Computer Security
Programming Assignment 2

Due: 10 April 2018

Overview

In this project you have to implement the DES (Data Encryption Standard) cryptography system. DES is the symmetric key block cipher algorithm designed by IBM in 1977. You are required to implement the encryption and decryption functions, including the DES round function, key schedule, cipher-block-chaining, permutation, etc.

You should write one program which facilitates both encryption and decryption; the program must accept the desired operation, input file, and key as inputs. You may either use command line arguments or interactive prompts to accept these parameters.

This project can be done either individually or in groups of two. Your code from this project will need to be integrated into the next project, so try to implement it properly to save you time in the future. As always, plagiarism and cheating will not be tolerated.

Requirements

- Use either C or C++ for this assignment.
- Your program must be able to compile and run on the machines in the CS department labs.
- Do not use any library functions (e.g., from openssl) to implement DES. You must implement the cipher from scratch.
- Your implementation should be compatible with openssl's DES-CBC implementation. That is, your program should be able to decrypt a ciphertext generated by openssl, and vice-versa.
- You should allow the user to supply their own key and IV, or generate random ones if they are not provided.
- The program must notify the user of any errors (e.g., missing input files) and fail gracefully.
- Code should be well-commented and modular.
- Include a Readme which describes how to use all of the features of your program, and a Makefile to compile your code.
- Your program should validate in valgrind (i.e., there should be no memory errors).
- You are to submit a tarball (.tar, .tgz) to Canvas including your code, the Readme, and the Makefile. Name the file `groupmember1name-groupmember2name.tar` (or .tgz).

Notes, hints, and resources

- You have been provided with a copy of the tables required by DES; i.e., the expansion box, substitution boxes, permutation boxes, etc. You may use these versions or find/implement your own, however the program must remain compatible with the DES standard as implemented by openssl.

- A DES key is 56 bits, but is traditionally stored as 64 bits (8 bytes); each byte contains one parity bit in addition to 7 bits of the actual key. You are not required to implement calculation or checking of these parity bits, however you should format the key with space for them to ensure compatibility with openssl.
- You should accept inputs for the key and IV in hexadecimal format.
- The openssl utilities typically use their own file format for storing a ciphertext along with its IV. You are not required to implement this file format. Instead, you can supply certain flags to openssl in order for it to read and produce plain binary ciphertexts:

  Encryption: `openssl enc -e -des-cbc -nosalt -K <hex key> -iv <hex iv>`
  Decryption: `openssl enc -d -des-cbc -nopad -nosalt -K <hex key> -iv <hex iv>`

  When using the utility in this way, input and output can be redirected using the `<` and `>` operators provided by the shell.
- Documentation on openssl's symmetric cipher commands can be found here:
  `https://www.openssl.org/docs/apps/enc.html`
- The FIPS standard for DES can be found here:
  `http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf`
- Wikipedia provides a decent overview of the DES algorithm in more concise terms than the FIPS document:
  `http://en.wikipedia.org/wiki/Data_Encryption_Standard#Description`