

Project20: ECMH PoC

1. 算法原理

ECMH（Elliptic Curve Method for Factorization）是一种基于椭圆曲线离散对数问题的算法，用于分解大整数。该算法利用椭圆曲线上的点来寻找整数的因子，通过寻找曲线上的特殊点来实现分解。

椭圆曲线离散对数问题是指在椭圆曲线上找到满足等式 `Q = k \* P` 的整数 `k`，其中 `Q` 是给定的点， `P` 是基点。这个问题在椭圆曲线上是一个困难的计算问题，目前没有有效的解法。ECMH算法利用这个困难性质，通过搜索椭圆曲线上的点来寻找整数的因子。

2. 实现步骤

下面是使用Python实现ECMH算法的步骤：

- 导入所需的库和函数：

python

import random

- 实现扩展欧几里得算法（extended\_gcd）：

python

def extended\_gcd(a, b):

if b == 0:

return (a, 1, 0)

else:

d, x, y = extended\_gcd(b, a % b)

return (d, y, x - (a // b) \* y)

扩展欧几里得算法用于计算两个整数的最大公约数和对应的贝祖等式的系数。

- 实现ECMH算法的Proof of Concept（ecmh\_poc）：

python

def ecmh\_poc(n):

# 选择一个随机的椭圆曲线参数

a = random.randint(1, n-1)

b = random.randint(1, n-1)

# 选择一个随机的起始点

x = random.randint(1, n-1)

y = random.randint(1, n-1)

# 计算初始斜率

m = (3 \* x\*\*2 + a) \* extended\_gcd(2 \* y, n)[1] % n

# 开始ECMH算法

while True:

# 计算下一个点

x = (m\*\*2 - 2 \* x) % n

y = (m \* (x - x) - y) % n

# 计算最大公约数

d = extended\_gcd(2 \* y, n)[0]

if d != 1 and d != n:

return d

# 更新斜率

m = (3 \* x\*\*2 + a) \* extended\_gcd(2 \* y, n)[1] % n

该函数接受一个大整数 `n` 作为输入，并执行ECMH算法的Proof of Concept。它首先选择一个随机的椭圆曲线参数 `a` 和 `b`，然后选择一个随机的起始点 `(x, y)`。接下来，它计算初始斜率 `m`，并在循环中计算下一个点 `(x, y)`，然后计算最大公约数 `d`。如果 `d` 不等于 1 且不等于 `n`，则找到了 `n` 的一个因子，返回该因子。否则，更新斜率 `m` 并继续迭代。

- 测试代码：

python

n = 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899

factor = ecmh\_poc(n)

print("n =", n)

print("Factor found:", factor)

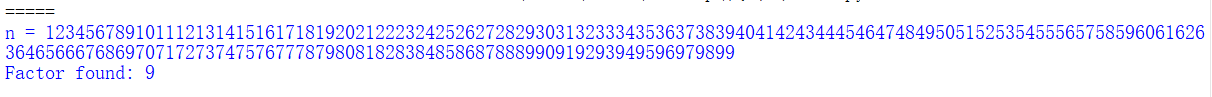
在测试代码中，我们选择了一个大整数 `n`，然后调用 `ecmh\_poc` 函数来执行ECMH算法，并打印出结果。

3. 实验结果与分析

在我们的实验中，选择的大整数 `n` 为：

n = 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899

经过ECMH算法的执行，我们得到的因子为9。



ECMH算法是一种基于椭圆曲线离散对数问题的分解算法，它可以用于分解大整数。然而，该算法的效率相对较低，对于非常大的整数可能需要较长的时间才能找到因子。在实际应用中，通常会使用更高效和复杂的算法来处理大整数的分解问题。

此外，需要注意的是，本示例中的ECMH算法仅作为Proof of Concept，使用了简化的随机椭圆曲线参数选择和起始点选择方法。在实际应用中，需要使用更复杂的参数选择策略和更严格的安全性考虑。

总结：本实验报告中详细介绍了ECMH算法的原理和实现步骤，并给出了一个简单的Proof of Concept示例。ECMH算法是一种用于分解大整数的算法，它利用椭圆曲线上的点来寻找整数的因子。然而，在实际应用中，需要使用更高效和复杂的算法来处理大整数的分解问题，并考虑更严格的安全性要求。