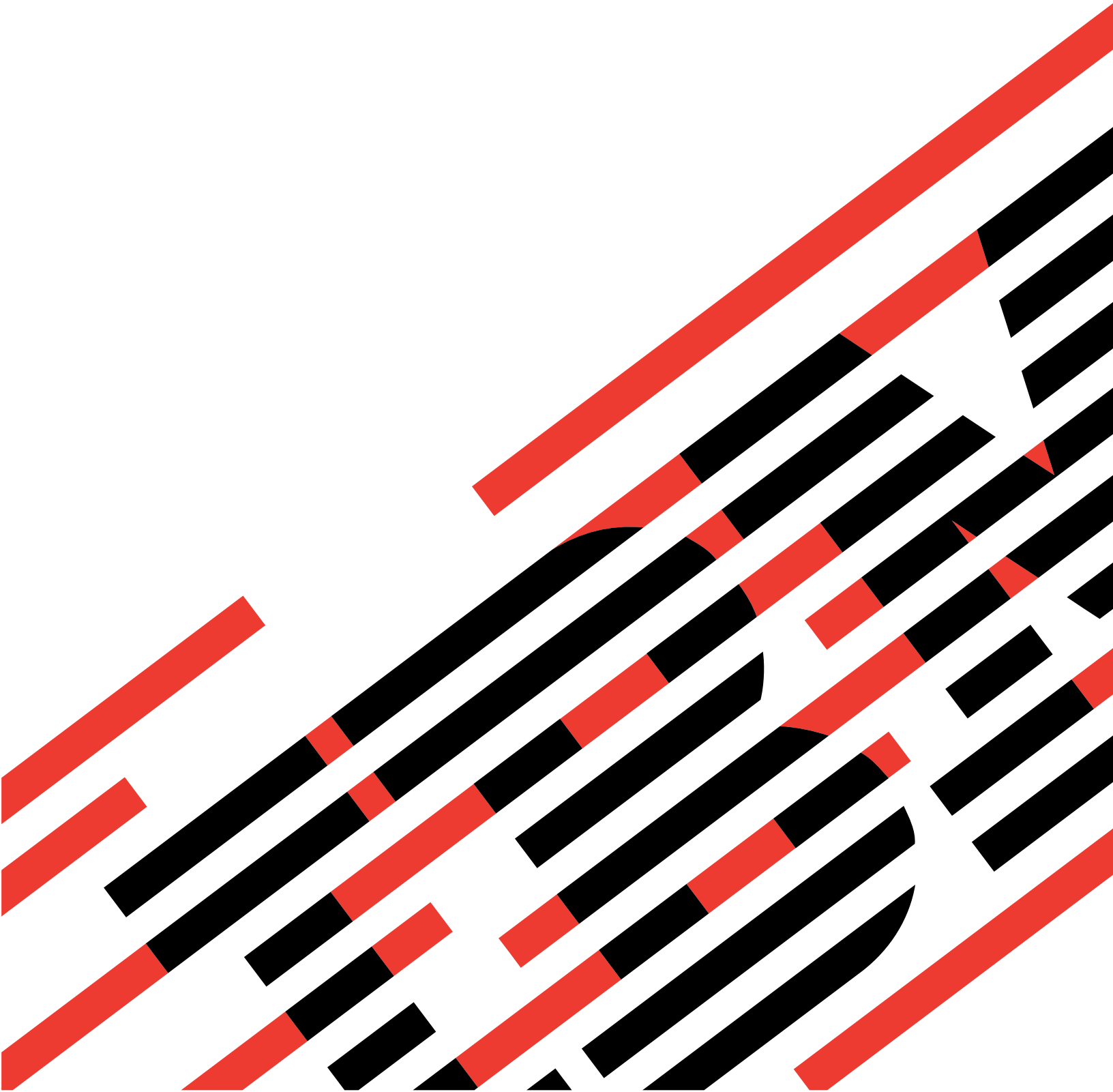
IBM

eserver

iSeries

Cryptographic Support for AS/400 Commands

Version 5 Release 3



IBM

@server

iSeries

Cryptographic Support for AS/400 Commands

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 39.

First Edition (May 2004)

This edition applies to version 5, release 3, modification 0 of Cryptographic Support for AS/400 (product number 5722-CR1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CICS models.

Contents

Add Cross-Domain Key (ADDCRSDMNK) 1

Change Cross-Domain Key (CHGCRSDMNK) 3

Change Master Key (CHGMSTK) 5

Cipher Data (CPHDTA) 7

Encrypt Cipher Key (ENCCPHK) 11

Encipher From Master Key (ENCFRMMSTK) 13

Encipher To Master Key (ENCTOMSTK) 15

Generate Cipher Key (GENCPHK) . . . 17

Generate Cross-domain Keys (GENCRSDMNK) 19

Generate MAC (GENMAC) 21

Generate PIN (GENPIN) 25

Remove Cross-Domain Key (RMVCRSDMNK) 27

Set Master Key (SETMSTK) 29

Translate PIN (TRNPIN). 31

Verify Master Key (VFYMSTK) 33

Verify PIN (VFYPIN) 35

Appendix. Notices 39

Add Cross-Domain Key (ADDCRSDMNK)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Add Cross-Domain Key (ADDCRSDMNK) command allows you to specify a new cross-domain key that is added into the cross-domain key table.

This command prompts you for the name of the key, its use, and the key value.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

None

Top

Error messages

Unknown

Top

Change Cross-Domain Key (CHGCRSDMNK)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Change Cross-Domain Key (CHGCRSDMNK) command allows you to change the value of a cross-domain key in the key table.

This command prompts you for the name of the key, its use, and the new key value.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

None

Top

Error messages

Unknown

Top

Change Master Key (CHGMSTK)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Change Master Key (CHGMSTK) command displays the Change Master Key prompt. Using this prompt, you can change the host master key on the system. You supply the values for two key parts, which are exclusive-ORed together to produce the new host master key. The new host master key must have odd parity in every byte.

The values in the cross-domain key table are reencrypted under the new host master key value.

To use this command, the system must be in a restricted state; enter the End System (ENDSYS) command to get to a restricted state. Once the master key is installed, you can return your system to a normal state by restarting your subsystems.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

None

Top

Error messages

Unknown

Top

Cipher Data (CPHDTA)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Cipher Data (CPHDTA) command is used to encrypt or decrypt a variable length of data.

Top

Parameters

Keyword	Description	Choices	Notes
DATA	Input data	*Character value*	Required, Positional 1
DTALEN	Input data length	1-65528	Required, Positional 2
CPHK	Cipher key	*Character value*	Required, Positional 3
RTNVAR	CL var for returned value	*Character value*	Required, Positional 4
OPTION	Option	*ECPH, *DCPH, *SCPH	Required, Positional 5
KTYPE	Key type	***CLEAR**, *CIPHER	Optional, Positional 6
CHAIN	Chain option	***NO**, *YES	Optional, Positional 7
ICV	Initial chaining value	*Character value*, ***NONE**	Optional, Positional 8
PAD	Pad option	***NO**, *YES	Optional, Positional 9
PADCHAR	Pad character	*Character value*, **X'00'**	Optional, Positional 10
RTNDTALEN	CL var for returned length	*Decimal number*	Optional, Positional 11

Top

Input data (DATA)

Specifies the data or, a variable containing the data, to be encrypted or decrypted. The data must be at least as long as the length specified in the **Input data length** prompt (DTALEN parameter). If *SCPH (specific encryption) is specified in the **Option** prompt (OPTION parameter), only values 0-9 and A-F may be specified in this parameter. This is a required parameter.

Top

Input data length (DTALEN)

Specifies the length or a variable containing the length of the data to be encrypted or decrypted.

When encrypting data, the length must be a multiple of 8 bytes if *NO is specified for both the **Pad option** prompt (PAD parameter) and the **Chain option** parameter (CHAIN parameter). The length must be less than 65,528 if the **Pad option** prompt (PAD parameter) is specified as *YES. The length must be less than or equal to 65,528 if the **Pad option** prompt (PAD parameter) is specified as *NO. If *SCPH (specific encryption) is specified in the **Option** prompt (OPTION parameter), this value must be 16.

When decrypting data, the length must be a multiple of 8 bytes unless cipher block chaining was used when the data was encrypted. The length must be less than or equal to 65,528. This is a required parameter.

Top

Cipher key (CPHK)

Specifies an 8-byte value, or a variable containing an 8-byte value, to be used as the key for the Data Encryption Algorithm. There are no restrictions on the value of this parameter. This is a required parameter.

Top

CL var for returned value (RTNVAR)

Specifies a variable to receive the results of the cipher operation. If *YES is specified for the **Pad option** (PAD parameter) and *ECPH is specified for the **Option** (OPTION parameter), the variable must be at least as long as the next 8-byte multiple past the value specified for the **Input data length** prompt (DTALEN parameter). Otherwise, it must be at least as long as the value specified for the **Input data length** prompt (DTALEN parameter). This is a required parameter.

Top

Option (OPTION)

Specifies the cipher function to be performed. This is a required parameter.

The possible values are:

***ECPH**
: A copy of the data in the **Input data** prompt (DATA parameter) is encrypted and placed in the variable specified in the **CL var for returned value** prompt (RTNVAR parameter)

***SCPH**
: The **Input data** prompt (DATA parameter) contains 16 hexadecimal characters representing 8 bytes of data to encrypt (for example, 'C1' represents 'A'). A copy of these 16 hexadecimal characters is converted to an 8-byte field and encrypted. The resulting 8 bytes of ciphertext are converted back to 16 hexadecimal characters and placed in the **CL var for returned value** prompt (RTNVAR parameter).

***DCPH**
: A copy of the data in the **Input data** prompt (DATA parameter) is decrypted and placed in the variable specified in the **CL var for returned value** prompt (RTNVAR parameter).

Top

Key type (KTYPE)

Specifies the format of the key specified in the **Cipher key** prompt (CPHK parameter).

The possible values are:

***CLEAR**
The cipher key is specified in plaintext. If *SCPH is specified for the **Option** prompt (OPTION parameter), the value specified must be *CLEAR.

***CIPHER**
The cipher key is enciphered under the host master key. If *YES is specified for the **Chain option** prompt (CHAIN parameter) or the **Pad option** prompt (PAD parameter), the value specified must be *CIPHER.

Top

Chain option (CHAIN)

Specifies whether cipher block chaining is to be used during the cipher operation.

The possible values are:

***NO** Cipher block chaining will not be used. If *SCPH is specified for the **Option** prompt (OPTION parameter) or *CLEAR is specified for the **Key type** prompt (KTYPE parameter), *NO must be specified.

***YES** Cipher block chaining will be used.

Top

Initial chaining value (ICV)

Specifies an 8-byte value, or a variable containing an 8-byte value, to be used as the initial chaining value when performing cipher block chaining. This parameter is ignored if *NO is specified for the **Chain option** prompt (CHAIN option).

The possible values are:

***NONE**
There is no initial chaining value.

initial-chaining-value
Specify the value to be used as the initial chaining value for cipher block chaining. There are no restrictions on the value for this parameter. A value must be specified if *YES is specified for the **Chain option** prompt (CHAIN option).

Top

Pad option (PAD)

Specifies whether padding is to be performed.

The possible values are:

***NO** Padding will not be performed. *NO must be specified if *SCPH is specified for the **Option** prompt (OPTION parameter).

***YES** Before encrypting, the data is padded out to the next 8-byte multiple using the pad character specified in the **Pad character** prompt (PADCHAR parameter). The last byte is then replaced with a count of the number of pad characters.

After the data is decrypted, the pad characters and count byte are left appended to the data. The length of the data without the pad characters is returned in the variable specified in the **CL var for returned length** prompt (RTNDTALEN parameter).

Top

Pad character (PADCHAR)

Specifies a 1-byte value, or a variable containing a 1-byte value, to be used as the pad character when *ECPH is specified for the **Option** prompt (OPTION parameter) and *YES is specified for the **Pad option** prompt (PAD parameter). The default (1-byte pad character) is hex 00.

Top

CL var for returned length (RTNDTALEN)

Specifies a variable to receive the length of the returned data. When *DCPH is specified for the **Option** prompt (OPTION parameter) and *YES is specified for the **Pad option** prompt (PAD parameter), this length is the length of the deciphered data without padding. Otherwise, this variable contains the length of the data placed in the variable specified in the **CL var for returned value** prompt (RTNVAR parameter).

Top

Examples

None

Top

Error messages

Unknown

Top

Encrypt Cipher Key (ENCCPHK)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Encrypt Cipher Key (ENCCPHK) command enciphers a plaintext data-encrypting key under the host master key.

Top

Parameters

Keyword	Description	Choices	Notes
CLRK	Clear key value	*Character value*	Required, Positional 1
CPHK	CL var for encrypted key	*Character value*	Required, Positional 2

Top

Clear key value (CLRK)

Specifies an 8-byte value, or a variable containing an 8-byte value, of a plaintext data-encrypting key. There are no restrictions on the value of this parameter. This is a required parameter.

Top

CL var for encrypted key (CPHK)

Specifies an 8-byte variable to receive the value of the data-encrypting key specified in the **Clear key value** prompt (CLRK parameter) after it is encrypted under the host master key. This is a required parameter.

Top

Examples

None

Top

Error messages

Unknown

Top

Encipher From Master Key (ENCFRMMSTK)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Encipher From Master Key (ENCFRMMSTK) command reenciphers a data-encrypting key from encryption under the host master key to encryption under one or two sending cross-domain keys.

Top

Parameters

Keyword	Description	Choices	Notes
CPHK	Enciphered key value	*Character value*	Required, Positional 1
CRSDMNK1	*SND cross-domain key name	*Name*	Required, Positional 2
KRTNVAR1	CL var for re-encrypted key	*Character value*	Required, Positional 3
CRSDMNK2	*SND cross-domain key name	*Name*, ***NONE**	Optional, Positional 4
KRTNVAR2	CL var for re-encrypted key	*Character value*	Optional, Positional 5

Top

Enciphered key value (CPHK)

Specifies an 8-byte value, or a variable containing an 8-byte value, that is the value of a data-encrypting key enciphered under the host master key. There are no restrictions on the value of this parameter. This is a required parameter.

Top

*SND cross-domain key name (CRSDMNK1)

Specifies the name, or a variable containing the name, of a sending cross-domain key. The value in the **Enciphered key value** prompt (CPHK parameter) is decrypted using the host master key and encrypted using this sending cross-domain key. This is a required parameter.

Top

CL var for re-encrypted key (KRTNVAR1)

Specifies an 8-byte variable to receive the data-encrypting key encrypted under the sending cross-domain key. This is a required parameter.

Top

*SND cross-domain key name (CRSDMNK2)

Specifies the name of a second sending cross-domain key. The value in the **Enciphered key value** prompt (CPHK parameter) is decrypted using the host master key and encrypted using this second sending cross-domain key.

The possible values are:

***NONE**
No sending cross-domain key is to be used to encrypt the data-encrypting key.

sending-cross-domain-key-name
Specify the name, or a variable containing the name, of a sending cross-domain key.

Top

CL var for re-encrypted key (KRTNVAR2)

Specifies an 8-byte variable to receive the data-encrypting key encrypted under the second sending cross-domain key.

Top

Examples

None

Top

Error messages

Unknown

Top

Encipher To Master Key (ENCTOMSTK)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Encipher To Master Key (ENCTOMSTK) command reenciphers a data-encrypting key from encryption under a receiving cross-domain key to encryption under the host master key.

Top

Parameters

Keyword	Description	Choices	Notes
CPHK	CL var for re-encrypted key	*Character value*	Required, Positional 1
CRSDMNK	*RCV cross-domain key name	*Name*	Required, Positional 2
ENCCPHK	Enciphered key value	*Character value*	Required, Positional 3

Top

CL var for re-encrypted key (CPHK)

Specifies an 8-byte variable to receive the data-encrypting key encrypted under the host master key. This is a required parameter.

Top

*RCV cross-domain key name (CRSDMNK)

Specifies the name, or a variable containing the name, of a receiving cross-domain key. The value in the **Enciphered key value** prompt (ENCCPHK parameter) is decrypted using the receiving cross-domain key and encrypted using the host master key. This is a required parameter.

Top

Enciphered key value (ENCCPHK)

Specifies the 8-byte value, or a variable containing the 8-byte value, of a data-encrypting key enciphered under the receiving cross-domain key specified in the ***RCV cross-domain key name** prompt (CRSDMNK parameter). This is a required parameter.

Top

Examples

None

Top

Error messages

Unknown

Top

Generate Cipher Key (GENCPHK)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Generate Cipher Key (GENCPHK) command generates a data-encrypting key enciphered under the host master key and optionally under a sending and/or receiving cross-domain key.

Top

Parameters

Keyword	Description	Choices	Notes
CPHK	CL var for generated key	*Character value*	Required, Positional 1
SNDCRSDMNK	*SND cross-domain key name	*Name*, ***NONE**	Optional, Positional 2
SNDRTNVAR	CL var for generated key	*Character value*	Optional, Positional 3
RCVCRSDMNK	*RCV cross-domain key name	*Name*, ***NONE**	Optional, Positional 4
RCVRTNVAR	CL var for generated key	*Character value*	Optional, Positional 5

Top

CL var for generated key (CPHK)

Specifies an 8-byte variable to receive the value of the generated data-encrypting key encrypted under the host master key. This is a required parameter.

Top

*SND cross-domain key name (SNDCRSDMNK)

Specifies the name of a sending cross-domain key that is to be used to encrypt the generated data-encrypting key.

The possible values are:

***NONE**
No sending cross-domain key is to be used to encrypt the data-encrypting key.

sending-cross-domain-key-name
Specify the name, or a variable containing the name, of a sending cross-domain key.

Top

CL var for generated key (SNDRTNVAR)

Specifies an 8-byte variable to receive the value of the generated data-encrypting key encrypted under the sending cross-domain key specified on the ***SND cross-domain key name** prompt (SNDCRSDMNK parameter).

Top

*RCV cross-domain key name (RCVCRSDMNK)

Specifies the name of a receiving cross-domain key that is to be used to encrypt the generated data-encrypting key.

The possible values are:

***NONE**
No receiving cross-domain key is to be used to encrypt the data-encrypting key.

receiving-cross-domain-key-name
Specify the name, or a variable containing the name, of a receiving cross-domain key.

Top

CL var for generated key (RCVRTNVAR)

Specifies an 8-byte variable to receive the value of the generated data-encrypting key encrypted under the receiving cross-domain key specified on the ***RCV cross-domain key name** prompt (RCVCRSDMNK parameter).

Top

Examples

None

Top

Error messages

Unknown

Top

Generate Cross-domain Keys (GENCRSDMNK)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Generate Cross-Domain Key (GENCRSDMNK) command generates random key values, installs them in the cross-domain key table, and prints a listing of the keys generated. The spooled file containing this listing is sent to the output queue associated with the job from which the command is submitted.

Note: To avoid any security exposure, you should use the Generate Cross-Domain Key (GENCRSDMNK) command only in a secure environment. Print the listing immediately and store it in a safe place.

Top

Parameters

Keyword	Description	Choices	Notes
CRSDMNK	Cross-domain key name base	*Name*	Required, Positional 1
NBR	Number to generate	1-9999	Required, Positional 2
SEED	Generation seed	*Character value*	Required, Positional 3
KUSE	Key use	*SND, *RCV, *PIN	Required, Positional 4

Top

Cross-domain key name base (CRSDMNK)

Specifies a valid system name, or a variable containing a name, to be used when generating the key names.

All generated key names will be 10 characters long with numerics in positions 7 through 10. When you specify a key name, any characters in positions 7 through 10 must be numeric. If you specify a key name base less than 10 characters, the name is filled in with 0's up to the last character, which is made a 1. This becomes the name of the first key generated.

For each succeeding key value, the last 4 characters of the key name, which is always a 4-digit number, is increased by 1 to become the next key name. This command does not generate any key names past 9999. This is a required parameter.

Top

Number to generate (NBR)

Specifies the number, or a variable containing the number, of keys to be generated. The value specified plus the 4-digit number from the first key name cannot exceed 10,000.

Top

Generation seed (SEED)

Specify 16 hexadecimal characters, or a character variable containing 16 hexadecimal characters, representing the 8-byte value to be used to initialize the random number generation routine. This is a required parameter.

Top

Key use (KUSE)

Specifies the use of the keys to be generated. This is a required parameter.

The possible values are:

***SND** The generated keys are added to the cross-domain key table as sending cross-domain keys.

***RCV** The generated keys are added to the cross-domain key table as receiving cross-domain keys.

***PIN** The generated keys are added to the cross-domain key table as personal identification number keys.

Top

Examples

None

Top

Error messages

Unknown

Top

Generate MAC (GENMAC)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Generate Message Authentication Code (GENMAC) command encrypts a variable length of data using cipher block chaining and returns the last 8 bytes to be used as a message authentication code.

Top

Parameters

Keyword	Description	Choices	Notes
DATA	Input data	*Character value*	Required, Positional 1
DTALEN	Input data length	1-32759	Required, Positional 2
CPHK	Cipher key	*Character value*	Required, Positional 3
ICV	Initial chaining value	*Character value*	Required, Positional 4
RTNVAR	CL var for returned value	*Character value*	Required, Positional 5
PAD	Pad option	***NO**, *YES	Optional, Positional 6
PADCHAR	Pad character	*Character value*, **X'00'**	Optional, Positional 7

Top

Input data (DATA)

Specifies the data, or a variable containing the data, to be encrypted. The data must be at least as long as the length specified in the **Input data length** prompt (DTALEN parameter). This is a required parameter.

Top

Input data length (DTALEN)

Specifies the length, or a variable containing the length, of the data to be encrypted. The length must be less than 32,760. This is a required parameter.

Top

Cipher key (CPHK)

Specifies an 8-byte value, or a variable containing an 8-byte value, to be used as the key for the data encryption algorithm. This value must be the value of the key encrypted under the host master key. There are no restrictions on the value of this parameter. This is a required parameter.

Top

Initial chaining value (ICV)

Specifies an 8-byte value, or a variable containing an 8-byte value, to be used as the initial chaining value when performing cipher block chaining. There are no restrictions on the value of this parameter. This is a required parameter.

Top

CL var for returned value (RTNVAR)

Specifies a variable to receive the 8-byte message authentication code. This is a required parameter.

Top

Pad option (PAD)

Specifies whether padding is to be performed.

The possible values are:

***NO** Padding will not be performed.

***YES** Before encrypting, the data is padded out to the next 8-byte multiple using the pad character specified in the **Pad character** prompt (PADCHAR parameter). The last byte is then replaced with a count of the number of pad characters.

Top

Pad character (PADCHAR)

Specifies a 1-byte value, or a variable containing a 1-byte value, to be used as the pad character when *YES is specified for the **Pad option** prompt (PAD parameter). The default (1-byte pad character) is hex 00.

Top

Examples

None

Top

Error messages

Unknown

Generate PIN (GENPIN)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Generate Personal Identification Number (GENPIN) command generates a personal identification number that is algorithmically related to your validation data. The generated number contains 16 decimal digits of which all or part may be assigned to a customer or used as an intermediate identification number if you have a preselected personal identification number value.

Top

Parameters

Keyword	Description	Choices	Notes
PINVLDK	PIN validation key name	*Name*	Required, Positional 1
DECTBL	Decimalization table	*Character value*	Required, Positional 2
VLDDTA	PIN validation data	*Character value*	Required, Positional 3
PINRTNVAR	Return variable	*Character value*	Required, Positional 4

Top

PIN validation key name (PINVLDK)

Specifies the name, or a variable containing the name, of a PIN validation key which is to be used to encrypt the validation data for the personal identification number. This validation key must exist in the cross-domain key table and be defined with a key use of personal identification number (*PIN specified on the **Key use** prompt (KUSE parameter) on the Generate Cross-Domain Key (GENCRSDMNK) or the Add Cross-Domain Key (ADDCRSDMNK command). This is a required parameter.

Top

Decimalization table (DECTBL)

Specifies 16 numeric digits (0-9), or a character variable containing 16 numeric digits, to be used as the table for conversion to decimal when generating the personal identification number. This is a required parameter.

Top

PIN validation data (VLDDTA)

Specifies 16 hexadecimal characters, or a character variable containing 16 hexadecimal characters, representing the 8 bytes of validation data to be used for generating the personal identification number (for example, hex C1 represents the character A). This is a required parameter.

Top

Return variable (PINRTNVAR)

Specifies a variable to receive the 16-digit generated personal identification number. This is a required parameter.

Top

Examples

None

Top

Error messages

Unknown

Top

Remove Cross-Domain Key (RMVCRSDMNK)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Remove Cross-Domain Key (RMVCRSDMNK) command removes one or more specified keys from the cross-domain key table.

Top

Parameters

Keyword	Description	Choices	Notes
CRSDMNK	Cross-domain key name	*Generic name, name*, *ALL	Required, Positional 1
KUSE	Cross-domain key use	*SND, *RCV, *PIN, *ALL	Required, Positional 2

Top

Cross-domain key name (CRSDMNK)

Specifies the name or generic name of the keys, with a key use that is specified in the **Cross-domain key use** prompt (KUSE parameter), that are to be removed from the cross-domain key table. You can also use this parameter to specify that all keys defined with the use specified in the **Cross-domain key use** prompt (KUSE parameter) are to be removed.

The possible values are:

***ALL** All the keys with a key use specified in the **Cross-domain key use** prompt (KUSE parameter) are to be removed from the cross-domain key table.

generic*-key-name
All keys with the same generic name with a key use specified in the **Cross-domain key use** prompt (KUSE parameter) are to be removed from the cross-domain key table. To specify a generic name, add an asterisk after the characters that are common in all the key names to be removed (ABC*, for example). If an asterisk is not included with the name, the system assumes that the name is a complete key name.

key-name
The key with the specified name with a key use specified in the **Cross-domain key use** prompt (KUSE parameter) are to be removed from the cross-domain key table.

Top

Cross-domain key use (KUSE)

Specifies the key use of the key(s) specified in the **Cross-domain key name** prompt (CRSDMNK parameter) that are to be removed from the cross-domain key table. You can also use this parameter to specify that all key uses of the named key are to be removed.

The possible values are:

***ALL** The keys named in the **Cross-domain key name** prompt (CRSDMNK parameter) with any key use are to be removed.

***SND** The keys named in the **Cross-domain key name** prompt (CRSDMNK parameter) with a key use of sending are to be removed.

***RCV** The keys named in the **Cross-domain key name** prompt (CRSDMNK parameter) with a key use of receiving are to be removed.

***PIN** The keys named in the **Cross-domain key name** prompt (CRSDMNK parameter) with a key use of personal identification number are to be removed.

Top

Examples

None

Top

Error messages

Unknown

Top

Set Master Key (SETMSTK)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Set Master Key (SETMSTK) command installs a new host master key on your system.

If an old master key exists, the values in the cross-domain key table are reencrypted under the new host master key value. You supply the values for two key parts, which are exclusive-ORed together to produce the new host master key. The new host master key must have odd parity in every byte.

To run this command, the system must be in a restricted state. (Enter the End System (ENDSYS) command to obtain a restricted state.) After installing the master key, return your system to a normal state by restarting your subsystems.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

None

Top

Error messages

Unknown

Top

Translate PIN (TRNPIN)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Translate Personal Identification Number (TRNPIN) command reenciphers a personal identification number (PIN) from encipherment under an input PIN protection key to encipherment under an output PIN protection key.

Top

Parameters

Keyword	Description	Choices	Notes
INPINPTCK	Input protection key name	*Name*	Required, Positional 1
ENCPIN	Enciphered PIN	*Character value*	Required, Positional 2
OUTPINPTCK	Output protection key name	*Name*	Required, Positional 3
RTNVAR	Return variable	*Character value*	Required, Positional 4

Top

Input protection key name (INPINPTCK)

Specifies the name, or a variable containing the name, of the input PIN protection key which the identification number is enciphered under. This key must exist in the cross-domain key table and be defined with a key use of personal identification number (*PIN specified on the **Key use** prompt (KUSE parameter) on the Generate Cross-Domain Key (GENCRSDMNK) or Add Cross-Domain Key (ADDCRSDMNK) command). This is a required parameter.

Top

Enciphered PIN (ENCPIN)

Specifies 16 hexadecimal characters, or a character variable containing 16 hexadecimal characters, representing a personal identification number enciphered under the input PIN protection key. This is a required parameter.

Top

Output protection key name (OUTPINPTCK)

Specifies the name, or a variable containing the name, of the output PIN protection key. The value specified in the **Enciphered PIN** prompt (ENCPIN parameter) is decrypted using the input protection key and encrypted using the output protection key. This key must exist in the cross-domain key table with a defined key use of sending. This is a required parameter.

Top

Return variable (RTNVAR)

Specifies a variable to receive the 16-character value of the identification number reenciphered under the output PIN protection key. This is a required parameter.

Top

Examples

None

Top

Error messages

Unknown

Top

Verify Master Key (VFYMSTK)

Where allowed to run:

- Interactive job (*INTERACT)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Verify Master Key (VFYMSTK) command compares the verification code you type in on the prompt display with the verification code of the current host master key.

If the verification codes do not match, the master key has been altered since you obtained your verification code or you have mistyped it. If the verification codes match, a message is shown at the bottom of the display, indicating that the entered verification code is correct.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

None

Top

Error messages

Unknown

Top

Verify PIN (VFYPIN)

Where allowed to run:

- Batch program (*BPGM)
- Interactive program (*IPGM)

Threadsafe: No

Parameters
Examples
Error messages

The Verify Personal Identification Number (VFYPIN) command determines if the customer's personal identification number is algorithmically related to the customer's validation data. The result of the test is returned in a 1-byte variable.

Top

Parameters

Keyword	Description	Choices	Notes
INPINPTCK	Input protection key name	*Name*	Required, Positional 1
ENCPIN	Enciphered PIN	*Character value*	Required, Positional 2
PINVLDK	PIN validation key name	*Name*	Required, Positional 3
DECTBL	Decimalization table	*Character value*	Required, Positional 4
VLDDTA	PIN validation data	*Character value*	Required, Positional 5
PINCHKLEN	PIN check length	1-16	Required, Positional 6
RTNVAR	Return variable	*Character value*	Required, Positional 7
PINPADCHAR	PIN pad character	*Character value*, ***NONE**	Optional, Positional 8
OFFSET	PIN offset data	*Character value*, ***NONE**	Optional, Positional 9

Top

Input protection key name (INPINPTCK)

Specifies the name, or a variable containing the name, of the input PIN protection key that the personal identification number is enciphered under. This key must exist in the cross-domain key table and be defined with a key use of personal identification number (*PIN specified on the **Key use** prompt (KUSE parameter) on the Generate Cross-Domain Key (GENCRSDMNK) or Add Cross-Domain Key (ADDCRSDMNK) command). This is a required parameter.

Top

Enciphered PIN (ENCPIN)

Specifies 16 hexadecimal characters, or a character variable containing 16 hexadecimal characters, representing a personal identification number in 3624 format enciphered under the input PIN protection key. This is a required parameter.

Top

PIN validation key name (PINVLDK)

Specifies the name, or a variable containing the name, of a validation key that is to be used to encrypt the validation data. This key must exist in the cross-domain key table and be defined with a key use of personal identification number (*PIN specified on the **Key use** prompt (KUSE parameter) on the Generate Cross-Domain Key (GENCRSDMNK) or Add Cross-Domain Key (ADDCRSDMNK) command). This is a required parameter.

Top

Decimalization table (DECTBL)

Specifies 16 numeric digits (0-9), or a character variable containing 16 numeric digits, to be used as the table for conversion to decimal when verifying the personal identification number. This is a required parameter.

Top

PIN validation data (VLDDTA)

Specifies 16 hexadecimal characters, or a character variable containing 16 hexadecimal characters, representing the 8 bytes of validation data to be used for verifying the personal identification number. This is a required parameter.

Top

PIN check length (PINCHKLEN)

Specifies the number, or a variable containing the number, of digits of the personal identification number to be verified.

Top

Return variable (RTNVAR)

Specifies a variable to receive the 1-byte verification status. If the personal identification number is valid, the status is set to 0. If the personal identification number is not valid, the status is set to 1.

Top

PIN pad character (PINPADCHAR)

Specifies the hexadecimal character that was used to pad the personal identification number before being encrypted. It is removed from the end of the personal identification number before verification.

The possible values are:

***NONE**
The personal identification number was not padded before being encrypted.

PIN-pad-character
Specify 1 hexadecimal character, or a variable containing 1 hexadecimal character, representing the value that was used to pad the personal identification number before it was encrypted

Top

PIN offset data (OFFSET)

Specifies a numeric value to be added to the intermediate personal identification number obtained from the validation data before comparing it with the supplied personal identification number.

The possible values are:

***NONE**
No offset value should be added to the intermediate personal identification number before the comparison. Specifying *NONE for this parameter is equivalent to entering a value of zero for the parameter.

offset-data
Specify 16 numeric digits (0-9), or a character variable containing 16 numeric digits, representing the value to be added to the intermediate personal identification number before the comparison.

Top

Examples

None

Top

Error messages

Unknown

Top

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY8809
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2
IBM
Infoprint
InfoWindow
iSeries
LPDA
OfficeVision

OS/400
Print Services Facility
RPG/400
SystemView
System/36
TCS
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

Personal Use: You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

Code disclaimer information

This document contains programming examples.

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you ″AS IS″ without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

Printed in USA