z/OS

Cryptographic Services Integrated Cryptographic Service Facility CCA Service Algorithm Updates APAR OA47781

Contents

Chapter 1. Overview

This document describes changes to the Integrated Cryptographic Service Facility (ICSF) product in support of the following CCA service algorithm updates:

- Support for the RSAES-OAEP format for the PKA Decrypt and PKA Encrypt callable services.
- Support in the Key Generate callable service for the CIPHER, DATAC, and DATAM key types in the OP, IM, or EX key forms.
- Operational Key Load support for HMAC keys loaded from the TKE workstation.
- Master key verification patterns on the ICSF Hardware Status panel.
- Access control point offsets on the Domain Role panel.

These changes are available through the application of the PTF for APAR OA47781 and apply to FMID HCR77B0, HCR77A1, and HCR77A0.

This document contains alterations to information previously presented in the following books:

- *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-03
- *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-03
- *z/OS Cryptographic Services ICSF System Programmer's Guide*, SC14-7507-03

The technical changes made to the ICSF product by the application of the PTF for APAR OA47781 are indicated in this document by a vertical line to the left of the change.

Chapter 2. Update of z/OS Cryptographic Services ICSF System Programmer's Guide, SC14-7507-03, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF System Programmer's Guide*, SC14-7507-03, for the CCA service algorithm updates provided by this APAR. Refer to this source document if background information is needed.

Parameters in the installation options data set

The installation options data set is an intended programming interface.

When specifying parameter values within parentheses, leading and trailing blanks are ignored. Embedded blanks may cause unpredictable results.

MASTERKCVLEN(2 or 3 or 4 or 5 or 6 or ALL)
Defines the number of hexadecimal digits to display on the ICSF Coprocessor Hardware Status panel (CSFCMP40) for the verification and hash patterns for the master keys. The patterns are also referred to as key check values. When an integer value is specified, that number of digits will be displayed. When ALL is specified, all the digits will be displayed.

The default is ALL.

This option can be used for compliance with the ISO11568 standard for the display of the key check values for master keys.

Callable services

For complete reference information on these callable services, refer to *z/OS Cryptographic Services ICSF Application Programmer's Guide*.

Table 1. Summary of new and changed ICSF callable services

Callable service	Release	Description
Key Generate	HCR77A0	**Changed:** Generate DES DATAC, DATAM, and CIPHER keys as a single key in key forms OP, IM, and EX.
PKA Decrypt	HCR77A0	**Changed:** Support formatting data as RSA-OAEP block and both SHA-1 and SHA-256 hashing.
PKA Encrypt	HCR77A0	**Changed:** Support formatting data as RSA-OAEP block and both SHA-1 and SHA-256 hashing.

Chapter 3. Update of z/OS Cryptographic Services ICSF Application Programmer's Guide, SC14-7508-03, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-03, for the CCA service algorithm updates provided by this APAR. Refer to this source document if background information is needed.

Key Generate (CSNBKGN and CSNEKGN)

Use the key generate callable service to generate either one or two odd parity DES keys of *any* type. The keys can be single-length (8 bytes), double-length (16 bytes), or, in the case of DATA keys, triple-length (24 bytes). The callable service does not produce keys in clear form and all keys are returned in encrypted form. When two keys are generated, each key has the same clear value, although this clear value is not exposed outside the secure cryptographic feature.

Use the key generate callable service to generate an AES key of DATA type. The callable service does not produce AES keys in clear form and all AES keys are returned in encrypted form. Only one AES key is generated.

The callable service name for AMODE (64) invocation is CSNEKGN.

Format

```
CALL CSNBKGN(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             key_form,
             key_length,
             key_type_1,
             key_type_2,
             KEK_key_identifier_1,
             KEK_key_identifier_2,
             generated_key_identifier_1,
             generated_key_identifier_2 )
```

Parameters

return_code

Direction	Type
Output	Integer

The return code specifies the general result of the callable service.

reason_code

Direction	Type
Output	Integer

The reason code specifies the result of the callable service that is returned to the application program. Each return code has different reason codes that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

The length of the data that is passed to the installation exit. The data is identified in the *exit_data* parameter.

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

key_form

Direction	Type
Input	Character String

A 4-byte keyword that defines the type of key or keys you want to generate. This parameter also specifies if each key should be returned for either operational, importable, or exportable use. The keyword must be in a 4-byte field, left-justified, and padded with blanks.

The first two characters refer to *key_type_1*. The next two characters refer to *key_type_2*.

These keywords are allowed: OP, IM, EX, OPIM, OPEX, IMEX, EXEX, OPOP, and IMIM. See Table 2 for their meanings.

If the *key_form* is OP, EX or IM, the KEK_*key_identifier_2*, *key_type_2*, and *generated_key_identifier_2* should be set to NULL.

Table 2. Key Form values for the Key Generate callable service

Keyword	Meaning
EX	One key that can be sent to another system.
EXEX	A key pair; both keys to be sent elsewhere, possibly for exporting to two different systems. The key pair has the same clear value.
IM	One key that can be locally imported. The key can be imported onto this system to make it operational at another time.
IMEX	A key pair to be imported; one key to be imported locally and one key to be sent elsewhere. Both keys have the same clear value.
IMIM	A key pair to be imported; both keys to be imported locally at another time.
OP	One operational key. The key is returned to the caller in the key token format. Specify the OP key form when generating AES keys.
OPEX	A key pair; one key that is operational and one key to be sent from this system. Both keys have the same clear value.

Table 2. Key Form values for the Key Generate callable service (continued)

Keyword	Meaning
OPIM	A key pair; one key that is operational and one key to be imported to the local system. Both keys have the same clear value. On the other system, the external key token can be imported to make it operational.
OPOP	A key pair; normally with different control vector values.

The key forms are defined as follows:

Operational (OP)
The key value is enciphered under a master key. The result is placed into an internal key token. The key is then operational at the local system.

Importable (IM)
The key value is enciphered under an importer key-encrypting key. The result is placed into an external key token.

Exportable (EX)
The key value is enciphered under an exporter key-encrypting key. The result is placed into an external key token. The key can then be transported or exported to another system and imported there for use. This key form cannot be used by any ICSF callable service.

The keys are placed into tokens that the *generated_key_identifier_1* and *generated_key_identifier_2* parameters identify.

Valid key type combinations depend on the key form. See Table 7 on page 13 for valid key combinations.

key_length

Direction	Type
Input	Character String

An 8-byte value that defines the length of the key. The keyword must be left-justified and padded on the right with blanks. You must supply one of the key length values in the *key_length* parameter.

Table 3. Key Length values for the Key Generate callable service

Value	Description	Algorithm
SINGLE or KEYLN8	The key should be a single length (8-byte) key.	DES
SINGLE-R	The key should be a double length (16-byte) key. The two key halves will be the same. This makes the key effectively a single length key.	DES
DOUBLE or KEYLN16	The key should be a double length (16-byte or 128-bit) key	AES or DES
DOUBLE-O	The key should be a double length (16-byte) key. Each of the two key halves will be unique (not the same value).	DES

Table 3. Key Length values for the Key Generate callable service (continued)

Value	Description	Algorithm
KEYLN24	The key should be a 24-byte (192-bit) key.	AES or DES
KEYLN32	The key should be a 32-byte (256-bit) key.	AES

DES Keys: Double-length (16-byte) keys have an 8-byte left half and an 8-byte right half. Both halves can have identical clear values or not. If you want the same value to be used in both key halves (refered to as replicated key values), specify *key_length* as SINGLE, SINGLE-R or KEYLN8. If you want different values to be the basis of each key half, specify *key_length* as DOUBLE, DOUBLE-O or KEYLN16.

Triple-length (24-byte) keys have three 8-byte key parts. This key length is valid for DATA keys only. To generate a triple-length DATA key with three different values to be the basis of each key part, specify *key_length* as KEYLN24.

Use SINGLE/SINGLE-R if you want to create a DES transport key that you would use to exchange DATA keys with a PCF system.

AES Keys: AES only allows KEYLN16, KEYLN24, KEYLN32. To generate a 128-bit AES key, specify key_length as KEYLN16. For 192-bit AES keys specify key_length as KEYLN24. A 256-bit AES key requires a key_length of KEYLN32. All AES keys are DATA keys.

This table shows the valid key lengths for each key type supported by DES keys. An **X** indicates that a key length is permitted for a key type. A **Y** indicates that the key generated will be a double-length key with replicated key values. It is preferred that SINGLE-R be used for this result.

Table 4. Key lengths for DES keys

Key Type	Single - KEYLN8	Single-R	Double - KEYLN16	DOUBLE-O	KEYLN24
MAC MACVER	X X	X X	X X	X X	
DATA	X		X		X
DATAC* DATAM DATAMV		X X X	X X X	X X X	
EXPORTER IMPORTER	Y Y	X X	X X	X X	
IKEYXLAT OKEYXLAT	Y Y	X X	X X	X X	
CIPHER DECIPHER ENCIPHER	X X X	X X X	X X X	X X X	

Table 4. Key lengths for DES keys (continued)

Key Type	Single - KEYLN8	Single-R	Double - KEYLN16	DOUBLE-O	KEYLN24
IPINENC	Y	X	X	X	
OPINENC	Y	X	X	X	
PINGEN	Y	X	X	X	
PINVER	Y	X	X	X	
CVARDEC*	X	X	X		
CVARENC*	X	X	X		
CVARPINE*	X	X	X		
CVARXCVL*	X	X	X		
CVARXCVR*	X	X	X		
DKYGENKY*		X	X	X	
KEYGENKY*		X	X	X	
CIPHERXI			X	X	
CIPHERXL			X	X	
CIPHERXO			X	X	

This table shows the valid key lengths for each key type supported by AES keys. An X indicates that a key length is permitted for that key type.

Table 5. Key lengths for AES keys

Key Type	128-byte	192-byte	256-byte
AESTOKEN	X	X	X
AESDATA	X	X	X

key_type_1

Direction	Type
Input	Character String

Use the *key_type_1* parameter for the first, or only key, that you want generated. The keyword must be left-justified and padded with blanks. Valid type combinations depend on the key form.

The 8-byte keyword for the *key_type_1* parameter can be one of the following:

- AESDATA, AESTOKEN, CIPHER, CIPHERXI, CIPHERXL, CIPHERXO, DATA, DATAC, DATAM, DATAMV, DECIPHER, ENCIPHER, EXPORTER, IKEYXLAT, IMPORTER, IPINENC, MAC, MACVER, OKEYXLAT, OPINENC, PINGEN and PINVER
- or the keyword TOKEN

If *key_type_1* is TOKEN, ICSF examines the control vector (CV) field in the *generated_key_identifier_1* parameter to derive the key type. When *key_type_1* is TOKEN, ICSF does not check for the length of the key for DATA keys. Instead, ICSF uses the *key_length* parameter to determine the length of the key.

If *key_type_1* is AESDATA or AESTOKEN, the key generated will be an AES key of type DATA. When *key_type_1* is AESTOKEN, ICSF uses the key_length parameter to determine the length of the key.

See Table 6 on page 12 and Table 7 on page 13 for valid key type and key form combinations.

key_type_2

Direction	Type
Input	Character String

Use the *key_type_2* parameter for a key pair, which is shown in Table 7 on page 13. The keyword must be left-justified and padded with blanks. Valid type combinations depend on the key form. *key_type_2* is only used when DES keys are generated.

The 8-byte keyword for the key_type_2 parameter can be one of the following:

- CIPHER, CIPHERXI, CIPHERXL, CIPHERXO, DATA, DATAC, DATAM, DATAMV, DECIPHER, ENCIPHER, EXPORTER, IKEYXLAT, IMPORTER, IPINENC, MAC, MACVER, OKEYXLAT, OPINENC, PINGEN and PINVER
- or the keyword TOKEN

If *key_type_2* is TOKEN, ICSF examines the control vector (CV) field in the *generated_key_identifier_2* parameter to derive the key type. When *key_type_2* is TOKEN, ICSF does not check for the length of the key for DATA keys. Instead, ICSF uses the *key_length* parameter to determine the length of the key.

If only one key is to be generated, *key_type_2* and *KEK_key_identifier_2* are ignored.

See Table 6 on page 12 and Table 7 on page 13 for valid key type and key form combinations.

KEK_key_identifier_1

Direction	Type
Input/Output	String

A 64-byte string of a DES internal key token containing the importer or exporter key-encrypting key, or a key label. If you supply a key label that is less than 64-bytes, it must be left-justified and padded with blanks. *KEK_key_identifier_1* is required for a *key_form* of IM, EX, IMEX, EXEX, or IMIM.

When key_form OP is used, parameters *KEK_key_identifier_1* and *KEK_key_identifier_2* are ignored. In this case, it is recommended that the parameters are initialized to 64-bytes of X'00'.

If the NOCV bit is on in the internal key token containing the key-encrypting key, the key-encrypting key itself (not the key-encrypting key variant) is used to encipher the generated key. For example, the key has been installed in the cryptographic key data set through the key generator utility program or the key entry hardware using the NOCV parameter; or you are passing the key-encrypting key in the internal key token with the NOCV bit on and your program is running in supervisor state or key 0-7.

KEK_key_identifier_1 cannot be an AES key token or label.

KEK_key_identifier_2

Direction	Type
Input/Output	String

A 64-byte string of a DES internal key token containing the importer or exporter key-encrypting key, or a key label of an internal token. If you supply a key label that is less than 64-bytes, it must be left-justified and padded with blanks. *KEK_key_identifier_2* is required for a *key_form* of OPIM, OPEX, IMEX, IMIM, or EXEX. This field is ignored for *key_form* keywords OP, IM and EX. When key_form OP is used, parameter *KEK_key_identifier_2* is ignored. In this case, it is recommended that the parameter is initialized to 64-bytes of X'00'.

If the NOCV bit is on in the internal key token containing the key-encrypting key, the key-encrypting key itself (not the key-encrypting key variant) is used to encipher the generated key. For example, the key has been installed in the cryptographic key data set through the key generator utility program or the key entry hardware using the NOCV parameter; or you are passing the key-encrypting key in the internal key token with the NOCV bit on and your program is running in supervisor state or in key 0-7.

KEK_key_identifier_2 cannot be an AES key token or label.

generated_key_identifier_1

Direction	Type
Input/Output	String

This parameter specifies either a generated:

- Internal DES or AES key token for an operational key form, or
- External DES key tokens containing a key enciphered under the *KEK_key_identifier_1* parameter.

If you specify a *key_type_1* of TOKEN, then this field contains a valid DES token of the key type you want to generate. Otherwise, on input, this parameter must be binary zeros. See *key_type_1* for a list of valid key types.

If you specify a *key_type_1* of IMPORTER or EXPORTER and a *key_form* of OPEX, and if the *generated_key_identifier_1* parameter contains a valid DES internal token of the SAME type, the NOCV bit, if on, is propagated to the generated key token.

When *key_type_1* parameter is AESDATA, then *generated_key_identifier_1* is ignored. In this case, it is recommended that the parameter be initialized to 64-bytes of X'00'. If you specify a key_type_1 of AESTOKEN, the *generated_key_identifier_1* parameter must be an internal AES key token or a clear AES key token. Information in this token can be used to determine the key type:

- The *key_type_1* parameter overrides the type in the token.
- The *key_length* parameter overrides the length value in the generated key token.

ICSF supports two methods of wrapping the key value in a symmetric key token: the original ECB wrapping and an enhanced CBC wrapping method which is ANSI X9.24 compliant. The output *generated_key_identifier_1* will use the default wrapping method unless a skeleton token is supplied as input. If a skeleton token is supplied as input, the wrapping method in the skeleton token will be used.

generated_key_identifier_2

Direction	Type
Input/Output	String

This parameter specifies either a generated:

- internal DES key token or
- external DES key token enciphered under *KEK_key_identifier_2*.

ICSF supports two methods of wrapping the key value in a symmetric key token: the original ECB wrapping and an enhanced CBC wrapping method which is ANSI X9.24 compliant. The output *generated_key_identifier_2* will use the default wrapping method unless a skeleton token is supplied as input. If a skeleton token is supplied as input, the wrapping method in the skeleton token will be used.

Restrictions

This callable service does not support version X'10' external DES key tokens (RKX key tokens).

Usage notes

SAF may be invoked to verify the caller is authorized to use this callable service, the key label, or internal secure key tokens that are stored in the CKDS or PKDS.

For key types CIPHERXI, CIPHERXL, and CIPHERXO, the key-encrypting keys in the KEK_key_identifier_1 and KEK_key_identifier_2 parameters must have a control vector with the key halves guaranteed unique flag on in the key form bits. An existing key-encrypting key can have its control vector updated using the restrict key attribute callable service.

Usage notes - Key type and key form combinations

Table 6 shows the valid key type and key form combinations for a single DES or AES key. Key types marked with an "*" must be requested through the specification of a proper control vector in a key token and through the use of the TOKEN keyword.

Note: Not all keytypes are valid on all hardware.

Table 6. Key Generate Valid Key Types and Key Forms for a Single Key

Key Type 1	Key Type 2	OP	IM	EX
AESDATA	Not applicable	X		
AESTOKEN	Not applicable	X		
CIPHER	Not applicable	X	X	X
DATA	Not applicable	X	X	X
DATAC*	Not applicable	X	X	X
DATAM	Not applicable	X	X	X
DKYGENKY*	Not applicable	X	X	X
KEYGENKY*	Not applicable	X	X	X
MAC	Not applicable	X	X	X
PINGEN	Not applicable	X	X	X

Table 7 shows the valid key type and key form combinations for a DES key pair. Key types marked with an "*" must be requested through the specification of a proper control vector in a key token and through the use of the TOKEN keyword.

See Table 8 on page 14 for an explanation of the differences between **E** as compared to **X**.

Table 7. Key Generate Valid Key Types and Key Forms for a Key Pair

Key Type 1	Key Type 2	OPEX	EXEX	OPIM, OPOP, IMIM	IMEX
CIPHER	CIPHER CIPHERXI CIPHERXL CIPHERXO DECIPHER ENCIPHER	X	X	X	X
CIPHERXI	CIPHER ENCIPHER	E	X	X	E
CIPHERXI	CIPHERXO	E	X		E
CIPHERXL	CIPHER	E	X	X	E
CIPHERXL	CIPHERXL	E	X		E
CIPHERXO	CIPHER DECIPHER	E	X	X	E
CIPHERXO	CIPHERXI	E	X		E
CVARDEC*	CVARENC* CVARPINE*	E			E
CVARENC*	CVARDEC* CVARXCVL* CVARXCVR*	E			E
CVARXCVL*	CVARENC*	E			E
CVARXCVR*	CVARENC*	E			E
CVARPINE*	CVARDEC*	E			E
DATA	DATA	X	X	X	X
DATAC*	DATAC*	X	X	X	X
DATAM	DATAM DATAMV	X	X	X	X
DECIPHER	CIPHER CIPHERXO ENCIPHER	X	X	X	X
DKYGENKY*	DKYGENKY*	X	X	X	X

Table 7. Key Generate Valid Key Types and Key Forms for a Key Pair (continued)

Key Type 1	Key Type 2	OPEX	EXEX	OPIM, OPOP, IMIM	IMEX
ENCIPHER	CIPHER CIPHERXI DECIPHER	X	X	X	X
EXPORTER	IKEYXLAT IMPORTER	X	X		X
IKEYXLAT	EXPORTER OKEYXLAT	X	X		X
IMPORTER	EXPORTER OKEYXLAT	X	X		X
IPINENC	OPINENC	X	X	E	X
KEYGENKY*	KEYGENKY*	X	X	X	X
MAC	MAC MACVER	X	X	X	X
OKEYXLAT	IKEYXLAT IMPORTER	X	X		X
OPINENC	IPINENC	X	X	E	X
OPINENC	OPINENC			X	
PINVER	PINGEN	X	X		X
PINGEN	PINVER	X	X		X

If you need to use NOCV key-encrypting keys, you need to enable NOCV IMPORTER and NOCV EXPORTER access control points

Access control points

The following table shows the access control points in the domain role that control the function of this service.

Table 8. Required access control points for Key Generate

Usage	Access Control Point
The key-form and key-type combinations shown with an 'X' in the Key_Form OP column in Table 6 on page 12.	Key Generate – OP
The key-form and key-type combinations shown with an 'X' in the Key_Form IM column in Table 6 on page 12.	Key Generate – Key set
The key-form and key-type combinations shown with an 'X 'in the Key_Form EX column in Table 6 on page 12.	Key Generate - Key set
The key-form and key-type combinations shown with an 'X' in Table 7 on page 13	Key Generate - Key set

Table 8. Required access control points for Key Generate (continued)

Usage	Access Control Point
The key-form and key-type combinations shown with an 'E' in Table 7 on page 13	Key Generate - Key set extended
The SINGLE-R key-length keyword is specified	Key Generate - SINGLE-R

To use a NOCV IMPORTER key-encrypting key with the key generate service, the **NOCV KEK usage for import-related functions** access control point must be enabled in addition to one or both of the access control points listed.

To use a NOCV EXPORTER key-encrypting key with the key generate service, the **NOCV KEK usage for export-related functions** access control point must be enabled in addition to one or both of the access control points listed.

To use the SINGLE-R rule array keyword, the **Key Generate – SINGLE-R** access control point must be enable.

If a key-encrypting key identifier is a weaker key than the key being generated, then:

- the service will fail if the **Prohibit weak wrapping - Transport keys** access control point is enabled.
- the service will complete successfully with a warning return code if the **Warn when weak wrap - Transport keys** access control point is enabled.

When the **Disallow 24-byte DATA wrapped with 16-byte Key** access control point is enabled, this service will fail if the source key is a triple-length DATA key and the DES master key is a 16-byte key or the key-encrypting key is a double-length key.

Required hardware

Table 9 lists the required cryptographic hardware for each server type and describes restrictions for this callable service.

Table 9. Key generate required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor Crypto Express2 Coprocessor	Key types CIPHERXI, CIPHERXL and CIPHERXO are not supported. Key length DOUBLE-O is not supported Secure AES keys are not supported. CIPHER keys in key form OP, EX, and IM are not supported.
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	Key types CIPHERXI, CIPHERXL and CIPHERXO are not supported. Key length DOUBLE-O is not supported Secure AES key support requires the Nov. 2008 or later licensed internal code (LIC). CIPHER keys in key form OP, EX, and IM are not supported.

Table 9. Key generate required hardware (continued)

Server	Required cryptographic hardware	Restrictions
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	Key types CIPHERXI, CIPHERXL and CIPHERXO are not supported. Key length DOUBLE-O is not supported Secure AES key support requires the Nov. 2008 or later licensed internal code (LIC). CIPHER keys in key form OP, EX, and IM are not supported.
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	Key types CIPHERXI, CIPHERXL and CIPHERXO are not supported. Key length DOUBLE-O is not supported CIPHER keys in key form OP, EX, and IM are not supported.
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	Generation of CIPHER keys in key form OP, EX, and IM requires the June 2015 or later licensed internal code (LIC).
IBM z13	Crypto Express5 CCA Coprocessor	Generation of CIPHER keys in key form OP, EX, and IM requires the July 2015 or later licensed internal code (LIC).

PKA Decrypt (CSNDPKD and CSNFPKD)

Use this service to decrypt (unwrap) a formatted key value. The service unwraps the key, parses it, and returns the parsed value to the application in the clear. PKCS 1.2, RSAES-OAEP, and ZERO-PAD formatting is supported. For PKCS 1.2, the decrypted data is examined to ensure it meets RSA DSI PKCS #1 block type 2 format specifications.

For PKA private keys, this service allows the use of clear or encrypted RSA private keys. If an external clear key token is used, the master keys are not required to be installed in any cryptographic coprocessor and PKA callable services does not have to be enabled. Requests are routed to a Cryptographic Accelerator if available when a clear key token is used. ZERO-PAD is only supported for external RSA clear private keys.

This service also supports the use of secure PKCS #11 private keys, which requires an active Enterprise PKCS #11 coprocessor. PKCS 1.2 formatting is supported.

The callable service name for AMODE(64) invocation is CSNFPKD.

Format

```
CALL CSNDPKD(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             rule_array_count,
             rule_array,
```

```
PKA_enciphered_keyvalue_length,
PKA_enciphered_keyvalue,
data_structure_length,
data_structure,
key_identifier_length,
key_identifier,
target_keyvalue_length,
target_keyvalue)
```

Parameters

return_code

Direction	Type
Output	Integer

The return code specifies the general result of the callable service.

reason_code

Direction	Type
Output	Integer

The reason code specifies the result of the callable service that is returned to the application program. Each return code has different reason codes that are assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

The length of the data that is passed to the installation exit. The data is identified in the *exit_data* parameter.

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. This value must be 1.

rule_array

Direction	Type
Input	String

The keyword that provides control information to the callable service. The keyword is left-justified in an 8-byte field and padded on the right with blanks.

Table 10. Keywords for PKA Decrypt

Keyword	Meaning
Recovery Method (required) specifies the method to use to recover the key value.	
PKCS-1.2	RSA PKCS #1 V1.5 block type 02 will be used to recover the key value.
PKCSOAEP	Specifies to recover the data formatted using the RSAES-OAEP encoding scheme defined in the RSA PKCS #1 v2.0 standard. This keyword is not valid when using a secure PKCS #11 private key.
ZERO-PAD	The input *PKA_enciphered_keyvalue* is decrypted using the RSA private key. The entire result (including leading zeros) will be returned in the *target_keyvalue* field. For PKA keys, the key_identifier must be an external RSA token or the labelname of a external token. This keyword is not valid when using a secure PKCS #11 private key.
Hash Method (one required when PKCSOAEP is specified. Otherwise, not allowed.)	
SHA-1	Specifies to use the SHA-1 method to calculate the OAEP message digest.
SHA-256	Specifies to use the SHA-256 method to calculate the OAEP message digest.

PKA_enciphered_keyvalue_length

Direction	Type
Input	Integer

The length of the *PKA_enciphered_keyvalue* parameter in bytes. The maximum size that you can specify is 512 bytes. The length should be the same as the modulus length of the *key_identifier*.

PKA_enciphered_keyvalue

Direction	Type
Input	String

This field contains the key value protected under an RSA public key. This byte-length string is left-justified within the *PKA_enciphered_keyvalue* parameter.

data_structure_length

Direction	Type
Input	Integer

The value must be 0.

data_structure

Direction	Type
Input	String

This field is currently ignored.

key_identifier_length

Direction	Type
Input	Integer

The length of the *key_identifier* parameter. When the *key_identifier* is a key label, this field specifies the length of the label. The maximum size that you can specify is 3500 bytes.

key_identifier

Direction	Type
Input	String

For PKA keys, an internal RSA private key token, the label of an internal RSA private key token, or an external RSA private key token containing a clear RSA private key in modulus-exponent or Chinese Remainder Theorem format.

For secure PKCS #11 keys, this is the 44-byte handle of the private key, prefixed with an EBCDIC equal sign character (‘=’ or x’7E’), and padded on the right with spaces for a total length of 64 bytes.

The corresponding public key was used to wrap the key value.

target_keyvalue_length

Direction	Type
Input/Output	Integer

The length of the *target_keyvalue* parameter. The maximum size that you can specify is 512 bytes. On return, this field is updated with the actual length of *target_keyvalue*.

If ZERO-PAD is specified, this length will be the same as the RSA modulus byte length.

target_keyvalue

Direction	Type
Output	String

This field will contain the decrypted, deformatted key value. If ZERO-PAD is specified, the decrypted key value, including leading zeros, will be returned.

Restrictions

The exponent of the RSA public key must be odd.

Authorization

To use this service with a secure PKCS #11 private key that is a public object, the caller must have SO (READ) authority or USER (READ) authority (any access) to the containing PKCS #11 token.

To use this service with a secure PKCS #11 private key that is a private object, the caller must have USER (READ) authority (user access) to the containing PKCS #11 token.

See *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications* for more information on the SO and User PKCS #11 roles.

Usage notes

SAF may be invoked to verify the caller is authorized to use this callable service, the key label, or internal secure key tokens that are stored in the CKDS, PKDS, or TKDS.

PKA RSA private key must be enabled for key management functions. Secure PKCS #11 private keys must be enabled for decryption.

For PKA keys, the hardware configuration sets the limit on the modulus size of keys for key management; thus, this service will fail if the RSA key modulus bit length exceeds this limit.

Access control points

For PKA keys, the **PKA Decrypt** access control point controls the function of this service.

There are access control points to disable a formatting rule. All of these controls are disabled in the domain role. Enabling these access control points will cause the request for the keyword to fail.

Table 11. PKA decrypt access controls

Access control point	Rule array keyword
PKA Decrypt – Disallow PKCS-1.2	PKCS-1.2
PKA Decrypt – Disallow PKCSOAEP	PKCSOAEP
PKA Decrypt – Disallow ZEROPAD	ZEROPAD

For secure PKCS #11 private keys, see 'PKCS #11 Access Control Points' in *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications* for more information on the access control points of the Enterprise PKCS #11 coprocessor.

Required hardware

This table lists the required cryptographic hardware for each server type and describes restrictions for this callable service.

Table 12. PKA decrypt required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor	RSA keys with moduli greater than 2048-bit length are not supported.
	Crypto Express2 Coprocessor	Keywords PKCSOAEP, SHA-1, and SHA-256 are not supported.
	PCI Cryptographic Accelerator	Only clear RSA private keys are supported. RSA keys with moduli greater than 2048-bit length are not supported.
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	RSA key support with moduli within the range 2048-bit to 4096-bit requires the Nov. 2007 or later licensed internal code (LIC). Keywords PKCSOAEP, SHA-1, and SHA-256 are not supported.
	Crypto Express2 Accelerator	Only clear RSA private keys are supported. RSA keys with moduli greater than 2048-bit length are not supported.
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	RSA key support with moduli within the range 2048-bit to 4096-bit requires the Nov. 2007 or later licensed internal code (LIC). Keywords PKCSOAEP, SHA-1, and SHA-256 are not supported.
	Crypto Express2 Accelerator Crypto Express3 Accelerator	Only clear RSA private keys are supported. RSA keys with moduli greater than 2048-bit length are not supported.
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	Keywords PKCSOAEP, SHA-1, and SHA-256 are not supported.
	Crypto Express3 Accelerator	Only clear RSA private keys are supported. RSA clear key support with moduli within the range 2048-bit to 4096-bit requires the Sep. 2011 or later licensed internal code (LIC).
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor (CEX4C)	Keywords PKCSOAEP, SHA-1, and SHA-256 require the June 2015 or later licensed internal code (LIC).
	Crypto Express3 Accelerator Crypto Express4 Accelerator (CEX4A)	Only clear RSA private keys are supported.
	Crypto Express4 Enterprise PKCS #11 coprocessor (CEX4P)	Required to use a secure PKCS #11 private key. Keywords ZEROPAD, PKCSOAEP, SHA-1, and SHA-256 are not supported.

Table 12. PKA decrypt required hardware (continued)

Server	Required cryptographic hardware	Restrictions
IBM z13	Crypto Express5 CCA Coprocessor (CEX5C)	Keywords PKCSOAEP, SHA-1, and SHA-256 require the July 2015 or later licensed internal code (LIC).
	Crypto Express5 CCA Accelerator (CEX5A)	Only clear RSA private keys are supported.
	Crypto Express5 Enterprise PKCS #11 coprocessor (CEX5P)	Required to use a secure PKCS #11 private key. Keywords ZEROPAD, PKCSOAEP, SHA-1, and SHA-256 are not supported.

PKA Encrypt (CSNDPKE and CSNFPKE)

This callable service encrypts a supplied clear key value under an RSA public key. The rule array keyword specifies the format of the key prior to encryption.

The callable service name for AMODE(64) invocation is CSNFPKE.

Format

```
CALL CSNDPKE(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             rule_array_count,
             rule_array,
             keyvalue_length,
             keyvalue,
             data_structure_length,
             data_structure,
             PKA_key_identifier_length,
             PKA_key_identifier,
             PKA_enciphered_keyvalue_length,
             PKA_enciphered_keyvalue)
```

Parameters

return_code

Direction	Type
Output	Integer

The return code specifies the general result of the callable service.

reason_code

Direction	Type
Output	Integer

The reason code specifies the result of the callable service that is returned to the application program. Each return code has different reason codes that are assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

The length of the data that is passed to the installation exit. The data is identified in the *exit_data* parameter.

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. This value can be 1, 2, or 3.

rule_array

Direction	Type
Input	String

A keyword that provides control information to the callable service. The keyword is left-justified in an 8-byte field and padded on the right with blanks.

Table 13. Keywords for PKA Encrypt

Keyword	Meaning
Formatting Method (required) specifies the method to use to format the key value prior to encryption.	
PKCS-1.2	RSA PKCS #1 V1.5 block type 02 format will be used to format the supplied key value.
ZERO-PAD	The key value will be padded on the left with binary zeros to the length of the PKA key modulus. The exponent of the public key must be odd.
MRP	The key value will be padded on the left with binary zeros to the length of the PKA key modulus. The RSA public key may have an even or odd exponent.
PKCSOAEP	Specifies to format the data using the RSAES-OAEP encoding scheme defined in the RSA PKCS #1 V2.0 standard. The formatted key is encrypted by the RSA public-key provided as the transport key and returned as an opaque data buffer.
Hash Method (one required when PKCSOAEP is specified. Otherwise, not allowed.)	
SHA-1	Specifies to use the SHA-1 method to calculate the OAEP message digest.
SHA-256	Specifies to use the SHA-256 method to calculate the OAEP message digest.
Key Rule (Optional)	

Table 13. Keywords for PKA Encrypt (continued)

Keyword	Meaning
KEYIDENT	This indicates that the value in the *keyvalue* field is the label of clear tokens in the CKDS. The *keyvalue_length* must be 64.

keyvalue_length

Direction	Type
Input	Integer

The length of the *keyvalue* parameter. The maximum field size is 512 bytes. The actual maximum size depends on the modulus length of *PKA_key_identifier* and the formatting method you specify in the *rule_array* parameter. When key rule KEYIDENT is specified, then the *keyvalue_length* parameter is required to be 64 bytes.

keyvalue

Direction	Type
Input	String

This field contains the supplied clear key value to be encrypted under the *PKA_key_identifier*. When key rule KEYIDENT is specified, the *keyvalue* parameter is assumed to contain a label for a valid CKDS clear key token.

data_structure_length

Direction	Type
Input	Integer

This value must be 0.

data_structure

Direction	Type
Input	String

This field is currently ignored.

PKA_key_identifier_length

Direction	Type
Input	Integer

The length of the *PKA_key_identifier* parameter. When the *PKA_key_identifier* is a key label, this field specifies the length of the label. The maximum size that you can specify is 3500 bytes.

PKA_key_identifier

Direction	Type
Input	String

The RSA public or private key token or the label of the RSA public or private key to be used to encrypt the supplied key value.

PKA_enciphered_keyvalue_length

Direction	Type
Input/Output	Integer

The length of the *PKA_enciphered_keyvalue* parameter in bytes. The maximum size that you can specify is 512 bytes. On return, this field is updated with the actual length of *PKA_enciphered_keyvalue*.

This length should be the same as the modulus length of the *PKA_key_identifier*.

PKA_enciphered_keyvalue

Direction	Type
Output	String

This field contains the key value protected under an RSA public key. This byte-length string is left-justified within the *PKA_enciphered_keyvalue* parameter.

Restrictions

The exponent for RSA public keys must be odd. When the modulus is greater than 2048, the public key exponent must be 3 or 65537.

Usage notes

- SAF may be invoked to verify the caller is authorized to use this callable service, the key label, or internal secure key tokens that are stored in the CKDS or PKDS.
- For RSA DSI PKCS #1 formatting, the key value length must be at least 11 bytes less than the modulus length of the RSA key.
- The hardware configuration sets the limit on the modulus size of keys for key management; thus, this service will fail if the RSA key modulus bit length exceeds this limit.
- The key value to be encrypted must be smaller than the modulus in the *PKA_key_identifier*.

Access control point

The **PKA Encrypt** access control point controls the function of this service.

There are access control points to disable a formatting rule. All of these controls are disabled in the domain role. Enabling these access control points will cause the request for the keyword to fail.

Table 14. PKA encrypt access controls

Access control point	Rule array keyword
PKA Encrypt – Disallow MRP	MRP
PKA Encrypt – Disallow PKCS-1.2	PKCS-1.2
PKA Encrypt – Disallow PKCSOAEP	PKCSOAEP
PKA Encrypt – Disallow ZEROPAD	ZEROPAD

Required hardware

This table lists the required cryptographic hardware for each server type and describes restrictions for this callable service.

Table 15. PKA encrypt required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor	Routed to a PCICA if one is available (ZERO-PAD and MRP only).
	Crypto Express2 Coprocessor	RSA keys with moduli greater than 2048-bit length are not supported. Keywords PKCSOAEP, SHA-1, and SHA-256 are not supported.
	PCI Cryptographic Accelerator	PKCS-1.2 keyword not supported. RSA keys with moduli greater than 2048-bit length are not supported.
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	Routed to a CEX2A if one is available (ZERO-PAD and MRP only). RSA key support with moduli within the range 2048-bit to 4096-bit requires the Nov. 2007 or later licensed internal code (LIC). Keywords PKCSOAEP, SHA-1, and SHA-256 are not supported.
	Crypto Express2 Accelerator	PKCS-1.2 keyword not supported. RSA keys with moduli greater than 2048-bit length are not supported.
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor	Routed to a CEX2A or CEX3A if one is available (ZERO-PAD and MRP only). RSA key support with moduli within the range 2048-bit to 4096-bit requires the Nov. 2007 or later licensed internal code (LIC). Keywords PKCSOAEP, SHA-1, and SHA-256 are not supported.
	Crypto Express3 Coprocessor	
	Crypto Express2 Accelerator Crypto Express3 Accelerator	PKCS-1.2 keyword not supported. RSA keys with moduli greater than 2048-bit length are not supported.
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	Routed to a CEX2A or CEX3A if one is available (ZERO-PAD and MRP only). Keywords PKCSOAEP, SHA-1, and SHA-256 are not supported.
	Crypto Express3 Accelerator	PKCS-1.2 keyword not supported. RSA clear key support with moduli within the range 2048-bit to 4096-bit requires the Sep. 2011 or later licensed internal code (LIC).

Table 15. PKA encrypt required hardware (continued)

Server	Required cryptographic hardware	Restrictions
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	Routed to a CEX3A or CEX4A if one is available (PKCSOAEP, ZERO-PAD, and MRP only). Keywords PKCSOAEP, SHA-1, and SHA-256 require the June 2015 or later licensed internal code (LIC).
	Crypto Express3 Accelerator Crypto Express4 CCA Accelerator	PKCS-1.2 keyword not supported.
IBM z13	Crypto Express5 CCA Coprocessor (CEX5C)	Routed to a CEX5A if one is available (PKCSOAEP, ZERO-PAD, and MRP only). Keywords PKCSOAEP, SHA-1, and SHA-256 require the July 2015 or later licensed internal code (LIC).
	Crypto Express5 CCA Accelerator (CEX5A)	PKCS-1.2 keyword not supported.

Access control points and callable services

The following table lists usage information using the following abbreviations:

AE Always enabled, cannot be disabled.

ED Enabled by default.

DD Disabled by default.

SC Usage of this access control point requires special consideration.

This following table lists access control points that affect specific services indicated in the access control point name. There is a description of the usage of the access control point in the Usage Notes section of the callable service description.

Note: If the domain role has been changed via the TKE workstation, all new access control points are disabled by default.

Table 16. Access control points – Callable Services

Name	Callable Service	Usage
PKA Decrypt – Disallow PKCS-1.2	CSNDPKD / CSNFPKD	DD
PKA Decrypt - Disallow ZEROPAD	CSNDPKD / CSNFPKD	DD
PKA Decrypt – Disallow PKCSOAEP	CSNDPKD / CSNFPKD	DD
PKA Encrypt – Disallow PKCS-1.2	CSNDPKE / CSNFPKE	DD
PKA Encrypt – Disallow ZEROPAD	CSNDPKE / CSNFPKE	DD
PKA Encrypt – Disallow MRP	CSNDPKE / CSNFPKE	DD
PKA Encrypt – Disallow PKCSOAEP	CSNDPKE / CSNFPKE	DD

PKA Encrypt

Chapter 4. Update of z/OS Cryptographic Services ICSF Administrator's Guide, SC14-7506-03, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-03, for the CCA service algorithm updates provided by this APAR. Refer to this source document if background information is needed.

Displaying coprocessor hardware status

You can use the ICSF panels to view the status of the cryptographic coprocessor key registers, the master key verification patterns, and other information about the cryptographic hardware. You can use this information for master key management.

When you enter and activate an AES, DES, ECC or RSA master key, you change the status of the registers. The cryptographic facility contains three key registers: one for the old master key, one for the new, and one for the current. The current master key register contains the active master key. The old master key is not lost when a new master key is loaded.

Note: The master key verification and hash patterns are displayed as hexadecimal digit strings on the Hardware Status panel. The number of valid digits is determined by the MASTERKCVLEN options data set keyword. See *z/OS Cryptographic Services ICSF System Programmer's Guide* for additional information.

Updated ICSF panels

The following ICSF panels have been updated:

ICSF Primary Menu panel

```
HCR77B0 ------------- Integrated Cryptographic Service Facility ---------

Enter the number of the desired option.

   1  COPROCESSOR MGMT    -  Management of Cryptographic Coprocessors
   2  KDS MANAGEMENT      -  Master key set or change, KDS processing
   3  OPSTAT              -  Installation options
   4  ADMINCNTL           -  Administrative Control Functions
   5  UTILITY             -  ICSF Utilities
   6  PPINIT              -  Pass Phrase Master Key/KDS Initialization
   7  TKE                 -  TKE PKA Direct Key Load
   8  KGUP                -  Key Generator Utility processes
   9  UDX MGMT            -  Management of User Defined Extensions

      Licensed Materials - Property of IBM
      5650-ZOS (C) Copyright IBM Corp. 1989, 2015.
      US Government Users Restricted Rights - Use, duplication or
      disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

 Press ENTER to go to the selected option.
 Press END   to exit to the previous menu.
```

CSFCMP00 — Coprocessor Management panel

```
CSFCMP00 ---------------- ICSF Coprocessor Management -------- Row 1 to 1 of 1

 Select the cryptographic features to be processed and press ENTER.
 Action characters are: A, D, E, K, R, S and V. See the help panel for details.

  CRYPTO    SERIAL
  FEATURE   NUMBER    STATUS                AES   DES   ECC   RSA   P11
  -------   --------  --------------------  ---   ---   ---   ---   ---
.   5C37    99EA6008  Master key incorrect   E     C     C     C
```

CSFSOP10 — Installation Options panel

```
CSFSOP10 ------------- ICSF - Installation Option Display -- Row 1 to 19 of 19
COMMAND ===>                                                  SCROLL ===> PAGE
         Active CKDS: CRYPTOR2.HCRICSF.CKDS
         Active PKDS: CRYPTOR2.HCRICSF.PKDS
         Active TKDS: CRYPTOR2.HCRICSF.TKDS
  OPTION                                                      CURRENT VALUE
  ------                                                      -------------
  CHECKAUTH      RACF check authorized callers                NO
  COMPAT         Allow CUSP/PCF compatibility                 NO
  CTRACE         CTRACE parmlib used at ICSF startup          CTICSF00
  DEFAULTWRAP    Default symmetric key wrapping - internal    ORIGINAL
  DEFAULTWRAP    Default symmetric key wrapping - external    ORIGINAL
  DOMAIN         Current domain index or usage domain index   0
  FIPSMODE       Operate PKCS #11 in FIPS 140-2 mode          NO,FAIL(NO)
  HDRDATE        Update the header record for all I/O ops     NO
  KDSREFDAYS     Number of days between reference updates     1
  KEYARCHMSG     Message for archived KDS record reference    NO
  MASTERKCVLEN   Length of master key verification patterns   6
  MAXSESSOBJECTS Max non-auth pgm PKCS #11 session objects     65535
  REASONCODES    Source of callable services reason codes     ICSF
  RNGCACHE       Random Number Generate cache enabled         YES
  SSM            Special Secure Mode enabled                  NO
  SYSPLEXCKDS    Sysplex consistency for CKDS updates         YES,FAIL(YES)
  SYSPLEXPKDS    Sysplex consistency for PKDS updates         YES,FAIL(YES)
  SYSPLEXTKDS    Sysplex consistency for TKDS updates         YES,FAIL(YES)
  USERPARM       User specified parameter data                USERPARM
  WAITLIST       Source of CICS Wait List if CICS installed   default
****************************** Bottom of data ********************************
```

Displaying the EP11 domain roles

Use the ICSF panels to display the enabled access control points for the Enterprise PKCS #11 coprocessor. All the access control points enabled will be listed.

1. Select option 1, COPROCESSOR MGMT, on the "ICSF Primary Menu panel" on page 29.
2. The Coprocessor Management panel appears. Refer to Figure 1 on page 31.

```
CSFGCMP0 ---------------- ICSF Coprocessor Management -------------

Select the coprocessors to be processed and press ENTER.
Action characters are: A, D, E, K, R, S, and V. See the help panel for details.

              Serial
CoProcessor    Number    Status               AES   DES   ECC   RSA    P11
-----------  ---------  ------               ---   ---   ---   ----   ___
__ 4P00      16BA6173   Active                                         A
__ 4C01      16BBP109   Master key incorrect  U     U     U     U
__ 4A02      N/A        Active
R  4P03      16BBP103   Active                                         A
__ 3C04      99001650   Active                A     A     A     A
__ 3C05      99001652   Active                A     A     A     A
__ 3A06      N/A        Active
__ 3C07      99002519   Master key incorrect  U     U     U     U
__ 3C08      91008972   Active                A     A     A     A
__ 3C09      90008301   Active                A     A     A     A
__ 4C14      16C35329   Active                A     A     A     A
__ 4P15      16C2H305   Active                                         A
```

Figure 1. Coprocessor Management Panel

3. Select the desired coprocessor by entering an 'R' to the left of the coprocessor. Press enter and the Status Display panel appears (Figure 2).

```
CSFCMP30 ---------------- ICSF Status Display -------------
COMMAND ===>

 Enabled access control points from the default role for 4P03 domain 0

Allow addition (activation) of Control Points
Allow backend to save semi-retained keys
Allow changes to key objects (usage flags only)
Allow clear passphrases for password-based-encryption
Allow clear public keys as non-attribute bound wrapping keys
Allow dual-function keys - digital signature and data encryption
Allow dual-function keys - key wrapping and data encryption
Allow dual-function keys - key wrapping and digital signature
Allow key derivation
Allow keywrap without attribute-bindings
Allow mixing external seed to RNG
Allow non-administrators to mark key objects TRUSTED
Allow non-administrators to mark public key objects ATTRBOUND
Allow non-BSI algorithms (as of 2009)
Allow non-BSI algorithms (as of 2011)
Allow non-FIPS-approved algorithms (as of 2011)
Allow removal (deactivation) of Control Points
Allow wrapping of stronger keys by weaker keys
```

Figure 2. CSFCMP30 — ICSF - Status Display

For the Access Control Points that are available on the Enterprise PKCS #11 coprocessor, see PKCS #11 Coprocessor Access Control Points in *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*.

Display CCA domain roles

Use the ICSF panels to display the coprocessor role for the coprocessor. All the access control points enabled will be listed.

1. Select option 1, COPROCESSOR MGMT, on the "ICSF Primary Menu panel" on page 29.

2. The Coprocessor Management panel appears. Refer to Figure 3.

```
 CSFGCMPOO ---------------- ICSF Coprocessor Management -------- Row 1 to 7 of 7
 COMMAND ===>

Select the cryptographic features to be processed and press ENTER.
Action characters are: A, D, E, K, R, S, and V. See the help panel for details.

  CRYPTO         SERIAL
  FEATURE        NUMBER         STATUS              AES DES ECC RSA P11
  -------       --------        ------------------- --- --- --- --- ---
. 4C00          16BA6173        Active               I   A   A   A
. 4C01          16BA6174        Master key incorrect I   A   C   E
. 4C02          16BA6175        Master key incorrect I   A   C   E
. 4A03          N/A             Active
. 4C04          16BA6199        Deactivated
. 4P05          16BA6200        Active                               A
. 4P06          16BA6201        Master key incorrect                 U
****************************** Bottom of data ********************************
```

Figure 3. Coprocessor Management Panel

3. Select the desired coprocessor by entering an 'R' or 'V' to the left of the coprocessor.

   The display shown when 'R' is used (see Figure 4 on page 33) lists all of the enabled access controls in alphabetic order.

   The display shown when 'V' is used (see Figure 8 on page 37) lists all of the enabled access controls and the offset within the role.The list can be ordered by the access control name or the offset. Press enter and the Domain Role Display panel appears.

   Note: A TKE is required in order to change the coprocessor role. See *z/OS Cryptographic Services ICSF TKE PCIX Workstation User's Guide*.

```
Access Control Manager - Read role
Authorize UDX
AES Master Key - Clear new master key register
AES Master Key - Combine key parts
AES Master Key - Load first key part
AES Master Key - Set master key
Clear Key Import/Multiple Clear Key Import - DES
Clear PIN Encrypt
Clear PIN Generate - GBP
Clear PIN Generate - Interbank
Clear PIN Generate - VISA PVV
Clear PIN Generate - 3624
Clear PIN Generate Alternate - VISA PVV
Clear PIN Generate Alternate - 3624 Offset
Control Vector Translate
Cryptographic Variable Encipher
CKDS Conversion2 - Allow use of REFORMAT
CKDS Conversion2 - Allow wrapping override keywords
CKDS Conversion2 - Convert from enhanced to original
CVV Key Combine
CVV Key Combine - Allow wrapping override keywords
CVV Key Combine - Permit mixed key types
Data Key Export
Data Key Export - Unrestricted
Data Key Import
Data Key Import - Unrestricted
Decipher - DES
Digital Signature Generate
Digital Signature Verify
Diversified Key Generate - Allow wrapping override keywords
Diversified Key Generate - CLR8-ENC
Diversified Key Generate - Single length or same halves
Diversified Key Generate - SESS-XOR
Diversified Key Generate - TDES-DEC
Diversified Key Generate - TDES-ENC
Diversified Key Generate - TDES-XOR
Diversified Key Generate - TDESEMV2/TDESEMV4
DATAM Key Management Control
DES Master Key - Clear new master key register
DES Master Key - Combine key parts
DES Master Key - Load first key part
DES Master Key - Set master key   Encipher - DES
Encrypted PIN Generate - GBP
Encrypted PIN Generate - Interbank
Encrypted PIN Generate - 3624
Encrypted PIN Translate - Reformat
```

Figure 4. CCA Coprocessor Role Display panel

```
Encrypted PIN Translate - Translate
Encrypted PIN Verify - GBP
Encrypted PIN Verify - Interbank
Encrypted PIN Verify - VISA PVV
Encrypted PIN Verify - 3624
ECC Diffie-Hellman
ECC Diffie-Hellman - Allow key wrap override
ECC Diffie-Hellman - Allow BP Curve 160
ECC Diffie-Hellman - Allow BP Curve 192
ECC Diffie-Hellman - Allow BP Curve 224
ECC Diffie-Hellman - Allow BP Curve 256
ECC Diffie-Hellman - Allow BP Curve 320
ECC Diffie-Hellman - Allow BP Curve 384
ECC Diffie-Hellman - Allow BP Curve 512
ECC Diffie-Hellman - Allow Prime Curve 192
ECC Diffie-Hellman - Allow Prime Curve 224
ECC Diffie-Hellman - Allow Prime Curve 256
ECC Diffie-Hellman - Allow Prime Curve 384
ECC Diffie-Hellman - Allow Prime Curve 521
ECC Diffie-Hellman - Allow PASSTHRU
ECC Master Key - Clear new master key register
ECC Master Key - Combine key parts
ECC Master Key - Load first key part
ECC Master Key - Set master key
HMAC Generate - SHA-1
HMAC Generate - SHA-224
HMAC Generate - SHA-256
HMAC Generate - SHA-384
HMAC Generate - SHA-512
HMAC Verify - SHA-1
HMAC Verify - SHA-224
HMAC Verify - SHA-256
HMAC Verify - SHA-384
HMAC Verify - SHA-512
Key Export
Key Export - Unrestricted
Key Generate - Key set
Key Generate - Key set extended
Key Generate - OP
Key Generate - SINGLE-R
Key Generate2 - Key set
Key Generate2 - OP
Key Import
Key Import - Unrestricted
Key Part Import - first key part
Key Part Import - middle and last
Key Part Import - Allow wrapping override keywords
Key Part Import - ADD-PART
Key Part Import - COMPLETE
Key Part Import - Unrestricted
Key Part Import2 - Add last required key part
Key Part Import2 - Add optional key part
Key Part Import2 - Add second of 3 or more key parts
Key Part Import2 - Complete key
Key Part Import2 - Load first key part, require 1 key parts
Key Part Import2 - Load first key part, require 2 key parts
Key Part Import2 - Load first key part, require 3 key parts
Key Test and Key Test2
Key Test2 - AES, ENC-ZERO
Key Translate
Key Translate2
Key Translate2 - Allow use of REFORMAT
Key Translate2 - Allow wrapping override keywords
Multiple Clear Key Import - Allow wrapping override keywords
Multiple Clear Key Import/Multiple Secure Key Import - AES
Multiple Secure Key Import - Allow wrapping override keywords
MAC Generate
MAC Verify
```

Figure 5. CCA Coprocessor Role Display panel - part 2

```
NOCV KEK usage for export-related functions
NOCV KEK usage for import-related functions
Operational Key Load
Prohibit Export
Prohibit Export Extended
PCF CKDS conversion utility
PCF CKDS Conversion - Allow wrapping override keywords
PIN Change/Unblock - change EMV PIN with IPINENC
PIN Change/Unblock - change EMV PIN with OPINENC
PKA Decrypt
PKA Encrypt
PKA Key Generate
PKA Key Generate - Clear ECC keys
PKA Key Generate - Clear RSA keys
PKA Key Generate - Clone
PKA Key Generate - Permit Regeneration Data
PKA Key Generate - Permit Regeneration Data Retain
PKA Key Import
PKA Key Import - Import an external trusted block
PKA Key Token Change RTCMK
PKA Key Translate - from source EXP KEK to target EXP KEK
PKA Key Translate - from source IMP KEK to target EXP KEK
PKA Key Translate - from source IMP KEK to target IMP KEK
PKA Key Translate - from CCA RSA to SC CRT Format
PKA Key Translate - from CCA RSA to SC ME Format
PKA Key Translate - from CCA RSA to SC Visa Format
Reencipher CKDS2
Reencipher CKDS
Reencipher PKDS
Remote Key Export - Gen or export a non-CCA node key
Restrict Key Attribute - Export Control
Restrict Key Attribute - Permit setting the TR-31 export bit
Retained Key Delete
Retained Key List
RSA Master Key - Clear new master key register
RSA Master Key - Combine key parts
RSA Master Key - Load first key part
RSA Master Key - Set master key
Secure Key Import - DES,IM
Secure Key Import - DES,OP
Secure Key Import2 - IM
Secure Key Import2 - OP
Secure Messaging for Keys
Secure Messaging for PINs
Symmetric token wrapping - external enhanced method
Symmetric token wrapping - external original method
Symmetric token wrapping - internal enhanced method
Symmetric token wrapping - internal original method
Symmetric Algorithm Decipher - secure AES keys
Symmetric Algorithm Encipher - secure AES keys
Symmetric Key Encipher/Decipher - Encrypted AES keys
Symmetric Key Encipher/Decipher - Encrypted DES keys
Symmetric Key Export - AES, PKCSOAEP, PKCS-1.2
Symmetric Key Export - AES, ZERO-PAD
Symmetric Key Export - AES,PKOAEP2
Symmetric Key Export - AESKW
Symmetric Key Export - DES, PKCS-1.2
Symmetric Key Export - DES, ZERO-PAD
Symmetric Key Export - HMAC,PKOAEP2
```

Figure 6. CCA Coprocessor Role Display panel – part 3

```
Symmetric Key Generate - Allow wrapping override keywords
Symmetric Key Generate - AES, PKCSOAEP, PKCS-1.2
Symmetric Key Generate - AES, ZERO-PAD
Symmetric Key Generate - DES, PKA92
Symmetric Key Generate - DES, PKCS-1.2
Symmetric Key Generate - DES, ZERO-PAD
Symmetric Key Import - Allow wrapping override keywords
Symmetric Key Import - AES, PKCSOAEP, PKCS-1.2
Symmetric Key Import - AES, ZERO-PAD
Symmetric Key Import - DES, PKA92 KEK
Symmetric Key Import - DES, PKCS-1.2
Symmetric Key Import - DES, ZERO-PAD
Symmetric Key Import2 - AES,PKOAEP2
Symmetric Key Import2 - AESKW
Symmetric Key Import2 - HMAC,PKOAEP2
Symmetric Key Token Change - RTCMK
Symmetric Key Token Change2 - RTCMK
SET Block Compose
SET Block Decompose
SET Block Decompose - PIN Extension IPINENC
SET Block Decompose - PIN Extension OPINENC
Transaction Validation - Generate
Transaction Validation - Verify CSC-3
Transaction Validation - Verify CSC-4
Transaction Validation - Verify CSC-5
Trusted Block Create - Activate an inactive block
Trusted Block Create - Create Block in inactive form
TR31 Export - Permit any CCA key if INCL-CV is specified
TR31 Export - Permit version A TR-31 key blocks
TR31 Export - Permit version B TR-31 key blocks
TR31 Export - Permit version C TR-31 key blocks
TR31 Export - Permit DATA to C0:G/C
TR31 Export - Permit DATA to D0:B
TR31 Export - Permit DKYGENKY:DKYL0+DALL to E4
TR31 Export - Permit DKYGENKY:DKYL0+DALL to E5
TR31 Export - Permit DKYGENKY:DKYL0+DDATA to E4
TR31 Export - Permit ENCIPHER/DECIPHER/CIPHER to D0:E/D/B
TR31 Export - Permit IPINENC to P0:D
TR31 Export - Permit KEYGENKY:UKPT to B0
TR31 Export - Permit MAC/DATA/DATAM to M1:G/C
TR31 Export - Permit MAC/DATA/DATAM to M3:G/C
TR31 Export - Permit MAC/MACVER:ANY-MAC to C0:G/C/V
TR31 Export - Permit MACVER/DATAMV to M0:V
TR31 Export - Permit MACVER/DATAMV to M1:V
TR31 Export - Permit MACVER/DATAMV to M3:V
TR31 Export - Permit OPINENC to P0:E
TR31 Export - Permit PINGEN:NO-SPEC/IBM-PIN/IBM-PINO to V1
TR31 Export - Permit PINGEN:NO-SPEC/VISA-PVV to V2
TR31 Export - Permit PINVER:NO-SPEC/IBM-PIN/IBM-PINO to V1
TR31 Export - Permit PINVER:NO-SPEC/VISA-PVV to V2
TR31 Import - Permit override of default wrapping method
TR31 Import - Permit version A TR-31 key blocks
TR31 Import - Permit version B TR-31 key blocks
TR31 Import - Permit version C TR-31 key blocks
TR31 Import - Permit E4 to DKYGENKY:DKYL0+DDATA
TR31 Import - Permit M0/M1/M3 to MAC/MACVER:ANY-MAC
TR31 Import - Permit P0:D to IPINENC
TR31 Import - Permit P0:E to OPINENC
TR31 Import - Permit V1 to PINGEN:IBM-PIN/IBM-PINO
TR31 Import - Permit V1 to PINVER:IBM-PIN/IBM-PINO
TR31 Import - Permit V2 to PINGEN:VISA-PVV
TR31 Import - Permit V2 to PINVER:VISA-PVV
UKPT - PIN Verify, PIN Translate
VISA CVV Generate
VISA CVV Verify
```

Figure 7. CCA Coprocessor Role Display panel – part 4

```
CSFCMP32 ----------------- ICSF - Domain Role Display ----- Row 1 to 35 of 278
COMMAND ===>

Sort by control value (Y/N) ==> N
Press END   to exit to the previous menu.

Enabled access controls from the domain role for 5C37 domain 0

0x0116   Access Control Manager - Read role
0x02B1   Authentication Parameter Generate
0x0240   Authorize UDX
0x0124   AES Master Key - Clear new master key register
0x0126   AES Master Key - Combine key parts
0x0125   AES Master Key - Load first key part
0x0128   AES Master Key - Set master key
0x01C0   Cipher Text Translate2
0x01C1   Cipher Text Translate2 - Allow translate from AES to TDES
0x01C2   Cipher Text Translate2 - Allow translate to weaker AES
0x01C3   Cipher Text Translate2 - Allow translate to weaker DES
0x00C3   Clear Key Import/Multiple Clear Key Import - DES
0x00AF   Clear PIN Encrypt
0x00A1   Clear PIN Generate - GBP
0x00A3   Clear PIN Generate - Interbank
0x00A2   Clear PIN Generate - VISA PVV
0x00A0   Clear PIN Generate - 3624
0x00BB   Clear PIN Generate Alternate - VISA PVV
0x00A4   Clear PIN Generate Alternate - 3624 Offset
0x00D6   Control Vector Translate
0x00DA   Cryptographic Variable Encipher
0x014C   CKDS Conversion2 - Allow use of REFORMAT
0x0146   CKDS Conversion2 - Allow wrapping override keywords
0x0147   CKDS Conversion2 - Convert from enhanced to original
0x0155   CVV Key Combine
0x0156   CVV Key Combine - Allow wrapping override keywords
0x0157   CVV Key Combine - Permit mixed key types
0x010A   Data Key Export
0x0277   Data Key Export - Unrestricted
0x0109   Data Key Import
0x027C   Data Key Import - Unrestricted
0x000F   Decipher - DES
0x0100   Digital Signature Generate
0x0101   Digital Signature Verify
0x02B8   Diversified Key Generate - TDES-CBC
0x013D   Diversified Key Generate - Allow wrapping override keywords
0x0040   Diversified Key Generate - CLR8-ENC
0x0044   Diversified Key Generate - Single length or same halves
0x0043   Diversified Key Generate - SESS-XOR
0x0042   Diversified Key Generate - TDES-DEC
0x0041   Diversified Key Generate - TDES-ENC
0x0045   Diversified Key Generate - TDES-XOR
0x0046   Diversified Key Generate - TDESEMV2/TDESEMV4
0x02D2   Diversified Key Generate2 - MK-OPTC
0x02CC   Diversified Key Generate2 - SESS-ENC
0x0275   DATAM Key Management Control
0x0032   DES Master Key - Clear new master key register
0x0019   DES Master Key - Combine key parts
0x0018   DES Master Key - Load first key part
0x001A   DES Master Key - Set master key
0x02C6   DK Deterministic PIN Generate
0x02CE   DK Migrate PIN
0x02C5   DK PAN Modify in Transaction
0x02C7   DK PAN Translate
```

Figure 8. CCA Domain Role Display panel

```
0x02C2   DK PIN Change
0x02C1   DK PIN Verify
0x02C3   DK PRW Card Number Update
0x02C4   DK PRW CMAC Generate
0x02C0   DK Random PIN Generate
0x02C8   DK Regenerate PRW
0x000E   Encipher - DES
0x00B1   Encrypted PIN Generate - GBP
0x00B2   Encrypted PIN Generate - Interbank
0x00B0   Encrypted PIN Generate - 3624
0x00B7   Encrypted PIN Translate - Reformat
0x00B3   Encrypted PIN Translate - Translate
0x00AC   Encrypted PIN Verify - GBP
0x00AE   Encrypted PIN Verify - Interbank
0x00AD   Encrypted PIN Verify - VISA PVV
0x00AB   Encrypted PIN Verify - 3624
0x0360   ECC Diffie-Hellman
0x0362   ECC Diffie-Hellman - Allow key wrap override
0x0368   ECC Diffie-Hellman - Allow BP Curve 160
0x0369   ECC Diffie-Hellman - Allow BP Curve 192
0x036A   ECC Diffie-Hellman - Allow BP Curve 224
0x036B   ECC Diffie-Hellman - Allow BP Curve 256
0x036C   ECC Diffie-Hellman - Allow BP Curve 320
0x036D   ECC Diffie-Hellman - Allow BP Curve 384
0x036E   ECC Diffie-Hellman - Allow BP Curve 512
0x0363   ECC Diffie-Hellman - Allow Prime Curve 192
0x0364   ECC Diffie-Hellman - Allow Prime Curve 224
0x0365   ECC Diffie-Hellman - Allow Prime Curve 256
0x0366   ECC Diffie-Hellman - Allow Prime Curve 384
0x0367   ECC Diffie-Hellman - Allow Prime Curve 521
0x0361   ECC Diffie-Hellman - Allow PASSTHRU
0x031F   ECC Master Key - Clear new master key register
0x0321   ECC Master Key - Combine key parts
0x0320   ECC Master Key - Load first key part
0x0322   ECC Master Key - Set master key
0x02D0   FPE Decrypt
0x02CF   FPE Encrypt
0x02D1   FPE Translate
0x00E4   HMAC Generate - SHA-1
0x00E5   HMAC Generate - SHA-224
0x00E6   HMAC Generate - SHA-256
0x00E7   HMAC Generate - SHA-384
0x00E8   HMAC Generate - SHA-512
0x00F7   HMAC Verify - SHA-1
0x00F8   HMAC Verify - SHA-224
0x00F9   HMAC Verify - SHA-256
0x00FA   HMAC Verify - SHA-384
0x00FB   HMAC Verify - SHA-512
0x0013   Key Export
0x0276   Key Export - Unrestricted
0x008C   Key Generate - Key set
0x00D7   Key Generate - Key set extended
0x008E   Key Generate - OP
0x00DB   Key Generate - SINGLE-R
0x00EB   Key Generate2 - Key set
0x00EC   Key Generate2 - Key set extended
0x00EA   Key Generate2 - OP
0x0012   Key Import
0x027B   Key Import - Unrestricted
0x001B   Key Part Import - first key part
0x001C   Key Part Import - middle and last
0x0140   Key Part Import - Allow wrapping override keywords
```

Figure 9. CCA Domain Role Display panel - part 2

```
0x0278   Key Part Import - ADD-PART
0x0279   Key Part Import - COMPLETE
0x027A   Key Part Import - Unrestricted
0x029B   Key Part Import2 - Add last required key part
0x029C   Key Part Import2 - Add optional key part
0x029A   Key Part Import2 - Add second of 3 or more key parts
0x029D   Key Part Import2 - Complete key
0x0299   Key Part Import2 - Load first key part, require 1 key parts
0x0298   Key Part Import2 - Load first key part, require 2 key parts
0x0297   Key Part Import2 - Load first key part, require 3 key parts
0x001D   Key Test and Key Test2
0x0021   Key Test2 - AES, ENC-ZERO
0x001F   Key Translate
0x0149   Key Translate2
0x014B   Key Translate2 - Allow use of REFORMAT
0x014A   Key Translate2 - Allow wrapping override keywords
0x0141   Multiple Clear Key Import - Allow wrapping override keywords
0x0129   Multiple Clear Key Import/Multiple Secure Key Import - AES
0x0142   Multiple Secure Key Import - Allow wrapping override keywords
0x0010   MAC Generate
0x0336   MAC Generate2 - AES CMAC
0x0011   MAC Verify
0x0337   MAC Verify2 - AES CMAC
0x0300   NOCV KEK usage for export-related functions
0x030A   NOCV KEK usage for import-related functions
0x0309   Operational Key Load
0x029E   Operational Key Load - Variable-Length Tokens
0x00CD   Prohibit Export
0x0301   Prohibit Export Extended
0x0303   PCF CKDS conversion utility
0x0148   PCF CKDS Conversion - Allow wrapping override keywords
0x00BD   PIN Change/Unblock - change EMV PIN with IPINENC
0x00BC   PIN Change/Unblock - change EMV PIN with OPINENC
0x011F   PKA Decrypt
0x011E   PKA Encrypt
0x0103   PKA Key Generate
0x0326   PKA Key Generate - Clear ECC keys
0x0205   PKA Key Generate - Clear RSA keys
0x0204   PKA Key Generate - Clone
0x027D   PKA Key Generate - Permit Regeneration Data
0x027E   PKA Key Generate - Permit Regeneration Data Retain
0x0104   PKA Key Import
0x0311   PKA Key Import - Import an external trusted block
0x0102   PKA Key Token Change RTCMK
0x031B   PKA Key Translate - from source EXP KEK to target EXP KEK
0x031C   PKA Key Translate - from source IMP KEK to target EXP KEK
0x031D   PKA Key Translate - from source IMP KEK to target IMP KEK
0x031A   PKA Key Translate - from CCA RSA to SC CRT Format
0x0319   PKA Key Translate - from CCA RSA to SC ME Format
0x0318   PKA Key Translate - from CCA RSA to SC Visa Format
0x033A   PKA Key Translate - from CCA RSA CRT to EMV CRT format
0x0338   PKA Key Translate - from CCA RSA CRT to EMV DDA format
0x0339   PKA Key Translate - from CCA RSA CRT to EMV DDAE format
0x00FF   PKA Key Translate - Translate external key token
0x00FE   PKA Key Translate - Translate internal key token
0x02B0   Recover PIN From Offset
0x001E   Reencipher CKDS
0x00F0   Reencipher CKDS2
0x0241   Reencipher PKDS
0x0312   Remote Key Export - Gen or export a non-CCA node key
0x00E9   Restrict Key Attribute - Export Control
0x0154   Restrict Key Attribute - Permit setting the TR-31 export bit
0x0203   Retained Key Delete
0x0230   Retained Key List
0x0060   RSA Master Key - Clear new master key register
```

Figure 10. CCA Domain Role Display panel - part 3

```
0x0054   RSA Master Key - Combine key parts
0x0053   RSA Master Key - Load first key part
0x0057   RSA Master Key - Set master key
0x00DC   Secure Key Import - DES,IM
0x00C4   Secure Key Import - DES,OP
0x00F3   Secure Key Import2 - IM
0x00F2   Secure Key Import2 - OP
0x0273   Secure Messaging for Keys
0x0274   Secure Messaging for PINs
0x013B   Symmetric token wrapping - external enhanced method
0x013C   Symmetric token wrapping - external original method
0x0139   Symmetric token wrapping - internal enhanced method
0x013A   Symmetric token wrapping - internal original method
0x012B   Symmetric Algorithm Decipher - secure AES keys
0x012A   Symmetric Algorithm Encipher - secure AES keys
0x0296   Symmetric Key Encipher/Decipher - Encrypted AES keys
0x0295   Symmetric Key Encipher/Decipher - Encrypted DES keys
0x0130   Symmetric Key Export - AES, PKCSOAEP, PKCS-1.2
0x0131   Symmetric Key Export - AES, ZERO-PAD
0x00FC   Symmetric Key Export - AES,PKOAEP2
0x0327   Symmetric Key Export - AESKW
0x02B3   Symmetric Key Export - AESKWCV
0x0105   Symmetric Key Export - DES, PKCS-1.2
0x023E   Symmetric Key Export - DES, ZERO-PAD
0x00F5   Symmetric Key Export - HMAC,PKOAEP2
0x02B5   Symmetric Key Export with Data
0x02B6   Symmetric Key Export with Data - Special
0x013E   Symmetric Key Generate - Allow wrapping override keywords
0x012C   Symmetric Key Generate - AES, PKCSOAEP, PKCS-1.2
0x012D   Symmetric Key Generate - AES, ZERO-PAD
0x010D   Symmetric Key Generate - DES, PKA92
0x023F   Symmetric Key Generate - DES, PKCS-1.2
0x023C   Symmetric Key Generate - DES, ZERO-PAD
0x0144   Symmetric Key Import - Allow wrapping override keywords
0x012E   Symmetric Key Import - AES, PKCSOAEP, PKCS-1.2
0x012F   Symmetric Key Import - AES, ZERO-PAD
0x0235   Symmetric Key Import - DES, PKA92 KEK
0x0106   Symmetric Key Import - DES, PKCS-1.2
0x023D   Symmetric Key Import - DES, ZERO-PAD
0x02B9   Symmetric Key Import2 - Allow wrapping override keywords
0x00FD   Symmetric Key Import2 - AES,PKOAEP2
0x0329   Symmetric Key Import2 - AESKW
0x02B4   Symmetric Key Import2 - AESKWCV
0x00F4   Symmetric Key Import2 - HMAC,PKOAEP2
0x0090   Symmetric Key Token Change - RTCMK
0x00F1   Symmetric Key Token Change2 - RTCMK
0x010B   SET Block Compose
0x010C   SET Block Decompose
0x0121   SET Block Decompose - PIN Extension IPINENC
0x0122   SET Block Decompose - PIN Extension OPINENC
0x0291   Transaction Validation - Generate
0x0292   Transaction Validation - Verify CSC-3
0x0293   Transaction Validation - Verify CSC-4
0x0294   Transaction Validation - Verify CSC-5
0x0310   Trusted Block Create - Activate an inactive block
0x030F   Trusted Block Create - Create Block in inactive form
0x0158   TR31 Export - Permit any CCA key if INCL-CV is specified
0x014D   TR31 Export - Permit version A TR-31 key blocks
0x014E   TR31 Export - Permit version B TR-31 key blocks
0x014F   TR31 Export - Permit version C TR-31 key blocks
0x0184   TR31 Export - Permit DATA to C0:G/C
0x0186   TR31 Export - Permit DATA to D0:B
0x01AB   TR31 Export - Permit DKYGENKY:DKYL0+DALL to E4
0x01AF   TR31 Export - Permit DKYGENKY:DKYL0+DALL to E5
0x01AA   TR31 Export - Permit DKYGENKY:DKYL0+DDATA to E4
```

Figure 11. CCA Domain Role Display panel - part 4

```
0x0185   TR31 Export - Permit ENCIPHER/DECIPHER/CIPHER to D0:E/D/B
0x0192   TR31 Export - Permit IPINENC to P0:D
0x0180   TR31 Export - Permit KEYGENKY:UKPT to B0
0x018D   TR31 Export - Permit MAC/DATA/DATAM to M1:G/C
0x018F   TR31 Export - Permit MAC/DATA/DATAM to M3:G/C
0x0183   TR31 Export - Permit MAC/MACVER:ANY-MAC to C0:G/C/V
0x018C   TR31 Export - Permit MACVER/DATAMV to M0:V
0x018E   TR31 Export - Permit MACVER/DATAMV to M1:V
0x0190   TR31 Export - Permit MACVER/DATAMV to M3:V
0x0191   TR31 Export - Permit OPINENC to P0:E
0x0196   TR31 Export - Permit PINGEN:NO-SPEC/IBM-PIN/IBM-PINO to V1
0x0198   TR31 Export - Permit PINGEN:NO-SPEC/VISA-PVV to V2
0x0195   TR31 Export - Permit PINVER:NO-SPEC/IBM-PIN/IBM-PINO to V1
0x0197   TR31 Export - Permit PINVER:NO-SPEC/VISA-PVV to V2
0x0153   TR31 Import - Permit override of default wrapping method
0x0150   TR31 Import - Permit version A TR-31 key blocks
0x0151   TR31 Import - Permit version B TR-31 key blocks
0x0152   TR31 Import - Permit version C TR-31 key blocks
0x0178   TR31 Import - Permit E4 to DKYGENKY:DKYL0+DDATA
0x0164   TR31 Import - Permit M0/M1/M3 to MAC/MACVER:ANY-MAC
0x0166   TR31 Import - Permit P0:D to IPINENC
0x0165   TR31 Import - Permit P0:E to OPINENC
0x0169   TR31 Import - Permit V1 to PINGEN:IBM-PIN/IBM-PINO
0x016A   TR31 Import - Permit V1 to PINVER:IBM-PIN/IBM-PINO
0x016B   TR31 Import - Permit V2 to PINGEN:VISA-PVV
0x016C   TR31 Import - Permit V2 to PINVER:VISA-PVV
0x01C8   Unique Key Derive
0x01CA   Unique Key Derive - Override default wrapping
0x00E1   UKPT - PIN Verify, PIN Translate
0x00DF   VISA CVV Generate
0x00E0   VISA CVV Verify
****************************** Bottom of data ********************************
```

Figure 12. CCA Domain Role Display panel - part 5

Printed in USA