УДК 004. 04

ШИФРОВАНИЕ ДАННЫХ НА БАЗЕ CRYPTO API

Курьянович Д. Ю.

Научный руководитель – Разоренов Н.А., к.т.н., доцент

Цель работы – разработать программу шифрования и дешифрования данных файла с генерацией ключа по алгоритму RC4.

С каждым годом все активнее развиваются информационные технологии, а информация приобретает все большую необходимость в защите. На смену бумажным документам пришли электронные, а личные контакты все чаще уступают место переписке по e-mail. Поэтому «шпионские штучки» вроде паролей и шифровок становятся все более привычными и необходимыми инструментами безопасности.

Развитие цифровизации резко усилило потребность в надежных алгоритмах шифрования для защиты конфиденциальной информации.Для обеспечения безопасности при передаче информации по открытым каналам связи необходимо соблюдение трех требований: конфиденциальность; аутентификация; целостность.

Шифрование — это преобразование данных в вид, недоступный для чтения без соответствующей информации (ключа шифрования). Задача состоит в том, чтобы обеспечить конфиденциальность, скрыв информацию от лиц, которым она не предназначена, даже если они имеют доступ к зашифрованным данным.

Шифрование информации гарантирует: недоступность информации для сторонних лиц, подлинность информации (то есть информации поступит к вам в неискаженном виде), целостность информации (данные, которые вы хотите передать останутся целыми в процессе передачи) [1].

Современные операционные системы Microsoft содержат множество криптографических подсистем различного назначения как прикладного уровня, так и уровня ядра. Ключевую роль в Windows играет интерфейс Crypto API (Cryptography API, CAPI) [2].

Функциональную часть разработанного на C++ приложения Win APP обеспечивают следующие функции:

HandleEncryptAction - выполняет действия, связанные с обработкой запроса на шифрование файла от пользователя;

HandleDecryptAction - выполняет действия, связанные с обработкой запроса на дефрование файла от пользователя;

GenerateKeyFile - генерирует файл с ключом, который в дальнейшем используется в данном приложении;

PopulateEditWithSelectedFileName - сохраняет имя файла, выбранного в стандартном диалоге открытия файла в элементе управления Edit;

SimplyGetSaveFileName - функция-обертка, получает имя файла, выбранное пользователем в стандартном диалоге «Сохранить файл»;
Error - функция-обработчик ошибок, Выводит сообщение об ошибке для пользователя.
На рисунке 1 представлена архитектура приложения.

Рисунок 1 – Архитектура приложения

Ниже приведен фрагмент кода реализации функции:

```
void GenerateKeyFile(char *keyfilename)
{
 HCRYPTPROV hProvider; HCRYPTKEY hKey;
 HCRYPTKEY hSessionKey;
 bool successful = true;
 successful = CryptAcquireContext(&hProvider, NULL, MS_DEF_PROV, PROV_RSA_FULL, CRYPT_VERIFYCONTEXT);
 if (!successful) {
        Debug("GenerateKey - CryptAcquireContext error");
        return;
 }
 successful = CryptImportKey(hProvider, key, sizeof(key), 0, 0, &hKey);
 if (!successful) {
        Debug("GenerateKey - CryptImportKey error");
        return;
 }
.
.
}
```

Приложение прошло функциональное тестирование, показало работоспособность и устойчивую работу.

Литература

1. Основные термины и определения. Классификация шифров. [Электронный ресурс]. Режим доступа: https://www.sites.google.com/site/anisimovkhv/ learning/kripto/lecture/tema3 – Дата доступа : 03.12.2019.
2. Вкратце о MicrosoftCrypto API [Электронный ресурс]. Режим доступа: https://habr.com/ru/sandbox/22763/ – Дата доступа : 05.12.2019