Gentry)发表了一篇文章[1,2]，公布了一项关于密码学的全新发现，是密码学上的一项真正的突破。Gentry 提出了基于格的全同态加密方案，方案可以对加密的数据进行处理得到一个输出，将这一输出进行解密，其结果与用同一方法处理未加密的原始数据得到的输出结果是一样的。

这一名为“全同态加密(fully homomorphic encryption)”的技术被冠以“密码学的圣杯”的称号。对数据进行加密给计算带来了难度，但如果能够在不解密的前提下进行计算则进一步提高了数据的安全性。例如，远程计算服务提供商收到客户发来的加密的医疗记录数据库，借助全同态加密技术，提供商可以像以往一样处理数据却不必破解密码。处理结果以加密的方式发回给客户，客户在自己的系统上进行解密读取。这一技术同样可以应用到网络邮件或在线办公软件套装中。全同态加密方案很好的将隐私保护和计算便利有效的结合在一起，利用加密算法，云端服务器不用解密就可以处理敏感数据。

Gentry 的基于格的全同态加密方案简单描述如下：①初始同态加密方案(SWHE)：首先构建初始同态加密方案 SWHE(Somewhat Homomorphic Encryption Scheme)，SWHE 包含密钥生成算法 KeyGen、加密算法 Encrypt、解密算法 Decrypt 和密文计算算法 Evaluate，一般方案中只需给出密文加法计算 Add 和乘法计算 Mult 即可。构建的 SWHE 可以同态计算次数较低的计算线路；②压缩(squash)：压缩 SWHE 的解密线路，通过在方案的公钥中加入私钥的提示信息来降低解密线路的深度，使得 SWHE 可以同态操作自己的解密线路并且有足够的剩余能力来完成其他所需运算；③自解(bootstrap)：在对密文进行运算时，通过在线路每层运算后进行自解步骤，即同态运行解密算法来刷新密文、降低密文的误差来保证最终解密正确，从而得到全同态加密方案。

Gentry 的全同态加密方案提出后，研究者相继提出了多个全同态加密实现方案[3-6]，几乎所有的方案都遵循 Gentry 提出的原始构建蓝本。但是因为构建蓝本中存在的自解步骤，需要将方案的私钥逐位加密后作为提示加入到公钥中，且为了安全需使用维度较高的格，所以难以避免的会造成方案密钥过长、对于维度较大的格耗时较长等缺点，最终导致这些全同态加密方案的实际操作效率较低。

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。本文主要介绍两个区别于传统全同态加密方案构建蓝本的优化方案，从去除方案的自解步骤、降低密文同态运算过程造成的误差增长问题入手来提高方案的实际有效性。

1　无需自解的全同态加密方案

Brakerski，Gentry 和 Vaikuntanathan[7]提出了一种全新的构建全同态加密方案的方法，新方法是基于误差学习问题 LWE(Learning With Error)或者环上的误差学习问题 RLWE(Ring Learning With Error)来构建全同态加密方案，本文简称为 BGV 方案。

BGV 方案使用密钥改变、模数改变优化技术，去除了传统方案的自解步骤，解决了密文同态计算后维度

增长、误差增大导致的解密困难、解密错误问题，有效的优化了全同态加密方案的实现效率。

1.1 LWE 问题和 RLWE 问题

对于安全参数 λ，整数维度 $n=n(\lambda)$，令 $q=q(\lambda)\geqslant 2$ 是个整数，$\chi=\chi(\lambda)$ 为整数上的分布。$\mathrm{LWE}_{n,q,\chi}$ 问题是区分两个分布：第一个分布为在 $\mathbf{Z}_q^{n+1}$ 上均匀采样$(\boldsymbol{a}_i, b_i)$，其中 $\boldsymbol{a}_i\in\mathbf{Z}_q^n$、$b_i\in\mathbf{Z}_q$；第二个分布中，首先在 $\mathbf{Z}_q^n$ 上均匀采样 $\boldsymbol{s}$、$\boldsymbol{a}_i$，在 χ 上采样 e_i，令 $b_i=<\boldsymbol{a}_i, \boldsymbol{s}>+e_i$，最终得到采样$(\boldsymbol{a}_i, b_i)\in\mathbf{Z}_q^{n+1}$。

对于安全参数 λ，$f(x)=x^d+1$(其中 d 是 2 的幂)，$q=q(\lambda)\geqslant 2$ 是个整数，令 $\mathbf{R}=\mathbf{Z}[x]/(f(x))$、$\mathbf{R}_q=\mathbf{R}/q\mathbf{R}$，$\chi=\chi(\lambda)$为 $\mathbf{R}$ 上的分布。$\mathrm{RLWE}_{d,q,\chi}$ 问题是区分如下两个分布：第一个分布为在 $\mathbf{R}_q^2$ 上均匀采样(g_i, h_i)；第二个分布中，首先在 $\mathbf{R}_q$ 上均匀采样 t、g_i，在 χ 上采样 e_i，令 $h_i=g_i t+e_i$，最终得到采样$(g_i, h_i)\in\mathbf{R}_q^2$。

对于某些 q 和高斯误差分布 χ，$\mathrm{LWE}_{n,q,\chi}$ 和 $\mathrm{RLWE}_{d,q,\chi}$ 可以归约到最坏情况格上难题，是难解问题可以用以构建加密方案[8]。

为了简化表示，Gentry 提出了普遍误差学习问题 GLWE(General Learning With Errors)。GLWE 问题是区分如下两个分布：第一个分布为在 $\mathbf{R}_q^{n+1}$ 上均匀采样$(\boldsymbol{a}_i, b_i)$；第二个分布中，首先在 $\mathbf{R}_q^n$ 上均匀采样 $\boldsymbol{s}$、$\boldsymbol{a}_i$，在 χ 上采样 e_i，$b_i=\langle\boldsymbol{a}_i, \boldsymbol{s}\rangle+e_i$，最终得到采样$(\boldsymbol{a}_i, b_i)\in\mathbf{R}_q^{n+1}$。当 $d=1$ 时，GLWE 即为 LWE 问题，当取 $n=1$ 时，GLWE 为 RLWE 问题。

1.2 基本加密方案

首先基于 GLWE 问题构建了一个不含任何同态操作的加密方案 E，方案 E 的具体算法描述如下。

(1) E. Setup(λ, μ, k)：用 $k\in\{0, 1\}$ 来确定 GLWE 为 LWE 或者 RLWE。选择一个 μ 位的模数 q，$d=d(\lambda, \mu, k)$，$n=n(\lambda, \mu, k)$，$n'=\lceil(2n+1)\log q\rceil$，$\chi=\chi(\lambda, \mu, k)$，$\mathbf{R}=\mathbf{Z}[x]/(x^d+1)$，输出参数 q，k，n，n'，χ。

(2) E. SecretKeyGen()：根据 E. Setup 输出的相关参数生成私钥 sk(secret key)。采样 $\boldsymbol{s}'\leftarrow\chi^n$，令 $\boldsymbol{s}=(1, \boldsymbol{s}'[1], \cdots, \boldsymbol{s}'[n])\in\mathbf{R}_q^{n+1}$，私钥 sk 为 $\boldsymbol{s}$。

(3) E. PublicKeyGen()：根据私钥 sk 生成公钥 pk(public key)。生成 $\boldsymbol{A}'\leftarrow\mathbf{R}_q^{n'\times n}$ 和向量 $\boldsymbol{e}\leftarrow\chi^{n'}$，令 $\boldsymbol{b}=\boldsymbol{A}'\boldsymbol{s}'+2\boldsymbol{e}$，令 $\boldsymbol{A}=[\boldsymbol{b}-\boldsymbol{A}']$ 为 n' 行 $n+1$ 列的矩阵，则 $\boldsymbol{A}\cdot\boldsymbol{s}=2\boldsymbol{e}$，$\boldsymbol{A}$ 即为公钥 pk。

(4) E. Enc(m)：根据公钥 pk 对明文 m 进行加密。明文 $m\in\mathbf{R}_2$，令 $\boldsymbol{m}'=(m, 0, \cdots, 0)\in\mathbf{R}_q^{n+1}$，取样 $\boldsymbol{r}\leftarrow\mathbf{R}_2^{n'}$，输出密文 $\boldsymbol{c}=\boldsymbol{m}'+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}\in\mathbf{R}_q^{n+1}$。

(5) E. Dec$(\boldsymbol{c})$：根据私钥 sk 对密文进行解密。输出 $m\leftarrow[[\langle\boldsymbol{c}, \boldsymbol{s}\rangle]_q]_2$。

因为$[[\langle\boldsymbol{c}, \boldsymbol{s}\rangle]_q]_2=[[\boldsymbol{c}^{\mathrm{T}}\cdot\boldsymbol{s}]_q]_2=[[(\boldsymbol{m}'^{\mathrm{T}}+\boldsymbol{r}^{\mathrm{T}}\boldsymbol{A})\cdot\boldsymbol{s}]_q]_2=[[\boldsymbol{m}'+2\boldsymbol{r}^{\mathrm{T}}\boldsymbol{e}]_q]_2=m$，所以可以保证方案解密算法正确。

1.3 密钥改变优化技术

在全同态加密方案中，一般用布尔线路来描述任意计算，所以只需考虑对密文的同态乘法和加法计算即可。对密文进行同态计算特别是进行乘法运算时，会导致结果密文的误差成倍或者平方增长，同时维度也相应增大，执行多次同态操作会出现解密错误。所以实现全同态加密方案的关键在于对密文进行同态计算后，如何有效的将结果密文的误差控制在可正确解密的范围内。在 Gentry 的构建蓝本中，通过自解步骤来降低同态计算过程中结果密文的误差，但是引入自解步骤后导致方案密钥过长、对于维度较大的格耗时较长。针对自解步骤的缺点，BGV 方案提出了无需自解步骤的优化技术——密钥改变优化技术(key switching)。

在基于 LWE 的密码方案中，通过计算私钥 $\boldsymbol{s}$ 和密文 $\boldsymbol{c}$ 的点积来进行解密 $m=[[\langle\boldsymbol{c}, \boldsymbol{s}\rangle]_q]_2$。假设密文 $\boldsymbol{c}_1$、$\boldsymbol{c}_2$ 分别是 m_1、m_2 的加密结果，对应同一个私钥 $\boldsymbol{s}$，令$\langle\boldsymbol{c}_1, \boldsymbol{s}\rangle\cdot\langle\boldsymbol{c}_2, \boldsymbol{s}\rangle=\langle\boldsymbol{c}_1\otimes\boldsymbol{c}_2, \boldsymbol{s}\otimes\boldsymbol{s}\rangle=Q_{\boldsymbol{c}_1,\boldsymbol{c}_2}(\boldsymbol{s})$，其中 $v\otimes w=(v[1]w[1], v[1]w[2], \cdots, v[n]w[n])$，可以解密得到 $m_1\cdot m_2=[[Q_{\boldsymbol{c}_1,\boldsymbol{c}_2}(\boldsymbol{s})]_q]_2$。$\boldsymbol{s}\otimes\boldsymbol{s}$ 较之 $\boldsymbol{s}$ 的维数增长为平方级，方案无法做到在维数增长的同时保持有效性，为了保证最终解密正确需要降低结果密文

的维度，即得到一个对应私钥 s' 的新密文 c'，$m=[[\langle c', s'\rangle]_q]_2$，$c'$、$s'$ 具有更低的维度。

定期的将一个对应解密私钥 s_1 的密文 c_1 转化为对应另一个解密私钥 s_2 的密文 c_2，c_2 和 c_1 对应同一明文，且 s_2、c_2 与 s_1、c_1 相比具有较低的维度，Brakerski 和 Vaikuntanathan[8] 将这一技术称为维度归约技术，BGV 方案即是在维度归约技术的基础上进一步扩展，提出了密钥改变技术。

密钥改变技术主要包括如下两个步骤：

(1) SwitchKeyGen(s_1, s_2)。输入两个向量，$s_1\in\mathbf{R}_q^{n_1}$，$s_2\in\mathbf{R}_q^{n_2}$（其中 $s_2[0]=1$，且 $n_2=n+1$），s_2 是一个有效的基于 GLWE 的加密系统 E 的私钥。

根据私钥 s_2 和参数 $n'=n_1\cdot\lceil\log q\rceil$，生成公钥 $A=$E.PublicKeygen()。

令 $B=A+$PowersofTwo(s_1, q)，其中 $PowersofTwo(s, q)=(s, 2\cdot s, \cdots, 2^{\lfloor\log q\rfloor}\cdot s)\in\mathbf{R}_q^{n'}$。

输出 $\tau_{s_1\to s_2}=B$。

(2) SwitchKey($\tau_{s_1\to s_2}$, c_1)。首先将 c_1 用二进制形式表示 $c_1=\sum_{j\in\{0,1,\cdots,\lfloor\log q\rfloor\}}2^j\cdot u_j(u_j\in\mathbf{R}_2^n)$，定义 BitDecomp($c_1$, q)$=(u_0, u_1, \cdots, u_{\lfloor\log q\rfloor})\in\mathbf{R}_2^{n'}$。输出 $c_2=$BitDecomp(c_1, q)$^{\mathrm{T}}\cdot B\in\mathbf{R}_q^{n_2}$。

因为向量 c 和 s 具有相同的维数，那么$\langle$BitDecomp(c, q), PowersOf2(s, q)$\rangle=\langle c, s\rangle \bmod q$。令 $A\cdot s_2=2e_2$，$c_2=$SwitchKey($\tau_{s_1\to s_2}$, c_1)，可知$\langle c_2, s_2\rangle=2\langle$BitDecomp($c_1$,$q$),$e_2\rangle+\langle c_1, s_1\rangle$。因为 BitDecomp($c_1$, $q)\in\mathbf{R}_2^{n'}$，所以$\langle$BitDecomp(c_1, q), $e_2\rangle$足够小，则 c_2 是 m 的对应私钥 s_2 的有效密文，误差仅仅增大了一个加法因子。

1.4　模数改变优化技术

在基于 LWE 的密码方案中，密文 c 是 $\mathbf{Z}_q$ 中的一个向量，通过计算私钥 s 和密文 c 的点积来进行解密 $m=[[\langle c, s\rangle]_q]_2$，当密文的误差量低于 $q/4$ 时能够保证解密正确。当进行密文同态计算时，密文的误差会增大，特别是乘法操作会使误差平方增长，比如对密文进行 L 次乘法后，误差会从原始的 e 增长到 e^{2^L}，为了解密正确需取一个的模数 $q\approx e^{2^L}$，但是大模数又会反过来影响方案的有效性和安全性。

基于上述问题，BGV 方案提出了在每次乘法操作后进行降低模数操作的优化技术——模数改变优化技术，模数改变技术将一个 $\mathbf{Z}_q$ 中的一个向量 c 变为 $\mathbf{Z}_{q/w}$ 中的向量 c/w（w 为缩放因子），将模数 q 变为 q/w 同时误差量由 e 将为 e/w。假设方案能够正确解密的误差范围为 D，将缩放因子 w 设为 D，那么每次密文乘法后误差会升为 D^2，但是通过模数改变技术将模数 q 降为 q/D 的同时可以将误差降回到 D，对密文进行更多次的乘法运算，模数一直在缩小但是误差总会降回到 D。当进行 L 次乘法后，误差仍然是 D，但是模数为 q/D^L，所以只需将初始模数设为 $q\approx D^{L+1}$ 即可保持方案的安全性。

在此将模数改变算法记为 $c'=$Scale(c, p, q)，算法输入对应模数 q 的密文 c，输出对应模数 p 的新密文 c'。假设 p、q 为两个整数，c 是一个整数向量，c' 是一个接近 $(p/q)\cdot c$ 的整数向量且 $c'=c \bmod 2$，则对任意满足 $\|[\langle c, s\rangle]_q\|<q/2-(q/p)\cdot l_1(s)$ 的 s，有 $[\langle c', s\rangle]_p=[\langle c, s\rangle]_q \bmod 2$ 且 $\|[\langle c', s\rangle]_p\|<(p/q)\cdot\|[\langle c, s\rangle]_q\|+l_1(s)$（其中 $l_1(s)=\sum_i\|s[i]\|$）。如果取 p 比 q 小，且 s 是短向量（在基于格的密码系统中，短向量可以作为私钥），则有 $\|[\langle c', s\rangle]_p\|<\|[\langle c, s\rangle]_q\|$，所以在不知道私钥仅知道私钥的长度限制的情况下，可以通过改变模数来有效降低密文的误差。

1.5　基于 GLWE 问题的无需自解的全同态加密方案

下面给出无需自解的全同态加密方案——BGV 方案的具体描述，其中参数 L 表示全同态加密方案能够同态处理的计算线路的深度。

(1) FHE.Setup(λ, L, k)。j 从 L 到 0，循环运行 E.Setup(λ, $(j+1)\cdot\mu$, k)得到参数 $\mathbf{P}_j=(q_j, d_j, n_j, n_j', \chi)$，其中 $L+1$ 个模数 $q_L, \cdots, q_0$ 逐次缩小。将所有参数中的 d_j、χ_j 均替换成 $d=d_L$、$\chi_j=\chi_L$，使得环的维度和误差分布不依赖于线路深度 L。

(2) FHE.KeyGen( )。根据参数，j 从 L 到 0 进行如下循环：令 $s_j=$E.SecretKeyGen($\mathbf{P}_j$) 和 $A_j=$E.PublicKeyGen($\mathbf{P}_j$, s_j)。

令 $\boldsymbol{s}_j'=\boldsymbol{s}_j\otimes\boldsymbol{s}_j\in\mathbf{R}_{q_j}^{\binom{n_j+1}{2}}$。

令 $\boldsymbol{s}_j''=\mathrm{BitDecomp}(\boldsymbol{s}_j',\ q_j)$。

$\tau_{\boldsymbol{s}_j''\to\boldsymbol{s}_{j-1}}=\mathrm{FHE.SwitchKeyGen}(\boldsymbol{s}_j'',\ \boldsymbol{s}_{j-1})$，当 $j=0$ 时此步省略。

私钥由所有的 $\boldsymbol{s}_j$ 构成，公钥由所有的 $\boldsymbol{A}_j$ 和 $\tau_{\boldsymbol{s}_j''\to\boldsymbol{s}_{j-1}}$ 组成。

(3) $\mathrm{FHE.Enc}(m)$。公钥 pk 为 $\boldsymbol{A}_L$，直接运行 $\mathrm{E.Enc}(m)$。

(4) $\mathrm{FHE.Dec}(\boldsymbol{c})$。如果密文是对应私钥 $\boldsymbol{s}_j$ 的，根据私钥 $\boldsymbol{s}_j$ 运行 $\mathrm{E.Dec}(\boldsymbol{c})$。

(5) $\mathrm{FHE.Add}(\boldsymbol{c}_1,\ \boldsymbol{c}_2)$。密文 $\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$ 对应同一 $\boldsymbol{s}_j$（如果不是，可以用 FHE.Refresh 算法使 $\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$ 对应同一私钥），令 $\boldsymbol{c}_3=\boldsymbol{c}_1+\boldsymbol{c}_2 \bmod q_j$，将 $\boldsymbol{c}_3$ 解释为对应 $\boldsymbol{s}_j'$ 的密文，然后输出 $\boldsymbol{c}_4=\mathrm{FHE.Refresh}(\boldsymbol{c}_3,\ \tau_{\boldsymbol{s}_j''\to\boldsymbol{s}_{j-1}},\ q_j,\ q_{j-1})$。

(6) $\mathrm{FHE.Mult}(\boldsymbol{c}_1,\ \boldsymbol{c}_2)$。密文 $\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$ 对应同一 $\boldsymbol{s}_j$，首先进行乘法运算，令 $\boldsymbol{c}_3=\boldsymbol{c}_1\cdot\boldsymbol{c}_2 \bmod q_j$，将 $\boldsymbol{c}_3$ 解释为对应 $\boldsymbol{s}_j'$ 的密文，然后输出 $\boldsymbol{c}_4=\mathrm{FHE.Refresh}(\boldsymbol{c}_3,\ \tau_{\boldsymbol{s}_j''\to\boldsymbol{s}_{j-1}},\ q_j,\ q_{j-1})$。

(7) $\mathrm{FHE.Refresh}(\boldsymbol{c},\ \tau_{\boldsymbol{s}_j''\to\boldsymbol{s}_{j-1}},\ q_j,\ q_{j-1})$。密文在 $\boldsymbol{s}_j'$ 下，辅助信息 $\tau_{\boldsymbol{s}_j''\to\boldsymbol{s}_{j-1}}$ 用来进行密钥改变算法，具体操作如下。

展开：$\boldsymbol{c}_1=\mathrm{PowersofTwo}(\boldsymbol{c},\ q_j)$，显然 $\langle\boldsymbol{c}_1,\ \boldsymbol{s}_j''\rangle=\langle\boldsymbol{c}_1,\ \boldsymbol{s}_j'\rangle \bmod q_j$。

模数变换：令 $\boldsymbol{c}_2=\mathrm{Scale}(\boldsymbol{c}_1,\ q_j,\ q_{j-1},\ 2)$，密文 $\boldsymbol{c}_2$ 对应私钥 $\boldsymbol{s}_j''$ 和模数 q_{j-1}。

密钥变换：输出 $\boldsymbol{c}_3=\mathrm{FHE.SwitchKey}(\tau_{\boldsymbol{s}_j''\to\boldsymbol{s}_{j-1}},\ \boldsymbol{c}_2,\ q_{j-1})$，密文 $\boldsymbol{c}_3$ 对应私钥 $\boldsymbol{s}_{j-1}$ 和模数 q_{j-1}。

通过引入密钥改变、模数改变优化技术，去除掉传统加密方案中自解步骤，上述 BGV 方案在实际效率上要远远优于传统的方案。以 RLWE 为例，方案对于深度为 L 的计算线路每个门的计算复杂度为 $\widetilde{O}(\lambda\cdot L^3)$，明显优于传统同态加密方案的 $\widetilde{\Omega}(\lambda^3)$。

2 无需模数改变的全同态加密方案

Brakerski[9] 提出了利用张量技术来构建基于 LWE 问题的全同态加密方案的方法，区别于以前的全同态方案中密文误差平方增长，此方案的密文误差在同态计算过程中只线性增长，而且方案在同态运算过程中不需要进行模数改变，同时在安全性方面可规约到最坏情况格上 GapSVP 问题。

2.1 基本加密方案

Brakerski 方案的基本加密方案即为 Regev[8] 公钥加密方案，具体方案描述如下。

(1) $\mathrm{E.SecretKeygen}(n)$。取样 $s\leftarrow\mathbf{Z}_q^n$，其中 q 是整数，χ 为整数上的分布，私钥 sk 即为 s。

(2) $\mathrm{E.PublicKeygen}(\boldsymbol{s})$。令 $n'=(n+1)(\log q+O(1))$，取样 $\boldsymbol{A}'\leftarrow\mathbf{Z}_q^{n'\times n}$，$\boldsymbol{e}\leftarrow\chi^{n'}$，计算 $\boldsymbol{b}=[\boldsymbol{A}'\boldsymbol{s}+\boldsymbol{e}]_q$，令 $\boldsymbol{A}=[\boldsymbol{b}\,\|\,-\boldsymbol{A}']\in\mathbf{Z}_q^{n'\times(n+1)}$，公钥 pk 即为 $\boldsymbol{A}$。

(3) $\mathrm{E.Encrypt}(m)$。$m\in\{0,\ 1\}$，取样 $\boldsymbol{r}\in\{0,\ 1\}^{n'}$，输出密文 $\boldsymbol{c}=[\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}+\left\lfloor\frac{q}{2}\right\rfloor\boldsymbol{m}']_q\in\mathbf{Z}_q^{n+1}$，其中 $\boldsymbol{m}'=(m,\ 0,\ \cdots,\ 0)\in\{0,\ 1\}^{n+1}$。

(4) $\mathrm{E.Decrypt}(\boldsymbol{c})$。$m=\left[\left\lfloor 2\cdot\frac{[\langle\boldsymbol{c},\ (1,\ \boldsymbol{s})\rangle]_q}{q}\right\rfloor\right]_2$，其中 $\lceil x\rfloor$ 输出与 x 最接近的整数。

在上述方案中，$\langle\boldsymbol{c},\ (1,\ \boldsymbol{s})\rangle=\left\lfloor\frac{q}{2}\right\rfloor m+e(\bmod q)$，如果如果 $|e|<q/4$，那么上述方案解密正确。

2.2 保比例不变的同态加密方案

Brakerski 在构建同态加密方案时，同样使用了密钥改变技术(key switching)，此技术前文已做详细介绍，只需令参数 $d=1$ 使得 GLWE 问题为 LWE 问题即可，此处不再累述。方案的同态计算能力仅仅依赖于 q/D，所以又称为保比例不变的同态加密方案 SI-HE(Scale Invariant Homomrphic Encryption scheme)。

令 $q=q(n)$ 是整数，$\chi=\chi(n)$ 为整数上的分布，$L=L(n)$ 是多项式，保比例不变同态加密方案 SI-HE 具体描述如下。

(1) SI-HE. Keygen(L, n)。利用 E. SecretKeygen 算法生成 $L+1$ 个私钥向量 $\boldsymbol{s}_0$, …, $\boldsymbol{s}_L$，然后利用 $\boldsymbol{s}_0$ 和 E. publicKeygen($\boldsymbol{s}_0$)计算一个公钥 $\boldsymbol{A}_0$，对所有的 $i\in[L]$，定义 $\boldsymbol{s}'_{i-1}=\mathrm{BitDecomp}((1,\boldsymbol{s}_{i-1}),q)\otimes\mathrm{BitDecomp}((1,\boldsymbol{s}_{i-1}),q)\in\{0,1\}^{((n+1)\lceil\log q\rceil)^2}$，计算 $\tau_{i-1\to i}=\mathrm{SwitchKeyGen}(\boldsymbol{s}'_{i-},\boldsymbol{s}_i)$，公钥 pk 为 $\boldsymbol{A}_0$，私钥 sk 为 $\boldsymbol{s}_L$，同态计算密钥 evk 为 $\{\tau_{i-1\to i}\}$。

(2) SI-HE. Enc(m)。与方案 E 相同。

(3) SI-HE. Add($\boldsymbol{c}_1,\boldsymbol{c}_2$)。根据同态计算密钥 evk 同态计算 $\boldsymbol{c}_1+\boldsymbol{c}_2$。$\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$ 对应同一私钥 $\boldsymbol{s}_{i-1}$，首先计算 $\boldsymbol{c}'_{\mathrm{add}}=\mathrm{PowersofTwo}(\boldsymbol{c}_1+\boldsymbol{c}_2)\otimes\mathrm{PowersofTwo}((1,0,\cdots,0))$，然后输出 $\boldsymbol{c}_{\mathrm{add}}=\mathrm{SwitchKey}(\tau_{i-1;i},\boldsymbol{c}'_{\mathrm{add}})\in\boldsymbol{Z}_q^{n+1}$。

(4) SI-HE. Mult($\boldsymbol{c}_1$, $\boldsymbol{c}_2$)。根据同态计算密钥 evk 同态计算 $\boldsymbol{c}_1\cdot\boldsymbol{c}_2$。$\boldsymbol{c}_1$ 和 $\boldsymbol{c}_2$ 对应同一私钥 $\boldsymbol{s}_{i-1}$，首先计算 $\boldsymbol{c}'_{\mathrm{mult}}=\left\lfloor\frac{2}{q}(\mathrm{PowersofTwo}(\boldsymbol{c}_1)\otimes\mathrm{PowersofTwo}(\boldsymbol{c}_2))\right\rfloor$，然后输出 $\boldsymbol{c}_{\mathrm{mult}}=\mathrm{SwitchKey}(\tau_{i-1;i},\boldsymbol{c}'_{\mathrm{mult}})\in\boldsymbol{Z}_q^{n+1}$。

(5) SI-HE. Dec($\boldsymbol{c}$)。$\boldsymbol{c}$ 是对应私钥 $\boldsymbol{s}_L$ 的(如果不是，可以利用密钥改变技术)，根据私钥 $\boldsymbol{s}_L$ 进行解密计算 $m=\mathrm{E.\ Decrypt}(\boldsymbol{c})$。

SI-HE 方案中参数 q 可取任意整数，去除了以往必须是 2 的幂数的限制要求，且方案不需要进行模数改变，全程仅需一个模数 q，方案描述相对简单。对于基于 $\mathrm{LWE}_{n,q,\chi}$ 问题的加密方案，如果 $q/D\geqslant(O(n\log q))^{L+O(1)}$，其中 D 为误差分布 χ 的范围($|\chi|\leqslant D$)，则 SI-HE 可同态处理深度为 L 的计算线路，且 SI-HE 同态计算能力仅仅依赖于 q/D。

3　结束语

全同态加密方案以其在隐私保护方面的优势，使其成为左右云计算能否普及的重要一环，如何提高全同态加密方案的实际有效性也成为众多研究者的关注点。Gentry 的开创性的工作使得全同态加密方案成为可能，但是因其构建过程的固有缺点使其难以投入实际应用。本文通过介绍两个不同于 Gentry 构建蓝本的全同态加密方案，详细介绍了密钥改变、模数改变等优化技术，通过在加密方案的构建过程中引入优化技术，提高了全同态方案的有效性，同时提供了构建全同态加密方案的全新视角，使全同态加密方案的实用化更近。

参考文献：

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

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

[3] Smart N P, Vercauteren F. Fully homomorphic encryption with relatively small key and ciphertext sizes[C]// PKC'10. Paris: Springer, 2010: 420-443.

[4] 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.

[5] 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.

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

[7] Brakerski Z, Gentry C, Vaikuntanathan V. Fully homomorphic encryption without bootstrapping[C]// The 3rd Innovations in Theoretical Computer Science Conference. Massachusetts: ACM Press, 2012: 211-230.

[8] Regev O. On lattices, learning with errors, random linear codes, and cryptography[C]// The 37th ACM Symposium on Theory of Computing. Baltimore: ACM Press, 2005: 84-93.

[9] Brakerski Z. Fully homomorphic encryption without modulus switching from classical GapSVP[C]// The 32nd International Cryptology Conference. California: Springer, 2012: 868-886.