Cryptography - Homework 3a

Samuel Breese

November 4, 2018

1 Problem

Implement Blum-Goldwasser probabilistic encryption algorithm with the following setup parameters

$$p = 499,\ q = 547,\ a = -57,\ b = 52,\ X_0 = 159201$$

The message m to be encrypted in binary is 1001110000010000l100.

2 What is the ciphertext?

We must know the modulus and public key $N = pq = 499 \times 547 = 272953$. Using our seed X_0, we must generate the least-significant bits of X_1 through X_{19} (since our message m is 19 bits long) use the Blum Blum Shub pseudo-random number generator. The following Haskell code does this:

```
data Natural = Succ Natural | Zero

bbs :: Integer -> Integer -> [Integer]
bbs x0 modulus = lsb . x <$> iterate Succ Zero
  where lsb :: Integer -> Integer
        lsb = flip rem 2
        x :: Natural -> Integer
        x Zero = x0
        x (Succ n) = let xn = x n in rem (xn * xn) modulus
```

Evaluating `take 20 $ bbs 159201 272953` yields `[1,1,0,1,0,0,0,1,0,0,0,0,0,0,1,1,0,1,0,0]`, which we can XOR with the message (as usual in a stream cipher) to obtain the encrypted message:

$C(m) = 10011100000100001100 \oplus 11010001000000110100 = 01001101000100111000$

We must also include $X_20 = X_0^{2^{20}} = 36858$ in the message.

3 Verify your answer by showing that $D(C(m)) = m$

To decrypt, we first compute $r_p = X_{20}^{\left(\frac{p+1}{4}\right)^{20}} \bmod p$ and $r_q = X_{20}^{\left(\frac{q+1}{4}\right)^{20}} \bmod q$, obtaining $r_p = 20$ and $r_q = 24$. From these, we can re-compute the seed:

$$X_0 = q \times b \times r_p + p \times a \times r_q \bmod N = 159201$$

Since this matches the original X_0, we can re-generate the stream of bits used to encrypt the message, and XOR those bits with the ciphertext to obtain the plaintext.