DOCUMENTATION ISG-kernel

Functional description
Decrypting and encrypting a NC program

Short Description:
FCT-C12

Documentation version: 1.2
07/11/2024

Preface

Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

Further information

Links below (DE)

https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

or (EN)

https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

Trade marks and patents

The name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

Copyright

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

➢ Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

⚠ CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

Attention

Restriction or error

This icon describes restrictions or warns of errors.

Notice

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.

Example

General example

Example that clarifies the text.

Programing Example

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

Release Note

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Table of contents

List of figures

1 Overview

Task

Controller/machine manufacturers supply encrypted NC programs that end-users are not allowed to modify and cannot view.

The NC kernel processes encrypted NC programs.

Characteristics

An encrypted NC program is recognised by its file extension. A key used for encryption and decryption must be defined for every file extension. Every file extension and the associated key define an encryption group.

A file is recognised as encrypted if its extension matches one of the encryption group file extensions. The CNC uses the associated key to decrypt the file automatically during NC program decoding.

The **program ISG Encrypter** is used.

Parameterisation

The user can define 3 different encryption groups. The keys are transmitted by CNC objects [▸ 18] to the NC kernel at controller start-up or before program start.

The file extensions assigned to the keys are configured by P-CHAN-00283 [▸ 17].

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2 Description

Initialisation

The following steps are required to use an encrypted NC program:

1. An NC program is encrypted with an individual key and saved to a folder.
2. The file extension is entered in the channel parameter list as an encrypted file type for the corresponding channel.
3. In parallel to the file extension, the associated key used to encrypt the file is entered in the NC kernel. The entry is made by using a write operation to a CNC object. This can be executed by the PLC.

Process

When an NC program is invoked, the NC kernel detects from the file extension whether it is encrypted. If the NC program is detected as encrypted, the kernel decrypts it using the specified key. If the file extension is defined as not encrypted, the program is processed as a normal NC program.

Attention

If the key is incorrect, the file is still decrypted. The NC kernel attempts to process the file and normally this then results in a syntax error.

CNC diagnosis

Entries of NC program parts in the CNC diagnostic data "diag_data.txt" are encrypted by a key from the controller manufacturer, i.e. they are not visible to users.

Flow chart

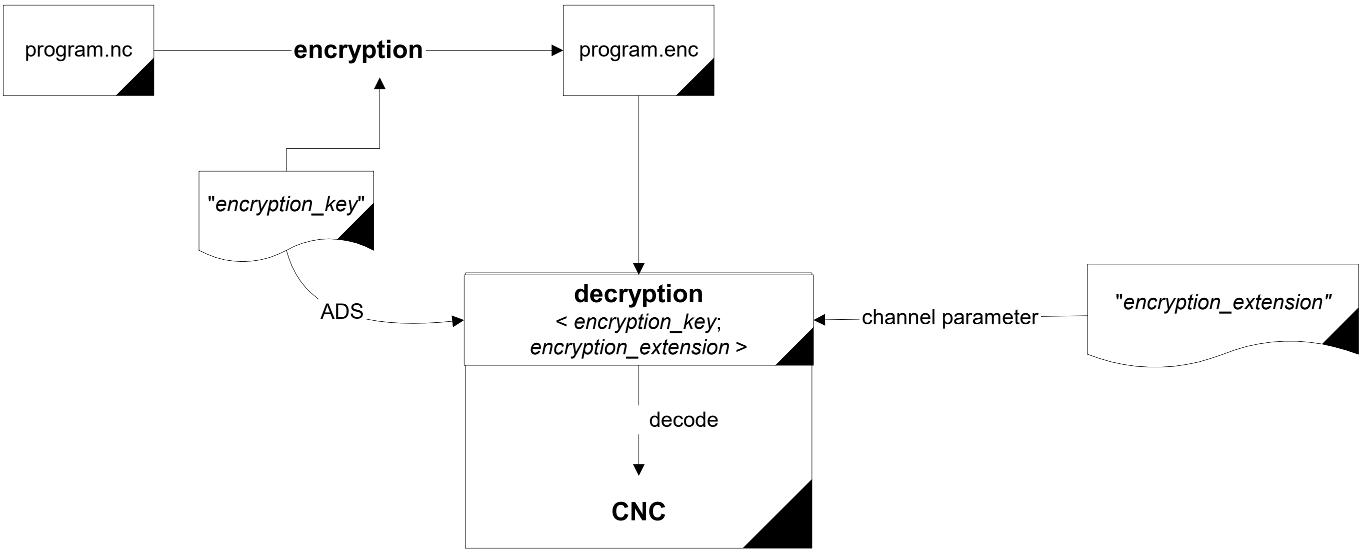


Fig. 1: Encryption/decryption flow chart of an NC program

3 Encryption groups and configuration

Groups

The user can define 3 different encryption groups for the NC kernel. Each of these groups consists of a pair comprising a key and a file extension. A key can contain a maximum of 56 characters plus '\0'.

A file extension must consist of 1 to 3 characters. When the NC kernel loads an NC program, it checks whether the extension of the NC program is entered in one of these 3 groups. If this is the case, the NC kernel decrypts the NC program with the key belonging to the associated group.

Default configuration

Users can use the groups as they wish. The pair assignment of key and file extension is depicted in the table below:

Group	Key	Channel parameters
1	mc_encryption_key_0	encryption_extension[0] P-CHAN-00283
2	mc_encryption_key_1	encryption_extension[1] P-CHAN-00283
3	mc_encryption_key_2	encryption_extension[2] P-CHAN-00283

3.1 Channel parameter list

Example

Parameterisation options for file extensions in the channel parameter list

The table below shows an example of a setting for the extensions in the channel parameter list. The file extensions for groups 1 to 3 (index 0, 1, 2) can be set.

Channel parameters	Value
encryption_extension[0]	enc
encryption_extension[1]	od
encryption_extension[2]	e

A further group also exists. This group is permanently specified by the controller or machine manufacturer and is used for the encryption of user-created NC programs (e.g. CNC cycles). These NC programs have the extension “ecy”.

Notice

It is strongly recommended that you do not re-assign the file extension “ecy” for your own definitions.

If you define the file extension “ecy”, it is not possible to use CNC cycles since NC programs encrypted by controller/machine manufacturers cannot be decrypted.

3.2 Setting keys via CNC objects

The keys for encryption groups must be set via CNC objects. Refer to the example below on how to address objects via the index group and index offset.

You can also set them online using the ISG Object Browser of the CNC.

As a security measure, all keys are only displayed hidden.

Access to encryption

All groups are accessible as shown below. The arrays can only be written via CNC objects.

Example

Channel 1

Task: COM

```
IDXGRP     :=16#00120101  ( Channel 1 )
IDXOFFS    :=16#00000094  ( mc_encryption_key_0)
IDXGRP     :=16#00120101  ( Channel 1 )
IDXOFFS    :=16#00000095  ( mc_encryption_key_1)
IDXGRP     :=16#00120101  ( Channel 1 )
IDXOFFS    :=16#00000096  ( mc_encryption_key_2)
```

ADS function block

Transfer takes place using the function block ADSWRITE(). The following applies to the example above:

```
fb_AdsWrite( NETID    :='',
             PORT     :=553,
             IDXGRP   :=16#00120101,
             IDXOFFS  :=16#00000094,
             SRCADDR  := ADR(mc_encryption_key_0),
             LEN      := SIZEOF (mc_encryption_key_0),
             WRITE    := TRUE
);
```

Attention

When writing CNC objects, note that it may be necessary to insert a "\0" at the string end.

4 Library methods

ISGEncryption.dll

This auxiliary DLL includes the methods for encrypting NC programs.

- encrypt_file()
- get_version()

Notice

The library was designed for the European/Western character set. If different character sets are used, it may result in unforeseen side effects.

4.1 Encryption

The specified input file is fully encrypted with the specified key and is saved as the output file.

long **encrypt_file** (char *pIn, char *pOut, char* encryption_key,
char* err_buffer, unsigned long err_buffer_size)

Parameter

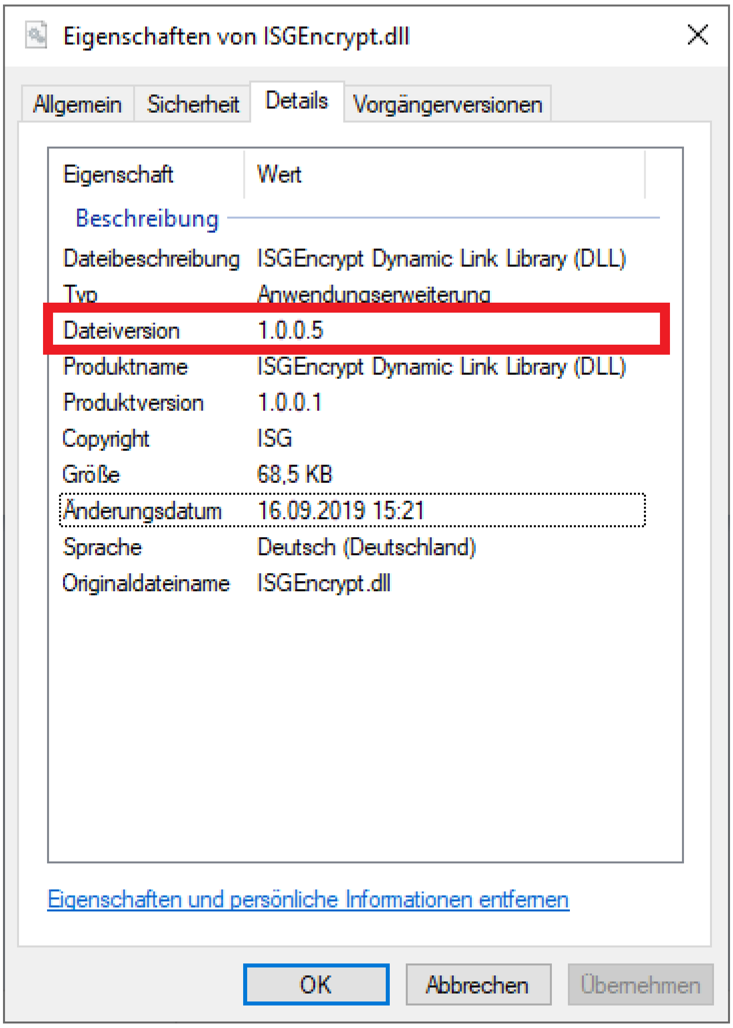
Name	Type	Meaning
pIn	char*	Name for input file
pOut	char*	Name for output
encryption_key	char*	Key
err_buffer	char*	Buffer for error messages: "Key is longer than 56 characters" "No key defined" "Could not open input file" "Could not open output file" If a blank string is entered, no error occurred. A maximum of 256 characters can be transmitted.
err_buffer_size	unsigned long	Size of the buffer err_buffer

Return values

Value	Meaning
-4	Output file cannot be opened
-3	Input file cannot be opened
-2	Maximum key length exceeded.
-1	Key missing.
0<x	Encryption of x characters successful.

4.2 Version number

The file version of the dll can be determined with the library method get_version(). This is the same version that is obtainable by right-clicking File => Properties => Details (see the figure below).


Eigenschaften von ISGEncrypt.dll
Allgemein
Sicherheit
Details
Vorgängerversionen
Eigenschaften und persönliche Informationen entfernen
OK
Abbrechen
Übernehmen

Eigenschaft	Wert
Beschreibung	
Dateibeschreibung	ISGEncrypt Dynamic Link Library (DLL)
Typ	Anwendungserweiterung
Dateiversion	1.0.0.5
Produktname	ISGEncrypt Dynamic Link Library (DLL)
Produktversion	1.0.0.1
Copyright	ISG
Größe	68,5 KB
Änderungsdatum	16.09.2019 15:21
Sprache	Deutsch (Deutschland)
Originaldateiname	ISGEncrypt.dll

Fig. 2: Determining the dll file version

Parameter

Name	Type	Meaning
buffer	unsigned char*	Buffer for version string
unsigned long	size	Size of buffer

Return values

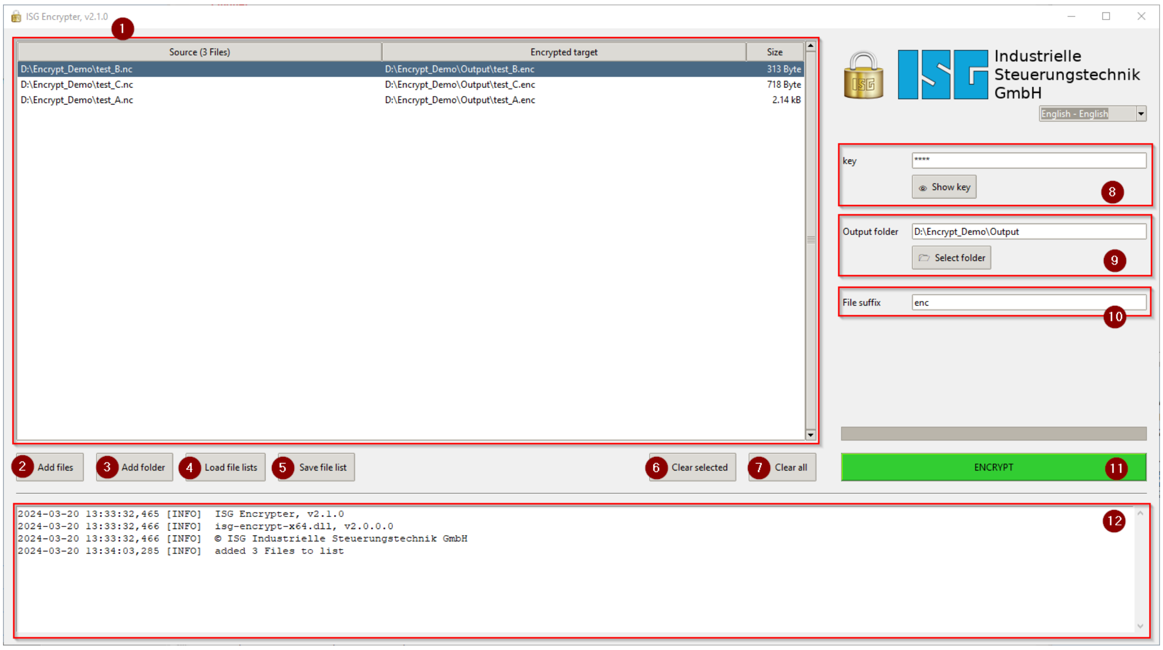
Value	Meaning
TRUE	Version string successfully copied to the buffer.

FALSE	Version string could not be written to the buffer.

5 Use of the dll in a test GUI

Example

ISG Encypt.dll in the application of the ISG Encypter



Source (3 Files)	Encrypted target	Size
D:\Encrypt_Demo\test_B.nc	D:\Encrypt_Demo\Output\test_B.enc	313 Byte
D:\Encrypt_Demo\test_C.nc	D:\Encrypt_Demo\Output\test_C.enc	718 Byte
D:\Encrypt_Demo\test_A.nc	D:\Encrypt_Demo\Output\test_A.enc	2.14 kB

key: ****

Output folder: D:\Encrypt_Demo\Output

File suffix: enc

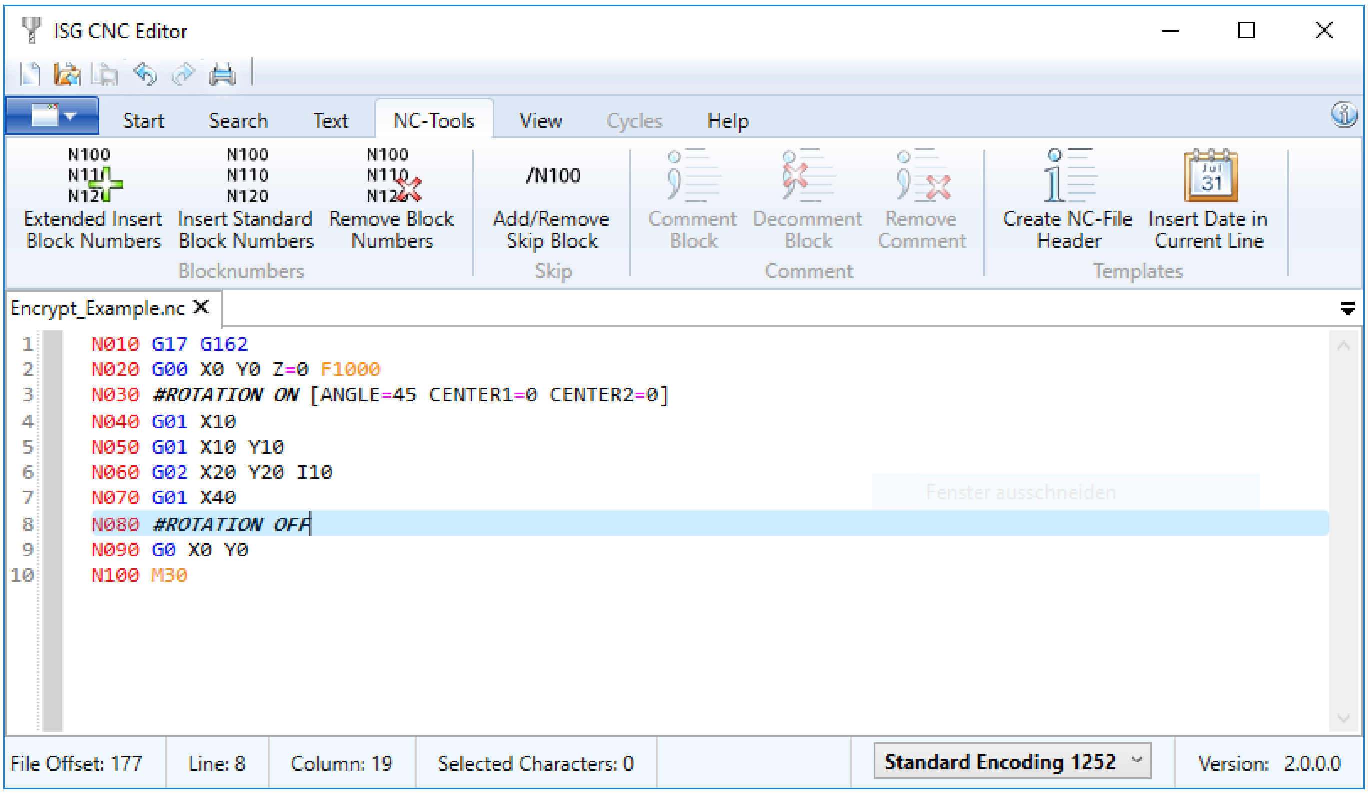
```
2024-03-20 13:33:32,465 [INFO]  ISG Encrypter, v2.1.0
2024-03-20 13:33:32,466 [INFO]  isg-encrypt-x64.dll, v2.0.0.0
2024-03-20 13:33:32,466 [INFO]  © ISG Industrielle Steuerungstechnik GmbH
2024-03-20 13:34:03,285 [INFO]  added 3 Files to list
```

Fig. 3: Overview of how to apply the ISG Encypter

Description of highlighted elements

index	Meaning	Description
1	File list	This lists all the files to be encrypted. The first column contains the file name of the source; the second column specifies the target file name.
2	Add files	Opens a dialog to add one or more files to the file list.
3	Add folder	Opens a dialog to add all the files in a folder (and all subfolders) to the file list.
4	Load file lists	Opens a dialog to load one or more file lists previously saved.
5	Save file list	Saves the current file list.
6	Remove selected	Removes selected entries from the file list.
7	Remove all	Removes all entries from the file list.
8	Key	The secret key used for encryption. The identical key must then be transferred to the CNC later, e.g. via the associated CNC object.
9	Output folder	This lets you specify a folder to save the encrypted files in. If this field remains empty, each encrypted file is stored in the folder of the associated source file.
10	File extension	Specifies the file extension used for encrypted files.
11	ENCRYPT	Starts encryption.
12	Log output	Displays information, warnings and errors.
13	Language	Changes the language of GUI elements.

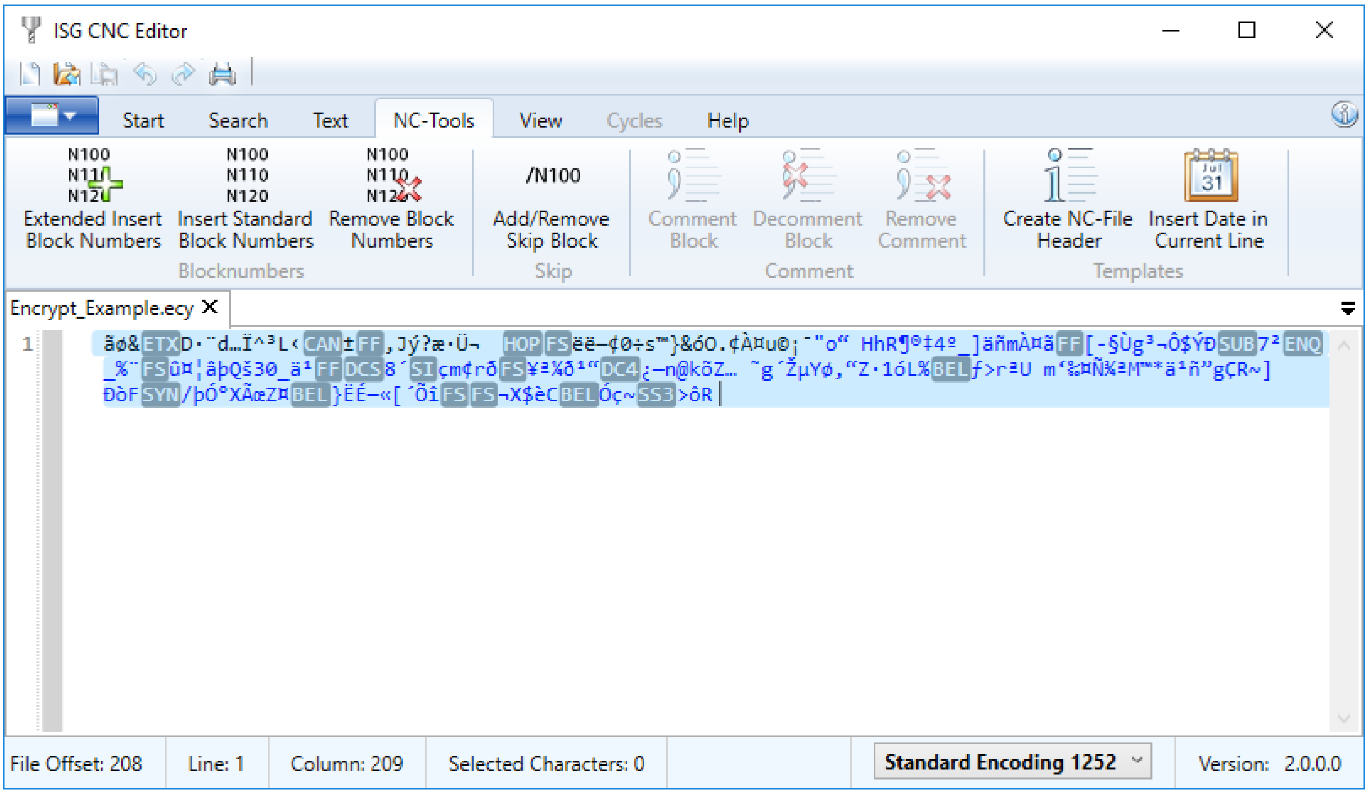
5.1 Encryption example



```
N010 G17 G162
N020 G00 X0 Y0 Z=0 F1000
N030 #ROTATION ON [ANGLE=45 CENTER1=0 CENTER2=0]
N040 G01 X10
N050 G01 X10 Y10
N060 G02 X20 Y20 I10
N070 G01 X40
N080 #ROTATION OFF
N090 G0 X0 Y0
N100 M30
```

File Offset: 177 | Line: 8 | Column: 19 | Selected Characters: 0 | Standard Encoding 1252 | Version: 2.0.0.0

Fig. 4: View of the source file with readable code



```
ãø&ETXD·¨d…Ï^³L‹CAN±FF,Jý?æ·Ü¬  HOPFSëë–¢0÷s™}&óO.¢À¤u®¡¯"o“ HhR¶®‡4º_]äñmÀ¤ãFF[-§Ùg³¬Ô$ÝÐSUB7²ENQ
_%¨FSû¤¦âþQš30_ä¹FFDCS8´SIçm¢rðFS¥ª%ð¹“DC4¿–n@kõZ… ˜g´ŽµYø,“Z·1óL%BELƒ>rªU m‘‰¤Ñ¾ªM™*ä¹ñ”gÇR~]
ÐòFSYN/þÓ°XÃæZ¤BEL}ËÉ–«[´ÕîFSFS¬X$èCBELÓç~SS3>ôR
```

File Offset: 208 | Line: 1 | Column: 209 | Selected Characters: 0 | Standard Encoding 1252 | Version: 2.0.0.0

Fig. 5: Encrypted file with encrypted code

Example

Procedure to encrypt a file

The file in the figure “View of the source file with readable code” is encrypted. ISGEncryption.dll must be in the same folder as the example application.

1. In this case, choose "asdf" as the password.
2. Select "Encrypt_Example.enc" (in the same folder as the input file) as the output file.
3. The result should be similar to the figure “Encrypted files”.

6 Parameter

6.1 Channel parameters

P-CHAN-00283	Define file extensions to encrypt NC programs
Description	The NC channel can process encrypted NC programs. Encryption is recognised by the file extension. A maximum of 3 self-defined file extensions are available in the channel parameter 'encryption_extension[...]' to configure file extensions. A file extension can consist of one to maximum 3 characters. No distinction is made between uppercase and lowercase letters in the file extension. A check is made whether the extension is entered in one of the 3 groups before opening an NC program. If the check is positive, the NC kernel decrypts the NC program with the key belonging to the related group. Both main programs and global subroutines can be encrypted. For more information about encryption see [FCT-C12 [▶ 6]].
Parameter	encryption_extension[i] where i = 0 2
Data type	STRING
Data range	Maximum of 3 characters
Dimension	----
Default value	*encryption_extension[0]* ---- *encryption_extension[1]* ---- *encryption_extension[2]* ---- *encryption_extension[3]* ecy *

Remarks	* File extensions can be set for the groups 1 to 3 (Index 0, 1, 2). A further group also exists. This group especially is pre-defined by the controller or machine manufacturer and is used for the encryption of self-created NC programs (e.g. cycles). The extension is 'ecy'. It is recommended not to re-use this extension for new user-defined definitions
	Parameterisation example:
	`encryption_extension[0] enc (1st group)`
	`encryption_extension[1] od (2nd group)`
	`encryption_extension[2] d (3rd group)`

6.2 CNC objects

Notes on addressing

<C_{ID}> Channel or channel ID starting with 1

For further information on addressing CNC objects, see [FCT-C13//Description].

Name	**mc_encryption_key_0**		
Description	This object specifies the key for the first encryption group. The encryption group is defined by the parameter P-CHAN-00283 [▸ 17] and refers to the specified file extension. This key acts on encryption_extension[**0**]		
Task	COM (Port 553)		
Indexgruppe	0x12010<C_{ID}>	**Index offset**	0x94
Data type	STRING	**Length/byte**	57
Attributes	write	**Unit**	-
Remarks			

Name	**mc_encryption_key_1**		
Description	This object specifies the key for the second encryption group. The encryption group is defined by the parameter P-CHAN-00283 and refers to the specified file extension. This key acts on encryption_extension[**1**]		
Task	COM (Port 553)		
Indexgruppe	0x12010<C_{ID}>	**Indexoffset**	0x95
Data type	STRING	**Length/byte**	57
Attributes	write	**Unit**	-
Remarks			

Name	**mc_encryption_key_2**		
Description	This object specifies the key for the third encryption group. The encryption group is defined by the parameter P-CHAN-00283 [▸ 17] and refers to the specified file extension. This key acts on encryption_extension[**2**]		
Task	COM (Port 553)		
Indexgruppe	0x12010<C_{ID}>	**Indexoffset**	0x96
Data type	STRING	**Length/byte**	57
Attributes	write	**Unit**	-
Remarks			

7 Appendix

7.1 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de. The latest documentation is posted in our Online Help (DE/EN):

QR code link: https://www.isg-stuttgart.de/documentation-kernel/

The link above forwards you to:

https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html

Notice

Change options for favourite links in your browser;

Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.

We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for download:

DE:
https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

EN:
https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

E-Mail: documentation@isg-stuttgart.de

Index