

# XOR Crypto

Write a **Go language** program that implements the XOR crypto method. This is a group programming assignment (i.e., **only one submission per group is needed**).

The method takes a message,  $m$  (either plaintext or ciphertext), of size  $b$  bytes and a key,  $k$ , also of size  $b$  bytes (i.e., they are exactly the same size). Each bit of  $m$  is XOR'd with each bit of  $k$ , one bit at a time. In practice, we often use a buffer of some size (e.g., 4,096 bytes or 4KB) and XOR a group of bits together for better performance.

## Requirements:

- Submit your source code only;
- Read the key from a file named *key* in the current directory (make sure that this works on Linux; i.e., don't use Windows-specific directory separators);
- Read the plaintext/ciphertext from **stdin**; and
- Send generated output (either plaintext or ciphertext) to **stdout**.

Please, no GUIs. Make this a command line application without frills that I can execute at the command line; for example, as follows: `./xor < plaintext > ciphertext`. This would take the contents of the file *plaintext*, XOR (encrypt) it with the contents of the file *key*, and store the resulting ciphertext to the file *ciphertext*.

The reverse: `./xor < ciphertext` would take the contents of the file *ciphertext*, XOR (decrypt) it with the contents of the file *key*, and send the resulting plaintext to **stdout**.

## Hints:

- Consider reading/writing all the data as binary data
- Consider an application of this program where the key is NOT the same size as the message (what would you do?)