z/OS Cryptographic Services
Integrated Cryptographic Service Facility

Support for AMEX CSC Algorithms – APAR OA38626

(October 24, 2012)

Contents

Chapter 1. Overview 1

Chapter 2. Update of z/OS Cryptographic Services ICSF Application Programmer's Guide, SA22-7522-15, information. . . . 3

PIN Change/Unblock (CSNBPCU and CSNEPCU) . 3

Format 4

Parameters 4

Usage Notes 8

Transaction Validation (CSNBTRV and CSNETRV) . . 9

Format 10

Parameters 10

Usage Notes 12

Chapter 1. Overview

Applying the PTF for APAR OA38626 provides support for a new CSC algorithm (CSC Version 2.0) defined by American Express. Specifically, this new algorithm is supported by the Transaction Validation (CSNBTRV and CSNETRV) callable service. Support is also being added for the AMEX output PIN creation format in the PCU verb. This support is being added to the PIN Change/Unblock (CSNBPCU and CSNEPCU) callable service. New rule array keywords are provided to enable applications to use either CSC algorithms.

This document contains alterations to information previously presented in *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SA22-7522-15.

The technical changes made to the ICSF product by the application of the PTF for APAR OA38626 are indicated in this document by a vertical line to the left of the change.

Chapter 2. Update of z/OS Cryptographic Services ICSF Application Programmer's Guide, SA22-7522-15, information

This chapter contains updates to the document *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SA22-7522-15. Refer to this source document if background information is needed.

PIN Change/Unblock (CSNBPCU and CSNEPCU)

The PIN Change/Unblock callable service is used to generate a special PIN block to change the PIN accepted by an integrated circuit card (smartcard). The special PIN block is based on the new PIN and the card-specific diversified key and, optionally, on the current PIN of the smartcard. The new PIN block is encrypted with a session key. The session key is derived in a two-step process. First, the card-specific diversified key (ICC Master Key) is derived using the TDES-ENC algorithm of the diversified key generation callable service. The session key is then generated according to the rule array algorithm:

- TDES-XOR - XOR ICC Master Key with the Application Transaction Counter (ATC)
- TDESEMV2 - use the EMV2000 algorithm with a branch factor of 2
- TDESEMV4 - use the EMV2000 algorithm with a branch factor of 4

The generating DKYGENKY key cannot have replicated halves. The *encryption_issuer_master_key_identifier* is a DKYGENKY key that permits generation of a SMPIN key. The *authentication_ issuer_master_key_identifier* is also a DKYGENKY key that permits generation of a double length MAC key.

The PIN block format is specified a keyword. The keyword specified refers to whether the current PIN is used in the generation of the new PIN. For PINs for the VISA ICC Card specification:

- VISAPCU1 would create a new PIN for a card without a PIN in an encrypted PIN-block in the *new_reference_PIN_block* variable. The contents of the five *input current_reference_PIN_x* variables are ignored.
- VISAPCU2 would provide the existing PIN for a card with a current PIN in an encrypted PIN-block in the *current_reference_PIN_block* variable, and supply the new PIN-value in an encrypted PIN-block in the *new_reference_PIN_block* variable.

For PINs for the American Express Hardware Security Module Function Requirements dated August 2011:

- AMEXPCU1 would create the output PIN from the new-reference PIN, the smart card-unique, intermediate key, and the current-reference PIN.
- AMEXPCU2 would create the output PIN from the new-reference PIN and the smart-card-unique, intermediate key.

An enhanced PIN security mode is available for extracting PINs from encrypted PIN blocks. This mode only applies when specifying a PIN-extraction method for an IBM 3621 or an IBM 3624 PIN-block. To do this, you must enable the PTR Enhanced PIN Security access control point in the default role. When activated, this mode limits checking of the PIN to decimal digits and a PIN length minimum of 4 is enforced. No other PIN-block consistency checking will occur.

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

Format

```
CALL CSNBPCU(
            return_code,
            reason_code,
            exit_data_length,
            exit_data,
            rule_array_count,
            rule_array,
            authentication_issuer_master_key_length,
            authentication_issuer_master_key_identifier,
            encryption_issuer_master_key_length,
            encryption_issuer_master_key_identifier,
            key_generation_data_length,
            key_generation_data,
            new_reference_PIN_key_length,
            new_reference_PIN_key_identifier,
            new_reference_PIN_block,
            new_reference_PIN_profile,
            new_reference_PIN_PAN_data,
            current_reference_PIN_key_length,
            current_reference_PIN_key_identifier,
            current_reference_PIN_block,
            current_reference_PIN_profile,
            current_reference_PIN_PAN_data,
            output_PIN_data_length,
            output_PIN_data,
            output_PIN_profile,
            output_PIN_message_length,
            output_PIN_message )
```

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 assigned to it that indicates 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 are supplying in the *rule_array* parameter. The valid values are 1 and 2.

rule_array

Direction: Input Type: String

Keywords that provides control information to the callable service. The keywords are left-justified in an 8-byte field and padded on the right with blanks. The keywords must be in contiguous storage. Specify one or two of these options:

Table 1. Rule Array Keywords for PIN Change/Unblock

Keyword	Meaning
Algorithm (optional)	
TDES-XOR	TDES encipher clear data to generate the intermediate (card-unique) key, followed by XOR of the final 2 bytes of each key with the ATC counter. This is the default.
TDESEMV2	Same processing as in the diversified key generate service.
TDESEMV4	Same processing as in the diversified key generate service.
PIN processing method (required)	
VISAPCU1	Form the new PIN from the new reference PIN and the smart-card-unique, intermediate key.
VISAPCU2	Form the new PIN from the new reference PIN and the smart-card-unique, the intermediate (card-unique) key and the current reference PIN.
AMEXPCU1	Form the new PIN from the new reference PIN, the smart-card-unique, intermediate key, and the current reference PIN.
AMEXPCU2	Form the new PIN from the new reference PIN and the smart-card-unique, intermediate key.

authentication_issuer_master_key_length

Direction: Input Type: Integer

The length of the *authentication_issuer_master_key_identifier* parameter. The value must be 64.

authentication_issuer_master_key_identifier

Direction: Input/Output Type: String

The label name or internal token of a DKYGENKY key type that is to be used to generate the card-unique diversified key. The control vector of this key must be a DKYL0 key that permits the generation of a double-length MAC key (DMAC). This DKYGENKY may not have replicated key halves.

encryption_issuer_master_key_length

Direction: Input Type: Integer

The length of the *encryption_issuer_master_key_identifier* parameter. The value must be 64.

encryption_issuer_master_key_identifier

Direction: Input/Output Type: String

The label name or internal token of a DKYGENKY key type that is to be used to generate the card-unique diversified key and the secure messaging session key for the protection of the output PIN block. The control vector of this key must be a DKYL0 key that permits the generation of a DMPIN key type. This DKYGENKY may not have replicated key halves.

key_generation_data_length

Direction: Input Type: Integer

The length of the *key_generation_data* parameter. This value must be 10, 18, 26 or 34 bytes.

key_generation_data

Direction: Input Type: String

The data provided to generate the card-unique session key. For TDES-XOR, this consists of 8 or 16 bytes of data to be processed by TDES to generate the card-unique diversified key followed by a 16 bit ATC counter to offset the card-unique diversified key to form the session key. For TDESEMV2 and TDESEMV4, this may be 10, 18, 26 or 34 bytes.

new_reference_PIN_key_length

Direction: Input Type: Integer

The length of the *new_reference_PIN_key_identifier* parameter. The value must be 64.

new_reference_PIN_key_identifier

Direction: Input/Output Type: String

The label name or internal token of a PIN encrypting key that is to be used to decrypt the *new_reference_PIN_block*. This must be an IPINENC or OPINENC key. If the label name is supplied, the name must be unique in the CKDS.

new_reference_PIN_block

Direction: Input Type: String

This is an 8-byte field that contains the enciphered PIN block of the new PIN.

new_reference_PIN_profile

Direction: Input Type: String

This is a 24-byte field that contains three 8-byte elements with a PIN block format keyword, a format control keyword (NONE) and a pad digit as required by certain formats.

new_reference_PIN_PAN_data

Direction: Input Type: String

This is a 12-byte field containing PAN in character format. This data may be needed to recover the new reference PIN if the format is ISO-0 or VISA-4. If neither is used, this parameter may be blanks.

current_reference_PIN_key_length

Direction: Input Type: Integer

The length of the *current_reference_PIN_key_identifier* parameter. The value must be 64. If the *rule_array* contains VISAPCU1 or AMEXPCU2, this value must be 0.

current_reference_PIN_key_identifier

Direction: Input/Output Type: String

The label name or internal token of a PIN encrypting key that is to be used to decrypt the *current_reference_PIN_block*. This must be an IPINENC or OPINENC key. If the labelname is supplied, the name must be unique on the CKDS. If the *rule_array* contains VISAPCU1 or AMEXPCU2, this value is ignored.

current_reference_PIN_block

Direction: Input Type: String

This is an 8-byte field that contains the enciphered PIN block of the new PIN. If the *rule_array* contains VISAPCU1 or AMEXPCU2, this value is ignored.

current_reference_PIN_profile

Direction: Input Type: String

This is a 24-byte field that contains three 8-byte elements with a PIN block format keyword, a format control keyword (NONE) and a pad digit as required by certain formats. If the *rule_array* contains VISAPCU1 or AMEXPCU2, this value is ignored.

current_reference_PIN_PAN_data

Direction: Input Type: String

This is a 12-byte field containing PAN in character format. This data may be needed to recover the new reference PIN if the format is ISO-0 or VISA-4. If neither is used, this parameter may be blanks. If the *rule_array* contains VISAPCU1 or AMEXPCU2, this value is ignored.

output_PIN_data_length

Direction: Input Type: Integer

The value of this parameter should be 0.

output_PIN_data

Direction: Input Type: String

This field is reserved.

output_PIN_profile

Direction: Input Type: String

This is a 24-byte field that contains three 8-byte elements with a PIN block format keyword (VISAPCU1, VISAPCU2, AMEXPCU1, or AMEXPCU2), a format control keyword, NONE, (left aligned and padded on the right with space characters) and 8 byte spaces.

output_PIN_message_length

Direction: Input/Output Type: Integer

The length of the *output_PIN_message* field. PIN block format keywords VISAPCU1 and VISAPCU2 require a byte length of 16. PIN block format keywords AMEXPCU1 and AMEXPCU2 require a byte length of 8.

output_PIN_message

Direction: Output Type: String

The reformatted PIN block with the new reference PIN enciphered under the SMPIN session key.

Usage Notes

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

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

Table 2. Required access control points for PIN Change/Unblock

PIN-block encrypting key-type	Access control point
OPINENC	PIN Change/Unblock - change EMV PIN with OPINENC
IPINENC	PIN Change/Unblock - change EMV PIN with IPINENC

When the *authentication_key_identifier* or *encryption_key_identifier* is specified with control vector bits (19 – 22) of B'1111', the **Diversified Key Generate - DKYGENKY – DALL** access control point must also be enabled.

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

Table 3. PIN Change/Unblock hardware

Server	Required cryptographic hardware	Restrictions
IBM @server zSeries 900		Not supported
IBM @server zSeries 990 IBM @server zSeries 890	PCI X Cryptographic Coprocessor Crypto Express2 Coprocessor	ISO-3 PIN block format is not supported. The AMEXPCU1 and AMEXPCU2 keywords are not supported.
IBM System z9 EC and z9 BC	Crypto Express2 Coprocessor	ISO-3 PIN block format requires the Nov. 2007 or later licensed internal code (LIC). For keywords AMEXPCU1 and AMEXPCU2, see the note below.
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	ISO-3 PIN block format requires the Nov. 2007 or later licensed internal code (LIC). For keywords AMEXPCU1 and AMEXPCU2, see the note below.
z196	Crypto Express3 Coprocessor	For keywords AMEXPCU1 and AMEXPCU2, see the note below.

Note: The AMEXPCU1 and AMEXPCU2 keywords require the following MCLs:

- IBM System z9 EC and z9 BC - Driver 67L, EC G40492, MCL G40942.028 -> 030 Bundle 70 to be available November 2012.
- IBM System z10 EC and IBM System z10 BC Crypto Express2 - Driver 79F, EC N24393, MCL N24393.017 -> 019 Bundle 65a Available 9/26/12
- IBM System z10 EC and IBM System z10 BC Crypto Express3 - Driver 79F, EC N24380, MCL N24380.047 -> 049 Bundle 65a Available 9/26/12
- IBM System z196 and z114 Crypto Express3 - Driver 93G, EC N48132, MCL N48132.007 -> 009 Bundle 37a Available 10/3/12

Transaction Validation (CSNBTRV and CSNETRV)

The transaction validation callable service supports the generation and validation of American Express card security codes (CSC). This service generates and verifies transaction values based on information from the transaction and a cryptographic key. Both versions 1.0 and 2.0 as defined in the American Express Hardware Security Module Function Requirements dated August 2011 are supported. You select the algorithm, validation method, and either the generate or verify mode, through rule-array keywords.

For the American Express process, the control vector supplied with the cryptographic key must indicate a MAC or MACVER class key. The key may be single or double length. DATAM and DATAMV keys are not supported. The MAC generate control vector bit must be on (bit 20) if you request CSC generation and MAC verify bit (bit 21) must be on if you request verification.

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

Format

```
CALL CSNBTRV(
            return_code,
            reason_code,
            exit_data_length,
            exit_data,
            rule_array_count,
            rule_array,
            transaction_key_identifier_length,
            transaction_key_identifier,
            transaction_info_length,
            transaction_info,
            validation_values_length,
            validation_values )
```

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 assigned to it that indicates 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 are supplying in the *rule_array* parameter. The valid values are 1, 2, or 3.

rule_array

Direction: Input   Type: Character String

Keywords that provides control information to the callable service. The keywords are left-justified in an 8-byte field and padded on the right with blanks. The keywords must be in contiguous storage. Specify one or two of the values inTable 4.

Table 4. Rule Array Keywords for Transaction Validation

Keyword	**Meaning**
American Express card security codes (required)	
CSC-3	3-digit card security code (CSC) located on the signature panel. **VERIFY** implied.
CSC-4	4-digit card security code (CSC) located on the signature panel. **VERIFY** implied.
CSC-5	5-digit card security code (CSC) located on the signature panel. **VERIFY** implied.
CSC-345	Generate 5-byte, 4-byte, 3-byte values when given an account number an an expiration date, **GENERATE** implied.
Operation (optional)	
VERIFY	Specifies verification of the value presented in the validation values variable.
GENERATE	Specifies generation of the value presented in the validation values variable.
Card Security Code Algorithm (One, optional)	
CSC-V1	Specifies use of CSC version 1.0 algorithm for generating or verifying the validation values. This is the default.
CSC-V2	Specifies use of CSC version 2.0 algorithm for generating or verifying the validation values.

`transaction_key_identifier_length`

Direction: Input Type: Integer

The length of the *transaction_key_identifier* parameter.

`transaction_key_identifier`

Direction: Input Type: String

The labelname or internal token of a MAC or MACVER class key. Key may be single or double length. When the CSC-V2 keyword is specified, the key must be a double-length key.

`transaction_info_length`

Direction: Input Type: Integer

The length of the *transaction_info* parameter. For American Express CSC codes, this length must be 19 if the algorithm for CSC v1.0 is specified and it must be 22 if the algorithm for CSC v2.0 is specified.

`transaction_info`

Direction: Input Type: String

Account information in character format. For American Express CSC-V1, this is a 19-byte field containing the concatenation of the 4-byte expiration data (in the format YYMM) and the 15-byte American Express account number. For CSC-V2, the string variable will contain the concatenation of the 4-byte expiration date in the format of (YYMM) , the 15-byte American Express account number and the 3-byte service code.

`validation_values_length`

Direction: Input/Output Type: Integer

The length of the *validation_values* parameter. Maximum value for this field is 64.

`validation_values`

Direction: Input Type: String

This variable contains American Express CSC values. The data is output for **GENERATE** and input for **VERIFY**.

Table 5. Output description for validation values

Operation	Element Description
GENERATE and **CSC-345**	5555544444333 where: 55555 = CSC 5 value 4444 = CSC 4 value 333 = CSC 3 value
VERIFY and **CSC-3**	333 = CSC 3 value
VERIFY and **CSC-4**	4444 = CSC 4 value
VERIFY and **CSC-5**	55555 = CSC 5 value

Usage Notes

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

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

Table 6. Required access control points for Transaction Validation

Operation keyword	Security code keyword	Access control point
GENERATE	CSC-345	Transaction Validation - Generate
VERIFY	CSC-3	Transaction Validation - Verify CSC-3
VERIFY	CSC-4	Transaction Validation - Verify CSC-4
VERIFY	CSC-5	Transaction Validation - Verify CSC-5

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

Table 7. Transaction validation required hardware

Server	Required cryptographic hardware	Restrictions
IBM @server zSeries 900		Not supported
IBM @server zSeries 990	PCI X Cryptographic Coprocessor	Requires May 2004 or later version of Licensed Internal Code (LIC)
IBM @server zSeries 890	Crypto Express2 Coprocessor	CSC-V1 and CSC-V2 keywords not supported.
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	For keywords CSC-V1 and CSC-V2, see the note below.
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	For keywords CSC-V1 and CSC-V2, see the note below.
z196	Crypto Express3 Coprocessor	For keywords CSC-V1 and CSC-V2, see the note below.

Note: The CSC-V1 and CSC-V2 keywords require the following MCLs:

- IBM System z9 EC and z9 BC - Driver 67L, EC G40492, MCL G40942.028 -> 030 Bundle 70 to be available November 2012.
- IBM System z10 EC and IBM System z10 BC Crypto Express2 - Driver 79F, EC N24393, MCL N24393.017 -> 019 Bundle 65a Available 9/26/12
- IBM System z10 EC and IBM System z10 BC Crypto Express3 - Driver 79F, EC N24380, MCL N24380.047 -> 049 Bundle 65a Available 9/26/12
- IBM System z196 and z114 Crypto Express3 - Driver 93G, EC N48132, MCL N48132.007 -> 009 Bundle 37a Available 10/3/12