探究报告

2311205 网络空间安全学院 李欣航

探究题目一 同态加密算法

同态加密算法是一种加密技术，允许在加密数据（即密文）上进行计算，而无需解密数据。这意味着，用户可以在云端等不受信任的环境中对加密数据进行处理，并且处理结果只有授权方才能解密得到原始结果。其核心优势在于能够在保证隐私的情况下进行计算，广泛应用于隐私保护计算、数据分析等领域。

加密方案：(Enc, Dec, Add)，Enc 和 Dec 分别表示加密和解密操作

明文：$m1, m2$ 公钥：pk 私钥：sk

生成密文：$c1 = \mathrm{Enc}(m1, \mathrm{pk}), c2 = \mathrm{Enc}(m2, \mathrm{pk})$

则有：$\mathrm{Dec}(\mathrm{Add}(c1, c2), \mathrm{sk}) = m1 + m2$

一、密码原语

1.部分同态加密

部分同态加密方案只支持对加密数据进行单一类型的数学运算（通常是加法或乘法）。这类算法在性能上相对较高，但无法处理复杂的多操作计算。

常见的部分同态加密算法：

（1）Paillier 加密（加法同态）：允许在密文上进行加法操作。它基于大整数分解的困难性，支持如下操作：

$E(a)+E(b)=E(a+b)$ $E(a)+E(b)=E(a+b)$ $E(a)+E(b)=E(a+b)$

Paillier 加密可以有效地处理一些隐私保护计算，如对加密数据的求和操作。

（2）RSA 加密（乘法同态）：RSA 算法是一种基于大数分解的公钥加密方案，具有乘法同态性质。

虽然 RSA 支持乘法同态，但由于其安全性依赖于大整数分解问题的困难性，RSA 并不适用于复杂同态加密应用。

2.全同态加密

全同态加密（FHE）是一类能够支持在加密数据上进行任意类型的计算（加法和乘法）的加密方案。全同态加密突破了部分同态加密的限制，可以支持更复杂的运算，如布尔运算、逻辑运算、更多的算术运算等。FHE 是同态加密领域的“终极目标”。

主要全同态加密算法：

（1）Gentry 算法：Gentry 是全同态加密领域的奠基人，他在 2009 年提出了第一个实用的全同态加密方案。该算法基于理想 lattices 和格基问题，能够在加密数据上执行任意数量的加法和乘法运算。Gentry 的方案在理论上具有突破意义，但在实践中仍存在效率问题。

（2）BGV 方案：是对 Gentry 方案的一种改进，采用了更高效的技术，支持加密数据上进行加法和乘法运算。BGV 方案基于学习与错误（LWE）问题，比 Gentry 的原始方案更加实用，且已在多个应用中取得实际进展。

（3）BFV 方案：是另一种改进方案，优化了同态加密的性能，特别是在加密数据的支持上，它引入了密文的“关键信息”来提高计算效率，适用于多种加密计算场景。

（4）CKKS 方案：是一种支持近似同态加密（Approximate Homomorphic Encryption）的方案，专为加速数值计算（如机器学习）而设计，能够进行近似运算，从而减小计算和存储的开销。

3.近似同态加密（Approximate Homomorphic Encryption，AHE）

近似同态加密允许在加密数据上进行近似计算，即计算结果可能会与明文结果略有不同。它适用于大多数机器学习和数据分析任务，能够大大提高效率，减少计算复杂度。

CKKS 方案：是一个典型的近似同态加密方案，专门设计用来支持加密后的数据处理，允许在密文上进行带有误差的加法和乘法运算，适合处理如浮点运算等。

二、应用场景

1.云计算：在云计算中，用户可以将计算和存储需求外包给云服务提供商，以节省自身的软硬件成本。然而，直接将明文数据交给云服务器存在安全风险。采用同态加密技术，用户可以在保护数据隐私的同时，利用云服务提供商的强大算力资源实现数据的托管存储和处理。

2.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。

3.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。

4.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。

（1）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。

（2）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。

三、数学问题

1.Paillier 算法

（1）密钥生成

生成两个大素数 p,q，

计算$n = pq, g = n + 1, \lambda = lcm(p - 1, q - 1)$,

定义函数$L(x) = \frac{x-1}{n}$

计算$\mu = \left(L\left(g^{\lambda} mod n^2\right)\right)^{-1} (mod\ n)$

得到公钥(n, g)，私钥(λ, μ).

（2）加密过程

选取明文m

选择随机数 r(0<r<n)

计算得到密文$c = g^m r^n (mod\ n^2)$

（3）解密过程

计算得到明文$m = L\left(c^{\lambda} mod n^2\right) \cdot \mu (mod n) = \frac{L(c^{\lambda} mod n^2)}{L(g^{\lambda} mod n^2)}$

（4）正确性分析

由最小公倍数可得$(p-1)|\lambda,(q-1)|\lambda$，可令$\lambda=k_1(p-1)=k_2(q-1)$，

由欧拉定理可得$g^{\varphi(p)}=g^{p-1}\equiv 1(\mathrm{mod}\,p)$，故$g^{\lambda}=g^{k_1(p-1)}\equiv 1(\mathrm{mod}\,p)$. 同理有$g^{\lambda}\equiv 1(\mathrm{mod}\,q)$.

所以$(g^{\lambda}-1)|p,(g^{\lambda}-1)|q$,所以$(g^{\lambda}-1)|pq$,即$(g^{\lambda}-1)|n$. 可得$(g^{\lambda}-1)|n^2,g^{\lambda}\equiv 1(\mathrm{mod}\,n^2)$.

$(g^{\lambda}\bmod n^2)\equiv 1(\mathrm{mod}\,n)$. 可令$g^{\lambda}\bmod n^2=nk_g+1$,则$L(g^{\lambda}\bmod n^2)=k_g$.

由二项式定理得$(1+kn)^m=C_m^0(kn)^0+C_m^1(kn)^1+C_m^2(kn)^2+\cdots+C_m^m(kn)^m\equiv 1+kmn(\mathrm{mod}\,n^2)$,

故$g^{m\lambda}=(nk_g+1)^m\equiv k_gmn+1(\mathrm{mod}\,n^2)$,同理$r^{n\lambda}=(nk_r+1)^n\equiv k_rn^2+1\equiv 1(\mathrm{mod}\,n^2)$.

故$L(c^{\lambda}\bmod n^2)=L(g^{m\lambda}r^{n\lambda}\bmod n^2)=L(k_gmn+1)=k_gm$,

所以$\dfrac{L(c^{\lambda}\bmod n^2)}{L(g^{\lambda}\bmod n^2)}=\dfrac{k_gm}{k_g}=m.$

（5）同态性分析

i)加法同态：

明文m_1,m_2 加密↓↓ 密文c_1,c_2

明文加法$m_1+m_2\Rightarrow$明文加密$\mathrm{Enc}(m_1+m_2)=g^{m_1+m_2}(r)^n\bmod n^2$

密文乘法$c_1\cdot c_2=g^{m_1+m_2}(r_1r_2)^n\bmod n^2$

ii)数乘同态：

$\mathrm{Enc}(km)=g^{km}(r)^n\bmod n^2$

$(\mathrm{Enc}(m))^k=c^k=(g^m(r)^n\bmod n^2)^k=g^{km}(r^k)^n\bmod n^2$

（6）安全性分析

i)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

ii)复合剩余类问题

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$z\equiv y^n(mod\ n^2)$即判断 z 是不是模n^2的 n 阶剩余是很困难的

2.ElGamal 算法

（1）密钥生成

生成一个大素数 p，g 是 p 的原根，随机选择一个私钥 x(1<x<p−1)，计算得公钥 $y\equiv g^x(mod\ p)$

（2）加密过程

选取明文 m，选择一个随机数 k(1<k<p−1)，计算得到密文$c_1\equiv g^k(mod\ p)$,

$c_2\equiv m\cdot y^k(mod\ p)$

（3）解密过程

计算$k' \equiv k^{-1}(mod\ p-1)$

计算得到明文$m \equiv c_2 c_1^{-x} y^{k' c_2}(mod\ p)$

（4）同态性

明文m_1, m_2

加密

密文c_1, c_2

有Dec$(c_1 c_2)$=$m_1 m_2$

（5）安全性分析

i) 离散对数问题

有限域上的离散对数指的是：设 p, q 为两个素数，$G = \{g^i | 0 \le i \le q-1, g \in Z_p^*\}$为阶为 q 的有限域$Z_p^*$上的乘法群。给定一个元素$y \in G$，找到一个整数$x \in Z_q$使得$y = g^x$的问题

离散对数问题的难点在于，对于某些群（特别是大素数阶的群），不存在已知的多项式时间算法来有效地求解 x。一些求解思路：穷举法、商客法（小步大步法）、$Pollard\ \rho$算法、Pohlig–Hellman 算法、指数积分算法等

探究题目二 素性检测

一、数学问题：素性判定

素数判定问题是指给定一个正整数 n，判断它是否为素数，素数是指只能被 1 和它本身整除的自然数。即对于一个整数 n，如果 n 除了 1 和 n 自身外没有其他正整数能整除它，那么 n 就是素数。

对于大数，素性检测变得尤为困难，因为传统的试除法判断素数的时间复杂度较高。随着密码学中大数的应用日益广泛，寻找更高效的素性检测算法成为了一个重要的研究方向。

二、算法思想及其特点

在素性检测中，最常用的两种概率性算法是 Miller–Rabin 算法和 Solovay–Strassen 算法。这两种算法都利用了数论中的数学性质，特别是费马小定理、欧拉定理等内容，能够高效地检测大数是否为素数。尽管它们是概率性算法，并不保证每次

都能准确地判断一个数是否为素数，但通过多次测试，可以极大地降低错误判定的概率。

2.1 Miller−Rabin 算法

米勒-拉宾素性检测算法是一种基于数论的概率性算法，广泛应用于大数素性测试。该算法的核心思想来源于费马小定理和欧拉定理，并且可以通过二次剩余和平方剩余等技巧来验证素性。其**基本步骤**如下：

1.费马小定理：若 p 是素数，且 a 是任意不等于 0 和 p 的整数，则有$a^{p-1} \equiv 1(mod\ p)$。米勒-拉宾算法利用了这一点来进行素性检测。

2.给定待检测的数 n，首先将 n−1 分解成$2^s \times d$的形式，其中 d 为奇数。

3.随机选取一个整数 a，并计算$a^d mod\ n$。如果结果为 1 或 n−1，则 n 可能是素数。如果结果不为 1 或 n−1，则进一步检查$a^{2^r \cdot d} mod\ n$（对于 r=0, 1, 2, ……, s−1）

4.如果在任何步骤找到了$a^{2^r \cdot d} mod\ n = n - 1$，则 n 可能是素数。如果没有找到，且算法中的每一轮都没有通过，则 n 不是素数。

时间复杂度：每一次测试的复杂度大约是 O(klog⁡3n)O(k \log^3 n)O(klog3n)，其中 kkk 是测试次数，nnn 是待测试的数。

优点：相较于其他算法，米勒-拉宾算法非常高效，适用于大数的素性检测。通过多次独立测试，可以大大降低错误率。

缺点：作为概率性算法，米勒-拉宾算法不能保证每次都能给出准确的结果，但通过增加测试次数，错误的概率可以变得极小。

2.2 Solovay−Strassen 素性测试

Solovay−Strassen 素性测试是一种与米勒-拉宾算法类似的概率性素性检测算法，但它依赖于 Jacobi 符号和欧拉定理。该算法的核心思想是利用二次剩余性质，通过检验某个数是否为二次剩余来判断其素性。其**基本步骤**如下：

1.给定待检测的数 n，首先将 n−1 分解成$2^s \times d$的形式，其中 d 为奇数。

2.随机选取一个整数$a(1 \leq a < n)$并计算 Jacobi 符号$(\frac{a}{n})$。

3.计算$a^d mod\ n$，如果结果为 1 或 n−1，则 n 可能是素数。

4.计算$a^d mod\ n$的结果不为 1 或 n−1，则计算$a^{2^r \cdot d} mod\ n$（对于 r=0, 1, 2, ……, s−1）, 并检查是否满足 Jacobi 符号的条件。

5.如果没有发现错误，且多次测试都通过，则 n 可能是素数。如果发现不满足条件，则可以确定不是素数。

时间复杂度：每次测试的复杂度大约是$O(klog^3n)$，其中 k 是测试次数，n 是待测试的数。

优点：Solovay−Strassen 算法相较于其他素性检测算法，在某些情况下有较好的性能，尤其是在处理大数时。通过 Jacobi 符号的快速计算，使得它在某些场景下优于米勒−拉宾算法。

缺点：同样作为概率性算法，Solovay−Strassen 算法也不能保证每次都能准确判断数是否为素数。通过增加测试次数，可以降低错误的概率。

四、素性检测在密码学中的应用

1.密钥生成:在许多密码学协议中，特别是在 RSA 和其他基于大素数分解的加密算法中，素数用于生成公私钥对。生成大素数是其中的关键步骤，而素性检测算法则用于确保所选的数是素数。例如，在 RSA 加密中，选取两个大素数 p 和 q，然后计算它们的乘积 n=p×q 作为公钥的一部分。如果 p 或 q 不是素数，则加密系统将失去安全性。

2.随机数生成：在一些基于素数的随机数生成算法中，素性检测用于验证生成的候选数是否为素数，以提高生成的随机数的质量。某些随机数生成方法需要选择较大的素数作为种子值，确保随机性和不可预测性。

3.公钥加密算法的安全性：许多公钥加密算法（如 RSA、ElGamal）依赖于素数的性质及其分解的困难性。素性检测算法的效率直接影响到这些加密算法的生成速度和安全性。例如，在 RSA 算法中，生成一个大素数对于加密系统的安全至关重要，而素性检测算法则决定了生成过程的效率。

探究题目三 RSA 问题

在现代密码学中，RSA 算法（Rivest−Shamir−Adleman）是一种公钥加密算法，被广泛应用于数据的加密与数字签名。其安全性依赖于大数分解问题的困难性，即对大数进行因数分解所需的计算时间呈指数级增长。RSA 的设计和实现涉及到数学中的许多概念，尤其是素数、欧拉函数、模幂运算等。

一、数学问题

RSA 算法的数学基础是数论中的两个难题：大数分解和离散对数问题。大数分解问题是指将一个大数分解成其素数因子的乘积，而离散对数问题则涉及到在有限域中找到一个数的离散对数。RSA 算法的安全性主要依赖于大数分解的困难性，即对于一个非常大的合数，要将其分解成两个质数的乘积是非常困难的。

二、算法思想及其特点

RSA 的工作原理

2.1 密钥生成

1.选择两个大素数 p 和 q，并计算它们的乘积$n = p \times q$

2.计算欧拉函数$\emptyset(n) = (p-1)(q-1)$。

3.选择公钥指数 e，使得$1 < \mathrm{e} < \emptyset(n)$且e 与$\emptyset(n)$互素。

4.计算私钥指数 d，使得$e \cdot d \equiv 1(mod\ \emptyset(n))$。

5.公钥是(e, n)，私钥是(d, n)。

2.2 加密过程

1.明文消息 M 被转换为一个整数 m$(0 \leq m < n)$。

2.使用公钥(e, n)对消息 m 进行加密，得到密文$c = m^e \ mod\ n$。

2.3 解密过程

使用私钥(d, n)对密文 c 进行解密，得到明文 m，$m = c^d \ mod\ n$。

RSA 算法能够确保加密和解密的过程是相互可逆的。

三、RSA 算法的优缺点

3.1 优点

1.安全性高：RSA 的安全性基于大数因数分解问题的计算困难性，目前尚未有有效的算法能够在多项式时间内因数分解足够大的数。尤其是在使用足够长的密钥时（如 2048 位或更高），RSA 的安全性得到了广泛认可。

2.公钥加密：RSA 是一种公钥加密算法，适合用于需要保护通信内容的场景。公钥可以公开，而私钥仅由接收者保存，确保了数据的安全性。

3.数字签名：RSA 不仅可以用于加密数据，还可以用于生成数字签名，确保数据的完整性和来源的可信性。

4.广泛应用：作为经典的公钥加密算法，RSA 被广泛应用于电子商务、SSL/TLS 加密、数字证书等场景。

3.2 缺点

1.计算效率低：由于 RSA 依赖于大数的模幂运算，尤其是在加密和解密时涉及到大数的指数运算，因此 RSA 的运算速度相对较慢。加密过程中的速度比对称加密算法（如 AES）慢很多。

2.密钥长度要求高：为了确保足够的安全性，RSA 通常需要较长的密钥（如 2048 位或更高）。较长的密钥使得计算过程更加耗时，并且增加了存储和通信的开销。

3.容易受到侧信道攻击：RSA 的实现容易受到侧信道攻击（如时间攻击、电磁攻击等）。因此，RSA 的实际实现需要进行额外的保护措施，如加密算法的随机化。

四、RSA 问题在密码学中的应用

1.数字签名：RSA 被广泛用于生成和验证数字签名。数字签名是一种用于验证信息来源和数据完整性的方法。发送者使用自己的私钥对消息进行签名，接收者则使用发送者的公钥来验证签名的有效性。RSA 数字签名可以应用于软件分发、电子邮件验证、电子商务等领域。

2.密钥交换：RSA 也可以用于密钥交换协议中，例如在 SSL/TLS 协议中，RSA 用于保护初始密钥交换过程。客户端和服务器通过 RSA 交换加密的密钥信息，从而为后续的通信提供对称密钥加密的基础。

3.SSL/TLS 协议：在 SSL/TLS 协议中，RSA 是用于加密和认证的核心算法之一。通过 RSA，服务器可以向客户端提供公钥，客户端使用该公钥加密对称加密密钥，确保在不安全的网络环境中传输的安全性。

4.电子支付与电子货币：RSA 广泛应用于电子支付系统中，确保交易数据的安全性和用户身份的认证。在使用 RSA 的支付系统中，银行或支付机构为客户生成公私钥对，客户使用私钥对交易进行签名或加密，确保交易的安全。

5.数字证书：RSA 是数字证书（如 X.509 证书）中的核心算法。数字证书通常由认证机构（CA）签发，其中包含了公钥及其所有者的身份信息。RSA 用于确保数字证书的真实性和有效性，防止伪造和篡改。