同态加密算法

同态加密(Homomorphic Encryption,HE)
满足同态运算性质的加密算法
即数据经过同态加密之后，对密文进行特定的计算，得到的密文计算结果再进行对应的同态解密后的明文等同于对明文数据直接进行相同的计算得到的结果。
同态映射定义
X与Y是两个环，若存在一个映射f:X→Y，使得∀x,y∈X都有
f(x+y)=f(x)+f(y) f(x•y)=f(x)•f(y)
则称f是一个从X到Y的同态映射或称环X与Y同态，记作X~Y.
同态加密**方案**
加密方案:(Enc,Dec,Add)，Enc和Dec分别表示加密和解密操作
明文:$m_1\ m_2$　公钥:pk　私钥:sk
生成密文:c_1=Enc(m_1,pk), c_2= Enc(m_2,pk)
则有:Dec(Add(c_1 , c_2),sk)=$m_1 + m_2$

同态加密算法的分类大致有部分同态加密(Partially Homomorphic Encryption,PHE)、近似同态加密/类同态加密 (Somewhat Homomorphic Encryption,SWHE)、全同态加密(Fully Homomorphic Encryption,FHE)三种。
部分同态加密：只支持加法运算或乘法运算中的一种。有乘法同态算法（RSA、ElGamal），或是加法同台算法Paillier
近似同态加密/类同态加密：支持有限次数的加法和乘法操作，但计算深度有限。（例如BNG,
支持任意次加法同态和一次乘法同态运算）
全同态加密：任意执行加法和乘法操作，理论上可以执行任意深度的运算，是同态加密研究的理想目标。
布尔电路实现：FHEW　TFHE　GSW
算术电路实现：BFV　· BGV　CKKS

下面是一些同态加密运算的**应用场景**：（云计算）用户的数据在加密状态下进行处理，云服务提供商无法直接访问或解读数据的原始内容，由云服务器直接对密文进行计算，计算结果解密后与直接在明文上操作的结果一致，从而确保了用户数据的隐私安全，无需担心数据泄露。（医疗健康）保护患者的病历、医疗影像等敏感信息，安全地分析医疗数据，而无需将数据解密，有效防止患者隐私泄露。（金融服务）处理大量包括个人身份信息、信用卡信息、贷款信息等的客户数据时保护客户数据的隐私和安全，避免因数据泄露而导致的法律和经济风险。
联邦学习：联邦学习(Federated Learning)是一种分布式机器学习方法，它允许多个客户端（如移动设备、浏览器或分布式服务器）协作训练一个共享模型，同时保持数据的隐私和安全。
联合建模过程中的参数交互计算:同态加密用于联邦学习中的参数交互计算过程，实现预测模型的联合确立。在联邦学习中多个参与方可以在保证各自数据隐私的同时实现联合机器学习建模，即在不获取对方原始数据的情况下利用对方数据提升自身模型的效果。
提高数据隐私保护:同态加密允许在加密数据上直接进行计算，这意味着参与方的数据在整个训练过程中始终处于加密状态，从而保护隐私。这种方式可以确保数据的隐私性，同时允许进行有效的模型训练。

以下是有关同态加密的数学问题：
以Paillier算法为例：
密钥生成
1. 生成两个大素数 p, q 。
2. 计算 n = pq， g = n + 1， $\lambda = lcm\left(p-1\ ,q-1\right)$
3. 定义函数 $\mathrm{L(x)} = \frac{x-1}{n}$
4. 计算 $u = \left(L\left(g^{\lambda}\ mod\ n^2\right)\right)^{-1}$
5. 得到公钥 (n, g)， 私钥 $(\lambda\ ,\mu)$ 。
加密过程
1. 选取明文 m 。
2. 选择随机数 r （0 < r < n）。
3. 计算得到密文 c =$g^m\ r^n\left(mod\ n^2\right)$
解密过程：
计算得到明文 m = $L\left(c^{\lambda}\ mod\ n^2\right)\cdot\mu\left(\mathrm{mod}\ n\right) = \frac{L\left(c^{\lambda}\ mod\ n^2\right)}{L\left(g^{\lambda}\ mod\ n^2\right)}$

以下是**Paillier算法正确性分析**
由最小公倍数可得(p-1)|λ,(q-1)|λ，可令λ=k_1(p-1)=k₂(q-1),
由欧拉定理可得$g^{\varphi(P)}=g^{P-1}\equiv 1(\mathrm{mod}\quad p)$,故$g^{\lambda}=g^{k_1\quad P-1}\equiv 1(\mathrm{mod}\quad p)$。 同理有$g^{\lambda}\equiv 1(\mathrm{mod}q)$.所以$(g^{\lambda}\text{-}1)|p$,$(g^{\lambda}\text{-}1)|q$,所以$(g^{\lambda}-1)|n$,即$(g^{\lambda}\text{-}1)|n^2$,可得$g^{\lambda}\equiv 1\ (mod\ n^2)$ $\{9^{\lambda}\bmod n_1^2\}\equiv 1\ (\mathrm{mod}\ n)$
可令$g^{\lambda}\ mod\ n^2=n\ k_g+1$。即L（$g^{\lambda}\ mod\ n^2$）=$k_g$.
由二项式定理得$(1+kn)^m\equiv 1+$ kmn(modn²),故$g^{m\lambda}=\left(n\ k_g+1\right)^m=kg\mathrm{mn}+1(\mathrm{mod}\quad)$,同理，故L($c^{\lambda}$modn²)= L($g^{m\lambda}\ r^{n\lambda}\ mod\ n^2$)=L($k_g$mn +1)= k_gm,所以$\frac{L\left(c^{\lambda}\ mod\ n^2\right)}{L\left(g^{\lambda}\ mod\ n^2\right)}$=m.

Paillier算法同态性分析
加法同态:
明文加法m_1+m_2
明文加密$Enc\left(m_1+m_2\right)=g^{m_1+m_2}(r)^n\bmod n^2$ r
密文乘法$C_1\cdot C_2=g^{m_1+m_2}\quad r_1+r_2\quad^n\bmod n^2$

...
Paillier算法安全性分析:
（大整数分解问题）：破解密码:已知公钥n=pq，难以推出素数p和q，故难以破解出私钥λ
（复合剩余类问题(Decisional Composite Residuolity Assumption,DCRA)）：复合剩余类问题指的是:给定一个合数n和整数z，很难确定是否存在一个整数y，使得z ≡y"(mod n-) .即判断z是不是模m²的n阶剩余是很困难的。复合剩余类问题的困难性是Paillicr算法的安全性的基础。由于复合剩余类问题的困难性，|Paillier算法能够抵抗多种攻击，包括选择明文攻击(CPA)和选择密文攻击(CCA)。这意味着即使攻击者拥有大量的密文和对应的明文，或者能够选择密文并获取对应的明文，也无法有效地破解Paillicr算法的加密。

素性检测

素性检测是密码学中的一个重要数学问题，它涉及到判断一个给定的数是否为素数。在密码学领域，素性检测的应用非常广泛，尤其是在公钥密码系统中。以下是素性检测在密码学中的一些**关键应用**:
1. RSA算法：RSA是最著名的公钥加密算法之一，它依赖于大素数的乘积作为其安全基础。素性检测用于生成RSA密钥对，确保所选的素数是安全的，从而保证加密系统的安全性。
2. ElGamal加密：这是一种基于离散对数问题的加密算法，同样需要大素数来保证安全性。素性检测在ElGamal算法中用于选择安全的素数，以构建加密和解密过程中所需的数学结构。
3.数字签名：素性检测在数字签名算法中也扮演着重要角色，如DSA（数字签名算法）和ECDSA（椭圆曲线数字签名算法）。这些算法依赖于素数来生成签名和验证签名的有效性。
4.椭圆曲线密码学（ECC）：在ECC中，素性检测用于选择椭圆曲线上的点，这些点的阶（即最小的正整数n，使得点乘以n后得到无穷远点）必须是素数。这确保了椭圆曲线的密码学性质，使得ECC能够提供与RSA相同级别的安全性，但使用更短的密钥。
5.素性测试算法：在密码学中，素性测试算法如Miller-Rabin测试和Solovay-Strassen测试被用来快速判断一个数是否为素数。这些算法对于构建和验证加密系统中的素数至关重要。
6.密钥生成：在许多加密系统中，素性检测是密钥生成过程中的一个关键步骤。它确保了密钥的随机性和不可预测性，从而提高了整个系统的安全性。
7.密码协议：在一些密码协议中，如零知识证明，素性检测可以用来验证参与者的密钥而不泄露密钥本身，这对于保护通信的隐私和完整性至关重要。
素性检测的数学问题在密码学中的应用是多方面的，它不仅关系到加密算法的安全性，还涉及到密钥管理和密码协议的有效实施。随着计算能力的提高，素性检测算法也在不断发展，以应对新的安全挑战。

其中有关**数学原理**的涉及的有Miller-Rabin测试和Solovay-Strassen测试，两者都是用于判断一个数是否为素数的概率性算法，它们在密码学中有着广泛的应用。这两种测试在某些方面有相似之处，但也存在一些关键的区别:
1. 原理
   - Miller-Rabin测试基于费马小定理的扩展，它通过检查一个数的某些属性来确定其是否为素数。如果一个数是合数，Miller-Rabin测试能够以很高的准确率识别出来。
   - Solovay-Strassen测试则基于欧拉准则和Jacobi符号，它同样可以有效地检测合数，但其正确性依赖于推广的黎曼猜想。
2. 随机性:

- Miller-Rabin测试是一种随机算法，它通过随机选择基数来测试一个数的素性。这种随机性使得算法在大多数情况下能够正确地识别素数和合数。
- Solovay-Strassen测试也是一种随机算法，但它的随机性体现在选择不同的基数上，用于计算Jacobi符号。

3. 错误概率：
- Miller-Rabin测试的错误概率可以调整，通过增加测试的轮数来降低错误概率。在实际应用中，可以通过多次测试来提高准确性。
- Solovay-Strassen测试的错误概率相对较高，因为它依赖于未经证实的数学猜想。然而，通过重复测试，也可以提高其准确性。

4. 应用范围：
- Miller-Rabin测试因其简单性和可调的“精度”而被广泛使用，尤其是在密码学和加密算法中，如RSA和ElGamal加密。
- Solovay-Strassen测试虽然在理论上有其优势，但由于其依赖于未经证实的猜想，因此在实际应用中可能不如Miller-Rabin测试受欢迎。

5. 时间复杂度：
- 两种算法的时间复杂度都为$O\left(k\left(\log n\right)^3\right)$，其中k为测试次数，n为待测数。这意味着随着待测数的增大，所需的计算时间会以多项式速度增长。

总的来说，Miller-Rabin测试因其较高的准确性和在密码学中的广泛应用而更受青睐，而Solovay-Strassen测试虽然在理论上有其价值，但在实际应用中可能受到其依赖于未经证实猜想的限制。

大整数分解问题

大整数分解问题是密码学中的一个核心数学问题，它涉及到将一个大的合数分解为其素数因子的乘积。这个问题在密码学中的运用非常广泛，尤其是在公钥密码体系的设计中。以下是大整数分解问题在密码学中的一些**关键应用**：

1. RSA算法：RSA是最著名的公钥加密算法之一，其安全性基于大整数分解问题的困难性。在RSA中，公钥由两个大素数的乘积构成，而私钥则是这两个素数。由于将大整数分解为素数因子极其困难，这保证了RSA算法的安全性。
2. Diffie-Hellman密钥交换：虽然Diffie-Hellman本身不直接依赖于大整数分解问题，但它通常与有限域上的离散对数问题结合使用，而离散对数问题的难度与大整数分解问题在某种程度上是相关的。
3. 椭圆曲线密码学（ECC）：ECC是一种基于椭圆曲线数学的密码学方法。虽然ECC的安全性依赖于椭圆曲线上的离散对数问题，但大整数分解问题在某些ECC实现中也扮演着重要角色，尤其是在密钥生成和签名验证过程中。
4. 整数分解算法：大整数分解问题的困难性促使了多种算法的发展，如二次筛法、椭圆曲线分解算法以及数域筛法。这些算法在密码学中用于破解或测试加密系统的安全性。
5. 密码学协议的安全性证明：在某些密码学协议中，安全性证明可能会归约到大整数分解问题的难度。这意味着如果能够高效解决大整数分解问题，那么这些协议的安全性也将受到威胁。
6. 密码学研究：大整数分解问题不仅是实际应用中的难题，也是密码学研究的重要课题。研究者们不断探索更有效的分解算法，同时也在寻找能够抵抗这些算法的新型加密方法。

大整数分解问题的困难性是现代密码学安全性的基石之一。随着计算技术的发展，对大整数分解问题的研究也在不断进步，这对密码学领域既是挑战也是机遇。

以下是有关其的一些数学原理：

大整数分解问题涉及到的数学原理主要包括数论中的几个核心概念：

1. 素数（Prime Numbers）：素数是只有两个正除数（1和它本身）的自然数。素数在整数分解问题中扮演基础角色，因为任何合数都可以分解为素数的乘积。
2. 合数（Composite Numbers）：合数是有多于两个正除数的自然数。合数可以被分解为两个或更多素数的乘积。
3. 最大公约数（Greatest Common Divisor, GCD）：两个或多个整数共有的最大正除数。在整数分解中，GCD可以用来简化问题，例如，通过欧几里得算法找到两个数的GCD。
4. 欧几里得算法（Euclidean Algorithm）：一种用于计算两个整数GCD的算法。它在整数分解的初步步骤中非常有用。
5. 费马小定理（Fermat's Little Theorem）：如果\(p \)是一个素数，\(a \)是一个不被\(p \)整除的整数，那么\(a^{p-1} \equiv 1 \mod p \)。这个定理在某些整数分解算法中被用来检测合数。
6. 中国剩余定理（Chinese Remainder Theorem, CRT）：如果一个人知道一个数除以几个整数的余数，那么他可以唯一确定这个数模这几个整数乘积的值。CRT在某些分解算法中

用于处理模运算。
7. 二次筛法（Quadratic Sieve, QS）：一种基于寻找平方同余的整数分解算法。它通过构建一个由平方数构成的集合，然后找到合适的线性组合来分解整数。
8. 椭圆曲线分解算法（Elliptic Curve Factorization, ECF）：虽然这种方法主要用于椭圆曲线密码学，但它也涉及到整数分解，特别是在选择椭圆曲线参数时。
9. 数域筛法（Number Field Sieve, NFS）：目前已知最快的通用整数分解算法，它通过构建数域并应用筛法来找到整数的因子。
10. 线性筛法（Linear Sieve）：一种改进的筛法，用于在数域筛法中更有效地找到关系。
大整数分解问题的困难性在于，随着整数的位数增加，所需的计算资源和时间呈指数级增长。这种困难性是许多现代加密算法（如RSA）安全性的基础。