z/OS

Cryptographic Services
Integrated Cryptographic Service Facility
Protected Key Read Support
APAR OA50450

Contents

Chapter 1. Overview

This document describes changes to the Integrated Cryptographic Service Facility (ICSF) product in support for Protected Key Read, which allows authorized callers to retrieve the protected-key CPACF form of an AES or DES CCA token by label.

These changes are available through the application of the PTF for APAR OA50450 and apply to FMID HCR77B1, HCR77B0, HCR77A1, and HCR77A0.

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

- *z/OS Cryptographic Services ICSF Overview*, SC14-7505-05
- *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-05
- *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-05

The technical changes made to the ICSF product by the application of the PTF for APAR OA50450 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, SC14-7505-05, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Overview*, SC14-7505-05, for the Protected Key Read support 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.

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 the protected-key CPACF form is needed, ICSF retrieves the key from the in-storage copy of the CKDS. 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.

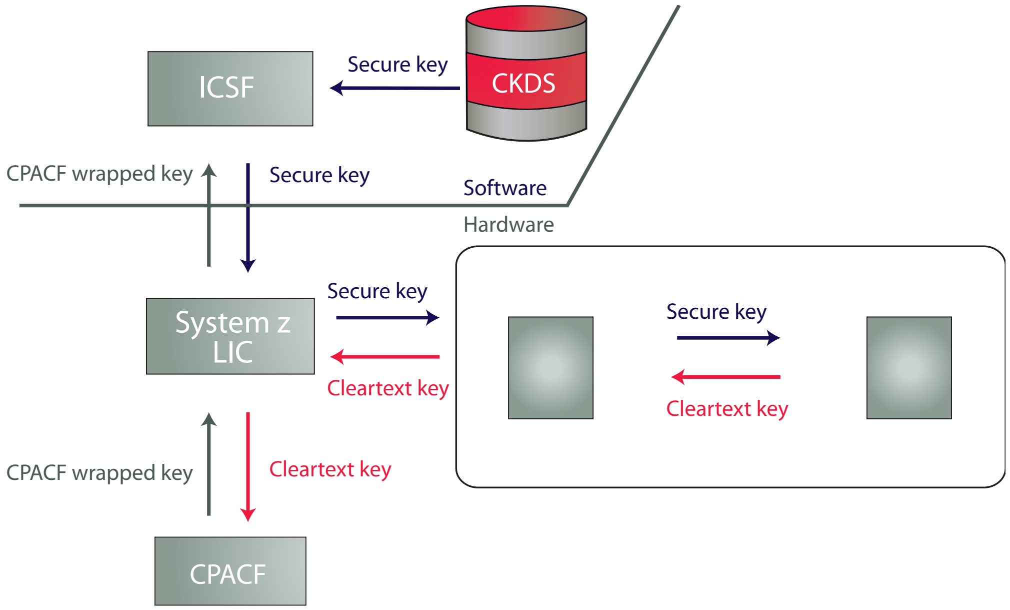

Figure 1. Transforming a CCA-encrypted key token into a CPACF-wrapped key

For more information about the 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, SC14-7506-05, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-05, for the Protected Key Read support provided by this APAR. Refer to this source document if background information is needed.

Enabling use of encrypted keys in callable services

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.

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 or Symmetric Key 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
```

Chapter 4. Update of z/OS Cryptographic Services ICSF Application Programmer's Guide, SC14-7508-05, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-05, for the Protected Key Read support provided by this APAR. Refer to this source document if background information is needed.

CKDS Key Record Read2 (CSNBKRR2 and CSNEKRR2)

Use this callable service to:

- Copy an internal fixed-length or variable-length AES, DES, or HMAC key token from the in-storage CKDS to application storage. Other cryptographic services can then use the copied key token directly.
- | Return the protected-key CPACF form of a fixed-length AES or DES key token.

The callable service name for AMODE(64) is CSNEKRR2.

Format

```
CALL CSNBKRR2(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             rule_array_count,
             rule_array,
             key_label,
             key_token_length,
             key_token )
```

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.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The value must be 0 or 2, inclusive.

rule_array

Direction	Type
Input	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 1. Keywords for CKDS Key Record Read2

Keyword	Meaning
Administrative rules (optional)	
ADMNREAD	The record is being read for administrative purposes. Reference date processing is bypassed. For more information, see the KDSREFDAYS option in *z/OS Cryptographic Services ICSF Administrator's Guide*. This keyword requires the application of the PTF for APAR OA49503 at ICSF FMID HCR77B1 and lower.
PROTKEY	The protected-key CPACF form of the secure key token is to be returned. This keyword requires the application of the PTF for APAR OA50450 at ICSF FMID HCR77B1 and lower.
Security checking rules (optional)	
BYPAUTH	ICSF will not perform an authorization check against CSFKEYS for the label because an authorized caller has already performed the relevant check. This rule is only allowed with PROTKEY. This keyword requires the application of the PTF for APAR OA50450 at ICSF FMID HCR77B1 and lower.

key_label

Direction	Type
Direction	String

The 64-byte label of a record in the CKDS to be retrieved.

key_token_length

Direction	Type
Input/Output	Integer

The length of the buffer for the output token or protected-key CPACF form of the token. On input, the length of the buffer. The minimum length is 8 bytes and the maximum length is 725 bytes. The length specified must be sufficient to contain the data returned. On output, this parameter will be updated with the length of the token returned in the *key_token* parameter.

key_token

Direction	Type
Output	String

The buffer into which the return key token or protected-key CPACF form of the token is written.

Usage Notes

- To retrieve a DES or AES encrypted DATA key from the CKDS in the protected-key CPACF form, the ICSF segment of the CSFKEYS class general resource profile associated with the specified key label must contain SYMCPACFWRAP(YES) and SYMCPACFRET(YES) and the invoker must be an authorized caller (either system key or supervisor state). For more information, see *z/OS Cryptographic Services ICSF Administrator's Guide*.
- The master keys must be loaded when using this service with the PROTKEY rule and specifying the label of an encrypted CCA key token.

Access control points

When the PROTKEY rule is specified with the label of an encrypted CCA key token, the appropriate access control point listed below must be enabled.

Table 2. Required access control points for CKDS Key Record Read2

Key algorithm	Access control point
AES	Symmetric Key Encipher/Decipher - Encrypted AES keys.
DES	Symmetric Key Encipher/Decipher - Encrypted DES keys.

Required hardware

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

Table 3. CKDS Key Record Read2 required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	None.	The PROTKEY rule is not supported.
IBM System z9 EC IBM System z9 BC	None.	The PROTKEY rule is not supported.

Table 3. CKDS Key Record Read2 required hardware (continued)

Server	Required cryptographic hardware	Restrictions
IBM System z10 EC IBM System z10 BC	CP Assist for Cryptographic Functions Crypto Express3 Coprocessor	When using the PROTKEY rule, encrypted keys require the CEX3C with the Nov. 2009 or later licensed internal code (LIC).
IBM zEnterprise 196 IBM zEnterprise 114	CP Assist for Cryptographic Functions Crypto Express3 Coprocessor	When using the PROTKEY rule, encrypted keys require the CEX3C with the Nov. 2009 or later licensed internal code (LIC).
IBM zEnterprise EC12 IBM zEnterprise BC12	CP Assist for Cryptographic Functions Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	
IBM z13 IBM z13s	CP Assist for Cryptographic Functions Crypto Express5 CCA Coprocessor	

Reason codes for return code 8 (8)

Table 4 lists reason codes returned from callable services that give return code 8.

Table 4. Reason codes for return code 8 (8)

Reason Code Hex (Decimal)	Description
C04 (3076)	The provided symmetric key label refers to an encrypted CCA key token, and the CSFKEYS profile covering it does not allow it to be returned in its protected-key CPACF form. **User Action:** Contact your ICSF or RACF administrator if you need to use this label in calls to the CKDS Key Record Read2 service with the PROTKEY rule.

Access control points and callable services

The following tables list 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 table lists access control points that affect multiple services or require special consideration when enabling the access control point. The Offset is the hexadecimal offset, or access-control-point code, for the control in the domain role in the coprocessor.

Table 5. Access control points affecting multiple services or requiring special consideration

Name	Callable services	Notes	Offset (hex)	Usage
High-performance secure AES keys	CSNBSYD / CSFNESYD, CSNBSYD1 / CSNESYD1, CSNBSYE / CSNESYE, CSNBSYE1 / CSNESYE1, CSNBFLD / CSNEFLD, CSNBFLE / CSNEFLE, and CSNBKRR2 / CSNEKRR2	When enabled, encrypted AES DATA key tokens in the CKDS can be used for the CPACF instructions.	0296	ED
High-performance secure DES keys	CSNBSYD / CSFNESYD, CSNBSYD1 / CSNESYD1, CSNBSYE / CSNESYE, CSNBSYE1 / CSNESYE1, CSNBFLD / CSNEFLD, CSNBFLE / CSNEFLE and CSNBKRR2 / CSNEKRR2	When enabled, encrypted DES DATA key tokens in the CKDS can be used for the CPACF instructions.	0295	ED

Printed in USA