Лабораторные работы по курсу
Защита информации (часть 1).

МГТУ им. Баумана
Кафедра ИУ-9
Юдаев П.В., 2014

Лабораторная работа № XX.
Программа шифрования файлов на диске filecrypt.

Проект лабораторной работы.

1 Задание

1.1 Основные требования

1.1.1. Программа запускается с системного диска, имеющего свободное место, и шифрует диск пользователя. Программа заменяет содержимое файлов на их шифротексты и наоборот. При этом размер файлов не изменяется.

1.1.2. Программа не позволяет злоумышленнику изменить содержимое зашифрованного файла. Если замечено изменение содержимого, файлы не расшифровываются. Подробней см. ниже в списке ошибочных ситуаций.

1.1.3. Программа может запускаться разными пользователями.
Можно добавлять и удалять пользователей программы.
У каждого пользователя свое имя и пароль.
Пароль имеет длину не менее 8 символов и содержит как минимум по 1 символу каждого типа:
- заглавные буквы A-Z
- строчные буквы a-z
- цифры 0-9

Программа должна препятствовать утечке или подбору пароля.

1.1.4. Шифр должен быть семантически стойким к атаке с выбором открытого текста. В частности, два файла с одинаковым содержимым должны преобразовываться в разные шифротексты. Два одинаковых фрагмента одного файла должны преобразовываться в разные фрагменты шифротекста.

1.1.5. Исполнимый файл (или скрипт) имеет имя filecrypt.
Запуск из командной строки:

`filecrypt -e|-d|-a|-r -u username -p password [filename1 filename2 .. filenameN]`
`-e` зашифровать файлы
`-d` расшифровать файлы
`-a` добавить пользователя
`-r` удалить пользователя

Пример использования программы:
флэшка заполнена несколькими файлами полностью. Зашифровать все или некоторые из этих файлов. Расшифровать их.

1.2 Прочие требования

1.2.1. В рамках данной лабораторной, чтобы программа не требовала права `root`, все пользователи программы запускают ее из-под одного аккаунта системы. Также программа всегда запускается из той папки, где размещен ее бинарный файл. Она хранит все необходимые данные в этой же папке.

1.2.2. В рамках данной лабораторной программа не обязана пользоваться низкоуровневыми вызовами, например, определять позицию файла на диске.

Факультативные сведения.
Если программа пользуется номером сектора t, то можно реализовать **Tweakable encryption** одним из способов.
а) $c := E_{tweak}(k, t, x) := E(E(k, t), x)$ - каждый сектор шифруется со своим ключом. Или
б) $c := E(k_1, x \oplus H(t)) \oplus H(t)$. Или
в) Конструкция XTS: $N := E(k_2, t)$, где t - сектор начала файла. i - номер блока внутри файла. $c := E(k_1, x \oplus H(N||i)) \oplus H(N||i)$. В 2 раза меньше операций шифрования, чем в варианте (а).
Утверждение: если шифр E стойкий к атакам с выбранным открытым текстом, то XTS - тоже.
Без доказательства. Обычно $E = AES$, тогда шифр называется XTS-AES.

1.2.3. Программа запускается с системного диска. Она может сохранять необходимую информацию на системном диске в фиксированной директории.
При этом объем хранимой информации должен быть минимальным.
Для каждого вида данных потребуется обосновать, почему его нужно хранить. Сжимать хранимые данные не требуется.

1.2.4. Утечка хранимых на диске данных не должна позволять злоумышленнику авторизоваться в программе или получить возможность расшифровать или подменить зашиф-

рованные файлы.

1.2.5. Для простоты, допускается, что утечка хранимых на системном диске данных может позволить злоумышленнику получить список имен пользователей.

1.2.6. Некоторые способы хранения данных на диске:
- текстовые файлы
- база данных, например, sqlite или MySQL.

1.3 Работа программы при отсутствии ошибок

1.3.1. Добавление пользователя
`filecrypt -a -u user_name -p Password1`
Сначала проверить, свободно ли это имя пользователя, потом проверить пароль, потом добавить пользователя.
Код возврата: 0 (инструкция, аналогичная `return 0;`)
Печатает в stdout: `Added user user_name`

1.3.2. Удаление пользователя
`filecrypt -r -u user_name -p Password1`
Авторизовать пользователя и удалить его.
Код возврата: 0
Печатает в stdout: `Removed user user_name`

1.3.3. Шифрование файлов
`filecrypt -e -u user_name -p Password1 filename1 filename2 filename3`
Сначала авторизовать пользователя, потом проверить наличие всех файлов, потом зашифровать их.
Код возврата: 0
Печатает в stdout: `Encrypted files: filename1 filename2 filename3`
Порядок имен файлов в stdout тот же, что в командной строке.

1.3.4. Расшифрование файлов
`filecrypt -e -u user_name -p Password1 filename1 filename2 filename3`
Сначала авторизовать пользователя, потом проверить наличие всех файлов, потом неизменность содержимого всех файлов, потом расшифровать их.
Код возврата: 0
Печатает в stdout: `Decrypted files: filename1 filename2 filename3`
Порядок имен файлов в stdout тот же, что в командной строке.

1.4 Обработка ошибочных ситуаций при работе программы

1.4.1. Ошибка командной строки

Параметры командной строки заданы неверно.

Код возврата: 1 (инструкция `return 1;` или аналогичная).

Программа печатает в stderr: Bad command line parameters

Пример:

```
filecrypt -y something
Bad command line parameters
```

1.4.2. Ошибка авторизации

При авторизации пользователя такой пользователь не найден или пароль не верный. Не должна раскрываться информация о том, существует этот пользователь или нет, в том числе по побочным каналам.

Код возврата: 2 или 3.

Программа печатает в stderr: Bad credentials

Пример:

```
filecrypt -e -u petya -p wrong_password file1
Bad credentials
```

1.4.3. Ошибка создания пользователя

А. При добавлении пользователя такой пользователь уже существует.

Код возврата: 4.

Программа печатает в stderr: User already exists

Пример:

```
filecrypt -a -u existing_user -p password
User already exists
```

Б. Такого пользователя нет, но пароль не удовлетворяет требованиям.

Код возврата: 5.

Программа печатает в stderr: Too short or weak password

Пример:

```
filecrypt -a -u new_user -p password
Too short or weak password
```

1.4.4. Файлы не найдены

При шифровании или расшифровании один или несколько файлов не найдены. Программа не шифрует (не расшифровывает) ни один файл.

Код возврата: 6.

Программа печатает в stderr: Files not found: <список только не найденных файлов через пробел в том порядке, как в командной строке>

Пример: если не найден файл file2:

```
filecrypt -d -u user -p password file1 file2 file3 file4
```

```
Files not found: file2
```

1.4.5. Изменение содержимого зашифрованных файлов
При запуске программы с ключом `-d` обнаружена подмена содержимого хотя бы одного файла из перечисленных в командной строке. Программа не расшифровывает ни один файл.
Код возврата: 7.
Программа печатает в stderr: Tampering detected: <список только подмененных файлов через пробел в том порядке, как в командной строке>
Пример: если изменены файлы file2 и file4:

```
filecrypt -d -u user -p password file1 file2 file3 file4
Tampering detected: file2 file4
```

2 Рекомендации по выполнению задания

1. Выбрать, в паре с кем будете выполнять работу и язык программирования. Сообщить это преподавателю.
2. Выбрать способ авторизации и хранения имен пользователей и паролей на диске.
3. Выбрать способ шифрования данных. Объем файла не должен меняться.
4. Определить, какая дополнительная информация должна сохраняться на диске при шифровании файлов. Выбрать, как ее хранить.
6. Обратиться к преподавателю (все авторы работы вместе). Описать выбранную архитектуру и обосновать ее. Использовать функции имеющихся библиотек, а не писать свои, везде где возможно. Получить одобрение выбранной архитектуры.
7. Запрограммировать в соответствии с принятыми ранее решениями.
8. Протестировать программу самостоятельно. Обратить внимание на точное соответствие кодов возврата и возвращаемых сообщений при правильной работе и при обработке ошибок.
9. Сдать работу преподавателю.

3 Литература к лабораторной работе

См. лекции про

- хранение паролей
- блочный шифр AES и другие шифры
- режимы использования блочных шифров
- криптографические хэш функции
- коды аутентификации сообщения (MAC), особенно HMAC