AES 加密

起源：早期的加密算法如 DES 在面对新的密码分析技术（差分密码分析和线性密码分析）时逐渐暴露出安全性不足的问题。为满足现代加密需求，美国国家标准与技术研究院（NIST）于 1997 年发起了高级加密标准（AES）的征集活动，旨在寻找一种更安全、高效的对称加密算法来替代 DES。最终，Rijndael 算法脱颖而出，被选定为 AES 算法。

算法实现

AES 是一种对称分组加密算法，其加密过程主要包括以下几个关键步骤：

1、字节代换（SubBytes）：利用一个预先定义的 S 盒（Substitution Box），将输入的每个字节按照特定规则替换为另一个字节。这个操作是非线性可逆的，使得密文与明文之间的关系更加复杂，增加了线性密码分析的难度。

2、行移位（ShiftRows）：对输入数据的矩阵表示（状态矩阵）进行上的循环移位操作。具体来说，第一行保持不变，第二行循环左移 1 个字节，第三行循环左移 2 个字节，第四行循环左移 3 个字节。改变字节在不同位置上的排列，进一步扩散了数据的信息。

3、列混合（MixColumns）：把状态矩阵中的每一列视为一个多项式，并与一个固定的多项式进行模乘运算，从而实现列的混合。（或者说将状态矩阵的每一列乘上一个固定的矩阵达到改变状态矩阵的效果）。此操作是线性的，保证了加密过程中的可逆性，同时也增强了数据在列方向上的扩散效果。

4、轮密钥加（AddRoundKey）：将状态矩阵与当前轮的子密钥进行按位异或操作。每一轮加密都使用不同的子密钥，这些子密钥是由原始密钥通过密钥扩展算法生成的。轮密钥加操作将密钥信息与数据紧密结合，使得密文对密钥具有高度的依赖性。

AES 算法根据密钥长度的不同，其加密轮数也有所区别。对于 128 位密钥，加密过程包含 10 轮；192 位密钥时为 12 轮；256 位密钥则需进行 14 轮加密。

注：无论密钥长度多少，最后一轮无需列混合操作。并且轮与轮之间还需进行密钥扩展，即由当前轮密钥得到下一轮子密钥。

安全性分析：

1. 抗密码分析能力强：AES 在设计时充分考虑了抵抗各种已知的密码分析攻击方法，如差分密码分析和线性密码分析等。其复杂结构和精心设计的操作使得攻击者难以通过分析密文获取明文信息。

2. 密钥空间大：支持 128 位、192 位和 256 位的密钥长度，密钥空间极为庞大。以 128 位密钥为例，密钥空间大小为2^{128}，这意味着穷举所有可能的密钥在计算上几乎是不可行的，从而有效抵御了暴力破解攻击。

3. 数学结构复杂：基于有限域上的代数结构进行设计，具有良好的数学性质。这种复杂的数学结构使得密码分析者难以找到算法的弱点，进一步提高了安全性。

应用场景

1、网络通信安全保障：在各类网络通信协议中广泛应用，如 VPN（虚拟专用网络）、SSL/TLS（安全套接层/传输层安全协议）等，用于加密传输的数据，确保数据在网络传输过程中的保密性和完整性，防止信息被第三方窃取或篡改。

2、数据存储加密：用于保护存储在各种存储设备（如硬盘、固态硬盘、闪存卡等）中的敏感数据。企业和个人可以使用 AES 对数据库中的机密信息、个人电脑中的隐私文件等进行加密存储，防止数据泄露。
3、电子商务交易安全：在电子商务领域发挥关键作用，保障在线交易过程中用户的个人信息（如信用卡号、密码、地址等）以及交易数据的安全。通过对这些信息进行加密处理，使得交易双方能够在安全的环境下进行交易，增强了用户对电子商务平台的信任。

大整数分解问题

定义：给定一个大整数 N，它是俩个大素数的乘积，将这两个满足 N=p×q 的大素数 p,q 找出来的问题称作大整数分解问题

算法实现
针对大数分解已有多种算法，其中一些较为著名的算法包括：
1. 试除法：最基本的大数分解方法。从最小的素数 2 开始，依次用素数去除待分解的大数，直到找到一个能整除该数的素数为止，然后继续对商进行分解，重复这个过程直到商为 1。优点是简单易懂，但对于较大的数，计算效率低。
2. Eratosthenes 筛法：该方法主要用于生成一定范围内的素数表，然后利用素数表来分解大数。它的基本思想是从 2 开始，标记所有 2 的倍数为合数（非素数），然后找到下一个未被标记的数（即素数），标记其所有倍数为合数，重复这个过程直到达到小于$\sqrt{n}$的最大素数。虽然该方法在生成素数表方面效率较高，但对于非常大的数字时，仍然面临效率问题。
3. 费马分解算法（Fermat's factorization method）：基于费马定理，如果一个奇数 n 可以表示为两个平方数之差，即 $n = x^2 - y^2$，那么 n 可以分解为 $(x + y)(x - y)$。费马分解法通过寻找满足条件的 x 和 y 来分解 n。然而，这种方法在实际应用中效率不高，尤其是对于没有合适平方数差表示的大数。
4. 二次筛法（Quadratic Sieve）：这是一种较为有效的大数分解算法。它基于对二次同余方程的求解来寻找数的因子。二次筛法通过选择一系列的数 x，计算 y = x^2 mod n，并尝试将 y 分解为小素数的乘积，然后通过分析这些分解结果来寻找 n 的因子。二次筛法在处理较大数时比前面几种方法更有效，但仍然面临计算资源需求大的问题。
5. 一般数域筛法（General Number Field Sieve, GNFS）：目前已知的最有效的大数分解算法之一。它基于代数数论的思想，将大数分解问题转化为在代数数域上的计算问题。一般数域筛法通过构造合适的数域和筛选过程来寻找数的因子。虽然一般数域筛法在理论上具有较高的效率，但它的计算复杂度仍然很高，并且在实际应用中需要大量的计算资源和时间，尤其是对于非常大的数（如数千位甚至更多位的整数）。

用途：
用于密码学安全性评估：在密码学研究和实践中，大数分解问题的难度被用于评估密码算法的安全性。通过分析特定大小的数在现有计算资源下的分解难度，来确定密码算法所使用的密钥长度是否足够安全。例如，随着计算能力的提升，为保证 RSA 算法的安全性，密钥长度从早期的 512 位逐渐增加到 1024 位、2048 位甚至更高，这一过程就是基于对大数分解问题难度的评估。

RSA 问题

1.定义：令 n=pq 是一个长度至少为 1024 比特的整数，并且 p-1 和 q-1 有大素数因子。e 是一个正奇数且满足$(e, \varphi(n)) = 1$，给定一个随机整数$c \in \mathbb{Z}_n^*$，将寻找一个整数 m 使其满足$m^e \equiv c(mod\ n)$的问题称作 RSA 问题。

2.求解方法

-暴力破解：最直接的方法就是从 2 到$\sqrt{n}$依次尝试能否整除 n，若找到一个能整除的数，就可以得到 p 或 q，但这种方法计算量极大，效率极低。

-基于数论性质的方法：

-Pollard's rho 算法：利用随机化算法和数论中的某些性质来寻找 n 的因子。它通过构造一个伪随机序列，根据序列中元素的特定运算结果来判断是否找到了 n 的因子。该算法在一定程度上比暴力破解效率高，但对于非常大的数仍然面临计算资源和时间的挑战。

-椭圆曲线分解法（Elliptic Curve Factorization Method）：基于椭圆曲线的数学性质来分解大整数。它在椭圆曲线上进行点运算，通过寻找椭圆曲线上的特殊点来获取 n 的因子。这种方法在某些情况下比传统的分解算法更有效，但实现较为复杂，计算复杂度仍然较高。

强 RSA 问题

1.定义：给定一个 RSA 模数 n = pq，p 和 q 为大素数，G 是$\mathbb{Z}_n^*$的一个循环子群。给定 G 中的一个随机元素 z，将找到一组整数$(u, e) \in G \times \mathbb{Z}_n^*$使得$z \equiv u^e(mod\ n)$。与普通 RSA 问题不同的是，这里的指数 e 是任意给定的，而不仅仅局限于与$\varphi(n)$互素的情况。

2.求解方法

-一般数域筛法（General Number Field Sieve, GNFS）变体思路：

类似于在普通 RSA 问题中对大数分解的 GNFS 算法思路，尝试在强 RSA 问题的求解中构建数域环境。通过选择合适的数域，对元素进行筛选和计算，试图找到满足 $z \equiv u^e(mod\ n)$的(u, e)。但由于指数 e 的任意性，使得数域中的计算和关系更为复杂，目前这种方法计算复杂度极高，在实际处理大的 n 和任意 e 时很难取得有效结果。

-基于格的方法探索：

利用格理论来处理强 RSA 问题。构建特定的格结构，将强 RSA 问题转化为格中的计算问题，如最短向量问题（Shortest Vector Problem, SVP）或最近向量问题（Closest Vector Problem, CVP）。然而，格问题本身计算难度大，虽然在理论研究上有一定进展，但实际应用中面临诸多挑战，距离能够高效求解强 RSA 问题还有很大差距。

3.对密码学安全性的重要意义

强 RSA 问题的困难性在现代密码学中具有关键意义。许多先进的密码协议和加密方案依赖于强 RSA 假设来确保其安全性。如果强 RSA 问题能够被有效解决，那么这些基于其假设构建的密码系统将面临严重的安全威胁。

4. 应用场景

-高级密码协议构建：

在一些复杂的密码协议设计中，RSA 问题的假设被广泛应用。例如，在某些基于身份的加密（IBE）方案中，利用强 RSA 问题的困难性来实现用户公钥与身份信息的直接关联，无需传统的证书颁发机构，简化了密钥管理过程。同时，在签密方案中，RSA 假设保证了签名和加密操作的安全性，使得通信双方可以在一个步骤中同时完成签名和加密，提高了通信效率和安全性。

-安全标准制定参考：

在密码技术相关的安全标准制定过程中，RSA 问题的特性是重要的参考依据。安全标准组织在确定不同安全级别的密码算法和协议要求时，会充分考虑基于 RSA 假设的技术在实际应用中的安全性和可行性。