Structure

Descriptor

Each hardware algorithm of a Guardant dongle is defined in its descriptor. Descriptors are kept in the dongle memory in protected items of special type and protected against reading and modifications. A part of the descriptor defines the properties of the algorithm. Its second part represents a hardware algorithm determinant. It plays a significant part in executing a specific algorithm while being a secret conversion key.

Properties of Hardware Algorithms and its use

A combination of algorithm's properties also participates in the process of creating a hardware algorithm descriptor. The properties are defined by a combination of special flags stored in the algorithm descriptor. You can fine-tune the required 'behavior' of a hardware algorithm by defining the combination of its properties.

Algorithm with limited number of runs

If flag **nsaf_GP_dec** is set in the algorithm descriptor, its counter (4-byte descriptor field **km_ad_GP)** will decrement on each run of the algorithm. When the counter reaches zero, the algorithm will automatically deactivate and stop converting data. This is an excellent way of creating a protected application with a limited license term.

For reverse algorithm activation, the counter of which reached zero, you need to locally reprogram the dongle or write a new value into **km_ad_GP** field of this algorithm's descriptor using TRU.

You can calculate the approximate number of calls to the algorithm during the suggested application (or its new version) term of operation and limit the number of times the algorithm will be executed. This can be one of the ways to withstand the brute force attacks on the algorithm.

Activation/deactivation of hardware algorithms

If these properties are defined in the algorithm's determinant, you will have an opportunity to activate or deactivate the algorithm when required using a special function. You can set passwords for activation/deactivation to be stored in the algorithm descriptor. Activation and deactivation enables efficient control over a specific set of algorithm descriptors engaged in the operation of protection system.

Dependence on ID

Flag **nsaf_ID** sets the dependence of a hardware algorithm on the dongle ID. This means that such algorithm will convert data in each dongle in a unique way even if all values of its descriptor are the same in all dongles.

Warning

If a dongle with such hardware algorithm gets accidentally damaged, it will be impossible to replace it with a dongle with the same algorithm, since there cannot be two dongles with the same IDs.

Operating time algorithm limitation

Guardant dongles with real time clocks allow controlling the activity of hardware algorithms using the real time timer with autonomous power source. For this special fields are provided for in the hardware algorithm descriptor, in which the time limitations are stored.

See **Using timer for controlling the hardware algorithm status** for more details.

Private Key (Determinant)

Determinant of a hardware algorithm is a set of bytes written in a special field of the algorithm descriptor and interpreted by the microprogram as a secret key of the algorithm.

Algorithm determinants represent critical data related to protection of applications and should always be kept a secret. Determinants shall be accessed by the authorized personnel only.

In the process of encryption/decryption or calculation of hash-function the determinant is kept inside the controller's memory. Algorithm's descriptor containing the determinant should be protected against reading/writing by hardware locks.

It is recommended to use reliable protected algorithms for generating random numbers for generating determinants, for example, hardware algorithm of type RND64. It is not recommended to use builtin functions of programming languages.

It is useful to change determinants from time to time. This is a very good and widespread practice for increasing the tamperproofness of the system. We recommend changing algorithm determinants upon release of a new version of protected application.