z/OS

Cryptographic Services Integrated Cryptographic Service Facility Support for secure key tokens with CSNBFLE and CSNBFLD APAR OA51102

Contents

Chapter 1. Overview

This document describes changes to the Integrated Cryptographic Service Facility (ICSF) product to allow encrypted key tokens that are not stored in the CKDS to be specified for the key identifier parameter of the CSNBFLE and CSNBFLD callable services.

These changes are available through the application of the PTF for APAR OA51102 and apply to FMID HCR77C0 and HCR77B1.

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

- *z/OS Cryptographic Services ICSF Overview*, SC14-7505-06
- *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-06
- *z/OS Cryptographic Services ICSF System Programmer's Guide*, SC14-7507-06
- *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-06

The technical changes made to the ICSF product by the application of the PTF for APAR OA51102 are indicated in this document by a vertical line to the left of the change.

Chapter 2. Update of z/OS Cryptographic Services ICSF Overview, SC14-7505-06, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Overview*, SC14-7505-06, for the new support for Field Level Encipher (CSNBFLE and CSNEFLE) and Field Level Decipher (CSNBFLD and CSNEFLD) provided by this APAR. Refer to this source document if background information is needed.

Protected-key CPACF

Protected-key CPACF provides both high performance and high security by taking advantage of the high speed of CPACF while utilizing encrypted keys. It does this by using CPACF wrapping keys to protect the key during CPACF processing instead of passing a clear key. These wrapping keys (one for Advanced Encryption Standard (AES) keys and one for Data Encryption Standard (DES) keys) are analogous to the coprocessor master keys and are visible only to licensed internal code (LIC) and never to operating system storage.

Five callable services support protected-key CPACF:

- CKDS Key Record Read2 (CSNBKRR2 and CSNEKRR2)
- Field Level Encipher (CSNBFLE and CSNEFLE)
- Field Level Decipher (CSNBFLD and CSNEFLD)
- Symmetric Key Encipher (CSNBSYE, CSNBSYE1, CSNESYE, CSNESYE1)
- Symmetric Key Decipher (CSNBSYD, CSNBSYD1, CSNESYD, CSNESYD1)

Field Level Encipher, Field Level Decipher, Symmetric Key Encipher, and Symmetric Key Decipher accept labels for the *key_identifier* parameter when the KEYIDENT keyword is provided in the *rule_array*. Before protected-key CPACF, this label was restricted to refer to a clear DATA key in the CKDS. With protected-key CPACF enabled, the label may now refer to an encrypted DATA key as well. Field Level Encipher and Field Level Decipher additionally support an encrypted DATA key token that does not reside in the CKDS for the *key_identifier* parameter.

CKDS Key Record Read2 with the PROTKEY rule returns the protected-key CPACF form of the CCA token to a caller with sufficient authority (either system key or supervisor state).

ICSF processes a secure key usable by a coprocessor (a CCA encrypted key token) into a secure key usable by CPACF (a CPACF-wrapped key). Each CPACF wrapped key is kept on hand after the first use so it can be used again for a subsequent encryption or decryption request.

To transform a CCA-encrypted key token into a CPACF-wrapped key, ICSF does the following:

1. Determines if the key has already been wrapped for use with CPACF. ICSF maintains a cache of CPACF-wrapped DATA keys by label. When a label is specified on a call to the Symmetric Key Encipher or Symmetric Key Decipher service or when a label or token is specified on a call to the Field Level Encipher or Field Level Decipher service, ICSF retrieves the key from the

in-storage copy of the CKDS or protected key token cache. If it is an encrypted DATA key, ICSF looks for a cached copy and uses it if one is present.

2. Determines if this key is a candidate for wrapping. If the key has not been wrapped for CPACF and cached, ICSF inspects a field in the covering CSFKEYS profile to check for permission. A CSFKEYS profile can contain an ICSF segment, which specifies rules for key use. The SYMCPACFWRAP field of the ICSF segment indicates whether ICSF can rewrap the encrypted key using the CPACF wrapping key. If there is no covering profile, or ICSF(SYMCPACFWRAP(NO)) is set, ICSF does not allow the operation. Additionally, for CKDS Key Record Read2 with the PROTKEY rule, the SYMCPACFRET field of the ICSF segment is checked to determine whether ICSF can return the protected-key CPACF form.
3. Requests the wrapping operation. ICSF builds a request to a Crypto Express3 Coprocessor (CEX3C) or later coprocessor. In the coprocessor, the encrypted DATA key is recovered from under the card master key. The clear form is presented back to the LIC layer, which wraps the clear key value under the corresponding CPACF wrapping key (either AES or DES) before returning the key to operating system storage. At no point during this operation is the clear key value visible in operating system storage.
4. Caches the returned CPACF-wrapped key for future use.

Figure 1 shows how ICSF transforms a CCA-encrypted key token into a CPACF-wrapped key.

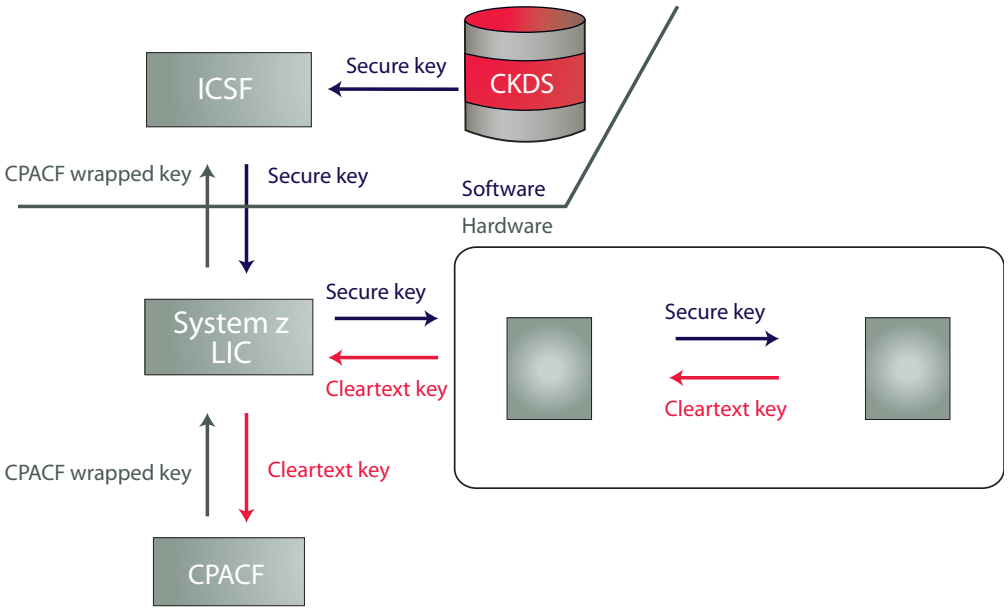

Figure 1. Transforming a CCA-encrypted key token into a CPACF-wrapped key

For more information about the Field Level Encipher and Field Level Decipher callable services, see *z/OS Cryptographic Services ICSF Application Programmer's Guide*.

For more information on the SYMCPACFWRAP and SYMCPACFRET fields of the ICSF segment of CSFKEYS profiles, see *z/OS Cryptographic Services ICSF Administrator's Guide*.

Chapter 3. Update of z/OS Cryptographic Services ICSF Administrator's Guide, SC14-7506-06, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-06, for the new support for Field Level Encipher (CSNBFLE and CSNEFLE) and Field Level Decipher (CSNBFLD and CSNEFLD) provided by this APAR. Refer to this source document if background information is needed.

Encrypted key support for Crypto Assist instructions

ICSF will exploit the performance of the CP Assist for Cryptographic Functions using encrypted AES and DES keys stored in the CKDS. Symmetric Key Encipher and Symmetric Key Decipher callable services will accept the label of an encrypted key as the key identifier. Field Level Encipher and Field Level Decipher will accept the label or token of an encrypted key as the key identifier. For more information about encryption using protected-key CPACF, see *z/OS Cryptographic Services ICSF Overview*.

Enabling use of encrypted keys in callable services that exploit CPACF

The Field Level Encipher, Field Level Decipher, Symmetric Key Encipher, and Symmetric Key Decipher callable services exploit CP Assist for Cryptographic Functions (CPACF) for improved performance.

The CKDS Key Record Read2 callable service can return the protected-key CPACF form of the CCA token to a caller with sufficient authority (either system key or supervisor state).

These services support encrypted AES and DES key tokens via the key label. This support requires the Crypto Express3 or later feature. The encrypted keys tokens must be stored in the CKDS and have a CSFKEYS profile with the ICSF segment.

A CSFKEYS profile can contain an ICSF segment, which specifies rules for key use. The SYMCPACFWRAP field of the ICSF segment enables you to specify whether ICSF can rewrap the encrypted key using the CPACF wrapping key. The specification:

- SYMCPACFWRAP(YES) indicates that encrypted keys covered by the profile can be rewrapped.
- SYMCPACFWRAP(NO), which is the default, indicates that encrypted keys covered by the profile cannot be rewrapped.

The Field Level Encipher and Field Level Decipher callable services exploit CP Assist for Cryptographic Functions (CPACF). These services support AES and DES clear key values and clear key tokens for the key identifier as well as AES and DES encrypted key tokens that are stored in the CKDS. These services have been enhanced to support encrypted key tokens that are not stored in the CKDS.

To use an encrypted key that does not reside in the CKDS, all of the following must be true:

1. The CSFKEYS class must be active and RACLISTed.

2. The ICSF segment of the CSFKEYS class general resource profile CSF-PROTECTED-KEY-TOKEN (or its generic equivalent) must contain SYMCPACFWRAP(YES).
3. The user associated with the application must have READ access to the profile.

Note: When key store policy is enabled, the default key label check is not enforced for encrypted key tokens. The CSF-PROTECTED-KEY-TOKEN profile is used in place of the CSF-CKDS-DEFAULT profile.

For all five services to utilize protected keys, SYMCPACFWRAP(YES) must be enabled in the covering profile.

For CKDS Key Record Read2, the SYMCPACFRET field of the ICSF segment enables you to specify whether ICSF can return the protected-key form of the CCA token to a caller. The specification:

- SYMCPACFRET(YES) indicates that keys covered by the profile can be returned to the caller in their protected-key CPACF form.
- SYMCPACFRET(NO), which is the default, indicates that keys covered by the profile cannot be returned to the caller in their protected-key CPACF form.

Rewrapping the encrypted key using the CPACF wrapping key is necessary in order to use an encrypted key as input to the Symmetric Key Encipher, Symmetric Key Decipher, Field Level Encipher, or Field Level Decipher callable services. You should be aware, however, that although the rewrapping operation ensures that no key is visible in application or system storage, the operation also requires the key to exist in the clear outside of the tamper-resistant hardware boundary. If your installation requires that a particular encrypted key must never exist outside of the tamper-resistant hardware boundary, do not use the SYMCPACFWRAP(YES) specification in a CSFKEYS profile that covers the key.

For example, say the CSFKEYS general resource profile DES.CHAOS.CAT covers an encrypted key stored in the CKDS that you would like to use as input to the Symmetric Key Encipher and Symmetric Key Decipher callable services. The following command modifies the SYMCPACFWRAP field of the profile's ICSF segment to allow this. The SETROPTS RACLIST command is used to refresh the CSFKEYS class in common storage.

```
RALTER CSFKEYS DES.CHAOS.CAT ICSF(SYMCPACFWRAP(YES))
SETROPTS RACLIST(CSFKEYS) REFRESH
```

The CSF-PROTECTED-KEY-TOKEN CSFKEYS profile can be defined for use with Field Level Encipher and Field Level Decipher using the following command:

```
RDEFINE CSFKEYS CSF-PROTECTED-KEY-TOKEN ICSF(SYMCPACFWRAP(YES))
UACC(NONE)
PERMIT CSF-PROTECTED-KEY-TOKEN ID(group-id) CLASS(CSFKEYS) ACCESS(READ)
SETR RACLIST(CSFKEYS) REFRESH
```

Callable services affected by key store policy

This table provides application programmers guidance on parameters covered by the key store policy controls.

Only the names of the 31-bit versions of the callable services are listed. However, 64-bit versions of the callable services and the ALET qualified versions of the services are also covered by the key store policy. The callable services that are affected by the TOKEN_CHECK key store policy controls are in the table below.

Table 1. Callable services and parameters affected by key store policy

ICSF callable service	31-bit name	Parameter checked
Field Level Decipher	CSNBFLD	key_identifier when the parameter contains an encrypted key token.
Field Level Encipher	CSNBFLE	key_identifier when the parameter contains an encrypted key token.

Chapter 4. Update of z/OS Cryptographic Services ICSF System Programmer's Guide, SC14-7507-06, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF System Programmer's Guide*, SC14-7507-06, for the new support for Field Level Encipher (CSNBFLE and CSNEFLE) and Field Level Decipher (CSNBFLD and CSNEFLD) provided by this APAR. Refer to this source document if background information is needed.

High Performance Encrypted Key (Subtype 28)

Symmetric Key Encipher (CSNBSYE, CSNBSYE1, CSNESYE and CSNESYE1), Symmetric Key Decipher (CSNBSYD, CSNBSYD1, CSNESYD and CSNESYD1), Field Level Encipher (CSNBFLE, CSNEFLE), and Field Level Decipher (CSNBFLD, CSNEFLD) callable services exploit CP Assist for Cryptographic Functions (CPACF) for improved key management performance. An encrypted DATA key stored in the CKDS can be used in these services, but only when SYMCPACFWRAP(YES) is specified in the ICSF segment of the CSFKEYS class profile that covers the key. For Field Level Encipher and Field Level Decipher, an encrypted DATA key that is not stored in the CKDS can be used, but only when SYMCPACFWRAP(YES) is specified in the ICSF segment of the CSF-PROTECTED-KEY-TOKEN CSFKEYS class profile. ICSF writes to subtype 28 at the completion of functions that attempt to wrap an encrypted key under the CPACF wrapping key. Subtype 28 will indicate if the rewrapping operation is:

- Permitted for this symmetric key
- Not permitted for this symmetric key

SMF records for this subtype will also contain server user and end user audit sections.

For more information about protected-key CPACF, see *z/OS Cryptographic Services ICSF Overview*.

Subtype 28

High performance encrypted key

Table 2. Subtype 28 High Performance Encrypted Key

Offsets		Name	Length	Format	Description
24	18	SMF82HPSK_FLAGS	4	binary	High Performance Encrypted Key flags: **Bit** **Meaning When Set** **0** Rewrapping operation is not permitted for this symmetric key. **1** Rewrapping operation was permitted for this symmetric key. **2** The list of labels is incomplete. **3** The key identifier was supplied as a key token, not as a label in the CKDS. Bits 4–31 are reserved.

Table 2. Subtype 28 High Performance Encrypted Key (continued)

Offsets		Name	Length	Format	Description
28	1C	SMF82HPSK_FUNCTION	8	EBCDIC	Name of the service that issues this SMF record. The name is in the form of CSFzzzz.
36	24	SMF82HPSK_SYM_LABEL_CNT	4	binary	Number of SYM labels present in this record.
The following is repeated SMF82HPSK_SYM_LABEL_CNT number of times.					
40	28	SMF82HPSK_SYM_LABELS	72	EBCDIC	SYM key label and type.

Chapter 5. Update of z/OS Cryptographic Services ICSF Application Programmer's Guide, SC14-7508-06, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-06, for the new support for Field Level Encipher (CSNBFLE and CSNEFLE) and Field Level Decipher (CSNBFLD and CSNEFLD) provided by this APAR. Refer to this source document if background information is needed.

Field Level Decipher (CSNBFLD and CSNEFLD)

Use the Field Level Decipher callable service to decrypt payment related database fields that have been previously encrypted using the field level encipher callable service. A database in this context is any structured data area or repository such as DB2, IMS, VSAM, or any column delineated data set or file.

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

Parameters

key_identifier

Direction	Type
Input/Output	String

For the KEY-CLR keyword, *key_identifier* specifies the cipher key. The parameter must be left justified.

For the KEYIDENT keyword, *key_identifier* specifies the internal clear or encrypted DES or AES DATA key token, or the label name of a clear or encrypted DES or AES DATA key token.

Note: To use a DES or AES encrypted DATA key in the CKDS, the ICSF segment of the CSFKEYS class general resource profile associated with the specified key label must contain SYMCPACFWRAP(YES). To use a DES or AES encrypted DATA key that does not reside in the CKDS, the ICSF segment of the CSFKEYS class general resource profile CSF-PROTECTED-KEY-TOKEN (or its generic equivalent) must contain SYMCPACFWRAP(YES). For more information, see *z/OS Cryptographic Services ICSF Administrator's Guide*.

If the token supplied was encrypted under the old master key, the token is returned encrypted under the current master key.

Usage notes

To use a CKDS encrypted key, the ICSF segment of the CSFKEYS class general resource profile associated with the specified key label must contain SYMCPACFWRAP(YES).

To use an encrypted key that does not reside in the CKDS, the ICSF segment of the CSFKEYS class general resource profile CSF-PROTECTED-KEY-TOKEN (or its generic equivalent) must contain SYMCPACFWRAP(YES) and the caller must have SAF access to the profile.

Field Level Encipher (CSNBFLE and CSNEFLE)

Use the Field Level Encipher callable service to encrypt payment related database fields, preserving the format of the fields. A database in this context is any structured data area or repository such as DB2, IMS, VSAM, or any column delineated data set or file. For example, you can encrypt a 16-digit EBCDIC credit card number where the resulting cipher text would also be 16 EBCDIC digits.

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

Parameters

key_identifier

Direction	Type
Input/Output	String

For the KEY-CLR keyword, *key_identifier* specifies the cipher key. The parameter must be left justified.

For the KEYIDENT keyword, *key_identifier* specifies the internal clear or encrypted DES or AES DATA key token, or the label name of a clear or encrypted DES or AES DATA key token.

Note: To use a DES or AES encrypted DATA key in the CKDS, the ICSF segment of the CSFKEYS class general resource profile associated with the specified key label must contain SYMCPACFWRAP(YES). To use a DES or AES encrypted DATA key that does not reside in the CKDS, the ICSF segment of the CSFKEYS class general resource profile CSF-PROTECTED-KEY-TOKEN (or its generic equivalent) must contain SYMCPACFWRAP(YES). For more information, see *z/OS Cryptographic Services ICSF Administrator's Guide*.

If the token supplied was encrypted under the old master key, the token is returned encrypted under the current master key.

Usage notes

To use a CKDS encrypted key, the ICSF segment of the CSFKEYS class general resource profile associated with the specified key label must contain SYMCPACFWRAP(YES).

To use an encrypted key that does not reside in the CKDS, the ICSF segment of the CSFKEYS class general resource profile CSF-PROTECTED-KEY-TOKEN (or its generic equivalent) must contain SYMCPACFWRAP(YES) and the caller must have SAF access to the profile.

Reason codes for return code 8 (8)

Table 3 lists reason codes returned from callable services that give return code 8.

Table 3. Reason codes for return code 8 (8)

Reason Code Hex (Decimal)	Description
BFB (3067)	The provided *key_identifier* refers to an encrypted CCA key token or a key label of an encrypted CCA key token, and the CSFKEYS profile covering it does not allow its use in high performance encrypted key operations. **User action:** Contact your ICSF or RACF administrator if you need to use this key with an ICSF service that supports secure keys for CPACF.
BE6 (3046)	A key token was passed to a service using high performance encrypted key operations and RACF failed your request to use the key token. **User action**: Contact your ICSF or RACF administrator if you need to pass key tokens to a service using high performance encrypted key operations.

Printed in USA