# ECE 404 Homework 1

Elias Talcott

January 23, 2020

## Contents

## 1 Python Code

```
#!/usr/bin/env python

### Homework Number: 1
### Name: Elias Talcott
### ECN Login: etalcott
### Due Date: January 23, 2020

import sys
from BitVector import *

###
## Decrypt message given an encrypted file and a bitvector key
###

def cryptBreak(ciphertextFile, key_bv):

    # Initialize variables for decryption
    PassPhrase = "Hopes and dreams of a million years"
    BLOCKSIZE = 16
    numbytes = BLOCKSIZE // 8

    # Reduce the passphrase to a bit array of size BLOCKSIZE
    bv_iv = BitVector(bitlist = [0] * BLOCKSIZE)
    for i in range(0, len(PassPhrase) // numbytes):
        textstr = PassPhrase[i * numbytes : (i + 1) * numbytes]
        bv_iv ^= BitVector(textstring = textstr)

    # Create a bitvector from the ciphertext hex string
    fpin = open(ciphertextFile)
    encrypted_bv = BitVector(hexstring = fpin.read())

    # Create a bitvector for storing the decrypted plaintext bit array
    msg_decrypted_bv = BitVector(size = 0)

    # Carry out differential XORing of bit blocks and decryption
    previous_decrypted_block = bv_iv
    for i in range(0, len(encrypted_bv) // BLOCKSIZE):
        bv = encrypted_bv[i * BLOCKSIZE : (i + 1) * BLOCKSIZE]
        temp = bv.deep_copy()
        bv ^=  previous_decrypted_block
        previous_decrypted_block = temp
        bv ^=  key_bv
        msg_decrypted_bv += bv

    return msg_decrypted_bv.get_text_from_bitvector()
```

```
###
## Brute force all possible possible key_bv values
###

if __name__ == "__main__":
    # Check and separate arguments
    if len(sys.argv) != 2:
        sys.exit("Wrong arguments!")
    _, infile = sys.argv

    # Test all 16-bit bitvector keys and check for Mark Twain quote
    for val in range(24000, 2 ** 16):
        key = BitVector(intVal = val, size = 16)
        decryptedMessage = cryptBreak(infile, key)
        if "Mark Twain" in decryptedMessage:
            print("\nKey: {}".format(val))
            print("\n{}".format(decryptedMessage))
            break
```

## 2 Plaintext Quote and Key

Using the function in cryptBreak.py, I was able to find the following secret message and key with a brute force attack.

**Secret Message:**

It is my belief that nearly any invented quotation, played with confidence, stands a good chance to deceive.

- Mark Twain

**Secret Key:** 25202

## 3 Code Explanation

The decryption algorithm in this problem uses the method of differential XORing to protect against the statistical attack. In this decryption case, differential XORing works by XORing each block with the previously decrypted block and then XORing that result with the decryption key. The resultant block is then added to the decrypted message. Since the first block does not have a previous block to be XORed with, we use a pass phrase to generate a stand-in previous block. This means that in order to successfully decrypt a message, one needs to know both the pass phrase and the decryption key.

In this problem, we are given the pass phrase "Hopes and dreams of a million years" and have to discover the decryption key using a brute force attack. This attack is carried out by iterating through every key in the key space, which has $2^{16} = 65,536$ possibilities. Using the decryption algorithm given in the lecture notes, it takes about 11 seconds to process 1,000 keys, meaning that the brute force attack would at most take $\sim$ 12 minutes and on average $\sim$ 6 minutes. Since the key was found to be 25,202, the attack took just under 5 minutes to complete in this case.

$e = \int\limits_{0}^{\infty} mc^2$