lab2

AUTHOR
Version
Sun Oct 20 2024

Table of contents

Main Page

Utility for encrypting and decrypting files using the Triple DES (3DES) algorithm. A user-provided password is converted into a key and initialization vector (IV) for 3DES.

Author

Ushakov Aleksandr (anushakov@ispras.ru)

Date

10/20/2024

Version

1.0

3DES File Encryption/Decryption Utility

This command-line utility allows you to encrypt and decrypt files using the Triple DES (3DES) algorithm in CBC mode. The cryptographic key and initialization vector (IV) are derived from a user-provided password using OpenSSL's key derivation functions.

Features

Encryption : Encrypt files using the 3DES (Triple DES) algorithm in CBC mode.
Decryption : Decrypt previously encrypted files.
Password-Based Key Derivation : The key and IV are derived securely from the user's password using SHA-256.

Prerequisites

OpenSSL development libraries (`libssl-dev`)
CMake
A C compiler (e.g., `gcc`)

Make sure OpenSSL is installed on your system. If not, you can install it using a package manager. For example:

On Ubuntu/Debian:

```
sudo apt-get install gcc libssl-dev cmake
```

On Fedora:

```
sudo dnf install gcc openssl-devel cmake
```

On macOS (with Homebrew):

```
brew install gcc openssl cmake
```

Building the Utility

To compile the program, follow these steps:

1. Clone the Repository (if applicable):

```
git clone https://git.miem.hse.ru/anushakov/lab2.git
cd lab2
```

1. Run CMake: this command generates the necessary build files based on the CMakeLists.txt configuration:

```
cmake -S . -B build
```

1. Build program:

```
cd build
make
```

Usage

The utility takes several command-line arguments to specify whether you want to encrypt or decrypt a file, along with input, output, and password parameters.

```
./des [-e|-d] -i input -o output -p password
```

Options:

-e : Encrypt the input file
-d : Decrypt the input file
-i : Path to the input file
-o : Path to the output file
-p : Password to use for encryption or decryption

Example Usage

Encrypt a File

To encrypt a file named plaintext.txt and output it as encrypted.dat:

```
./des -e -i plaintext.txt -o encrypted.dat -p your_password
```

Decrypt a File

To decrypt the encrypted.dat file back to decrypted.txt:

```
./des -d -i encrypted.dat -o decrypted.txt -p your_password
```

Contributing

Contributions are welcome! Please feel free to submit issues and pull requests.

Licensing and distribution

Utility is distributed under the `Apache License, Version 2.0`.

Acknowledgements

This utility uses the OpenSSL library for cryptographic operations.

File Index

File List

Here is a list of all files with brief descriptions:

File Documentation

main.c File Reference

```
#include <openssl/evp.h>
#include <getopt.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Macros

#define **BUF_SIZE** 1024*1024

Functions

int **generate_key_iv** (const char *password, unsigned char *key, unsigned char *iv)

Generates a key and initialization vector (IV) based on a password.

void **encrypt_file** (const char *input_file, const char *output_file, const char *password)

Encrypts the input file using the 3DES algorithm in CBC mode.

void **decrypt_file** (const char *input_file, const char *output_file, const char *password)

Decrypts the input file using the 3DES algorithm in CBC mode.

void **print_usage** (const char *prog_name)

Prints the usage instructions for the program.

int **main** (int argc, char **argv)

The main function of the program.

Macro Definition Documentation

#define BUF_SIZE 1024*1024

Function Documentation

void decrypt_file (const char * *input_file*, const char * *output_file*, const char * *password*)

Decrypts the input file using the 3DES algorithm in CBC mode.

Parameters

in	*input_file*	Name of the input file to decrypt.
in	*output_file*	Name of the output file where the decrypted data will be written.
in	*password*	The password used to generate the key and IV.

void encrypt_file (const char * *input_file*, const char * *output_file*, const char * *password*)

Encrypts the input file using the 3DES algorithm in CBC mode.

Parameters

in	*input_file*	Name of the input file to encrypt.
in	*output_file*	Name of the output file where the encrypted data will be written.
in	*password*	The password used to generate the key and IV.

int generate_key_iv (const char * *password*, unsigned char * *key*, unsigned char * *iv*)

Generates a key and initialization vector (IV) based on a password.

The key is derived by applying a password-based key derivation function (PBKDF).
A salt can be optionally provided, but in this implementation, it is not used (set to NULL).

Parameters

in	*password*	A string containing the password.
out	*key*	Buffer to store the generated key.
out	*iv*	Buffer to store the generated initialization vector (IV).

Returns

1 on success, exits the program with an error message on failure.

int main (int *argc*, char ** *argv*)

The main function of the program.

Processes command-line arguments and runs either encryption or decryption.

Parameters

in	*argc*	The number of command-line arguments.
in	*argv*	Array of command-line arguments.

Returns

0 on successful execution of the program.

void print_usage (const char * *prog_name*)

Prints the usage instructions for the program.

Parameters

in	*prog_name*	The name of the program invoked in the command line.

README.md File Reference

Index

INDEX