同态加密的数学原理

内容：杨峥芃、陆皓喆　　PPT：郝志成　　展示：李晨阳

1 同态加密的原理

什么是同态加密？简单来说，同态加密就是一种委托数据处理的方法，但是让你不丧失对数据的所有权。

举一个很简单的例子：

$Alice$想让工人加工自己的金子但是不信任工人，害怕其在操作过程中偷取自己的金子，那么就相出了以下的办法：$Alice$将金子锁在一个密闭的盒子里面，这个盒子安装了一个手套。工人可以带着这个手套，对盒子内部的金子进行处理。但是盒子是锁着的，所以工人不仅拿不到金块，连处理过程中掉下的任何金子都拿不到。加工完成后。$Alice$拿回这个盒子，把锁打开，就得到了金子。

这其中抽象出的概念就是：

- $Alice$：数据持有方
- 工人： 不可信的服务提供第三方
- 盒子：加密算法
- 盒子上的锁：用户密钥
- 将金块放在盒子里面并且用锁锁上：将数据用同态加密方案进行加密
- 加工：应用同态特性，在无法取得数据的条件下直接对加密结果进行处理
- 开锁：对结果进行解密，直接得到处理后的结果

这样，我们就完成了我们信息的传输，而且保证了信息的安全性。

同态加密对于数据安全来说，不像一般的加密方案只关注数据存储安全，攻击者无法从密文中获得任何信息, 对加密数据的任何改动操作都会造成解密的错误。而同态加密关注于数据的处理安全，其提供了一种对加密数据处理的功能，且处理过程中无法得知原始内容，同时数据经过操作后还能够解密得到处理好的结果。

同态加密（$Homomorphic\ Encryption$）允许对密文处理后仍然是加密的结果。即对密文直接进行处理，跟对明文进行处理后再对处理结果加密，得到的结果相同。从抽象代数的角度讲，保持了同态性。其中的关键问题就是“等比例映射”，它让加密后的数据按比例代表原始数据，实现了加密数据的运算。在同态加密中，加密数据被表示为群或环上的元素，加密算法也属于这个群或环，因此可以对加密数据进行增量计算，同时保持密文的安全性。

总的来说，同态加密是一种高度安全的加密方式，可以实现加密数据的运算，同时保持数据的安全性。它通过等比例映射和解决离散对数问题，实现了诸如私有带宽调度、安全计算、秘密共享和加密搜索等复杂的安全计算。

2 同态加密的数学定义

我们给出一个公式：

$$E(m_1) \star E(m_2) = E(m_1 \star m_2)\ , \forall m_1, m_2 \in M$$

在定义中我们假设加密算法E应用于所有可能的信息集合M。如果加密算法E满足公式(1)，则称其在★运算上具有同态加密的特性。目前的同态加密算法主要支持两种运算上的同态性：加法和乘法。

需要强调的是，公式(1)仅用于更清晰地说明同态加密的性质，实际的同态加密算法可能略有不同。例如，$Paillier$算法对加法具有同态性，根据公式(1)，其密文的求和应等于密文的求和，但实际情况是密文的乘积等于求和后的密文。因此，我们通常只关注密文结果是否与预期的计算结果相同，而不会具体要求密文上的计算方式（通常由加密算法确定）。

3 同态加密的分类

一般包括四种类型：加法同态、乘法同态、减法同态和除法同态。

如果同时满足加法同态和乘法同态，则意味着是代数同态，称为**全同态**($Full\ Homomorphic$)；同时满足四种同态性，则称为**算数同态**。对于计算机来说，实现了全同态就可以实现所有操作的同态性。

下面列举了一些近年来的同态加密的发展过程：

类型	算法	时间	说明	实际应用
半同态加密-乘法同态	RSA算法	1977	非随机化加密，具有乘法同态性的原始算法面临一系列选择明文攻击	在非同态场景中使用非常广泛
半同态加密-乘法同态	$ElGamal$算法	1985	随机化加密	DSS数字签名标准基于$ElGamal$数字签名算法的变形
半同态加密-加法同态	$Paillier$算法	1999	应用最为成熟	联邦学习
半同态加密-有限次数全同态	$Boneh\text{-}Goh\text{-}Nissim$方案	2005	仅支持一次乘法同态运算	无
全同态加密	$Gentry$方案	2009	第一代全同态加密，性能较差	无
全同态加密	BGV方案	2012	第二代全同态加密，性能相对较好	$IBM\ HElib$开源库
全同态加密	BFV方案	2012	第二代全同态加密，和BGV类似	微软$SEAL$开源库
全同态加密	GSW方案	2013	第三代全同态加密，基于近似特征向量	$TFHE$开源库
全同态加密	$CKKS$方案	2017	可实现浮点数近似运算	$HElib$和$SEAL$

4 同态加密的常用算法

4.1 RSA

该算法是一种半同态加密算法的乘法运算。我们首先需要生成密钥，我们有以下的步骤：

> 1. **选择质数：**随机选择两个大且不同的质数p和q。
> 2. **计算模数：**计算p和q的乘积$n = p \times q$。这个n将作为公钥和私钥的一部分，并且是公开的。
> 3. **计算欧拉函数：**计算$\varphi(n) = (p-1) \times (q-1)$。注意，$\varphi(n)$是私钥生成的关键部分，但不应该被公开。
> 4. **选择加密指数：**选择一个整数e，使得$1 < e < \varphi(n)$，并且e与$\varphi(n)$互质。这个e将作为公钥的一部分，用于加密操作。
> 5. **计算解密指数：**找到一个整数d，使得$(e \times d - 1)$能被$\varphi(n)$整除。换句话说，求解模反元素d，满足$e \times d \equiv 1 (mod\ \varphi(n))$。这个$d$将作为私钥的一部分，用于解密操作。

这样，我们就得到了我们的公钥(n, e)和私钥(n, d)。我们利用$C = M^e\ (mod\ n)$来进行加密，而利用$M = C^d (\ mod\ n)$来进行解密。

同态性的证明：

$$E(m_1) * E(m_2) = (m_1^e (\bmod n)) * (m_2^e (\bmod n)) = (m_1 * m_2)^e (\bmod n) = E(m_1 * m_2)$$

RSA的同态性质表明，$E(m1 * m2)$ 可以直接使用 $E(m1)$ 和 $E(m2)$ 来评估它们，而无需解密它们。换句话说，RSA只是在乘法上是同态的。因此，它不允许密文的同态加法。

4.2 El-Gamal

生成密钥：

使用生成器g生成具有阶数n的循环群G。在循环群中，可以使用其自身元素之一的幂生成群的所有元素。然后，$h = g^y$ 为随机选择的 $y \in Z_n*$ 计算。最后，公钥是$(G，n，g，h)$，x是方案的密钥。

加密：

消息m使用g和x进行加密，其中x是从集合$\{1，2，\ldots，n-1\}$ 中随机选择的，加密算法的输出是密文对 $(c = (c1，c2))$：

$$c = E(m) = (g^x, mh^x) = (g^x, mg^{xy}) = (c_1, c_2)$$

解密：

要解密密文c，首先，$s = {c_1}^y$ 计算，其中y是密钥。然后，解密算法的工作原理如下：

$$c_2 \cdot s^{-1} = mg^{xy} \cdot g^{-xy} = m$$

同态性的证明：

$$E(m_1) * E(m_2) = (g^{x_1}, m_1 h^{x_1}) * (g^{x2}, m_2 h^{x_2}) = (g^{x1+x_2}, m_1 * m_2 h^{x_1+x_2}) = E(m_1 * m_2)$$

4.3 Paillier

该算法是一种加法同态的加密运算。

我们随机选择两个大质数p和q满足$gcd(pq,(p-1)(q-1))=1$。这个属性是保证两个质数长度相等。gcd即求两个数的最大公约数。计算$n=pq$和$\lambda=lcm(p-1,q-1)$。Lcm即求两个数的最小公倍数。

选择随机整数$g,g\in\mathbf{Z}_{n^2}^*$,使得$\gcd(L(g^\lambda \bmod n^2),n)=1$。此处定义$L(x)=(x-1)/n$。计算$\mu=(L(g^\lambda \bmod n^2))^{-1} \bmod n$，公钥为$(n,g)$,私钥为$(\lambda,\mu)$。

加密的过程如下所示：

- 明文m，$0<=m<=n$
- 随机数r，$0<r<n$，且$gcd(r,n)=1$
- 加密结果：$c=g^m\times r^n \bmod n^2$

对应的解密过程就是：

$$m=L(c^\lambda) \bmod n^2\times\mu \bmod n$$

同态性的证明：

对于明文m_1,m_2，P加密后的结果为：

$$E(m_1)\equiv g^{m_1}r_1^n \bmod n^2$$

$$E(m_2)\equiv g^{m_2}r_2^n \bmod n^2$$

$$E(m1)E(m2)\equiv g^{m_1}r_1^n\cdot g^{m_2}r_2^n \bmod n^2\equiv g^{m_1+m_2}(r_1r_2)^n \bmod n^2\equiv E(m_1+m_2)$$

这样就证明了该算法的同态性质。

4.4 第一代全同态加密方案——Gentry方案

$Gentry$方案是一种基于电路模型的全同态加密算法，支持对每个比特进行加法和乘法同态运算。$Gentry$方案的基本思想是构造支持有限次同态运算的同态加密算法并引入“$Bootstrapping$”方法控制运算过程中的噪音增长，这也是第一代全同态加密方案的主流模型。

“$Bootstrapping$”方法通过将解密过程本身转化为同态运算电路，并生成新的公私钥对对原私钥和含有噪音的原密文进行加密，然后用原私钥的密文对原密文的密文进行解密过程的同态运算，即可得到不含噪音的新密文。但是，由于解密过程本身的运算十分复杂，运算过程中也会产生大量噪音，为了给必要的同态运算需求至少预留足够进行一次乘法运算的噪音增长空间，需要对预先解密电路进行压缩简化，即将解密过程的一些操作尽量提前到加密时完成。

4.5 第二代全同态加密方案——BGV/BFV方案

$Gentry$方案之后的第二代全同态加密方案通常基于$LWE/RLWE$假设，其安全性基于代数格上的困难问题，典型方案包括BGV方案和BFV方案等。

BGV（$Brakerski-Gentry-Vaikuntanathan$）方案是目前主流的全同态加密算法中效率最高的方案。在BGV方案中，密文和密钥均以向量表示，而密文的乘积和对应的密钥乘积则为张量，因此密文乘法运算会造成密文维数的爆炸式增长，导致方案只能进行常数次的乘法运算。BGV方案采用密钥交换技术控制密文向量的维数膨胀，在进行密文计算后通过密钥交换将膨胀的密文维数恢复为原密文的维数。同时，BGV方案可采用模交换技术替代$Gentry$方案中的“$Bootstrapping$”过程，用于控制密文同态运算产生的噪声增长，而不需要通过复杂的解密电路实现。因此，在每次进行密文乘法运算后，首先需要通过密钥交换技术降低密文的维数，然后通过模交换技术降低密文的噪声，从而能够继续进行下一次计算。

BFV（$Brakerski/Fan-Vercauteren$）方案是与$BGV$方案类似的另一种第二代全同态加密方案，同样可基于$LWE$和$RLWE$构造。$BFV$方案不需要通过模交换进行密文噪声控制，但同样需要通过密钥交换解决密文乘法带来的密文维数膨胀问题。

目前，最为主流的两个全同态加密开源库$HElib$和$SEAL$分别实现了BGV方案和BFV方案。

4.6 第三代全同态加密方案——GSW方案

GSW（$Gentry-Sahai-Waters$）方案是一种基于近似特征向量的全同态加密方案。该方案基于LWE并可推广至$RLWE$，但其的性能不如BGV方案等其他基于$RLWE$的方案。GSW方案的密文为矩阵的形式，而矩阵相乘并不会导致矩阵维数的改变，因此GSW方案解决了以往方案中密文向量相乘导致的密文维数膨胀问题，无需进行用于降低密文维数的密钥交换过程。

4.7 浮点数全同态加密方案——CKKS方案

$CKKS$（$Cheon-Kim-Kim-Song$）方案是2017年提出的一种新方案，支持针对实数或复数的浮点数加法和乘法同态运算，得到的计算结果为近似值，适用于机器学习模型训练等不需要精确结果的场景。由于浮点数同态运算在特定场景的必要性，$HElib$和$SEAL$两个全同态加密开源库均支持了$CKKS$方案。