Hardware Symmetric Data Encryption

Symmetric hardware algorithms GSII64 and AES128 are implemented in Guardant dongles. They perform hardware encryption of small volumes of data (*S mall size of converted data is based on the relatively low speed of hardware algorithm operation. Big size of data can significantly slow down protected application.*). See Section **Hardware Algorithms** for more information on symmetric algorithms.

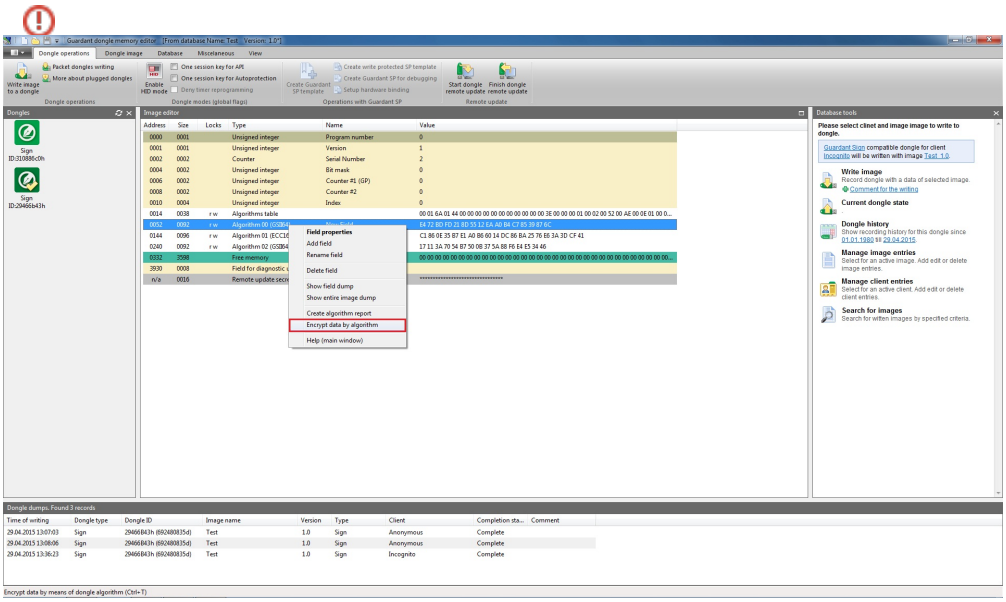
Previously encrypted data can be stored in a protected application or separate files and decrypted right before use.

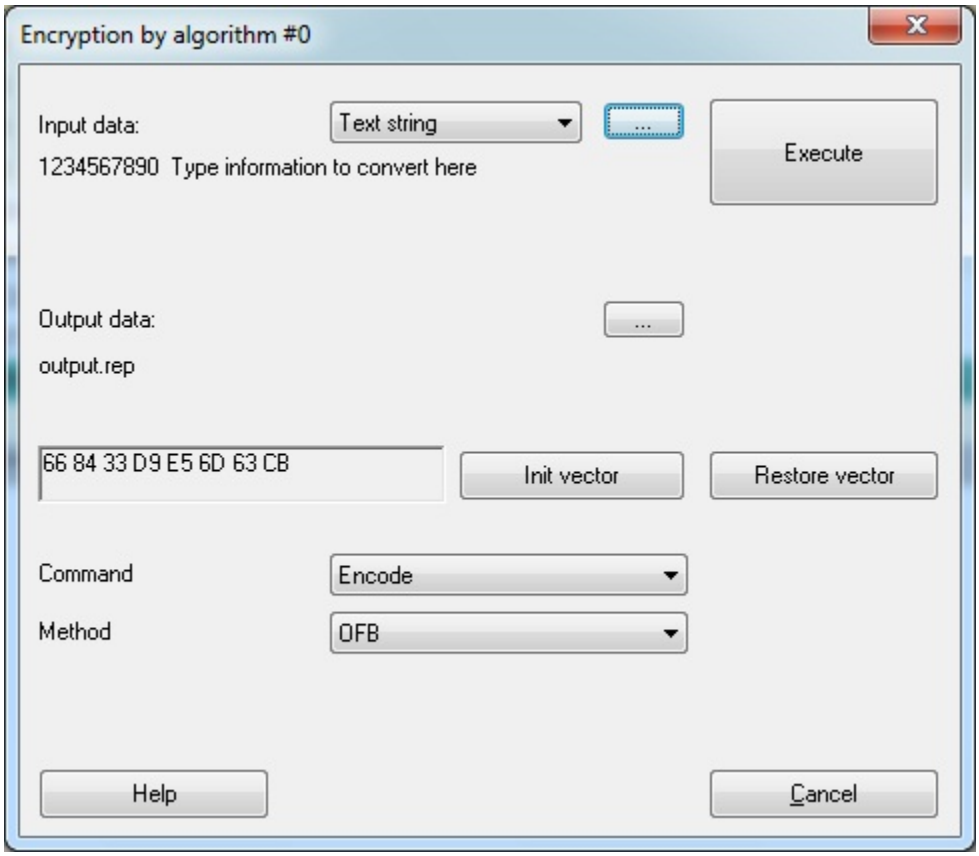
GrdCrypt function is used for calling a symmetric algorithm from within the application. See Guardant API help system **(GrdAPI.chm)** for more information on Guardant API operations.

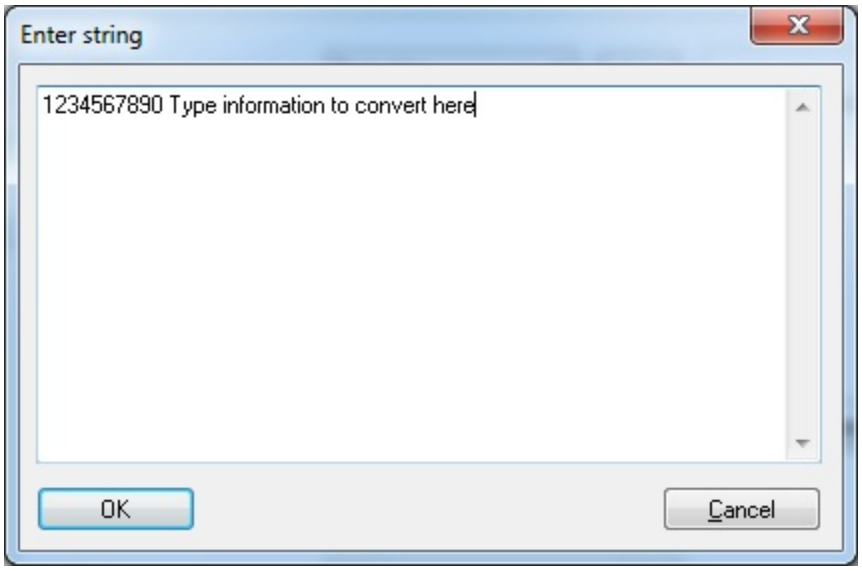
GrdUtil.exe provides an easy-to-use service for preparing the encryption data in advance. You can encrypt and decrypt information using this utility. The prepared data is further used for application protection.

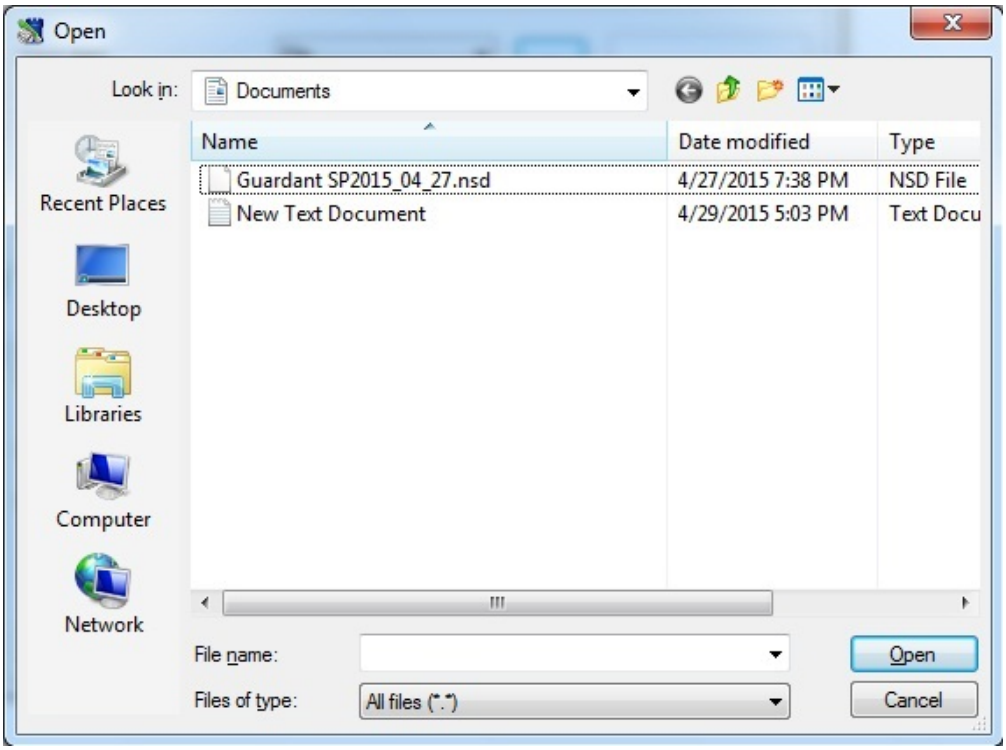
Preparation of data for encryption

Select the required symmetric algorithm from the list of fields of Mask Editor and execute command **Dongle | Encryption. Encryption by algorithm No. N** (*N is the index number of GSII64 algorithm*) dialog box will appear on the screen:




Encryption by algorithm #0
Input data:
Text string
...
Execute
1234567890 Type information to convert here
Output data:
...
output.rep
66 84 33 D9 E5 6D 63 CB
Init vector
Restore vector
Command
Encode
Method
OFB
Help
Cancel


Enter string
1234567890 Type information to convert here
OK
Cancel



Define the following parameters in the dialog box:

- Input data and their type
- Initialization vector
- Output data and their type
- Direction and method of encryption
- Programming language (if output data are presented as source code)

Input data

Data to be encrypted can be presented as: string of characters or file of any format.

A dropdown list in the top part of the dialog box serves for selecting the type of input data.

Clicking **[…]** against the list opens **Enter string** dialog for entering a string of symbols or standard system dialog for specifying the filename and its path.

The defined string or filename with data and its path appear in Input data entry field.

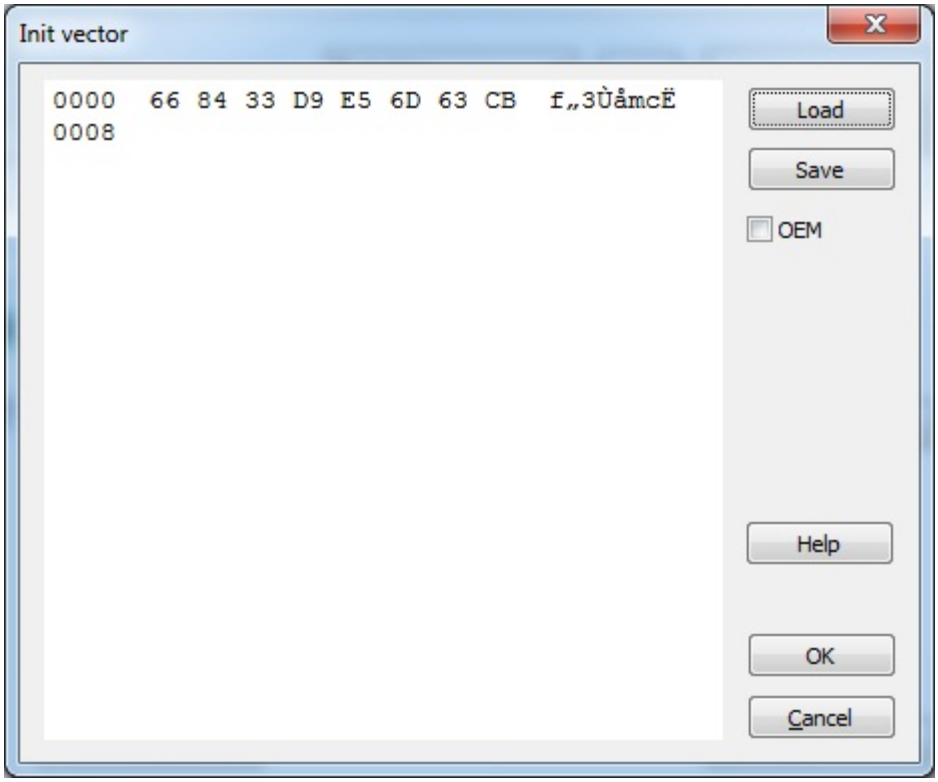
Initialization vector

Initialization vector – a random number used for running symmetric algorithm in ECB, CBC and OFB operating modes. The GSII64 initialization vector is 8 bytes, AES128 – 16 bytes.

Initialization vector is generated automatically upon opening **Encryption by algorithm No** dialog box and is displayed in the respective entry field.

When necessary the default value of the initialization vector can be changed. Clicking **[Init vector]** button launches a hexadecimal editor serving for changing or setting a new initialization vector value.

Initialization vector dialog box:



Initialization vector dialog box control elements:

Interface element	Description of purpose
Hexadecimal editor window	Enter initialization vector value
[Load] button	Load dump from *.dmp file
[Save] button	Save dump into *.dmp file
OEM flag	Select Windows/DOS encoding. Windows (ANSI) encoding is used by default – OEM option is off.

The vector value changes during the encryption. **[Restore]** buttons serves for restoring the original initialization vector value.

Output data

Encrypted data can have the following form:

Output data	Description
Source code	A text file containing encrypted data in a form of array of numbers and created according to the syntax rules of one of the basic programming languages: Assembler, C/C++, Pascal/Delphi
Binary code	Encrypted sequence of bytes

A dropdown menu in the middle part of the dialog box serves for selecting the type of data presentation.

Clicking **[…]** against the list opens a standard system dialog for specifying the name of file containing encrypted data *(Output.rep* by default) and its path.

The name of data file and its path appear in **Output data** entry field.

[Execute] button

Clicking **[Execute]** button initiates the process of data encryption (decryption). The button is enabled after filling out the **Input data** and **Output data** sections.

Programming language

Programming language dropdown list located in the lower part of the dialog box becomes enabled only if the encrypted data is represented as source code in the selected programming language.

The following programming languages are available: Assembler, C/C++, Pascal/Delphi.

Encryption and decryption

Command dropdown list serves for selecting an operation, which will be performed with the input data: encryption or decryption.

Encryption Method

Symmetric algorithms have 4 operating modes different by their properties and purposes. GSII64 and AES128 algorithm is described in details in Chapter **Hardware Algorithms.**

Select method of encryption using the dropdown list.

Performing encryption

Clicking **[Execute]** button located in the top part of the dialog box initiates the encryption.

Writing mask data into dongle memory

The utility requests for confirmation of writing the mask data into the dongle before the encryption is initiated:

Preliminary writing of mask into dongle is required if a new algorithm is used or an algorithm with changed determinant.

Saving report

After that the encryption report saving dialog box will appear on the screen where filename *(report.rep* by default) and path should be specified.

The encryption report is a text file generated according to the syntax rules of the selected programming language. The report contains statistical information on encryption parameters & sets password in the form of array:

Process of encryption

After saving the report a progress bar will appear on the screen.

The encrypted data are saved in the specified file in the form of array or sequence of bytes.

Decryption

The process of decryption is similar to the process of encryption (see above). The input data here is a file with encrypted data. The direction of encryption is changed to decryption **(Command** list).

Important information

For correct decryption of data you need to use the same algorithm, an initialization vector and method of encryption that were used for encryption.