Lalita Sharkey 01063704

CS 764/864: Blockchains and Cryptocurrencies: Fundamentals, Technologies, and Economics

Spring 2020 Module 5 Homework 5

Q1 Blind Digital Signatures

1.1 Given p=5, q=11, determine the public key (e,n) and the private key (d,n). Assume e=7.

n = p*q = 55, Phi(n) = (p-1)*(q-1) = 40

public key (e,n) = (7, 55)

private key (d,n) = (23, 55)

1.2 Given b= 6, determine the blinding factor using the public key derived in (i)

BF = (b^e mod n) = 41

1.3 Given m=11, determine the blinded message using the information in (i) and (ii)

BM = m*BF mod n = 11

1.4 Determine the signature on the blinded message in (iii) using the private key (d,n)

s = BM^d mod n = 11

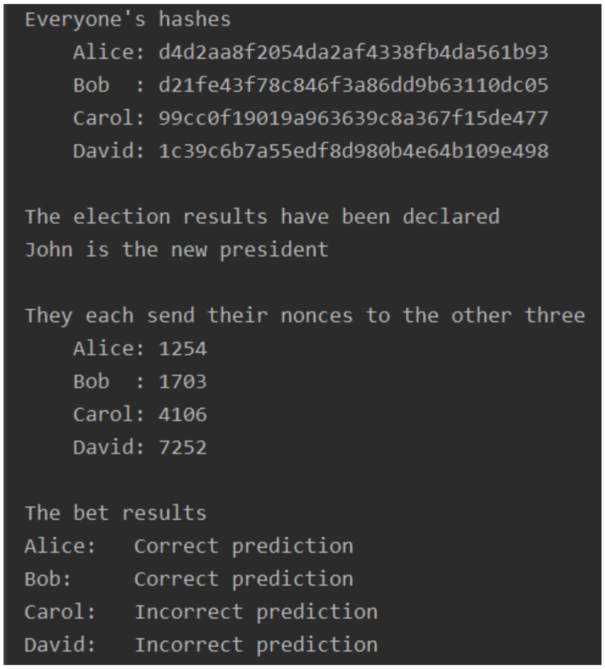
1.5 Unblind the signature derived in (iv) to determine the signature on the original message m=11

unblinded: (s/b) mod n = (11 + 1*55)/6 mod 55 = 11

1.6 Verify that the blind signature is the same as the signature on m using (d,n).

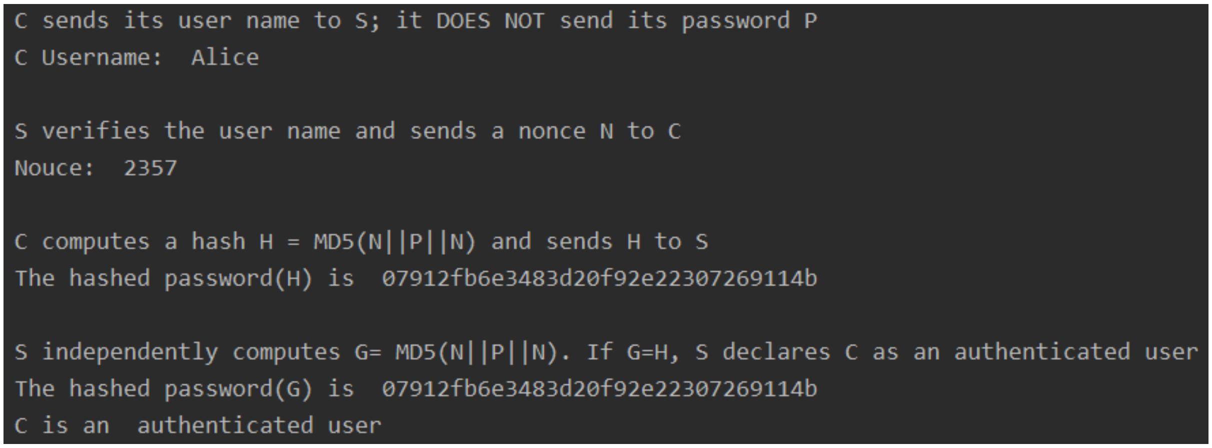
signature = (m^d) mod n = 11
Therefore, the signatures match.

Q2. Bit-commitment protocol



```
Everyone's hashes
    Alice: d4d2aa8f2054da2af4338fb4da561b93
    Bob  : d21fe43f78c846f3a86dd9b63110dc05
    Carol: 99cc0f19019a963639c8a367f15de477
    David: 1c39c6b7a55edf8d980b4e64b109e498

The election results have been declared
John is the new president

They each send their nonces to the other three
    Alice: 1254
    Bob  : 1703
    Carol: 4106
    David: 7252

The bet results
Alice:   Correct prediction
Bob:     Correct prediction
Carol:   Incorrect prediction
David:   Incorrect prediction
```

Q3. Zero-knowledge proofs



```
C sends its user name to S; it DOES NOT send its password P
C Username:  Alice

S verifies the user name and sends a nonce N to C
Nouce:  2357

C computes a hash H = MD5(N||P||N) and sends H to S
The hashed password(H) is  07912fb6e3483d20f92e22307269114b

S independently computes G= MD5(N||P||N). If G=H, S declares C as an authenticated user
The hashed password(G) is  07912fb6e3483d20f92e22307269114b
C is an  authenticated user
```

Q4 Discrete Zero-Knowledge Proofs

Alice and Bob know: p=17, A=5, and B=9. Alice knows x is 10 but does not reveal it to Bob.

1. Alice chooses a random number r < p.
   a. Let Alice picks 5 as r
2. Alice sends q = (A^r mod p) to Bob.
   a. q = 14
3. Bob sends a random bit to Alice.
   a. Random bit = 1
4. Alice computes s = (r + i*x) mod (p-1) and sends to Bob.
   a. S = 15
5. Bob computes C = A^s mod p. This should be equal to D = q*(B^i) mod p.
   a. C = 7
   b. D = 7
6. Since C is equal to D, it means Alice knows the value of x.

Lalita Sharkey 01063704

Readme for Q2 and Q3 codes

Requirement:

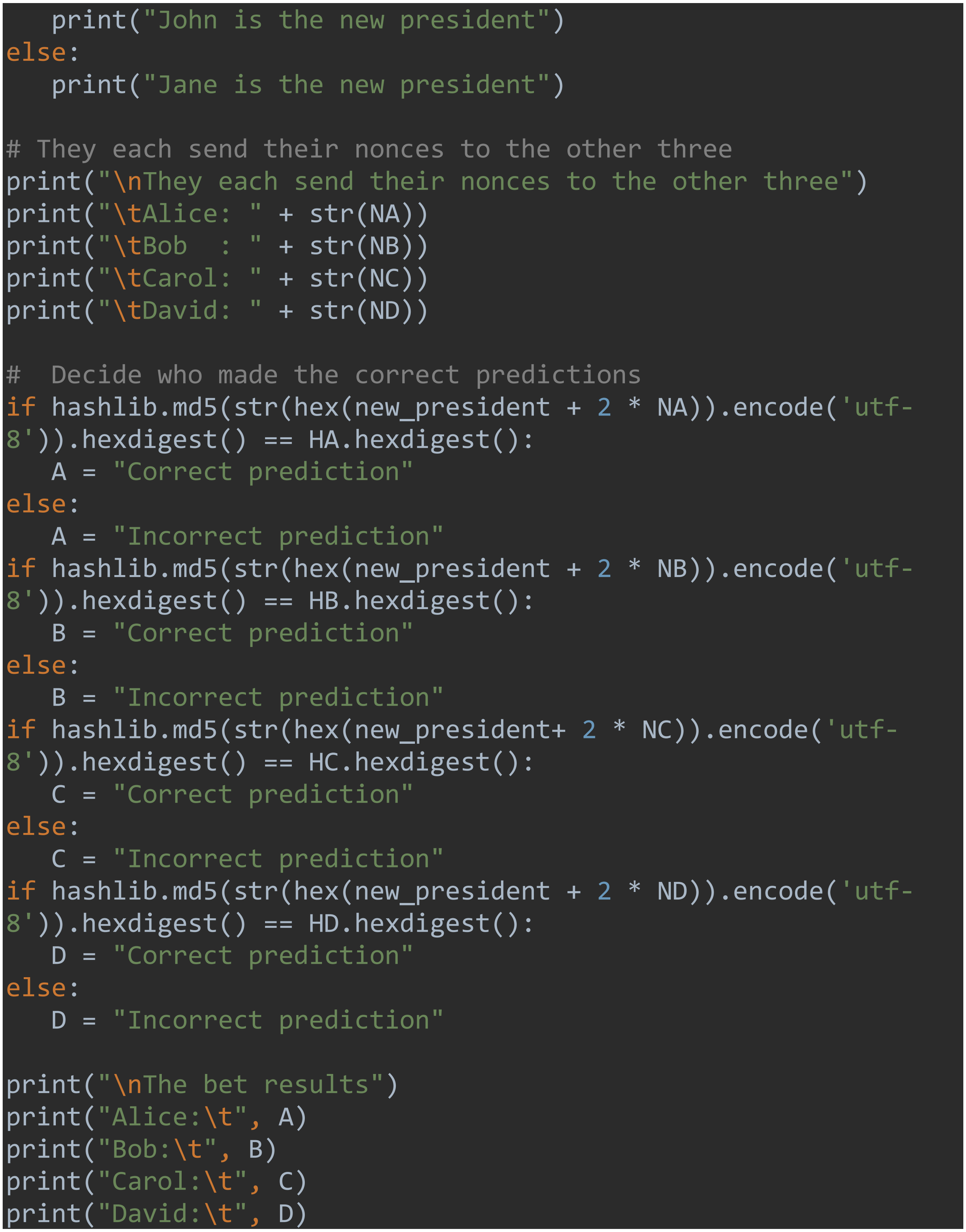
- Python 3.6+
- random and hashlib library installed

To run the code:

- Simply double click on the .py files or using terminal and use the `python ./hw_q2.py` `python ./hw_q3.py` commands

Source Code for Q2

```
import random
import hashlib

# Generate random nonces for the 4 friends
NA = random.randint(10,10000)
NB = random.randint(10,10000)
NC = random.randint(10,10000)
ND = random.randint(10,10000)

# Winner bit
john_bit = 0
jane_bit = 1

# Compute hashes  Hi = MD5(Predicted winner bit || Nonce)
# Given that Alice and Bob predicted “John” a
# Carol and David predicted “Jane”
HA = hashlib.md5(str(hex(john_bit + 2*NA)).encode('utf-8'))
HB = hashlib.md5(str(hex(john_bit + 2*NB)).encode('utf-8'))
HC = hashlib.md5(str(hex(jane_bit + 2*NC)).encode('utf-8'))
HD = hashlib.md5(str(hex(jane_bit + 2*ND)).encode('utf-8'))

print("Everyone's hashes")
print("\tAlice: " + HA.hexdigest())
print("\tBob  : " + HB.hexdigest())
print("\tCarol: " + HC.hexdigest())
print("\tDavid: " + HD.hexdigest())

# Randomly select a winner
print("\nThe election results have been declared")
new_president = random.randint(0,1)
if new_president == john_bit :
```



```
    print("John is the new president")
else:
    print("Jane is the new president")

# They each send their nonces to the other three
print("\nThey each send their nonces to the other three")
print("\tAlice: " + str(NA))
print("\tBob  : " + str(NB))
print("\tCarol: " + str(NC))
print("\tDavid: " + str(ND))

#  Decide who made the correct predictions
if hashlib.md5(str(hex(new_president + 2 * NA)).encode('utf-
8')).hexdigest() == HA.hexdigest():
    A = "Correct prediction"
else:
    A = "Incorrect prediction"
if hashlib.md5(str(hex(new_president + 2 * NB)).encode('utf-
8')).hexdigest() == HB.hexdigest():
    B = "Correct prediction"
else:
    B = "Incorrect prediction"
if hashlib.md5(str(hex(new_president+ 2 * NC)).encode('utf-
8')).hexdigest() == HC.hexdigest():
    C = "Correct prediction"
else:
    C = "Incorrect prediction"
if hashlib.md5(str(hex(new_president + 2 * ND)).encode('utf-
8')).hexdigest() == HD.hexdigest():
    D = "Correct prediction"
else:
    D = "Incorrect prediction"

print("\nThe bet results")
print("Alice:\t", A)
print("Bob:\t", B)
print("Carol:\t", C)
print("David:\t", D)
```

Source Code for Q3

```
import hashlib

# Given
N = 2357
username = "Alice"
password = "alice123"

# (i)
print("C sends its user name to S; it DOES NOT send its
password P")
print("C Username: ", username)

# (ii)
print("\nS verifies the user name and sends a nonce N to C")
print("Nouce: ", str(N))

# (iii)
hashed_password = hashlib.md5(str(hex(N) + password +
hex(N)).encode('utf-8'))
print("\nC computes a hash H = MD5(N||P||N) and sends H to S")
print("The hashed password(H) is ",
hashed_password.hexdigest())

#(iV)
g = hashlib.md5(str(hex(N) + password + hex(N)).encode('utf-
8'))
print("\nS independently computes G= MD5(N||P||N). If G=H, S
declares C as an authenticated user")
print("The hashed password(G) is ", g.hexdigest())

if g.hexdigest() == hashed_password.hexdigest():
   print("C is an  authenticated user")
else:
   print("C is NOT an  authenticated user")
```