HW1 Lecture Notes

CS 24000 L04 Week 2

HW1 Problem Explanation

- Implement Julius Caesar's ROT-13 encryption scheme
- Generalize to ROT-n encryption
- Add an xor checksum function

ROT-13/ROT-n Encryption

- Given a message, g_msg, add **n** to each individual letter
  - If a letter goes past ‘z’, subtract 26 to wrap back around
  - n is defined to be positive or 0
- Save this new, encrypted message to g_cip

encrypt(13, 0, 12)

g_msg	h	e	l	l	o	,		w	o	r	l	d
g_cip	u	r	y	y	b	,		j	b	e	y	d

XOR Checksum

- Start with a checksum value equal to zero
- Loop over g_msg
  - For each character in g_msg, xor the value of checksum with the character

h	e	l	l	o	,		w	o	r	l	d
0x68	0x65	0x6c	0x6c	0x6f	0x2c	0x20	0x77	0x6f	0x72	0x6c	0x64

0x68 ^ 0x65 ^ 0x6c ^ 0x6c ^ 0x6f ^ 0x2c ^ 0x20 ^ 0x77 ^ 0x6f ^ 0x72 ^ 0x6c ^ 0x64
= 0x0c (12)

Important Notes (AKA Hints)

- ARRAY_SIZE is a global defining the length of the array
- n, l, and r are arguments to encode
  - n for defining ROT-N
  - l for left (aka start) and r for right (aka end)
- g_msg holds the original message, do not touch
- g_cip holds the ciphertext (the encrypted message)