Crypter.

*File encryption software*

***This is a draft document automatically created from russian spelled file.***

Content

[General information](https://translate.googleusercontent.com/translate_f" \l "h.6pd5adq65k7y)

[Purpose](https://translate.googleusercontent.com/translate_f" \l "h.c1f42s2vcymg)

[Execution](https://translate.googleusercontent.com/translate_f" \l "h.dcdxtwdtu7yi)

[Application](https://translate.googleusercontent.com/translate_f" \l "h.15m1plvbc9j3)

[Interface](https://translate.googleusercontent.com/translate_f" \l "h.kth3kj5kecnz)

[Input parameters](https://translate.googleusercontent.com/translate_f" \l "h.gpb9kdmyjdd8)

[-i <input file>](https://translate.googleusercontent.com/translate_f" \l "h.m9383ite2pda)

[-o <output file>](https://translate.googleusercontent.com/translate_f" \l "h.18dazfrgdcp7)

[-c <configuration file>](https://translate.googleusercontent.com/translate_f" \l "h.46236w1pmiru)

[-k <encryption key>](https://translate.googleusercontent.com/translate_f" \l "h.fln6f26xlmqc)

[-n <initialization vector>](https://translate.googleusercontent.com/translate_f" \l "h.rpyv637lfd5o)

[-a (add nonce to the name of the output file)](https://translate.googleusercontent.com/translate_f" \l "h.6uo7j0fuj838)

[Configuration file](https://translate.googleusercontent.com/translate_f" \l "h.bkoes38yuil)

[Algorithm](https://translate.googleusercontent.com/translate_f" \l "h.s8je6e5ekut3)

[Nonce update](https://translate.googleusercontent.com/translate_f" \l "h.pa731xhg1088)

[Requirements](https://translate.googleusercontent.com/translate_f" \l "h.rfx3sx9rgy3u)

[operating system](https://translate.googleusercontent.com/translate_f" \l "h.lx390rlsxusp)

[Required libraries and frameworks](https://translate.googleusercontent.com/translate_f" \l "h.31ae9r834sm0)

[Application Development Environment (Application Project)](https://translate.googleusercontent.com/translate_f" \l "h.qqpmm6u0bgpf)

[Work results](https://translate.googleusercontent.com/translate_f" \l "h.c3bmk9dgzxks)

[Data for testing](https://translate.googleusercontent.com/translate_f" \l "h.dqiz9abm28lk)

1. General information

**2.1.****Purpose**

To encrypt files using the AES algorithm in CTR mode (CTR mode). It is supposed to use the application to encrypt the files of the microcontroller's firmware, for subsequent placement on the server.

**2.2.****Execution**

Win32 console application. Run from the command line with input parameters (options). The application can be written in an interpreted language.

**2.3.****Application**

A call from the command line with input parameters in the call line and / or configuration file.

1. Interface

              Implemented by the command line.

The interface must comply with the requirements of the [POSIX.1-2008 standard](https://translate.google.com/translate?hl=en&prev=_t&sl=ru&tl=en&u=http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap12.html) ( [POSIX Conventions for Command Line Arguments](https://translate.google.com/translate?hl=en&prev=_t&sl=ru&tl=en&u=http://www.math.uni-hamburg.de/doc/java/tutorial/essential/attributes/_posix.html) ). If the requirements for the interface given in this TK contradict the specified standards, then this contradiction should be indicated in this document as a comment for the purpose of subsequent amendments to the TK requirements.

**3.1.**             **Input parameters**

Input parameters are passed to the application as a string after the command to call the application via the console and / or from the configuration file, the path to which can be specified in the input arguments when the application is started. Parameters are divided into mandatory and optional. Required parameters must be specified when starting the application in the parameter list, or in the configuration file, the path to which is specified when the application is called.

Parameters can be listed in any order.

For example, the string:

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

starts up encryption the file . \ build \ input\_file.bin writing result in . \ fw\_images \ output\_file.bin. Input options are taken from . \ misc \ crypter.cfg

**3.1.1.**             **-i <input file>**

Required. The file to be encrypted.

**3.1.2.**             **-o <output file>**

Required. The file in which the result of the input file encryption is to be placed. If the file exists, it must be overwritten.

**3.1.3.**             **-c <configuration file>**

Optional parameter. The file that contains the configuration settings for the application.

**3.1.4.**             **-k <encryption key>**

Required. 256 bit encryption key, consisting of ASCII characters (no more than 32 characters). If the specified key contains fewer characters, then the missing characters are replaced with a character with the code 0xFF. For example, input parameter

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

matches the key

|  |
| --- |
| 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 |

If there are more than 32 characters, the characters following the 32nd character are not counted.

**3.1.5.**             **-n <initialization vector>**

Optional parameter. 128 bit initialization vector (or nonce) required for AES encryption in CTR mode (CTR mode). Consists of ASCII characters (no more than 16 characters). If the specified value contains fewer characters, then the missing characters are replaced with a character with the code 0xFF. If an application is launched without specifying the value of the initialization vector, then it should be generated randomly within the application. In this case, the generated value of the initialization vector must be added to the file name.

For example, when launching an application, the output file *output\_file.bin* is specified and the value nonce is not specified. While the application is running, a nonce is generated with a value of <qWErtyasDFghzxcv>. As a result, an output file will be created with the name *output\_file\_qWErtyasDFghzxcv.bin*

When generating nonce randomly, the values ​​of individual bytes may correspond to service characters (non-printable). Such bytes must be converted to values ​​corresponding to characters from the <a ... z>, <A ... Z> and <0 ... 9> series. The transformation method should be chosen by the performer and briefly described in this TK.

**3.1.6.**             **-a (add nonce to the name of the output file)**

Optional parameter. Add the value of the used nonce to the name of the output file. For example, when launching an application, the output file *-o output\_file.bin* , *-n 123456789* and the *-a*parameter are specified . As a result, an output file will be created with the name *output\_file\_123456789.bin*

**3.2.**             **Configuration file**

The path to the file is optionally specified when the application is invoked using the *-c* input parameter .

The parameters contained in the file correspond to the syntax of the parameters specified when the application is invoked on the command line, except that the CR LF characters can also separate parameters apart from the space.

Sample configuration file:

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

1. Algorithm

Encryption is performed using the [AES](https://translate.google.com/translate?hl=en&prev=_t&sl=ru&tl=en&u=http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation%23Counter_.28CTR.29" \l "Counter_.28CTR.29) algorithm [in CTR mode (CTR mode)](https://translate.google.com/translate?hl=en&prev=_t&sl=ru&tl=en&u=http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation%23Counter_.28CTR.29" \l "Counter_.28CTR.29) . 256 bit encryption key.

It should be noted that the CTR mode has the following property: if the data A is encrypted using nonce B and the C key, then some result D is obtained. If, in this case, the data D is encrypted using the nonce B and the C key, the result is A.

**4.1.**             **Nonce update**

The nonce value is updated from block to data block by increment of value. Those.

|  |
| --- |
| nonce\_2 = nonce\_1 ++; // where nonce\_1, nonce\_2 is 128 bit numbers |

1. Requirements

**5.1.**             **operating system**

The application should run in the following OS:

        Windows XP (x86)

        Windows Vista (x86 / x64)

        Windows 7 (x86 / x64)

**5.2.**             **Required libraries and frameworks**

All libraries used by the application must be part of the application binary file. An application may require the presence of the .NET Framework or the presence of an interpreter (when writing in an interpreted language).

**5.3.**             **Application Development Environment (Application Project)**

It is preferable to use a development environment that supports one or more of the following operating systems:

        Windows XP (x86)

        Windows Vista (x86 / x64)

        Windows 7 (x86 / x64)

The use of development environments that do not support any of these operating systems should be pre-agreed.

1. Work results

As a result of the work, the performer must provide:

        Project files

        with indication

■        names and versions of the application development environment

■        OS used

        the code should contain comments, which are preferably left in the Doxygen format

        If the development system requires special settings for assembling the application, other than the default values, then these settings should be described or contained in the settings export file

        Executive (binary) application file

        A brief description of the application with an example of use, as well as the input and output file used in the example for verification by the customer

**6.1.**             **Data for testing**

When testing, use the following data:

|  |
| --- |
| #define AES\_TEXT\_SIZE 64  / \* key size 256 bits; +1 for \ 0 \* /  char AES256key [32 + 1] = "qwertyuiQWERTYUIasdfghjkASDFGHJK" ;    / \* initialization vector (nonce); +1 for \ 0 \* /  char IV\_1 [16 + 1] = "zxcvbnmmZXCVBNMM" ;    / \* plaintext \* /  char Plaintext [AES\_TEXT\_SIZE + 1] = "testTESTtestTEST" \                                                               "TESTtestTESTtest" \                                                               "testTESTtestTEST" \                                                               "TESTtestTESTtest";    / \* result Plaintext processing \* /  / \*  0x42, 0x4F, 0xE6, 0xA5, 0x00, 0xAF, 0xAE, 0xFE,  0x91, 0x1E, 0x7A, 0x49, 0x2F, 0x45, 0xF1, 0xDC,  0xA8, 0x88, 0xFC, 0xD2, 0x46, 0x21, 0x0A, 0x2D,  0x44, 0xE0, 0x10, 0xF9, 0x08, 0x46, 0xB1, 0xAC,  0x4A, 0x78, 0x0C, 0x3F, 0xE4, 0xA9, 0x2E, 0xD3,  0xDD, 0xCD, 0x57, 0xF9, 0xDF, 0x33, 0x78, 0xE7,  0x23, 0xA8, 0x74, 0xA9, 0x57, 0xFE, 0x9A, 0x11,  0x83, 0x05, 0x39, 0xC4, 0x6C, 0xB9, 0x84, 0x0E  \* / |

Additional test data (AES256key and IV\_1 do not comply with the terms of reference):

|  |
| --- |
| #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 \* /    / \* result Plaintext processing \* /  / \*  =======================================  Encrypted Data with AES 256 Mode CTR  ---------------------------------------  [0xDC] [0x7E] [0x84] [0xBF] [0xDA] [0x79] [0x16] [0x4B] [0x7E] [0xCD] [0x84] [0x86] [0x98] [0x5D] [0x38] [0x60] Block 0  [0xD5] [0x77] [0x78] [0x8B] [0x8D] [0x8A] [0x85] [0x74] [0x55] [0x13] [0xA5] [0xD5] [0x0F] [0x82] [0x1F] [0x30] Block 1  [0xFF] [0xE9] [0x6D] [0x5C] [0xF5] [0x4B] [0x23] [0x8D] [0xCC] [0x8D] [0x67] [0x83] [0xA8] [0x7F] [0x3B] [0xEA] Block 2  [0xE9] [0xAF] [0x54] [0x63] [0x44] [0xCB] [0x9C] [0xA4] [0xD1] [0xE5] [0x53] [0xFF] [0xC0] [0x6B] [0xC7] [0x3E] Block 3  \* /      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 \* /    / \* result Ciphertext processing \* /  / \*  =======================================  Decrypted Data with AES 256 Mode CTR  ---------------------------------------  [0xC1] [0xF6] [0x91] [0xF1] [0x75] [0x20] [0x3B] [0x9B] [0x59] [0x19] [0x74] [0x0C] [0xF9] [0x27] [0x36] [0x37] Block 0  [0x2B] [0xDC] [0x39] [0x47] [0xC3] [0xFB] [0x30] [0x06] [0x5E] [0x7F] [0xDB] [0x43] [0xDB] [0x5B] [0xE9] [0xD3] Block 1  [0xBC] [0x9F] [0xA7] [0xA2] [0xB5] [0xD6] [0xB3] [0xA7] [0x58] [0x60] [0x40] [0x04] [0x90] [0x57] [0xFC] [0x13] Block 2  [0x20] [0xC1] [0xBA] [0x87] [0xF3] [0x9B] [0xAB] [0xBA] [0x6E] [0xC0] [0xD8] [0xB4] [0x53] [0x81] [0x11] [0x89] Block 3  \* / |