全同态加密算法深入解析-2

原创 致远博士 格密链 2018-11-30 10:00

同态加密方案提供了一种惊人的能力——能够在不解密的情况下，对密文数据进行计算。这使得您无需破坏敏感源数据，同时可以对数据进行处理。我们将分两次内容对全同态加密算法做深入地解析，此为第二部分。

前面已经介绍了一些关于全同态加密的数学知识和加解密是如何工作的，今天的内容包含全同态加密中最有意思的部分——同态操作(同态加法和乘法)以及Relinearisation等话题。

1 同态操作

人们对这类密码体制如此感兴趣的一个主要原因是，它们允许同态加法和乘法(来自希腊语*homo* - same和*morphe* - shape)。这意味着您可以在数字仍然加密的情况下进行加法和乘法运算，而不必先解密它们。这是一个令人惊叹的功能，有望在数据保护和安全方面构建一个新的黄金标准。

同态加法

最简单的情况是两个加密数字的加法。假设我们已经用相同的公钥加密了两个多项式m_1和m_2：

$$\mathbf{a} = ([\mathbf{pk}_0 u_1 + e_1 + qm_1/t]_q, [\mathbf{pk}_1 u_1 + e_2]_q),$$
$$\mathbf{b} = ([\mathbf{pk}_0 u_2 + e_3 + qm_2/t]_q, [\mathbf{pk}_1 u_2 + e_4]_q).$$

注意，我们需要使用两个不同的、小的多项式u_1和u_2，以及4个小的噪音多项式$e_1 \cdots e_4$。

如果我们仅仅是将密文中的元素相加，就会得到一个新的密文

$$\mathbf{a}+\mathbf{b}=\left([\mathbf{pk}_0(u_1+u_2)+(e_1+e_3)+q(m_1+m_2)/t]_q,[\mathbf{pk}_1(u_1+u_2)+(e_2+e_4)]_q\right)$$

由于消息仅存在于具有缩放的密文中，所以加法的结果与m_1 + m_2加密的形式相同，只是增加了新的噪音：

$$\mathbf{c}=\left([\mathbf{pk}_0(u_3)+(e_5)+q(m_1+m_2)/t]_q,[\mathbf{pk}_1(u_3)+(e_6)]_q\right)$$

近似解密(舍入之前)将是

$$[q(m_1+m_2)/t+e_5+eu_3+e_6s]_q$$

这意味着只要新的噪音不是太大，消息m_1 + m_2将正确解密。噪音有三种类型:

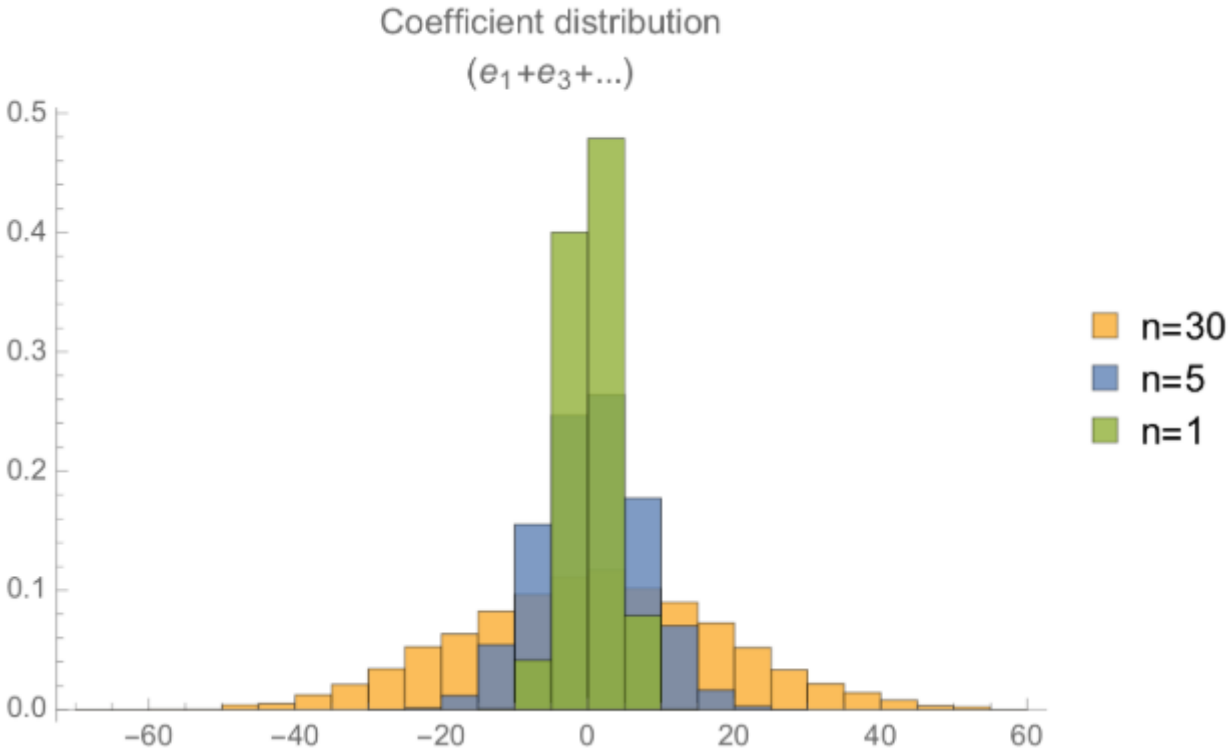
$$e_5=e_1+e_3$$

$$eu_3=e(u_1+u_2)$$

$$e_6s=(e_2+e_4)s$$

我们担心的是，当这些项变得足够大，以至于噪音多项式中的一个系数大于$q/(2t)$时，解密就会失败，因为解密过程结束时的四舍五入操作会四舍五入到错误的数字。

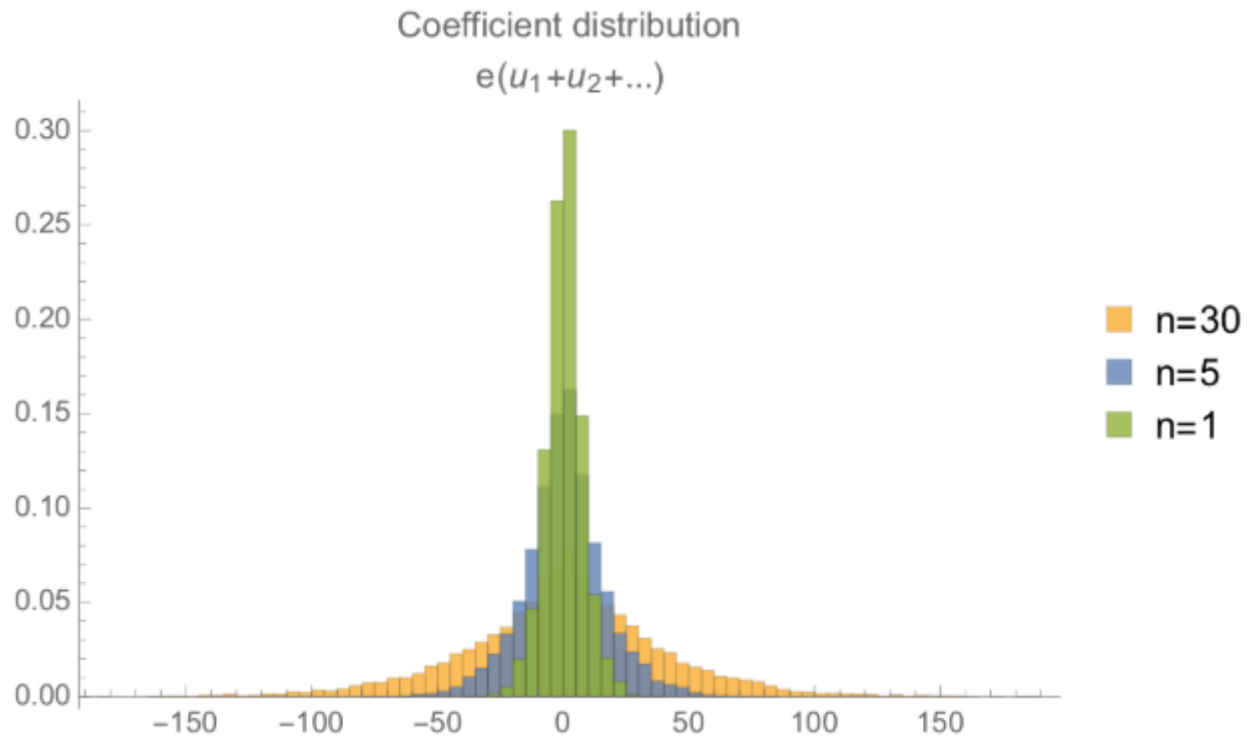
如果我们只考虑第一项，那么我们就把来自离散高斯分布的多项式中的系数相加。这意味着，在某些情况下，我们会把一个负系数加到一个正系数上，结果会更接近于零。在其他情况下，系数会有相同的符号，所以结果会更大。我们可以做很多的同态加法，看看噪音是如何随着加法的数量增加而增加的，这是很有指导意义的。系数的分布如下图所示，其中我们添加了1、5和30个噪音多项式(随机地进行了数百次试验)。



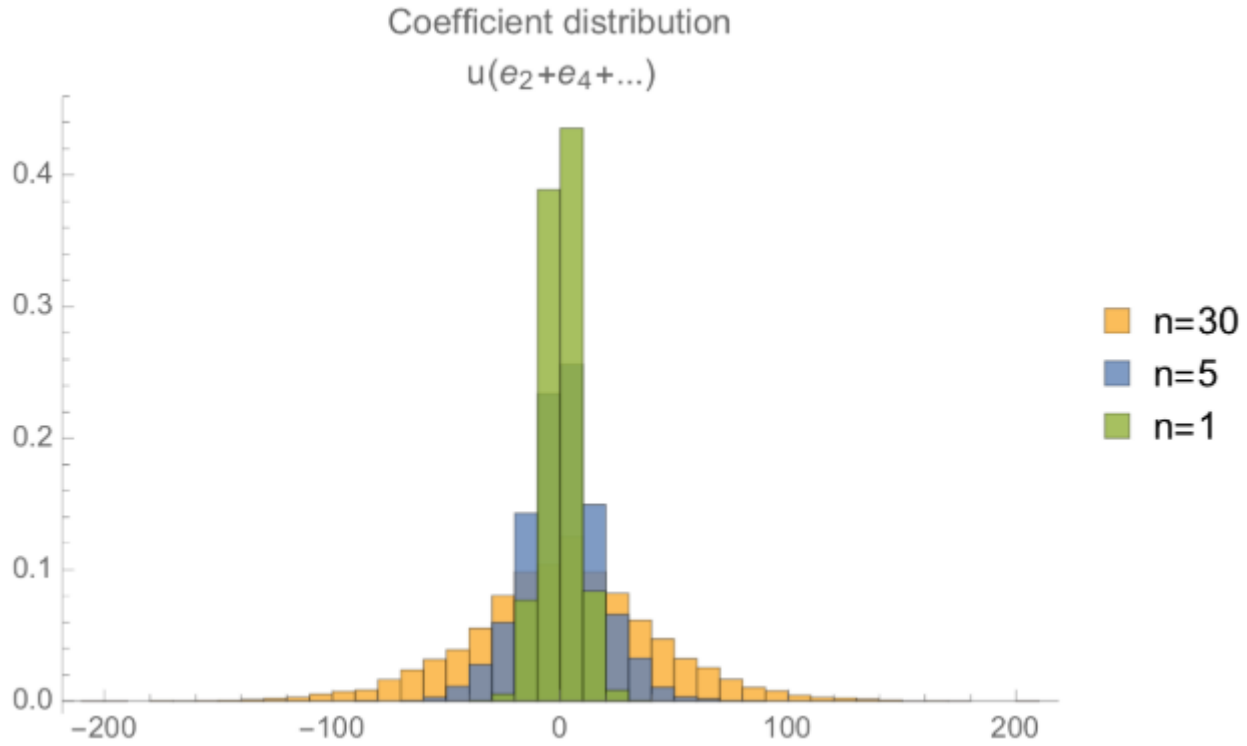
当我们添加了30个噪音多项式时，某些系数有可能会大于64，即超过了q/t的一半，所以解密不会产生正确的结果。

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。一个噪音多项式和一个小的多项式的乘积的系数大约将是随机正负号的噪音多项式系数的2/3rds的总和。这意味着这个噪音与多项式的最高次的平方根$\sqrt{n}$一致。

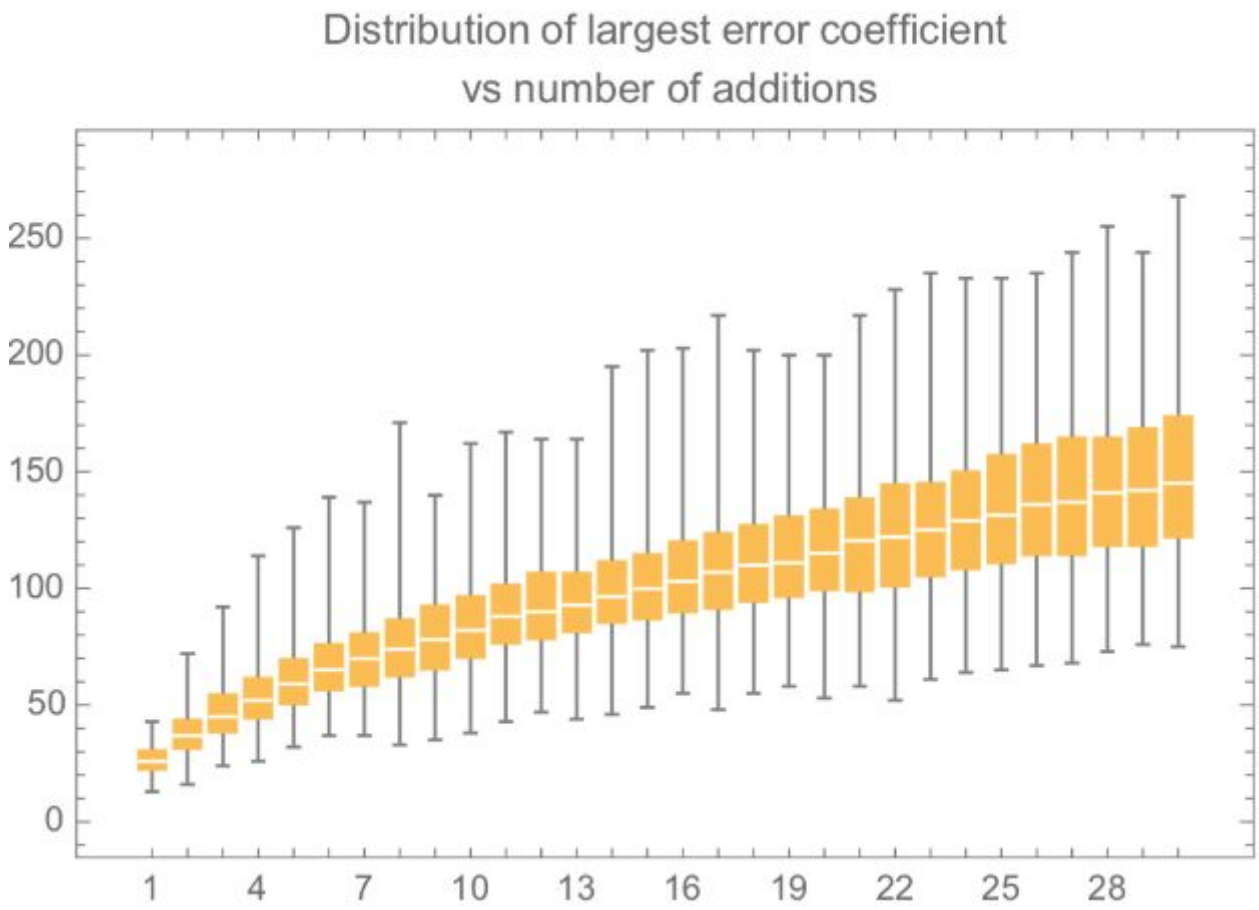
对这一项绘制与上面相同的分布可以看出，它比第一项大得多，而且即使对于我们示例中的参数，也存在错误解密的危险，即使只是添加了几个参数。



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:



结合起来，我们可以画出这三项的最大系数的增长，作为已经发生的加法数量的函数。这是一个须状图，给出了这些最大值的可变性。(注意噪音的均值接近于零，这是最大系数的幅值分布。)



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。

不幸的是，由密文的同态乘法引入的噪音量又要大得多。

同态乘法

同态乘法在程序上很简单，但是比加法复杂得多。如上所述，消息以qm_1/t的比例出现在密文的第一个元素中。因此，将两个密文的第一个元素相乘，再乘以t/q，就会得到一个带有$qm_1 m_2/t$的项——如果我们仍然能够除去掩码项，这个项就可以恢复。

因此，要理解同态乘法的机制，关键在于了解如何从密文的乘积中去掉掩码项。要做到这一点，我们的想法是把密文看作是私钥s的幂次方的一个简单多项式。这是这篇文章中使用多项式的第三种不同的方法，所以它有点令人困惑，但是它是理解同态乘法如何工作的关键。

我们可以写出解密过程的第一部分，使密文的每个元素都是s的多项式的系数:

$$\left[\mathbf{ct}_0 + \mathbf{ct}_1 s^1\right]_q$$

请记住，**ct**和*s*本身就是多项式，所以这个方程是一个多项式乘以一个多项式(s^0)加上一个多项式乘以另一个多项式，然后所有这些都取多项式模$x^d + 1$和系数模q。

现在，我们在上面看到解密产生了一个与掩码项*au*无关的量。

$$\left[\mathbf{ct}_0 + \mathbf{ct}_1 s^1\right]_q \rightarrow \frac{q}{t} m + noise$$

好了，现在考虑两个密文**a**和**b**，它们被定义为两个消息m_1和m_2的加密，它们可以被解密:

$$\left[\mathbf{a}_0 + \mathbf{a}_1 s^1\right]_q \rightarrow \frac{q}{t} m_1 + n_1$$

$$\left[\mathbf{b}_0 + \mathbf{b}_1 s^1\right]_q \rightarrow \frac{q}{t} m_2 + n_2$$

其中n_1和n_2表示密文中的噪声。

如果我们取它们的乘积，我们有:

$$\left[\mathbf{a}_0 + \mathbf{a}_1 s^1\right]_q \left[\mathbf{b}_0 + \mathbf{b}_1 s^1\right]_q \rightarrow \left(\frac{q}{t} m_1 + n_1\right)\left(\frac{q}{t} m_2 + n_2\right)$$

右边的表达式与计算**a**和**b**所用的掩码无关，所以左边也必须与它们无关。

如果我们把左边展开成*s*的形式(为了方便起见，再乘以*t*/*q*)就得到了

$$mult(\mathbf{a}, \mathbf{b}) = \mathbf{c}_0 + \mathbf{c}_1 s + \mathbf{c}_2 s^2$$

其中

$$\mathbf{c}_0 = \left[\frac{t}{q}\mathbf{a}_0\mathbf{b}_0\right]_q$$

$$\mathbf{c}_1 = \left[\frac{t}{q}(\mathbf{a}_1\mathbf{b}_0 + \mathbf{a}_0\mathbf{b}_1)\right]_q$$

$$\mathbf{c}_2 = \left[\frac{t}{q}\mathbf{a}_1\mathbf{b}_1\right]_q$$

这样做意味着我们可以计算出一个新的密文的组成部分，它比原来的密文多一个元素，并且可以正确地使用密钥*s*的幂次方进行解密。

解密的形式展开如下:

$$\left[\left\lfloor\frac{t}{q}\left[\mathbf{ct}_0 s^0 + \mathbf{ct}_1 s^1 + \mathbf{ct}_2 s^2\right]_q\right\rceil\right]_t$$

这只是增加了另一项即多项式乘以多项式的平方。有很多簿记要做，但它只是学校级代数(直到模数部分!)这是解密步骤的概括，它允许我们解密同态乘法的结果。

要了解这一切是如何显式地工作的，请考虑**a**和**b**在加密过程中的展开式

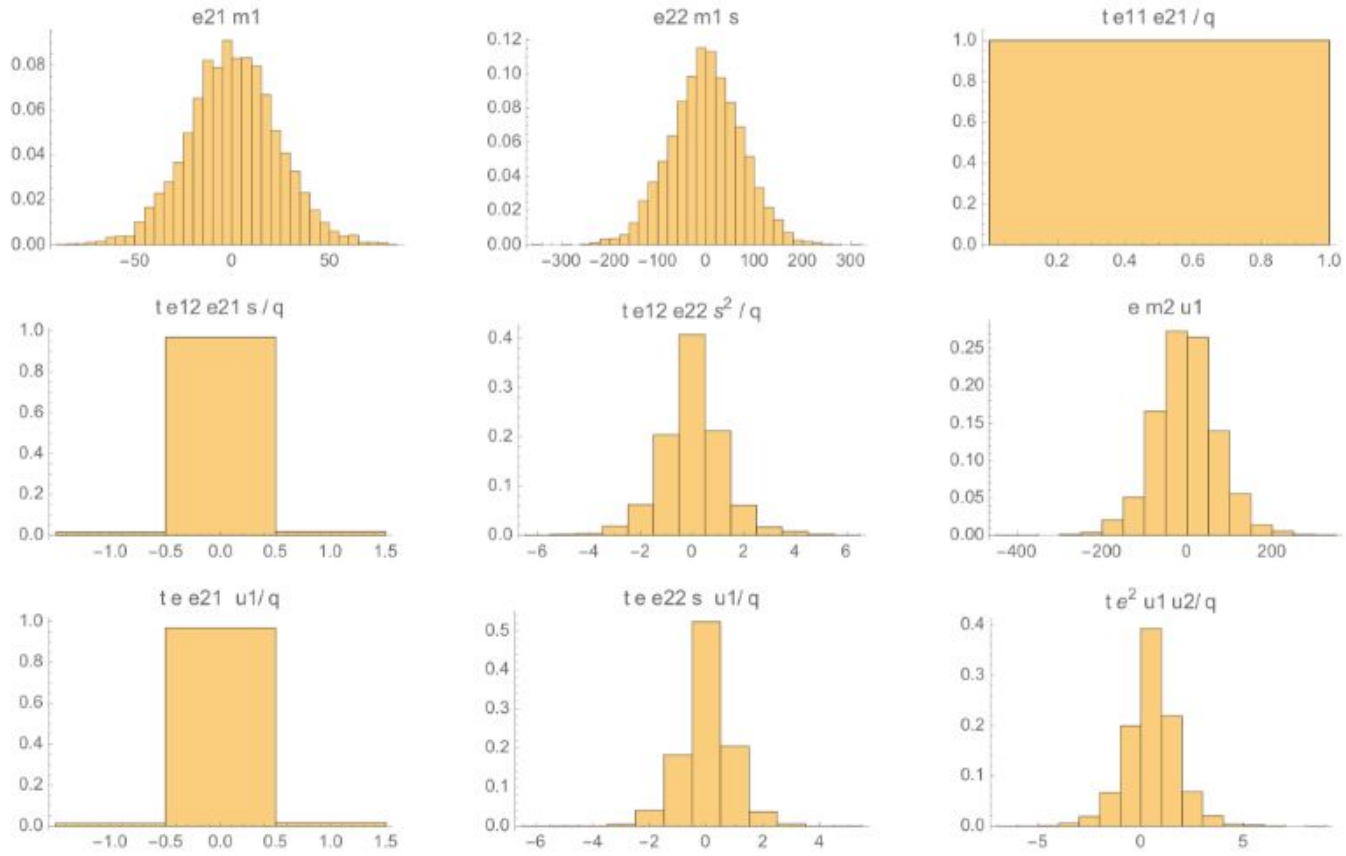
$$\mathbf{a} = [\mathbf{pk}_0 u_1 + e_{11} + qm_1/t, \mathbf{pk}_1 u_1 + e_{12}]$$
$$\mathbf{b} = [\mathbf{pk}_0 u_2 + e_{21} + qm_2/t, \mathbf{pk}_1 u_2 + e_{22}]$$

如果我们展开乘法的定义，同时对结果进行部分解密(即解密到除以*q*/*t*和整数之前)，那么得到的表达式就很复杂。但是，由于每个密文的组件都是在解密过程中被构造成能够删除掩码项(*au*ᵢ)的，所以这个展开的结果完全不依赖于来自公钥的掩码项!!!得到的表达式如下：

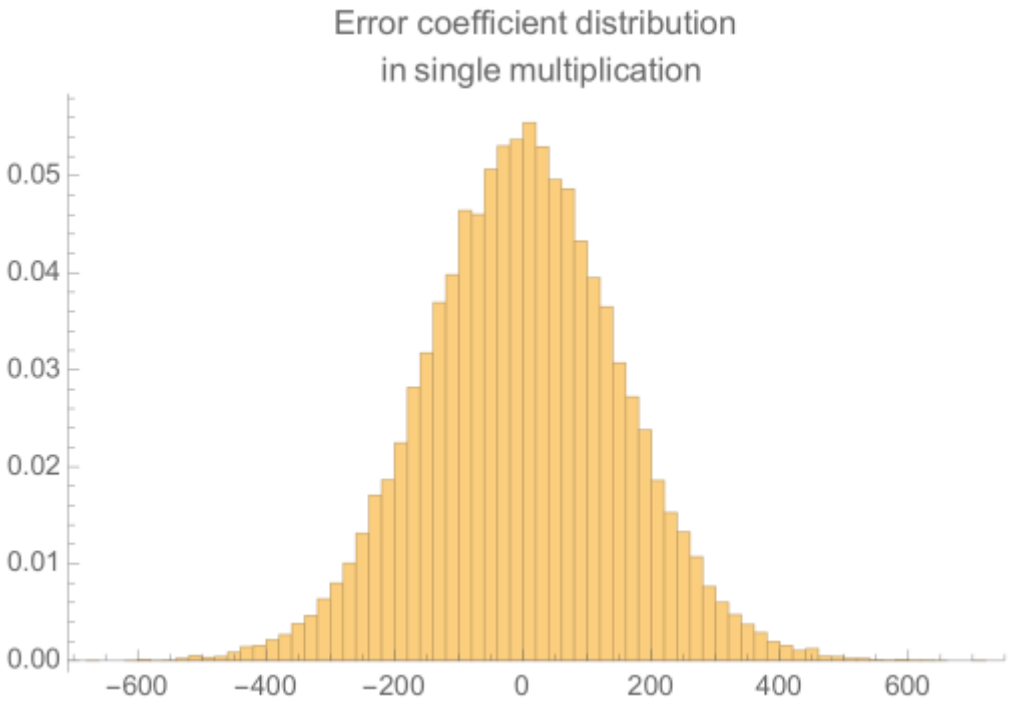
$$\mathbf{c}_0 s^0 + \mathbf{c}_1 s^1 + \mathbf{c}_2 s^2$$
$$= \frac{q}{t}m_1 m_2 + e_{22}m_1 s + e_{12}m_2 s + em_2 u_1 + em_1 u_2 + e_{21}m_1 + e_{11}m_2$$
$$+\frac{t}{q}e^2 u_1 u_2 + \frac{t}{q}e_{12}e_{22}s^2 + \frac{t}{q}e_{22}esu_1 + \frac{t}{q}e_{12}esu_2$$
$$+\frac{t}{q}e_{12}e_{21}s + \frac{t}{q}e_{11}e_{22}s + \frac{t}{q}e_{21}eu_1 + \frac{t}{q}e_{11}eu_2 + \frac{t}{q}e_{11}e_{21}$$

这里有很多项，但是现在我们已经去掉了掩码，问题是，噪音(除了第一个)与$q/(2t)$的“噪音预算”相比有多大?

为了感受这一点，我们模拟了大量加密的随机信息的乘法，$d = 16$，$t = 7$，$q = 7168 = 1024 \times t$。各类型噪音的系数大小分布如下图所示。请注意，总的噪音需要大于$q/(2t) = 512$才能导致解密错误。在这些项中，涉及噪音多项式的项、消息和私钥$e_{22}m_1 + \mathbf{e}_{12}m_2 s$是最大的贡献者。



上图显示，对于这些参数，最大的贡献来自于包含噪音多项式乘以消息多项式和私钥的项。这种噪音的最大系数约为300。这里有两项，其他的项更小。把所有的噪音合并成一个，就得到了乘法结果的总噪音。这些系数的分布如下图所示：



这表明没有足够的预算来安全地进行单次乘法，然后解密这些参数的正确结果(无论如何都是不安全的!)——大约1/4000的系数将具有大于512的噪音，导致约1%的解密错误率。

因此，如果我们将它们视为*s*的多项式，则可以进行密文的乘法，从而在解密时抵消它们自己的掩码项。将它们相乘，并分别跟踪*s*的幂次方的系数和噪音量，以便我们确信它们能够正确解密。

— 2 —
Relinearisation和其他话题

上面概述的乘法策略允许我们进行多次乘法，但代价是每次乘法都将密文的大小增加一个多项式。密文在大小上增长可能是一个问题。事实证明，有一些方法可以将密文的大小还原为两个多项式，但代价是增加噪音。这就是所谓的Relinearisation（再线性化），因为你要去掉*s*多项式中的二次项和更高的项。

另一项使这种加密方案切实可行的重要技术是将多个消息打包到一个明文中，通过并行化提高吞吐量。

— 3 —
结论

粗略地说，加密是将消息隐藏在一个环上的多项式中，并添加一些噪音。每个密文都包含足够的信息，可以在给定私钥的情况下除去自己的掩码。因为嵌入只涉及到消息的缩放，所以

仍然可以对它们执行加法和乘法，并使用一些巧妙的结构来在之后移除掩码。该方案的安全性来自于在不知道私钥的情况下，在噪声存在的情况下很难去除掩码。这个问题的难度导致了一些优秀的安全性能，例如没有已知的量子算法来攻击这些系统。

如果您已经了解了这些，我们希望您现在能够更好地理解基于Ring Learning with Errors问题的同态加密方案(或者至少是这些方案中的FV方案)的工作原理。

你可能还会喜欢：

黎曼猜想是否会对密码学的安全产生影响
比特币必须灭亡
50+区块链如何引领世界的例子
Token化如何将传统资产搬上区块链
Token的价值
区块链是糟糕的技术
谁将赢得区块链比赛 ——中国正在赢得500年来最重要比赛
全同态加密：从理论到实践-1
给六岁小孩讲区块链
我为什么受够了Chrome
传统资产正迈入Token化时代
解析比特币白皮书之交易
Coin和Token间的区别到底是什么?
区块链实力哪国强

欢迎收听“区块链杂谈”节目，国内最有质量的区块链知识分享节目。

宁波格密链网络科技有限公司，专注于区块链上的密码技术研发。

格密链

专注于区块链上的密码学技术

长按扫码可关注