z/OSCryptographic Services
Integrated Cryptographic Service Facility

HMAC Enablement (APAR OA33260) and HMAC Coexistence (APAR OA34402)

(February 10, 2011)

Contents

Chapter 1. Overview

Information on HMAC key support is provided in the Cryptographic Support for z/OS V1R10-R12 (ICSF FMID HCR7780) documentation. The documented HMAC key support must be enabled for HCR7780 systems with the PTF for APAR OA33260.

Releases of ICSF prior to HCR7780 require coexistence support for a sysplex sharing a variable-length record format CKDS with a fixed-length record format CKDS. HMAC coexistence can be enabled with the PTF for APAR OA34402.

The HMAC key support described in this document, and enabled by the PTF for APAR OA33260. requires the following MCL level:

Driver 86E, EC N29766, MCL 21, Bundle 22

The HMAC keys are variable length (80-2048 bit) symmetric keys protected by the AES master key and used to generate and verify MACs using the FIPS-198 algorithm. To support these variable length keys, a new variable length record format is available for CKDS records.

To store HMAC keys in an existing CKDS, the CKDS must first be converted to the variable length record format. ICSF provides a CKDS conversion program, CSFCNV2, that converts a fixed length record format CKDS to a variable length record format.

To define a CKDS for variable length records, ICSF provides a new sample CKDS allocation job (member CSFCKD2) in SYS1.SAMPLIB.

The variable-length record format CKDS cannot be used on z800 and z900 systems with the Cryptographic Coprocessor Feature (CCF). On CCF systems:
- ICSF will terminate if started with a variable-length record format CKDS
- attempts to refresh a variable-length record format CKDS will fail
- attempts to reencipher a variable-length record format CKDS will fail
- conversion of a fixed-length to variable-length record format CKDS cannot be performed
- the pass phrase initialization utility (PPINIT) and key generator utility program (KGUP) will fail if you are using a variable-length record format CKDS

Although the HMAC key support information is provided in the ICSF FMID HCR7780 documentation, this document contains alterations to the information previously presented in the following books:
- *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521-15
- *z/OS Cryptographic Services ICSF System Programmer's Guide*, SA22-7520-15
- *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SA22-7522-14
- *z/OS Cryptographic Services ICSF Messages*, SA22-7523-14

Alterations to the information are identified in this document by revision bars (|) in the left margin of the page.

Chapter 2. Update of z/OS Cryptographic Services ICSF Administrator's Guide, SA22-7521-15, information

This chapter contains updates to the document *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521-15. Refer to this source document if background information is needed.

Return and reason codes for the CSFEUTIL program

When you invoke the CSFEUTIL program as a batch job, you receive the return code in a message when the job completes. The meanings of the return codes are:

Return Code	**Meaning**
0	Process successful.
4	Parameters are incorrect.
8	RACF authorization check failed.
12	Process unsuccessful.
72 or 104	CKDS processing has failed. A return code 72 indicates the error was detected in the new KDS. A return code 104 indicates the error was detected with the old KDS.

When the program is invoked from another program, the invoking program receives the reason code in General Register 0 along with the return code in General Register 15. The following list describes the meaning of the reason codes. If a particular reason code is not listed, refer to the listing of ICSF and TSS return and reason codes in the *z/OS Cryptographic Services ICSF Application Programmer's Guide*.

Return code 0 has this reason code:

Reason Code	**Meaning**
36132	CKDS reencipher/Change MK processed only tokens encrypted under the DES master key.

Return code 8 has this reason code:

Reason Code	**Meaning**
16000	Invoker has insufficient RACF access authority to perform function.

Return code 12 has these reason codes:

Reason Code	**Meaning**
36000	Unable to change master key. Check hardware status.
36008	Crypto master key register(s) in improper state.
36020	Input CKDS is empty or not initialized (authentication pattern in the control record is invalid).
36036	The new master key register for Coprocessor 1 (C1) is not full, but C0 is ready and the current master key is valid.

36040	The new master key register for C0 is not full, but C1 is ready and the current master key is valid.
36044	The master key authentication pattern for the CKDS does not match the authentication pattern of the coprocessors, which are not equal.
36048	The master key authentication pattern for the CKDS does not match the authentication pattern of either of the coprocessors, which are not equal.
36052	A valid new master key is present in C0, but its authentication pattern does not match that of C1 or the CKDS, which are equal.
36056	A valid new master key is present in C1, but its authentication pattern does not match that of C0 or the CKDS, which are equal.
36060	The new master key register(s) is/are not full.
36064	Both new master key registers are full but not equal.
36068	The input KDS is not enciphered under the current master key.
36076	The new master key register for C0 is not full, but the CPUs are online.
36080	The new master key register for C1 is not full, but the CPUs are online.
36084	The master key register cannot be changed since ICSF is running in compatibility mode.
36104	Option not available. There were no Cryptographic Coprocessors available to perform the service that was attempted.
36108	PKA callable services are enabled, and the PKDS is the active PKDS as specified in the options data set.
36120	The CKDS is unusable. The CKDS does not support record level authentication.
36124	The CKDS is unusable. The CKDS only supports encrypted AES keys and encrypted DES support is required.
36128	The CKDS is unusable. The CKDS does not support encrypted DES keys which is required.
36160	The attempt to reencipher the CKDS failed because there is an enhanced token in the CKDS.
36001	A variable-length record format CKDS cannot be used on a system with a Cryptographic Coprocessor Feature.
36168	The LRECL attribute of the input CKDS doesn't match the LRECL of the output CKDS.

Return code 72 or 104 has these reason codes:

Reason Code	**Meaning**
6008	A service routine has failed. The service routines that may be called are: **CSFMGN** MAC generation

CSFMVR
MAC verification

CSFMKVR
Master key verification

6012	The single-record, read-write installation exit (CSFSRRW) returned a return code greater than 4.
6016	An I/O error occurred reading or writing the CKDS.
6020	The CSFSRRW installation exit abended and the installation options EXIT keyword specifies that the invoking service should end.
6024	The CSFSRRW installation exit abended and the installation options EXIT keyword specifies that ICSF should end.
6028	The CKDS access routine could not establish the ESTAE environment.
6040	The CSFSRRW installation exit could not be loaded and is required.
6044	Information necessary to set up CSFSRRW installation exit processing could not be obtained.
6048	The system keys cannot be found while attempting to write a complete CKDS data set.
6052	For a write CKDS record request, the current master key verification pattern (MKVP) does not match the CKDS header record MKVP.
6056	The output CKDS is not empty.
36001	A variable-length record format CKDS cannot be used on a system with a Cryptographic Coprocessor Feature.
36168	The LRECL attribute of the input CKDS doesn't match the LRECL of the output CKDS.

Note: It is possible that you will receive MVS reason codes rather than ICSF reason codes, for example, if the reason code indicates a dynamic allocation failure. For an explanation of Dynamic Allocation reason codes, see *z/OS MVS Programming: Authorized Assembler Services Guide*.

Chapter 3. Update of z/OS Cryptographic Services ICSF System Programmer's Guide, SA22-7520-15, information

This chapter contains updates to the document *z/OS Cryptographic Services ICSF System Programmer's Guide*, SA22-7520-15. 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.

Support is provided for the use of system symbols in the installation options data set. System symbols can be used as values for any of the parameters. System symbols are specified in the IEASYMxx member of SYS1.PARMLIB; the IEASYM statement of the LOADxx member of SYS1.PARMLIB specifies the IEASYMxx member(s) to be used for the resolution of system symbols. This example shows the use of a system symbol for specifying the domain to be used for the start of ICSF:

```
DOMAIN(&PARDOM.)
```

When the Installation Options Data Set is processed during the start of ICSF, the value of the system symbol PARDOM will be substituted as the value of the DOMAIN parameter.

For the first start, you specified an empty VSAM data set name for the CKDS in the CKDSN option, an empty VSAM data set name for the PKDS in the PKDSN option, and SSM(YES). You may want to change these and other options for subsequent starts. Here is a complete list of installation options:

BEGIN(fmid)

Specifies that keywords following this BEGIN keyword are supported in release *fmid* and later. There must be an END statement to complete the current section. If not, an error message will be issued and ICSF will terminate.

There may be any number of BEGIN/END pairs in the data set, but they can't be nested within each other. A BEGIN must have a matching END before another BEGIN can be specified.

If the release of ICSF you're running is at this release or later, the keywords will be parsed and processed. If release of ICSF you're running is an earlier release, the keywords will be ignored.

It recommended that when your systems are all running releases that support newer keywords that the BEGIN/END pair be removed.

The following FMIDs are supported: HCR7740, HCR7750, HCR7751, HCR7770, and HCR7780.

Here is an example of the usage of the BEGIN/END keywords.

```
keyword4       /* keyword4 is supported by all releases */
BEGIN(HCR7751)
keyword1       /* keyword1 added in HCR7751 */
keyword3       /* keyword3 added in HCR7751 */
END
BEGIN(HCR7770)
keyword2       /* keyword2 added in HCR7770 */
END
keyword5       /* keyword5 is supported by all releases */
```

CHECKAUTH(YES or NO)
Indicates whether ICSF performs security access control checking of Supervisor State and System Key callers. If you specify CHECKAUTH(YES), ICSF issues RACROUTE calls to perform the security access control checking and the results are logged in RACF SMF records that are cut by RACF. If you specify CHECKAUTH(NO), the authorization checks against resources in the CSFSERV class are not performed resulting in a significant performance enhancement for supervisor state and system key callers. However, the authorization checks are not logged in the RACF SMF records.

If you do not specify the CHECKAUTH option, the default is CHECKAUTH(NO).

If you configure CHECKAUTH(YES) in the ICSF options dataset, the Health Checker address space user identity must be authorized to the CSFRKL profile in class CSFSERV for the ICSFMIG7731_ICSF_RETAINED_RSAKEY migration check to successfully execute. However, you have no action to take if you choose not to run the migration check. If you configure CHECKAUTH(NO), there is no requirement to authorize the Health Checker user identity for any ICSF profiles or classes, since the check routine executes in supervisor state. This is not an implementation consideration, but rather a check deployment or activation time customer administration consideration.

CKDSN(data-set-name)
Specifies the CKDS name the system uses to start ICSF. Whenever you restart ICSF, the CKDS named in the CKDSN option becomes the active in-storage CKDS. (At first-time startup, you should specify the name of an empty VSAM data set you created to use as the CKDS.)

If you do not specify this keyword, ICSF does not become active. There is no default for this option, so you must specify a value.

CKTAUTH(YES or NO)
Decides if authentication will be performed for every CKDS record read from DASD.

| **Note:** If the active CKDS is a variable-length record format CKDS or a
| fixed-length record format CKDS that does not use record level
| authentication, the CKTAUTH option will be ignored. It will be
| displayed as DISABLED on the Installation Options Display panel.

YES	Indicates authentication will be performed.
NO	Indicates no authentication will be performed.

COMPAT(YES, NO, or COEXIST)
Indicates whether ICSF runs in compatibility mode, non-compatibility mode, or coexistence mode with PCF.

YES	Indicates **compatibility mode**.

In compatibility mode, you can run a PCF application on ICSF, because ICSF supports the PCF macros. You do not have to reassemble PCF applications to do this. You cannot start PCF at the same time as ICSF on the same operating system.

NO Indicates **non-compatibility mode**. In noncompatibility mode, you run PCF applications on PCF and ICSF applications on ICSF. You cannot run PCF applications on ICSF, because ICSF does not support the PCF macros in this mode. PCF can be started at the same time as ICSF on the same operating system. You can start ICSF and then start PCF, or you can start PCF and then start ICSF.

You should use noncompatibility mode unless you are migrating from PCF to ICSF.

COEXIST Indicates **coexistence mode**.

In coexistence mode, you can run a PCF application on PCF, or you can reassemble the PCF application to run on ICSF. To do this, you reassemble the application against coexistence macros that are shipped with ICSF. You can start PCF at the same time as ICSF on the same operating system.

If you do not specify the COMPAT option, the default value is COMPAT(NO).

When you initialize ICSF for the first time, noncompatibility mode must be active. Therefore, at first-time startup, you must specify COMPAT(NO) or allow the default to be used.

COMPENC(DES or CDMF)
This keyword is no longer supported but is tolerated.

DEFAULTWRAP(internal_wrapping_method,external_wrapping_method)
Specifies the default key wrapping for DES keys. Any token generated or updated by a service will be wrapped using the specified method unless overridden by rule array keyword or a skeleton token. The default wrapping method for internal and external tokens is specified independently.

Valid values for *internal_wrapping_method* and *external_wrapping_method* are:

ORIGINAL
Specifies the original CCA token wrapping be used: ECB wrapping for DES.

ENHANCED
Specifies the new X9.24 compliant CBC wrapping used. Note that the enhanced wrapping method is available only on the z196 with a CEX3C.

If the DEFAULTWRAP keyword is not specified, the default wrapping method will be ORIGINAL for both internal and external tokens.

DOMAIN(n)
Specifies the number of the domain that you want to use for this start of ICSF. You can specify only one domain in the options data set.

DOMAIN is an optional parameter. The DOMAIN parameter is only required if more than one domain is specified as the usage domain on the PR/SM panels or if running in native mode. If specified in the options data set, it will be used and it must be one of the usage domains for the LPAR.

If DOMAIN is not specified in the options data set, ICSF determines which domains are available in this LPAR. If only one domain is defined for the LPAR, ICSF will use it. If more than one is available, ICSF will issue error message CSFM409E.

The cryptographic processors support multiple sets of master key registers, which the specific domain values identify.

- The Cryptographic Coprocessor Feature has a master key register for the DES master key, the auxiliary DES master key, the signature master key and the key management master key. The auxiliary DES master key register may contain either the new or old DES master key. On the PCI Cryptographic Coprocessor, each domain has a master key register for the current, new, and old SYM-MK and RSA-MK.
- The PCIXCC/CEX2C has master key registers for the DES-MK, AES-MK and RSA-MK master keys. Each domain has a master key register for the current, new, and old DES-MK, AES-MK and RSA-MK.
- The CEX3C has master key registers for the DES-MK, AES-MK, RSA-MK, and ECC-MK master keys. Each domain has a master key for the current, new, and old DES-MK, AES-MK, RSA-MK, and ECC-MK.

For more information about partitions and running different configurations, see *z/OS Cryptographic Services ICSF Overview*.

If you run ICSF in compatibility or coexistence mode, you cannot change the domain number without re-IPLing the system. A re-IPL ensures that a program does not access a cryptographic service with a key that is encrypted under a different master key. If you are certain that no cryptographic applications are still running, you can:

1. Stop CSF
2. Start CSF in COMPAT(NO) mode with a different domain number
3. Stop CSF
4. Start CSF in compatibility or coexistence mode with a different domain number.

END Specifies the end of a section of keywords for the *fmid* from the **BEGIN(fmid)**. There must be a **BEGIN(fmid)** prior to the END. There must be an END for each **BEGIN(fmid)**. See the description for BEGIN for an example of the usage of the BEGIN and END keywords.

EXIT(ICSF-name,load-module-name,FAIL(fail-option))
Indicates information about an installation exit.

The *ICSF-name* is the identifier for each exit. Table 1 on page 11 lists all the ICSF exit names and explains when ICSF calls each exit. The load module name is the name of the module that contains the exit. The name can be any valid name your installation chooses.

Using the FAIL keyword of the EXIT statement, you specify the action ICSF, the KGUP, or the PCF conversion program takes if the exit ends abnormally. The fail action that you specify applies to subsequent calls of the exit. If an exit ends abnormally, ICSF takes a system dump. The exit is protected with an ESTAE or the ICSF service functional recovery routine (FRR).

In general, you can specify one of these values for a fail option:

NONE
No action is taken. The exit can be called again and will end abnormally again.

EXIT The exit is no longer available to be called again.

SERVICE
The service or program that called the exit is no longer available to be called again.

ICSF ICSF or the key generator utility program or the PCF conversion program is ended, depending on the exit.

Some fail options are not valid for a specific exit. If you specify a fail option that is not valid, ICSF uses the next valid fail option. For example, if SERVICE is not a valid fail option for an exit, ICSF uses the EXIT option. EXIT is responsible for logging in SMF the results of any authorization checks that are made.

Table 1. Exit Identifiers and Exit Invocations

Exit Identifiers	**Exit Invocations**
CSFEXIT1	Gets control after the operator issues the START command, but before processing takes place. **Note:** You must not specify an EXIT statement for the first mainline exit, CSFEXIT1.
CSFEXIT2	Gets control after ICSF reads and interprets the installation options data set.
CSFEXIT3	Gets control before ICSF completes initialization.
CSFEXIT4	Gets control after the operator issues the STOP command to stop ICSF.
CSFEXIT5	Gets control when the operator issues the MODIFY command to modify ICSF.
CSFEMK	Gets control during the compatibility service for the PCF EMK macro.
CSFGKC	Gets control during the compatibility service for the PCF GENKEY macro.
CSFRTC	Gets control during the compatibility service for the PCF RETKEY macro.
CSFEDC	Gets control during the compatibility service for the PCF CIPHER macro.
CSFCKDS	Gets control when a callable service retrieves an entry from the CKDS.
CSFKGUP	Gets control during key generator utility program initialization, processing, and termination.
CSFCONVX	Gets control when you run the PCF CKDS conversion program.
CSFSRRW	Gets control when an access to a single record in the CKDS is made using the key entry hardware.
CSFAEGN	Gets control during the ANSI X9.17 EDC generate callable service.
CSFAKEX	Gets control during the ANSI X9.17 key export callable service.
CSFAKIM	Gets control during the ANSI X9.17 key import callable service.
CSFAKTR	Gets control during the ANSI X9.17 key translate callable service.
CSFATKN	Gets control during the ANSI X9.17 transport key partial notarize callable service.
CSFCKI	Gets control during the clear key import callable service.
CSFCPE	Gets control during the clear PIN encrypt callable service.
CSFCPA	Gets control during the clear PIN generate alternate callable service.

Table 1. Exit Identifiers and Exit Invocations (continued)

Exit Identifiers	Exit Invocations
CSFCTT	Gets control during the ciphertext translate callable service.
CSFCTT1	Gets control during the ciphertext translate (with ALET) callable service.
CSFPGN	Gets control during the Clear PIN generate callable service.
CSFCVT	Gets control during the control vector translate callable service.
CSFCVE	Gets control during the cryptographic variable encipher callable service.
CSFDKX	Gets control during the data key export callable service.
CSFDKM	Gets control during the data key import callable service.
CSFDEC	Gets control during the decipher callable service.
CSFDEC1	Gets control during the decipher (with ALET) callable service.
CSFDCO	Gets control during the decode callable service.
CSFDSG	Gets control during the digital signature generate service.
CSFDSV	Gets control during the digital signature verify callable service.
CSFDKG	Gets control during the diversified key generate callable service.
CSFENC	Gets control during the encipher callable service.
CSFENC1	Gets control during the encipher (with ALET) callable service.
CSFECO	Gets control during the encode callable service.
CSFEPG	Gets control during the encrypted PIN generate callable service.
CSFPTR	Gets control during the encrypted PIN translate callable service.
CSFPVR	Gets control during the encrypted PIN verify callable service.
CSFKEX	Gets control during the key export callable service.
CSFKGN	Gets control during the key generate callable service.
CSFKGN2	Gets control during the key generate2 callable service.
CSFKIM	Gets control during the key import callable service.
CSFKPI	Gets control during the key part import callable service.
CSFKPI2	Gets control during the key part import2 callable service.
CSFKRC	Gets control during the key record create callable service.
CSFKRC2	Gets control during the key record create2 callable service.
CSFKRD	Gets control during the key record delete callable service.
CSFKRR	Gets control during the key record read callable service.
CSFKRR2	Gets control during the key record read2 callable service.
CSFKRW	Gets control during the key record write callable service.
CSFKRW2	Gets control during the key record write2 callable service.
CSFKYT	Gets control during the key test callable service.
CSFKYT2	Gets control during the key test2 callable service.
CSFKYTX	Gets control during the key test extended callable service.
CSFMDG	Gets control during the MDC generate callable service.
CSFKTR	Gets control during the key translate callable service.
CSFKTR2	Gets control during the key translate2 callable service.
CSFMGN1	Gets control during the MAC generate (with ALET) callable service.
CSFMVR	Gets control during the MAC verify callable service.

Table 1. Exit Identifiers and Exit Invocations (continued)

Exit Identifiers	Exit Invocations
CSFMVR1	Gets control during the MAC verify (with ALET) callable service.
CSFMDG1	Gets control during the MDC generate (with ALET) callable service.
CSFMGN	Gets control during the MAC generate callable service.
CSFCKM	Gets control during the multiple clear key import callable service.
CSFSKM	Gets control during the multiple secure key import callable service.
CSFOWH	Gets control during the one-way hash generate callable service.
CSFOWH1	Gets control during the one-way hash generate (with ALET) callable service.
CSFPCI	Gets control during the PCI interface callable service.
CSFPCU	Gets contol during the PIN Change/Unblock callable service
CSFPEX	Gets control during the prohibit export callable service.
CSFPEXX	Gets control during the prohibit export extended callable service.
CSFPKD	Gets control during the PKA decrypt callable service.
CSFPKE	Gets control during the PKA encrypt callable service.
CSFPKG	Gets control during the PKA key generate callable service.
CSFPKI	Gets control during the PKA key import callable service.
CSFPKT	Gets control during the PKA key translate callable service.
CSFPKTC	Gets control during the PKA key token change callable service.
CSFPKX	Gets control during the PKA Public Key Extract callable service.
CSFPKRC	Gets control during the PKDS record create callable service.
CSFPKRD	Gets control during the PKDS record delete callable service.
CSFPKRR	Gets control during the PKDS record read callable service.
CSFPKRW	Gets control during the PKDS record write callable service.
CSFPKSC	Gets control during the PKSC interface callable service.
CSFRKA	Gets control during the restrict key attribute callable service.
CSFRNG	Gets control during the random number generate callable service.
CSFRNGL	Gets control during the random number generate long callable service.
CSFRKD	Gets control during the retained key delete callable service.
CSFRKL	Gets control during the retained key list callable service.
CSFRKX	Gets control during the remote key export callable service.
CSFSKI	Gets control during the secure key import callable service.
CSFSKI2	Gets control during the secure key import2 callable service.
CSFSKY	Gets control during the secure messaging for keys callable service.
CSFSMG	Gets control during the symmetric MAC generate callable service.
CSFSMG1	Gets control during the symmetric MAC generate (with ALET) callable service.
CSFSMV	Gets control during the symmetric MAC verify callable service.
CSFSMV1	Gets control during the symmetric MAC verify (with ALET) callable service.
CSFSPN	Gets control during the secure messaging for PINs callable service.
CSFSBC	Gets control during the SET block compose callable service.
CSFSBD	Gets control during the SET block decompose callable service.
CSFSYX	Gets control during the symmetric key export callable service.

Table 1. Exit Identifiers and Exit Invocations (continued)

Exit Identifiers	Exit Invocations
CSFSYG	Gets control during the symmetric key generate callable service.
CSFSYI	Gets control during the symmetric key import callable service.
CSFSYI2	Gets control during the symmetric key import2 callable service.
CSFTBC	Gets control during the trusted block create callable service.
CSFTCK	Gets control during the transform CDMF key callable service.
CSFTRV	Gets control during the transaction validation callable service
CSFUDK	Gets control during the user derived key callable service.
CSFCSG	Gets control during the VISA CVV service generate callable service.
CSFCSV	Gets control during the VISA CVV service verify callable service.
CSFHMG	Gets control during the HMAC generate callable service.
CSFHMV	Gets control during the HMAC Verify callable service.

Note: z/OS no longer ships IBM-supplied security exit routines; the security exit points remain. Users of z/OS should use the Security Server (RACF) or an equivalent product to obtain access checking of services and keys. ICSF no longer needs these exit routines.

FIPSMODE(YES or COMPAT or NO,FAIL(fail-option))

Indicates whether z/OS PKCS #11 services must run in compliance with the Federal Information Processing Standard Security Requirements for Cryptographic Modules, referred to as FIPS 140-2. FIPS 140-2, published by the National Institute of Standards and Technology (NIST), is a standard that defines rules and restrictions for how cryptographic modules should protect sensitive or valuable information. The standard is available at http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

By configuring z/OS PKCS #11 services to operate in compliance with FIPS 140-2 specifications, installations or individual applications can use the z/OS PKCS #11 services in a way that allows only the cryptographic algorithms (including key sizes) approved by the standard, and restricts access to the algorithms that are not approved. For more information, refer to *z/OS Cryptographic Services ICSF Writing PKCS #11 Applications*.

YES Indicates that the z/OS PKCS #11 services will operate in *FIPS standard mode*. Any application using the PKCS #11 services will be forced to use those services in a FIPS-compliant manner. Applications will not have access to the algorithms or key sizes not approved by FIPS 140-2. In addition, ICSF initialization will test that it is running on an IBM System z model type, and a version and release of z/OS, that supports FIPS. If so, then ICSF will perform a series of cryptographic known answer tests as required by the FIPS 140-2 standard. If any of these initialization tests should fail, the action the ICSF initialization process takes will depend on the *fail-option* specified.

COMPAT

Indicates that the z/OS PKCS #11 services will operate in *FIPS compatibility mode*. This mode is intended for installations where only certain z/OS PKCS #11 applications must comply with the FIPS 140-2 standard, while other applications do not. In this mode, the PKCS #11 services can be further configured so that the applications that do not need to comply with the FIPS 140-2 standard are not restricted from using any of the PKCS #11 algorithms, while applications that must comply with the standard are restricted from using the non-approved algorithms. By default, the COMPAT option will have the same effect as the YES option, and all applications using the PKCS #11 services will be forced to use those services in a FIPS-compliant manner. However, additional specifications can be made:

- at the PKCS #11 token and application level, by creating FIPSEXEMPT.*token-name* resource profiles in the CRYPTOZ class. A FIPSEXEMPT.*token-name* resource exists for each token. User IDs with READ access authority to a FIPSEXEMPT.*token-name* are exempt from FIPS compliance, while user IDs with access authority NONE can only use the PKCS #11 services in a FIPS-compliant manner.
- within applications themselves for individual keys. When an application creates a key, the application can specify that the key must be used in a FIPS 140-2 compliant fashion. The application can specify this by setting the Boolean key attribute CKA_IBM_FIPS140 to TRUE.

When the COMPAT option is specified, ICSF initialization will test that it is running on an IBM System z model type, and a version and release of z/OS, that supports FIPS. If so, then ICSF will perform a series of cryptographic known answer tests as required by the FIPS 140-2 standard. If any of these initialization tests should fail, the action the ICSF initialization process takes will depend on the *fail-option* specified.

NO Indicates that no z/OS PKCS #11 applications at the installation need to comply with the FIPS 140-2 standard, and ICSF will bypass the extra processing that is required to ensure FIPS compliance. FIPSEXEMPT.*token-name* profiles, if they exist, will not be examined. Requests to generate or use a key with CKA_IBM_FIPS140=TRUE will result in a failure return code.

The *fail-option* is either YES or NO, and indicates what action the ICSF initialization process should take if any of the initialization tests (performed when **FIPSMODE** is **YES** or **COMPAT**) should fail.

YES indicates that ICSF should end abnormally if any of the tests fail.

NO Specifies that ICSF initialization process should continue even if one or more of the tests fail. However, z/OS PKCS #11 support will be limited or nonexistent depending on the test that failed.

- If ICSF is running on an IBM system z model type or with a version of z/OS that does not support FIPS, most FIPS processing is bypassed. PKCS #11 callable services will be available, but ICSF will not adhere to FIPS 140 restrictions. Requests to generate or use a key with CKA_IBM_FIPS140=TRUE will result in a failure return code.
- If a known answer test failed, all ICSF PKCS #11 callable services will be unavailable.

KEYAUTH(YES or NO)
Indicates whether or not ICSF authenticates a key entry after ICSF retrieves one from the in-storage CKDS. If you specify KEYAUTH(YES), ICSF authenticates the key. ICSF generates a message authentication code (MAC)

for each key entry in the CKDS when you create or update the entry. If you specify KEYAUTH(YES), ICSF performs a MAC verification to ensure that the entry has not changed. If you specify KEYAUTH(NO), ICSF does not perform this authentication and gains a small performance enhancement. If you do not specify the KEYAUTH option, the default value is KEYAUTH(NO).

| **Note:** If the active CKDS is a variable-length record format CKDS or a
| fixed-length record format CKDS that does not use record level
| authentication, the KEYAUTH option will be ignored. It will be
| displayed as DISABLED on the Installation Options Display panel.

MAXLEN(n)
Defines the maximum length of characters in a text string, including any necessary padding, for some callable service requests. For example, this option defines the maximum length of the text the encipher service encrypts for each call. Specify *n* as a decimal value from 1024 through 2147483647. If you do not specify the MAXLEN option, the default value is MAXLEN(65535).

The MAXLEN parameter may still be specified in the options data set, but only the maximum value limit will be enforced (2147483647). If a value greater than this is specified, an error will result and ICSF will not start.

Note: MAXLEN is no longer displayed on the Installation Option Display panel.

PKDSCACHE
This keyword is no longer supported but is tolerated.

PKDSN(data-set-name)
Specifies the PKDS name the system uses to start ICSF. Whenever you restart ICSF, the PKDS named in the PKDSN option becomes the active PKDS. (At first-time startup, you should specify the name of an empty VSAM data set you created to use as the PKDS.)

If you do not specify this keyword, ICSF does not become active. There is no default for this option, so you must specify a value.

REASONCODES(ICSF or TSS)
Specifies which set of reason codes are to be returned from callable services. If you do not specify the REASONCODES option, the default of REASONCODES(ICSF) is used. If you specify REASONCODES(TSS), TSS reason codes will be returned. If there is a 1-to-1 mapping, the codes will be converted.

If you specified REASONCODES(ICSF) and your service was processed on a PCICC, PCIXCC, CEX2C, or CEX3C, a TSS reason code may be returned if there is no 1–1 corresponding ICSF reason code.

SERVICE(service-number,load-module-name,FAIL(fail-option))
Indicates information about an installation-defined service.

ICSF allows an installation to define its own service similar to an ICSF callable service. The *service-number* specifies a number that identifies the service to ICSF. The valid service numbers are 1 through 32767, inclusive. This set of service numbers is valid for both installation-defined services and UDX services. The service number of an installation-defined service must not be the same as the service number of a UDX service. The *load-module-name* is the name of the module that contains the service. During ICSF startup, ICSF loads this module and binds it to the *service-number* you specified.

The *fail-option* is YES or NO, indicating the action ICSF should take if loading the service ends abnormally.

YES	Specifies that ICSF ends abnormally if your service cannot be loaded.
NO	Specifies that ICSF continues to start if your service cannot be loaded.

If the service itself ends abnormally, ICSF does not end, but takes a system dump instead. The ICSF service functional recovery routine (FRR) protects the service.

SSM(YES or NO)

Specifies whether or not an installation can enable special secure mode (SSM) while running ICSF. SSM lowers the security of your system to let you enter clear keys and generate clear PINs. You must enable SSM for KGUP to permit generation or entry of clear keys and to enable the secure key import or clear pin generate callable services.

YES	Indicates that you can enable the SSM.
NO	Indicates that you cannot enable the SSM.

If you do not specify the SSM option, the default value is SSM(NO).

Note: **CCF Systems only**: When you initialize ICSF for the first time, SSM must be active. Therefore, at first-time startup, you must specify SSM(YES).

If you are running with the IBM @server zSeries 900, IBM @server zSeries 800, S/390 Enterprise Servers and S/390 Multiprise servers, you must perform these tasks to make SSM active:
- Specify SSM(YES) in the installation options data set
- Enable SSM in the cryptographic hardware
- When running under a logical partition (LPAR), enable SSM for each partition.

SSM must be enabled or disabled in ALL places or errors may be logged and functions will not work as expected.

Note: The setting of the Environment Control Mask (ECM) enables SSM. Without TKE, the supplied ECM enables SSM. With TKE, you can set the ECM directly; the supplied ECM enables SSM, but you have the ability to disable it. For details, refer to *Support Element Operations Guide* and *z/OS Cryptographic Services ICSF TKE Workstation User's Guide*.

SYSPLEXCKDS(YES or NO,FAIL(*fail-option*))

SYSPLEXCKDS(YES,FAIL(*fail-option*))

ICSF will join the ICSF sysplex group SYSICSF and this system will participate in sysplex-wide consistency for CKDS data.

SYSPLEXCKDS(YES,FAIL(YES))

Indicates ICSF initialization will end abnormally if the ICSF cross-system services environment cannot be established during ICSF initialization due to a failure creating the CKDS latch set or a failure to join the ICSF sysplex group.

SYSPLEXCKDS(YES,FAIL(NO))
Indicates ICSF initialization processing will continue even if the request to create a CKDS latch set fails or the request to join the ICSF sysplex group fails. The system will not be notified of updates to the CKDS by other members of the ICSF sysplex group. A value of either FAIL(YES) or FAIL(NO) will be ignored with SYSPLEXCKDS(NO,...).

SYSPLEXCKDS(NO,FAIL(*fail-option*))
CKDS update processing proceeds as it does today (i.e. no Cross-System Services task will be initialized, nor will any XCF signalling be performed when an update to a CKDS record occurs).

If you do not specify the SYSPLEXCKDS option, the default value is SYSPEXCKDS(NO,FAIL(NO)).

SYSPLEXPKDS(YES or NO,FAIL(*fail-option*))
ICSF will join the ICSF sysplex group SYSICSF and this system will participate in sysplex-wide consistency for PKDS data.

SYSPLEXPKDS(YES,FAIL(*fail-option*))
ICSF will join the ICSF sysplex group SYSICSFP and this system will participate in sysplex-wide consistency for PKDS data.

SYSPLEXPKDS(YES,FAIL(YES))
Indicates ICSF initialization will fail to join the sysplex if the ICSF cross-system services environment cannot be established during ICSF initialization due to a failure creating the PKDS latch set or a failure to join the ICSF sysplex group.

SYSPLEXPKDS(YES,FAIL(NO))
Indicates ICSF initialization processing will continue even if the request to create a PKDS latch set fails or the request to join the ICSF sysplex group fails. The system will not be notified of updates to the PKDS by other members of the ICSF sysplex group. A value of either FAIL(YES) or FAIL(NO) will be ignored with SYSPLEXPKDS(NO,...).

SYSPLEXPKDS(NO,FAIL(*fail-option*))
PKDS update processing proceeds without trying to join the ICSF sysplex group.

If you do not specify the **SYSPLEXPKDS** option, the default value is **SYSPEXPKDS(NO,FAIL(NO))**.

SYSPLEXTKDS(YES or NO,FAIL(*fail-option*))

ICSF will join the ICSF sysplex group SYSICSF and this system will participate in sysplex-wide consistency for TKDS data.

Note: TKDSN needs to be specified for this to work. See on page 19.

SYSPLEXTKDS(NO,FAIL(*fail-option*))
Indicates no XCF signalling will be performed when an update to a TKDS record occurs.

SYSPLEXTKDS(YES,FAIL(*fail-option*))
Indicates the system will be notified of updates made to the TKDS by other members of the sysplex who have also specified SYSPLEXTKDS(YES,FAIL(*fail-option*)).

SYSPLEXTKDS(YES,FAIL(YES))
Indicates ICSF will terminate abnormally if there is a failure creating the TKDS latch set.

SYSPLEXTKDS(YES,FAIL(NO))
Indicates ICSF initialization processing will continue even if the request to create a TKDS latch set fails with an environment failure. This system will not be notified of updates to the TKDS by other members of the ICSF sysplex group.

If you do not specify the SYSPLEXTKDS option, the default value is SYSPLEXTKDS(NO,FAIL(NO)) is the default.

TKDSN(data-set-name)
The name of an existing TKDS or an empty VSAM data set to be used as the TKDS. To enable applications to create and use persistent PKCS #11 tokens and objects using the PKCS #11 services, this option must be specified.

TRACEENTRY(n)
Specifies the number, *n*, of trace buffers to allocate for ICSF tracing. Specify *n* as a decimal value from 10000 through 500000, inclusive. The default is 10000.

You should set this parameter to the maximum in case you ever need this trace material.

UDX(UDX-id,service-number,load-module-name,'comment_text',FAIL(fail-option))
ICSF allows the development of User Defined Extensions for the PCICC, PCIXCC, CEX2C, or CEX3C. The *UDX-id* is supplied to the installation when the UDX is developed. The *service-number* specifies a number that identifies the service to ICSF. The valid service numbers are 1 to 32767, inclusive. This set of service numbers is valid for both installation-defined services and UDX services. The service number of a UDX service must not be the same as the service number of an installation-defined service. The *load-module-name* is the name of the module that contains this service. During ICSF startup, ICSF loads this module and binds it to the service-number that was specified. A *comment* may be specified. The positional parameter is required. The comment consists of up to 40 EBCDIC characters, and may include imbedded blank characters. The comment text is enclosed by single quotes. If no comment text is desired, two contiguous single quotes should be specified.

The *fail-option* is YES or NO, indicating the action ICSF should take if loading the service ends abnormally. If the service itself ends abnormally, ICSF does not end, but takes a system dump instead.

YES Specifies that ICSF ends abnormally if your service cannot be loaded.

NO Specifies that ICSF continues to start if your service cannot be loaded.

The User Defined Extension (UDX) is responsible for logging in SMF the results of any authorization checks that were made.

USERPARM(value)
Specifies an 8-byte field for installation use. The Installation Option Display panel displays this value, which is stored in the Cryptographic Communication Vector Table (CCVT) in the *CCVT_USERPARM* field. An application program or installation exit can examine this field and use it to set system environment information. The default is eight blanks.

WAITLIST(data_set_name)
This optional parameter can be used if you have ICSF with CICS (CICS 4.1 or higher) installed. It specifies a customer modifiable data set will be used to determine names of the services to be placed into the ICSF CICS Wait List. A sample data set is provided by ICSF via member CSFWTL00 of SYS1.SAMPLIB with CCFs/PCICCs and CSFWTL01 for systems with PCIXCCs, CEX2Cs, or CEX3Cs. The sample data set contains the same entries as the default ICSF CICS Wait List (i.e., the data set contains the names of all ICSF callable services which, by default, will be driven through the CICS TRUE). The WAITLIST option should be added to the Installation Options data set under these conditions.

- Non-CICS customers will not specify a WAITLIST keyword. You must ensure, however, that if you have any existing CICS applications which invoke any of the ICSF services in the Wait List and if these applications were linked with ICSF stubs at a pre-OS/390 V2R10 level, then these applications must be re-linked with the current ICSF stubs.

  If running on a z990, z890, z9 EC or z9 BC however, you must also ensure that any existing CICS applications which invoke any of these services are re-linked to ensure that the correct version of the stub is used: CSNBCKI, CSNBCKM, CSNBDEC, CSNBENC, CSNBKYTX, CSNBMGN, CSNBMVR, CSNBPEXX, CSNBRNG
- CICS customers who do not want to make use of CICS TRUE must either not enable the TRUE or must specify a WAITLIST keyword and point to an empty wait list data set (or specify WAITLIST(DUMMY)) in the Installation Options data set.
- CICS customers who wish to modify the ICSF default CICS Wait List should modify the sample Wait List data set supplied in member CSFWTL00 or CSFWTL01 of SYS1.SAMPLIB. The WAITLIST keyword in the Installation Options Data Set should be set to point to this modified data set. If you have any existing CICS applications which invoke any of the ICSF services in the Wait List and if these applications were linked with ICSF stubs at a pre-OS/390 V2R10 level, then these applications must be re-linked with the current ICSF stubs.

  If running on a z990, z890, z9 EC or z9 BC any existing CICS applications which invoke any of these services are re-linked to ensure that the correct version of the stub is used: CSNBCKI, CSNBCKM, CSNBDEC, CSNBENC, CSNBKYTX, CSNBMGN, CSNBMVR, CSNBPEXX, CSNBRNG.

CICS Attachment Facility

If you have the CICS Attachment Facility (CICS 4.1 or higher) installed and you specify your own CICS wait list data set, you need to modify the wait list data set to include the new callable services.

On a z900 or z800 machine, modify and include:

- HCR7780: CSFKRC2, CSFKRW2
- HCR7770: CSF1DMK, CSF1DVK, CSF1GKP, CSF1GSK, CSF1PKS, CSF1PKV, CSF1SAV, CSF1SKE, CSF1TRC, CSF1TRD, CSF1UWK, CSF1WPK, CSFTBC, CSFRKX
- HCR7751, HCR7750, HCR7740: CSF1GKP, CSF1GSK, CSF1PKS, CSF1PKV, CSF1SAV, CSF1TRC, CSF1TRD, CSF1UWK, CSF1WPK, CSFTBC, CSFRKX
- HCR7731: CSFTBC, CSFRKX

Note: If no Wait List is specified on a z900 or z800, the default wait list will be used. (See sample CSFWTL00 for the contents of the default wait list for z900 or z800.)

On a z990 or z890 (or later) class machine, modify and include:

- HCR7780: CSFHMG, CSFHMG1, CSFHMV, CSFHMV1, CSFKGN2, CSFKPI2, CSFKTR2, CSFKYT2, CSFRKA, CSFSKI2, CSFSYI2, CSFKRC2, CSFKRW2
- HCR7770: CSNBSYD, CSNBSYD1, CSNBSYE, CSNBSYE1, CSFPKT, CSF1DMK, CSF1DVK, CSF1SKD, CSF1SKE, CSF1HMG, CSF1HMV, CSF1OWH, CSF1PRF, CSNBSAD, CSNBSAD1, CSNBSAE, CSNBSAE1, CSFRNGL, CSF1GKP, CSF1GSK, CSF1PKS, CSF1PKV, CSF1SAV, CSF1TRC, CSF1TRD, CSF1UWK, CSF1WPK, CSFTBC, CSFRKX
- HCR7751: CSNBSAD, CSNBSAD1, CSNBSAE, CSNBSAE1, CSFRNGL, CSF1GKP, CSF1GSK, CSF1PKS, CSF1PKV, CSF1SAV, CSF1TRC, CSF1TRD, CSF1UWK, CSF1WPK, CSFTBC, CSFRKX
- HCR7750: CSFRNGL, CSF1GKP, CSF1GSK, CSF1PKS, CSF1PKV, CSF1SAV, CSF1TRC, CSF1TRD, CSF1UWK, CSF1WPK, CSFTBC, CSFRKX
- HCR7740: CSF1GKP, CSF1GSK, CSF1PKS, CSF1PKV, CSF1SAV, CSF1TRC, CSF1TRD, CSF1UWK, CSF1WPK, CSFTBC, CSFRKX
- HCR7731: CSFTBC, CSFRKX

Note: If no Wait List is specified on a z990, z890, z9 EC, z9 BC, z10 EC and z10 BC the default wait list will be used. (See sample CSFWTL01 for the contents of the default wait list for z990, z890, z9 EC, z9 BC, z10 EC and z10 BC.)

Chapter 4. Update of z/OS Cryptographic Services ICSF Application Programmer's Guide, SA22-7522-14, information

This chapter contains updates to the document *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SA22-7522-14. Refer to this source document if background information is needed.

Key Test2 (CSNBKYT2 and CSNEKYT2)

Use this callable service to generate or verify a secure, cryptographic verification pattern for keys. The key to test can be in the clear or encrypted under the master key. Keywords in the rule array specify whether the callable service generates or verifies a verification pattern.

The callable service name for AMODE(64) invocation is CSNEKYT2.

Format

```
CALL CSNBKYT2(
            return_code,
            reason_code,
            exit_data_length,
            exit_data,
            rule_array_count,
            rule_array,
            key_identifier_length,
            key_identifier,
            key_encrypting_key_identifier_length,
            key_encrypting_key_identifier,
            reserved_length,
            reserved,
            verification_pattern_length,
            verification_pattern )
```

Parameters

return_code

Direction: Output	Type: Integer

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

reason_code

Direction: Output	Type: 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: Input/Output	Type: Integer

The length of the data that is passed to the installation exit. The length can be from X'00000000' to X'7FFFFFFF' (2 gigabytes). The data is identified in the *exit_data* parameter.

exit_data

Direction: Input/Output Type: String

The data that is passed to the installation exit.

rule_array_count

Direction: Input Type: Integer

The number of keywords you supplied in the *rule_array* parameter. The value must be 3.

rule_array

Direction: Input Type: String

The *rule_array* contains keywords that provide control information to the callable service. The keywords must be in contiguous storage with each of the keywords left-justified in its own 8-byte location and padded on the right with blanks.

Table 2. Keywords for Key Test2 Control Information

Keyword	Meaning
Token algorithm (Required)	
HMAC	Specifies the key token is an HMAC key token.
Process rule (One required)	
GENERATE	Generate a verification pattern for the specified key.
VERIFY	Verify that a verification pattern matches the specified key.
Verification pattern calculation algorithm (Optional)	
SHA2VP1	Specifies to use the SHA-256 based verification pattern calculation algorithm. This is the default.

key_identifier_length

Direction: Input Type: Integer

The length of the *key_identifier* in bytes. The maximum value is 725.

key_identifier

Direction: Input Type: String

The key for which to generate or verify the verification pattern. The parameter is a variable length string of an internal token or the 64-byte label of a key in the CKDS.

If an internal token was supplied and was encrypted under the old master key, the token will be returned encrypted under the current master key.

key_encrypting_key_identifier_length

Direction: Input	Type: Integer

The byte length of the *key_encrypting_key_identifier* parameter. The value must be zero.

key_encrypting_key_identifier

Direction: Input/Output	Type: String

This parameter is ignored.

reserved_length

Direction: Input	Type: Integer

The byte length of the reserved parameter. The value must be zero.

reserved

Direction: Input/Output	Type: String

This parameter is ignored.

verification_pattern_length

Direction: Input/Output	Type: Integer

The byte length of the *verification_pattern* parameter.

On input: For GENERATE, the length must be at least 8 bytes; For VERIFY, the length must be 8 bytes.

On output for GENERATE, the length of the verification pattern returned.

verification_pattern

Direction: Input/Output	Type: String

For GENERATE, the verification pattern generated for the key.

For VERIFY, the supplied verification pattern to be verified.

Restrictions

This callable service only supports HMAC keys.

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.

You can generate the verification pattern for a key when you generate the key. You can distribute the pattern with the key and it can be verified at the receiving node. In this way, users can ensure using the same key at the sending and receiving locations. You can generate and verify keys of any combination of key forms: clear, operational or external.

This table shows the access control points in the ICSF role that control the function of this service.

Table 3. Required access control points for Key Test2

Access control point
Key Test and Key Test2 **Note:** This access control point cannot be disabled. It is required for ICSF master key validation

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

Table 4. Key Test2 required hardware

Server	Required cryptographic hardware	Restrictions
IBM @server zSeries 800 IBM @server zSeries 900		This service is not supported.
IBM @server zSeries 990 IBM @server zSeries 890		This service is not supported.
IBM System z9 EC IBM System z9 BC		This service is not supported.
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor	This service is not supported.
	Crypto Express3 Coprocessor	HMAC keys not supported.
z196	Crypto Express3 Coprocessor	

Reason Codes for Return Code 8 (8)

A return code of 8 indicates that the call to the service was unsuccessful. The following reason code has been added:

Table 5. Reason Codes for Return Code 8 (8)

Reason Code Hex (Decimal)	Description
841 (2113)	A key token contains invalid payload. **User action:** Recreate the key token.

Chapter 5. Update of z/OS Cryptographic Services ICSF Messages, SA22-7523-14, information

This chapter contains updates to the document *z/OS Cryptographic Services ICSF Messages*, SA22-7523-14. Refer to this source document if background information is needed.

CSFMnnnn Messages (ICSF Address Space)

CSFMnnnn messages communicate Integrated Cryptographic Service Facility mainline task issues. The following CSFMnnnn Message has been added:

| **CSFM128E CRYPTOGRAPHIC KEY DATA SET, *dsname*, CANNOT BE USED ON THIS SYSTEM.**

| **Explanation:** The cryptographic key data set (CKDS) cannot be used on this system. There are several reasons for
| this occurring.

| • Some of the keys required by this system are missing from the CKDS. This can occur when a CKDS is initialized
| on a system that requires fewer system keys. For a z900 system, the CKDS must be initialized on a z900.
| • The CKDS was initialized on a system without cryptographic coprocessors, but the current system has
| cryptographic coprocessors.
| • The CKDS is a variable-length record format CKDS and can not be used on z900 systems.

| **System action:** ICSF terminates.

| **Operator response:** Contact your system programmer.

| **System programmer response:** Update the ICSF installation options data set with the correct CKDS and restart ICSF.

CSFGnnnn Messages (Key Generator Utility Processing)

CSFGnnnn messages are issued by the key generator utility program (KGUP). These messages are sent to the KGUP diagnostic data set (CSFDIAG). For information about defining the CSFDIAG data set, see *z/OS Cryptographic Services ICSF System Programmer's Guide*. The following CSFGnnnn Message has been added:

| **CSFG0876 ENTRY *label* IS AN HMAC KEY TOKEN. *verb* NOT PERFORMED.**

| **Explanation:** The entry with the key index label is an HMAC key which can not be deleted or updated by KGUP.

| **System action:** Processing for the UPDATE, DELETE or RENAME statement ends.

| **User response:** Correct the KGUP statement so that the label is not of a HMAC key.

| **CSFG0886 CKDS IS NOT USABLE.**

| **Explanation:** The cryptographic key data set specified can not be used by KGUP. The CKDS has variable-length
| records and the release of ICSF running doesn't support the variable-length record format CKDS.

| **System action:** Processing ends.

| **User response:** Specify a CKDS that is compatible with your system.