Antonio Cebreiro Bernárdez
Seguridad de la Información

Lab 2

En este laboratorio se ha implementado en Python el Bit Commitment Protocol entre dos instancias. Este algoritmo es un mecanismo de intercambio de información seguro en cuanto a integridad se refiere pues ninguna de las partes puede repudiar lo que declara en el manifiesto entregado.

El algoritmo presenta una estructura lineal debido a la falta de tiempo, pero lo ideal sería la implementación mediante 2 partes lógicamente separadas. Por ejemplo, mediante sockets.

```python
import cryptography as cry
import sys
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms
#Preshared key
key = b"00000000000000000000000000000000"
nonce = b"0000000000000000"
algorithm = algorithms.ChaCha20(key,nonce)
cipher = Cipher(algorithm, mode=None)
encryptor = cipher.encryptor()
#commit function
def commit(s,r,b0):
    encryptor = cipher.encryptor()
    if b0 == 0:
        ct = encryptor.update(s)
        x = bin(int.from_bytes(ct,byteorder=sys.byteorder))
        return x
    ct = encryptor.update(s)
    x = int.from_bytes(ct, byteorder=sys.byteorder)
    r = int.from_bytes(r, byteorder=sys.byteorder)
    a = bin(x ^ r)
    return a
#Alice chooses a random r ∈ R and send r to Bob.
r = b"0"
print(r)
print()
#Bob chooses a random s ∈ S and computes c ← commit(s, r, b0)
b0 = 0
s = b"1"
c = commit(s,r,b0)
print(c)
#Bob sends (b0, s) to Alice, and she accepts the opening if c = commit(s, r, b0).
#bAlice = b0 # If b0 is not what Bob says bit commitment will fail.
bAlice =  not b0 # If b0 is what Bob says bit commitment will be succesfull.
cAlice = commit(s,r,bAlice)
print(cAlice)
if cAlice == c:
    print("Bit commitment succesfull")
else:
    print("Bit commitment failed")
```