Assignment 1: Codes, Number Theory, and Symmetric Encryption
Cryptography and Secure Communications (MITS 5500)
Prepared by: Miguel V. Martin, PhD, PEng
Due: October 2, 2020 by 11:59 pm

This is an individual assignment; collaboration is not allowed.

1. Unconventional Cryptosystem [10%]. You intercept the following sequence of characters from a suspicious channel. Not sure whether the message is in English or not. What's the message? Explain how you broke it. Note: A solution to this problem will not be provided after the due date.

```
)90)  (225(  )90)  (225>  )360  <45)  )315)  )315)  (135(  )90)
(135>  <45)  )315)  )90)  <315)  (225>  225(  270(  (225(  135(
(225(  <360)
```

Submit your answer to this question in a 1-page Word/PDF document.

---

2. Number Theory and cryptography.io [90%]. Submit a Python program called `assignment1.py` based on SymPy with the following menu:

A. Primality Test using Miller-Rabin
B. Chinese Reminder Theorem
C. Symmetric Encryption
D. Exit

The behaviour of each of these menu options should be as follows:

A. Miller-Rabin [30%]. Under this option, the program takes as input from the console an integer number $n < 10{,}000$, and uses the Miller-Rabin algorithm, as described in the course notes, to determine whether n is prime or not. The program should output the result of every iteration of the loop in the Miller-Rabin algorithm, and finally whether the number is prime or not. Of course, SymPy has the `mr` function available but this assignment is asking you to code the algorithm from scratch.

B. Chinese Reminder Theorem [30%]. Under this option, the program takes as input from the keyboard positive integers a, b, and c, and also pairwise relatively prime positive integers r, s, and t (i.e., $\gcd(r, s) = \gcd(s, t) = \gcd(r, t) = 1$). The program should find the value of x that satisfies a system of congruencies of the form:

© 2020 Miguel V. Martin

$$
\begin{aligned}
x &\equiv a \pmod{r}\\
x &\equiv b \pmod{s}\\
x &\equiv c \pmod{t}
\end{aligned}
$$

You may use SymPy functions such as `crt` and others.

C. Symmetric Encryption in Python [30%]. Under this option, the program takes as input from the keyboard a plaintext message and encrypts the message using either AES or 3DES encryption. The program automatically generates the necessary key and outputs the corresponding ciphertext on the screen along with its decryption back to the original plaintext and the key used for encryption. Feel free to use example code from: https://cryptography.io/en/latest/hazmat/primitives/symmetric-encryption/.

© 2020 Miguel V. Martin