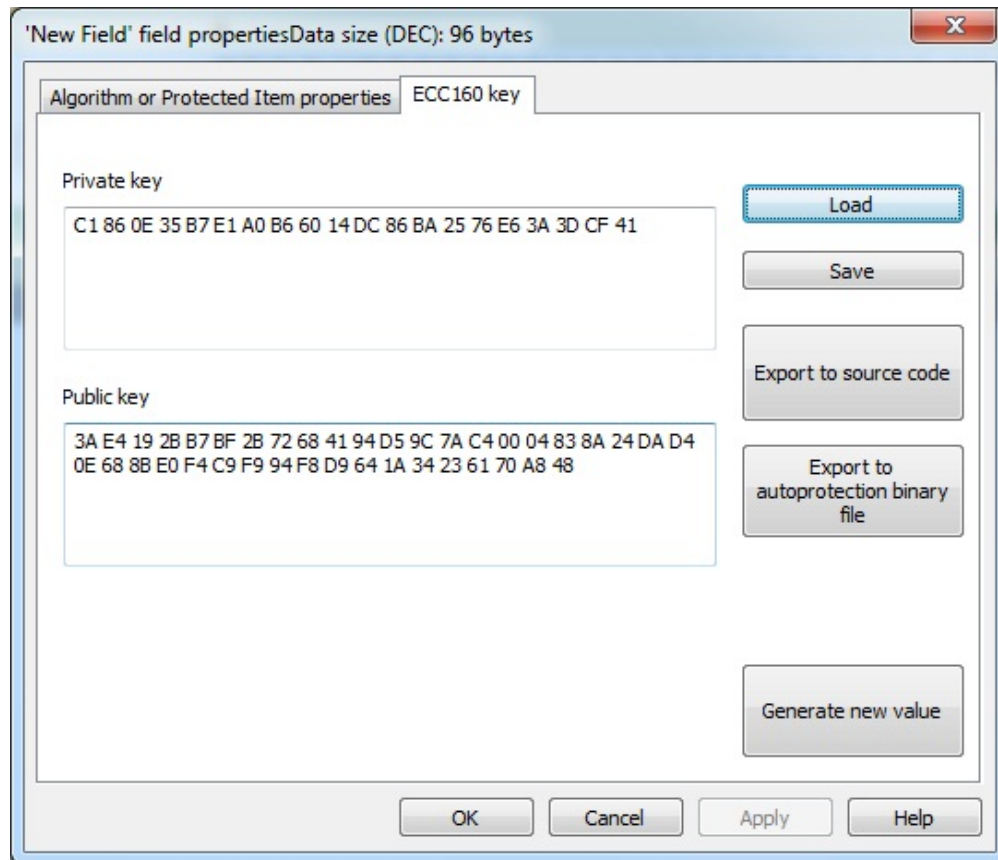


ECC160 key pair

When creating/editing an elliptic curve digital signature algorithm (ECC160), the **Determinant Editing** dialog is replaced with **ECC160 Key**.

The **ECC160 Key** dialog serves to generating the key pair of the digital signature algorithm ECC160.

The private key appears in the top part of the dialog box, and the public key – in the bottom of the dialog box. To the right there are buttons for performing service operations on the key pair.



By default a key pair of an ECC160 algorithm is generated randomly. The pair can be changed by automatically creating a new one using the **[Generate new value]** button.

The **[Save]** button allows for saving the key pair into an external file (*.ecc). Clicking **[Load]** allows the previously saved pair to be loaded from a *.ecc file.

In order to use ECC160 algorithm for autoprotection you need to specify the path to the file with a public key as the parameter of SIGN option. **[Export to autoprotection binary file]** allows saving the public key into a file (PublicKey.bin, by default).

The **[Export to source code]** button saves the key pair into the C++ header file (EccKeySource.h, by default).

After editing the determinant (or ECC160 key pair) and clicking **[Finish]**, the algorithm creation dialog box will close and a new algorithm will appear in the list of mask fields. Whereas GrdUtil.exe will automatically assign an index number (*Unified numbering is used for protected items and hardware algorithms*) to the algorithm and correct the bound of hardware locks taking the added algorithm into account.

Now all you have to do is to write the mask into the dongle and the newly created algorithm will be available for use.