Indect Block Cipher Application

User's manual

7 UE FP INDECT Project Department of Telecommunications AGH University of Science and Technology, Krakow

1 Application

ICB application is an implementation of Indect Block Cipher. The main idea of this symmetric algorithm is the unique approach to key scheme. It is still a pseudo random byte sequence but instead of XOR-ing its bits with binary data in some part of the algorithm, we use it as a base to create special AES like matrices (S-boxes). Each of those matrices represents a unique nonlinear transformation. They can be used both as substitution and permutation transformations.

1. Each of those matrices are code-able by a 64 bit chain (key values must be a multiple of 64). There are 4 key lengths chosen for practical use:

    - **128 bits** (2 S-boxes are coded: one for substitution transformation and another for permutation transformation. With a 128-bit key we propose 8 rounds of algorithm).
    - **192 bits** (3 S-boxes are coded: two for substitution transformation and one for permutation transformation. With a 192-bit key we propose that the number of rounds equals 10).
    - **320 bits** (5 S-boxes are coded: four for substitution transformation and one for permutation transformation. With a 320-bit key we propose that the number of rounds equals 12).
    - **576 bit**s (9 S-boxes are coded: eight for substitution transformation and one for permutation transformation. With a 128-bit key we propose that the number of rounds equals 14).

2. Every of those key lengths and suiting number of rounds ensures unique level of protection.

3. The IBC algorithm operates on 256 bit blocks of plain text (the plaintext is divided into 256 bit blocks).

4. Round of the algorithm is composed of substitution and permutation transformations in this particular order. Substitution operates on 8 bit parts of block and permutation operates on the whole block.

5. If the length of the last block of plain text is smaller than 256 bits it is padded with zeroes and then encrypted.

2 Interface

The visual interface of Indect Block Cipher application is presented in the Figure 2.1. The presented configuration is a default configuration that is automatically applied to the interface short after the start of the program.

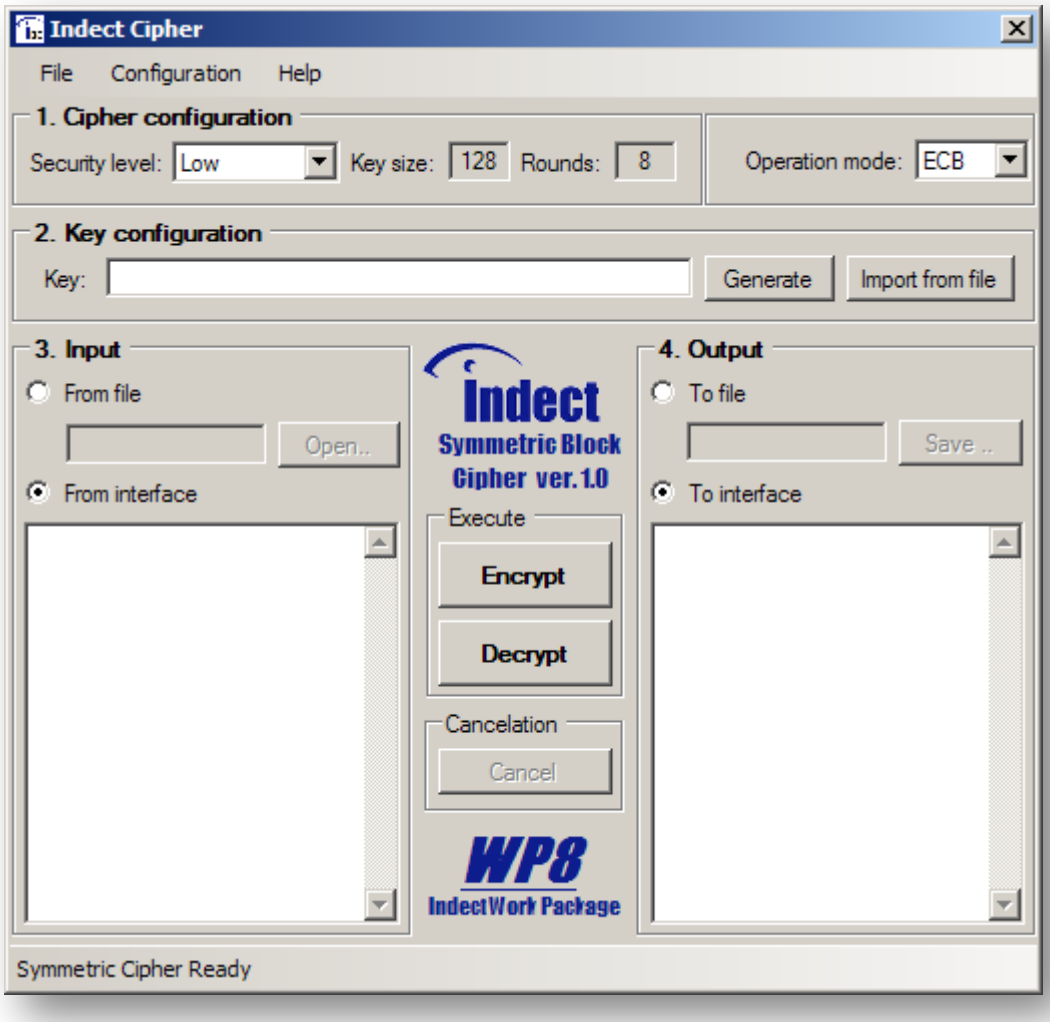

Figure 2.1: Default interface

Figure 2.2 presents example of encryption process. In the further sections each control block of the interface will be described in detail.

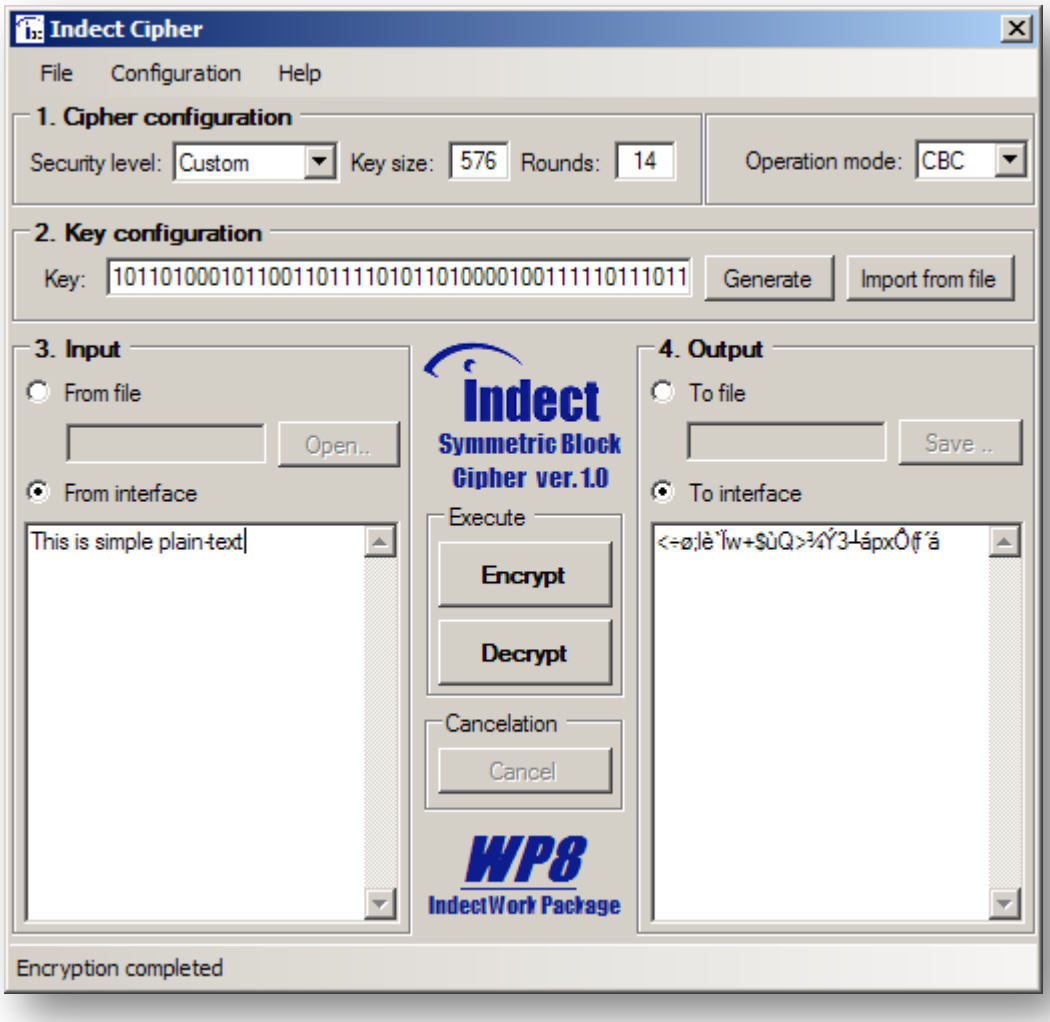

Figure 2.2: Example of configuration

2.1 Cipher configuration

Configuration of the cipher properties is the first operation that should be carried out by the user before the actual encryption. Controls that set ciphers values and operation modes are grouped in a block named “Cipher configuration” that is shown in Figure 2.3.

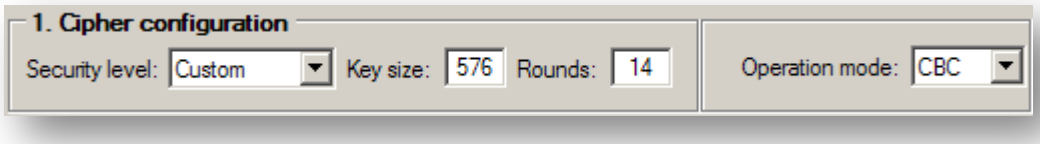

Figure 2.3: Cipher configuration

1. Security level – That combo-box corresponds to the configuration set determining the security level of encryption process. Short after choosing security level the values of key size and number of rounds are automatically set to its respective values. The user can choose between 4 defined security levels and one custom level that allows user to manually define the properties of the cipher. These levels are:

   - **Low** – key size set to 128 bits with 8 rounds of encryption process,
   - **Medium** – key size set to 192 bits with 10 rounds of encryption process,
   - **High** – key size set to 320 bits with 12 rounds of encryption process,
   - **Very High** – key size set to 576 bits with 14 rounds of encryption process,
   - **Custom** – key size and rounds controls are unlocked, and the user is allowed to manually change each value.

2. Key size – defines the size of the key in bits. Allowed values are: 128, 190, 320 and 576. This text box is automatically unlocked when the Custom security level is chosen.

3. Rounds – defines the number of rounds of encryption process. This text box is automatically unlocked when the Custom security level is chosen.

4. Operation mode – defines the method of encryption/decryption of data larger than one 256 bit block. The user can chose between the following modes:

   - **ECB** – Electronic Codebook. Each block of the data is encrypted separately and independently.
   - **CBC** – Cipher-block chaining. Provides better security but makes each block dependent of the previous one.

2.2 Key configuration

Figure 2.4 presents the configuration of the key. This block consists of an input textbox that should contain a valid key in binary format. Key can be entered manually, generated by the application or imported from a text file.

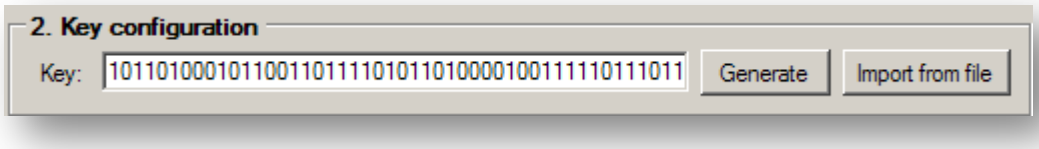

Figure 2.4: Key configuration

1. Key – textbox that should contain a valid key – that is: with a correct length, corresponding to the one configured in block “cipher configuration”, in binary format and mathematically applicable to the algorithm (the idea of encryption key was presented in the chapter 6 of deliverable D8.3 “Specification of new constructed block cipher and evaluation of its vulnerability to errors”).

2. Generate – button that invokes the generation of a valid key with length defined in block “cipher configuration”.

3. Import from file – button that allows user to choose a text file containing the key.

2.3 Input

This block represents input configuration. User is allowed to enter text manually or choose a file she/he wants to be encrypted or decrypted (Figure 2.5).

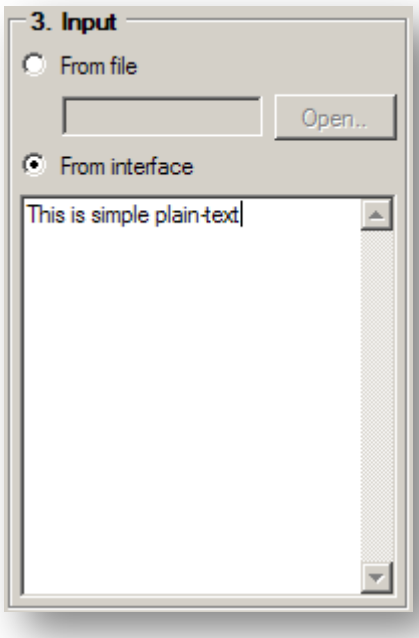

Figure 2.5: Input

1. From file – radio-button that sets container of input data to be a file,

2. File path – text-box that should contain a valid path to file,

3. Open… - button that allows user to choose a file,

4. From interface – radio-button that sets the input textbox to be the data source,

5. Input – text-box that is used to enter input data

2.4 Output

This block corresponds to the output configuration. Output can be set to be a file or a text-box. It is highly recommended to use a file as an output to the encryption process, because of the encoding differences between the set of encrypted letters and those that can be printed out on the screen. Interface output is used only to show the process, but in case of real encryption a file should be used as the destination of encryption instead (Figure 2.6).

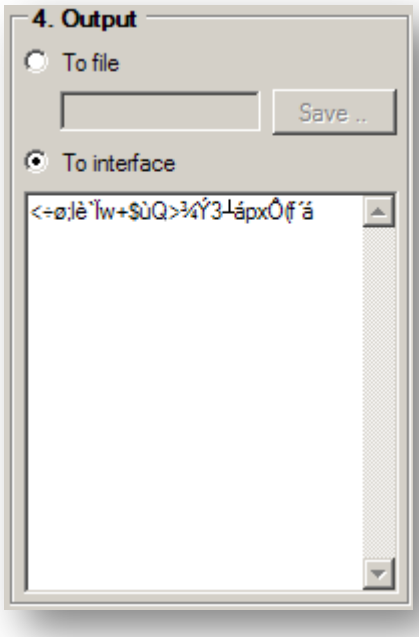

Figure 2.6: Output

1. To file – radio-button that sets the container of output data to be a file – highly recommended,

2. File path – text-box that should contain a valid path to the output file,

3. Save… - button that allows users to specify a file,

4. To interface – radio-button that sets the textbox to be the data output,

5. Output – text-box that is used to print output data – used only for demonstration purposes.

2.5 Execute

This block consists of two buttons that start the encryption or decryption process (Figure 2.7).

Figure 2.7: Execute block

1. Encrypt – button that when pressed starts the encryption process,

2. Decrypt – manually starts the decryption process.

2.6 Cancelation

This is a block containing only one button and providing user a possibility to cancel the operation and control the interface again (Figure 2.8).

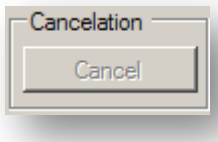

Figure 2.8: Cancelation

Cancel – button that is unlocked when the encryption or decryption process occurs. By pressing it, user stops operation and goes back to the interface.

2.7 Main Menu

This control allows user to change configuration manually without using a mouse (keyboard only). All settings are accessible and configurable through this control. Menu is categorized in three groups: File, Configuration and Help (Figure 2.9).

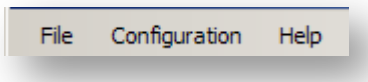

Figure 2.9: Main menu

File – Contains options:

- Clear – clears every field and sets the default configuration to the interface,
- Open file.. – allows user to choose an input file,
- Save file.. – allows user to choose an output file,
- Exit – exits the application.

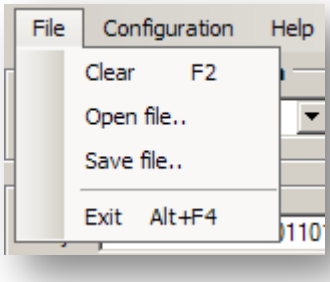

Figure 2.10: File menu

Configuration – Contains options:

- Define operation mode – provides the same choice as the operation mode combo box from the cipher configuration block.

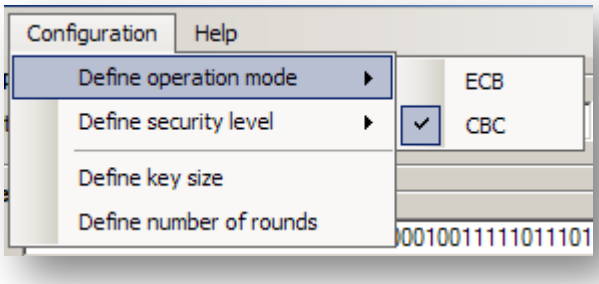

Figure 2.11: Operation mode submenu

- Define security level – provides the same choice as the security level combo box from the cipher configuration block.

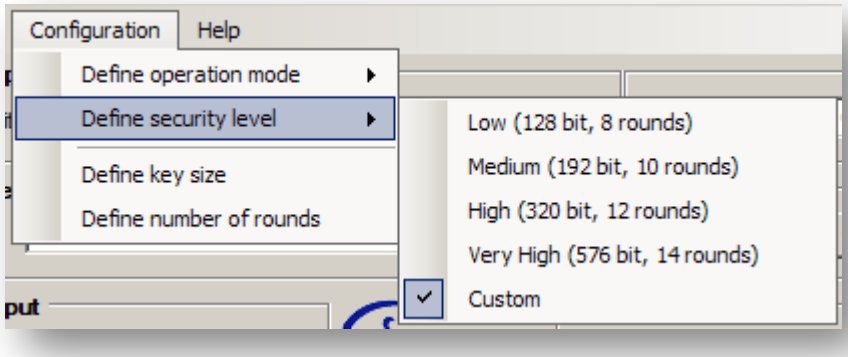

Figure 2.12: Security level submenu

- Define key size – moves the cursor to the text-box where key size is defined. This option is unlocked only when the Custom security level has been chosen.

- Define number of rounds – moves the cursor to the text-box where the number of rounds is defined. This option is unlocked only if the Custom security level has been chosen.

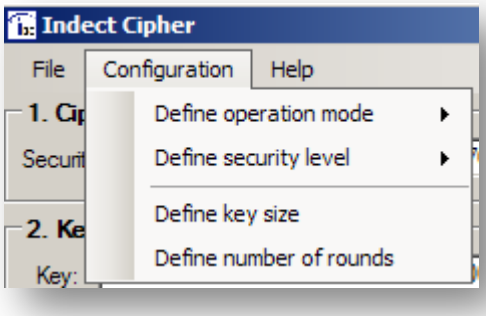

Figure 2.13: Configuration menu

Help – Contains options:

- About – opens a dialog with general information about the application and its authors,
- Help – opens a file with additional documentation,

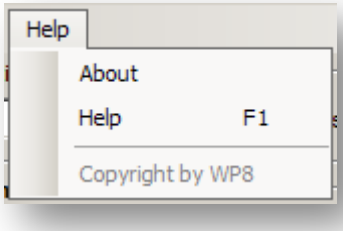

Figure 2.14: Help menu

2.8 Status bar

Status bar indicates the current state of the encryption engine. It also presents the progress of the encryption or decryption (Figure 2.15).

Figure 2.15: Status bar

Possible messages are:

- Symmetric Cipher Ready – shown after starting the application or clearing configuration,
- Encryption in progress – shown when encryption is in progress,
- Decryption in progress – shown when decryption process occurs,

- Encryption cancelled – shown when encryption was interrupted by pressing cancel button,
- Decryption cancelled – shown when decryption process was canceled by user.

Progress bar shows percentage of completion of work, and it is only visible when actual encryption or decryption occurs.

2.9 About Dialog

Presents general information about the authors, the INDECT project website, software license, organization and WP8 INDECT Work Package.

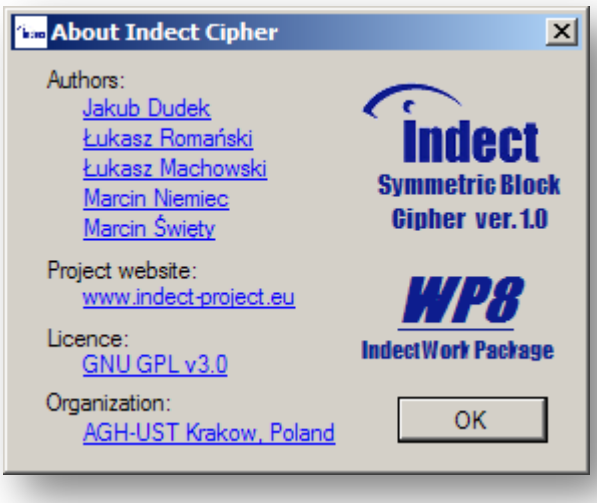

Figure 2.16: About dialog

3 Example of use

This chapter presents an example of encryption/decryption process performed by the IBC application.

3.1 Getting started

In the following section, an example usage of this application is presented with screens corresponding to each action.

1. Firstly, the user should decide on the configuration parameters of the encryption process. Let's assume that the information that we wish to encrypt is delicate very sensitive one. Let it be a file. Therefore we choose the Very High security level and CBC operation mode.

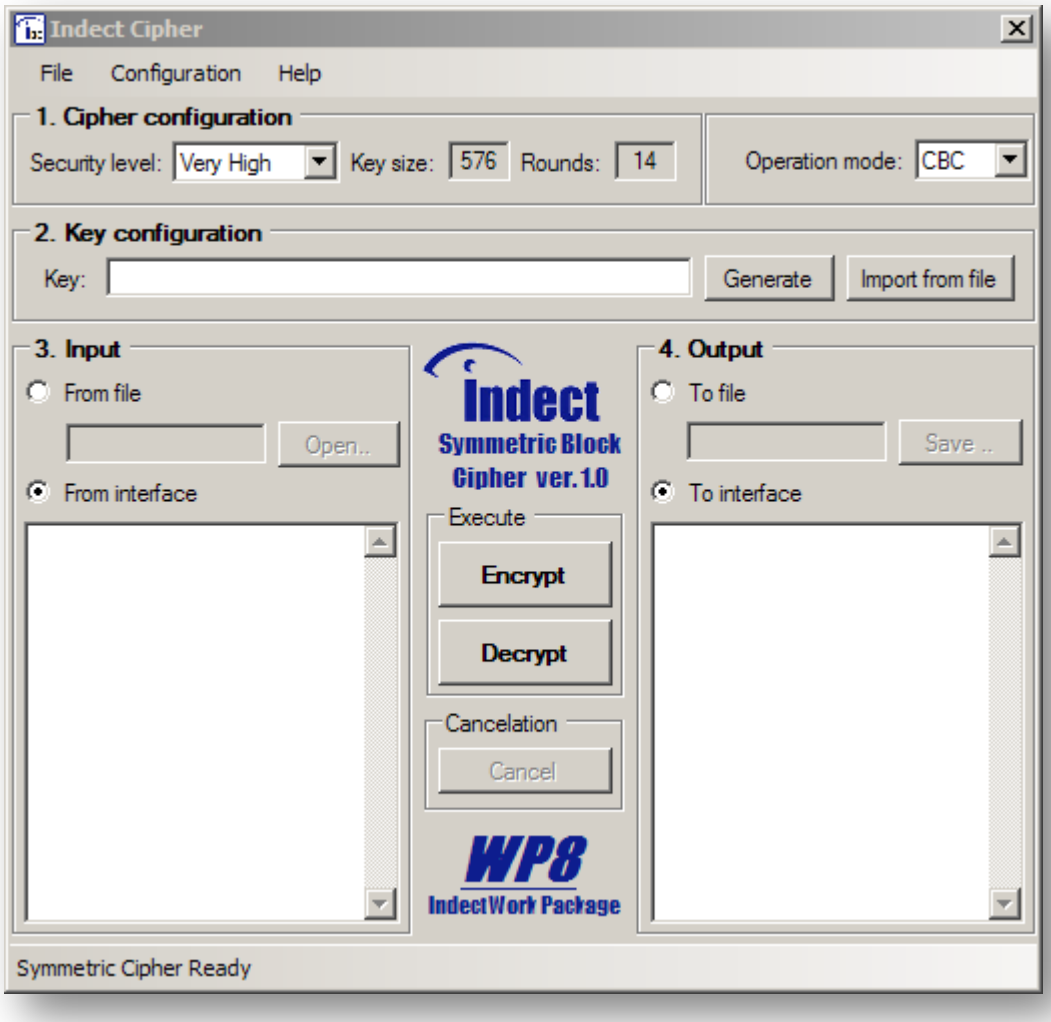

Figure 3.1: Step 1. Cipher configuration

2. Next step is key configuration. Because we are encrypting a file, we generate a 576 bit key. Then we copy this key for further decryption process.

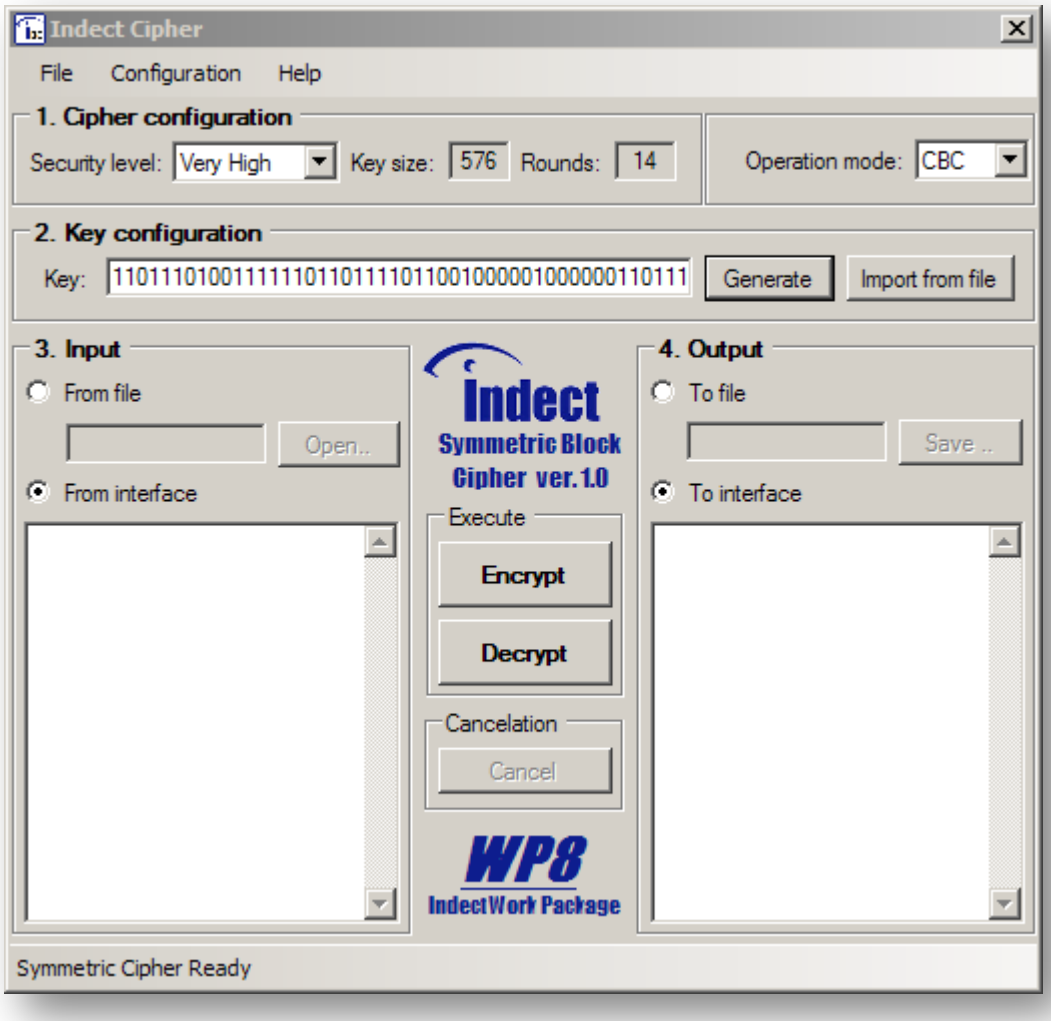

Figure 3.2: Step 2. Key configuration

3. After completing the configuration of cipher and key blocks, we should choose the file that we wish to encrypt. We choose the output to be a file. Interface automatically adds an “output” word to the file name, so we are not required to choose the new destination file.

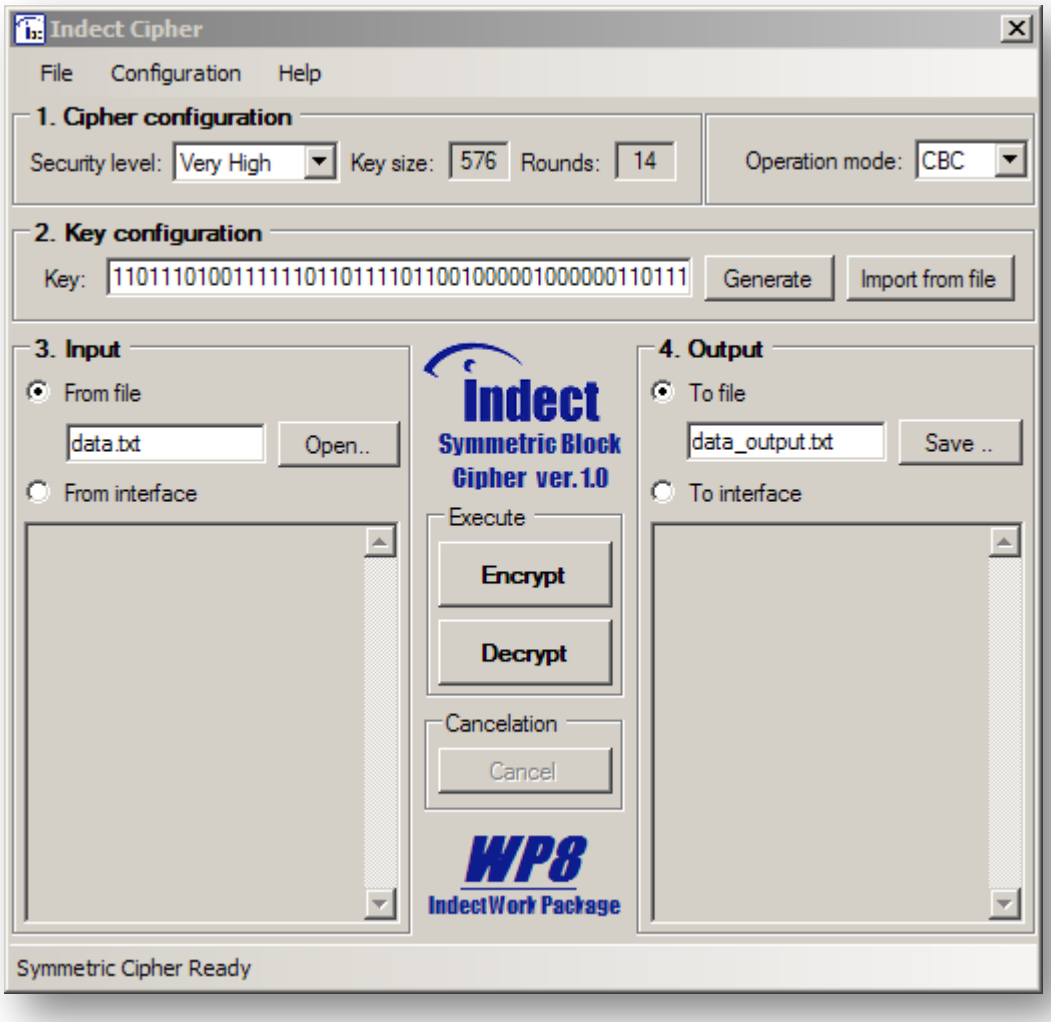

Figure 3.3: Step 3. Input/Output configuration

4. Then we press the “Encrypt” button and wait till the process ends, which is indicated by a descriptive message on the status bar.

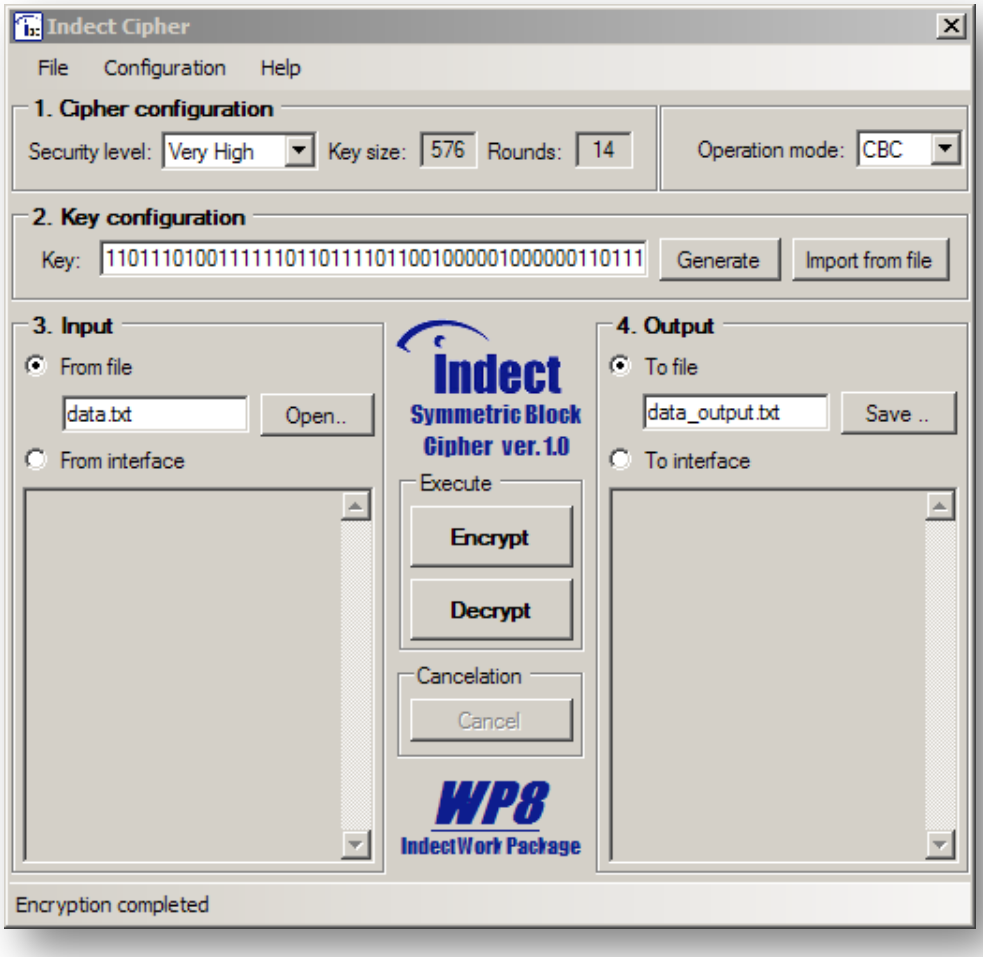

Figure 3.4: Step 4. Encryption process completed

5. After that, we have an encrypted file named “data_output.txt”. If we have saved the key somewhere, we could then delete the original file from disk and close the application.

6. When we run the application again, we set the configuration to the same state as previously shown in points 1-2.

7. Then we select the encrypted file “data_output.txt” as an input.

8. Output file should be now selected (default name would be data_output_output.txt).

9. After clicking “Decrypt” button, the process starts.

10. When process is finished, we have decrypted the file back to the plain form.