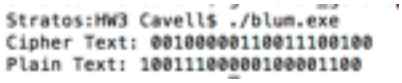
Cavell Teng
HW3 - Blum-Goldwasser

1.

Plain text = 10011100000100001100
Cipher text = 00100000110011100100

2.

Running through the program returns both the encrypted cipher text and the decrypted plain text.



First, Alice generates a public key using the given p and q values. Normally, the p and q values are random values generated such that they are prime numbers congruent to 3 mod 4. This public key is used by Bob (with seed value generated by him) to encrypt the messages that are sent to Alice. When Alice receives the encrypted value, she uses her private values, p and q, in addition to generate the initial seed value that Bob used from the final x value given with the cipher text.

In the encryption function, a plain text, randomly generated seed value, and public key are used. The plain text is split into equally sized groups of bits. An x value is generated using the $seed^2$ mod key. The least significant bits are XOR'd with the groups of bits. The resulting group of bits is the cipher text. This group, along with the final x value calculated are returned.

In the decryption function, the cipher text, the public key, p, q, a, b, and the final x value, are given. Using the p and q, the rp and rq values are obtained. These values are used as the exponent for the modular exponentiation of the final x value with the modulo of p and q for each operation. The two results are multiplied with a and p and b and q respectively before being summed and mod'd with the public key. This results in the initial seed used by Bob. Once this seed is obtained, the same process of generating x values and XORing is repeated with the cipher text. The result is the plain text.