

本文主要记录隐私计算中的同态加密（Homomorphic Encryption, HE）技术，包括部分同态加密（RSA、GM、ElGamal、Paillier）、近似同态加密（BGN）、有限级数全同态加密和全同态加密（DGHV、BGV、BFV、CKKS、GSW、FHEW、TFHE）等技术，**仅供参考**。

由于**篇幅限制**，将同态加密技术的介绍分为四个部分，**第一部分**讲述部分同态加密和近似同态加密技术；**第二部分**讲述 Bootstrapping 和 DGHV、BGV 全同态加密方案；**第三部分**讲述 SIMD 打包技术与 BFV、CKKS 全同态加密方案；**第四部分**讲述 GSW、FHEW、TFHE 全同态加密方案。

在本文中为**简化表示**，将加密记为 $Enc(\cdot)$ ，解密记为 $Dec(\cdot)$ ，不标明使用的公钥和私钥代表只有一对公私钥。本文中介绍的加密方案基本都依赖于各种**计算难题**，在之前的文章中有过介绍。

四、（有限级数）全同态加密（续）

本文中不单独区分有限级数全同态加密和全同态加密，事实上有限级数全同态加密可以通过 Bootstrapping 技术转换为全同态加密。

按照时间顺序，一般认为到目前为止**共有四代**全同态加密方案，其中 CKKS 方案由于支持浮点数计算（实际是定点数），可以用于隐私机器学习等场景中，按照时间发展通常归属于第四代全同态加密，但其使用的技术以及特性本质上是第二代全同态加密技术，因此本文将其**归类于第二代**进行介绍，总共介绍**三代全同态加密**方案。

第一个全同态加密方案由 Gentry 在 2009 年提出，关键点在于提出了 Bootstrapping 的思想，给后续研究人员提供了一个设计全同态加密算法的蓝图^[6,7,8]。Gentry 提出的同态加密方案较为复杂，且难以实际应用，本文只介绍其中的 Bootstrapping 思想。随后，在 2010 年 van Dijk、Gentry、Halevi 和 Vaikuntanathan 发表了 **DGHV 加密方案**^[9]，该方案基于 Gentry 的设计蓝图，构造了一个在整数上的全同态加密方案，其概念简单但效率较低，在本文中把这两个方案归为**第一代方案**。

第二代全同态加密方案由 BGV 方案^[11]、BFV 方案^[12]和 CKKS 方案^[13]组成，这些方案都利用 **LWE 或 RLWE 问题**来实现。由于这类方案都提供了**控制噪声增长**的手段，且它们 Bootstrapping 的实现较为复杂低效，实际应用中通常把它们作为**有限级数全同态加密**使用。这类方案可以利用 SIMD 技术打包向量进行**并行计算**，有效提高了大批量数据的处理能力。

第三代全同态加密方案由 GSW 方案^[14]、FHEW 方案^[15]和 TFHE 方案^[16]组成。GSW 方案利用 LWE 问题构造了基于**近似特征值**的加密方案，给出了一种新型的加密路径，并关注于**逻辑电路**的全同态计算。随后 FHEW 和 TFHE 在此基础上优化逻辑门的同态计算以及 Bootstrapping 的构造，使得这类方案较好地**支持 Bootstrapping 操作**，效率很高，但不支持像第二代全同态加密方案中的 SIMD 打包技术。

代数	代表方案	核心技术	同态操作类型	SIMD 支持	Bootstrapping	应用特点
第一代	Gentry, DGHV	Ideal Lattice, 整数加密	加法乘法	不支持	必须使用（效率极低）	原理性突破，复杂，主要展示思想
第二代	BGV, BGV, CKKS	LWE / RLWE	加乘等算术操作	支持	可选（效率低）	高效有限次运算，适合批量处理和ML场景
第三代	GSW, FHEW, TFHE	LWE + 基于矩阵的表示	逻辑操作	不支持	高效支持	高速 Bootstrapping，适合电路计算

在介绍上述三代方案前，首先了解一下全同态加密中的关键技术：Bootstrapping 技术。

1. BootStrapping 技术

有限级数全同态加密在同态运算过程中会**增加噪声**，一旦噪声超过**某个界限**，就会掩盖住明文，也就**无法解密**成功了。Bootstrapping 技术的提出就是为了解决噪声增加的问题，它提供了一种重新**刷新噪声**的方案，思想很巧妙也比较容易理解，本文简单介绍一下它的主要思想。

在同态运算过程中，如果噪声快要到达极限，此时可以使用另一个公钥把当前的**密文**和**私钥**分别加密，然后在加密的状态下进行**同态解密**，这就是 Bootstrapping 技术的核心思想。

具体地说，设 (pk_0, sk_0) 和 (pk_1, sk_1) 为两对公私钥，密文 c_0 由 pk_0 加密（即 $c_0 = Enc_0(m)$ ，其中 m 为明文），且由于经历过若干次同态运算噪声快要达到极限，此时利用 pk_1 分别加密 c_0 和 sk_0 得到： $c' = Enc_1(c_0)$ 和 $sk'_0 = Enc_1(sk_0)$ ，随后在 pk_1 加密的系统下同态解密 c_0 得到：

$$Evaluate(c', sk'_0) = Enc_1(Dec_0(c_0)) = Enc_1(m) = c_1$$

此时 c_1 中的噪声极小，可以继续参与同态运算。Bootstrapping 技术给出了一种可以安全更换加密密钥的方法，当 c_1 经过若干同态计算后噪声也达到极限，可以继续利用其他的公钥重复执行上述操作，从而实现无限次的同态运算，也就在理论上实现了真正的全同态加密。

2. DGHV 加密方案

DGHV 加密方案^[9]可以实现对一个比特位的消息 $m \in \{0, 1\}$ 加密并支持全同态运算，可以通过设计算术电路实现对大整数的加密和运算，本文以对 $m \in \{0, 1\}$ 的加密进行介绍。

DGHV 加密方案的安全性依赖于近似最大公约数问题（Approximate Greatest Common Divisor, AGCD），介绍如下：

近似最大公约数问题：简单理解为对于一组整数 $S = \{x_0, x_1, \dots, x_\tau\}$ ，求解这一组数的近似最大公约数 $p \approx \gcd(x_0, x_1, \dots, x_\tau)$ 。在实际应用中，一般取 $x_i = q_i p + r_i$ ，其中 $q_i \gg p$ 且 $r_i \ll p$ 。可以理解为 r_i 是每个整数的一个噪音，正是这些噪音的存在使得求解 S 的近似最大公约数 p 困难。

近似最大公约数问题并不是真的要求解最大公约数，其目的是给定一组表达式为 $x_i = q_i p + r_i$ 的整数，求解其中的 p （即使没有噪音存在， p 也并非一定是真正的“最大”公约数）。

上述近似最大公约数问题要求解的 p 其实就是 DGHV 加密方案的私钥，在该加密方案中会有模 p 运算，该方案定义了所有的模 p 运算结果取 $(-\frac{p}{2}, \frac{p}{2}]$ （所有的模运算都这么操作），例如当 $p = 5$ 时：

- $\{0, 1, 2\}$ 中元素模 p 结果集合仍为 $\{0, 1, 2\}$
- $\{3, 4\}$ 中元素模 p 结果集合变为 $\{-2, -1\}$

定义这种特殊的取模方法是因为 DGHV 方案的加密思想简化后本质上是给 m 加上一个偶数 E （可以简单理解为加上一个噪音）得到的密文 $c = m + E$ 不改变奇偶性，解密时根据奇偶性还原 m 。由于加密时这个偶数可能为负数，而算法中要求模数 p 为奇数，负数对奇数取模会改变奇偶性。例如对 $m = 1$ 加密，模数为 $p = 11$ ，加密噪声 $E = -4$ 时，

- 若采用正常取模方式，会得到： $c = m + E = 1 - 4 = -3 = 8 \pmod{11}$ ，解密时会根据 c 是偶数判断 $m = 0$ 从而解密失败。
- 若采用修改后的取模算法，会得到： $c = m + E = 1 - 4 = -3 \pmod{11}$ ，解密时会根据 c 是奇数判断 $m = 1$ 从而解密成功。

到这里就可以发现，如果加密引入的噪音太大，使得密文超出了 $(-\frac{p}{2}, \frac{p}{2}]$ 的范围，密文的奇偶性改变就会导致无法解密，也就是 DGHV 加密方案中所谓的“噪音掩盖明文”的场景，也是引入 Bootstrapping 技术要解决的问题。

上述加密方案显然是不安全的（因为根据密文的奇偶性就能判断 m 的值），只是用作解释为什么 DGHV 方案要定义这样的取模方式，以及该方案中的噪音是什么。真正的 DGHV 加密方案如下所示。

2.1 密钥生成算法

这里为简化理解省略了安全参数的要求，详细信息可以参考论文。

1. 随机选择一个大奇数 p ，随机生成一组数 $S = \{x_0, x_1, \dots, x_\tau\}$ ，其中每个数表达时为 $x_i = q_i p + r_i$ 。生成过程中要求 x_0 为这组数中最大的数，且 x_0 为奇数， x_0 模 p 为偶数。
2. 上述生成随机数过程中满足 $q_i \gg p$ 而 $|r_i| \ll p$ 。
3. 公钥为 $S = \{x_0, x_1, \dots, x_\tau\}$ ，私钥为 p 。

2.2 加密算法

对于明文 $m \in \{0, 1\}$ ，随机选择一个随机数 r 满足 $|r| \ll p$ （可以理解为噪音），再随机选择 S 的一个不含 x_0 的子集 $S' \subseteq S$ ，计算密文：

$$c = Enc(m) = \left(m + 2r + 2 \sum_{x \in S'} x \right) \bmod x_0$$

这里模 x_0 运算也是 DGHV 中定义的取模方式，本质是为了能够表示负数。

2.3 解密算法

对于密文 c ，使用私钥解密得到明文消息 m 的过程如下：

$$m = Dec(c) = (c \bmod p) \bmod 2$$

正确性

显然密文可以写成 $c = m + 2r + 2 \sum_{x \in S'} x + kx_0$ 的形式（其中 k 为整数），而由于 $x_i = q_i p + r_i$ ，可以得到 $x_i \bmod p = r_i$ ，因此：

$$c \bmod p = \left(m + 2r + 2 \sum_{x_i \in S'} r_i + kr_0 \right) \bmod p$$

考虑到 $r_i \ll p$ ，噪声 $2r + 2 \sum_{x_i \in S'} r_i + kr_0$ 没有超过界限时，可以认为：

$$c \bmod p = m + 2r + 2 \sum_{x_i \in S'} r_i + kr_0$$

由于在密钥生成阶段要求 $x_0 \bmod p$ 为偶数，则 $kx_0 \bmod p = kr_0$ 是偶数，因此可以得到：

$$Dec(c) = (c \bmod p) \bmod 2 = \left(m + 2r + 2 \sum_{x_i \in S'} r_i + kr_0 \right) \bmod 2 = m$$

安全性

由于在密钥生成阶段要求 x_0 为奇数，则加密结果 c 的取值是对 x_0 取模后的值。可以观察到在取模之前 c 的奇偶性和 m 的奇偶性保持一致，但对奇数 x_0 取模会混淆奇偶性，例如 $x_0 = 11$ 时，若取模前 $c = 16$ ，则 $16 \bmod 11 = 5$ 奇偶性发生改变。因此攻击方无法通过密文的奇偶性来判断 m 的取值。

由于加密时会随机选择随机数 r 和子集 $S' \subseteq S$ ，因而容易理解 DGHV 加密方案具有语义安全性。

集合 S 的元素数量以及随机数的生成应当满足一定安全条件，使得攻击方无法通过暴力枚举破解密文。此外，根据近似最大公约数问题的困难性，可以认为公开集合 S 不会暴露私钥 p 。

加法同态性

DGHV 方案满足加法同态性：

$$Enc(m_0) + Enc(m_1) = (m_0 + m_1 + E) \bmod x_0 = Enc(m_0 + m_1)$$

其中， $E = 2r_{m_0} + 2r_{m_1} + 2 \sum_{x \in S'_0} x + 2 \sum_{x \in S'_1} x$ ，该值的大小只在解密阶段起作用，每次加法同态运算后 E 中包含的 r_i 的数量都会增加（即噪声增加），一旦 $E \bmod p = \sum r_i$ 的取值超出某个界限，即使得 $|m_0 + m_1 + \sum r_i| \geq \frac{p}{2}$ ，在解密时密文的奇偶性就会发生改变，也就无法正确解密了。因此当噪声累积到一定程度时需要使用 Bootstrapping 技术刷新噪声，从而实现无限的加法同态运算。

乘法同态性

DGHV 方案满足乘法同态性：

$$Enc(m_0) \cdot Enc(m_1) = (m_0 \cdot m_1 + E) \bmod x_0 = Enc(m_0 m_1)$$

其中， E 类似于加法同态，在模 p 后也是若干 r_i 的和。同样的，每次乘法同态运算后 E 中包含的 r_i 的数量（噪声）都会增加，当噪声累积到一定程度时需要使用 Bootstrapping 技术刷新噪声，从而实现无限的乘法同态运算。

3. BGV 加密方案

自 Brakerski 和 Vaikuntanathan 在 2011 年提出模数转换技术后^[10]，他们在 2012 年和 Gentry 一起基于该技术提出了一套全同态加密方案，即 BGV 加密方案^[11]。该方案利用 RLWE / LWE 难题构造，利用模数转换有效减少了 Bootstrapping 的使用，同时提出了若干优化方法，效率相较于第一代算法提高很多，代表了第二代全同态加密的起点。

下面以基于 **RLWE** 问题构造的 BGV 方案为例进行介绍，设置消息空间为多项式环 $m \in \mathbb{Z}_t[x]/(x^d + 1)$ ，其中 t, d 根据实际应用和安全参数进行选取。该方案中取模操作也沿用 DGHV 方案中定义的操作方式，并将 x 对 q 取模后的结果记作 $[x]_q \in (-\frac{q}{2}, \frac{q}{2}]$ 。

对于多项式环中的元素 $m \in \mathbb{Z}_t[x]/(x^d + 1)$ ，简记为 $m \in R_t$ ，其形式为一个**多项式**：

$$m = a_0 + a_1x + a_2x^2 + \cdots + a_{d-1}x^{d-1}$$

其中系数 a_i 都对 t 取模，且多项式本身对 $x^d + 1$ 取模。

RLWE 问题可以简单理解为，随机选取 $a, s, e \in R_t$ ，其中 s 的范数较小（可理解为**系数较小**）， e 为从某个小噪声分布中采样而来的噪声（**系数也较小**），计算 $b = as + e$ 。此时在安全参数设置合理的情况下，公开 a, b 求解 s 是困难的。

3.1 密钥生成算法

这里为简化理解省略了安全参数的要求，详细信息可以参考论文。

1. 随机选取 L 个大素数 q_1, \dots, q_L 作为模数，满足 $q_L > q_{L-1} > \cdots > q_1 > t$ ，且 $[q_L]_t = [q_{L-1}]_t = \cdots = [q_1]_t = 1$ 。其中 L 根据实际情况自行选取，决定明文空间大小的参数 t 一般也较小（例如 $t = 2$ ）。此外这里 L 个模数的大小关系还有更多的要求，会在后文噪声分析中体现。
2. 随机选取**符合 RLWE 问题**的 $a, s, e \in R_{q_L}$ ，并计算 $b = [as + te]_{q_L}$ 。（此处 s 的系数可以看作只含 $\{-1, 0, 1\}$ ）。
3. 公钥为 a, b ，私钥为 s 。

3.2 加密算法

对于明文 $m \in R_t$ ，随机选取 $r, e_0, e_1 \in R_{q_L}$ ，其中 r 为小元素， e_0, e_1 从小噪声分布中采样。使用公钥加密过程如下：

$$\vec{c} = Enc(m) = (c_0, c_1) = ([m + br + te_0]_{q_L}, [ar + te_1]_{q_L})$$

得到的密文 \vec{c} 可以看作是一个元素为多项式的**二维向量**。

3.3 解密算法

可以把私钥包装为一个二维向量 $\vec{s} = (1, -s)$ ，使用该私钥解密过程如下：

$$m = Dec(\vec{c}) = [[\vec{c} \cdot \vec{s}]_{q_L}]_t$$

正确性

可以观察到在解密时：

$$[\vec{c} \cdot \vec{s}]_{q_L} = [m + br + te_0 - asr - ste_1]_{q_L} = [m + t(re + e_0 - se_1)]_{q_L}$$

上述式子中的 $t(re + e_0 - se_1)$ 部分就属于噪声了，由于噪声中各变量都属于小变量（范数小），因此可以认为 $||m + t(re + e_0 - se_1)|| < \frac{q_L}{2}$ （这里令范数 $||x||$ 为**欧几里德范数**，即取多项式 x 中系数向量的欧几里德范数，相较于无穷范数更方便且紧凑，下同），即 $[m + t(re + e_0 - se_1)]_{q_L} = m + t(re + e_0 - se_1)$ ，从而有：

$$Dec(\vec{c}) = [m + t(re + e_0 - se_1)]_t = m$$

安全性

由于解决 RLWE 问题是困难的，BGV 方案公开 a, b 作为公钥不会泄露 s 的取值，且在加密时 $r, e_0, e_1 \in R_{q_L}$ 都是随机选取的，可以认为 BGV 方案是安全的。

加法同态性

BGV 加密方案具有加法同态性，可以验证：

$$\begin{aligned} Enc(m_0) + Enc(m_1) &= [\vec{c}_0 + \vec{c}_1]_{q_L} \\ &= (m_0 + m_1 + b(r_0 + r_1) + t(e_{00} + e_{01}), a(r_0 + r_1) + t(e_{10} + e_{11})) \\ &= Enc(m_0 + m_1) \end{aligned}$$

在解密时以 $\vec{s} = (1, -s)$ 为私钥执行解密算法即可，正确性可参考前文。

注意到执行同态加法后，噪声变为 $t((r_0 + r_1)e + e_{00} + e_{01} - s(e_{10} + e_{11}))$ ，即同态加法运算会使得**噪声增加**，不过加法产生的噪声增加可以看作是**线性增长**的，较为缓慢，只需注意噪声不要超过界限即可。

乘法同态性

BGV 加密方案具有乘法同态性。记两个明文 m_0, m_1 对应的密文为 $\vec{c}_0 = Enc(m_0) = (c_{00}, c_{10})$ 和 $\vec{c}_1 = Enc(m_1) = (c_{01}, c_{11})$ ，在噪声可控情况下，显然有如下关系成立：

$$\begin{aligned} [c_{00} - c_{10}s]_{q_L} &= m_0 + te_{s0} \\ [c_{01} - c_{11}s]_{q_L} &= m_1 + te_{s1} \end{aligned}$$

从而有：

$$\begin{aligned} [(c_{00} - c_{10}s) \cdot (c_{01} - c_{11}s)]_{q_L} &= [c_{00}c_{01} - (c_{00}c_{11} + c_{10}c_{01})s + c_{10}c_{11}s^2]_{q_L} \\ &= m_0m_1 + te_{s2} + te_s^2 \end{aligned}$$

其中 e_{s0}, e_{s1}, e_{s2} 都为若干从小噪声分布中抽样得到的**噪声的和**， e_s^2 为若干**噪声平方的和**，它们本质上都为噪声，这里不需要关注具体的取值是多少，而是主要关注噪声的**增长幅度**（线性增长和平方增长）。

显然当噪声可控时，即 $||m_0m_1 + te_{s2} + te_s^2|| < \frac{q_L}{2}$ 时，有：

$$[(c_{00} - c_{10}s) \cdot (c_{01} - c_{11}s)]_t = [m_0m_1 + te_{s2} + te_s^2]_t = m_0m_1$$

因此，可以将乘法同态计算后的密文记作 $\vec{c} = (c_{00}c_{01}, c_{00}c_{11} + c_{10}c_{01}, c_{10}c_{11})$ ，而解密私钥为 $\vec{s} = (1, -s, s^2)$ ，从而有：

$$Dec(\vec{c}) = [[\vec{c} \cdot \vec{s}]_{q_L}]_t = m_0m_1$$

到这里似乎 BGV 方案可以支持乘法同态计算了，但是会存在两个问题：

- 密文和私钥在乘法同态计算后从二维向量变成了三维向量，即出现了**维度扩张**，无法支持无限次的乘法同态。
- 噪声在经过乘法同态计算后出现了平方级别的增长**，容易掩盖明文。

于是 BGV 方案为了解决这两个问题分别提出了两个方案：**密钥转换**和**模数转换**。

a. 密钥转换

密钥转换在后续 BFV 方案的介绍中也称作 Relineralisation，其作用是把乘法同态计算后得到的密文 $\vec{c} = (c_{00}c_{01}, c_{00}c_{11} + c_{10}c_{01}, c_{10}c_{11})$ （为方便表示简记为 $\vec{c} = (c_0, c_1, c_2)$ ）转换为 $\vec{c}' = (c'_0, c'_1)$ ，解密密钥转换为 $\vec{s}' = (1, -s)$ ，即维度重新降为**二维**。

在 BGV 的论文中采用的是把密钥转换为**张量积**的形式实现的，这里直接使用 BFV 方案中（版本1）的**多项式表示法**，便于理解，两者的原理本质上是一样的。

密钥转换的实现方式也很简单，只需要用 s 对 s^2 进行加密即可：

$$b = s^2 + as + te$$

此时可以得到：

$$\begin{aligned} c'_0 &= c_0 + c_2b \\ c'_1 &= c_1 + c_2a \end{aligned}$$

验证如下：

$$\begin{aligned}\vec{c} \cdot \vec{s} &= c'_0 - c'_1 s = c_0 - c_1 s + c_2(b - as) \\ &= c_0 - c_1 s + c_2 s^2 + tc_2 e\end{aligned}$$

显然会增加一个噪声项 $tc_2 e$, 若该噪声足够小, 那么正确性就得到了证明。但问题是该**噪声并不小**, 因为 c_2 的取值可能很大, 这时 BGV 为了解决这个噪声问题选择将 c_2 进行**分解** (这里以二进制分解为例, 事实上别的进制也可以), 记 D 为 c_2 的二进制长度, 对 c_2 二进制分解后得到:

$$c_2 = \sum_{i=0}^{D-1} c_{2i} \cdot 2^i$$

并将对 s^2 的加密转换为对 D 个 $s^2 \cdot 2^i$ 的加密:

$$b_i = s^2 \cdot 2^i + a_i s + t e_i \quad (0 \leq i < D)$$

从而有:

$$\begin{aligned}c'_0 &= c_0 + \sum_{i=0}^{D-1} c_{2i} b_i \\ c'_1 &= c_1 + \sum_{i=0}^{D-1} c_{2i} a_i\end{aligned}$$

再次验证如下:

$$\begin{aligned}\vec{c} \cdot \vec{s} &= c'_0 - c'_1 s = c_0 - c_1 s + \sum_{i=0}^{D-1} c_{2i} (b_i - a_i s) \\ &= c_0 - c_1 s + \sum_{i=0}^{D-1} (c_{2i} \cdot 2^i \cdot s^2 + c_{2i} t e_i) \\ &= c_0 - c_1 s + c_2 s^2 + t e_s\end{aligned}$$

其中 $e_s = \sum_{i=0}^{D-1} c_{2i} e_i$, 由于进行了**二进制分解**, $\|c_{2i}\| \leq \sqrt{d}$, 因此噪声项 $t e_s$ 的取值可以认为较小, 相较于二进制分解前的噪声 $tc_2 e$ 大大减小。

所以通过密钥转换技术, BGV 方案能够以较小的噪声代价将密文和私钥缩回二维向量, 解决了上述乘法同态计算中的第一个问题。

b. 模数转换

对于乘法同态计算中引入的**平方级的噪声增长**, BGV 方案并没有直接执行 Bootstrapping 操作 (很费时), 而是引入了**模数转换**技术。

注意到在密钥生成算法中, 生成了 L 个模数, 前文中只提到了其中的 q_L , 这里会给出将模数从 q_i 转换为 q_{i-1} 并降低噪声的方法。

模数转换是将模 q_i 的密文 \vec{c}_i 转换为模 q_{i-1} 的密文 \vec{c}_{i-1} , 与此同时保证解密时的正确性:

$$m \equiv [\vec{c}_i \cdot \vec{s}]_{q_i} \equiv [\vec{c}_{i-1} \cdot \vec{s}]_{q_{i-1}} \bmod t$$

BGV 方案指出这样的 \vec{c}_{i-1} 可以直接计算: 首先计算 $\frac{q_{i-1}}{q_i} \cdot \vec{c}_i$ (计算结果显然会引入小数), 然后取距这个结果最近的与 \vec{c}_i 满足模 t 同余的值, 作为 \vec{c}_{i-1} 的取值。

这里理解起来可能比较晕, 简单举一个例子, 不妨设 $t = 4$ 、 $q_i = 11$ 、 $q_{i-1} = 7$ 、 $\vec{c}_i = (10 + 9x, 9 + 8x)$, 那么计算:

$$\frac{q_{i-1}}{q_i} \cdot \vec{c}_i = \frac{7}{11} \cdot (10 + 9x, 9 + 8x) \approx (6.4 + 5.7x, 5.7 + 5.1x)$$

由于 $[\vec{c}_i]_t = (2 + x, 1)$, 则与 $\frac{q_{i-1}}{q_i} \cdot \vec{c}_i$ 最近且与 \vec{c}_i 满足模 t 同余的值为 $\vec{c}_{i-1} = (6 + 5x, 5 + 4x)$, 通过这样的计算就得到了模数转换的结果 \vec{c}_i 。

注意, 上述内容是 BGV 方案直接告诉我们计算 \vec{c}_i 的**一种方法**, 但并未给出为什么这么计算是正确的, 下面给出证明。

不妨设 $[\vec{c}_i \cdot \vec{s}]_{q_i} = \vec{c}_i \cdot \vec{s} + kq_i$, 其中 k 为某个整数, 随后令 $e_q = \vec{c}_{i-1} \cdot \vec{s} + kq_{i-1}$, 可以证明 $[\vec{c}_{i-1} \cdot \vec{s}]_{q_{i-1}} = e_q$ 。显然这个证明可以转化为证明 $\|e_q\| < \frac{q_{i-1}}{2}$, 证明如下:

$$\begin{aligned}
\|e_q\| &= \|\vec{c}_{i-1} \cdot \vec{s} + kq_{i-1}\| \\
&= \|\vec{c}_{i-1} \cdot \vec{s} + kq_{i-1} + (\frac{q_{i-1}}{q_i} \cdot \vec{c}_i) \cdot \vec{s} - (\frac{q_{i-1}}{q_i} \cdot \vec{c}_i) \cdot \vec{s}\| \\
&= \|(\vec{c}_{i-1} - \frac{q_{i-1}}{q_i} \cdot \vec{c}_i) \cdot \vec{s} + \frac{q_{i-1}}{q_i} \cdot (\vec{c}_i \cdot \vec{s} + kq_i)\| \\
&\leq \|(\vec{c}_{i-1} - \frac{q_{i-1}}{q_i} \cdot \vec{c}_i)[0] \cdot \vec{s}[0]\| + \|(\vec{c}_{i-1} - \frac{q_{i-1}}{q_i} \cdot \vec{c}_i)[1] \cdot \vec{s}[1]\| + \frac{q_{i-1}}{q_i} \cdot \|[\vec{c}_i \cdot \vec{s}]_{q_i}\| \\
&\leq \gamma_R \|(\vec{c}_{i-1} - \frac{q_{i-1}}{q_i} \cdot \vec{c}_i)[0]\| \cdot \|\vec{s}[0]\| + \gamma_R \|(\vec{c}_{i-1} - \frac{q_{i-1}}{q_i} \cdot \vec{c}_i)[1]\| \cdot \|\vec{s}[1]\| + \frac{q_{i-1}}{q_i} \cdot \|[\vec{c}_i \cdot \vec{s}]_{q_i}\|
\end{aligned}$$

其中膨胀因子 γ_R 为控制环上多项式乘法范数膨胀的常数，满足关系式 $\|a \cdot b\| \leq \gamma_R \cdot \|a\| \cdot \|b\|$ 。

注意到 $(\vec{c}_{i-1} - \frac{q_{i-1}}{q_i} \cdot \vec{c}_i)[0]$ 和 $(\vec{c}_{i-1} - \frac{q_{i-1}}{q_i} \cdot \vec{c}_i)[1]$ 实际上是衡量在模数转换计算时， \vec{c}_{i-1} 与 $\frac{q_{i-1}}{q_i} \cdot \vec{c}_i$ 有多“近”，考虑到 \vec{c}_{i-1} 为与 $\frac{q_{i-1}}{q_i} \cdot \vec{c}_i$ 最近且与 \vec{c}_i 满足模 t 同余的值，那么这个距离一定不会超过 $\frac{t}{2}$ (与 x 最近模 t 同余的值一定能在 $x \pm \frac{t}{2}$ 范围内找到)。由于 d 是多项式的度，假设系数全为 $\frac{t}{2}$ ，多项式的范数就为 $\frac{t}{2} \cdot \sqrt{d}$ 。设 $\ell_1(\vec{s}) = \|\vec{s}[0]\| + \|\vec{s}[1]\|$ 为二维向量的 ℓ_1 范数，从而有如下放缩成立：

$$\|e_q\| \leq \gamma_R \cdot \frac{t}{2} \cdot \sqrt{d} \cdot \ell_1(\vec{s}) + \frac{q_{i-1}}{q_i} \cdot \|[\vec{c}_i \cdot \vec{s}]_{q_i}\|$$

假设 $\|[\vec{c}_i \cdot \vec{s}]_{q_i}\| < \frac{q_i}{2} - \frac{q_{i-1}}{q_i} \cdot \gamma_R \cdot \frac{t}{2} \cdot \sqrt{d} \cdot \ell_1(\vec{s})$ (当 $q_{i-1} \ll q_i$ 时容易成立)，就有：

$$\|e_q\| \leq \gamma_R \cdot \frac{t}{2} \cdot \sqrt{d} \cdot \ell_1(\vec{s}) + \frac{q_{i-1}}{q_i} \cdot \|[\vec{c}_i \cdot \vec{s}]_{q_i}\| < \frac{q_{i-1}}{2}$$

因此可以证明出 $[\vec{c}_{i-1} \cdot \vec{s}]_{q_{i-1}} = e_q$ ，又根据 $[q_i]_t = [q_{i-1}]_t$ 且 $[\vec{c}_i]_t = [\vec{c}_{i-1}]_t$ ，从而有：

$$[[\vec{c}_{i-1} \cdot \vec{s}]_{q_{i-1}}]_t = [e_q]_t = [\vec{c}_{i-1} \cdot \vec{s} + kq_{i-1}]_t = [\vec{c}_i \cdot \vec{s} + kq_i]_t = [[\vec{c}_i \cdot \vec{s}]_{q_i}]_t = m$$

证明完毕，可以得出将模 q_i 的密文 \vec{c}_i 转换为模 q_{i-1} 的密文 \vec{c}_{i-1} 的计算方法没有问题。但是我们费尽心思进行模数转换是为了什么呢？我们关注一下模数转换前后的**噪声变化**，上述证明过程给出了如下不等式：

$$\|[\vec{c}_{i-1} \cdot \vec{s}]_{q_{i-1}}\| = \|e_q\| \leq \frac{q_{i-1}}{q_i} \cdot \|[\vec{c}_i \cdot \vec{s}]_{q_i}\| + \gamma_R \cdot \frac{t}{2} \cdot \sqrt{d} \cdot \ell_1(\vec{s})$$

可以看出，若 $\frac{q_{i-1}}{q_i}$ 足够小，可以认为在模数转变后噪声近似变为了原来的 $\frac{q_{i-1}}{q_i}$ 倍，即通过模数转换可以**减少噪声的绝对大小**。

总结完整的同态乘法计算步骤如下：

1. 对齐两个密文的模数（大模数转换为小模数）；
2. 进行**同态乘法**计算，得到三维密文；
3. 进行**密钥转换**，将密文转换为二维；
4. 进行**模数转换**，转为更小的模数。（若 L 个模数使用完，就无法执行同态乘法，必须利用 Bootstrapping 技术刷新）

注意到在模数转换后，模数变成了更小的 q_{i-1} ，这意味着噪音的**上限**也对应着变小了，这时需要人为设置合理的参数来保证通过模数转换缩小噪音是有意义的。

BGV 方案中能够一个噪音上界 B ，并证明对于模任意一个模数 q_i 的合法密文都能保证其噪音小于 B 。显然，刚加密得到的密文满足噪音小于 B ，以同态乘法计算为例（噪音增长最大的情况），给定两个噪音小于 B 的同模数的密文，它们在执行完同态乘法计算后得到的密文噪音也小于 B ，证明如下。

1. 两个密文进行**同态乘法**计算后得到的三维密文噪音上界为 $\gamma_R \cdot B^2$ （具体证明可参考论文，容易理解噪音上界一定位于原始噪音的**平方级别**）。
2. 设执行密钥转换引入的噪音上界为 η_0 ，执行模数转换后额外引入的噪音上界为 η_1 （都可以算出来），从而有同态乘法运算后噪音上界为：

$$\frac{q_{i-1}}{q_i} \cdot (\gamma_R \cdot B^2 + \eta_0) + \eta_1$$

只要通过合理设置参数，满足如下两个条件：

$$\begin{aligned} B &\geq 2 \cdot (\eta_0 + \eta_1) \\ \frac{q_i}{q_{i-1}} &\geq 2 \cdot B \cdot \gamma_R \end{aligned}$$

就可以得到：

$$\begin{aligned} \frac{q_{i-1}}{q_i} \cdot (\gamma_R \cdot B^2 + \eta_0) + \eta_1 &< \frac{q_{i-1}}{q_i} \cdot \gamma_R \cdot B^2 + \eta_0 + \eta_1 \\ &\leq \frac{\gamma_R \cdot B^2}{2 \cdot B \cdot \gamma_R} + \frac{B}{2} \\ &\leq B \end{aligned}$$

得证。

因此 BGV 方案可以通过模数转换技术减少 Bootstrapping 技术的使用，大大提升同态运算的效率。

下一部分将介绍 SIMD 打包技术与 BFV、CKKS 全同态加密方案。

参考文献（续）

- [6] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 2009.
- [7] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proceedings of the forty-first annual ACM symposium on Theory of computing, Bethesda, MD, USA: ACM, May 2009, pp. 169–178.
- [8] C. Gentry, "Computing arbitrary functions of encrypted data," Commun. ACM, vol. 53, no. 3, pp. 97–105, Mar. 2010.
- [9] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," in Advances in Cryptology – EUROCRYPT 2010, H. Gilbert, Ed., Berlin, Heidelberg: Springer, 2010, pp. 24–43.
- [10] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Oct. 2011, pp. 97–106.
- [11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, Massachusetts: ACM, Jan. 2012, pp. 309–325.
- [12] J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," 2012, 2012/144. Accessed: Jul. 15, 2025. [Online]. Available: <https://eprint.iacr.org/2012/144>
- [13] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in Advances in Cryptology – ASIACRYPT 2017, Cham: Springer International Publishing, 2017, pp. 409–437.
- [14] C. Gentry, A. Sahai, and B. Waters, "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based," in Advances in Cryptology – CRYPTO 2013, R. Canetti and J. A. Garay, Eds., Berlin, Heidelberg: Springer, 2013, pp. 75–92.
- [15] L. Ducas and D. Micciancio, "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second," in Advances in Cryptology – EUROCRYPT 2015, E. Oswald and M. Fischlin, Eds., Berlin, Heidelberg: Springer, 2015, pp. 617–640.
- [16] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast Fully Homomorphic Encryption Over the Torus," J. Cryptol., vol. 33, no. 1, pp. 34–91, Jan. 2020.

本文为作者在学习相关知识时的一种记录，便于以后的回顾。作者并没有系统地学习过密码学，因此在表述上可能会存在不严谨甚至出错的地方，文章仅供参考，欢迎大家与我交流，一起进步！

其他平台：

- 知乎（Totoro）：<https://www.zhihu.com/people/totoro-14-60>
- CSDN（_Totoro_）：https://blog.csdn.net/orz_Totoro

- B站 (Totoro_134) : <https://space.bilibili.com/279377771>
- Github (Totoro134) : <https://github.com/Totoro134>
- 公众号 (知识长生所)