**广州大学学生实验报告**

**开课学院及实验室：**网络空间安全学院 **2023年 5 月 30日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | 网络空间安全学院 | **年级/专业/班** | 网安211 | **姓名** |  | **学号** |  |
| **实验课程名称** | 密码技术及应用 | | | | | **成绩** |  |
| **实验项目名称** | 数字签名算法实验 | | | | | **指导**  **老师** | 李树栋 |

1. 实验目的

**（1）实现基于离散对数的盲签名方案，并举例验证盲签名过程**

设在该方案中，是一个大素数，是中的本原元，签名者 的密钥为，，公钥为。基于离散对数的盲签名方案如下:

### 签名过程

1) 接收者 Alice选择，随机选取，计算，将发送给 Bob。

2)Bob 选择一个随机数，计算，将发送给 Alice。

3)Alice 随机选取计算，

将发送给 Bob。

4)Bob 计算。将发送给 Alice。

5)Alice计算，的签名为。

上述签名过程如下图

|  |
| --- |
| 接收者Alice 签名者Bob |
| 随机选取，计算    随机数，计算    随机选取,计算      计算： |

#### 验签过程

验证是否成立。若成立，则签名有效；若不成立，则签名无效。

#### 签名验证过程的证明

由得

 .. (1)

若成立，则，

由此可得：  (2)

将(1)带入(2)得：

 .

比较 k 前面的系数和常数项，可得:



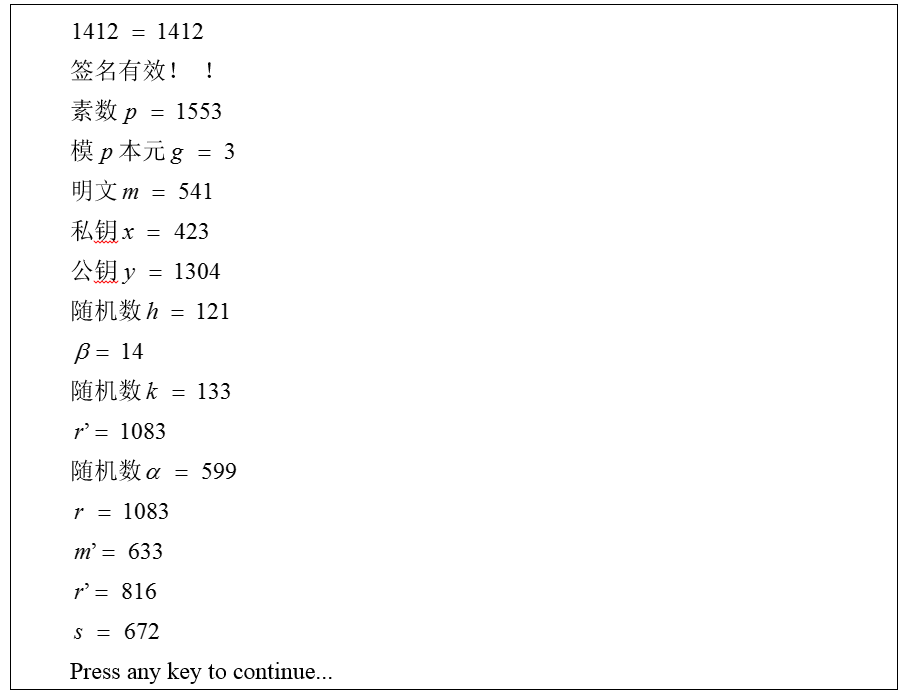


上述过程逆推，即得

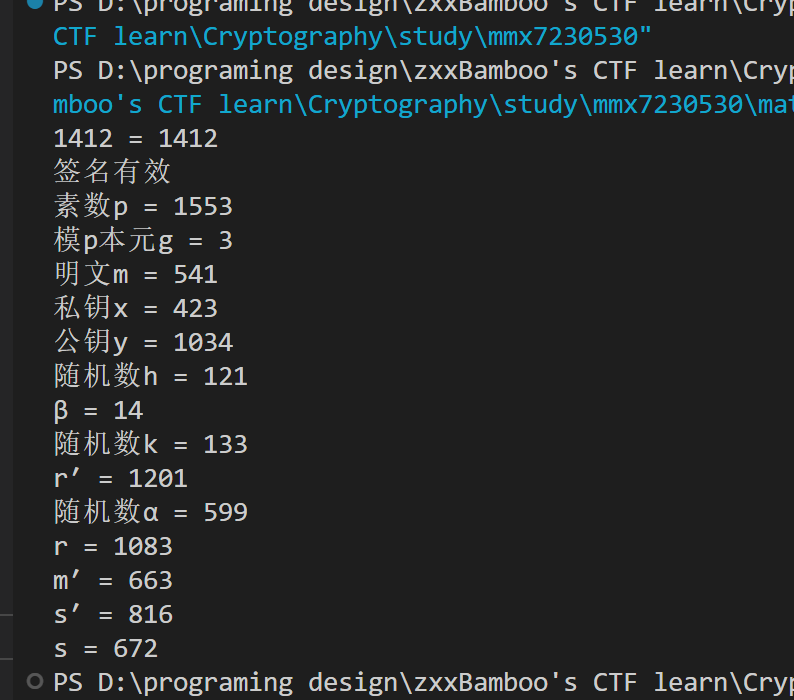
1. 实验内容

（1）实现上述基于离散对数的盲签名方案

（2）举例：假设素数，模p本原根，明文，私钥。随机数，随机数，随机数。请针对该例，验证该盲签名的实验结果。



1. 实验过程及结果
2. import random
3. from gmpy2 import invert
4. def gcd(m, n):
5. if n == 0:
6. return x
7. else:
8. return gcd(n, m % n)
9. def husu(n):  # 随机选取与n互素的数
10. m = random.randint(0, n)
11. if gcd(m, n) == 1:
12. return m
13. return husu(n)
14. p = 1553
15. g = 3
16. m = 541
17. x = 423
18. # h = husu(p-1)
19. h = 121
20. b = pow(g, h, p)  # β
21. # k = husu(p-1)
22. k = 133
23. r2 = pow(b, k, p)  # r'
24. # a = husu(p-1)
25. a = 599  # α
26. r = pow(r2, a, p)
27. r\_ni = invert(r, p - 1)  # 求逆
28. m2 = (m \* r2 \* r\_ni) % (p - 1)  # m'
29. k\_ni = invert(k, p - 1)
30. s2 = (k\_ni \* (m2 \* x - r2)) % (p - 1)  # s'
31. a\_ni = invert(a, p - 1)
32. h\_ni = invert(h, p - 1)
33. r2\_ni = invert(r2, p - 1)
34. s = (a\_ni \* h\_ni \* r \* r2\_ni \* s2) % (p - 1)
35. y = pow(g, x, p)
36. t1 = pow(y, m, p)
37. t2 = (pow(r, s) \* pow(g, r)) % p
38. if t1 == t2:
39. print(t1, '=', t2)
40. print("签名有效")
41. else:
42. print(t1, '!=', t2)
43. print("签名无效")
44. print("素数p =", p)
45. print("模p本元g =", g)
46. print("明文m =", m)
47. print("私钥x =", x)
48. print("公钥y =", y)
49. print("随机数h =", h)
50. print("β =", b)
51. print("随机数k =", k)
52. print("r′ =", r2)
53. print("随机数α =", a)
54. print("r =", r)
55. print("m′ =", m2)
56. print("s′ =", s2)
57. print("s =", s)



1. 实验总结

1. 安全性增强：基于离散对数的盲签名方案依赖于离散对数问题的困难性。可以研究更强的密码学假设，如椭圆曲线密码学，来设计更安全的盲签名方案。

2. 可扩展性改进：盲签名方案在大规模环境下可能会遇到效率和可扩展性的挑战。可以研究和设计更高效的算法、数据结构或协议来改进盲签名方案的性能和可扩展性。

3. 匿名性和隐私保护：盲签名方案通常涉及用户的匿名性和隐私保护。可以考虑采用零知识证明或混淆技术来增强用户的匿名性和保护用户的隐私。

4. 实现优化：在实际代码实现盲签名方案时，可以优化代码以提高效率。例如，采用更快速的离散对数算法、使用并行计算或硬件加速等。

5. 应用领域扩展：盲签名方案可以应用于许多领域，如数字货币、电子投票、身份验证等。可以研究如何将盲签名方案扩展到特定领域，并解决该领域的具体需求和挑战。