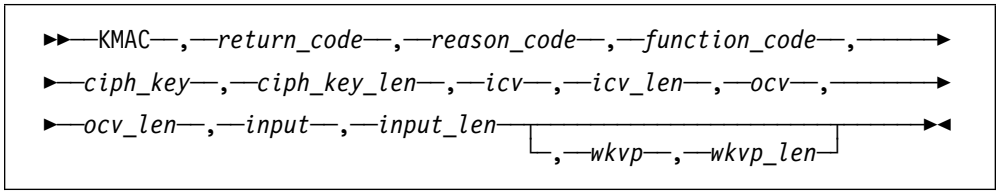
KMAC - Compute message authentication code

Purpose

Compute the message authentication code of a record using one of three encryption algorithms: the Data Encryption Standard (DES), the triple DES standard (TDES) or the Advanced Encryption Standard (AES), using the KMAC ('B91E') hardware instruction.

Format



Parameters

KMAC
(input, CHAR, 8) can be passed as a literal or in a variable.

return_code
(output, INT, 4) is a variable for the return code from KMAC.

reason_code
(output, INT, 4) is a variable for the reason code from KMAC.

function_code
(input, INT, 4) specifies which encryption algorithm to use. The list of valid codes is given in the Usage Notes.

ciph_key
(input, CHAR, *ciph_key_len*) is the cryptographic key to use. If the *wkvp* is specified, this is the encrypted cryptographic key value, as returned by the PCKMO Rexx exec.

ciph_key_len
(input, INT, 4) is a variable containing the length of the preceding character parameter, *ciph_key*

icv
(input, CHAR, *icv_len*) Initial chaining value

icv_len
(input, INT, 4) is a variable containing the length of the preceding character parameter, *icv*

ocv
(output, CHAR, *ocv_len*) Output chaining value

ocv_len
(input, INT, 4) is a variable containing the length of the preceding character parameter, *ocv*

input
(input, CHAR, *input_len*) is the record whose message authentication code (MAC) value is to be computed.

input_len
(input, INT, 4) is a variable containing the length of the preceding character parameter, *input*

wkvp
(input, CHAR, *wkvp_len*) the 24 or 32 byte wrapping key verification pattern as returned by the PCKMO instruction

wkvp_len
(input, INT, 4) is a variable containing the length of the wrapping key verification pattern as returned by the PCKMO instruction. This value must be either 24 or 32.

Usage

1. Valid function codes are:

Code	Function	Description
1	DES	use the Data Encryption Standard (DES) algorithm with a 8 byte key and a data block size of 8 bytes.
2	TDES_128	use the triple DES algorithm with a 16 byte key and a data block size of 8 bytes.
3	TDES_192	use the triple DES algorithm with a 24 byte key and a data block size of 8 bytes.
9	Encrypted_DES	use the Data Encryption Standard (DES) algorithm with an encrypted 8 byte key, along with the 24 bytes DES wrappingkey verification pattern value, and a data block size of 8 bytes.
10	Encrypted_TDES_128	use the triple DES algorithm with an encrypted 16 byte key along with the 24 bytes DES wrappingkey verification pattern value, and a data block size of 8 bytes.
11	Encrypted_TDES_192	use the triple DES algorithm with an encrypted 24 byte key along with the 24 bytes DES wrappingkey verification pattern value, and a data block size of 8 bytes.
18	AES_128	use the Advanced Encryption Standard algorithm with a 16 byte key and a data block size of 16 bytes.
19	AES_192	use the Advanced Encryption Standard algorithm with a 24 byte key and a data block size of 16 bytes.
20	AES_256	use the Advanced Encryption Standard algorithm with a 32 byte key and a data block size of 16 bytes.
26	Encrypted_AES_128	use the Advanced Encryption Standard algorithm with an encrypted 16 byte key, along with the 32 bytes AES wrapping key verification pattern value, and a data block size of 16 bytes.
27	Encrypted_AES_192	use the Advanced Encryption Standard algorithm with an encrypted 24 byte key, along with the 32 bytes AES wrapping key verification pattern value, and a data block size of 16 bytes.
28	Encrypted_AES_256	use the Advanced Encryption Standard algorithm with an encrypted 32 byte key, along with the 32 bytes AES wrapping key verification pattern value, and a data block size of 16 bytes.

All other function code values are unassigned.

2. Definitions of these functions for Rexx are in KMACREXX COPY and for PL/I are in KMACPLI COPY.

3. The length of the input record, *input* must be an integer multiple of the data block size..
4. The *icv* and *ocv* variables can be the same; that is, the output chaining value can overwrite the initial chaining value.
5. Before processing the first part of a message, the program must set the initial values for the initial chaining value field, *icv*. To comply with ANSI X9.9, the initial chaining value must be set to all binary zeros.
6. The current *wkvp* wrapping key verification pattern can be found by running the PCKMO Rexx exec.
7. Since both the wrapping key and wrapping key verification pattern registers are changed each time CMS, or zCMS, is IPL-ed, or the CP SYSTEM RESET command is issued in the virtual machine, using encrypted keys in this environment is problematic. Support for *wkvp* is included only for completeness.

Return codes and Reason codes

Return codes:

Severity	Code	Meaning
OK	0	The operation was successful.
WARNING	4	The operation was successful, but a warning condition was encountered.
ERROR	8	The operation was unsuccessful.
SEVERE	12	The operation was unsuccessful, and the operation can no longer continue.

Reason codes:

Severity	Code	Description
ERROR	8	bad key length, key not 8, 16, 24, or 32 bytes long
ERROR	9	bad chaining value length, cv is not 8 or 16 bytes long
ERROR	10	unknown function code specified
ERROR	11	invalid direction specified, must be either "E" or "D"
ERROR	12	bad cipher feedback (LCFB) length
ERROR	13	bad input record length, not an integer multiple of data block size
ERROR	14	bad output record length, not equal to input record length
ERROR	15	bad wrapping key verification pattern length; this value must be either 24 or 32
ERROR	16	unsupported function code value
ERROR	17	Verification pattern mismatch; the value specified does not match the contents of the wrapping key verification pattern register