ANUC 1100 – Introduction to Programming and Algorithms

Semester 1, 2018

Assignment 2

Due date: 06:00pm, 1 June, 2018

This assignment will be marked out of 15. It will count for 15% of the final grade. Below you will find the specifications.

Instructions:

- All functions you developed for this assignment must be included in one single .hs file and submitted to the assignment dropbox link on Wattle by 06:00pm, 1 June, 2018. Make sure you include your name and student ID in the comment section at the top of your source file. If you have any other supporting documents, you can zip all files into a single archive Assignment_1_<name>_<ID>.zip where <name> and <ID> are your details.
- Late submission without an extension are penalised at the rate of 5% of the possible marks available per working day or part thereof. The assignment is not accepted after 10 working days after the due date.
- Plagiarism will attract academic penalties in accordance with the ANU guidelines.

Good luck and enjoy the time you will spend on this assignment

Implementation of encryption/decryption algorithms

1. Introduction to cryptography

The fundamental objective of cryptography is to allow two people, e.g., Alice and Bob, to communicate securely over an insecure (public) channel in a way that a third person, e.g., Osca, is unable to understand what is being exchanged.

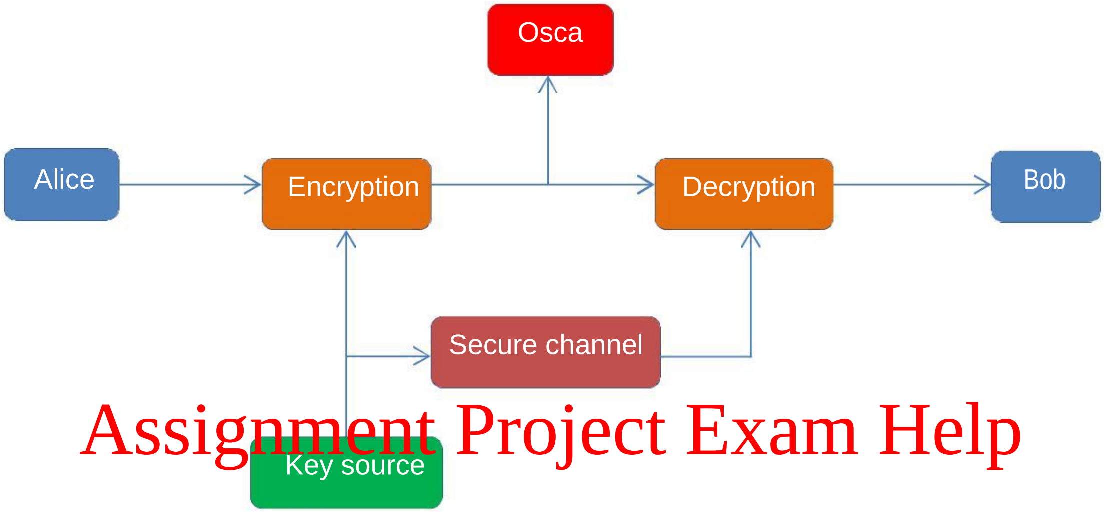

Figure 1: Encryption process

The most popular method is cryptography which employs encryption to convert a plaintext into an incomprehensible version, namely cipher text. The reverse process which converts encrypted messages into understandable versions is called decryptions.

1.1. Classic cryptography

This old-styled cryptography is mainly based on either the rearrangement of order of letters in a message (transposition ciphers) or the replacement of letters or groups of letters by other letters or groups of letters (substitution ciphers). An early substitution cipher was the Ceasar cipher which replaces each letter in a message by another letter at a predefined position (known as a key) in the alphabet.

1.2. Modern cryptography

The development of digital computers and electronics has made it possible to implement more complex encryption of any kind of data representable in binary format.

Two well-known fields of modern cryptography are Symmetric-key and Asymmetric-key (also known as Public-key). Symmetric-key cryptography refers to encryption methods in which both sender, Alice, and receiver, Bob, are having a same key for encrypting and decrypting messages. Symmetric-key cryptography can be implemented as either stream ciphers or block ciphers.

Block ciphers encrypt inputs as blocks of characters or blocks of bits. The most popular block cipher standards are Data Encryption Standard (DES) and Advanced Encryption Standard (AES).

2. **Assignment task**: **Ceasar Cipher program** (15 marks)

Ceasar cipher is a substitution cipher in which each letter in the plaintext is shifted to a certain position in the alphabet to become a different letter. The shifted position is a number which is called a key, k.

For example:

- Plaintext: “This is the plaintext!”
- K=3
- Ciphertext: “Wklv lv wkh sodlqwhaw!”

The Ceasar cipher only applied to the set of 26 alphabetic letters, ‘a’ .. ‘z’. Mathematically, each letter is firstly bound to a number, i.e., ‘a’ = 0, ‘b’ = 1, .., ‘z’ = 25. Implementation of the Ceasar encryption is as following:

E(x) = (x + k) mod 26, where x is the number corresponding to a letter and k is the key.

Decryption is a reverse process of encryption. That is: D(x) = (E(x) – k)) mod 26

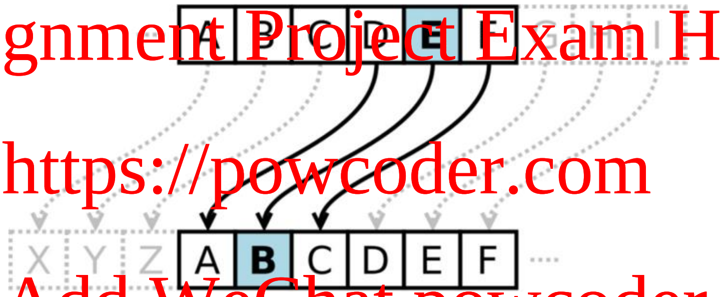

Figure 2: Ceasar cipher (Wikipedia)

You are required to:

Write a Haskell program to implement extended Ceasar cipher applicable to all characters in the ASCII 128 table instead of 26 alphabetic letters. For example:

- Plaintext: “This is the plaintext!”
- K=3
- Ciphertext: "Wklv#lv#wkh#sodlqwh{w$".

The difference in here is all spaces and other characters, such as ‘!’ are also encrypted

Notes:

1. Names used in your program must be meaningful.
2. Submit a file which can be compiled.
3. Include at least 3 test cases in your .hs file which prints the test results.