JavaCrypTool (www.cryptool.org)

Online help for the usage of the grille analysis

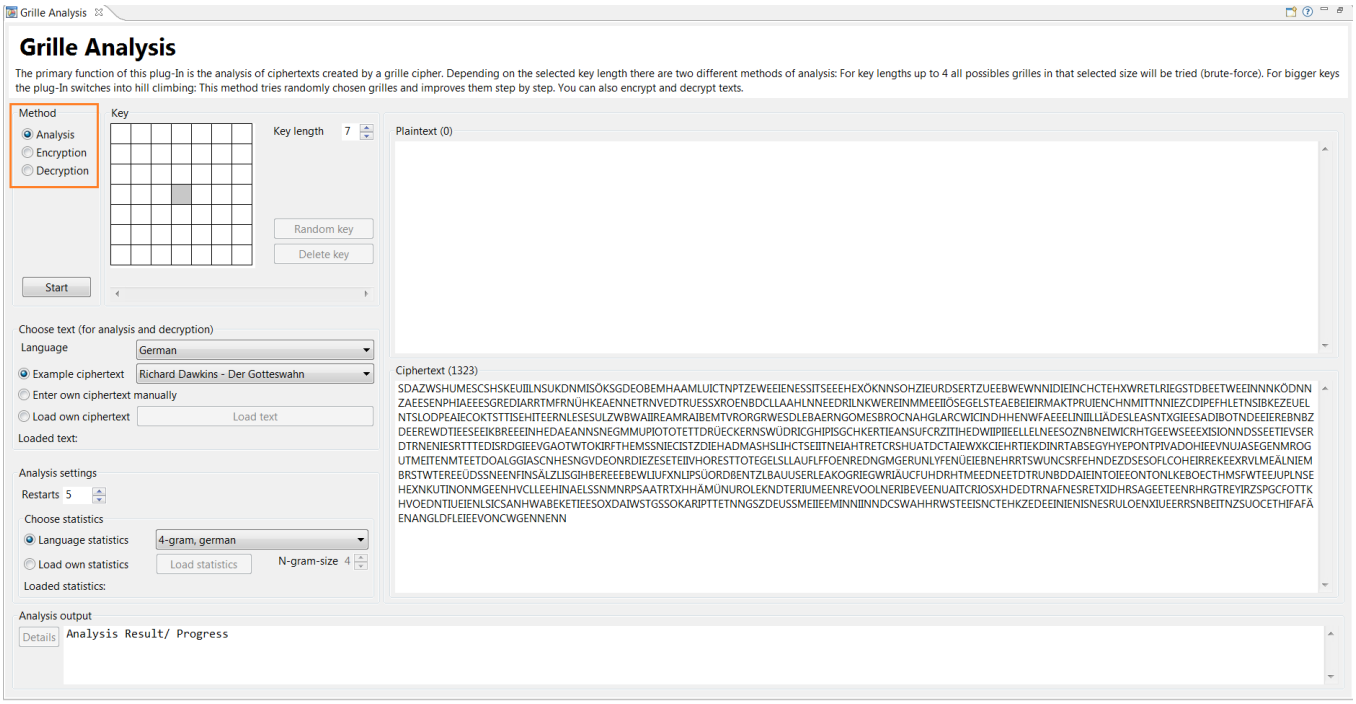
Contents

The main function of this plug-in is the analysis of a ciphertext that has been encrypted using the grille cipher. The content of the template is used as key. Somewhat unusual here is the characterization of the key length as the side length of the square field (template).

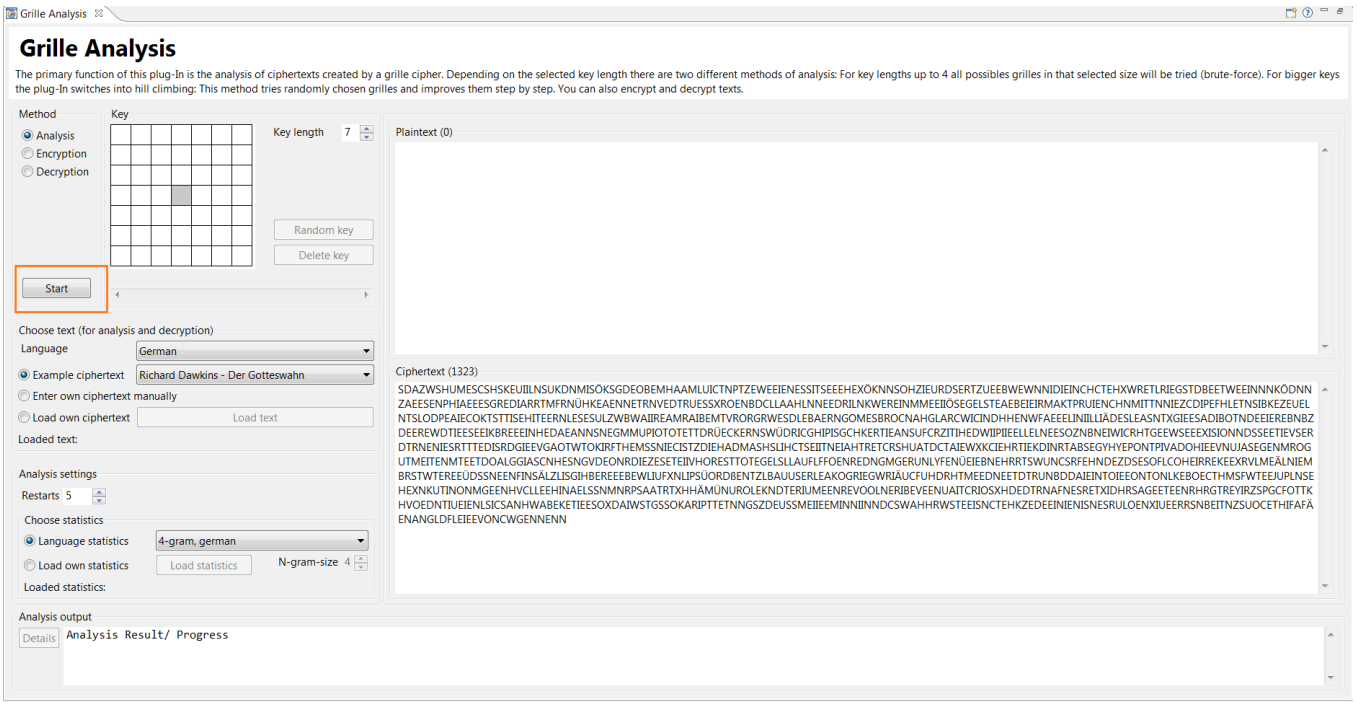
In addition to the analysis function, the functions Encryption and Decryption are also available in this plug-in.

1 Analysis

In the initial setting of the plug-in (default setting), the radio button is set on the analysis function. It is selected in the first aggregation „Method“. The method can be changed to either encryption or decryption.

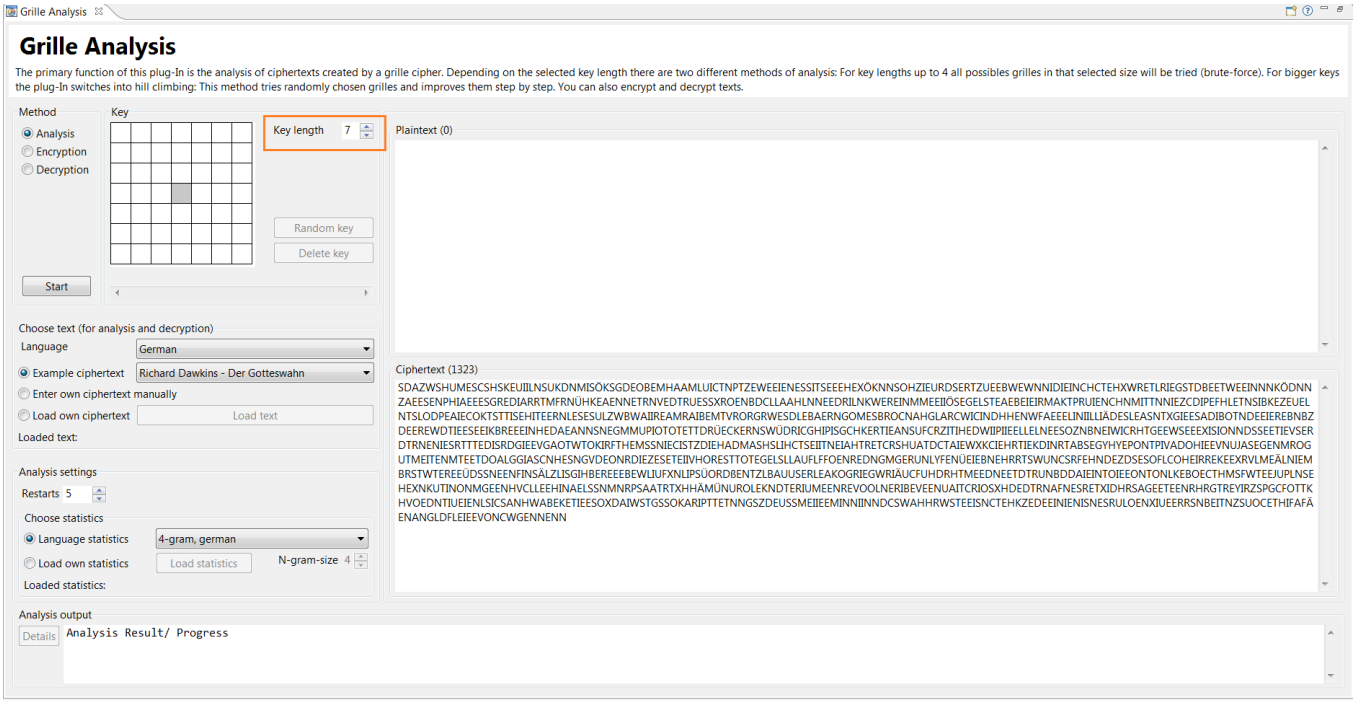


Every relevant parameter for the analysis as well as the ciphertext has a default setting, so as to start an analysis immediately by pressing the „Start“ button below the method selection.



1.1 Key length

Since the key used for encrypting the text is secret and is supposed to be found by the analysis, the key field itself is disabled for this method. However, the key length is important for a successful analysis and is made available next to the key field. This length should match that of the key used for this ciphertext.



For the default setting „Example ciphertext“ in the section Choose text the ciphertext will be adapted

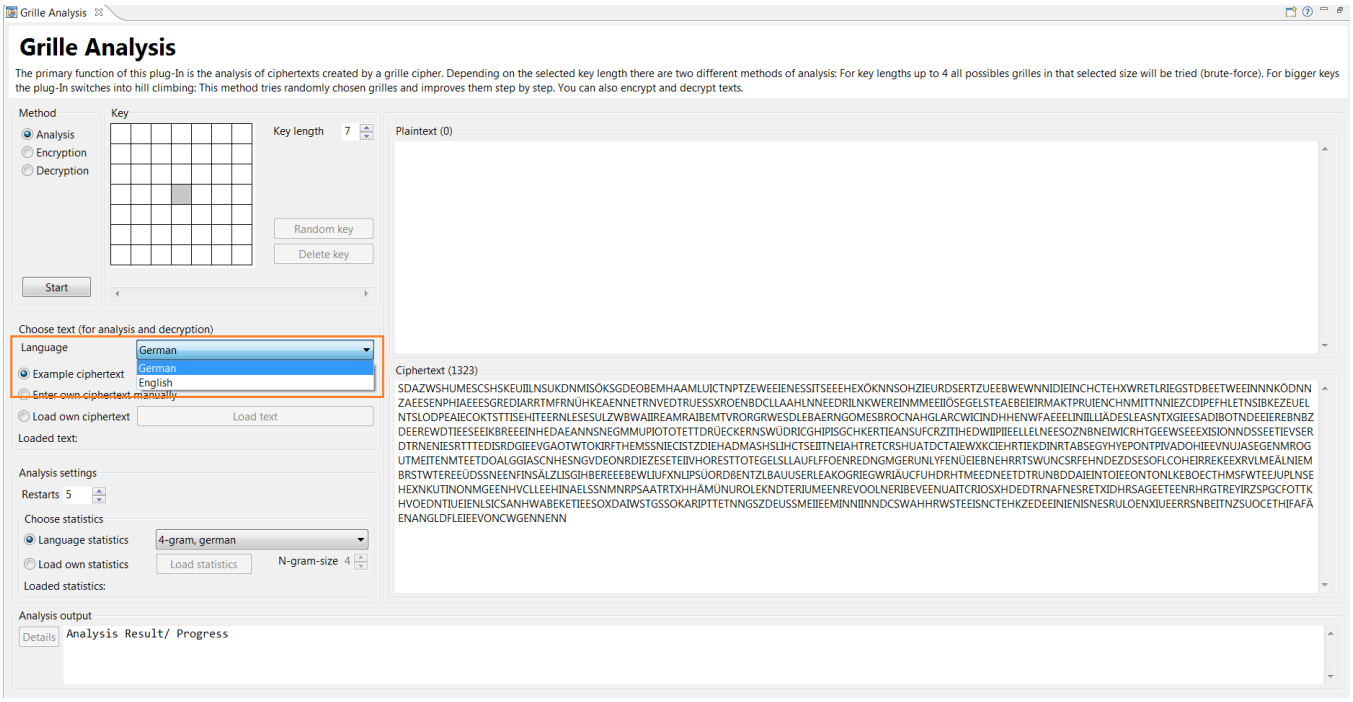
to the respective key length. The corresponding key will be randomly generated and discarded after the encryption.

1.2 Choose text

1.2.1 Language

You can select either „German“ or „English“ as language. The ciphertext displayed by selecting the option „Example ciphertext“ (default) – below the language selection in the section Text input – will be updated according to the choosen language.

If a ciphertext is typed in manually or loaded, the language must be adjusted accordingly. Since the analysis is based on language-specific occurences of alphabetic strings, the text being analyzed must be compatible with the selected language.



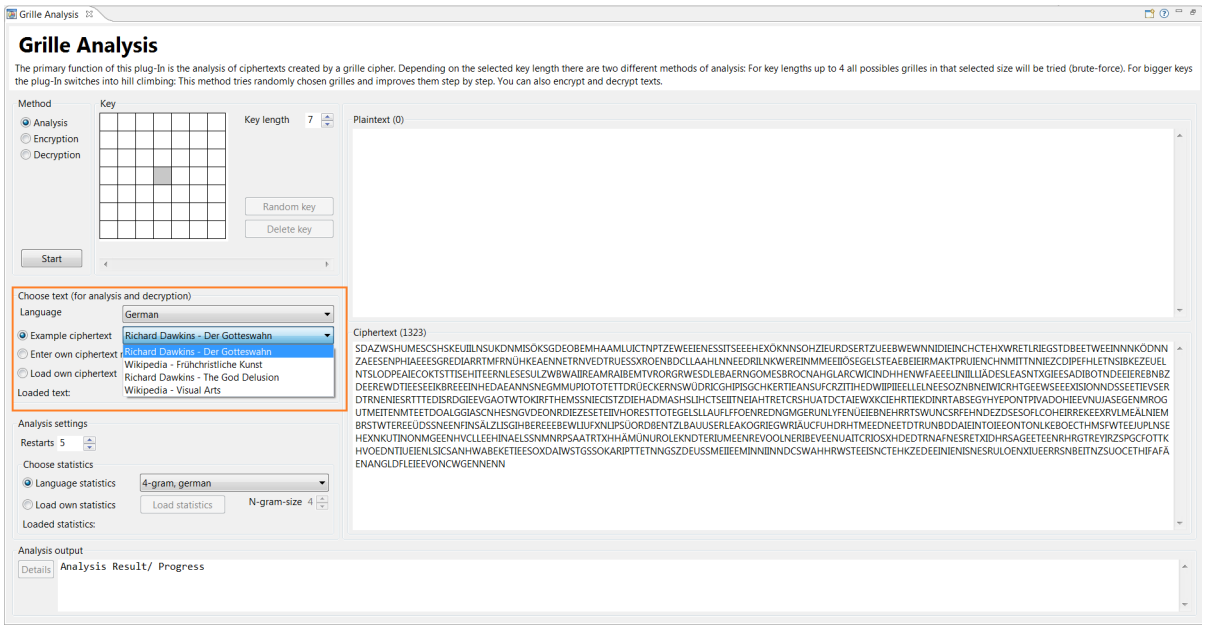
1.2.2 Text input

In this aggregation the type of text input can be selected.

(a) **Example ciphertext**

The selection „Example ciphertext“ is choosen by default. There are two german and two english ciphertexts available for selection. [1]

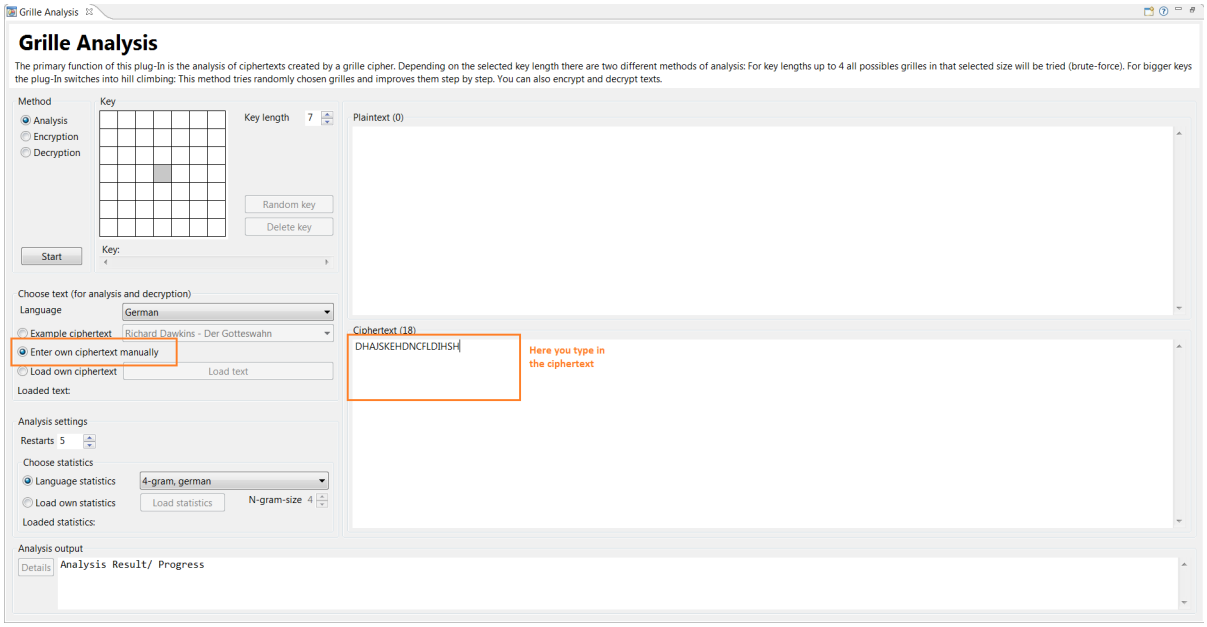
The text encryption will be consistent with the current key length.



[1] **Sources:**
Richard Dawkins - Der Gotteswahn
Richard Dawkins - The God Delusion
`https://de.wikipedia.org/wiki/Bildende_Kunst#Fr%C3%BChchristliche_und_byzantinische_Kunst`
`https://en.wikipedia.org/wiki/Visual_arts`

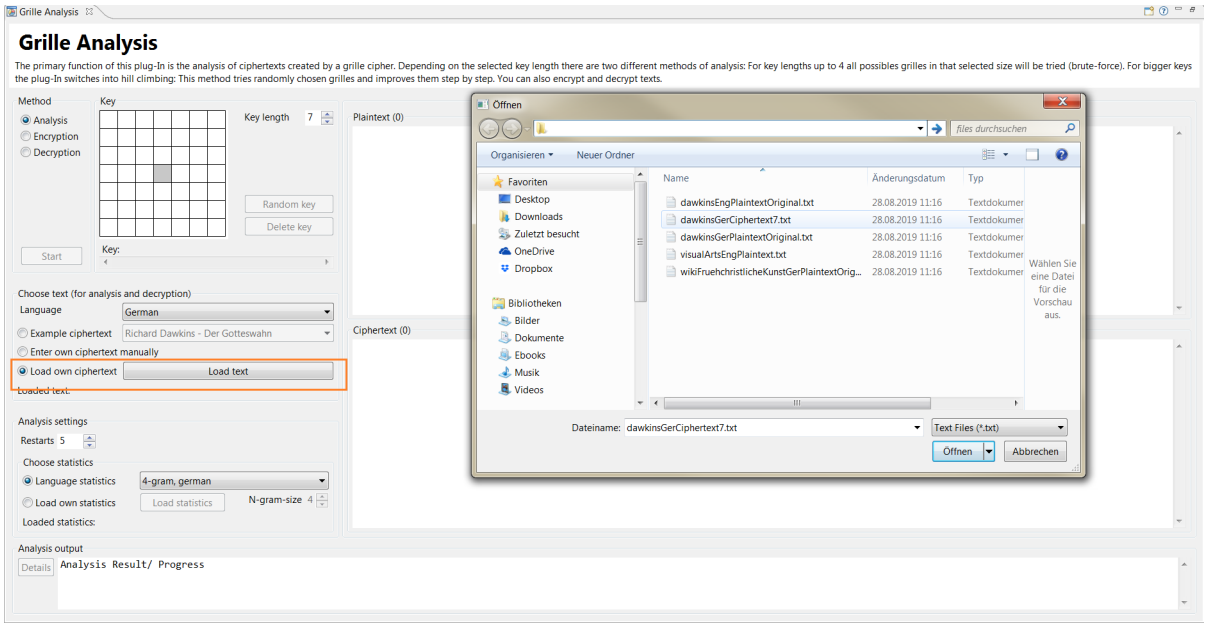
(b) Enter own ciphertext manually

By selecting the „Enter own ciphertext manually“ option, the ciphertext field on the right will be available for entering a text.



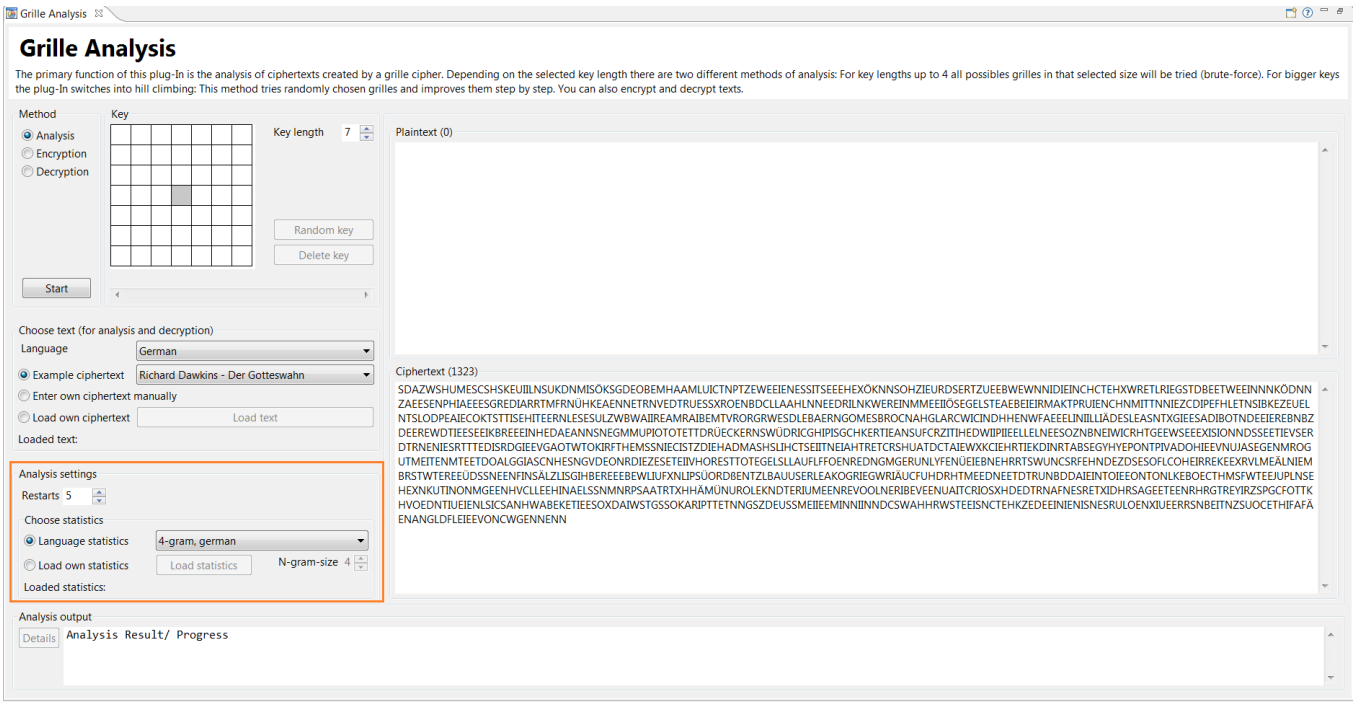
(c) Load own ciphertext

The last selection „Load own ciphertext “ is available for loading a ciphertext from a file. With this selection the button „Load text “ is activated which can then be used to load a text file (*.txt). The text is then displayed in the ciphertext field.



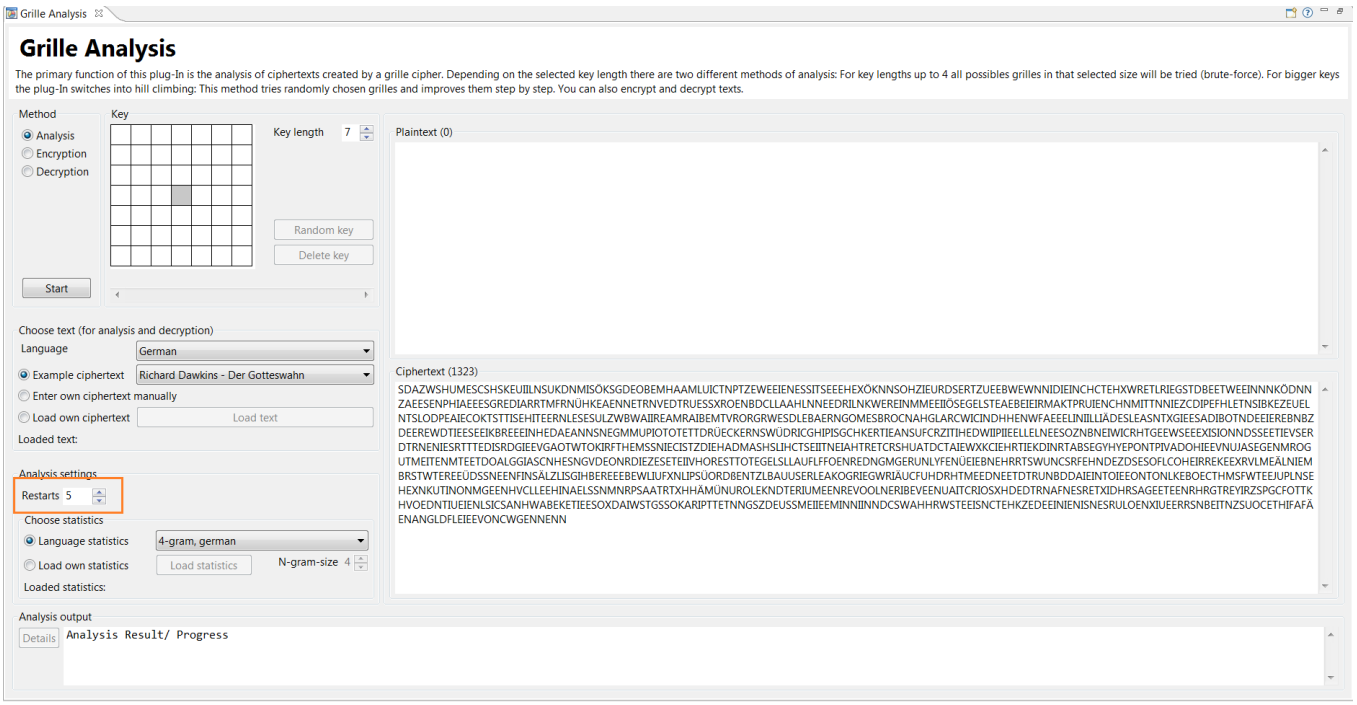
1.3 Analysis settings

The „Analysis settings“ section is located directly below the „Choose text“ section and is only activated for the analysis function.



1.3.1 Restarts

The number of „restarts“ is relevant and thus only enabled for the analysis of ciphertexts that were encryped with a key of size 5 or higher. For a key length with the aforementioned size a „hill climbing“- algorithm will be implemented for the analysis. Every restart creates a new random key for the selected key size. This respective key is then improved step by step by only changing one hole at a time and comparing the quality of the resulting decrypted text. Every improvement is saved and then used as the basis for the next improvement search. This pattern will be repeated until no further improvements can be made by changing one hole. The higher the number of restarts, the higher the chance of finding the right key through the analysis will be.

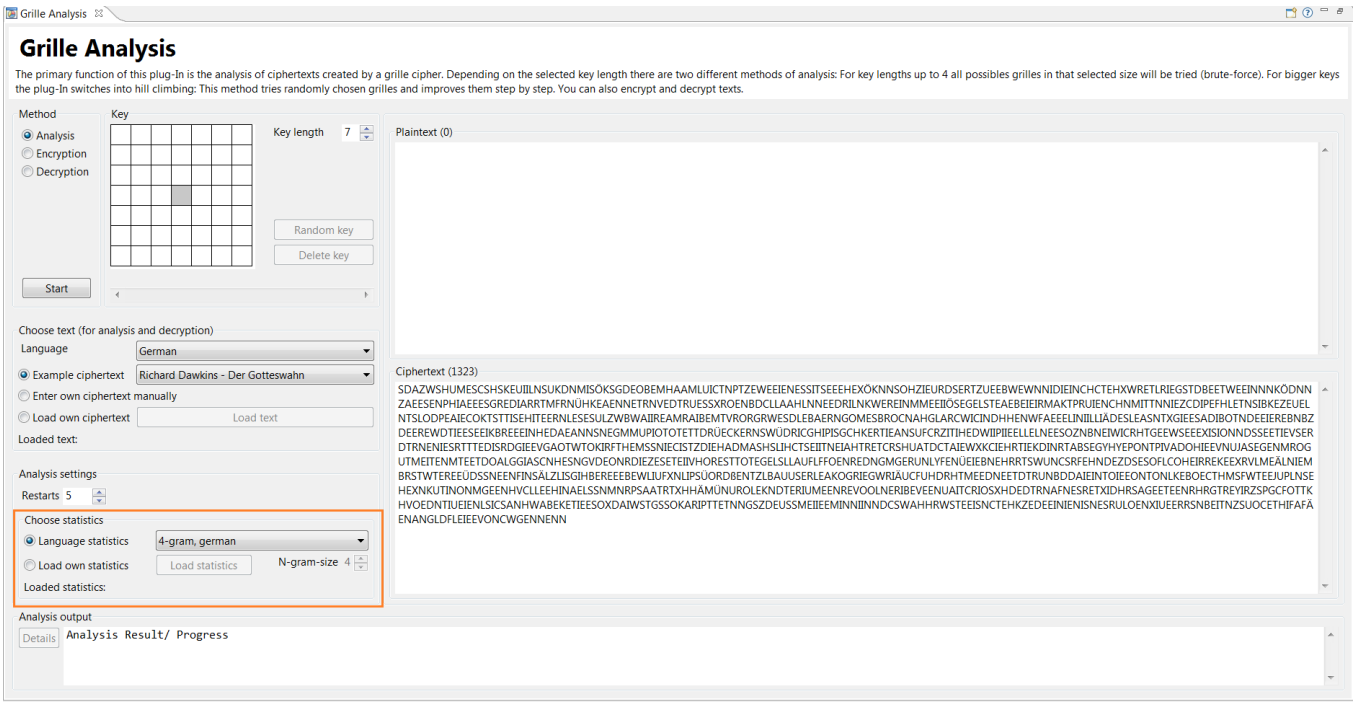


Note: Expect a longer processing time for larger keys and a higher number of restarts. [2]

1.3.2 Language statistics

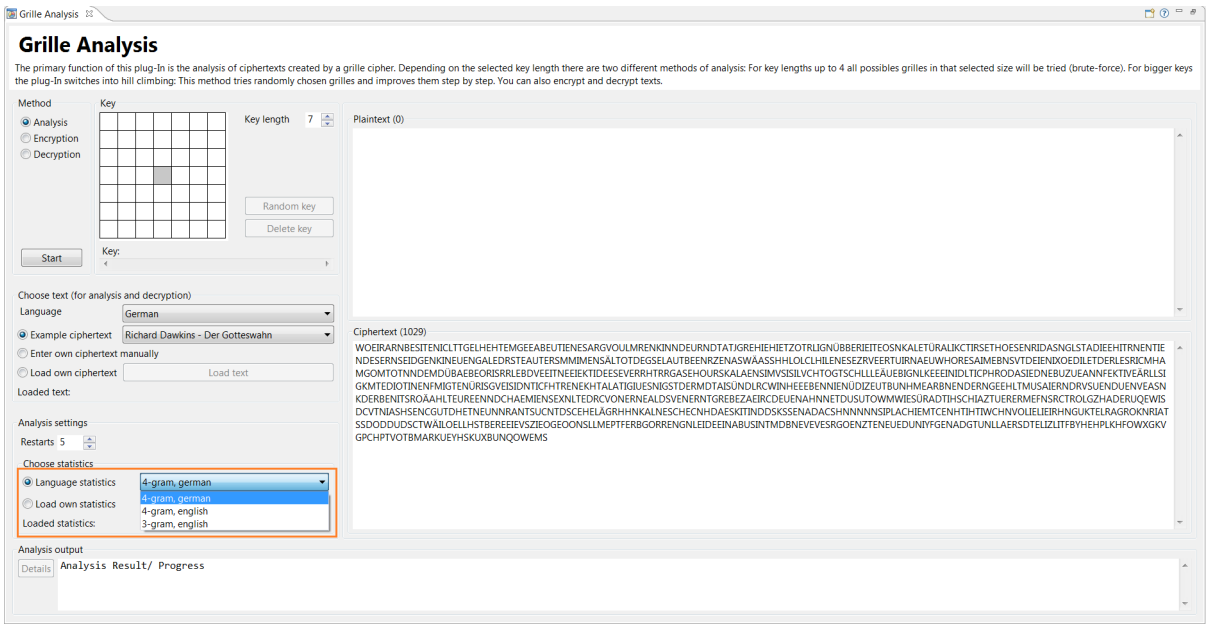
The language statistics are essential for the analysis of a ciphertext. It contains the probability of occurrence of all coherent character strings of a certain length *n* (n-grams) in the respective language. The language of the ciphertext must agree with the selected or loaded statistics. Otherwise, the probability of success drops considerably.

[2] Appendix: Evaluation of the analysis



(a) **Language statistics**

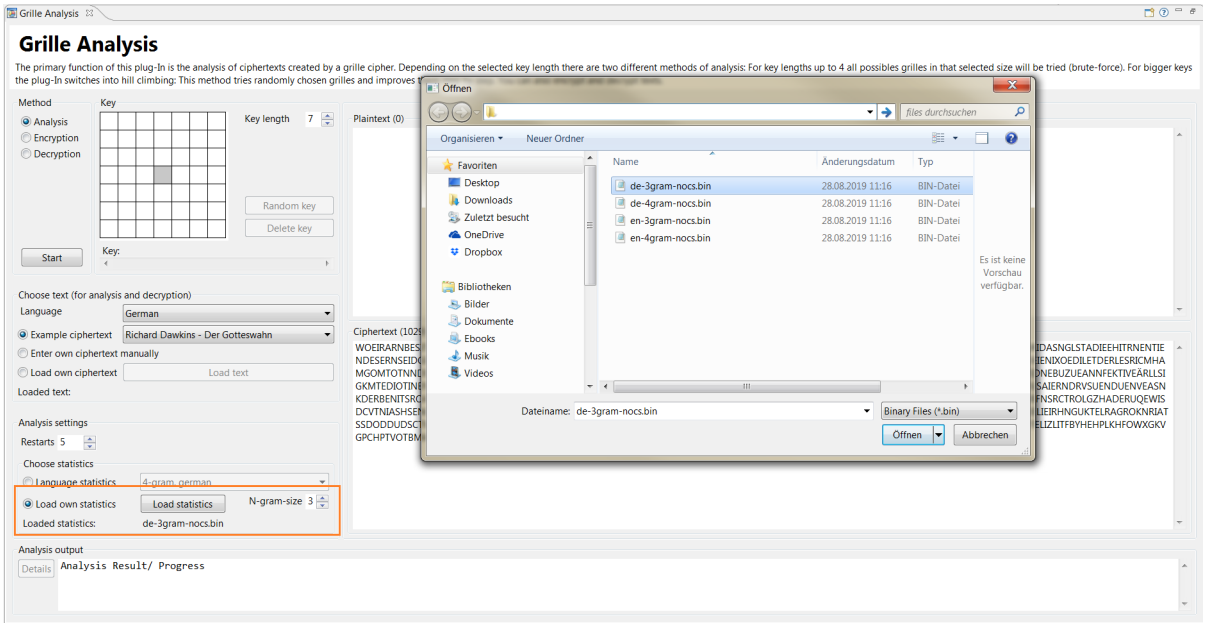
To carry out the analysis, one of the three stored statistics can be applied (one in german and two in english).



(b) **Load own statistics**

The option to load language statistics files is also available. The following are *requirements* for the format of a usable language statistics: Compared to other language statistics, the actual n-grams for this plug-in should not be in the file. The language statistics used should only contain the logarithmized values of the respective n-grams. For this purpose, the language statistics are built up like an n -dimensional cube, where the position of the letters in the

alphabet and the position of each letter in the n-gram together form the index for the value of the respective n-gram.



Example: To save and retrieve the quadgram „This“ in an english quadgram statistic, proceed as follows (with lower case letters being treated like upper case letters):

The english alphabet is given in the form „ABC ... XYZ“ of length 26, whereby each letter is assigned an index from 0 to 25. For the quadgram „This“, each letter index of the quadgram is now multiplied by a power of the alphabet length depending on the position in the quadgram. These four values are summed together resulting with the index of the quadgram value in the language statistics. As mentioned above, the letters T, h, i, and s are each assigned the index 19, 7, 8, and 18. The calculation mentioned above

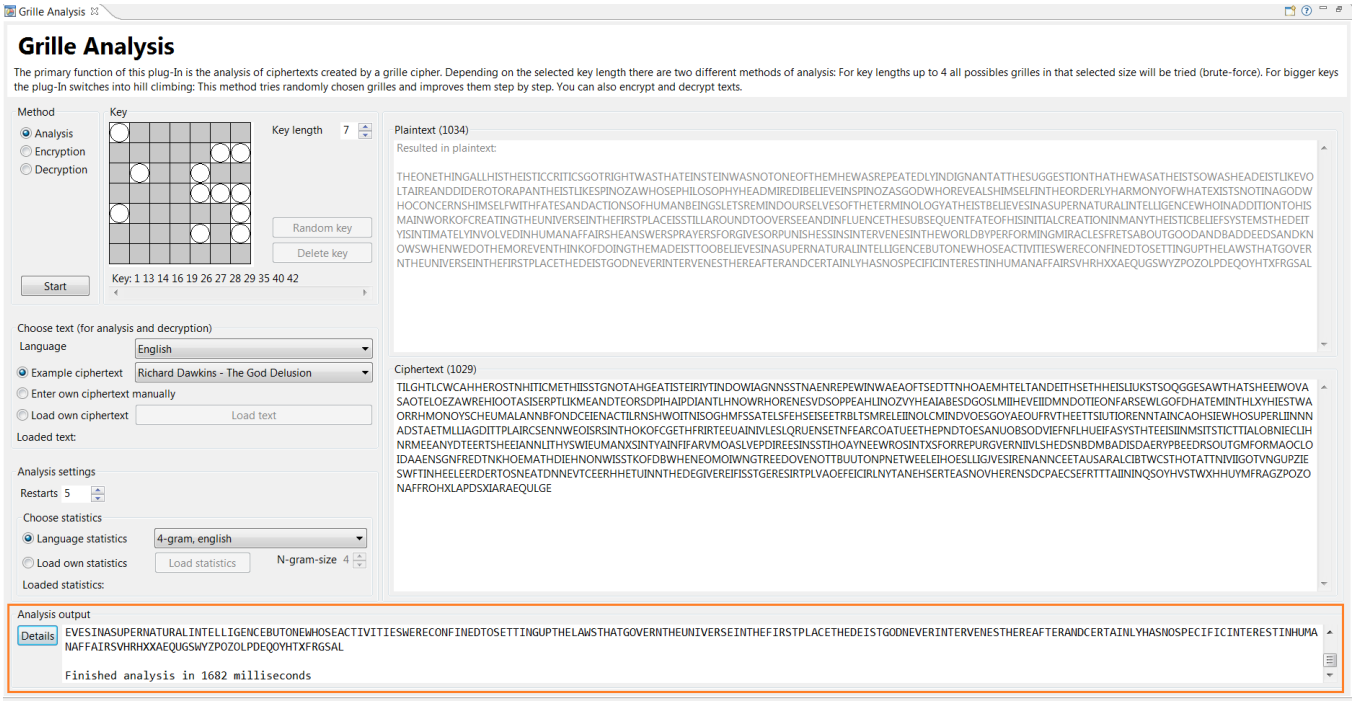
$$19 * 26^3 + 7 * 26^2 + 8 * 26^1 + 18 * 26^0 = 19 * 17576 + 7 * 676 + 8 * 26 + 18 = 338,902$$

for the index of the quadgram „This“ in an english quadgram statistic can now be carried out.

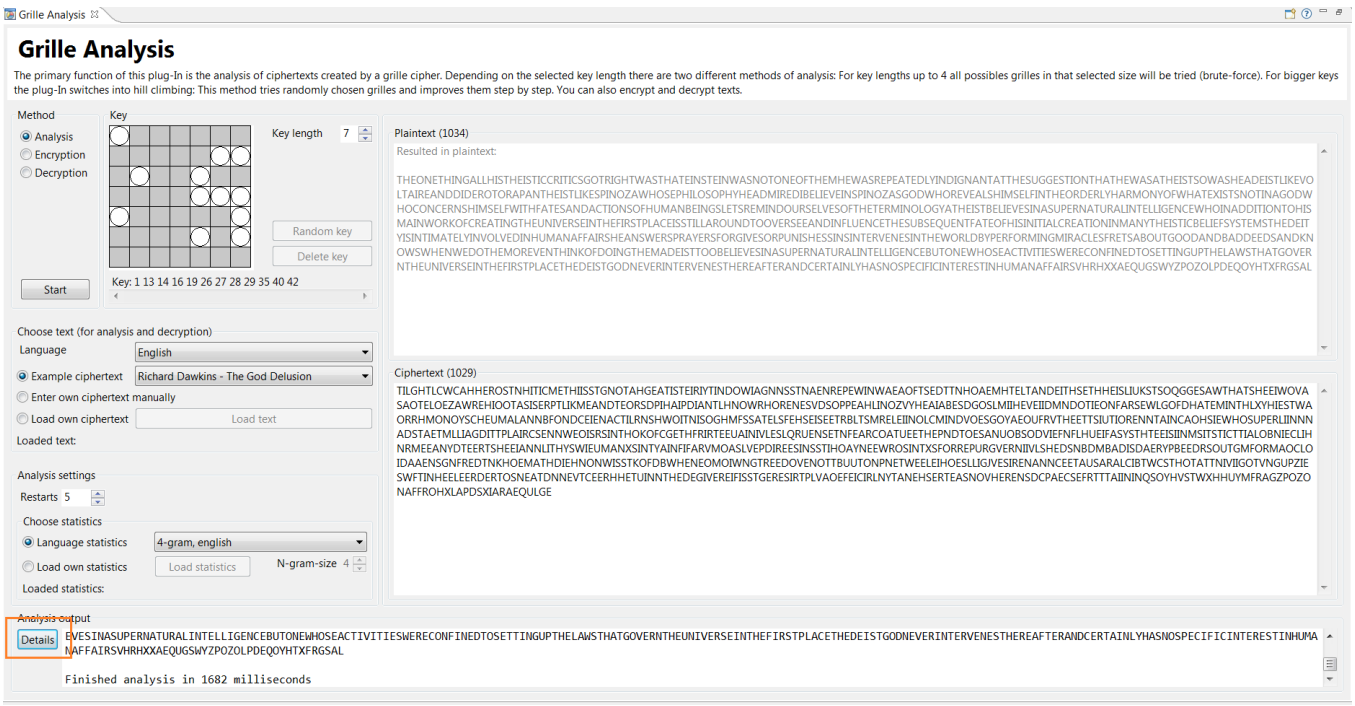
Loading customized language statistics requires manual entry of the size of the n-grams in this aggregation „Choose statistics“.

1.4 Analysis output

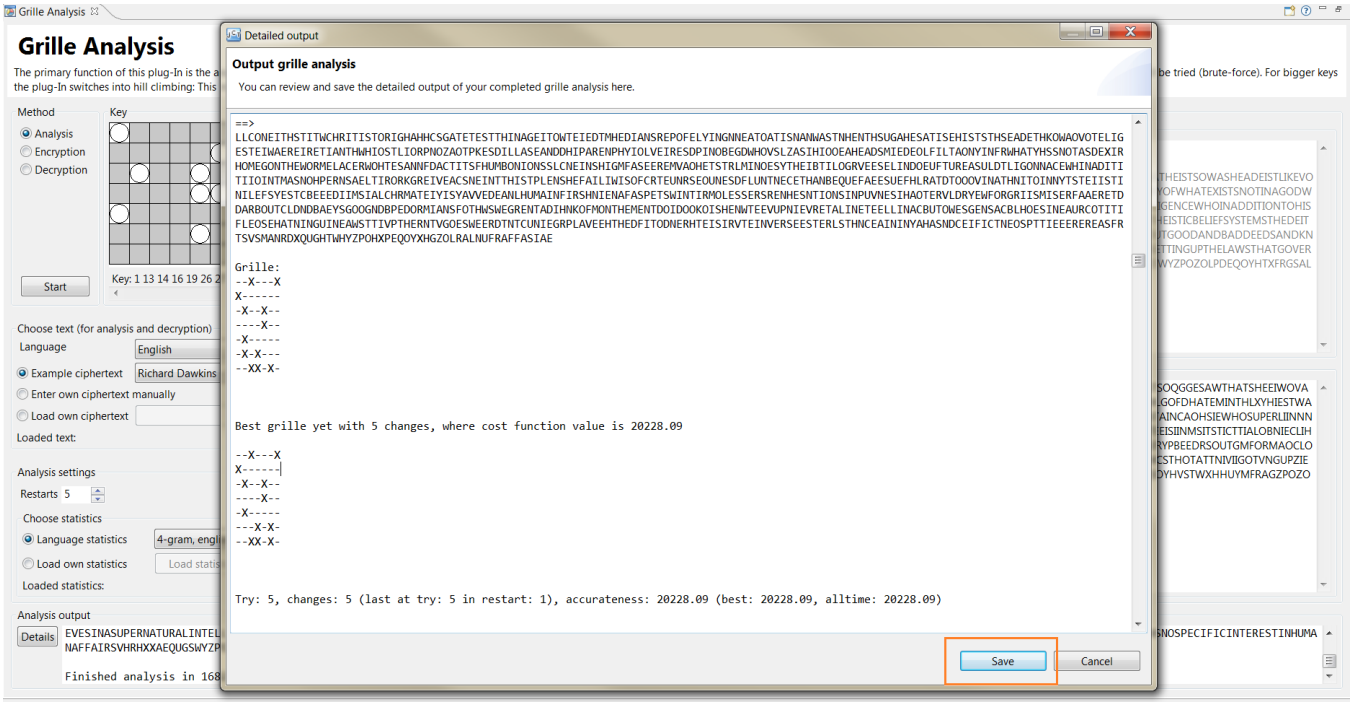
The analysis output window is located at the bottom of the screen. The selected parameters are displayed at the start of the analysis. After completing the analysis, additional information such as the key found, the resulting plaintext, and the time required for the run are displayed.



The „Details“ button on the lefthand side of the output field provides additional information.



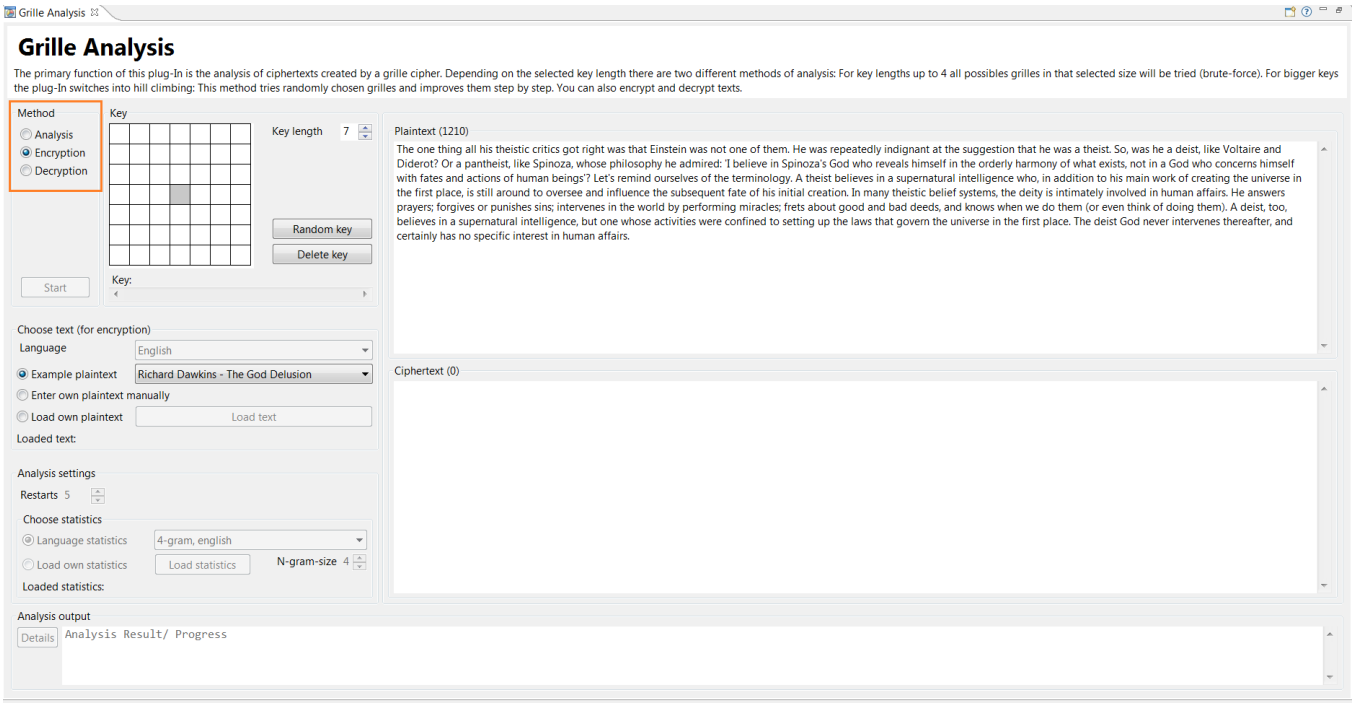
This opens a new dialog window in which information from the output window and intermediate outputs from the analysis process are displayed. This output can then be saved as a text file (*.txt).



After completition of the analysis, the key found is also displayed in the key field itself, in order to be applied, for example, directly for decryption and thus validation of the analysis result.

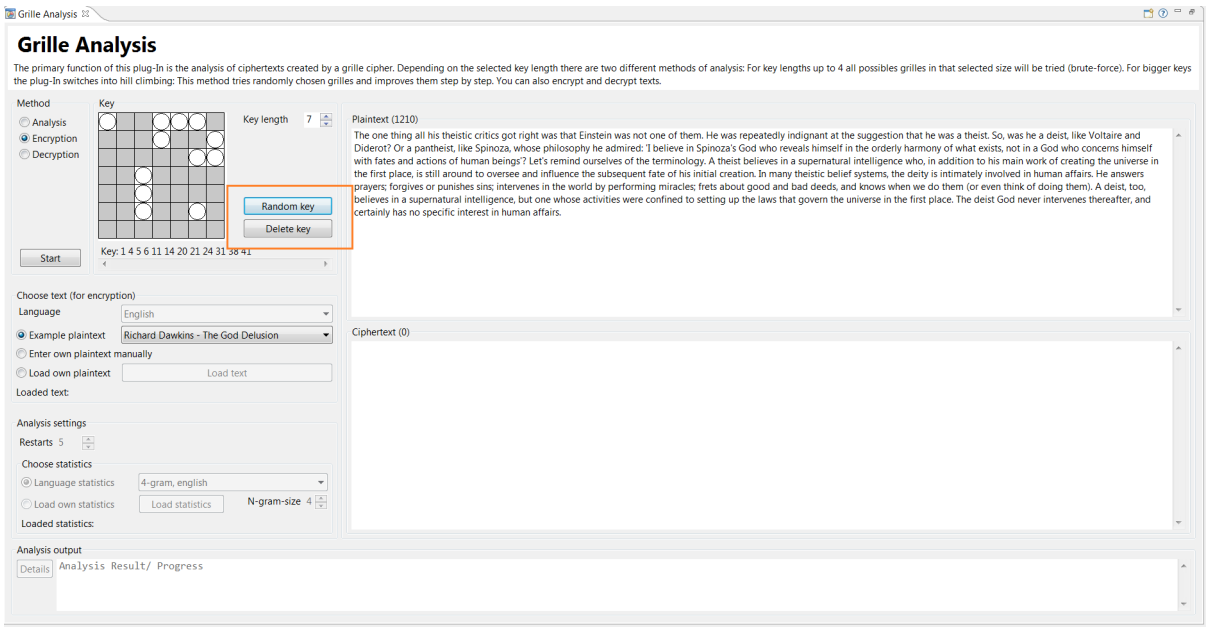
2 Encryption

In this plug-in, plaintexts can also be encrypted. For this purpose, select the function „Encryption“ in the aggregation „Method“. This entails that the buttons „Random key“ and „Delete key“ will be enabled (in case, analysis is selected, both buttons are disabled).



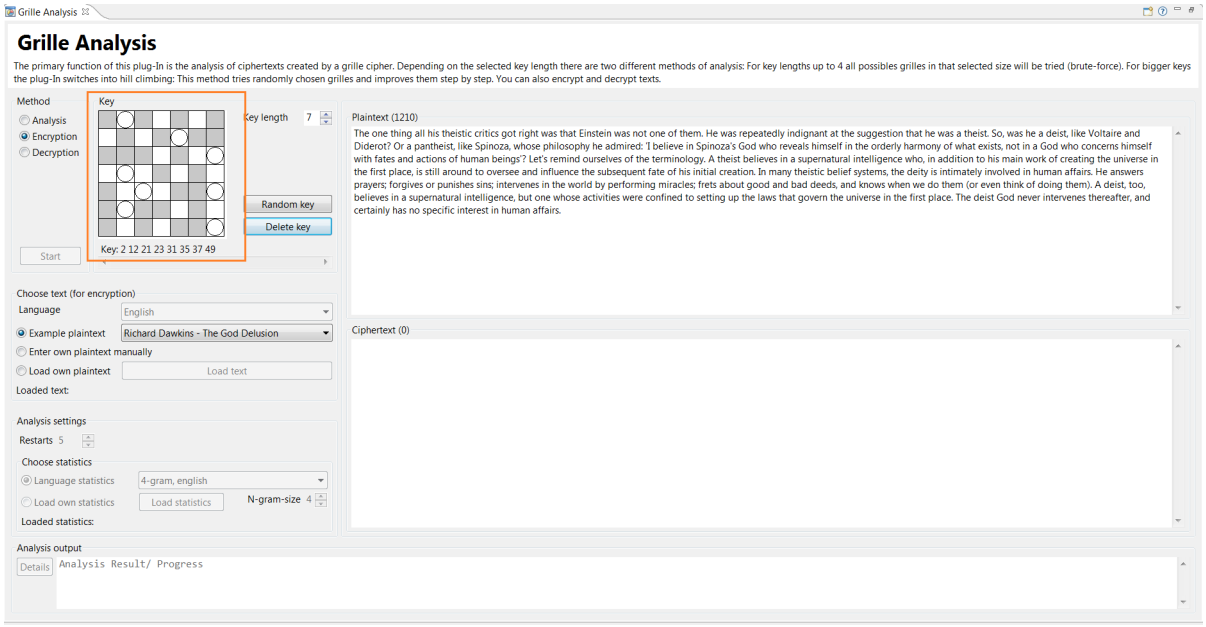
(a) Random key

By pressing the button „Random key“ a random key is generated and displayed.



(b) **Creating a key manually**

To create a key manually, select the fields in which the template should contain the holes for entering the plain text. For each selected field, the three associated fields in the other three quadrants, that are required by the rotation in the encryption process are blocked. If an already selected field is clicked again, the selection of this field is removed.



Click on the „Delete key“ button to deselect all fields simultaneously.

The selected holes in the key are also displayed in numerical form under the key field. The fields are numbered from top left to bottom right (starting with 1).

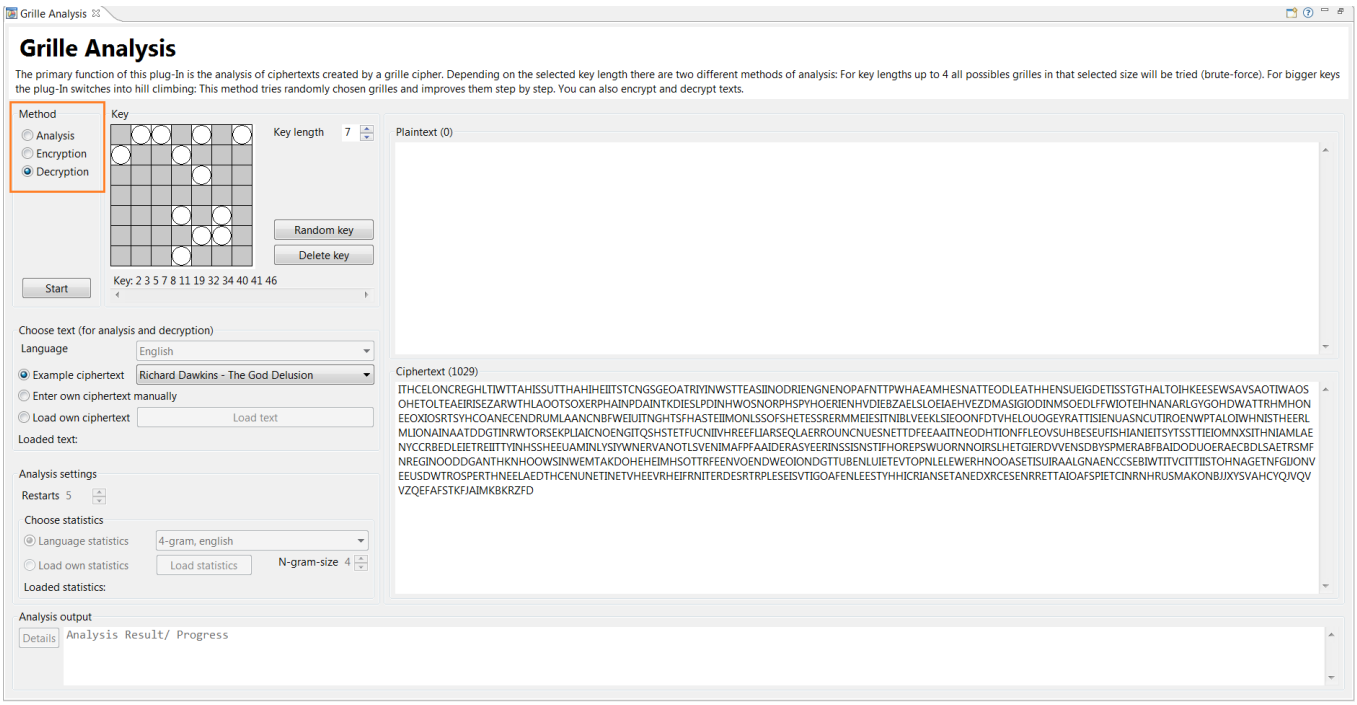
In addition to the key, a plaintext is also required for encryption. As in the analysis function, this can be selected from a number of example texts, typed in, or imported. After selecting example texts, the selected function of the plug-in is recognized and accordingly a plaintext or ciphertext is loaded into the corresponding window. This distinction must be made for manually customized texts, meaning that the plug-in does not recognize whether there is a plaintext or a ciphertext.

If there is a valid key and a text in the plain text field, the „Start“ button will be activated and the encryption can be carried out.

After changing the method selection to „Analyze“ or „Decryption“ continue with the self-generated ciphertext.

3 Decryption

As a final functionality, the plug-in offers decryption, allowing for key implementation obtained from an analysis as well as ciphertext decryption.



A valid key along with a non-empty ciphertext field and a non-empty plaintext field are required for both decryption and encryption, respectively. The operation of the text selection is analogous to the two methods already described.