

Alexander Frederick

Data Structures and Algorithms II

Project 1

User's Manual

Setup and Compilation

1. Download and unzip the submission from eLearning on a Linux box in the multi-platform lab.
2. The submission includes:
 - main.cpp (main driver program)
 - HashTable.cpp and HashTable.h (hash table implementation with chaining)
 - List.cpp and List.h (linked list for collision resolution)
 - Node.cpp and Node.h (node structure for storing user credentials)
 - encryption.cpp and encryption.h (Vigenère cipher implementation)
 - passwords.cpp and passwords.h (random password generation)
 - filemaker.cpp and filemaker.h (file creation utilities)
 - tokenizer.cpp and tokenizer.h (file parsing utilities)
 - names.txt
 - UsersManual.pdf
 - UMLpng
1. **Environment:** This program has been tested in the multi-platform lab and will run there.
2. **Compiling:** This program includes a Makefile. At the command line in Linux, type **make**. The program produces an executable entitled **main**

Running the program: Be sure names.txt is in the same directory as the executable. Issue the command **./main**. No command line arguments are required or checked.

Note: *The program uses a fixed encryption key ("riddle") for Vigenère cipher encryption. This key is hardcoded in main.cpp.*

Program Description

This program demonstrates a secure password authentication system using the following components:

- **Password Generation:** Creates random 9-character passwords using lowercase letters (a-z).
- **Vigenère Cipher Encryption:** Encrypts passwords using a repeating key ("riddle") to protect stored credentials.
- **Hash Table Storage:** Stores username-password pairs in a hash table with chaining (using linked lists) for collision resolution.
- **Authentication Testing:** Tests both legal passwords (correct) and illegal passwords (modified) to verify the encryption and retrieval system.

The program creates two files during execution:

- **rawdata.txt:** Contains username-password pairs in plaintext
- **encrypteddata.txt:** Contains username-encrypted password pairs

User Input

No user interaction with the program is required. The program automatically:

1. Reads usernames from names.txt
2. Generates random passwords for each user
3. Encrypts the passwords using Vigenère cipher
4. Populates the hash table with encrypted credentials
5. Performs authentication tests on selected users

Output

All output goes to the console. The program displays two test sections:

Legal Password Testing

Tests passwords that should match (correct credentials). Output format:

Legal:

Userid	Password(file)	Password(table/un)	Result
alice	xyznopqrs	xyznopqrs	match
charlie	abcdefghi	abcdefghi	match

Illegal Password Testing

Tests modified passwords that should not match (incorrect credentials). The first letter of each password is changed to 'z'. Output format:

Illegal:

Userid	Password(mod)	Password(table)	Result
alice	zyznopqrs	xyznopqrs	no match
charlie	zbcdefghi	abcdefghi	no match

Note: The encrypted password values shown will vary each time the program runs due to random password generation. This may result in a match on the illegal password test. Run again and it is likely to not match again. This best fit the project sample output.