网络首发时间：2024-05-17 18:49:28
网络首发地址：https://link.cnki.net/urlid/11.4494.TN.20240515.1450.006

全同态加密技术的研究现状及发展路线综述

戴怡然[①②③] 张 江[②] 向斌武[①②③] 邓 燚[*①③]

[①](中国科学院信息工程研究所信息安全国家重点实验室，北京 100093)

[②](密码科学技术全国重点实验室，北京 100878)

[③](中国科学院大学网络空间安全学院，北京 100049)

摘 要：随着物联网、云计算、人工智能的应用与普及，数据安全与隐私保护成为人们关注的焦点。全同态加密，作为隐私安全问题的有效解决办法，允许对加密数据执行任意同态计算，是一种强大的加密工具，具有广泛的潜在应用。该文总结了自2009年以来提出全同态加密方案，并根据方案的核心技术划分成4条技术路线，分析讨论了各类方案的关键构造，算法优化进程和未来发展方向。首先，全面介绍了全同态加密相关的数学原理，涵盖了全同态加密方案的基础假设和安全特性。随后，按照4条全同态加密方案的技术路线，归纳了加密方案的结构通式，总结了自举算法的核心步骤，讨论了最新研究进展，并在此基础上综合分析比较了各类方案的存储效率及运算速度。最后，展示了同态算法库对每条技术路线下加密方案的应用实现情况，分析了在当前时代背景下全同态加密方案的机遇与挑战，并对未来的研究前景做出了展望。

关键词：全同态加密；自举；BGV；GSW；CKKS

中图分类号：TN918.4; TP309 **文献标识码**：A **文章编号**：1009-5896(2024)05-0001-16

DOI: 10.11999/JEIT230703

Overview on the Research Status and Development Route of Fully Homomorphic Encryption Technology

DAI Yiran[①②③] ZHANG Jiang[②] XIANG Binwu[①②③] DENG Yi[①③]

[①](*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093, *China*)

[②](*State Key Laboratory of Cryptology, Beijing* 100878, *China*)

[③](*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract: With the application and popularization of IoT, cloud computing, and artificial intelligence, data security and privacy protection have become the focus of attention. Fully homomorphic encryption, as an effective solution to the privacy security problem, allows performing arbitrary homomorphic computation on encrypted data, and is a powerful encryption tool with a wide range of potential applications. The paper summarizes the proposed fully homomorphic encryption schemes since 2009, and divides them into four technical routes based on the core technologies of the schemes, analyzes and discusses the key constructs, algorithm optimization processes, and future development directions of each type of scheme. The paper firstly introduces fully homomorphic encryption-related mathematical principles, covering the basic assumptions and security features of fully homomorphic encryption schemes. Subsequently, according to the technical routes of the four fully homomorphic encryption schemes, it summarizes the structural general formulas of the encryption schemes, summarizes the core steps of the bootstrap algorithms, discusses the latest research progress, and on the basis of this, comprehensively analyzes and compares the storage efficiencies and computing speeds of various schemes. The paper finally shows the application implementation of homomorphic algorithm library for encryption schemes under each technical route, analyzes the opportunities and challenges of fully homomorphic encryption schemes in the current era, and makes an outlook on the future research prospects.

Key words: Fully Homomorphic Encryption; Bootstrapping; BGV; GSW; CKKS

收稿日期：2023-09-28；改回日期：2024-01-22；

*通信作者： 邓燚 deng@iie.ac.cn

基金项目：国家重点研发计划(2023YFB4503203)，国家自然科学基金(62372447, 61932019)

Foundation Items: The National Key Research and Development Project of China (2023YFB4503203), The National Natural Science Foundation of China (62372447, 61932019)

1 引言

传统的密码体制往往涉及加密信息的交换，需要在交换的节点中分享密钥，这种操作容易引发隐私泄露的问题，特别是需要将敏感数据交给第三方的场景，例如云计算中服务器、提供商和云运营商都可以接触到用户的敏感信息。同态加密，作为密码学里一种特殊的加密方式，可以解决这个问题[1,2]：在不提供解密密钥的情况下，同态密文可以发给任意的第三方进行计算，第三方返回密文形式的计算结果，最后由用户自行下载解密。全同态加密的概念最早于上世纪70年代被提出，1978年Rivest等人[3]第一次提到了通过对密文进行计算，获得对应明文被执行相同运算操作的系统构想，后人把这一想法重新总结命名为全同态加密。

根据密文数据上允许进行运算的种类和次数，同态加密方案可以分为3大类：单同态加密、类同态加密和全同态加密。单同态加密(Partially Homomorphic Encryption, PHE)的特点是仅能支持一种同态运算，但是可以执行无限次该同态运算。例如用ElGamal函数[4]：$g^{m_0}\cdot g^{m_1}=g^{m_0+m_1}$可以构造支持加法同态的单同态加密方案，RSA算法[5]：$m_0^e\cdot m_1^e=(m_0\cdot m_1)^e$则可以用来构造支持乘法同态的单同态加密方案。它们的局限性在于只能进行单一的加法或乘法运算，无法支持同时需要多种同态运算的计算场景。类同态加密(Somewhat Homomorphic Encryption, SWHE)的特点是可以支持多种同态运算，但是允许的运算次数有限。例如基于配对的循环群加密算法[6]可以同时支持加法和乘法的同态运算，但是乘法只能进行至多1次。此外，还有一种层次型同态加密(Leveled Fully Homomorphic Encryption)，它可以在给定深度的电路上完成有限次的函数同态运算，执行同态运算的次数与电路深度成正相关，本质上仍属于类同态加密。相比于单同态仅支持单一运算，类同态已经初步实现支持混合电路同态运算，但对于一个任意逻辑和深度的函数仍无法实现完整的同态计算。全同态加密(Fully Homomorphic Encryption, FHE)，顾名思义，可以支持无限次、所有种类的电路同态运算。全同态加密方案支持对密文进行任意次数同态计算的特点，可以借此为任何想要的功能构建程序，这些程序可以直接在密文输入上运行，产生密文为相应明文运算结果的加密。全同态加密在私有计算外包方面具有很大的实际意义，例如，在云计算[7,8]、机器学习[9]、医疗[10]等领域，由于不需要数据的解密就可直接运算，它可以由不受信任的一方运行且不会暴露其输入和内部状态，用户的数据安全得到了有效保障。

2009年Gentry[11]提出了第1个基于理想格(ideal lattice)构造的全同态方案，它可以支持任意电路的求值。Gentry在他的论文中不仅提出了第1个全同态方案，而且设计了一种由类同态方案构造全同态方案的方法——自举。自举是目前全同态加密方案构造最通用途径，它可以理解为加密方案同态的运行自己的解密算法，将一个噪音接近临界值的密文“刷新”成一个噪声很低的新密文，进而得以继续新一轮的运算。因为对称同态加密方案对云计算的适用性有限，且存在通用转换可以将按比特加密的对称全同态转换成公钥全同态加密，所以全同态加密方案的研究工作主要基于公钥加密体制。在Gentry思想的影响下，涌现了许多新结构的同态加密方案，除了基于格上困难学习假设(Learning With Errors, LWE)的方案之外，还有基于近似最大公约数假设(Approximate Greatest Common Divisor, AGCD)的方案例如Pereira[12]，基于NTRU(Nth-degree Truncated polynomial Ring Unit)算法的有FINAL[13]、Kamil[14]和国内向斌武等人[15]的最新研究成果，其中基于AGCD和NTRU的方案目前均从基于LWE的全同态加密方案“移植”得到，因此本文不对基于这两个困难问题的全同态加密方案做重点讨论。

本文提供了全同态加密自2009年Gentry取得突破性进展以来的一个全面的视角，探究不同技术路线下全同态加密方案的发展脉络和相关应用。本文的第2节补充了格的数学定义，以及基于格的密码学中涉及的相关数学问题，帮助理解后续给出的全同态方案的概念和构造；第3节，根据全同态方案的关键技术特点，分成4个部分详细讲述了每类代表性全同态方案的优化进程和自举算法，并给出了具体实现的相关数据；第4节，根据前文提到的代表性全同态方案，介绍了一些基于这些加密方案设计的全同态算法库及其支持的优化算法；第5节讨论了全同态领域发展的现状和目前存在的一些问题和挑战，总结回顾了本文内容，对未来全同态加密算法的改进优化方向做出了展望。

2 背景知识

本节的第1部分介绍格的数学定义，并描述基于格的同态加密方案安全性所依赖的数学困难问题。第2部分则介绍全同态方案中的基本算法模型和同态加密相关的安全性定义。

2.1 格定义及格上困难问题

一个k维格是R^n的离散加性子群。令

$B=(\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_k)$为$R^n$中线性无关的向量集合，则$B$中元素的所有整数线性组合的集合可以定义为$B$生成的格$\mathcal{L}(B)$

$$\mathcal{L}=\mathcal{L}(B)=\left\{\Sigma_{i=1}^{k}\gamma_i\boldsymbol{b}_i:\gamma_i\in\mathbb{Z},\boldsymbol{b}_i\in B\right\} \quad (1)$$

其中B称为格$\mathcal{L}(B)$的基。

大多数已知格上的定义都是基于距离的概念。具体来说，对于R^n中的任意向量$\boldsymbol{t}$和格$\mathcal{L}$中的任意元素$\boldsymbol{v}$，这两个向量之间的距离定义为$\operatorname{dist}(\boldsymbol{t},\boldsymbol{v})=|\boldsymbol{t}-\boldsymbol{v}|_2$。因此，$\boldsymbol{t}$与$\mathcal{L}$中任意元素之间的最小距离为

$$\operatorname{dist}(\boldsymbol{t},\mathcal{L})=\min|\boldsymbol{t}-\boldsymbol{v}|_2:\boldsymbol{v}\in\mathcal{L} \quad (2)$$

定义格$\mathcal{L}$的最小距离为$\lambda_1(\mathcal{L})$，即$\mathcal{L}$中最短的非零向量的长度，即

$$\lambda_1(\mathcal{L})=\min\{|\boldsymbol{v}|_2:\boldsymbol{v}\in\mathcal{L},\boldsymbol{v}\neq\boldsymbol{0}\} \quad (3)$$

许多基于格的全同态方案的安全性依赖于最短向量问题(Shortest Vector Problem, SVP)及其变体的困难性。SVP表示在给定格中寻找最短的非零向量。如果将给定的格限制为理想格，则可以得到IdealSVP[16]。

定义1 SVP问题[16]

(1) γ–近似最短向量问题(SVP_γ)：确定一个近似最短的向量。给定$\gamma>1$，在格上找到一个非零向量$\boldsymbol{v}\in\mathcal{L}$使得$|\boldsymbol{v}|\leq\gamma\cdot\lambda_1(\mathcal{L})$。

(2) 判定最短向量问题($\mathrm{GapSVP}_{\gamma,r}$)：判定最短向量的边界是否正确。给定$\gamma>1,r>0$，判断是否满足$\lambda_1(\mathcal{L})\leq r$或$\lambda_1(\mathcal{L})\geq\gamma\cdot r$。

注意，如果最短向量问题是可解的，那么判定最短向量问题也是可解的，即$\mathrm{GapSVP}_{\gamma,r}\leq\mathrm{SVP}$。

最近向量问题(Closest Vector Problem, CVP)是最短向量问题的一个推广。给定非零向量$\boldsymbol{t}\in R^n$，在格$\mathcal{L}$中寻找一个最接近$\boldsymbol{t}$的向量$\boldsymbol{v}$，即$\operatorname{dist}(\boldsymbol{t},\boldsymbol{v})=\operatorname{dist}(\boldsymbol{t},\mathcal{L})$。

定义2 CVP问题[17]

(1) γ–近似最短向量问题(CVP_γ)：在格中找到一个与目标向量最接近的向量。给定$\gamma\geq1$，$\boldsymbol{t}\in R^n$，找到向量$\boldsymbol{v}\in\mathcal{L}$使得$\operatorname{dist}(\boldsymbol{t},\boldsymbol{v})\leq\gamma\cdot\operatorname{dist}(\boldsymbol{t},\mathcal{L})$。

(2) 判定最近向量问题($\mathrm{DCVP}_{\gamma,r}$)：给定$\gamma\geq1,r>0,\boldsymbol{t}\in R^n$，判断是否满足$\operatorname{dist}(\boldsymbol{t},\mathcal{L})\leq r$或$\operatorname{dist}(\boldsymbol{t},\mathcal{L})\geq\gamma\cdot r$。

错误学习问题(LWE)是由Regev[18]在2005年引入的，作为Blum等人[19]的“奇偶校验学习”问题的扩展，在密码学的发展中起到了关键作用。

定义3 LWE问题[18]

给定一个向量$\boldsymbol{b}\in\mathbb{Z}_q^m$和矩阵$\boldsymbol{A}\in\mathbb{Z}_q^{m\times n}$，LWE问题即为找到一个向量$\boldsymbol{s}\in\mathbb{Z}_q^n$，使得$\boldsymbol{A}\cdot\boldsymbol{s}+\boldsymbol{e}=\boldsymbol{b}$，其中$\boldsymbol{e}\in\mathbb{Z}_q^m$为误差分布$\chi$的抽样。

2005年，Regev[18]通过量子归约将格中最坏情况下的判定最短向量GapSVP归约为LWE问题，即如果能找到一个算法在多项式时间内解决LWE问题，那么也能在多项式时间内解决GapSVP问题。因此，基于LWE的同态加密方案的安全性与SVP及其变体的格上困难问题密切相关。

定义4 RLWE问题[20]>

给定多个抽样$(a,b=a\cdot s+e)\in R_q\times R_q$，寻找$s\in R_q$，其中$a$为$R_q$中均匀随机的抽样，$e\in R_q$为误差分布$\chi$的抽样。

环上错误学习(Ring Learning With Errors problem, RLWE)问题是LWE问题的环版本。从实际应用角度来看，基于RLWE问题的同态加密方案比基于LWE的方案计算效率更高，密文大小更紧凑。

2.2 全同态加密算法模型及安全性质

全同态加密方案可以解释为这样一种加密方案:给定一些密文，对明文的加法乘法操作都可以通过直接在密文上进行且无需要解密。记全同态加密方案为ε，共有4个算法：密钥生成算法KeyGen、加密算法Enc、解密算法Dec、同态运算算法Eval，其主要工作流程为：

(1)给定安全参数λ，运行密钥生成算法$\mathrm{KeyGen}(1^\lambda)$，生成公钥pk，私钥sk，同态计算公钥evk。

(2)给定明文$m_1,m_2,\cdots,m_k$和公钥pk，对$i\in[1,k]$，运行加密算法$\mathrm{Enc}(\mathrm{pk},m_i)$，生成密文$\mathrm{ct}_i$。

(3)给定函数f和一些密文$\mathrm{ct}_1,\mathrm{ct}_2,\cdots,\mathrm{ct}_k$，同态计算公钥evk，运行同态运算算法$\mathrm{Eval}(f,\mathrm{evk},\mathrm{ct}_1,ct_2,\cdots,\mathrm{ct}_k)$，获得密文$\mathrm{ct}_1,\mathrm{ct}_2,\cdots,\mathrm{ct}_k$在经过函数$f$同态运算的密文结果$\mathrm{ct}_f$。

(4)用私钥sk对密文ct_f运行解密算法$\mathrm{Dec}(\mathrm{sk},\mathrm{ct}_f)$，获得相应的明文$m_f=f(m_1,m_2,\cdots,m_k)$。

其中第3步的同态运算算法就是同态加密方案的核心，对Eval输入密文可以实现任意函数的计算，更进一步还可以计算解密函数，这也是构造全同态加密方案的重要途径。

一个安全且合理的全同态加密方案具有4个安全性质：正确性，语义安全性，同态性，紧凑性。

(1) 正确性：对于任意的消息$m\in\{0,1\}$，经同态加密方案加密的消息都能正确解密，即$m=\mathrm{Dec}(\mathrm{sk},\mathrm{Enc}(\mathrm{pk},m))$。

(2) 语义安全性：一个任意多项式时间敌手A成功区分两个加密不同消息的密文，它的优势是

可忽略的，也可以称这个加密方案是不可区分选择明文安全的(IND-CPA安全)

$$\Pr\left[A\left(\mathrm{pk},\mathrm{Enc}\left(\mathrm{pk},m=0\right)=1\right)\right]-\Pr\left[A\left(\mathrm{pk},\mathrm{Enc}\left(\mathrm{pk},m=1\right)=1\right)\right]=\mathrm{negl}\left(\lambda\right) \tag{4}$$

(3) 同态性：给定一个电路f，任意一组消息$m_1,m_2,\cdots,m_k$，有

$$\mathrm{Dec}\left(\mathrm{sk},\mathrm{Eval}\left(f,\mathrm{evk},\mathrm{Enc}\left(\mathrm{pk},m_1\right),\mathrm{Enc}\left(\mathrm{pk},m_2\right),\cdots,\mathrm{Enc}\left(\mathrm{pk},m_k\right)\right)\right)=f\left(m_1,m_2,\cdots,m_k\right) \tag{5}$$

在同态加密密文上所做运算解密后得到的明文，与直接对明文做对应运算得到的结果相同。

(4) 紧凑性：对于任意电路f，任意一组密文$\mathrm{ct}_1,\mathrm{ct}_2,\cdots,\mathrm{ct}_k$，经过同态运算后得到的密文长度与电路$f$无关，即

$$\left|\mathrm{Eval}\left(f,\mathrm{evk},\mathrm{ct}_1,\mathrm{ct}_2,\cdots,\mathrm{ct}_k\right)\right|=\mathrm{poly}\left(\lambda\right) \tag{6}$$

其中正确性和语义安全性跟公钥加密方案中的定义相同，这是一个健全的加密体系的必要条件；同态性和简短性则是同态加密方案特有的安全性定义，同态性是同态加密方案独特属性，紧凑性是同态算法有意义的前提[21]，如果没有紧凑性的限制，假设将原始密文输入运算函数$f\left(\mathrm{ct}_1,\mathrm{ct}_2,\cdots,\mathrm{ct}_k\right)$直接作为新密文输出，解密时就需要先对原始密文$\mathrm{ct}_1,\mathrm{ct}_2,\cdots,\mathrm{ct}_k$解密再将解密后的内容$m_1,m_2,\cdots,m_k$放入函数$f$中运算才能得到经过同态运算后的消息$f\left(m_1,m_2,\cdots,m_k\right)$，导致的后果是密文会随着同态运算次数的增加越来越大，这样不仅毫无意义，而且对于一些需要非公开计算的函数f，最后输出与f强相关的密文结果也会泄露f的信息。

自举是由类同态构造全同态加密方案的核心步骤，运算过程中涉及公开加密密钥的密文，为了保证密钥安全性，密钥需要满足循环安全(Circular Security)，这也是Gentry在方案中提出的一个额外假设，下面给出在公钥密码体制(PKE)下的定义：

定义5 循环安全假设[22]

令$n=n(\lambda)$为多项式，$\mathrm{PKE}=(\mathrm{KeyGen},\mathrm{Enc},\mathrm{Dec})$为一个比特公钥加密方案，其私钥长度为$l_{\mathrm{sk}}$。对任意多项式时间的敌手$A$，若此公钥加密方案是$n$-key 循环安全的，则有

$$\mathrm{Adv}^{\mathrm{circ}}_{\mathrm{PKE},\mathrm{A},n}(\lambda):=2\left|\Pr\left[\mathrm{Expt}^{\mathrm{circ}}_{\mathrm{PKE},\mathrm{A},n}(\lambda)=1\right]-\frac{1}{2}\right|=\mathrm{negl}(\lambda) \tag{7}$$

其中$\mathrm{negl}(\lambda)$是可忽略函数，因此敌手获胜的概率是可忽略的。

该定义说明循环安全假设保证了敌手无法区分随机消息的密文与加密私钥的密文，满足了公钥加密方案中语义安全的经典安全性要求。

3 技术路线

根据全同态方案核心技术，可以按照技术路线划分为如下几类：第1类的代表是Gentry[11]在2009年提出的全同态加密方案，它找到了理想格这个既支持同态加法又支持同态乘法的工具，还开创性地提出了自举的思想，使得全同态加密方案可以通过类同态加密方案构造。第2类是Brakerski和Vaikuntanathan[23]在2011年提出的类同态加密方案，它率先提出了密钥切换操作控制密文乘法的维数扩展、模切换操作控制噪声增长速度的想法。在此基础上，后续的同态方案对BV11的效率和噪声控制进行了优化，并设计了适配的自举算法。第3类是Gentry, Sahai和Waters[24]提出的基于矩阵的近似特征向量技术的同态加密方案，它的技术特点是使用了矩阵的密文形式，矩阵密文做乘法避免了向量密文的维数扩张问题，同时将乘法噪声的增长速度由指数级降低到线性级。第4类是Jung Hee Cheon团队[25]在2017年提出近似同态加密方案，它适用于一些不需要精确计算的场景，在自举操作中用多项式函数近似计算替代了精确的比特提取。本节将对每类代表性全同态加密方案的理论基础，加密方案的结构通式，核心的技术特点，算法的发展过程，目前方案存在的局限性以及实现效果等方面进行叙述分析。图1展示了4类全同态方案诞生的时间顺序，括号内列举了其分支下的重要论文，箭头表示两类方案存在一定的相关性但发表时间有先后：BGV方案继承了Gentry类方案的主要思想，但方案构造的基础假设不同，控制噪声的技术也不同。GSW方案与BGV的基础假设相同，但改变了密文结构；CKKS方案的密文结构、同态操作与BGV基本相同，但消息空间不同、适用运算场景不同。

3.1 理想格和自举理论

以2009年Gentry的工作为代表的同态加密方案，这类方案构造的安全性是基于理想格上的困难问题[50]。格上的两个著名困难问题是最近向量问题(CVP)[17]和最短向量问题 (SVP)[16]，分别是在格上找一个已知格点的最近向量和最短向量 ，后续又在此基础上引申出了格约简问题[51]，为给定的格找一组尽可能接近正交的、既正交又短的“好”格基。若已知一组“好”格基，则求解CVP和SVP问题只需要多项式时间，反之则需要指数级的时间。在Gentry的方案中，密钥就相当于一组“好”格基，求解密文可以归约到求解CVP和SVP问题。

3.1.1 具体方案

因为理想本身支持加法和乘法，所以是用于构

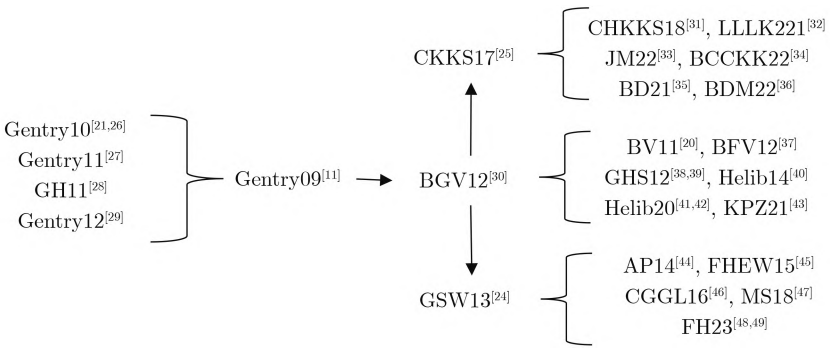

图 1 全同态加密的不同技术路线分支及其代表性方案

建全同态加密方案的自然数学对象。Gentry在方案中定义的理想具有两个性质：首先在理想中假设一个随机元素来加密消息，保证加密的安全性；其次生成一个消除理想的秘密陷门。依据这两个属性可以实现密文的同态加法和同态乘法。方案的主要思想是先构造一个支持有限次加法和乘法同态计算的类同态加密方案，然后借助稀疏子集和问题(sparse subset sum problem,SSSP)压缩解密电路[26]，降低电路深度，使得这个解密电路可以被同态计算，接着又基于循环安全假设(Circular Security)进行同态解密操作，最终实现全同态加密。这里给出以Gentry09为代表的全同态加密方案的一般结构：

密钥生成：对于给定的环R和理想I的基B_I，$\mathrm{IdealGen}\left(R,B_I\right)=\left(B_J^{\mathrm{sk}},B_J^{\mathrm{pk}}\right)$，满足$I+J=R$，其中$B_J^{\mathrm{sk}},B_J^{\mathrm{pk}}$为理想$J$的基$B_J$组成的私钥和公钥。

加密：选择随机向量$\boldsymbol{r},g$，消息$m\in\{0,1\}$，生成密文$\mathrm{ct}=\mathrm{Enc}\left(m\right)=m+\boldsymbol{r}\cdot B_I+\boldsymbol{g}\cdot B_J^{\mathrm{pk}}$

解密：$\mathrm{Dec}(\mathrm{ct},B_J^{\mathrm{sk}})=(\mathrm{ct}-B_J^{\mathrm{sk}}\cdot(B_J^{\mathrm{sk}})^{-1}\cdot\mathrm{ct})\mathrm{mod}\ B_I$

密文加法：$\mathrm{ct}_1+ct_2=\mathrm{Enc}\left(m_1\right)+\mathrm{Enc}\left(m_2\right)=m_1+m_2+\left(\boldsymbol{r}_1+\boldsymbol{r}_2\right)B_I+\left(\boldsymbol{g}_1+\boldsymbol{g}_2\right)\cdot B_J^{\mathrm{pk}}$

密文乘法：

$$\begin{aligned}&\mathrm{ct}_1\cdot ct_2=\mathrm{Enc}\left(m_1\right)\cdot\mathrm{Enc}\left(m_2\right)\\&=m_1m_2+\left(m_1\boldsymbol{r}_2+m_2\boldsymbol{r}_1+\boldsymbol{r}_1\boldsymbol{r}_2B_I\right)B_I\\&\quad+\left(m_1\boldsymbol{g}_2+m_2\boldsymbol{g}_1+\boldsymbol{g}_1\boldsymbol{g}_2B_I^{\mathrm{pk}}\right)B_J^{\mathrm{pk}}\\&\quad+\left(\boldsymbol{r}_1\boldsymbol{g}_2+\boldsymbol{r}_2\boldsymbol{g}_1\right)B_IB_J^{\mathrm{pk}}\end{aligned}$$

此类同态方案正确解密的要求是噪声量$\mathrm{ct}\mathrm{mod}B_J^{\mathrm{pk}}$要小于$B_J^{\mathrm{pk}}/2$，因此乘法操作后剧烈增长的噪声会导致密文解密失败，限制了同态操作的继续进行。

3.1.2 自举算法

Gentry在著作中提出了的一个关键技术——自举，这个术语表示刷新一个密文的操作过程：自举电路由同态方案的解密电路组成，旧密文输入这个自举电路产生一个新的密文，该密文加密相同的消息，但具有较低的噪声水平，可以对其进行更多的同态操作。自举是迄今为止大部分全同态加密方案的核心，更通俗的讲，自举过程就是把一个噪音马上达到临界点的同态密文二次加密，并且用加密的私钥同态计算解密算法，把里层的密文解密成明文，只留外层的加密，这样就获得了一个全新的低噪音同态密文且加密的内容不变。为了实现自举，Gentry设计了一种压缩类同态方案解密函数的方法。这种方法将原来的类同态方案ε转换成另一个具有相同的同态容量和更简单的允许自举的解密函数全同态加密方案ε^*。同时Gentry[26]证明了在同态计算公钥中添加一些关于私钥的“额外信息”就足够了，降低了解密算法的复杂性。因为自举过程中用到了私钥的加密，所以需要针对私钥设置一个额外的安全假设，称为循环安全假设。这个假设意味着公开发布用公钥加密的私钥密文不会影响私钥的安全性，即它保证了敌手无法观察到该方案在公钥下加密的私钥信息。目前的全同态加密方案结构都需要在自举中用到私钥的加密，因此所有已知的全同态结构都要求循环安全性，但是目前循环安全性并没有严谨的数学证明，因此它通常被作为方案基础安全假设之上的附加假设。

接下来将给出自举的具体算法流程。考虑一个类同态加密方案ε，两个公私钥对$(\mathrm{pk}_1,\mathrm{sk}_1)$，$(\mathrm{pk}_2,\mathrm{sk}_2)$，加密算法Enc，公钥$\mathrm{pk}_1$加密消息$m$得到的密文$\mathrm{ct}=\mathrm{Enc}\left(\mathrm{pk}_1,m\right)$。对密文ct的自举分为以下3步：

(1) 用公钥pk_2加密私钥sk_1，生成自举密钥$\boxed{\mathrm{sk}_1}$：$\mathrm{Enc}\left(\mathrm{pk}_2,\mathrm{sk}_1\right)\rightarrow\boxed{\mathrm{sk}_1}$

(2) 用公钥pk_2加密密文ct(非必需步骤[1])：$\mathrm{Enc}\left(\mathrm{pk}_2,\mathrm{ct}\right)\rightarrow\boxed{\mathrm{ct}}$

(3) 使用自举密钥同态解密新密文：$\mathrm{Dec}\left(\boxed{\mathrm{sk}_1},\boxed{\mathrm{ct}}\right)$。

[1] 通常自举算法只需自举密钥与密文做乘法即可同时完成密文的解密与重加密

通过这种方式，就可以在公钥pk_2下获得同一消息m的加密$\mathrm{ct}_{\mathrm{new}}=\mathrm{Enc}\left(\mathrm{pk}_2,m\right)$。▫表示对方框里内容的加密。

自举是为了降低密文同态操作中引入的噪声。从概念上讲，自举调用解密函数并同时执行重加密，这些操作生成了“新”密文，这个“新”密文的噪声包含了重加密的噪声加上解密电路同态计算导致的噪声。因此，自举得到的“新”密文噪声大于用加密算法得到的初始密文噪声，但小于用比解密电路深度更高的同态求值函数得到的密文噪声。将自举应用于在多次同态运算后的密文，这使得构造任意大深度函数的同态电路成为可能，前提是解密电路必须可同态运算。2011年，Gentry和Halevi[28]尝试用类同态加密和乘法同态加密的混合体构造全同态加密方案，用Decision Diffie-Hellman替换SSSP假设，这个方案仍然依赖于自举。Gentry等人[29]的研究结果证明，构造一个全同态方案的问题足以转移到构造一个只能对特定的小次函数集求值的方案，即类同态方案，且当该方案满足循环安全假设。满足上述情况的类同态加密方案，则可以应用自举(必要时还可以压缩)来获得全同态方案，因此，自举是由类同态加密方案构造全同态加密方案的唯一途径。不幸的是，自举在计算上很复杂，同时还需要很大的内存空间。

3.1.3 研究现状

Gentry的工作做出了许多理论贡献，但方案中包含的一些数学概念过于复杂难以实现，而且运算成本过高，单个比特的自举就需要30 min[27]。虽然此时全同态方案巨大的开销制约了其实用性，但是自举思想的提出为后续全同态加密方案的诞生提供了实现途径。后来的全同态加密方案设计都据此思想：先构造一个高效的类同态加密方案，再根据方案设计相应的高效自举算法，最后令类同态加密方案调用自举算法实现全同态加密。此时Gentry方案的性能表明全同态加密距离投入实际应用还有一段路程，简化方案结构、降低成本、提升效率都是当时全同态加密方案设计需要解决的问题。

3.2 密钥切换和模切换

以BFV/BGV为代表的这类全同态加密方案是基于LWE问题和RLWE问题构造的。LWE问题在2005年由Regev提出，且解决LWE问题不比解决格上的CVP和SVP简单[52]；RLWE问题则是LWE问题的代数变体，其安全性可以归约到理想格上最坏情况问题，在2011年被Brakerski和Vaikuntanathan[20]提出。同一时期人们发现，基于RLWE构造全同态加密方案[53]似乎比基于LWE更有优势。在BGV方案中，LWE型密文的明文空间为$\mathbb{Z}_t$，RLWE型密文的明文空间为$R_t=\mathbb{Z}_t\left[X\right]/X^n+1$，但一个LWE密文仅能存储一个明文，而一个RLWE密文最多能存储n个明文，空间利用率大幅度提高。进一步，引入了单指令多数据(Single Instruction Multiple Data, SIMD)操作[54]之后， RLWE密文的n位对应明文全都有效使用，因此后续BGV类的全同态加密方案大多基于RLWE问题设计。

3.2.1 具体方案

接下来以BGV为例，给出这类全同态加密方案的一般结构：

密钥生成：给定安全参数λ，$\mathrm{KeyGen}\left(1^\lambda\right)\to\mathrm{pk},\mathrm{sk}$，生成公钥pk，私钥$\mathrm{sk}=(1,s)$

加密：$\mathrm{Enc}(m,\mathrm{pk})=\mathrm{ct}=(c_0,c_1)\in R_q^2$

解密：$\mathrm{Dec}(\mathrm{ct},\mathrm{sk})=((c_0+c_1\cdot s\mod q)\mod t)=m\in R_t$

密文加法：$\mathrm{ct}_1+\mathrm{ct}_2=(c_{10},c_{11})+(c_{20},c_{21})=(c_{10}+c_{20},c_{11}+c_{21})=\mathrm{ct}_{\mathrm{add}}$

密文乘法：$\mathrm{ct}_1\cdot ct_2=(c_{10},c_{11})\cdot(c_{20},c_{21})=(c_{10}c_{20},c_{10}c_{21}+c_{11}c_{20},c_{11}c_{21})=(c_0,c_1,c_2)=\overline{\mathrm{ct}}$，

$$\overline{\mathrm{ct}}=(c_0,c_1,c_2)\xrightarrow{\text{keyswitch}}(c_0{}',c_1{}')=\mathrm{ct}',$$

$$\mathrm{ct}'\mod q'\xrightarrow{\text{modswitch}}\mathrm{ct}_{\mathrm{mult}}\mod q$$

2011年，Brakerski和Vaikuntanathan[23]基于LWE问题设计了一个不使用自举技术就能增加计算轮次的类同态加密方案(BV)，它用重线性化技术来控制密文的维数膨胀，用模切换技术来控制噪声增长，将密文的尺寸缩小到了$O\left(\lambda\log_2\lambda\right)$。但由于LWE问题本身的数学结构限制，导致BV方案的空间利用率并不高，所以在BV方案的基础上，2012年就出现了基于RLWE的优化BV同态加密方案[37,30]，分别从不同角度改进拓展了BV的核心技术，使得BV类同态加密方案的性能得到进一步提升。

2012年，Fan和Vercauteren[37]将基于LWE问题的Brakerski全同态方案移植到RLWE中构造了BFV方案，提出了两个优化版本的重线性化算法，用来解决乘法后密文维数增加的问题。与BV方案相比，BFV的算法缩小了重线性化密钥，计算速度得到提升。第1种重线性化方法是选一组合适的基T，将重线性密钥rlk在这组基上进行分解，减小重线性化过程中由于密文乘法引入的噪声。

$$\begin{aligned}&\mathrm{rlk}\to\Sigma_i^l\mathrm{rlk}\left[i\right],\ \overline{\mathrm{ct}}=(c_0,c_1,c_2)\\&\quad\to\left(c_0,c_1,\Sigma_1^lT^ic_2\left[i\right]\right),\\&\mathrm{ct}_{\mathrm{mult}}=(c_0,c_1)+c_2\cdot\mathrm{rlk}\\&\qquad=\Big(\left[c_0+\Sigma_i^l\mathrm{rlk}\left[i\right]\left[0\right]\cdot c_2\left[i\right]\right]_q,\\&\qquad\quad\left[c_1+\Sigma_i^l\mathrm{rlk}\left[i\right]\left[1\right]\cdot c_2\left[i\right]\right]_q\Big)\end{aligned}\tag{8}$$

另一种重线性化方法则借助模切换的思想，先将密文切换到大模数，做完重线性化之后再切换回小模数，利用模约简缩小重线性化乘法中引入的密文噪声。

$$\text{rlk} \bmod (pq)，\ \overline{\text{ct}} \bmod q \to \overline{\text{ct}} \bmod (pq)，$$

$$\begin{aligned}\text{ct}_{\text{mult}} \bmod q &= (c_0, c_1) + [c_2 \cdot \text{rlk}]_p \\ &= \left(c_0 + \frac{c_2 \cdot \text{rlk}[0]}{p}, c_1 + \frac{c_2 \cdot \text{rlk}[1]}{p}\right) \bmod q\end{aligned} \tag{9}$$

BGV方案是Brakerski等人[30]在2012年提出的一种基于RLWE无自举的全同态方案，主要贡献是优化了密钥切换技术(keyswitch)和控制密文噪声增长速度的模切换技术(modswitch)。

$$\text{keyswitch} : \overline{\text{ct}} = (c_0, c_1, c_2) \to (c_0', c_1') = \text{ct}' \tag{10}$$

$$\text{modswitch} : \text{ct}' \bmod q^l \to \text{ct}'/q = \text{ct}_{\text{mult}} \bmod q^{l-1} \tag{11}$$

密文同态乘法的实质是向量张量积，会造成密文维度的增加，此时就需要密钥切换技术来降低维数，确保密文能正确解密。同时乘法后的密文噪声也会增加，用模切换技术将密文切换到小模数可以同步的缩小密文和噪声，减小乘法噪声的增长速度，使密文支持更多次的同态乘法。

此时BGV方案1次只能对单个密文进行运算，运算效率低下，为了降低BGV方案的开销、提升计算效率，人们开始思考如何同时处理多个密文。在2014年，Smart和Vercauteren[54]在BGV方案中引入了SIMD操作，通过在多项式环上使用中国剩余定理，令密文呈现多槽 (slots)结构 ，允许把明文数组插进槽中，这样对两个密文进行同态操作的时候，即对数组向量按分量进行同态操作，提高了空间利用率和计算效率，降低了密文转换率，使得基于RLWE的同态加密方案相较于LWE方案呈现了巨大的优势。2012年，Gentry团队[38,39]针对SIMD算法的特点优化了计算电路，将输入消息打包到l个数组中，每层使用SIMD操作，再将输出作为到下一层的输入，对于浅层算数电路可以实现polylog的低开销：对于安全参数λ，任意t个门，深度L，平均宽度$\Omega(\lambda)$的电路，可以在时间$t \cdot \text{polylog}_2(\lambda)$内同态计算。虽然BGV方案中模数会随着同态计算逐渐变小，但上述算法对深度的依赖可以通过使用自举操作来规避：一旦模数小到无法进行进一步的计算，则自举刷新密文，用模数较大的新密文来支持后续计算。

基于(R)LWE问题构造的这些全同态方案，在算法上有一些相似性，但同时也有它们各自的特点。当BGV, BFV的明文空间都是R_t，明文模t，密文模q时，BGV会将消息放在低位加密：$\text{Dec}(\text{ct}_{\text{BGV}}) = m + t \cdot \boldsymbol{e}$，而BFV会把消息放在高位加密：$\text{Dec}(\text{ct}_{\text{BFV}}) = \frac{q}{t \cdot m} + \boldsymbol{e}$，因此在t较大时BGV噪声的增长速度会比BFV小。总而言之，对于较大的明文模数，BGV方案的噪声增加速度较小，计算效率较高，而较小明文模数的情况下，BFV的运算效率较高，表现更好。

3.2.2 自举算法

BFV/BGV类同态加密方案从类同态转变为全同态的方法使用了Gentry的思想，即在密文噪声达到最大噪声界前，用自举步骤刷新密文降低噪声，这样得到的新密文噪声只与同态解密电路相关，与初始噪声无关。自举的具体流程为：模切换→线性变换→比特提取→逆线性变换。模切换步骤的目的是将密文提升到大模数下，需要使用线性变换操作将多项式系数放到明文槽中，比特提取系数的主比特同时去噪，最后进行线性变换的逆过程将明文槽中的系数放回原位，其中复杂度最高的部分是比特提取。2016年，IBM基于BGV开发了一个开源的全同态运算库HElib[40–42]，Halevi和Shoup详细描述了如何在Helib中实现BGV的自举算法。对消息进行稀疏打包的情况下，平均每比特自举仅需1.3 ms，具有实际应用价值。

3.2.3 研究现状

跟据BFV和BGV的这些特点，Andrey Kim团队[43]在2021年对这两个经典全同态加密方案做出了进一步优化，如表1。有趣的是，这两个经典方案的改进方法几乎都源于对方的技术特点。经过上述优化之后，明文模较小时(如取t=2)改进BFV的速度占优，但随着明文模的增大，改进BGV的速度

表 1　Andrey Kim团队对BFV和BGV方案的算法优化

BFV	BGV	优化效果
$\left\lfloor \frac{q}{t} \right\rfloor m \to \left\lfloor \frac{qm}{t} \right\rfloor$预计算$qm$	\	降低初始噪声
将乘法中的模切换步骤置于重线性化之后	\	降低模切换维数
将BFV的模数分层，乘法之后切换到小模数	\	减缓噪声增长速度
\	静态噪声管理	增强方案的实用性

更快：对于所有明文模量，改进BFV方案运算速度均有提升，当明文模取t=2电路深度为20时仅需50 s运算时间，相比原始方案的150 s快了3倍，这归功于在BFV加密过程中将部分内容进行了预计算；当进行7层乘法深度的二叉树计算时，改进BGV仅需1.27 s，快于改进BFV的1.54 s，这两个结果都优于原始BFV的2.48 s。

3.3 密文矩阵和非对称噪声

2013年Gentry等[24]提出了基于矩阵特征值和特征向量构造的GSW方案，标志着全同态方案引入了新的密文结构——矩阵，因为矩阵的特征值与特征向量具有线性关系$\boldsymbol{Ax}=\lambda\boldsymbol{x}$，在原始体系中引入随机噪音$\boldsymbol{e}$，就可以得到一个LWE型的加密方案$\boldsymbol{C}\cdot\boldsymbol{s}=\mu\boldsymbol{s}+\boldsymbol{e}$，其中$C$为密文，$\boldsymbol{s}$为密钥，$\mu$为加密的消息，$\boldsymbol{e}$为随机噪声。GSW方案的安全性可以归约到与BFV/BGV方案相同的LWE问题和RLWE问题，不同的是，以BGV, BFV为代表加密方案的密文和密钥都是以向量形式表示，需要额外的密钥切换技术来控制乘法运算后密文的维数扩展，而以GSW为代表加密方案的矩阵密文做乘法不会改变密文维数，避免了密钥交换技术需要用到的大量密钥交换的矩阵而导致公钥长度的增长。

3.3.1 具体方案

GSW方案继承了BGV方案的设计思路，并在此基础上加入了自己的核心技术——近似特征向量技术，构造了一个无需同态计算公钥就可以进行同态运算的“层次型”同态加密方案。GSW类方案的特点是同态密文运算为简单的矩阵运算，相比于向量结构的密文，矩阵结构虽然不支持消息打包，但避免了维数膨胀问题和密钥切换运算。GSW类同态加密方案的一般结构如下：

密钥生成：随机抽取一个向量$\bar{\boldsymbol{s}}\in\mathbb{Z}_q^{n-1}$，生成私钥$\boldsymbol{s}\to(\bar{\boldsymbol{s}},-1)^{\mathrm{T}}\in\mathbb{Z}_q^n$

加密：给定消息u和n阶单位矩阵$\boldsymbol{I}_n$，随机选取矩阵$\boldsymbol{A}\in\mathbb{Z}_q^{n\times(n-1)}$，噪声向量$\boldsymbol{e}\to\chi_{\boldsymbol{e}}^n$，生成密文矩阵$\boldsymbol{C}=(\boldsymbol{A},\boldsymbol{A}\cdot\bar{\boldsymbol{s}}+\boldsymbol{e})+u\cdot\boldsymbol{I}_n\in\mathbb{Z}_q^{n\times n}$

解密：$\mathrm{Dec}(\boldsymbol{C},\boldsymbol{s})=\boldsymbol{C}\cdot\boldsymbol{s}=u\cdot\boldsymbol{s}-\boldsymbol{e}$，这里噪音向量$\boldsymbol{e}$表示引入的LWE噪声，后文统一取正号

密文加法：矩阵加法$\boldsymbol{C}_1+\boldsymbol{C}_2$，$\mathrm{Dec}(\boldsymbol{C}_1+\boldsymbol{C}_2)=(u_1+u_2)\cdot\boldsymbol{s}+(\boldsymbol{e}_1+\boldsymbol{e}_2)$

密文乘法：矩阵乘法$\boldsymbol{C}_1\cdot\boldsymbol{C}_2$，$\mathrm{Dec}(\boldsymbol{C}_1\cdot\boldsymbol{C}_2)=(u_1\cdot u_2)\cdot\boldsymbol{s}+u_2\cdot\boldsymbol{e}_1+\boldsymbol{C}_1\cdot\boldsymbol{e}_2$

可以观察到密文乘法中噪声并不是对称增长的，即$\boldsymbol{C}_1\cdot\boldsymbol{C}_2$与$\boldsymbol{C}_2\cdot\boldsymbol{C}_1$引起的噪声增长不同。因为密文矩阵$\boldsymbol{C}$远大于消息$u$，所以$\boldsymbol{C}\cdot\boldsymbol{e}$在乘法噪声中占主导地位。因为GSW密文正确解密的噪声上界为$q/4$，所以为了控制乘法中产生的噪声项$\boldsymbol{C}\cdot\boldsymbol{e}$，就需要限制密文$\boldsymbol{C}$的大小，GSW方案中使用二进制分解的思想定义了函数

$$\begin{aligned}&\mathrm{BitDecomp}(\boldsymbol{a})\\&=(a_{1,1},a_{1,2},\cdots,a_{1,m},a_{2,1},\cdots a_{2,m},a_{n,1},\cdots,a_{n.m})\end{aligned}\tag{12}$$

为向量$\boldsymbol{a}=(a_1,a_2,\cdots,a_n)$从最低位到最高位每项系数的比特分解

$$\begin{aligned}&\mathrm{BitDecomp}^{-1}\left(\boldsymbol{b}\in\mathbb{Z}_q^{nm}\right)\\&=\left(\sum\nolimits_j^{m-1}2^jb_{1,j}\bmod q,\right.\\&\quad\sum\nolimits_j^{m-1}2^jb_{2,j}\bmod q,\cdots,\\&\quad\left.\sum\nolimits_j^{m-1}2^jb_{n,j}\bmod q\right)\end{aligned}\tag{13}$$

为$\mathrm{BitDecomp}(\cdot)$的逆函数

$$\mathrm{Flatten}\left(\boldsymbol{b}\in\mathbb{Z}_q^{nm}\right)=\mathrm{BitDecomp}\left(\mathrm{Bitcomp}^{-1}(\boldsymbol{b})\right)\tag{14}$$

生成一个系数为$\{0,1\}$的向量

$$\begin{aligned}\mathrm{Powersof2}(\boldsymbol{s})=&\left(s_1,2s_1,\cdots,2^{m-1}s_1,s_2,2s_2,\cdots,\right.\\&\left.2^{m-1}s_n\right)\bmod q\end{aligned}\tag{15}$$

对于任意$\boldsymbol{a},\boldsymbol{s}\in\mathbb{Z}_q^n$，满足$\langle\boldsymbol{a},\boldsymbol{s}\rangle=\langle\mathrm{BitDecomp}(\boldsymbol{a}),\mathrm{Powersof2}(\boldsymbol{s})\rangle$;

对于任意$\boldsymbol{b}\in\mathbb{Z}_q^{nm}$，$\boldsymbol{s}\in\mathbb{Z}_q^n$，满足

$$\begin{aligned}\langle\boldsymbol{b},\mathrm{Powersof2}(\boldsymbol{s})\rangle&=\langle\mathrm{BitDecomp}^{-1}(\boldsymbol{b}),\boldsymbol{s}\rangle\\&=\langle\mathrm{Flatten}(\boldsymbol{b}),\mathrm{Powersof2}(\boldsymbol{s})\rangle\end{aligned}$$

使用上述函数，对密文矩阵$\boldsymbol{C}$进行二进制分解，即可实现降低乘法项的密文尺寸，控制噪声增长的目的。取$m=n\cdot\log_2q$，根据上述函数定义一个$m\times n$维的二进制重组矩阵

$$\mathbf{G}=\begin{pmatrix}1&0&\cdots&0\\2&0&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\2^{\log_2q-1}&0&\cdots&0\\0&1&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&2^{\log_2q-1}\end{pmatrix}\tag{16}$$

在加密的时候，生成密文$\boldsymbol{C}=(\boldsymbol{A},\boldsymbol{A}\cdot\bar{\boldsymbol{s}}+\boldsymbol{e})+u\cdot\boldsymbol{G}$，然后不直接输出$\boldsymbol{C}$，而是输出密文分解后的二进制矩阵$\hat{\boldsymbol{C}}=\boldsymbol{G}^{-1}(\boldsymbol{C})$。密文$\widehat{\boldsymbol{C}_1}$与$\widehat{\boldsymbol{C}_2}$进行乘法运算之后，$\mathrm{Dec}\left(\widehat{\boldsymbol{C}_1}\cdot\widehat{\boldsymbol{C}_2}\right)=(\boldsymbol{u}_1\cdot\boldsymbol{u}_2)\cdot\boldsymbol{G}\cdot\boldsymbol{s}+\boldsymbol{u}_2\cdot\boldsymbol{e}_1+\widehat{\boldsymbol{C}_1}\cdot\boldsymbol{e}_2$这里的$\widehat{\boldsymbol{C}_1}$为二进制矩阵，当噪声界取$\delta$时，$\widehat{\boldsymbol{C}_1}\cdot\mathbf{e}_2$的上界为$n\cdot\log_2q\cdot\delta$，远小于$\boldsymbol{C}_1\cdot\boldsymbol{e}_2$的上界$m\cdot q\cdot\delta$。

除此以外，观察到$\boldsymbol{G}$的第1行$\boldsymbol{G}_1=(1,0,\cdots,0)$只有第1项是1其余为0，在解密阶段$\mathrm{Dec}\left(\hat{\boldsymbol{C}},\boldsymbol{s}\right)=\hat{\boldsymbol{C}}\cdot\boldsymbol{G}\cdot s=u\boldsymbol{G}\cdot\boldsymbol{s}+\boldsymbol{e}$，则$u\mathbf{G}\cdot\mathbf{s}$的第1行有$(u\cdot\boldsymbol{G}\cdot\boldsymbol{s})_1=u\boldsymbol{G}_1\cdot s=u\boldsymbol{s}[1]$，因为$\boldsymbol{s}[1]\in\mathbb{Z}_q$大概率是个远大于噪声上限的数字，所以只需计算观察结果向量的第一项，就可以得到消息u。

3.3.2 自举算法

目前GSW类方案主要为单比特的加密，因此自举过程不涉及密文的打包与解包，仅需同态计算解密函数，主要涉及两个步骤，第一步是用于同态解密的累加：包括内积操作和模约简，第二步是比特提取：进行同态舍入操作，最后得到自举后的新密文。

对GSW类加密方案来说，调整方案构造，使用比较短的参数就可以满足安全性，这样便于引入一些非常高效的自举技术，因此后续的一系列研究都专注于提升方案的自举性能。Jacob Alperin-Sheriff和 Chris Peikert[44]在2014年利用对称群和置换矩阵的思想，设计了一个GSW方案的单比特快速自举算法，为了完全重新随机化误差，在构造格陷门时把使用的矩阵$\boldsymbol{G}^{-1}=\mathrm{BitDecomp}\left(\boldsymbol{I}_N\right)$从一个确定性的比特分解，重新定义为一个随机抽样的矩阵。此外他们还依据中国剩余定理将大的加法群嵌入小的对称群，这样自举需要的同态操作次数就从$\tilde{O}\left(\lambda^3\right)$降到了$\tilde{O}\left(\lambda\right)$次。在2015年，Ducas和Micciancio[45]将AP14方案的思想扩展到环上，用模4操作来同态计算两个密文的NAND与非门，构造了一个基于环GSW的同态累加器用于执行密文的自举过程，将自举速度提升到每次1 s，实验中自举1个比特单次只需要0.69 s。Ilaria Chillotti 团队[46]在2016年对GSW密文与LWE密文、GSW密文与GSW密文的乘法做出了进一步的优化，设计了一种新的一般化乘积机制，并在此基础上改进了自举步骤，将单个门自举时间减少到0.1 s。

3.3.3 研究现状

GSW类的全同态加密方案以其优秀的自举速度被认为是目前最理想的全同态加密方案，但是GSW类不同于BGV/BFV类的向量密文结构，矩阵密文无法自然的令一个密文打包加密多条消息或者多个密文并行运算，所以主流使用的GSW类加密方案仍是单比特加密。2018年Steven Myers 和 Adam Shull[47]提出了一种打包密文的自举算法，通过在FHEW的基础上使用递归的计算方法，将计算复杂度降低到$O\left(3^\rho\lambda^{1/\rho}\right),\rho=O\left(1\right)$。进一步，2023年Liu Feng-Hao和Wang Han[48,49]在MS18方案打包算法的基础上，引入了适用FHEW的SIMD算法，将计算复杂度直接优化到$\tilde{O}\left(1\right)$。上述方案对提升GSW类加密方案的自举效率具有巨大理论价值，但是素数幂明文环的设置，盲旋转算法中test向量的构造等实例化参数选择，是该类方案目前的实现瓶颈，因此如何设计具有实用价值的密文打包方法和并行计算算法，进一步提升自举效率，是当前亟待解决的问题。

3.4 非精确解密场景的适应性方案

为了配合一些不需要精确解密的应用场景，Jung Hee Cheon团队[25]在2017年提出了第1个近似同态加密方案——CKKS，该方案BGV为基础，引入正则嵌入的消息编码方式，具有支持浮点数运算的特性。

3.4.1 具体方案

CKKS主要思想是将加密噪声视为消息的一部分，跟BFV, BGV相比，CKKS的特点有两个：一是对引入的噪声跟明文加密时并未进行高低比特位上的分离，即$\langle\mathbf{ct},\mathbf{sk}\rangle=\boldsymbol{m}+\boldsymbol{e}$；二是解密后的明文是原始明文的近似值，即$\mathrm{Dec}\left(\mathrm{Enc}\left(\boldsymbol{m}\right)\right)\approx\boldsymbol{m}$。这些特点让CKKS方案的使用更加灵活，可以适配多种多情况的应用场景。CKKS的密文加法与乘法与BGV类方案计算过程基本相同，故不再赘述。CKKS需要额外关注的是它的编码过程，因为其编码消息是浮点数，不同与其他全同态方案的整数消息，CKKS需要借助正则嵌入矩阵将浮点数消息编码成整数明文。对于一个待编码的消息$z=(a_1+\mathrm{i}b_1,a_2+\mathrm{i}b_2,\cdots,a_{n/2}+\mathrm{i}b_{n/2})\in C^{n/2}$，给定精度因子$\Delta$，正则嵌入矩阵$\boldsymbol{\sigma}$，函数$\pi$，$\zeta$为多项式环$\Phi_M=X^n+1$的本源单位根，满足$\zeta^n=1\mod M$，记编码后的明文$\boldsymbol{m}\in R_t^n$，具体编解码公式为

编码： $\boldsymbol{m}=\mathrm{Encode}(\boldsymbol{z},\Delta)=\boldsymbol{\sigma}^{-1}\left\lceil\Delta\pi^{-1}(\boldsymbol{z})\right\rfloor$

解码： $\boldsymbol{z}=\mathrm{Encode}(\boldsymbol{m},\Delta)=\pi\circ\boldsymbol{\sigma}(\boldsymbol{m}/\Delta)$

其中函数π的作用是将复数向量取共轭再连接到原向量后面，即$\pi^{-1}(\boldsymbol{z})=(a_1+\mathrm{i}b_1,a_2+\mathrm{i}b_2,\cdots,a_{n/2}+\mathrm{i}b_{n/2},a_1-\mathrm{i}b_1,\cdots,a_{n/2}-\mathrm{i}b_{n/2})$。CKKS方案加密的复数域上消息向量，通过正则嵌入打包成整数环上的明文向量，后续就与其他全同态加密方案一样做整数多项式环上的同态运算。

3.4.2 自举算法

CKK方案的自举流程与BGV/BFV类方案基本相同，因为CKKS方案主要针对于无需精确解密场景的应用，所以自举过程中的比特提取步骤可以用近似模函数完成，无需进行同态的精确解密。同BGV/BFV类方案相同，CKKS也将消息存储在明文槽中，因为模切换是对密文系数进行操作，这种操作不会改变消息，所以它在自举过程中不能直接

进行模切换，也需要先进行线性变换，将系数向量放入明文槽中，再通过一个从小模数换到大模数的模切换操作来完成全部自举过程，实现减小密文噪声扩大密文模数的目标。

CKKS的换模操作需要用多项式来近似模约简函数，这个近似过程中产生的近似误差是自举误差的主要来源，如表2。目前主流的思想是选择一个在原点附近与模约简函数形状相近的函数作为模约简函数与近似多项式之间的桥梁，例如sin函数[31]。但是近似函数与模约简函数之间同样会产生近似误差，因此需要找到更接近模约简函数的近似函数。2021年Joon-Woo Lee团队[32]用复合函数的方法在原本sin函数近似的基础上添加了一个逆sin函数，去掉了sin函数与模约简函数之间的近似误差，但是需要额外计算一个逆sin函数的多项式近似。总体而言，此方法提高了自举精度，特别是在小明文模数的情况下具有优异表现，因为在小明文模情况下近似函数与模约简函数之间的近似误差占主导地位，此时自举精度可以达到40.5 bit，计算时间仅需94.7 s。随后的2022年，Jutla和Manohar[33]使用了一个新的近似函数——正弦级数近似，可以实现几乎任意精度的CKKS自举，使用n阶正弦级数近似时误差约为$O\left(\varepsilon^{2n+1}\right)$，优于正弦函数近似误差$O\left(\varepsilon^3\right)$，自举精度也达到了44 bit。CKKS自举过程中由于不可避免的模切换操作引入了误差导致自举精度始终受限，2022年Youngjin Bae团队[34]针对该问题提出了一种新的自举模式——META BTS，是一个类似比特提取的算法，通过重复两次给定的任意自举算法，可以获得一个精度加倍的自举结果，即迭代低精度自举来获得高精度自举，在目标精度n足够大的情况下，n位精度自举的密文模数近似为$O\left(n^{5/4}\right)$。该方案的优势在于将自举算法作为黑盒，可以用于CKKS类的所有自举算法，因此如果未来出现了更先进的自举算法，则同样可以令META BTS在精度方面得到更好的结果。

3.4.3 研究现状

理想状态下只要噪声与消息相比足够小，则该噪声不会破坏消息的有效数字，但人们一直对CKKS的安全性有所质疑。Baiyu Li和Daniele Micciancio[35]在2021年发表的论文中提出，虽然CKKS方案满足传统的IND-CPA安全，但是对于更强的带有特殊解密Oracle敌手——IND-CPA$^{\mathrm{D}}$安全性，可以用恢复密钥攻击打破。这一结论使得人们的目光又聚焦到CKKS方案的安全性优化，2022年Li Baiyu团队[36]利用了noise flood技术，通过给解密后的明文添加噪声令CKKS方案满足IND-CPA$^{\mathrm{D}}$安全性，但这一方法同时也会令解密明文损失将近一半的精度。不论是自举算法中的精度改进，还是近似解密相关安全性研究，CKKS方案始终与噪声联系密切，噪声控制分析也将是CKKS类方案的重点研究内容。

3.5 对比分析

表3列出了4条技术路线全同态加密方案的参数设置、技术特点，运算效率对比。表中的s为稀疏子集的大小，S为密钥长度，d为格的行列式，N为理想格维数，$f(\lambda)$为安全参数λ的函数，n为环维数，q为密文模，t为明文模。首先，因为Gentry类方案基础假设中的理想格参数设置难度较大，而BGV类，GSW类，CKKS类方案都是基于更加简单易用的(R)LWE困难问题，所以这3条技术路线下的全同态加密方案都更加具有实用性且目前已被广泛应用，并且这些方案都有多个算法库支持代码实现，所以Gentry类方案由于实现存在困难逐渐退出人们的研究视野。接下来将主要对除Gentry类以外的3类全同态加密方案进行对比分析。

BGV类方案基于LWE和RLWE构造的全同态加密方案效率有所不同，得益于RLWE的环结构，可以令单个密文加密多个消息，提高了密文利用率，同时降低了公钥的尺寸，平均单比特的自举时间约1 ms，与以自举速度著称的GSW类方案同样做到毫秒级。虽然文中提到的一些最新成果对GSW类方案的SIMD和打包操作出理论支撑，但目前因为技术的局限性，在代码实现上仍不支持上述两种操作。

表中数据显示，GSW类方案密文的矩阵结构导致密文尺寸大于其他方案。矩阵密文的优势在于同态乘法不会有维数扩张问题，而且噪声增长的速

表 2 CKKS 类全同态加密方案自举的不同近似函数及自举精度对比

方案名称	近似函数(方法)	自举精度(bit)
CHKKS[31]	$\sin\left(\frac{2\pi x}{q}\right)\approx x \bmod q$	24
LLLK[32]	$\left(\frac{2\pi x}{q}\right)\arcsin\left(\sin\left(\frac{2\pi x}{q}\right)\right)=x \bmod q$	40.5
JM[33]	$\frac{q}{2\pi}\sum_{k=1}^{n}\beta_k\sin\left(\frac{2\pi k x}{q}\right)\approx x \bmod q$	44
BCCKK[24]	META BTS	255

表 3　Gentry 类,BGV 类,GSW 类,CKKS 类全同态加密方案对比

	Gentry类	BGV类		GSW类	CKKS类
基础假设	理想格	LWE	RLWE	(R)LWE	RLWE
密文空间	$\mathbb{Z}_d$	$\mathbb{Z}_q^n$	R_q^n	$\mathbb{Z}_q^{n\times n}$	R_q^n
密文大小	$f(\lambda)$	$(n+1)\lceil\log_2 q\rceil$	$2n\lceil\log_2 q\rceil$	$(n+1)^2\lceil\log_2 q\rceil^3$	$2n\lceil\log_2 q\rceil$
明文空间	$\mathbb{Z}_2$	$\mathbb{Z}_t$	R_t	$\mathbb{Z}_2$	\
单个密文存储明文数	1	1	n	1	n
公钥大小	$s\cdot\left\lceil 2\sqrt{S}\right\rceil\cdot\log_2 d$	$2n(n+1)\log_2 q$	$2n\lceil\log_2 q\rceil$	$2n(n+1)\log_2 q$	$2n\lceil\log_2 q\rceil$
是否支持消息打包	\	否	是	暂不支持	是
是否支持SIMD	\	是	是	暂不支持	是
同态乘法噪声增长速度	$N^{\Omega(1)}$	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n^2)$
(平均)单比特自举时间	30 min	\	1.3 ms	13 ms	6.43 s

度也会因噪声向量被分解成二进制矩阵而降低，由指数级降低为线性级。GSW类方案的经典方案构造基于LWE假设，仅在自举步骤用到RLWE的环结构，并不涉及密文的打包，因此表中数据除自举时间外都是基于LWE的方案参数。

CKKS因为近似加密的特点会把加密引入的LWE噪声看作消息的一部分，解密明文的精度由精度因子Δ决定，所以与其他全同态加密方案不同，只有消息空间，没有明文空间。

在自举效率方面，Gentry类方案[27]代码实现的效果并不理想，在72 bit的安全强度下，公钥大小约为2.3 GB，自举刷新但比特密文需要30min。相较于基于(R)LWE的全同态加密方案中，公钥最小仅需几十MB，自举速度已经提升到毫秒级，Gentry类方案的自举速度和内存需求明显缺乏竞争力。BGV类方案在算法库Helib[42]的实现中设定128 bit的安全参数，稀疏自举[2)]可以做到平均单比特自举仅用时1.3 ms。GSW类方案在TFHE库中取安全参数为128 bit可以做到13 ms的单比特自举。CKKS类方案的自举时间与精度成正相关，所需精度越高，自举过程中用来近似模函数的近似多项式维数越高，自举时间也就越长。目前CKKS类方案的实现效果[34]聚焦于自举后的密文精度以及同态乘法的最大深度。在算法库HEAAN中选择128 bit安全参数，精度因子$\Delta=2^{36}$，取环维度2^{15}时，自举所需时间为6.43 s，自举后支持的最大乘法深度为14；取环维度2^{16}时，自举所需时间为16.3 s，自举后支持的最大乘法深度为29。

4　应用实现

随着数据安全的普及，各类云服务的广泛应用，全同态加密算法拥有广阔的应用场景和发展空间，目前已经有一些团队根据上述全同态方案构造了相应的算法库，将理论成果转化为高质量实现。

Helib[55]由IBM开发设计，是一个用C++语言编写的同态加密开源软件库，可以在Windows, macOS, Ubuntu, CentOS等操作系统平台上进行安装部署，底层依赖于NTL数论运算库和GMP多精度运算库，支持BGV和CKKS方案的实现。此外Helib还提供许多提升算法运行效率的算法代码，例如Smart-Vercauteren密文打包技术和Gentry-Halevi-Smart优化算法，支持“set”，“add”，“multiply”，“shift”等基本操作指令。2018年IBM发布的新版本Helib库，优化了重线性化算法，使效率进一步提升了15～75 倍。

Microsoft SEAL[56]由 Microsoft 的密码学和隐私研究小组开发，是一个用C++编写的开源同态加密库，可以在多种环境中运行，支持BFV, BGV和CKKS三种同态加密方案。SEAL库的使用要求用户理解许多同态加密的特定概念，它可以进行密文的同态加法和乘法操作，但无法进行密文的比较排序等操作。

PALISADE是一个开源项目，提供格密码构建实例和最新同态加密方案的高效实现。PALISADE专为可用性而设计，支持BGV, BFV, CKKS和FHEW方案以及TFHE方案的变体，包括对应的自举算法。PALISADE还提供后量子公钥加密、代理重加密、多方计算阈值同态、基于身份的加密、基于属性的加密和数字签名等加密方案。目前PALISADE的主创团队已将该项目合并到后继者OpenFHE[57]库中。

TFHE库基于Chillotti团队[58]同年提出的方案

2) 每个明文槽都只包含字段（或环）的一个元素，而不是扩展字段（或环）。

进行设计，它实现了GSW方案的类同态和带自举的全同态。与其他库不同，TFHE库对电路门的数量或其组成没有限制，因此它的电路可由用户自主定义，或用带有自动电路生成的工具生成，可以实现对任意电路的密文同态运算。

HEAAN[59]是由CKKS方案作者团队开发的一个开源算法库，支持定点算术和有理数的近似计算，支持自举算法基于CHKKS18[31]，引入的近似误差基本同于浮点运算误差，取决于具体参数的设置。HEAAN库支持C++和NTL library语言，目前更新到V2.1版本。

OpenFHE[57]是Duality Technologies团队在2022年发布的一个支持目前大部分全同态算法的开源算法库。OpenFHE库继承了PALISADE库的所有功能，并融合了HElib和HEAAN的特定功能，它支持用户友好模式和编译器友好模式，可以由库或外部编译器直接调用同态密文的运算算法，提供了更简单的API、模块化、跨平台支持和硬件加速器集成。

除了上述主流同态实现库之外，还有主要用于探索使用低级处理器原语的高性能同态加密的NFLlib库，探索使用GPU来加速同态加密的cuFHE库，用Go语言编写的Lattigo库等 。表4针对一些基本的使用需求对比了这些实现库的应用特点。

随着全同态库的功能逐渐强大、应用逐渐广泛，全同态加密方案的标准化进程也在逐步推进。同态加密标准联盟一直致力于开发同态加密的社区标准。该组织基于标准化研讨会的三份关于同态加密的安全性、API和应用白皮书，于2018年发布了《同态加密标准》的第一版，并计划在未来版本中增加用于同态加密的标准 API 和编程模型的内容。2021年，国际标准ISO/IEC 18033-8《信息技术-安全技术-第8部分：完全同态加密》立项，旨在推进全同态加密工作机制的规范化。提议的国际标准规定了全同态的加密机制：提供了定义、符号和格式，同时定义了安全模型、假设、消息空间等适合标准化的方案加密机制。

5 总结与展望

自从Genrty开创了全同态的方案构造，经过十多年的发展，全同态加密技术正在从理论走向实际，从一个理论上的方案转变成一种实际可用的工具，也涌现了一些具体的应用场景和性能指标。多方安全计算、密码数据库、区块链隐私保护、机器学习隐私等多种应用领域不断涌现同态密码技术的身影，并将在今后继续发挥出巨大的作用。

一方面，为了适应云环境下多用户快速、稳定地存储和计算密文，全同态加密的方案设计开始面向多用户：门限全同态[60](ThresholdFHE)，多密钥的全同态[61]、多方参与的全同态[62]，这些带交互的方案设计进一步拓宽了全同态加密的应用场景。

另一方面，随着计算机算力的提升，硬件数据处理能力的增强，针对全同态方案基础参数的许多实验性改进将更有实际意义:选择非2次幂的明文环次数[63]，$t=X^m+b$的明文模数[64]，比SIMD更复杂但高效的域指令多数据(Field Instruction Multiple Data, FIMD)打包[65]等，这些技术都致力于缩减同态加密方案的内存占用，提升存储效率，具有推广到所有同态加密方案的潜力。

目前已有的4条同态加密方案技术路线各有不同的挑战和机遇：Gentry基于理想格做出的一系列全同态相关领域研究，奠定了全同态加密的理论基础，但是理想格等数学概念代码实现困难，所以相关研究逐渐遇冷。BGV类方案基于简单易用的(R)LWE假设，便于用计算机编程检验方案性能，此外还擅长多组打包密文的并行处理，因此即使自

表 4 不同实现库的应用特点对比

实现库名称	实现语言	支持加密方案	自举	特色功能
Helib	C++	BGV, CKKS	√	支持HE 汇编语言
SEAL	C++	BFV, CKKS	√	跨平台编译
PALISADE	C++	BGV, BFV, CKKS, FHEW, TFHE	√	跨平台编译，BGV/BFV/CKKS的多方扩展
TFHE	C/C++	GSW, FHEW, TFHE	√	可编程自举
HEAAN	C++	CKKS	√	GPU加速
NFLlib	C++	\	\	基于NTT的快速格计算
cuFHE	C++	TFHE	√	CUDA加速
Lattigo	Go	BGV, BFV, CKKS	√	跨平台编译，剩余数系统(RNS)
OpenFHE	C++	BGV, BFV, CKKS, FHEW, TFHE	√	剩余数系统(RNS), BGV/BFV/CKKS的多方扩展

举时间相对GSW类方案更长，单比特的平均自举时间基本为同一水平。目前BGV类方案的优化方向一是加速自举算法，改进自举中最耗时的比特提取步骤，提升密文同态解密速度；二是优化降噪技术，增加同态乘法的最大深度，尽可能避免复杂的自举操作。GSW类加密方案利用矩阵的代数性质加速了密文乘法，特有的非对称噪声增长也将噪声增长速度从BGV类的指数级降低到了线性级，密文的自举性能相比其他技术路线的同态方案具有极大优势。此外根据GSW类方案的特点还可以用于构造基于身份或属性的加密方案，因为GSW类密文除了具有同态处理密文数据的性质，还支持可编程自举，可以进行非线性运算，所以将认证消息加密成GSW类密文，则可以在不泄露认证消息的前提下，对认证消息的密文进行同态操作，完成身份认证。然而目前身份认证协议主要使用CCA2-secure的传统加密方案[66,67]，现有的全同态加密方案仅支持CCA1的安全性，因此如何令全同态加密方案满足CCA2-secure也是限制其应用领域的一个共性问题。此外GSW类加密方案如何实现消息的打包和密文的并行处理仍是尚未彻底解决的难题，因为目前的理论成果往往需要一些严苛的参数要求，所以距离代码实现投入使用仍有一段距离。CKKS类方案的主要结构与BGV类相似，因此除了BGV类方案存在的问题之外，CKKS类方案始终关注如何选择更好的近似函数提升自举精度，削弱噪声泄露的信息对方案安全性的影响两大问题。总体来说，上述全同态加密方案目前仍无法满足实际应用中云端对大量数据的密文快速处理需求，如何进一步加速理论成果转化，改善全同态加密方案实用性能，仍然是未来研究的重点。

参 考 文 献

[1] BRICKELL E F and YACOBI Y. On privacy homomorphisms (Extended Abstract)[C]. The Workshop on the Theory and Application of Cryptographic Techniques, Amsterdam, The Netherlands, 1988: 117–125. doi: 10.1007/3-540-39118-5_12.

[2] CRAMER R, DAMGÅRD I, and NIELSEN J B. Multiparty computation from threshold homomorphic encryption[C]. The International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, 2001: 280–300. doi: 10.1007/3-540-44987-6_18.

[3] RIVEST R L, ADLEMAN L, and DERTOUZOS M L. On data banks and privacy homomorphisms[J]. *Foundations of Secure Computation*, 1978, 4(11): 169–179.

[4] ELGAMAL T. A public key cryptosystem and a signature scheme based on discrete logarithms[M]. BLAKLEY G R and CHAUM D. Advances in Cryptology. Berlin Heidelberg: Springer, 1985: 10–18. doi: 10.1007/3-540-39568-7_2.

[5] RIVEST R L, SHAMIR A, and ADLEMAN L. A method for obtaining digital signatures and public-key cryptosystems[J]. *Communications of the ACM*, 1978, 21(2): 120–126. doi: 10.1145/359340.359342.

[6] BONEH D, GOH E J, and NISSIM K. Evaluating 2-DNF formulas on ciphertexts[C]. The Second Theory of Cryptography Conference, Cambridge, USA, 2005: 325–341. doi: 10.1007/978-3-540-30576-7_18.

[7] BEHERA S and PRATHURI J R. Design of novel hardware architecture for fully homomorphic encryption algorithms in FPGA for real-time data in cloud computing[J]. *IEEE Access*, 2022, 10: 131406–131418. doi: 10.1109/ACCESS.2022.3229892.

[8] LI Juyan, QIAO Zhiqi, ZHANG Kejia, *et al*. A lattice-based homomorphic proxy re-encryption scheme with strong anti-collusion for cloud computing[J]. *Sensors*, 2021, 21(1): 288. doi: 10.3390/s21010288.

[9] LEE J W, KANG H, LEE Y, *et al*. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network[J]. *IEEE Access*, 2022, 10: 30039–30054. doi: 10.1109/ACCESS.2022.3159694.

[10] 李腾, 方保坤, 马卓, 等. 基于同态加密的医疗数据密文异常检测方法[J]. 中国科学: 信息科学, 2023, 53(7): 1368–1391. doi: 10.1360/SSI-2022-0214.

LI Teng, FANG Baokun, MA Zhuo, *et al*. Homomorphic encryption-based ciphertext anomaly detection method for e-health records[J]. *Scientia Sinica Informationis*, 2023, 53(7): 1368–1391. doi: 10.1360/SSI-2022-0214.

[11] GENTRY C. Fully homomorphic encryption using ideal lattices[C]. The 41st Annual ACM Symposium on Theory of Computing, Bethesda, USA, 2009: 169–169. doi: 10.1145/1536414.1536440.

[12] PEREIRA H V L. Bootstrapping fully homomorphic encryption over the integers in less than one second[C/OL]. The 24th IACR International Conference on Practice and Theory of Public Key Cryptography, 2021. doi: 10.1007/978-3-030-75245-3_13.

[13] BONTE C, ILIASHENKO I, PARK J, *et al*. FINAL: Faster FHE instantiated with NTRU and LWE[C]. The 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, China, 2022. doi: 10.1007/978-3-031-22966-4_7.

[14] KLUCZNIAK K. NTRU-v-um: Secure fully homomorphic encryption from NTRU with small modulus[C]. The 2022 ACM SIGSAC Conference on Computer and Communications

Security, Los Angeles, USA, 2022: 1783–1797. doi: 10.1145/3548606.3560700.

[15] XIANG Binwu, ZHANG Jiang, DENG Yi, *et al*. Fast blind rotation for bootstrapping FHEs[C]. The 43rd Annual International Cryptology Conference, Santa Barbara, USA, 2023. doi: 10.1007/978-3-031-38551-3_1.

[16] STEHLÉ D, STEINFELD R, TANAKA K, *et al*. Efficient public key encryption based on ideal lattices[C]. The 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, 2009. doi: 10.1007/978-3-642-10366-7_36.

[17] SVP challenge[EB/OL]. https://www.latticechallenge.org/svp-challenge/.

[18] REGEV O. On lattices, learning with errors, random linear codes, and cryptography[C]. Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, Baltimore, USA, 2005: 84–93. doi: 10.1145/1060590.1060603.

[19] BLUM A, FURST M, KEARNS M, *et al*. Cryptographic primitives based on hard learning problems[C]. The 13th Annual International Cryptology Conference, Santa Barbara, USA, 1994. doi: 10.1007/3-540-48329-2_24.

[20] BRAKERSKI Z and VAIKUNTANATHAN V. Fully homomorphic encryption from ring-LWE and security for key dependent messages[C]. The 31st Annual Cryptology Conference, Santa Barbara, USA, 2011. doi: 10.1007/978-3-642-22792-9_29.

[21] GENTRY C, HALEVI S, and VAIKUNTANATHAN V. i-Hop homomorphic encryption and rerandomizable Yao circuits[C]. The 30th Annual Cryptology Conference, Santa Barbara, USA, 2010. doi: 10.1007/978-3-642-14623-7_9.

[22] KITAGAWA F and MATSUDA T. Circular security is complete for KDM security[C]. Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, 2020. doi: 10.1007/978-3-030-64837-4_9.

[23] BRAKERSKI Z and VAIKUNTANATHAN V. Efficient fully homomorphic encryption from (standard) LWE[C]. The IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, USA, 2011: 97–106. doi: 10.1109/FOCS.2011.12.

[24] GENTRY C, SAHAI A, and WATERS B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based[C]. The 33rd Annual Cryptology Conference, Santa Barbara, USA, 2013. doi: 10.1007/978-3-642-40041-4_5.

[25] CHEON J H, KIM A, KIM M, *et al*. Homomorphic encryption for arithmetic of approximate numbers[C]. The 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, 2017. doi: 10.1007/978-3-319-70694-8_15.

[26] GENTRY C. Computing arbitrary functions of encrypted data[J]. *Communications of the ACM*, 2010, 53(3): 97–105. doi: 10.1145/1666420.1666444.

[27] GENTRY C and HALEVI S. Implementing Gentry's fully-homomorphic encryption scheme[C]. The 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, 2011: 129–148. doi: 10.1007/978-3-642-20465-4_9.

[28] GENTRY C and HALEVI S. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits[C]. The IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, USA, 2011. doi: 10.1109/FOCS.2011.94.

[29] GENTRY C, HALEVI S, and SMART N P. Better bootstrapping in fully homomorphic encryption[C]. The 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, 2012: 1–16. doi: 10.1007/978-3-642-30057-8_1.

[30] BRAKERSKI Z, GENTRY C, and VAIKUNTANATHAN V. (Leveled) fully homomorphic encryption without bootstrapping[C]. The 3rd Innovations in Theoretical Computer Science Conference, Cambridge, USA, 2012. doi: 10.1145/2090236.2090262.

[31] CHEON J H, HAN K, KIM A, *et al*. Bootstrapping for approximate homomorphic encryption[C]. The 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, 2018. DOI: 10.1007/978-3-319-78381-9_14.

[32] LEE J W, LEE E, LEE Y, *et al*. High-precision bootstrapping of RNS-CKKS homomorphic encryption using optimal minimax polynomial approximation and inverse sine function[C]. The 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 2021. doi: 10.1007/978-3-030-77870-5_22.

[33] JUTLA C S and MANOHAR N. Sine series approximation of the mod function for bootstrapping of approximate HE[R]. Paper 2021/572, 2021.

[34] BAE Y, CHEON J H, CHO W, *et al*. META-BTS: Bootstrapping precision beyond the limit[C]. The 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, USA, 2022: 223–234. doi: 10.1145/3548606.3560696.

[35] LI Baiyu and MICCIANCIO D. On the security of homomorphic encryption on approximate numbers[C]. The 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 2021. doi: 10.1007/978-3-030-77870-5_23.

[36] LI Baiyu, MICCIANCIO D, SCHULTZ M, *et al*. Securing approximate homomorphic encryption using differential privacy[R]. Paper 2022/816, 2022.

[37] FAN Junfeng and VERCAUTEREN F. Somewhat practical fully homomorphic encryption[R]. Paper 2012/144, 2012.

[38] GENTRY C, HALEVI S, and SMART N P. Fully homomorphic encryption with polylog overhead[C]. The 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 2012. doi: 10.1007/978-3-642-29011-4_28.

[39] GENTRY C, HALEVI S, and SMART N P. Homomorphic evaluation of the AES circuit[C]. The 32nd Annual Cryptology Conference, Santa Barbara, USA, 2012. doi: 10.1007/978-3-642-32009-5_49.

[40] HALEVI S and SHOUP V. Algorithms in HElib[C]. The 34th Annual Cryptology Conference, Santa Barbara, USA, 2014. doi: 10.1007/978-3-662-44371-2_31.

[41] HALEVI S and SHOUP V. Design and implementation of HElib: A homomorphic encryption library[R]. Paper 2020/1481, 2020.

[42] HALEVI S and SHOUP V. Bootstrapping for HElib[J]. *Journal of Cryptology*, 2021, 34(1): 7. doi: 10.1007/s00145-020-09368-7.

[43] KIM A, POLYAKOV Y, and ZUCCA V. Revisiting homomorphic encryption schemes for finite fields[C]. Proceedings of the 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, 2021. DOI: 10.1007/978-3-030-92078-4_21.

[44] ALPERIN-SHERIFF J and PEIKERT C. Faster bootstrapping with polynomial error[C]. The 34th Annual Cryptology Conference, Santa Barbara, USA, 2014: 297–314. doi: 10.1007/978-3-662-44371-2_17.

[45] DUCAS L and MICCIANCIO D. FHEW: Bootstrapping homomorphic encryption in less than a second[C]. The 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, 2015: 617–640. doi: 10.1007/978-3-662-46800-5_24.

[46] CHILLOTTI I, GAMA N, GEORGIEVA M, *et al*. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds[C]. The 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, 2016. DOI: 10.1007/978-3-662-53887-6_1.

[47] MYERS S and SHULL A. Practical revocation and key rotation[C]. The Cryptographers' Track at the RSA Conference 2018, San Francisco, USA, 2018. DOI: 10.1007/978-3-319-76953-0_9.

[48] LIU Fenghao and WANG Han. Batch bootstrapping II: Bootstrapping in polynomial modulus only requires $O(1)$ FHE multiplications in amortization[C]. The 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, 2023: 353–384. doi: 10.1007/978-3-031-30620-4_12.

[49] LIU Fenghao and WANG Han. Batch bootstrapping I: A new framework for SIMD bootstrapping in polynomial modulus[C]. The 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, 2023: 321–352. doi: 10.1007/978-3-031-30620-4_11.

[50] MICCIANCIO D and REGEV O. Lattice-based cryptography[M]. BERNSTEIN D J, BUCHMANN J, and DAHMEN E. Post-Quantum Cryptography. Berlin, Heidelberg: Springer, 2009. doi: 10.1007/978-3-540-88702-7_5.

[51] AJTAI M. Generating hard instances of lattice problems (extended abstract)[C]. The Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, USA, 1996: 99–108. doi: 10.1145/237814.237838.

[52] LYUBASHEVSKY V, PEIKERT C, and REGEV O. On ideal lattices and learning with errors over rings[C]. The 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, France, 2010. doi: 10.1007/978-3-642-13190-5_1.

[53] DUCAS L and DURMUS A. Ring-LWE in polynomial rings[C]. The 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, 2012, pp. 34–51. doi: 10.1007/978-3-642-30057-8_3.

[54] SMART N P and VERCAUTEREN F. Fully homomorphic SIMD operations[J]. *Designs, Codes and Cryptography*, 2014, 71(1): 57–81. doi: 10.1007/s10623-012-9720-4.

[55] homenc. HElib[EB/OL].https://github.com/homenc/HElib.

[56] microsoft. SEAL[EB/OL]. https://github.com/microsoft/SEAL.

[57] openfheorg. Openfhe-development[EB/OL]. https://github.com/openfheorg/openfhe-development.

[58] tfhe. Tfhe[EB/OL].https://github.com/tfhe/tfhe.

[59] snucrypto. HEAAN[EB/OL]. https://github.com/snucrypto/HEAAN.

[60] CHOWDHURY S, SINHA S, SINGH A, *et al*. Efficient threshold FHE with application to real-time systems[R]. Paper 2022/1625, 2022.

[61] YUAN Minghao, WANG Dongdong, ZHANG Feng, *et al*. An examination of multi-key fully homomorphic encryption and its applications[J]. *Mathematics*, 2022, 10(24): 4678. doi: 10.3390/math10244678.

[62] PARK J and ROVIRA S. Efficient TFHE bootstrapping in the multiparty setting[R]. Paper 2023/759, 2023.

[63] DI GIUSTO A and MARCOLLA C. Breaking the power-of-two barrier: Noise estimation for BGV in NTT-friendly rings[R]. Paper 2023/783, 2023.

[64] CHEN Hao, ILIASHENKO I, and LAINE K. When HEAAN Meets FV: A new somewhat homomorphic encryption with reduced memory overhead[C]. Proceedings of the 18th IMA International Conference, 2021. doi: 10.1007/978-3-030-92641-0_13.

[65] AUNG K M M, LIM E, SIM J J, *et al*. Field instruction multiple data[R]. Paper 2022/771, 2022.

[66] LI Zengpeng and WANG Ding. Achieving one-round password-based authenticated key exchange over lattices[J]. *IEEE Transactions on Services Computing*, 2022, 15(1): 308–321. doi: 10.1109/TSC.2019.2939836.

[67] WANG Qingxuan, WANG Ding, CHENG Chi, *et al*. Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices[J]. *IEEE Transactions on Dependable and Secure Computing*, 2023, 20(1): 193–208. doi: 10.1109/TDSC.2021.3129512.

戴怡然：女，博士生，研究方向为全同态加密.

张 江：男，副研究员，研究方向为公钥密码可证明安全理论、抗量子密码、多方安全计算协议设计与分析研究.

向斌武：男，博士生，研究方向为抗量子密码算法的设计与安全性分析、 全同态加密.

邓 燚：男，研究员，研究方向为零知识证明、安全归约方法.

责任编辑：马秀强