CIS5371 – Cryptography **Fall 2025**

Homework 1: Deadline Tuesday 9/23

Instructor: Viet Tung Hoang

Recall that your solution must be typed via Latex.

1. (**Reuse of one-time pad**) (50 points) In this problem all characters are represented as 8-bit bytes with the usual US-ASCII encoding (e.g. "A" is encoded as 0x41). Here are two 8-character English words (that you can find in a dictionary such that /usr/share/dict/words in Unix) encrypted with a "one-time pad". Decide whether they were encrypted with the same pad or with different pads. If they are different pads, then explain why they cannot be the same pad. If they are the same pad, then decrypt the ciphertexts. You don't have to submit your code.

   ```
   e9 3a e9 c5 fc 73 55 d5
   f4 3a fe c7 e1 68 4a df
   ```

2. (**General one-time pad**) (60 points) Let $N, n > 0$ be integers such that $N > 2^n$. Alice and Bob share a secret key K that is chosen uniformly from $\{0, 1, \ldots, N-1\}$. Alice now wishes to send a secret message $M \in \{0,1\}^n$ to Bob by saying something aloud; the message M has no relation with the key K. Eavesdropper Eve is listening in: she hears everything Alice says, but doesn't know the key.

   a) (30 points) Describe a (**very simple**) encryption scheme so that Alice can communicate M to Bob and achieve perfect secrecy. In particular, you need to specify both the encryption and decryption algorithms.

   b) (30 points) Prove that your scheme indeed achieves perfect secrecy by showing that for any fixed message M, if we pick K uniformly from $\{0, 1, \ldots, N-1\}$ then the corresponding ciphertext will be uniformly distributed over the ciphertext space.

   Note: A common error is to parse K into an n-bit string L and encrypt via $C \leftarrow L \oplus M$. This doesn't work. For example, consider $n = 1$ and $N = 3$, and let's say we get $L \leftarrow K \bmod 2$. Fix $M = 0$. Since $K \xleftarrow{\$} \{0, 1, 2\}$, the string L is 1 with probability 1/3, and 0 with probability 2/3. Then $C = 1$ with probability 1/3, and $C = 0$ with probability 2/3. In other words, the distribution of the ciphertext is *not* uniform, and thus this scheme fails to achieve perfect secrecy.

3. (**Breaking substitution cipher**) (70 points) In this problem, you will write a program to implement the Monte Carlo method of breaking substitution cipher. Your program will take as input a file that contains a ciphertext, and then output the plaintext. **Make sure that your program can run in `linprog`.**

   In the course webpage, you will also find two files "`testInput.txt`" and "`testOutput.txt`". They are a sample input file and the corresponding output file.

   Some implementation details. Recall that you need bigram (two-letter) frequencies of English. In the course website, you'll find a file "`war-and-peace.txt`" that contains (a lower-case version) of the English translation of the novel War and Peace. Compute bigram frequencies based on that file and store it in a file "`bigram.txt`". Your submission code should just read the file `bigram.txt` instead of re-computing it.

   Deliverables. You need to upload to Canvas a zip file containing your source code (and the file `bigram.txt`). This includes a `README.txt` that informs me how I should run the program. There's no

restriction on your choice of programming language. In the past, people often used Python because it was easy to implement, and it was fast enough. Still, make sure that your program is fast enough to finish 10,000 iterations on the test input within a few minutes.

After each iteration, your program should print out the iteration number and the best decrypted message that you've found so far. I'll terminate your program after 10,000 iterations or when the correct message is found. Since the algorithm is probabilistic, I'll run your program up to 3 times.