桂林電子科技大學
GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY

本科毕业设计（论文）

题目： 关于隐私保护分布式统计的算法研究

学　　号：
姓　　名：
学　　院： 数学与计算科学
专　　业：
指 导 教 师：
指导教师职称：

2022 年 5 月 6 日

摘 要

在分布式统计计算中，若自身的设备无法处理庞大的计算量，可交由第三方服务计算商代为处理.若数据为隐私、机密数据，直接发送可能会造成数据的泄露，引发不可估量的后果.所以将数据加密后，第三方服务计算商对加密后的密文进行运算，接着将结果返回这样保证了数据的隐私性、正确性和机密性.本文通过全同态加密 BFV 加密算法加密数据，在半诚实敌手模型下，每个计算参与方拿到密文的一部分，对密文进行同态操作，接着返回同态操作之后的密文给数据所有者.最后对 BFV 算法在分布式系统下的正确性，安全性，局限性进行分析.

关键词：同态加密；安全多方计算；半诚实敌手模型；全同态加密.

Abstract

In distributed statistical computing, if its own equipment cannot handle the huge amount of computation, it can be handled by a third-party service computing provider. If the data is private and confidential, sending it directly may result in data leakage and incalculable consequences. Therefore, after encrypting the data, the third-party service computing provider operates on the encrypted ciphertext, and then returns the result, which ensures the privacy, correctness and confidentiality of the data. In this paper, data is encrypted by fully homomorphic encryption, and two different schemes are constructed for the case of one data provider and multiple third-party computing service providers, and multiple data providers and one third-party computing service provider. The article gives the complexity, correctness and security of the scheme.

Key words: Homomorphic encryption; secure multi-party computation; secure computing protocol; fully homomorphic encryption

目　　录

1 绪论

1.1 研究背景及意义

信息大爆炸的今天，海量的数据扑面而来.面对这些激增的数据，如果自己的设备无法处理，那么就可以交由有能力的第三方代行计算，这时就涉及到了数据的隐私与机密的问题.例如，一款软件若要在本地处理数据，若用户设备性能低，在本地处理会大大消耗设备资源，造成使用卡顿，用户体验性极差.而将这些数据提交到云服务器上去计算，则能大大解决这一问题.这时迎来了有一个问题，若将用户的隐私数据直接提交给云服务器，就会造成数据的泄露，这就是大数据下的隐私问题.而将数据进行加密，再安全传输协议发送至云服务器，由云服务器代为处理，这也许是一个好的办法.

1.2 研究现状

针对密文计算问题，Rivest 等人[1]在 20 世纪 70 年代首先提出的“同态加密”能实现基本的加密操作之外，还能实现密文间的多种计算功能，即先计算后解密可等价于先解密后计算.这使得数据能由多方提供给云服务器进行密文计算，对数据的机密性，完整性和隐私性得到了有效的保护.同态加密方案支持对密文的计算，并能由密文的计算结果解密得到正确的明文计算结果.在后来的同态加密方案中，有些只支持加法同态，如 Paillier 方案[2]和 Goldwasser-Micali 方案[3]；有些只支持乘法同态，如 Unpadded-RSA 方案[4]和 ElGamal 方案[5].而能同时支持加法和乘法运算的加密方案则较少.Boneh[6]等人提出了 Boneh-Goh-Nissim 加密方案，能够支持任意次的加法操作，但只能支持一次乘法操作.Centry[7,8]提出了一种基于理想格的加密方案，并利用提提出的“Bootstrappable”技术，通过对密文的重加密使密文支持任意次数的同态运算，是真正意义上的全同态加密，也称为全同态加密的研究奠定重要基础.后续的很多同态加密方案[7~13]也都基于 Centry 的“Bootsrappable”技术，这些方案对文献[7]都有不同程度的改进.如 Smart 等人[9]改用整数和多项式实现全同态加密，减少了密文和密钥的长度，Dijk 等人[10]使用整数进行加密，更利于理解.但由于“Bootstrappable”技术本身的复杂性，即使具有较低安全性时，一次“Bootstrappable”操作也需要大约 30s 的时间[13]，因此，这些加密方案都较难应用到实际中，后来，Centry 等人[14,15]从加强全同态算法中的自展技术、全同态加密算法的解密循环的分解技术、实现方法等方面展开了研究，虽然降低了同态加密的复杂性，但仍较难于实际应用.黄汝维等人[16]针对云计算环境的隐私保护问题提出了基于向量和矩阵运算的加密可计算方案（CESVMC），算法运行效率较高，但其算术运算方案不支持加法和乘法的混合匀速那，且仅支持一次乘法和除法匀速那，运算后的密文长度会增大，而且需要记录密文经过的运算类型以完成解密.

1985 年，ElGamal [17]基于有限域上的离散对数困难假设设计了 ElGamal 加密算法，

该加密方法具有乘法同态性，并且满足选择明文不可区分(IND-CPA)安全，但是 ElGamal 的一个不足之处是它的密文成倍扩张.应用最为广泛的当属 Paillier[18]加密系统，基于高阶合数度剩余类困难问题，且具有 IND-CPA 安全.Goldwasser-Micali [19]加密系统属于异或同态加密系统，该加密系统基于二次剩余困难问题，虽具有 IND-CPA 安全，但每次只能加密单比特，因此加密效率会比较低. 1999—2005 年间出现了不少浅同态加密方案，其中最著名浅同态加密方案当属 Boneh[20]等基于理想成员判定困难假设设计的加密方案.该方案能执行一次乘法和若干次加法运算，Boneh [21]等虽然用它成功解决了 2NF 问题，但是该方案在解密时需要搜索解密，因此基于此方案的 2NF 保密计算协议效率很低.2009 年 9 月，身处 IBM 的研究员 Craig Gentry[22]发表一篇论文于 STOC，他解决了一项棘手的数学问题，该问题自几十年前公钥加密发明以来，一致困扰着科学家们.他基于“理想格 idea lattice”的数学对象，使人们可以充分操作加密状态的数据，即可以在不解密的情况下对加密数据进行任何可以在明文上的计算，对加密后的信息仍能深入和无限的分析，保证数据的隐私性、完整性和保密性.这个加密技术被称为全同态加密(full homomorphic encryption).

目前对密文计算在分布式统计方面研究较少，马飞，蒋建国[23]在分布式计算的基础上，运用 Pailier 同态加密提出了在 SMH 模式下，采用 CClient，KClient，SServer 的拓扑结构进行统计分析的计算方案.本文想运用 BFV 全同态加密算法在分布式统计情况下，研究其算法的正确性，安全性和局限性.

2 理论与本文工作

2.1 数据通过何种方式加密.

BFV 算法加密方案来源于文章 "Somewhat Practical Fully Homomorphic Encryption"，它是基于 RLWE (Ring-Learning With Errors)难题的全同态加密方案.密文形式为两个模为$mod\ \ d$的多项式，依据多项式相同幂级项相加的特性，可应用于分布式系统，交由多个计算参与方计算，且计算参与方之间互不干扰.

2.2 本文结构

第三章部分本文的相关理论基础.第二部分分析计算协议对数据的安全.第三部分对方案正确性，安全性，复杂性以及局限性进行分析.

对于本文工作.介绍了在分布式情况下，基于半诚实模型，数据所有者于计算参与方之间时如何完成委托计算的过程，以及数据所有者在保护数据隐私的情况下，如何对数据进行加密，以至于保证数据不被泄露的 BFV 全同态加密算法.最后模拟在多个计算参与方进行多次同态加法之后，对算法进行正确性，安全性，局限性分析.

3 理论基础

3.1 背景知识

同态加密是一种加密形式，具有额外的评估能力，可以在不访问密钥的情况下计算加密数据.这种计算的结果仍然是加密的.同态加密可以看作是公钥密码学的扩展.同态是指代数中的同态，加密和解密函数可以被认为是明文空间M 和密文空间L 之间的同态.同态加密的方案中，有密钥生成算法Gen_ζ，加密算法Enc_ζ，解密算法Dec_ζ

1）密钥生成算法$Gen_\zeta : M \to b_{key}$.$Gen_\zeta$ 根据明文空间M 的随机数生成密钥b_{key}.

2）加密算法$Enc_\zeta : (b_{key}, p_\zeta) \to c_\zeta$.$p_\zeta$ 表示明文空间，c_ζ 表示密文空间，Enc_ζ 利用密钥b_{key}加密明文p_ζ，返回密文c_ζ.

3）解密算法$Dec_\zeta : (b_{key}, c_\zeta) \to p_\zeta$.$Dec_\zeta$ 利用密钥b_{key}解密密文c_ζ，返回明文p_ζ

对于明文空间M 上的多项式

$$f(a_0, a_1, a_2, a_3, \cdots, a_n) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n, a_n (n = 1, 2, 3, \cdots, n) \subset M \qquad (3\text{-}1)$$

及数据$p_1, p_2, p_3, \cdots, p_n$，使用密钥生成算法$Gen_\zeta$，数据$p_1, p_2, p_3, \cdots, p_n$经过加密算法$Enc_\zeta$加密后，得到密文$c_1, c_2, c_3, \cdots, c_n$.这时候第三方对密文$c_1, c_2, c_3, \cdots, c_n$进行计算处理，得到加密后的结果$e$，第三方将结果$e$返回给数据拥有者$F$，数据拥有者$F$ 通过私钥g，结合解密算法Dec_ζ就可以将结果还原为明文形式，即

$$p+q=Dec(Enc(b,p)+Enc(b,q)) \tag{3-2}$$

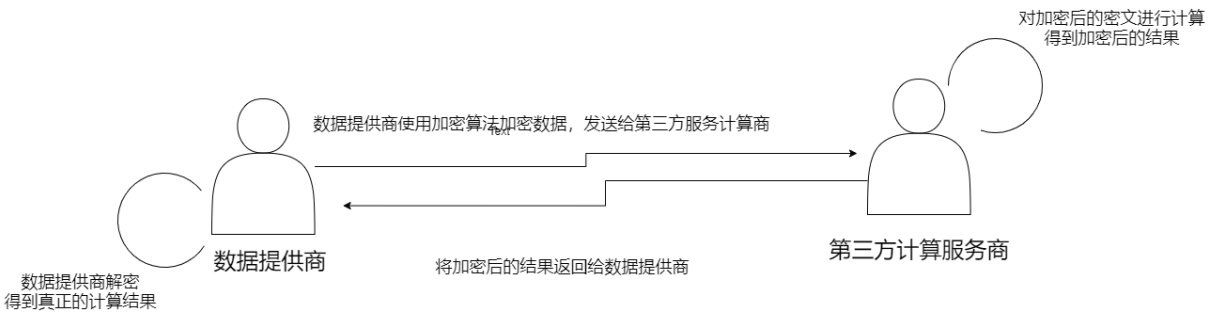

图 3-1 加密解密操作流程图

加密后进行加法运算得到的结果解密后与明文运算的一致，称为加法同态加密.同理满足减法、乘法的称为减法同态加密，乘法同态加密，而能够对密文进行任意计算的称为全同态加密.

3.2 加密算法原理

BFV 加密是主流的基于容错学习问题的全同态加密方案之一，其主要思想是将明文放在密文的高位，通过整除的方式，去除低位的噪声.比如有一个 BFV 的密文ct，私钥s，则 BFV 的解密为$m+e=c\times s$，其中e是解密噪声，显然，当噪声足够小的时候，我们可以通过除以t取整获得明文m.

3.2.1 背景介绍

对于一个简单多项式

$$4x^2+2x+1 \tag{3-3}$$

每一个系数都是整数，并且需要$\bmod t$.假设$t=12$，这就像一个 12 月份的日历，9 加 6 得到 3.多项式的所有系数都是这样处理的.

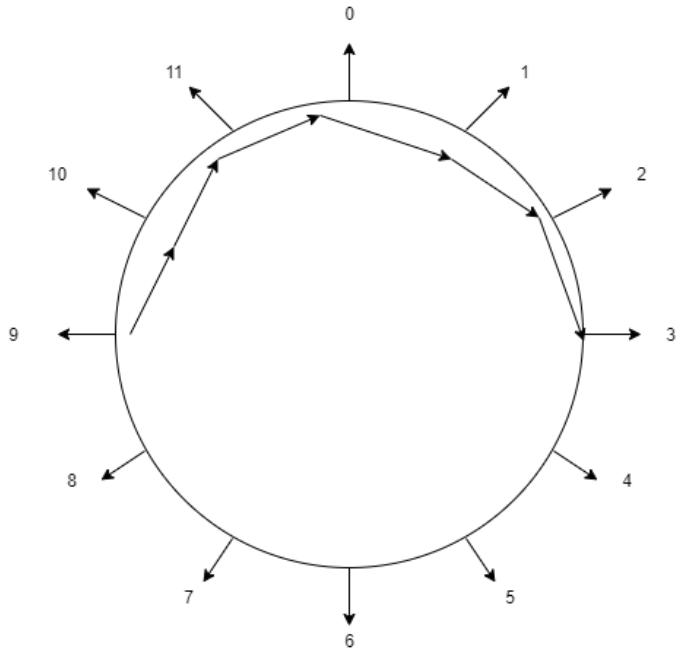

图 3-2 $\bmod t=12$ 的多项式环

或者我们可以将数字考虑在-5 到 6 之间，这样我们就可以方便地求负数.注意到，这只是一个方便系数——余数为-1 和余数为 11（除以 12 时）之间没有区别.

第二点，在于这种使用余数的思想不仅适用于多项式的系数，也适用于多项式本身.

根据 BFV 加密方案，定义了一个特殊的多项式，称为多项式模，并且只考虑多项式乘以该多项式模后的余数.BFV 方案中该多项式模的具体形式为 $x^{16}+1$，其中对于某些 n，有 $d=2n$.为了说明这一点，我们取 $n=4$，因此多项式为 $x^{16}+1$.

因为接下来考虑的是关于模 $x^{16}+1$ 之后的余数，所以我们只需要考虑幂从 x^0 到 x^{15} 的多项式.任何更高次的幂都会乘以该多项式模而消去.这也可以被理解为，$x^{16}=-1(\text{mod } x^{16}+1)$，这意味着 x^{16} 可以被 -1 替换，以将 x 的更高次幂归约到 0 到 16 的范围内.

所以接下来考虑的多项式都是这种形式的

$$\begin{aligned} &a_{15}x^{15}+a_{14}x^{14}+a_{13}x^{13}+a_{12}x^{12}+a_{11}x^{11}+a_{10}x^{10}+a_9x^9+a_8x^8 \\ &+a_7x^7+a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x+a_0 \end{aligned} \tag{3-4}$$

其中这 16 系数（即 a_i）中的每一个范围都是从 0 到 $t-1$ 我们可以用系数的环面来说明，如下所示

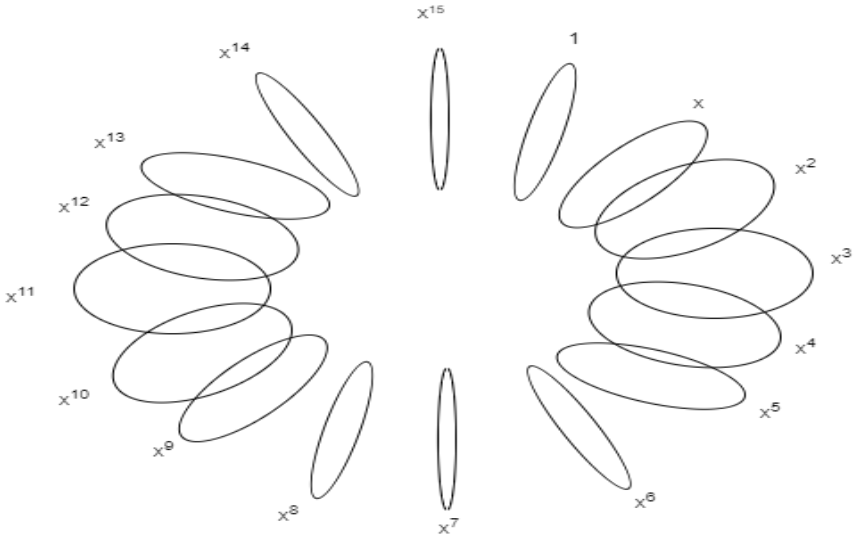

图 3-3 mod $d=16$ 系数环面

在这个图中，每个循环表示多项式中 x 的幂次方前的系数都是取值范围（包含 24 个可能值）.绿点代表系数取 0 时所处的位置.这为我们提供了一种很好的方法来可视化多项式，这在我们考虑加密和解密步骤如何工作时将会有所帮助.

BFV 加密方案涉及大量的多项式乘法.当我们把 x 的两个幂次方相乘，比如 $2x^{14}$ 和 x^4 时，我们把他们指数相加，得到 $2x^{18}$.有人可能会假设，求这个多项式关于多项式模的余数可能需要在 x^{16} 处将指数转回 0，得到 $2x^2$ ，就像上面所示的整数系数那样.如果多项式模式 x^{16} ，情况就是这样.然而，我们的多项式模 $x^{16}+1$ ——如上所示，额外的+1 因子引入了一个符号变化这有助于进一步干扰乘法的结果.

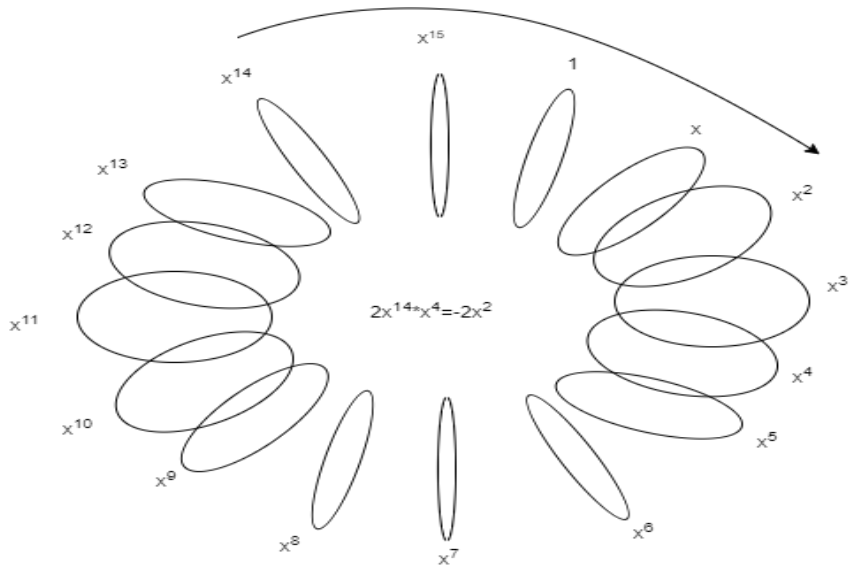

图 3-4 环面变化

如上图所示，当 $2x^{14}$ 乘以 x^4 后模 $x^{16}+1$ 时，取这个项，向前旋转环面 4 个幂，然后

从 0 处调整系数的值，得到$22x^2$（或$-2x^2$，如果我们认为数字时从-12 到 11 而不是从 0 到 23 时）.

这种形式的多项式具有非常丰富的结构和许多不错的特性.他们时多元多项式的子集.使用其中一个作为多项式模并不是严格必须的，但是这样做会更加方便快速.

3.2.2 使用环上的多项式加密

我们已经介绍了 FV 加密方案中使用的换上的多项式的一些属性，现在我们可以讨论加密和解密的工作原理.首先，我们需要讨论如何生成私钥和公钥，然后讨论如何使用他们进行加密和解密.

1)私钥和公钥

加密采用明文，并使用从私钥派生的公钥将其转换为密文.从明文到密文的转换是通过一种只有在自己知道私钥的情况下才容易可逆的方式完成的.

更具体地说，明文是环上的多项式，其具有多项式模x^d+1，其中$d=2n$，以及系数模t.明文加密后为密文，其是由两个环上的、具有相同多项式模的多项式构成的，但系数模为q，通 常 q 远大于t.

例如，多项式模为$x^{4096}+1$，这意味着明文和密文中的多项式都有$d=4096$个系数.明文多项式的系数需要模$t=290764801$，密文多项式的系数需要模 q = 92143472475614740 48 或更大.

在第一部分中，为了更直观，我们将使用$d=20$、$t=20480$和$q=26214400$.

对于私钥或密钥，我们用s表示，它是我们随机生成的一个系数为-1、0 或 1 的多项式.例如，

$$s=-x^{18}-x^{17}+x^{16}+x^{14}+x^{13}-x^{12}-x^{11}-x^{10}+x^9-x^7+x^6-x^5-x^4+x^3-x-1 \qquad (3\text{-}5)$$

接下来，我们从密文空间中随机生成一个多项式(用于生成公钥)，其系数模为q，我们用a表示.

$$\begin{aligned} a=&6232991x^{19}+4468759x^{18}+8071583x^{17}-9205735x^{16}+2480145x^{15}\\ &+7880745x^{14}+664117x^{13}-10042664x^{12}+700992x^{11}+6068675x^{10}\\ &+1080889x^9+4403536x^8-11676926x^7-5038114x^6-6847784x^5\\ &+389783x^4+8303932x^3+7824347x^2-10936607x+8220445 \end{aligned} \qquad (3\text{-}6)$$

我们还定义了一个噪音多项式，它是“小”的，因为它是从离散高斯分布中取的一个小系 数.这个多项式只在这里使用一次，然后丢弃.

$$e = 5x^{18} + x^{17} + x^{16} + 4x^{15} + 3x^{14} - x^{13} - 4x^{12} - x^{11} - 3x^{10} \\ -4x^9 + 2x^8 + 3x^7 + 2x^6 + 3x^5 - 2x^2 + 2x - 3 \tag{3-7}$$

然后将公钥定义为一对多项式，即 $pk = ([-as + e]_q, a)$，其中多项式都是模多项式模和系数模 q 的.

对于上面给出的示例，公钥的第一个多项式被构造为

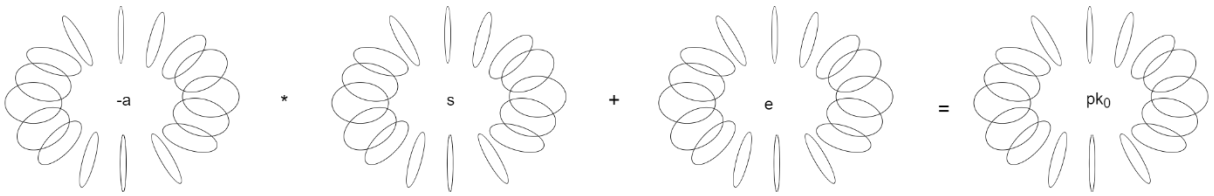

图 3-5 公钥生成

$$a = 7770124x^{19} + 11873819x^{18} - 10547374x^{17} - 4160027x^{16} + 7970426x^{15} \\ + 4383494x^{14} - 11926778x^{13} - 5351847x^{12} + 8805182x^{11} - 12993620x^{10} \\ + 6085770x^9 + 12168500x^8 - 2814833x^7 + 12795407x^6 - 10238010x^5 \\ - 2547658x^4 + 12185838x^3 - 10264067x^2 + 9206842x + 691008 \tag{3-8}$$

其中第一个乘法取多项式 a，它的系数 mod q，然后乘以系数为-1，0 或 1 的 s.由于模上多项式模的多项式的乘法具有“旋转和反射”性质，有效地混合和打乱了 a 的所有系数，并进一步增加了小的噪音.多项式 a 有效地掩盖了公钥中的私钥.

通过从公钥中找到 s 的方式来破解加密方案，其主要涉及的计算为 $([-as + e]_q, a)$.唯一的因素是该方案中包含了噪音——如果 e 为零，则很容易从公钥中计算出 s.当 e 足够大，但又不太大时，这是一个难题.

以上的示例中，私钥可以通过暴力攻击恢复——–尝试每个可能的 s (只有 $3^{20} = 3486784401$ 组合)，然后计算 $-as + e$ 来寻找出一个接近公钥的第一项的答案.对于真正的参数 3^{4096} $(d = 4096)$，这种暴力攻击的方法是完全不可行的.

加密

加密过程看起来有点像公钥生成过程.

加密明文的过程是将一个系数模为 t 的多项式转换为一对系数模为 q 的多项式.本例中，我们 将加密一个非常简单的多项式(称为消息)—— $m = 4x^2 + 2x + 1$ 只有三个不为零的系数.

加密还需要三个小的多项式.两个噪音多项式来自于相同的离散高斯分布(即和公钥中的噪音 多项式的取法一样)，另一个多项式我们称之为 u，它的系数为-1、0 或 1，就

像私钥一样.

$$\begin{aligned} e_1 = &-3x^{19}+3x^{18}+2x^{16}-2x^{15}-3x^{14}+2x^{13}-3x^{12}-4x^{11}+5x^{10} \\ &-x^9-2x^8+x^7+4x^6-4x^5-x^4+3x^3-3x^2+2x-3 \end{aligned} \tag{3-9}$$

$$\begin{aligned} e_2 = &-x^{19}+3x^{18}+x^{17}+3x^{16}-x^{15}-4x^{14}+x^{12}+x^{11}-x^{10} \\ &-3x^9+2x^8-3x^7-3x^6+4x^5-3x^4+x^3+4x^2+3x-2 \end{aligned} \tag{3-10}$$

和

$$\begin{aligned} u = &x^{19}+x^{18}+x^{17}-x^{16}-x^{14}+x^{13}-x^{12}-x^{10} \\ &+x^9+x^8-x^7+x^6-x^5+x^4-x^3+x^2+x+1 \end{aligned} \tag{3-11}$$

这些多项式只在加密过程中使用，然后丢弃.

密文是由两个多项式组成的，通过如下计算得到

$$ct = ([pk_0u + e_1 + \frac{qm}{t}]_q, [pk_1u + e_2]_q) \tag{3-12}$$

请注意消息中的值是在 mod t 的范围内，而在我们的示例中，它们被缩放为 q/t (即 1280)，使它们覆盖 mod q 的范围.这是消息被插入到密文时的唯一更改.这些值通过添加到第一项来掩盖，第一项的值是在 mod q 的范围内，与随机的噪音没有区别.u 的随机性改变了每次加密中使用的掩码，从而确保相同的明文在每次加密时产生不同的密文.

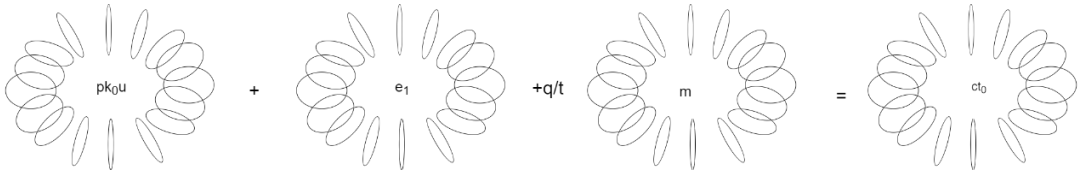

图 3-6 通过公钥加密明文为密文

同态加法和乘法之所以有效，是因为消息在密文中以比例来表示.其他项用于掩盖消息，而 且可以证明它们是有效的，只有在您知道私钥的情况下才能删除它们.

使用上面给出的多项式显式地计算密文的第一个元素

$$\begin{aligned} ct_0 = &-6704816x^{19}-3387669x^{18}-1263755x^{17}-7566503x^{16}-5553801x^{15} \\ &+10480374x^{14}-2983465x^{13}+8685054x^{12}-8427950x^{11}-12902530x^{10} \\ &+11941655x^9-5799749x^8-7949675x^7+12500833x^6-1304873x^5 \\ &-5505880x^4-10725454x^3+12010517x^2+4880686x-3589928 \end{aligned} \tag{3-13}$$

代入公钥，我们可以看到密文的第一个元素展开为

$$ct_0 = (e_1 + eu - aus + \frac{qm}{t})_q \tag{3-14}$$

在这个 表达式中，前两项是“小”的，与噪音成比例，后两项是“大”的.第一个大项有效地掩盖 了第二个大项，即消息.

密文的第二个元素是这样计算的:

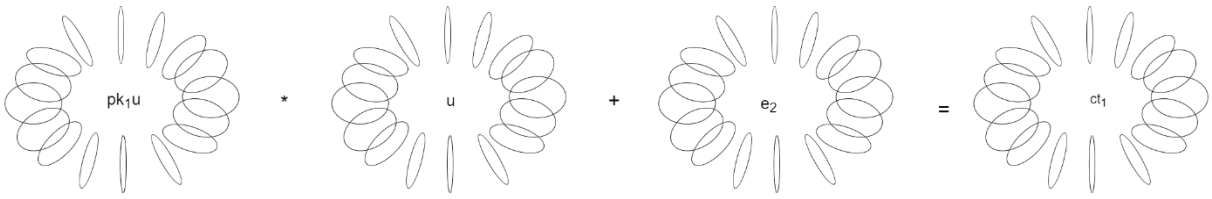

图 3-7 密文多项式的第二项

$$\begin{aligned} ct_1 = &-3163925x^{19} + 7314912x^{18} - 2859765x^{17} - 8189681x^{16} - 10348910x^{15} \\ &+ 8949536x^{14} - 7163151x^{13} + 10404596x^{12} + 4234890x^{11} + 9542582x^{10} \\ &- 6915366x^9 - 9859055x^8 - 8420035x^7 + 1103403x^6 + 1514767x^5 \\ &+ 4765440x^4 - 10020300x^3 + 10549635x^2 + 7785811x + 5495934 \end{aligned} \tag{3-15}$$

代入公钥，我们看到密文的第二个元素展开为$ct_1 = [au + e_2]_q$.这说明了解密是如何工作的 ——如果我们知道s，就可以计算出$ct_1 s = [aus + e_2 s]_q$，它可以用来消除密文的第一个元素中的非消息大项.

综上所述，密文可以用公钥(public key)、私钥(private key)、掩码(mask)、噪音(noise) 和消息(message)表示为

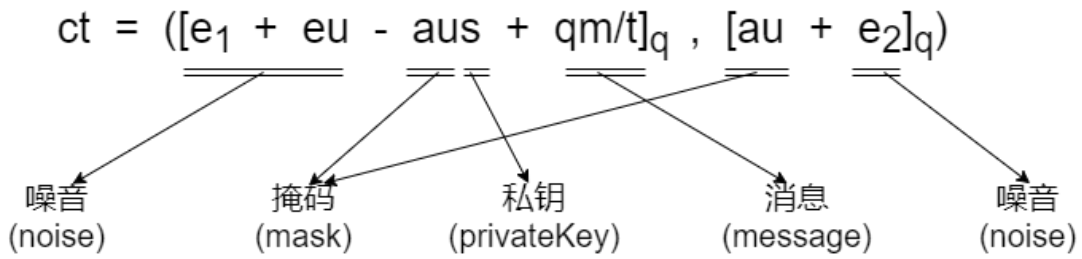

图 3-8 密文的组成

如上所述，解密相对简单.首先，我们计算$[ct_0 + ct_1 s]_q$，它将从消息中完全移除掩码.这给我们一个多项式，它可以展开为$[qm/t + e_1 + e_2 s]_q$,也就是说缩放后的信息加上一些噪声.因此，只要噪声不太大，我们就可以恢复消息.

明确地，

$$ct_1 s + ct_0 = 5x^{19} + 13x^{18} - 8x^{17} + 10x^{16} + 22x^{14} - 18x^{13} - 32x^{12} - 5x^{11} + 8x^{10} + 11x^9 + 33x^8 + 12x^7 - 12x^6 - 7x^5 + 5097x^4 + 4x^3 + 2558x^2 + 2x + 1292 \quad (3\text{-}16)$$

在这里您可以看到，除了明文的三个非零系数(x^4, x^2 和 x^0)之外，所有的系数都小于 $q/t = 1280$.如果我们把这个多项式缩放回 mod t 范围内的值，那么我们就得到

$$\frac{5x^{19}}{1280} + \frac{13x^{18}}{1280} - \frac{8x^{17}}{1280} + \frac{10x^{16}}{1280} + \frac{22x^{14}}{1280} - \frac{18x^{13}}{1280} - \frac{32x^{12}}{1280} - \frac{5x^{11}}{1280} + \frac{8x^{10}}{1280} + \frac{11x^9}{1280} + \frac{33x^8}{1280} + \frac{12x^7}{1280} - \frac{12x^6}{1280} - \frac{7x^5}{1280} + \frac{5097x^4}{1280} + \frac{4x^3}{1280} + \frac{2558x^2}{1280} + \frac{2x}{1280} + \frac{1292}{1280} \quad (3\text{-}17)$$

四舍五入这些系数可以恢复我们的消息 $m = 4x^4 + 2x^2 + 1$. 我们通过将系数四舍五入，来舍入到接近的整数后得到我们的信息.

把它们放在一起，我们通过如下计算来解密密文

$$m' = [[\frac{t}{q}[ct_0 + ct_1 s]_q]] \quad (3\text{-}18)$$

[]表示舍入到接近的整数(四舍五入).

如果系数中噪音太大，那么它们终会更接近一个与正确整数不同的整数，然后解密会(悄无声息地)失败并产生错误的结果.在上面的示例中，大的噪音为 $33/1280$ ，所以仍然有一些 空间允许产生更多的噪音，并且能够正确解密.噪音的含量可以通过将 q/t 的比值变大或变小来调节.

3.3 典型的统计计算公式变换与同态计算

3.3.1 统计计算公式变换

(1)算术平均：假设样本空间为 $\{x_1, \ldots, x_n\}$,算数平均可以表示为

$$\bar{x} = \sum_{i=1}^{n} x_i \quad (3\text{-}19)$$

(2)方差：方差表征变量 X 取值得散度，其变换后得计算公式如下

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - (\frac{1}{n}\sum_{i=1}^{n} x_i)^2 \quad (3\text{-}20)$$

(3)线性回归：设有数据集： $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$,一元线性回归的目的是找到线性方程 $y = a + bx$ 去拟合这个数据集，最常采用最小二乘法来确定参数 a 与 b ，等价变换后公式如下

$$a = \frac{1}{n}\sum_{i=1}^{n} y_i - b\frac{1}{n}\sum_{i=1}^{n} x_i \quad (3\text{-}21)$$

$$b=\frac{n\sum_{i=1}^{n}x_iy_i-\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i}{n\sum_{i=1}^{n}x_i^2-(\sum_{i=1}^{n}x_i)^2} \tag{3-22}$$

(4)相关系数：变量 X 与 Y 的相关系数是用来衡量两者之间线性关系的强度与方向，其变换后的计算公式如下

$$b=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{3-23}$$

$$=\frac{n\sum_{i=1}^{n}x_iy_i-\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i}{\sqrt{n\sum_{i=1}^{n}x_i^2-(\sum_{i=1}^{n}x_i)^2}\sqrt{n\sum_{i=1}^{n}y_i^2-(\sum_{i=1}^{n}y_i)^2}} \tag{3-24}$$

3.3.2 加法同态性与统计计算

令中间结果： $w_x=\sum_{i=1}^{n}x_i^2$ ， $u_x=\sum_{i=1}^{n}x_i$ ， $w_y=\sum_{i=1}^{n}y_i^2$ ， $u_y=\sum_{i=1}^{n}y_i$ ， $z_{xy}=\sum_{i=1}^{n}x_iy_i$.用密钥 b_{key} 分别加密 x_i ， x_i^2 ， y_i ， y_i^2 ， x_iy_i .根据全同态加密的加法同态和乘法同态，有一下等式成立

$$Enc_\zeta(w_x)=\prod_{i=1}^{n}Enc_\zeta(x_i^2),Enc_\zeta(u_x)=\prod_{i=1}^{n}Enc_\zeta(x_i) \tag{3-25}$$

$$Enc_\zeta(w_y)=\prod_{i=1}^{n}Enc_\zeta(y_i^2),Enc_\zeta(u_y)=\prod_{i=1}^{n}Enc_\zeta(y_i) \tag{3-26}$$

$$Enc_\zeta(z_{xy})=\prod_{i=1}^{n}Enc_\zeta(x_iy_i),Enc_\zeta(n)=\prod_{i=1}^{n}Enc_\zeta(1) \tag{3-27}$$

3.4 半诚实敌手模型

半诚实敌手模型应用于多方安全计算中，而对于安全多方计算拥有两种安全模型，一是半诚实敌手模型(The Semi-Honest Model)，二是恶意敌手模型(The Malicious Model).

对于半诚实敌手模型而言，计算参与方都是诚实的，都在按照协议的规定的流程执行.但是可能会被恶意攻击者监听获取到在协议执行过程中的自己的输入输出信息以及在协议运行过程中获得的信息.满足半诚实敌手模型是安全多方计算的基础条件，而恶意敌手模型是建立在主动攻击的基础上的，在现实生活中，主动攻击相对来说更复

杂和困难，而被动攻击却时很常见的，因此根据成本和开销，很多的协议都是基于保证半诚实敌手模型下的安全而建立.

4 实验概述

通过 matlab 软件编写私钥、随机噪声多项式、公钥、加密、解密函数代码，接着通过 matlab 的 deploy 命令讲编写的函数打成 jar 包，最后在 Java 语言平台上运用 apache 开源的 mina 框架中的 ServerSocket 与 ClientSocket 模拟分布式情况，观察分析其同态操作效果.

4.1 实验环境

4.1.1 环境要求和配置

硬件配置：Dell 笔记本电脑；Windows 10 家庭中文版；Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40GHz.

代码语言：matlab，Java.

编译工具：matlab2017R，IntelliJ IDEA 2021.2.

4.1.2 实验数据参数

模多项式次数 $d=16$，模系数 $q=896$ 以及系数模 $t=7$

4.1.3 分析依据

通过数百次的模拟，统计计算出其在分布式情况下的加密后的密文在多次同态操作后的解密成功率，之后分析其正确性，安全性以及局限性.

5 算法的分析

5.1 同态加法

在分布式系统下，将一个明文输入数据加密后得到一个多项式的密文.可以根据计算参与方的数量，决定 BFV 加密算法的多项式模 x^d+1 中 d 参数的选择.例如，拥有 16 个计算参与方，此时取 $d=16$，则生成的密文产生了两个最高次次数不超过 16 的多项式，将两个多项式写成一个 2 行 16 列的向量，例如密文 ct

$$\begin{aligned} ct_0 = {} & 217x^{15}-53x^{14}+13x^{13}-249x^{12}-392x^{11}-238x^{10}+252x^9+115x^8 \\ & +5x^7+184x^6-201x^5-258x^4-247x^3+144x^2+23x+42 \end{aligned} \tag{5-1}$$

$$\begin{aligned} ct_1 = {} & 42x^{15} - 256x^{14} - 393x^{13} - 229x^{12} + 447x^{11} - 369x^{10} - 212x^9 + 107x^8 \\ & + 52x^7 + 70x^6 - 138x^5 + 322x^4 + 186x^3 - 282x^2 - 60x + 84 \end{aligned} \tag{5-2}$$

将其转换为矩阵形式

$$CT = \begin{bmatrix} ct_0 \\ ct_1 \end{bmatrix} \tag{5-3}$$

$$\begin{bmatrix} 217 & -53 & 13 & -249 & -392 & -238 & 252 & 115 & 5 & 184 & -201 & -258 & -247 & 144 & 23 & 42 \\ 42 & -256 & -393 & -229 & 447 & -369 & -212 & 107 & 52 & 70 & -138 & 322 & 186 & -282 & -60 & 84 \end{bmatrix}$$

将得到的矩阵对同一列的进行拆分，保留行数，结果得到

$$\left(\begin{pmatrix} 217 \\ 42 \end{pmatrix}, \begin{pmatrix} -53 \\ -256 \end{pmatrix}, \begin{pmatrix} -249 \\ -229 \end{pmatrix}, \cdots, \begin{pmatrix} 23 \\ -60 \end{pmatrix}, \begin{pmatrix} 42 \\ 84 \end{pmatrix} \right) \tag{5-4}$$

的密文对.将每个密文对分发给每个计算参与方，参与方拿到密文对后，对其进行同态加法操作.

由于这是单个数据加密得到的密文，若有 n 个数据进行加密，那么就能得到 n 个密文 ct，那么就会有 n 个 2 行 $d=16$ 列的矩阵，也将其对应的拆分，发送给参与方.那么此时的计算参与方拥有 n 个 2 行 1 列的向量

$$\left(\begin{pmatrix} a^1_{ct_0^1} \\ a^1_{ct_1^1} \end{pmatrix}, \begin{pmatrix} a^2_{ct_0^2} \\ a^2_{ct_1^2} \end{pmatrix}, \cdots, \begin{pmatrix} a^{n-1}_{ct_0^{n-1}} \\ a^{n-1}_{ct_1^{n-1}} \end{pmatrix}, \begin{pmatrix} a^n_{ct_0^n} \\ a^n_{ct_1^n} \end{pmatrix} \right) \tag{5-5}$$

当计算参与方对每一个进行相加，得到一个 2 行 1 列的列向量

$$ct = \begin{bmatrix} \sum_{i=1}^{n} a^i_{ct_0^i} \\ \sum_{i=1}^{n} a^i_{ct_1^i} \end{bmatrix} \tag{5-6}$$

将此向量 ct 返回给数据所有者，即可完成同态加法的操作.

5.2 正确性

对于同态加法.数据所有者加密得到的多项式密文可以看作一个 $2\times d$ 维的矩阵 CT，对多项式密文的加法操作，等同于对矩阵的加法操作，再将其细分，观察其内部，可以发现：

1)计算参与方 P_i 拿到的 n 个 2×1 维向量数据对其进行加法操作，在矩阵计算中，属于不同矩阵中同一位置元素进行相加得到结果.所以对于计算参与方通过加法操作密文数据，得到的结果是正确的.

2)对于将密文多项式看作 $2\times d$ 维的矩阵 CT .多项式的加法中，幂次相等的项相加与矩阵的同一位置元素相加的操作是一致的.那么此过程是保证了同态加法的正确性.

而 BFV 加密算法方案是拥有同态乘法的全同态加密，对于同态乘法包括两个步骤：第一步很简单，基本上是将多项式 ct_1 和 ct_2 相乘，然后按 t/q 缩放.但问题是我们最终得到的密文是由 3 个环元素组成的，而不是 2 个.且由于多项式乘法的性质，一个计算参与方拥有的数据只是这一幂级的系数，无法拥有另一个幂级的系数，这就需要计算参与方之间的通信.若要完成乘法操作，那么获得其他幂级的系数就相当于获取了整个密文的所有系数，那么对于分布式系统来说，这就南辕北辙了.

5.3 安全性

为所有参与方都是半诚实的，且使用的加密体制是安全的，计算参与方获取到的为加密数据的一部分，此时计算参与方是无法推算出原始数据，保证其安全性.

5.4 局限性

对 BFV 加密算法而言，其本质是通过对多项式取模，对系数取模，接着添加噪声多项式来掩盖真实的消息，最后对密文多项式缩放，四舍五入取整确定系数.那么，由于添加噪声多项式的关系，且加法具有加法交换律，那么密文

$$\begin{aligned} a &= ([pk_0u_1 + e_1 + qm_1/t]_q, [pk_1u_1 + e_2]_q), \\ b &= ([pk_0u_2 + e_3 + qm_2/t]_q, [pk_1u_2 + e_4]_q) \end{aligned} \tag{5-7}$$

相加，得到新密文

$$\begin{aligned} &a+b \\ &= ([pk_0(u_1+u_2) + (e_1+e_3) + q(m_1+m_2)/t]_q, [pk_1(u_1+u_2) + (e_2+e_4)]_q) \end{aligned} \tag{5-8}$$

而根据解密算法，解密的多项式为

$$[\frac{q(m_1+m_2)}{t} + (e_1+e_3) + e(u_1+u_2) + (e_2+e_4)s]_q \tag{5-9}$$

随着同态加法操作的不断增加，噪声多项式的系数不断累积，当对解密多项式乘以 t/q 消去消息 (m_1+m_2) 的系数，噪音多项式的系数可能会出现有一个介于 $k[q/t, q/(2t)]$ 的值，那么对系数四舍五入取整时，会将噪音多项式 e 的系数给加上，那么解密就会解出错误的结果.

针对此问题，将噪音分为三类

$$\begin{aligned} e_5 &= e_1 + e_3 \\ eu_3 &= e(u_1 + u_2) \\ e_6 s &= (e_2 + e_4)s \end{aligned} \tag{5-10}$$

为了直观感受与分析，分别取 $q = 896, t = 7, d = 16$ 对这三种噪音多项式类型分别添加了 1，5 和 30 个噪音多项式，观察其系数分布

1) $e_5 = e_1 + e_3$ 类型噪音多项式

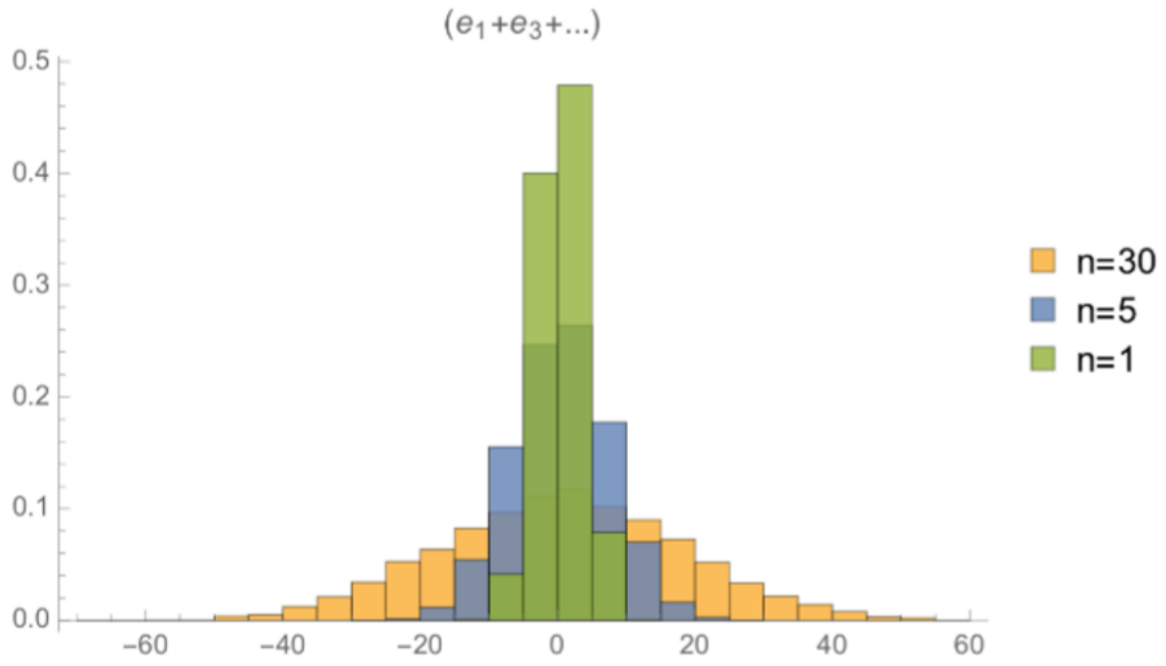

图 5-1 $e_5 = e_1 + e_3$ 类型各噪声多项式系数分布

当我们添加 30 个噪音多项式时，某些系数有可能会大于 64，即超过了 $q/t = 128$ 的一半，所以解密不会产生正确的结果.

另外两项表示不同的情况——第二项是一个噪音多项式乘以一些“小的多项式”(系数为-1、 0 或 1)的总和.这种乘法会产生更大的噪音.这意味着这个噪音与多项式的最高次的平方根 $\sqrt{n}$ 一致.

对这一项绘制与上面相同的分布可以看出，它比第一项大得多，而且即使对于我们示例中的参数，也存在错误解密的危险，即使只是添加了几个参数.

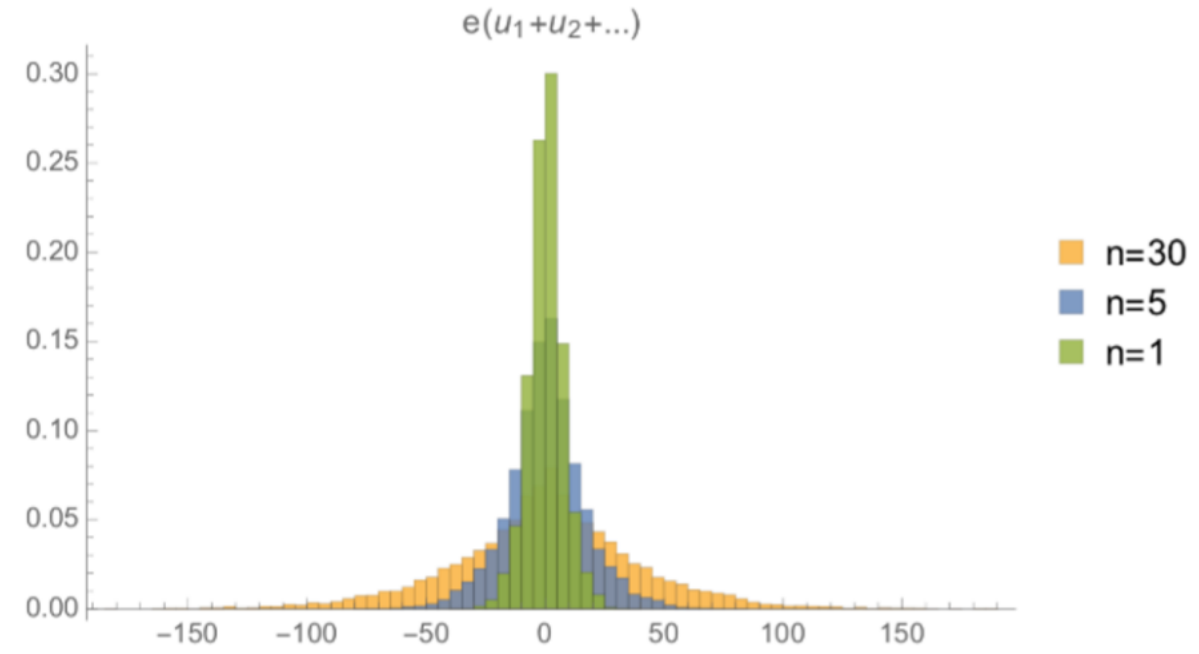

图 5-2 $eu_3 = e(u_1 + u_2)$ 类型各噪声多项式系数分布

第三项是类似的——一组噪音多项式之和，乘以一个“小的多项式”.它的噪音分布是这样的:

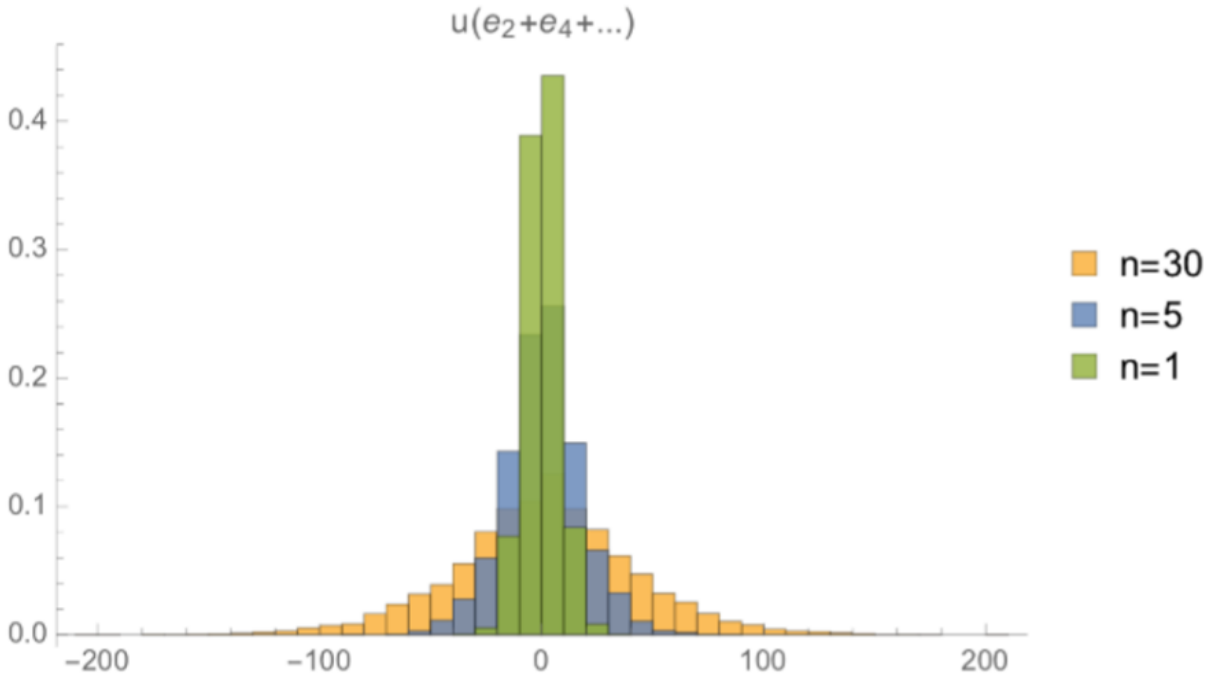

图 5-3 $e_6 s = (e_2 + e_4)s$ 类型各噪声多项式系数分布

这表明，对于我们所选择的参数，由两个以上加法产生的密文，解码错误的概率很高，而且两次加法失败的概率也很高.这是因为有时最大错误大于 64，当 q/t = 128 时，

会导致不正确的解密，就像这里一样.为了给这样的操作提供更多的空间，我们需要使用更大的 q/t 比值， 这可以应对通常由所执行的操作数量引入的噪音量.

6 结束语

本文在半诚实模型下，关于数据在分布式计算下的隐私性对全同态加密的 BFV 方案进行分析，对方案进行了正确性，安全性与局限性的分析.能支持对数据的加密与解密， 保证数据的隐私不被泄露.

通过仿真模拟发现，当同态操作达到一定次数时，由于噪声多项式的累加影响，会对最后的解密成功率产生影响，使得解密失败，这也就导致了 BFV 方案不能进行任意次数的同态加法操作.而又由于分布式特点，使得 BFV 算法本具有的同态乘法操作与分布式概念相冲突.同时，由于解密算法对 q/t 的系数进行四舍五入化简，使得同态计算只能进行整数类型的加密，运算和解密操作.这对于分布式统计来说是远远不够的.

所以对未来的展望是，加入 CKKS 算法，让其能拓展明文空间范围，能计算浮点数类型，最好是能进行任意此的同态操作.

谢 辞

经过了三个多月的学习和工作，我最终完成了本次的论文的论文.从开始接到论文题目到系统的实现，再到论文文章的完成，每走一步对我来说都是新的尝试与挑战，这也是我大学期间独立完成的一个大型项目.然我的论文作品不是很成熟，还有很多不足之处，但我能够自豪的说，那里面的每一段代码，都有我的劳动.当看着自我的程序，自我成天相伴的系统能够健康的运行，真是莫大的幸福和欣慰.我相信其中的酸甜苦辣最终都会化为甜美的甘泉.

这次做论文的经历也会使我终身受益，我感受到做论文是要真真正正用心去做的一件事情，是真正的自我学习的过程和研究的过程，没有学习就不可能有研究的潜力，没有自我的研究，就不会有所突破，那也就不叫论文了.期望这次的经历能让我在以后学习中激励我继续提高.

感激我们一齐在学校努力的同学，我们彼此关心、互相支持和帮忙，留下了许多难忘的回忆.

感激我的父母和家人，感激他们对我学习、生活给予的支持和照顾.在论文的写作过程中，还获得了许许多多人的帮忙与先前研究工作者的宝贵资料，论文的研究成果离不开你们的协作和帮忙，在此对你们表示深切的谢意.期望能够以本文向你们汇报，以感激你们对我的关怀与帮忙，感激一向以来对我的支持与鼓励.你们永远是我的精神支柱和继续前进的动力.

参考文献

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

[2] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[A]. Proc of the Advances in Cryptology(EUROCRYPT99)[C]. Prague, Czech Republic, 1999.233-238

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

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

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

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

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

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

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

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

[11] CORON J, MANDAL A, NACCACHE D, et al, Fully homomorphic encryption over the integers with shorter public keys[A]. Proc of the Advances in Cryptology (CRYPTO 201 1)[C]. Santa Barbara, California, USA, 2011.487-504.

[12]BRAKERSKI Z, VAIKUNTANATHAN V. Fully homomorphic encryption from ring-lwe and security for key dependent messages[A]. Proc of the Advances in Cryptology (CRYPTO 201 1)[C]. Santa Barbara, Califomia, USA, 2011.505 -524.

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

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

[15]GENTRY C, HALEVI S. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits[A]. Proc of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science[C]. Palm Springs, CA, USA, 201 1.107-109.

[16]黄汝维，桂小林，余思等.云环境中支持隐私保护的可计算加密方法[J].计算机学报, 2011, 34(12): 2391-2402.

[17]T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469-472, July 1985, doi: 10.1109/TIT.1985.1057074.

[18]Paillier P . Public-key cryptosystems based on composite degree residuosity classes[J]. Advances in Cryptology Leurocrypt, 2004.

[19]Goldwasser, S.; Micali, S.; Rivest, R. L. (1988). "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks". SIAM Journal on Computing. 17 (2): 281.

[20]D Boneh, EJ Goh, K Nissim (April 2006). "Evaluating 2-DNF Formulas on Ciphertexts"

[21]STOC '09: Proceedings of the forty-first annual ACM symposium on Theory of computingMay 2009 Pages 169–178https://doi.org/10.1145/1536414.1536440

[22]杨攀,桂小林,姚婧,林建财,田丰,张学军.支持同态算术运算的数据加密方案算法研究[J].通信学报,2015,36(01):171-182.