Crypter. ТЗ на ПО

ПО для шифрования файлов

Содержание

[Как работать с настоящим документом](#h.ivh45sydi4sc)

[TODO](#h.cf2cclmrg03r)

[Замечания](#h.hypcufj0scel)

[Вопросы](#h.mkz56mo3nqnw)

[Общие сведения](#h.6pd5adq65k7y)

[Назначение](#h.c1f42s2vcymg)

[Исполнение](#h.dcdxtwdtu7yi)

[Применение](#h.15m1plvbc9j3)

[Интерфейс](#h.kth3kj5kecnz)

[Входные параметры](#h.gpb9kdmyjdd8)

[-i <входной файл>](#h.m9383ite2pda)

[-o <выходной файл>](#h.18dazfrgdcp7)

[-c <конфигурационный файл>](#h.46236w1pmiru)

[-k <ключ шифрования>](#h.fln6f26xlmqc)

[-n <инициализационный вектор>](#h.rpyv637lfd5o)

[-a (добавить значение nonce к имени выходного файла)](#h.6uo7j0fuj838)

[Конфигурационный файл](#h.bkoes38yuil)

[Алгоритм](#h.s8je6e5ekut3)

[Обновление nonce](#h.pa731xhg1088)

[Требования](#h.rfx3sx9rgy3u)

[Операционная система](#h.lx390rlsxusp)

[Необходимые библиотеки и фреймворки](#h.31ae9r834sm0)

[Среда разработки приложения (проекта приложения)](#h.qqpmm6u0bgpf)

[Результаты работы](#h.c3bmk9dgzxks)

[Данные для тестирования](#h.dqiz9abm28lk)

# Как работать с настоящим документом

Настоящий документ является рабочим, т.е. он допускает правки и дополнения как заказчиком так и исполнителем, которые предварительно указываются в одном из следующих подпунктов или с помощью комментария на полях документа.

## TODO

## Замечания

## Вопросы

# Общие сведения

## Назначение

Для шифрования файлов по алгоритму AES в режиме CTR (CTR mode). Предполагается использовать приложение для шифрования файлов образа прошивки микроконтроллера, для последующего размещения на сервере.

## Исполнение

Консольное приложение Win32. Запуск из командной строки с указанием входных параметров (опций). Приложение может быть написано на интерпретируемом языке.

## Применение

Вызов из командной строки с указанием входных параметров в строке вызова и/или конфигурационном файле.

# Интерфейс

Реализуется средствами командной строки.

Интерфейс должен соответствовать требованиям стандарта [POSIX.1-2008](http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap12.html) ([POSIX Conventions for Command Line Arguments](http://www.math.uni-hamburg.de/doc/java/tutorial/essential/attributes/_posix.html)). Если требования к интерфейсу приведённые в данном ТЗ противоречат указанным стандартам, то это противоречие должно быть указано в данном документе в виде комментария с целью последующего внесения в требования ТЗ правок.

## Входные параметры

Входные параметры передаются приложению в виде строки после команды вызова приложения через консоль и/или из конфигурационного файла, путь к которому может быть указан во входных аргументах при запуске приложения. Параметры делятся на обязательные и необязательные. Обязательные параметры должны быть указаны при запуске приложения в списке параметров, либо в конфигурационном файле, путь к которому указывается при вызове приложения.

Параметры могут быть перечислены в любом порядке.

Например, строка:

|  |
| --- |
| crypter.exe -i .\build\input\_file.bin -o .\fw\_images\output\_file.bin -c .\misc\crypter.cfg |

запускает шифрование файла .\build\input\_file.bin записывая результат в .\fw\_images\output\_file.bin. Входные параметры берутся из .\misc\crypter.cfg

### -i <входной файл>

Обязательный параметр. Файл, который должен быть зашифрован.

### -o <выходной файл>

Обязательный параметр. Файл, в который должен быть помещён результат шифрования входного файла. Если файл существует, то он должен быть перезаписан.

### -c **<**конфигурационный файл>

Необязательный параметр. Файл, в котором содержатся конфигурационные параметры приложения.

### -k <ключ шифрования>

Обязательный параметр. 256 битный ключ шифрования, состоящий из ASCII символов (не более 32 символов). Если указанный ключ содержит меньшее количество символов, то недостающие символы заменяются на символ с кодом 0xFF. К примеру, входной параметр

|  |
| --- |
| -k abcdefghij12345ABC |

соответствует ключу

|  |
| --- |
| LSB MSB |
| 61 62 63 64 65 66 67 68 69 6A 31 32 33 34 35 41 42 43 FF FF FF FF FF FF FF FF FF FF FF FF FF FF |

Если символов больше чем 32, то символы следующие за 32-ым символом не учитываются.

### -n <инициализационный вектор>

Необязательный параметр. 128 битный инициализационный вектор (или nonce), необходимый для шифрования по алгоритму AES в режиме CTR (CTR mode). Состоит из ASCII символов (не более 16 символов). Если указанное значение содержит меньшее количество символов, то недостающие символы заменяются на символ с кодом 0xFF. Если запуск приложения происходит без указания значения инициализационного вектора, то он должен быть сгенерирован случайным образом внутри приложения. При этом сгенерированное значение инициализационного вектора должно быть добавлено к имени файла.

К примеру, при запуске приложения указан выходной файл *output\_file.bin* и не указано значение nonce. Во время работы приложения, сгенерирован nonce со значением <qWErtyasDFghzxcv>. В итоге будет создан выходной файл с именем *output\_file\_qWErtyasDFghzxcv.bin*

При генерации nonce случайным образом, значения отдельных байт могут соответствовать служебным символам (непечатным). Такие байты должны быть преобразованы в значения, соответствующие символам из ряда <a...z>, <A...Z> и <0...9>. Способ преобразования должен быть выбран исполнителем и кратко описан в настоящем ТЗ.

### -a (добавить значение nonce к имени выходного файла)

Необязательный параметр. Добавить значение использованного nonce к имени выходного файла. К примеру, при запуске приложения указан выходной файл *-o output\_file.bin* , *-n 123456789* и параметр *-a*. В итоге будет создан выходной файл с именем *output\_file\_123456789.bin*

## Конфигурационный файл

Путь к файлу опционально указывается при вызове приложения с помощью входного параметра *-c*.

Параметры содержащиеся в файле соответствуют синтаксису параметров указываемых при вызове приложения в командной строке за исключением того, что разделять параметры кроме пробела могут ещё и символы CR LF.

Пример конфигурационного файла:

|  |
| --- |
| -k abcdefghij12345ABC  -o .\fw\_images\output\_file.bin  -i .\build\input\_file.bin  -a |

# Алгоритм

Шифрование ведётся по алгоритму [AES в режиме CTR (CTR mode)](http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Counter_.28CTR.29). Ключ шифрования 256 бит.

## Обновление nonce

Значение nonce обновляется от блока к блоку данных инкрементом значения. Т.е.

|  |
| --- |
| nonce\_2 = nonce\_1++; // где nonce\_1, nonce\_2 - 128 битные числа |

# Требования

## Операционная система

Приложение должно запускаться в следующих ОС:

* Windows XP (x86)
* Windows Vista (x86/x64)
* Windows 7 (x86/x64)

## Необходимые библиотеки и фреймворки

Все библиотеки используемые приложением должны входить в состав бинарного файла приложения. Приложение может требовать для работы наличия NET Framework или наличия интерпретатора (при написании на интерпретируемом языке).

## Среда разработки приложения (проекта приложения)

Предпочтительно использование среды разработки, поддерживающей одну или несколько следующих ОС:

* Windows XP (x86)
* Windows Vista (x86/x64)
* Windows 7 (x86/x64)

Использование сред разработки неподдерживающих какую-либо из указанных ОС должно предварительно согласовываться.

# Результаты работы

В результате работы исполнитель должен предоставить:

* Файлы проекта
  + с указанием
    - названия и версии среды разработки приложения
    - используемой ОС
  + код должен содержать коментарии, которые предпочтительно оставлять в формате Doxygen
  + если для сборки приложения система разработки требует специальных настроек, отличных от значений по-умолчанию, то эти настройки должны быть описаны или содержаться в файле экспорта настроек
* Исполнительный (бинарный) файл приложения
* Краткое описание работы приложения с примером использования, а так же входной и выходной файл используемый в примере для проверки заказчиком

## Данные для тестирования

При тестировании использовать следующие данные:

|  |
| --- |
| #define AES\_TEXT\_SIZE 64  uint8\_t AES256key[32] = {0x60,0x3d,0xeb,0x10,0x15,0xca,0x71,0xbe,  0x2b,0x73,0xae,0xf0,0x85,0x7d,0x77,0x81,  0x1f,0x35,0x2c,0x07,0x3b,0x61,0x08,0xd7,  0x2d,0x98,0x10,0xa3,0x09,0x14,0xdf,0xf4}; /\* key size 256 bits \*/  uint8\_t IV\_1[16] = {0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,  0x08,0x09,0x0a,0x0b,0x0c,0x0d,0x0e,0x0f}; /\* initialization vector (nonce) \*/  uint8\_t Plaintext[AES\_TEXT\_SIZE] =  {0x6b,0xc1,0xbe,0xe2,0x2e,0x40,0x9f,0x96,  0xe9,0x3d,0x7e,0x11,0x73,0x93,0x17,0x2a,  0xae,0x2d,0x8a,0x57,0x1e,0x03,0xac,0x9c,  0x9e,0xb7,0x6f,0xac,0x45,0xaf,0x8e,0x51,  0x30,0xc8,0x1c,0x46,0xa3,0x5c,0xe4,0x11,  0xe5,0xfb,0xc1,0x19,0x1a,0x0a,0x52,0xef,  0xf6,0x9f,0x24,0x45,0xdf,0x4f,0x9b,0x17,  0xad,0x2b,0x41,0x7b,0xe6,0x6c,0x37,0x10}; /\* plaintext \*/  uint8\_t Ciphertext[AES\_TEXT\_SIZE] =  {0x76,0x49,0xab,0xac,0x81,0x19,0xb2,0x46,  0xce,0xe9,0x8e,0x9b,0x12,0xe9,0x19,0x7d,  0x50,0x86,0xcb,0x9b,0x50,0x72,0x19,0xee,  0x95,0xdb,0x11,0x3a,0x91,0x76,0x78,0xb2,  0x73,0xbe,0xd6,0xb8,0xe3,0xc1,0x74,0x3b,  0x71,0x16,0xe6,0x9e,0x22,0x22,0x95,0x16,  0x3f,0xf1,0xca,0xa1,0x68,0x1f,0xac,0x09,  0x12,0x0e,0xca,0x30,0x75,0x86,0xe1,0xa7}; /\* ciphertext \*/ |