Loadable code (Guardant Code / Code Time only)

A dongle with the capability of loading the application code represents some kind of trusted platform allowing performing significant calculations outside of the CPU and computer's RAM, on which the protected application is running.

Traditionally dongles could perform two main things:

a. store some data critical for the protected application operation,
b. internally execute algorithms converting data using some algorithms.

These algorithms, as a rule, are represented as symmetric encryption algorithms, or unidirectional functions with secret conversion key.

The developers of protection systems based on "traditional" dongles are often confused by the choice of data to be converted. As a rule, they would generate such data artificially and not using the stream of real data, on which the calculations are based.

This creates some problems related to the fact that the data converted with hardware algorithm is, first, not always can be directly used in the application, and secondly, you should try real hard to make sure that the flow of these data is diverse over a significantly long timeframe.

The loadable code's protection mechanism is based on the fact that the algorithm, built into the dongle by the developer, processes natural data received in the process of application execution. The processed data can be used in the application directly, overriding the validation, which, as a rule, boils down to one or two assembler commands. The natural data flow is not constant and diverse. Therefore the loadable code algorithm will much likely perform calculations on the constantly changing data if this algorithm is properly selected and correctly implemented.