Diffie-Hellman密钥交换计算过程

1. 公共参数选择  
   ①素数p=101 g=2  
   ②验证原根代码

|  |
| --- |
| p = 101  g = 2  sequence = [pow(g, i, p) for i in range(1, p)]  assert len(set(sequence)) == p - 1, "g 不是原根"  print("g 是原根") |

③结果



1. 私钥与公钥计算  
   ①成员A的私钥a = 19，公钥A = g^a mod p = 98  
   ②成员B的私钥b = 45，公钥B = b^a mod p = 41
2. 共享密钥计算  
   ①A的共享密钥Ka = B^a mod p = 69  
   ②B的共享密钥Kb = A^b mod p = 69  
   验证Ka = Kb , 共享密钥一致
3. 结果  
   使用D-H密钥交换生成的共享密钥为69  
     
     
     
     
     
     
     
   附录1 D-H验证

|  |
| --- |
| import random  # 公共参数  p = 211 # 素数  g = 2 # 原根  # A选择私钥和公钥  a = random.randint(1, p - 1)  A = pow(g, a, p)  # B选择私钥和公钥  b = random.randint(1, p - 1)  B = pow(g, b, p)  # 计算共享密钥  K\_A = pow(B, a, p) # A的共享密钥  K\_B = pow(A, b, p) # B的共享密钥  # 打印结果  print(f"p = {p}, g = {g}")  print(f"A的私钥: {a}, 公钥: {A}")  print(f"B的私钥: {b}, 公钥: {B}")  print(f"A计算的共享密钥: {K\_A}")  print(f"B计算的共享密钥: {K\_B}")  assert K\_A == K\_B, "共享密钥不一致！" |

附录2 题目所给的p，g验证

图片包含 文本

描述已自动生成图片包含 日历

描述已自动生成日历

低可信度描述已自动生成文本

描述已自动生成文本

描述已自动生成文本

中度可信度描述已自动生成