z/OS

Cryptographic Services Integrated Cryptographic Service Facility EMV Simplification Services APAR OA47016

Contents

Chapter 1. Overview

This document describes changes to the Integrated Cryptographic Service Facility (ICSF) product in support of the following EMV simplification services:

- Six new callable services:
  - Derive ICC MK (CSNBDCM and CSNEDCM)
  - Derive Session Key (CSNBDSK and CSNEDSK)
  - EMV Scripting Service (CSNBESC and CSNEESC)
  - EMV Transaction (ARQC/ARPC) Service (CSNBEAC and CSNEEAC)
  - EMV Verification Functions (CSNBEVF and CSNEEVF)
  - Generate Issuer MK (CSNBGIM and CSNEGIM)

These changes are available through the application of the PTF for APAR OA47016 and apply to FMID HCR77B0, HCR77A1, and HCR77A0.

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

- *z/OS Cryptographic Services ICSF Overview*, SC14-7505-03
- *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-03
- *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-03
- *z/OS Cryptographic Services ICSF System Programmer's Guide*, SC14-7507-03

The technical changes made to the ICSF product by the application of the PTF for APAR OA47016 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, SC14-7505-03, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Overview*, SC14-7505-03, for the EMV simplification services provided by this APAR. Refer to this source document if background information is needed.

EMV integrated circuit card specifications

EMV stands for Europay, MasterCard, and Visa, the three companies that originally created the common standard for retail terminals accepting chip cards. Chip cards are also called stored value cards or smart cards. An algorithm or formula is stored in the chip. Chip card transactions are PIN-based for maximum security.

In addition to the EMV specification for contact chip cards, there are also specifications for contactless chip cards, common payment applications (CPA), card personalization, mobile payments, and tokenisation. These other specifications are being used by mobile payment systems and e-wallets.

The EMV standard is now managed by EMVCo, a consortium with control split equally among American Express, China UnionPay, Discover, Japan Credit Bureau (JCB), Mastercard, and Visa. EMVCo also has a variety of associates that include retailers, banks, payment processors, and other credit card companies and financial institutions. These associates provide both technical and strategic business input to EMVCo.

With ICSF, you can develop EMV ICC integrated circuit card applications using diversified key generate (CSNBDKG), secure messaging for PINs (CSNBSPN) and secure messaging for keys (CSNBSKY) services. ICSF supports the PIN change algorithms specified in the VISA Integrated Circuit Card Specification. PIN block/change (CSNBPCU), provides this support. Additionally, the diversified key generate service (CSNBDKG) supports the EMV2000 key generation algorithm.

ICSF callable services also simplify payment processing. These services use parameters that are specifically for EMV functions and call the correct sequence of existing ICSF services and cryptographic coprocessor services to perform EMV functions.

Generate Issuer Master Key Service (CSNBGIM)
This service helps with the initial steps of EMV setup by generating and storing the issuer master keys. The master keys are returned in either internal or external key tokens for key management.

Derive ICC Master Key Service (CSNBDCM)
This service generates an ICC master key from an issuer master key. The ICC master keys are needed for ICC personalization, EMV transaction processing, and EMV scripting. The master keys are returned in either internal or external key tokens for key management.

Derive Session Key Service (CSNBDSK)
This service generates a session key from either an issuer master key or an ICC master key. Session keys are needed for EMV transaction processing and EMV scripting.

EMV Transaction (ARQC/ARPC) Service (CSNBEAC)
This service simplifies EMV ARQC and ARPC transaction processing.

EMV Scripting Service (CSNBESC)
This service simplifies EMV scripting. Scripts may be encrypted for confidentiality, MAC'd for integrity, or both.

EMV Verification Service (CSNBEVF)
This service provides additional functions used by MasterCard:

- Verification of data authentication codes.
- Verification of ICC dynamic numbers.
- Decryption of encrypted counters.

Standards

IBM cryptographic coprocessor features and ICSF support cryptographic algorithms and techniques from International and geographical standards organizations.

EMV Integrated Circuit Card Specifications for Payment Systems.

ICSF provides a comprehensive set of key management services that allow you to create, generate, derive, import, and export keys needed for EMV online authorization processing.

ICSF provides services you can use in secure communications with EMV smart cards.

EMV ICC integrated circuit card applications can use the following callable services:

- Derive ICC MK (CSNBDCM and CSNEDCM).
- Derive Session Key (CSNBDSK and CSNEDSK).
- Diversified Key Generate (CSNBDKG and CSNEDKG).
- EMV Scripting Service (CSNBESC and CSNEESC).
- EMV Transaction Service (CSNBEAC and CSNEEAC).
- EMV Verification Functions (CSNBEVF and CSNEEVF).
- Generate Issuer MK (CSNBGIM and CSNEGIM).
- Secure Messaging for Keys (CSNBSKY and CSNESKY).
- Secure Messaging for PINs (CSNBSPN and CSNESPN).

The Diversified Key Generate (CSNBDKG and CSNEDKG) service supports the EMV2000 key derivation methods.

Summary of callable service support by hardware configuration

In Table 1 on page 5, letters represent various configurations according to:

- Letter A (**PCIXCC/CEX2C**) - IBM eServer zSeries 990 or IBM eServer zSeries 890 with CP Assist for Cryptographic Functions DES/TDES Enablement and PCIXCC/CEX2C.
- Letter B (**CEX2C/CEX3C**) - z9 EC, z9 BC, z10 EC, and z10 BC with CP Assist for Cryptographic Functions DES/TDES Enablement and CEX2C, or z10 EC and z10 BC with CP Assist for Cryptographic Functions DES/TDES Enablement and CEX3C.
- Letter C (**CEX3C**) - z114/z196 with CP Assist for Cryptographic Functions DES/TDES Enablement and CEX3C.

- Letter D (**CEX3C/CEX4C**) - IBM zEnterprise EC12 and BC12 with CP Assist for Cryptographic Functions DES/TDES Enablement and CEX3 and CEX4C.
- Letter E (**CEX5C**) - IBM z13 with CP Assist for Cryptographic Functions DES/TDES Enablement and CEX5C.

Table 1. Summary of ICSF callable services support

Service Name	Function	A	B	C	D	E
Derive ICC MK	Derives ICC master keys from issuer master keys.	X	X	X	X	X
Derive Session Key	Derives session keys from either issuer master keys or ICC master keys.	X	X	X	X	X
EMV Scripting Service	Simplifies EMV scripting. Scripts can be encrypted for confidentiality, MAC'd for integrity, or both.	X	X	X	X	X
EMV Transaction Service	Simplifies ARQC verification and ARPC generation.	X	X	X	X	X
EMV Verification Functions	Provides EMV functions used by MasterCard.	X	X	X	X	X
Generate Issuer MK	Generates issuer master keys and stores the keys in the CKDS.	X	X	X	X	X

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

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

DES key types

The DES keys are 64-bit, 128-bit, and 192-bit keys that use the DES algorithm to perform the cryptographic function. A 64-bit key is referred to as a single-length key. A 128-bit key is referred to as a double-length key. Triple-length keys are 192-bits in length. Only DATA keys can be triple-length.

For installations that do not support double-length key-encrypting keys, effective single-length keys are provided. For an effective single-length key, the clear key value of the left key half equals the clear key value of the right key half.

Table 2. Descriptions of DES key types and service usage

DES key type	**Usable with services**
DATA class (data operation keys) These key are used to encrypt and decrypt data. Single-length keys can be used to generate and verify MACs and CVVs. DATA keys can be single-length, double-length, or triple-length. DATAM and DATAMV keys are double-length.	
DATA	Authentication Parameter Generate, Cipher Text Translate2, CVV Key Combine, Decipher, Encipher, EMV Verification Functions, Field Level Decipher, Field Level Encipher, MAC Generate, MAC Verify, Symmetric Key Encipher, Symmetric Key Decipher, VISA CVV Generate, VISA CVV Verify
DATAM	MAC Generate, MAC Verify
DATAMV	MAC Verify
Cipher class (data operation keys) These key are used to encrypt and decrypt data. The keys can be single-length or double-length.	
CIPHER	Cipher Text Translate2, Decipher, Encipher, FPE Decipher, FPE Encipher, FPE Translate
DECIPHER	Cipher Text Translate2, Decipher, FPE Encipher, FPE Translate
ENCIPHER	Cipher Text Translate2, Encipher, FPE Decipher, FPE Translate
CIPHERXL class (cipher text translate keys) These key are used to translate cipher text. The keys are double--length.	
CIPHERXI	Cipher Text Translate2 (translate inbound key only)
CIPHERXL	Cipher Text Translate2 (translate inbound and outbound key)
CIPHERXO	Cipher Text Translate2 (translate outbound key only)
MAC class (data operation keys) These keys are used to generate and verify MACs, CVVs, and CSCs. The keys can be single-length or double-length keys.	

Table 2. Descriptions of DES key types and service usage (continued)

DES key type	Usable with services
MAC	CVV Key Combine, MAC Generate, MAC Verify, Transaction Validation, VISA CVV Generate, VISA CVV Verify
MACVER	CVV Key Combine, MAC Verify, Transaction Validation, VISA CVV Verify
PIN class These keys are used generate and verify PINs and PIN offsets. The keys are double-length keys.	
PINGEN	Clear PIN Generate, Clear PIN Generate Alternate, Encrypted PIN Generate, Recover PIN from Offset
PINVER	Encrypted PIN Verify
These keys are used wrap and unwrap PIN blocks:	
IPINENC	Authentication Parameter Generate, Clear PIN Generate Alternate, EMV Scripting Service, Encrypted PIN Translate, Encrypted PIN Verify, PIN Change/Unblock, Secure Messaging for PINs
OPINENC	Clear PIN Encrypt, Clear PIN Generate Alternate, EMV Scripting Service, Encrypted PIN Generate, Encrypted PIN Translate, PIN Change/Unblock, Recover PIN from Offset
Key-encrypting key class These keys are used to wrap other keys. The keys are double-length keys.	
EXPORTER	Control Vector Translate, Data Key Export, Derive ICC MK, ECC Diffie-Hellman, Generate Issuer MK, Key Export, Key Generate, Key Test2, Key Test Extended, Key Translate, Key Translate2, PKA Key Generate, PKA Key Translate, Prohibit Export Extended, Remote Key Export, Secure Messaging for Keys, Symmetric Key Generate, TR-31 Export, TR-31 Import, Unique Key Derive
IMPORTER	Control Vector Translate, Data Key Import, ECC Diffie-Hellman, Generate Issuer MK, Key Generate, Key Import, Key Test2, Key Test Extended, Key Translate, Key Translate2, Multiple Secure Key Import, PKA Key Generate, PKA Key Import, PKA Key Translate, Prohibit Export Extended, Remote Key Export, Restrict Key Attribute, Secure Key Import, Secure Messaging for Keys, Symmetric Key Generate, TR-31 Export, TR-31 Import
IMP-PKA	PKA Key Import, Remote Key Export, Trusted Block Create
IKEYXLAT, OKEYXLAT	Control Vector Translate, Key Translate, Key Translate2, TR-31 Export,TR-31 Import
Key-generate key class These keys are used to derive keys. The keys are double-length keys. The key usage flags in the control vector determine which services the KEYGENKY key may be used with.	
KEYGENKY	Diversified Key Generate, Encrypted PIN Translate, Encrypted PIN Verify, FPE Decipher, FPE Encipher, FPE Translate, Unique Key Derive
DKYGENKY	Derive ICC MK, Derive Session Key, Diversified Key Generate, EMV Scripting Service, EMV Transaction (ARQC/ARPC) Service, EMV Verification Functions, Generate Issuer MK, PIN Change/Unblock

Table 2. Descriptions of DES key types and service usage (continued)

DES key type	Usable with services
Cryptographic-variable class These keys are used in the special verbs that operate with cryptographic variables The keys are single-length keys.	
CVARENC	Cryptographic Variable Encipher
CVARXCVL	Control Vector Translate
CVARXCVR	Control Vector Translate
Secure-messaging class (data operation keys) These keys are used to encrypt keys or PINs. The keys are double-length keys. The key usage flags in the control vector determine which services the key may be used with.	
SECMSG	Diversified Key Generate, Secure Messaging for Keys, Secure Messaging for PINs

EMV simplification services

EMV simplification services are provided to assist in implementing EMV processing. These services provide key management support for generating master keys and deriving session keys. They provide EMV processing and functions for MasterCard, EMV, and Visa standards.

For Visa, the Cryptogram Version 10 key derivation is used. See Visa specification, Appendix D2. Padding is with binary zeroes until the length is a multiple of 8 bytes.

For MasterCard, M/CHIP 2 key derivation is used. EMV padding rules are used.

For EMV, the session key derivation is used as described in EMV Book 2, Annex A1.3. EMV padding rules are used.

Master keys are:

Application Cryptogram Key (AC)
Used during EMV Transaction Processing (ARQC/ARPC).

Secure Messaging Authentication Key (MAC)
Used to provide integrity for EMV scripting.

Secure Confidentiality Key (ENC)
Used to provide confidentiality for EMV scripts containing PINs.

DATA Key (DATA)
Used to encrypt and decrypt data used in EMV Verification Functions.

Derive ICC Master Key callable service (CSNBDCM and CSNEDCM)

This service generates an ICC master key from an issuer master key. The ICC master keys are needed for ICC personalization, EMV transaction processing, and EMV scripting. Optionally, this service returns the ICC master key as an external token under a key-encrypting key (KEK).

Inputs are:

- Issuer master key (key token or CKDS label).
- PAN and PAN sequence number.
- Key-encrypting key to wrap the generated master key (optional).

Outputs are:

- Internal or external ICC master key token.

See "Derive ICC MK (CSNBDCM and CSNEDCM)" on page 12 for more information.

Derive Session Key callable service (CSNBDSK and CSNEDSK)

This service generates a session key from either the Issuer or ICC master key. Session keys are needed for EMV transaction processing or EMV scripting.

Inputs are:

- Issuer or ICC master key (key token or CKDS label).
- PAN, PAN sequence number, Application Transaction Counter (ATC), and unpredictable number.

Outputs are:

- Internal session key token.

See "Derive Session Key (CSNBDSK and CSNEDSK)" on page 19 for more information.

EMV Scripting callable service (CSNBESC and CSNEESC)

EMV Scripting is a mechanism for sending commands to a payment card. The commands are used to update card parameters including potentially the PIN. Commands may be encrypted for confidentiality or MAC'd for integrity or both.

Scripts are generated by the issuer, or the issuer's agent, when a transaction is received from a payment card. This service receives the script as input, encrypts it, MAC's it, or both, and then returns either the encrypted script, the MAC, or both. The output is intended to be sent back to the payment card along with the response.

This service provides the following functions:

- Scripting with integrity.
- Scripting with confidentiality (for protection of scripts that may or may not contain a PIN).
- Scripting with confidentiality and integrity.
- PIN change/unblock.

Inputs are:

- Issuer MAC and ENC master key or keys (key token or CKDS label).
- PAN, PAN sequence number, script message, Application Transaction Counter (ATC), and unpredictable number.
- PIN Block, PIN Key, and PIN Format (optional).

Outputs are:

- Script message.
- MAC (optional).

See "EMV Scripting Service (CSNBESC and CSNEESC)" on page 32 for more information.

EMV Transaction (ARQC/ARPC) callable service (CSNBEAC and CSNEEAC)

This service provides EMV Authorization Request Cryptogram (ARQC) and Authorization Response Cryptogram (ARPC) transaction processing.

An ARQC is generated by the EMV card upon request from the point of sales terminal to obtain authorization for payment. The ARQC is forwarded across the payment network to the issuer for verification. After the issuer has verified the ARQC, the issuer generates an ARPC (the response). The ARPC is sent back through the payment network to the point of sales terminal to authorize the transaction.

This service performs the following EMV functions:

- Verification of the Authorization Request Cryptogram (ARQC).
- Generation of the Authorization Response Cryptogram (ARPC).
- Both operations combined: Verify the ARQC and generate the ARPC.

Inputs are:

- Issuer AC master key or keys (key token or CKDS label).
- PAN, PAN sequence number, Cryptogram Information, Application Transaction Counter (ATC), Authorization Response Code (ARC), ARQC, and unpredictable number.

Output is:

- ARPC.

See "EMV Transaction (ARQC/ARPC) Service (CSNBEAC and CSNEEAC)" on page 44 for more information.

EMV Verification callable service (CSNBEVF and CSNEEVF)

This service provides the following additional functions used by MasterCard:

- Verification of data authentication codes.
- Verification of ICC dynamic numbers.
- Decryption of encrypted counters.

Inputs are:

- Issuer master key (key token or CKDS label).
- PAN, PAN sequence number, data field, Application Transaction Counter (ATC), and unpredictable number.

Outputs are:

- Return and reason code.
- Decrypted counters.

See “EMV Verification Functions (CSNBEVF and CSNEEVF)” on page 51 for more information.

Generate Issuer Master Key callable service (CSNBGIM and CSNEGIM)

This service is intended to help with the initial steps of EMV setup by generating and storing the issuer master keys in the CKDS. Optionally, the keys can be returned as external tokens under a key-encrypting key (KEK) that is shared with the ICC personalization system.

Inputs are:

- CKDS label for the issuer master key.
- Key-encrypting key (KEK) to wrap the generated master key (optional).

Outputs are:

- Issuer master key in the CKDS.
- Internal or external issuer master key token.

See “Generate Issuer MK (CSNBGIM and CSNEGIM)” on page 25 for more information.

Managing Symmetric Cryptographic Keys

Derive ICC MK (CSNBDCM and CSNEDCM)

The Derive ICC MK callable service generates ICC master keys from issuer master keys. ICC master keys are needed for ICC personalization, EMV transaction processing, and EMV scripting. Optionally, this service returns the ICC master key as an external token wrapped under a key-encrypting key (KEK). Use the TKE workstation to establish the KEK that is optionally used by this service.

The following ICC master keys can be generated:

ICC Master Application Cryptogram Key (AC)
This key is used to generate and verify the ARQC and ARPC.

ICC Master Secure Messaging Authentication Key (MAC)
This key is used to provide integrity for EMV scripting.

ICC Master Secure Messaging Confidentiality Key (ENC)
This key is used to provide confidentiality for EMV scripting.

ICC Master Data Key (DATA)
This key is used for functions that require encryption and decryption of EMV fields.

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

Format

```
CALL CSNBDCM(
            return_code,
            reason_code,
            exit_data_length,
```

```
exit_data,
rule_array_count,
rule_array,
issuer_master_key_identifier_length,
issuer_master_key_identifier,
icc_master_key_identifier_length,
icc_master_key_identifier,
transport_key_identifier_length,
transport_key_identifier,
pan_length,
pan,
pan_seq_number,
reserved1_length,
reserved1,
reserved2_length,
reserved2)
```

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 assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

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

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The minimum value is 3 and the maximum value is 5.

rule_array

Direction	Type
Input	String

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 3. Rule array keywords for Derive ICC MK

Keyword	Meaning
Algorithm (Required)	
TDES	Specifies the use of Triple-DES.
Key mode (One required). Defines the key derivation mechanism.	
VISA	Use this key mode for Visa Cryptogram Version 10 key derivation.
MC	Use this key mode for MasterCard M/CHIP 2.1 key derivation.
Output key type (One required). See the *issuer_master_key_identifier* and *icc_master_key_identifier* parameters for more information.	
AC	Derives the ICC Master Application Cryptogram Key. This key is used to generate and verify the ARQC and ARPC. When the VISA key mode is specified, the issuer master key must be of a DKYL0 DKYGENKY key and the derived ICC master key will be of type MAC. When the MC key mode is specified, the issuer master key must be a DKYL1 DKYGENKY key and the derived ICC master key will be a DKYL0 DKYGENKY key.
MAC	Derives the ICC Master Secure Messaging Authentication Key. This key is used to provide integrity for EMV scripting. When the VISA key mode is specified, the issuer master key must be of a DKYL0 DKYGENKY key and the derived ICC master key will be of type MAC. When the MC key mode is specified, the issuer master key must be a DKYL1 DKYGENKY key and the derived ICC master key will be a DKYL0 DKYGENKY key.
ENC	Derives the ICC Master Secure Messaging Confidentiality Key. This key is used to provide confidentiality for EMV scripting. When the MC key mode is specified, the issuer master key must be a DKYL1 DKYGENKY key and the derived ICC master key will be a DKYL0 DKYGENKY key. Not valid with key mode VISA.

Table 3. Rule array keywords for Derive ICC MK (continued)

Keyword	Meaning
DATA	Derives the ICC Master DATA Key. This key is used for functions that require encryption and decryption of EMV fields. When the MC key mode is specified, the issuer master key must be a DKYL1 DKYGENKY key and the derived ICC master key will be a DKYL0 DKYGENKY key. Not valid with key mode VISA.
Key encryption (Optional)	
MASTER	Specifies to return the ICC master key as an internal token encrypted under the master key. This is the default.
XPORT	Specifies to return the ICC master key as external token encrypted under the *transport_key_identifier*.
Control flag (Optional)	
APPANSEQ	Specifies to append the PAN sequence number when the card specific master key is derived. See the descriptions of *pan* and *pan_seq_number*. The default is not to append the PAN sequence number.

issuer_master_key_identifier_length

Direction	Type
Input	Integer

Specifies the length of the *issuer_master_key_identifier* parameter in bytes. The value must be 64.

issuer_master_key_identifier

Direction	Type
Input/Output	String

A 64-byte DES key identifier (either an internal token or key label) for the issuer master key. The issuer master key is the key from which the ICC master key is derived.

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

The key algorithm must be DES and the key type must be DKYGENKY. The value subtype and key usage attributes required are listed in Table 4.

Table 4. Derive ICC MK: Key requirements

Master key	VISA	MC
Application Cryptogram Key (AC)	DMAC, DKYL0	DMAC, DKYL1
Secure Messaging Authentication Key (MAC)	DMAC, DKYL0	DMAC, DKYL1
Secure Messaging Confidentiality Key (ENC)	N/A	DMPIN, DKYL1
Data Key (DATA)	N/A	DDATA, DKYL1

icc_master_key_identifier_length

Direction	Type
Input	Integer

This parameter specifies the length of the *icc_master_key_identifier* parameter in bytes. The value must be 64.

icc_master_key_identifier

Direction	Type
Output	String

A 64-byte CCA DES key identifier for the ICC master key. The ICC master key is the DES key from which session keys are derived.

On output, this is the derived key token containing the ICC master key. If the XPORT rule is specified, the key token is returned in external format wrapped by the *transport_key_identifier*. Otherwise, it is returned in internal format.

The attributes of the generated key (See the output key type rules for a description of key types derived by this service based on the selected key mode):

Table 5. Derive ICC MK: Key type and key usage attributes of the generated keys

Master key	VISA	MC
Application Cryptogram Key (AC)	MAC	DKYGENKY, DMAC, DKYL0
Secure Messaging Authentication Key (MAC)	MAC	DKYGENKY, DMAC, DKYL0
Secure Messaging Confidentiality Key (ENC)	N/A	DKYGENKY, DMPIN, DKYL0
Data Key (DATA)	N/A	DKYGENKY, DDATA, DKYL0

transport_key_identifier_length

Direction	Type
Input	Integer

This parameter specifies the length of the *transport_key_identifier* parameter in bytes. When the XPORT keyword is specified, the value must be 64. Otherwise, the value must be 0.

transport_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to wrap the generated keys. This key must be an EXPORTER key type specified as an operational key token or as a key label of an EXPORTER key in key storage. When the *transport_key_identifier_length* is zero, this parameter is ignored.

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

The NOCV bit is shown in Table 390. Internal Key Token Format on page 997.

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

pan_length

Direction	Type
Input	Integer

Length in bytes of the *pan* parameter. The value must be 10.

pan

Direction	Type
Input	String

The 10-byte EMV card's Primary Account Number. The data must be in compressed numeric format and right justified in a 10-byte field, padded to the left with zeroes. For example, PAN 1234567890 must be provided as x'00000000001234567890'.

This data is used in combination with the PAN sequence number to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

pan_seq_number

Direction	Type
Input	String

The 1-byte sequence number of the EMV card's Primary account Number. If the APPANSEQ control flag rule array keyword was specified, this PAN sequence number is used in combination with the PAN to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

reserved1_length

Direction	Type
Input	Integer

Length in bytes of the *reserved1* parameter. The value must be 0.

reserved1

Direction	Type
Input	String

This field is ignored.

reserved2_length

Direction	Type
Input	Integer

Length in bytes of the *reserved2* parameter. The value must be 0.

reserved2

Direction	Type
Input	String

This field is ignored.

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.

Cryptographic services used by Derive ICC MK

The following CCA cryptographic services are used by Derive ICC MK:

- CSNBKTB - Key Token Build
- CSNBDKG – Diversified Key Generate
- CSNBKEX - Key Export

The caller does not require authorization to each of these services, only to Derive ICC MK. Additionally, the caller must have the required access control points enabled (see information below about required access control points).

Access control points

The following access control points must be enabled to use Derive ICC MK:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-XOR
- Diversified Key Generate - TDESEMV2/TDESEMV4

Required hardware

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

Table 6. Derive ICC MK required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor Crypto Express2 Coprocessor	
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	

Table 6. Derive ICC MK required hardware (continued)

Server	Required cryptographic hardware	Restrictions
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	
IBM z13	Crypto Express5 CCA Coprocessor	

Derive Session Key (CSNBDSK and CSNEDSK)

The Derive Session Key callable service derives a session key from either an issuer master key or an ICC master key. The session key can be used for EMV transaction processing or EMV scripting.

The following session keys can be derived for Visa Cryptogram Version 10 processing method (VISA):

- Application Cryptogram Session Key (AC) for ARQC and ARPC processing.
- Secure Messaging Authentication Session Key (MAC) for scripting.
- Secure Messaging Confidentiality Session Key (ENC) for scripting.

The following session keys can be derived for MasterCard M/CHIP 2.1 processing method (MC):

- Application Cryptogram Session Key (AC) from the issuer master key ARQC and ARPC processing.
- ARPC key (AC) from the issuer ARPC master key for ARPC processing.
- Secure Messaging Authentication Session Key (MAC) from either the issuer or ICC master key for scripting.
- Secure Messaging Confidentiality Session Key (ENC) from either the issuer or ICC master key for scripting.
- DATA Session Key (DATA) from either the issuer or ICC master key for encryption and decryption of EMV fields.

The following session keys can be derived for EMV Book 2, Annex A1.3, Visa Cryptogram Version 14, and MasterCard M/CHIP 4.MasterCard M/CHIP 2.1 processing method (EMV):

- Application Cryptogram Session Key (AC) for ARQC and ARPC processing.
- Secure Messaging Authentication Session Key (MAC) for scripting.
- Secure Messaging Confidentiality Session Key (ENC) for scripting.
- DATA Session Key (DATA) for encryption and decryption of EMV fields.

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

Format

```
CALL CSNBDSK(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
```

```
rule_array_count,
rule_array,
master_key_identifier_length,
master_key_identifier,
session_key_identifier_length,
session_key_identifier,
pan_length,
pan,
pan_seq_number,
atc,
unpredictable_number_length,
unpredictable_number,
reserved1_length,
reserved1,
reserved2_length,
reserved2)
```

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 assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

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

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The minimum value is 3 and the maximum value is 5.

rule_array

Direction	Type
Input	String

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 7. Rule array keywords for Derive Session Key

Keyword	Meaning
Algorithm (Required)	
TDES	Specifies the use of Triple-DES.
Key mode (One required). Defines the key derivation mechanism. See the output key type for more information.	
VISA	Specifies to use the Visa Cryptogram Version 10 key derivation. The card's master key is used as the session key (the keys are the same for each session). See Visa specification, Appendix D2. Padding is with binary zeroes until the length is a multiple of 8 bytes.
MC	Specifies to use the MasterCard M/CHIP 2.1 key derivation. The ATC and an unpredictable number is encrypted with the card's master key. The card's master key is used when generating the ARPC. Padding is according to EMV.
EMV	Specifies to use the session key derivation as described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.3. Use this key mode for Visa Cryptogram Version 14 and MasterCard M/CHIP 4. EMV padding rules apply.
Output key type (One required). See the *issuer_master_key_identifier* description for the requirements for the key-generating key.	
AC	Specifies to derive the Application Cryptogram Session Key (AC). The derived session key will be of the type MAC.
MAC	Specifies to derive the Secure Messaging Authentication Session Key (MAC). The derived session key will be of the type MAC.
ENC	Specifies to derive the Secure Messaging Confidentiality Session Key (ENC). The derived session key will be of the type SECMSG.
DATA	Specifies to derive the DATA Session Key (DATA). The derived session key will be of the type DATA. Not valid with key mode VISA.
Control flag (Optional)	
APPANSEQ	Specifies to append the PAN sequence number when the card specific master key is derived. See the descriptions of *pan* and *pan_seq_number*. The default is not to append the PAN sequence number.
Branch Factor (One optional, valid only with key mode EMV). The branching factor is to be used in EMV session key derivation.	
TDESEMV2	Specifies a branch factor of 2 for a height of 16. This is the default.

Table 7. Rule array keywords for Derive Session Key (continued)

Keyword	Meaning
TDESEMV4	Specifies a branch factor of 4 for a height of 8.

master_key_identifier_length

Direction	Type
Input	Integer

Specifies the length of the *issuer_master_key_identifier* parameter in bytes. The value must be 64.

master_key_identifier

Direction	Type
Input/Output	String

A 64-byte DES key identifier (either an internal token or key label) for the issuer master key or the ICC master key. This key is the DES key from which card specific keys and session keys are derived.

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

The key algorithm must be DES and the key type must be DKYGENKY. The required subtype and key usage attributes for the master key type and key mode combinations are listed in Table 8.

Table 8. Derive Session Key: Key requirements

Master key	VISA	MC	EMV
Issuer AC	DMAC, DKYL0	DMAC, DKYL1	DMAC, DKYL0
Issuer MAC	DMAC, DKYL0	DMAC, DKYL1	DMAC, DKYL0
Issuer ENC	DMPIN, DKYL0	DMPIN, DKYL1	DMPIN, DKYL0
Issuer DATA	N/A	DDATA, DKYL1	DDATA, DKYL0
Issuer APRC	N/A	DMAC, DKYL0	N/A
ICC AC	N/A	DMAC, DKYL0	N/A
ICC MAC	N/A	DMAC, DKYL0	N/A
ICC ENC	N/A	DMPIN, DKYL0	N/A
ICC DATA	N/A	DDATA, DKYL0	N/A

session_key_identifier_length

Direction	Type
Input/Output	Integer

This parameter specifies the length of the *session_key_identifier* parameter in bytes. The value must be 64.

session_key_identifier

Direction	Type
Output	String

The 64-byte parameter for the derived session key token. The session key can be used for EMV transaction processing or EMV scripting.

The session key type generated based on the master key and the key mode specified are shown in Table 9.

Table 9. Derive Session Key: Attributes of the key generated

Session Key	VISA	MC	EMV
AC	MAC	MAC	MAC
MAC	MAC	MAC	MAC
ENC	SECMSG, SMPIN	SECMSG, SMPIN	SECMSG, SMPIN
DATA	N/A	DATA	DATA
ARPC (AC)	N/A	MAC	N/A

pan_length

Direction	Type
Input	Integer

Length in bytes of the *pan* parameter. The value must be 10.

pan

Direction	Type
Input	String

The 10-byte EMV card's Primary Account Number. The data must be in compressed numeric format and right justified in a 10-byte field, padded to the left with zeroes. For example, PAN 1234567890 must be provided as x'0000000001234567890'.

This data is used in combination with the PAN sequence number to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

pan_seq_number

Direction	Type
Input	String

The 1-byte sequence number of the EMV card's Primary account Number. If the APPANSEQ control flag rule array keyword was specified, this PAN sequence number is used in combination with the PAN to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

atc

Direction	Type
Input	String

The 2-byte application transaction counter that is used for session key derivation. See the key mode rules for more information on session key derivation.

Note: The first byte is the high-order byte and the second byte is the low order byte.

unpredictable_number_length

Direction	Type
Input	Integer

Specifies the length of the *unpredictable_number* in bytes. The value must be 4 or 8.

unpredictable_number

Direction	Type
Input	String

The 4-byte or 8-byte unpredictable number used in the MasterCard M/Chip 2.1 session key derivation scheme.

For EMV scripting with the MC key mode, this value is expected to be 8 bytes. For EMV transaction processing and verification functions using the MC key mode, this value is expected to be 4 bytes.

The data in this field will not be reformatted by the API before use.

reserved1_length

Direction	Type
Input	Integer

Length in bytes of the *reserved1* parameter. The value must be 0.

reserved1

Direction	Type
Input	String

This field is ignored.

reserved2_length

Direction	Type
Input	Integer

Length in bytes of the *reserved2* parameter. The value must be 0.

reserved2

Direction	Type
Input	String

This field is ignored.

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.

Cryptographic services used by Derive Session Key

The following CCA cryptographic services are used by Derive Session Key:

- CSNBKTB - Key Token Build
- CSNBDKG – Diversified Key Generate

The caller does not require authorization to each of these services, only to Derive Session Key. Additionally, the caller must have the required access control points enabled (see information below about required access control points).

Access control points

The following access control points must be enabled to use Derive Session Key:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-XOR
- Diversified Key Generate - TDESEMV2/TDESEMV4

Required hardware

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

Table 10. Derive Session Key required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor Crypto Express2 Coprocessor	
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	
IBM z13	Crypto Express5 CCA Coprocessor	

Generate Issuer MK (CSNBGIM and CSNEGIM)

The Generate Issuer MK callable service helps with the initial steps of EMV setup by generating and storing the issuer master keys. Optionally, the issuer master keys can be returned as external tokens wrapped under a key-encrypting key

(KEK) that is shared with the ICC personalization system. Use the TKE workstation to establish the KEK that is optionally used by this service.

Use the Generate Issuer MK service to generate the issuer master keys for use in EMV processing. The master keys are double-length DES key of key type DKYGENKY. The subtype and key usage attributes depends on the type of master key and key mode.

The generated key is stored in the CKDS using the label supplied and returned to the caller in an internal token. The label must not exist in the CKDS. Optionally, the key can be returned as external tokens wrapped under a key-encrypting key.

The following master keys can be generated:

Issuer Master Application Cryptogram Key (AC)
For VISA and EMV, this key is used for deriving session keys for ARQC verification and ARPC generation.

For MasterCard, two keys are generated:

- The issuer master key used for deriving session keys for ARQC verification.
- The ARPC master key used for deriving session keys for ARPC generation.

Issuer Master Secure Messaging Authentication Key (MAC)
This key is used for deriving session keys to provide integrity for EMV scripting.

Issuer Master Secure Messaging Confidentiality Key (ENC)
This key is used for deriving session keys to provide confidentiality for EMV scripting.

Issuer Master Data Key (DATA)
This key is used for deriving session keys for functions that require encryption and decryption of EMV fields for EMV and MasterCard.

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

Format

```
CALL CSNBGIM(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             rule_array_count,
             rule_array,
             issuer_master_key_identifier_length,
             issuer_master_key_identifier,
             issuer_ARPC_master_key_identifier_length,
             issuer_ARPC_master_key_identifier,
             transport_key_identifier_length,
             transport_key_identifier,
             reserved1_length,
             reserved1,
             reserved2_length,
             reserved2)
```

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 assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

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

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The minimum value is 3 and the maximum value is 4.

rule_array

Direction	Type
Input	String

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 11. Rule array keywords for Generate Issuer MK

Keyword	Meaning
Algorithm (Required)	
TDES	Specifies the use of Triple-DES.
Key mode (One required)	

Table 11. Rule array keywords for Generate Issuer MK (continued)

Keyword	Meaning
VISA	Use this key mode for Visa Cryptogram Version 10.
MC	Use this key mode for MasterCard M/CHIP 2.1.
EMV	Use this key mode for EMV book 2, Annex A1.3, Visa Cryptogram Version 14, and MasterCard M/CHIP 4.
Output key type (One required). See the *issuer_master_key_identifier* and *issuer_ARPC_master_key_identifier* parameters for the attributes of the generated keys.	
AC	Generates the Issuer Master Application Cryptogram Key. This key is used for deriving session keys for ARQC verification and ARPC generation.
MAC	Generates the Issuer Master Secure Messaging Authentication Key. This key is used to provide integrity for EMV scripting.
ENC	Generates the Issuer Master Secure Messaging Confidentiality Key. This key is used to provide confidentiality for EMV scripting.
DATA	Generates the Issuer Master Data Key. This key is used for functions that require encryption and decryption of EMV fields. Not valid with the VISA key mode.
Key encryption (Optional)	
MASTER	Specifies to return the ICC Master Key as an internal token encrypted under the master key. This is the default.
XPORT	Specifies to return the issuer master key or issuer master keys as external tokens wrapped under the *transport_key_identifier*.

issuer_master_key_identifier_length

Direction	Type
Input	Integer

Specifies the length of the *issuer_master_key_identifier* parameter in bytes. The value must be 64.

issuer_master_key_identifier

Direction	Type
Input/Output	String

The key identifier of the master key being generated.

On input, this is a 64-character label of the record to be added to the CKDS. The supplied CKDS label must not already exist. This service will not overwrite existing CKDS records.

On output, the generated key is stored in the CKDS and is returned in a key token in this parameter. When the XPORT keyword is specified, the token is returned in external format wrapped under the key-encrypting key supplied in the *transport_key_identifier* parameter. Otherwise, it is returned in internal format.

All master keys generated are double-length DES keys of key type DKYGENKY. The subtype and key usage attributes for the master keys are listed in Table 12.

Table 12. Generate Issuer MK: Attributes of the generated key

Master key	VISA	MC	EMV
Application Cryptogram Key (AC)	DMAC, DKYL0	DMAC, DKYL1	DMAC, DKYL0
Secure Messaging Authentication Key (MAC)	DMAC, DKYL0	DMAC, DKYL1	DMAC, DKYL0
Secure Messaging Confidentiality Key (ENC)	DMPIN, DKYL0	DMPIN, DKYL1	DMPIN, DKYL0
Data Key (DATA)	N/A	DDATA, DKYL1	DDATA, DKYL0

issuer_ARPC_master_key_identifier_length

Direction	Type
Input	Integer

This parameter specifies the length of the *issuer_ARPC_master_key_identifier* parameter in bytes. When the key mode keyword is MC and the output key type keyword is AC, the value must be 64. Otherwise, the value must be zero.

issuer_ARPC_master_key_identifier

Direction	Type
Input/Output	String

The key identifier of the master key being generated. If the *issuer_ARPC_master_key_identifier* parameter is zero, this parameter is ignored.

On input, this is a 64-character label of the record to be added to the CKDS. The supplied CKDS label must not already exist. This service will not overwrite existing CKDS records.

On output, the generated key is stored in the CKDS and is returned in a key token in this parameter. When the XPORT keyword is specified, the token is returned in external format wrapped under the key-encrypting key supplied in the *transport_key_identifier* parameter. Otherwise, it is returned in internal format.

The attributes of the generated key:

- Key algorithm is DES.
- Key type is DKYGENKY.
- Subtype is DKYL0.
- Key usage is DMAC.

transport_key_identifier_length

Direction	Type
Input	Integer

This parameter specifies the length of the *transport_key_identifier* parameter in bytes. When the XPORT keyword is specified, the value must be 64. Otherwise, the value must be 0.

transport_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to wrap the generated keys. The key identifier is an operational token or the key label of an operational token in key storage. When the *transport_key_identifier_length* is zero, this parameter is ignored.

The key identifier must be a DES IMPORTER or EXPORTER. If an EXPORTER key is supplied, it may be a NOCV EXPORTER. If an IMPORTER is supplied and the MasterCard AC master keys are being generated, it cannot be an NOCV IMPORTER. A NOCV IMPORTER may be used for the generation of other master keys.

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

The NOCV bit is shown in Table 390. Internal Key Token Format on page 997.

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

reserved1_length

Direction	Type
Input	Integer

Length in bytes of the *reserved1* parameter. The value must be 0.

reserved1

Direction	Type
Input	String

This field is ignored.

reserved2_length

Direction	Type
Input	Integer

Length in bytes of the *reserved2* parameter. The value must be 0.

reserved2

Direction	Type
Input	String

This field is ignored.

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.

Cryptographic services used by Generate Issuer MK

The following CCA cryptographic services are used by Generate Issuer MK:

- CSNBKEX - Key Export
- CSNBKGN - Key Generate
- CSNBKIM - Key Import
- CSNBKRC2 - CKDS Key Record Create2
- CSNBKTB - Key Token Build

The caller does not require authorization to each of these services, only to Generate Issuer MK. Additionally, the caller must have the required access control points enabled (see information below about required access control points).

Access control points

The following access control points must be enabled to use Generate Issuer MK:

- Key Export
- Key Generate - OP
- Key Import
- NOCV KEK usage for export-related functions

Required hardware

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

Table 13. Generate Issuer MK required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor Crypto Express2 Coprocessor	
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	
IBM z13	Crypto Express5 CCA Coprocessor	

Financial Services

EMV Scripting Service (CSNBESC and CSNEESC)

The EMV Scripting Service is a mechanism for sending commands to an EMV payment card. The commands are used to update card parameters including potentially the PIN. Commands may be encrypted for confidentiality or MAC'd for integrity or both.

Scripts are generated by the issuer, or the issuer's agent, when a transaction is received from a payment card. This service receives the script as input, encrypts, MAC's it or both, and returns either the encrypted script, the MAC, or both. The output is intended to be sent back to the payment card along with the response.

This service performs the following EMV scripting functions:

- Scripting with integrity.

  The message is MAC'd with a session key derived from the issuer master key specified in the *issuer_integrity_master_key_identifier* parameter.
- Scripting with confidentiality (for protection of scripts that may or may not contain a PIN).

  The message is encrypted with a session key derived from the issuer master key specified in the *issuer_confidentiality_master_key_identifier* parameter.
- Scripting with confidentiality and integrity (for protection of scripts that may or may not contain a PIN).

  The message is first encrypted (for confidentiality) with a session key derived from the issuer master key specified in the *issuer_confidentiality_master_key_identifier* parameter and then it is MAC'd (for integrity) with a session key derived from the issuer master key specified in the *issuer_integrity_master_key_identifier* parameter.
- PIN change/unblock.

  Visa PIN change/unblocking as described in VISA Integrated Circuit Card Specification, v1.4.0. The PIN can be changed by either specifying the new PIN only or specifying both the current PIN and the new PIN. See *new_PIN_encrypting_key_identifier* and *current_PIN_encrypting_key_identifier* parameters for additional information. The message is encrypted with a session key derived from the issuer master key specified in the *issuer_confidentiality_master_key_identifier* parameter.

This service can be used in the following specific brand modes:

- Visa
- MasterCard
- EMV

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

Format

```
CALL CSNBESC(
            return_code,
            reason_code,
```

```
exit_data_length,
exit_data,
rule_array_count,
rule_array,
issuer_integrity_master_key_identifier_length,
issuer_integrity_master_key_identifier,
issuer_confidentiality_master_key_identifier_length,
issuer_confidentiality_master_key_identifier,
new_PIN_encrypting_key_identifier_length,
new_PIN_encrypting_key_identifier,
current_PIN_encrypting_key_identifier_length,
current_PIN_encrypting_key_identifier,
new_PIN_block,
current_PIN_block,
pan_length,
pan,
pan_seq_number,
atc,
unpredictable_number,
input_message_length,
input_message,
PIN_offset,
PIN_format,
output_message_length,
output_message,
mac_length,
mac,
reserved1_length,
reserved1,
reserved2_length,
reserved2)
```

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 assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

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

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The minimum value is 3 and the maximum value is 5.

rule_array

Direction	Type
Input	String

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 14. Rule array keywords for EMV Scripting Service

Keyword	Meaning
Algorithm (Required)	
TDES	Specifies the use of Triple-DES.
Action (One required)	
SMINT	Secure messaging with integrity.
SMCON	Secure messaging with confidentiality. Only MC and EMV key modes supported.
SMCONPIN	Secure messaging with confidentiality for commands containing a PIN.
SMCONINT	Secure messaging with both confidentiality and integrity. Only MC and EMV key modes supported.
SMCIPIN	Secure messaging with both confidentiality and integrity for commands containing a PIN.
VISAPIN	Visa PIN change/unblocking as described in VISA Integrated Circuit Card Specification, v1.4.0.
Key mode (One required). Defines the key derivation mechanism.	
VISA	Specifies to use the Visa Cryptogram Version 10 key derivation. Not valid with SMCON and SMCONINT.
MC	Specifies to use the MasterCard M/CHIP 2.1 key derivation. The *random_number* parameter is used. Padding is according to EMV rules.
EMV	Specifies to use the session key derivation as described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.3.
Control flag (Optional)	

Table 14. Rule array keywords for EMV Scripting Service (continued)

Keyword	Meaning
APPANSEQ	Specifies to append the PAN sequence number when the card specific master key is derived. See the descriptions of *pan* and *pan_seq_number*. The default is not to append the PAN sequence number.
Branch Factor (One optional, valid only with key mode EMV). The branching factor is to be used in EMV session key derivation.	
TDESEMV2	Specifies a branch factor of 2 for a height of 16. This is the default.
TDESEMV4	Specifies a branch factor of 4 for a height of 8.

issuer_integrity_master_key_identifier_length

Direction	Type
Input	Integer

Specifies the length of the *issuer_integrity_master_key_identifier* parameter in bytes. The value must be 0 or 64. When the action keywords SMCON or SMCONPIN is specified, the value must be 0 and the *issuer_integrity_master_key_identifier* parameter is ignored. Otherwise, the value must be 64 and a key token or label must be supplied in the *issuer_integrity_master_key_identifier* parameter.

issuer_integrity_master_key_identifier

Direction	Type
Input/Output	String

The 64-byte DES key identifier (either an internal token or key label) for the issuer master key to be used for secure messaging with integrity (actions SMINT, SMCONINT, and SMCIPIN) or the issuer master key for authentication (action VISAPIN).

The issuer master key is the DES key from which the card specific keys and session keys for scripting are derived.

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

Table 15. EMV Scripting Service: Key requirements

Key mode keyword	Key derivation used	Key type	Subtype	Key generating bits
VISA	SESS-XOR	DKYGENKY	0	Must be '0010' (keyword DMAC).
MC	MasterCard M/Chip 2.1	DKYGENKY	1	Must be '0010' (keyword DMAC).
EMV	EMV key derivation	DKYGENKY	0	Must be '0010' (keyword DMAC).

Note: For action VISAPIN, this is the issuer master key for authentication. The key is used for preparing the material used to form the PIN block for PIN change/unblock. The card specific key is derived using the same PAN data as

for the other issuer master keys. The key must be of type DKYGENKY, subtype 0, and the key generating bits must be '0010' (keyword DMAC).

issuer_confidentiality_master_key_identifier_length

Direction	Type
Input	Integer

This parameter specifies the length of the *issuer_confidentiality_master_key_identifier* parameter in bytes. The value must be 0 or 64. When action keyword SMINT is specified, the value must be 0 and the *issuer_confidentiality_master_key_identifier* parameter is ignored. Otherwise, the value must be 64 and a key token or label must be supplied in the *issuer_confidentiality_master_key_identifier* parameter.

issuer_confidentiality_master_key_identifier

Direction	Type
Input/Output	String

The 64-byte DES key identifier (either an internal token or key label) for the issuer master key to be used for secure messaging with confidentiality (actions SMCON, SMCONPIN, SMCONINT, SMCIPIN, VISAPIN). The issuer master key is the DES key from which card specific keys and session keys for scripting are derived.

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

Table 16. Key type requirements for actions SMCON and SMCONINT

Key mode keyword	Key derivation used	Key type	Subtype	Key generating bits
VISA	SESS-XOR	DKYGENKY	0	Must be '0001' (keyword DDATA).
MC	MasterCard M/Chip 2.1	DKYGENKY	1	Must be '0001' (keyword DDATA).
EMV	EMV key derivation	DKYGENKY	0	Must be '0001' (keyword DDATA).

Table 17. Key type requirements for actions SMCONPIN, SMCIPIN, and VISAPIN

Key mode keyword	Key derivation used	Key type	Subtype	Key generating bits
VISA	SESS-XOR	DKYGENKY	0	Must be '1001' (keyword DMPIN).
MC	MasterCard M/Chip 2.1	DKYGENKY	1	Must be '1001' (keyword DMPIN).
EMV	EMV key derivation	DKYGENKY	0	Must be '1001' (keyword DMPIN).

new_PIN_encrypting_key_identifier_length

Direction	Type
Input	Integer

Specifies the length of the *new_PIN_encrypting_key_identifier* parameter in bytes. The value must be 0 or 64. When action keyword VISAPIN is specified, the value must be 64. Otherwise, the value must be 0 and the *new_PIN_encrypting_key_identifier* parameter is ignored.

new_PIN_encrypting_key_identifier

Direction	Type
Input/Output	String

The 64-byte DES key identifier (either an internal token or key label) for the key that encrypts the new PIN block. The key type can be either IPINENC or OPINENC. There are separate Cryptographic Coprocessor access points required based on the rules selected. For additional information, see "Cryptographic services used by EMV Scripting Service" on page 42 and "Access control points" on page 42.

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

current_PIN_encrypting_key_identifier_length

Direction	Type
Input	Integer

Specifies the length of the *current_PIN_encrypting_key_identifier* parameter in bytes. The value must be 0 or 64. When action keyword SMCONPIN, SMCIPIN, or VISAPIN is specified, the value must be 64. Otherwise, the value must be 0 and the *current_PIN_encrypting_key_identifier* parameter is ignored.

current_PIN_encrypting_key_identifier

Direction	Type
Input/Output	String

The 64-byte DES key identifier (either an internal token or key label) for the key that encrypts the PIN block with the current PIN.

For actions SMCONPIN and SMCIPIN, the key type must be IPINENC. For action VISAPIN, the key type can be either IPINENC or OPINENC.

There are separate Cryptographic Coprocessor access points required depending on the rules selected. For additional information, see "Cryptographic services used by EMV Scripting Service" on page 42 and "Access control points" on page 42.

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

new_PIN_block

Direction	Type
Input	String

The 8-byte PIN block encrypted by the *new_PIN_encrypting_key_identifier*.

For VISAPIN, this is the new PIN. All PIN block formats are supported. This parameter is ignored for all other action keywords.

current_PIN_block

Direction	Type
Input	String

The 8-byte PIN block for the current PIN encrypted by the *current_PIN_encrypting_key_identifier*.

When VISAPIN is specified, this is the current PIN. All Pin block formats are supported.

When SMCONPIN or SMCIPIN is specified, PIN block formats supported are ISO-0, ISO-1, and ISO-2.

When the action keyword is SMCON, SMCONINT, or SMINT, this parameter is ignored.

pan_length

Direction	Type
Input	Integer

Length in bytes of the *pan* parameter. The value must be 10.

pan

Direction	Type
Input	String

The 10-byte EMV card's Primary Account Number. The data must be in compressed numeric format and right justified in a 10-byte field, padded to the left with zeroes. For example, PAN 1234567890 must be provided as x'0000000001234567890'.

This data is used in combination with the PAN sequence number to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

pan_seq_number

Direction	Type
Input	String

The 1-byte sequence number of the EMV card's Primary Account Number. If the APPANSEQ control flag rule array keyword was specified, this PAN sequence number is used in combination with the PAN to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

atc

Direction	Type
Input	String

The 2-byte application transaction counter that is used for session key derivation. See the key mode rules for more information on session key derivation.

This parameter must be 2 bytes.

Note: The first byte is the high-order byte and the second byte is the low order byte.

unpredicatable_number

Direction	Type
Input	String

The 8-byte random number for secure messaging with MasterCard M/CHIP 2.1 processing. Only used when key mode is MC. Otherwise, this parameter is ignored.

input_message_length

Direction	Type
Input	Integer

The length of the message supplied in the *input_message* parameter.

This value must be between 8 and 320, inclusive. For key mode rule VISA, the value must be between 8 and 255, inclusive.

input_message

Direction	Type
Input	String

The message to be secured. Padding is as follows:

For action VISAPIN:
To the left one byte that contains the length of the PIN block (which is 8 bytes) and with an '80' byte followed by a number of '00' bytes until the length is 16 bytes. The output PIN block is 16 bytes for this action. Encryption is done in ECB mode.

For scripting with confidentiality:
The message is only padded if it is not a multiple of eight bytes. EMV padding is used.

For scripting with integrity:
The message is only padded if it is not a multiple of eight bytes. EMV padding is used.

The formatting of the message to the EMV card is the responsibility of the application.

PIN_offset

Direction	Type
Input	Integer

The offset in the message to be secured where the PIN block is to be placed. The first position has offset 0.

This parameter is ignored when the VISAPIN, SMINT, and SMCON keywords are specified.

PIN_format

Direction	Type
Input	String

The 17 bytes consists of two 8 byte fields for input and output PIN block format respectively and a 1 byte for the PAD digit, in case a PAD digit is needed for one of the PIN formats.

The following PIN formats are supported:

For actions SMCONPIN and SMCIPIN:
ISO-0, ISO-1, and ISO-2. Both 8 byte blocks must be fully specified.

For action VISAPIN:
All PIN formats are supported. See *ICSF Application Programmer's Guide*. Only the first 8 bytes (and maybe the PAD digit) are used for this action.

The pad digit must be specified only when specifying "3624" for the PIN format.

This parameter is ignored when the SMINT and SMCON keywords are specified.

output_message_length

Direction	Type
Input	Integer

On input, the length of the buffer to receive the processed message. The value must be at least as long as the input message plus any padding.

On output, the actual length of the message returned in the *output_message_length* parameter.

This parameter is ignored when keyword SMINT is specified.

output_message

Direction	Type
Input	String

The encrypted message when the action keyword is SMCON, SMCONPIN, SMCONINT, SMCIPIN, or VISAPIN.

The formatting of the message to the EMV card is the responsibility of the application.

mac_length

Direction	Type
Input	Integer

The number of bytes of the MAC that are to be returned in the *mac* parameter. The MAC is returned for secure messaging with integrity. Values 4, 6, and 8 are supported.

This parameter is ignored when keyword SMCOM, SMCONPIN, or VISAPIN is specified.

mac

Direction	Type
Input	String

The MAC value that is calculated for the actions SMINT, SMCONINT, and SMCIPIN.

Note that for actions SMCONINT and SMCIPIN, the result of the integrity protection, the MAC, is output in this parameter and the result of the encryption is in the output message parameter.

reserved1_length

Direction	Type
Input	Integer

Length in bytes of the *reserved1* parameter. The value must be 0.

reserved1

Direction	Type
Input	String

This field is ignored.

reserved2_length

Direction	Type
Input	Integer

Length in bytes of the *reserved2* parameter. The value must be 0.

reserved2

Direction	Type
Input	String

This field is ignored.

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.

Cryptographic services used by EMV Scripting Service

The following CCA cryptographic services are used by EMV Scripting Service:

For action SMINT:

- CSNBKTB - Key Token Build
- CSNBDKG - Diversified Key Generate
- CSNBMGN - MAC Generate

For action SMCON:

- CSNBKTB - Key Token Build
- CSNBDKG - Diversified Key Generate
- CSNBENC - Encipher

For action SMCONPIN:

- CSNBKTB - Key Token Build
- CSNBDKG - Diversified Key Generate
- CSNBSPN - Secure Message for PINs

For action SMCONINT:

- CSNBKTB - Key Token Build
- CSNBDKG - Diversified Key Generate
- CSNBMGN - MAC Generate
- CSNBENC - Encipher

For action SMCIPIN:

- CSNBKTB - Key Token Build
- CSNBDKG - Diversified Key Generate
- CSNBMGN - MAC Generate
- CSNBSPN - Secure Message for PINs

For action VISAPIN:

- CSNBKTB - Key Token Build
- CSNBPCU - VISA PIN change/unblock

The caller does not require authorization to each of these services, only to the EMV Scripting Service. Additionally, the caller must have the required access control points enabled (see information below about required access control points).

Access control points

The following access control points must be enabled to use the EMV Scripting Service:

For action SMINT:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-XOR
- Diversified Key Generate - SESS-XOR
- Diversified Key Generate - TDESEMV2/TDESEMV4
- MAC Generate

For action SMCON:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-XOR
- Diversified Key Generate - SESS-XOR

- Diversified Key Generate - TDESEMV2/TDESEMV4
- Encipher - DES

For action SMCONPIN:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-XOR
- Diversified Key Generate - SESS-XOR
- Diversified Key Generate - TDESEMV2/TDESEMV4
- Secure Messaging for PINs

For action SMCONINT:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-XOR
- Diversified Key Generate - TDESEMV2/TDESEMV4
- MAC Generate
- Encipher - DES

For action SMCIPIN:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-XOR
- Diversified Key Generate - TDESEMV2/TDESEMV4
- MAC Generate
- Secure Messaging for PINs

For action VISAPIN:

- PIN change/unblock - change EMV PIN with OPINENC
- PIN change/unblock - change EMV PIN with IPINENC

Required hardware

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

Table 18. EMV Scripting Service required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor Crypto Express2 Coprocessor	
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	

Table 18. EMV Scripting Service required hardware (continued)

Server	Required cryptographic hardware	Restrictions
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	
IBM z13	Crypto Express5 CCA Coprocessor	

EMV Transaction (ARQC/ARPC) Service (CSNBEAC and CSNEEAC)

The EMV Transaction (ARQC/ARPC) Service simplifies EMV Authorization Request Cryptogram (ARQC) and Authorization Response Cryptogram (ARPC) transaction processing. An ARQC is generated by the EMV card upon request from the point of sales terminal to obtain authorization for payment. The ARQC is forwarded across the payment network to the issuer for verification. After the issuer has verified the ARQC, the issuer generates an ARPC (the response). The ARPC is sent back through the payment network to the point of sales terminal to authorize the transaction.

The EMV Transaction (ARQC/ARPC) Service performs the following EMV functions:

- Verifying the Authorization Request Cryptogram (ARQC).
- Generating the Authorization Response Cryptogram (ARPC).
- Both verifying the ARQC and generating the ARPC.

This service can be used in the following specific brand modes:

- Visa
- MasterCard
- EMV

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

Format

```
CALL CSNBEAC(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             rule_array_count,
             rule_array,
             issuer_master_key_identifier_length,
             issuer_master_key_identifier,
             issuer_ARPC_master_key_identifier_length,
             issuer_ARPC_master_key_identifier,
             pan_length,
             pan,
             pan_seq_number,
             cryptogram_info_length,
             cryptogram_info,
             atc,
             arc,
             argc,
             arpc,
```

```
                unpredictable_number,
                reserved1_length,
                reserved1,
                reserved2_length,
                reserved2)
```

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 assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

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

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The minimum value is 3 and the maximum value is 4.

rule_array

Direction	Type
Input	String

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 19. Rule array keywords for EMV Transaction (ARQC/ARPC) Service

Keyword	Meaning
Algorithm (Required)	
TDES	Specifies the use of Triple-DES.
Action (One required)	
VERARQC	Specifies to verify the input Authorization Request Cryptogram.
GENARPC	Specifies to generate the Authorization Response Cryptogram from the input Authorization Request Cryptogram and Authorization Response Code (ARC).
VERGEN	Specifies to both verify the Authorization Request Cryptogram and generate the Authorization Response Cryptogram.
Key mode (One required). Defines the key derivation mechanism.	
VISA	Specifies to use the Visa Cryptogram Version 10 key derivation. The card's master key for application cryptograms is used as the session key (the keys are the same for each session). See Visa specification, Appendix D2. Padding is with binary zeroes until the length is a multiple of 8 bytes.
MC	Specifies to use the MasterCard M/CHIP 2.1 key derivation. The ATC and an unpredictable number is 3DES encrypted with the card's master key. The card's master key is used when generating the ARPC. EMV padding rules apply.
EMV	Specifies to use the session key derivation as described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.3. Use this key mode for Visa Cryptogram Version 14 and MasterCard M/CHIP 4. EMV padding rules apply.
Control flag (Optional)	
APPANSEQ	Specifies to append the PAN sequence number when the card specific master key is derived. See the descriptions of *pan* and *pan_seq_number*. The default is not to append the PAN sequence number.
Branch Factor (One optional, valid only with key mode EMV). The branching factor is to be used in EMV session key derivation.	
TDESEMV2	Specifies a branch factor of 2 for a height of 16. This is the default.
TDESEMV4	Specifies a branch factor of 4 for a height of 8.

issuer_master_key_identifier_length

Direction	Type
Input	Integer

Specifies the length of the *issuer_master_key_identifier* parameter in bytes. The value must be 64.

issuer_master_key_identifier

Direction	Type
Input/Output	String

A 64-byte DES key identifier (either an internal token or key label) for the issuer master key for Application Cryptograms (AC). The issuer master key is the DES key from which the card specific keys are derived and from the card specific keys, the session keys for application cryptograms are derived.

When using the key mode of VISA or EMV, this key is used for both the verification of the ARQC (VERARQC and VERGEN) and the generation of the ARPC (GENARPC and VERGEN).

When using the key mode of MC, this key is used only for the verification of the ARQC (VERARQC and VERGEN). The *issuer_ARPC_master_key_identifier* must supply the ARPC generation key (GENARPC and VERGEN).

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

Table 20. EMV Transaction (ARQC/ARPC) Service: Key requirements

Key mode keyword	Key type	Subtype	Key usage attributes
VISA	DKYGENKY	DKYL0	Must specify a MAC key will be derived (keyword DMAC).
MC	DKYGENKY	DKYL1	Must specify a MAC key will be derived (keyword DMAC).
EMV	DKYGENKY	DKYL0	Must specify a MAC key will be derived (keyword DMAC).

Note: For MasterCard M/Chip 2.1, you need the issuer master key in two versions: one of each subtype (DKYL1 and DKYL0). If action VERARQC or VERGEN is specified, this key must be the subtype DKYL1 and is used to derive the session key for ARQC verification. The key to be used for ARPC generation (GENARPC and VERGEN) must be specified in the *issuer_ARPC_master_key_identifier* input parameter.

issuer_ARPC_master_key_identifier_length

Direction	Type
Input	Integer

This parameter specifies the length of the *issuer_ARPC_master_key_identifier* parameter in bytes. When the key mode keyword MC is specified, the value must be 64. Otherwise, the value must be zero.

issuer_ARPC_master_key_identifier

Direction	Type
Input/Output	String

The 64-byte CCA DES key identifier (either an internal token or key label) for the issuer master key for Application Response Cryptograms (ARPC) when using the MC key mode. The issuer ARPC master key is the DES key from which a session key for ARPC generation is derived.

Only used when action is GENARPC or VERGEN and key mode is MC, where this key is the issuer master key to be used for deriving the key to use for ARPC generation.

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

Key Type Requirements: The key type must be DKYGENKY, the subtype must be DKYL0, and the key usage must specify that a MAC key will be derived (keyword DMAC).

Note: For MasterCard M/Chip 2.1, you need the issuer master key in two versions: one of each subtype (DKYL1 and DKYL0). If the action GENARPC or VERGEN is specified, this key must be the subtype DKYL0 and is used to derive the session key for ARPC generation. The key to be used for ARQC verification (VERARQC and VERGEN) must be specified in the *issuer_master_key_identifier* input parameter.

pan_length

Direction	Type
Input	Integer

Length in bytes of the *pan* parameter. The value must be 10.

pan

Direction	Type
Input	String

The 10-byte EMV card's Primary Account Number. The data must be in compressed numeric format and right justified in a 10-byte field, padded to the left with zeroes. For example, PAN 1234567890 must be provided as x'0000000001234567890'.

This data is used in combination with the PAN sequence number to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

pan_seq_number

Direction	Type
Input	String

The 1-byte sequence number of the EMV card's Primary Account Number. If the APPANSEQ control flag rule array keyword was specified, this PAN sequence number is used in combination with the PAN to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

cryptogram_info_length

Direction	Type
Input	Integer

The length of the cryptogram information supplied in the *cryptogram_info* parameter. This value must be between 1 and 252 inclusive.

cryptogram_info

Direction	Type
Input	String

The cryptogram information on which the ARQC is generated. The data must not be padded.

The cryptogram information is padded as follows:

Key derivation mechanism	Padding
VISA	Visa ICC Card's Specification V1.4.0, Appendix D2 and D3
MC, EMV	EMV, Book 2, Annex A1.2

atc

Direction	Type
Input	String

The 2-byte application transaction counter that is used for session key derivation. See the key mode rules for more information on session key derivation.

This parameter must be 2 bytes.

Note: The first byte is the high-order byte and the second byte is the low order byte.

arc

Direction	Type
Input	String

The 2-byte authorization response code that must be used for generating the ARPC. Only used if action GENARPC or VERGEN is specified.

This parameter must be 2 bytes.

Note: The first byte is the high-order byte and the second byte is the low order byte.

arqc

Direction	Type
Input	String

The 8-byte authorization request cryptogram received from the payment card.

This parameter must be 8 bytes.

arpc

Direction	Type
Output	String

The 8-byte authorization response cryptogram to be sent back to the payment card. The ARPC is obtained by enciphering the ARQC XOR-ed with the ARC (See also EMV, Book 2, 8.2).

This parameter must be 8 bytes.

unpredictable_number

Direction	Type
Input	String

The 4-byte unpredictable number used in the MasterCard M/Chip 2.1 session key derivation scheme.

The data in this field will not be reformatted by the API before use.

This parameter must be 4 bytes.

reserved1_length

Direction	Type
Input	Integer

Length in bytes of the *reserved1* parameter. The value must be 0.

reserved1

Direction	Type
Input	String

This field is ignored.

reserved2_length

Direction	Type
Input	Integer

Length in bytes of the *reserved2* parameter. The value must be 0.

reserved2

Direction	Type
Input	String

This field is ignored.

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.

Cryptographic services used by EMV Transaction (ARQC/ARPC) Service

The following CCA cryptographic services are used by EMV Transaction (ARQC/ARPC) Service:

- CSNBKTB - Key Token Build
- CSNBDKG - Diversified Key Generate
- CSNBMGN - MAC Generate
- CSNBMVR - MAC Verify

The caller does not require authorization to each of these services, only to the EMV Transaction (ARQC/ARPC) Service. Additionally, the caller must have the required access control points enabled (see information below about required access control points).

Access control points

The following access control points must be enabled to use the EMV Transaction (ARQC/ARPC) Service:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-XOR
- Diversified Key Generate - TDESEMV2/TDESEMV4
- MAC Generate
- MAC Verify

Required hardware

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

Table 21. EMV Transaction (ARQC/ARPC) Service required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor Crypto Express2 Coprocessor	
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	
IBM z13	Crypto Express5 CCA Coprocessor	

EMV Verification Functions (CSNBEVF and CSNEEVF)

Provides additional functions used by MasterCard for their EMV cards in addition to application cryptograms and scripting.

This service performs the following EMV scripting functions:

- Verification of data authentication codes.
- Verification of dynamic numbers.
- Decryption of encrypted counters.

This service can be used in the following specific brand modes:

- MasterCard
- EMV

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

Format

```
CALL CSNBEVF(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             rule_array_count,
             rule_array,
             key_identifier_length,
             key_identifier,
             pan_length,
             pan,
             pan_seq_number,
             atc,
             unpredictable_number,
             data_length,
             data,
             reserved1_length,
             reserved1,
             reserved2_length,
             reserved2)
```

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 assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

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

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The minimum value is 3 and the maximum value is 5.

rule_array

Direction	Type
Input	String

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 22. Rule array keywords for EMV Verification Functions

Keyword	Meaning
Algorithm (Required)	
TDES	Specifies the use of Triple-DES.
Function to be performed (One required)	
DYNVER	Specifies to verify the dynamic number.
DACVER	Specifies to verify the data authentication code.
DECCNT	Specifies to decrypt the encrypted counters.
Key mode (One required). Defines the key derivation mechanism.	
MC	Specifies to use the MasterCard M/CHIP 2.1 mode. The ATC and an unpredictable number is encrypted with the card's master key. The card's master key is used when generating the ARPC. Padding is according to EMV.
EMV	Specifies to use the session key derivation as described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.3. Use this key mode for Visa Cryptogram Version 14 and MasterCard M/CHIP 4. EMV padding rules apply.
Control flag (Optional)	
APPANSEQ	Specifies to append the PAN sequence number when the card specific master key is derived. See the descriptions of *pan* and *pan_seq_number*. The default is not to append the PAN sequence number.
Branch Factor (One optional, valid only with key mode EMV). The branching factor is to be used in EMV session key derivation.	
TDESEMV2	Specifies a branch factor of 2 for a height of 16. This is the default.
TDESEMV4	Specifies a branch factor of 4 for a height of 8.

key_identifier_length

Direction	Type
Input	Integer

Specifies the length of the *key_identifier* parameter in bytes. The value must be 64.

key_identifier

Direction	Type
Input/Output	String

The 64-byte CCA DES key identifier (either an internal token or key label) of the key to perform the specified function.

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

Table 23. EMV Verification Functions: Key requirements

Function	Key mode	Key type	Subtype	Key usage
DACVER	N/A	Must be a double-length DATA session key.	N/A	N/A
DECCNT	MC	DKYGENKY	DKYL1	Must specify a DATA key to be derived (DDATA).
DECCNT	EMV	DKYGENKY	DKYL0	Must specify a DATA key to be derived (DDATA).
DYNVER	N/A	DKYGENKY	DKYL0	Must specify a DATA key to be derived (DDATA).

pan_length

Direction	Type
Input	Integer

Length in bytes of the *pan* parameter. The value must be 10.

pan

Direction	Type
Input	String

The 10-byte EMV card's Primary Account Number. The data must be in compressed numeric format and right justified in a 10-byte field, padded to the left with zeroes. For example, PAN 1234567890 must be provided as x'00000000001234567890'.

This data is used in combination with the PAN sequence number to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

pan_seq_number

Direction	Type
Input	String

The 1-byte sequence number of the EMV card's Primary Account Number. If the APPANSEQ control flag rule array keyword was specified, this PAN sequence number is used in combination with the PAN to derive the card's master key. The exact set of rules is described in EMV Integrated Circuit Card Specification for Payment Systems Version 4.2 (EMV4.2) Book 2, Annex A1.4.

If the APPANSEQ keyword was not specified, the PAN sequence number is not appended to the PAN when deriving the card's specific master key.

atc

Direction	Type
Input	String

The 2-byte application transaction counter that is used for session key derivation. See the key mode rules for more information on session key derivation.

This parameter must be 2 bytes.

Note: The first byte is the high-order byte and the second byte is the low order byte.

unpredicatable_number

Direction	Type
Input	String

The 4-byte unpredictable number used in the MasterCard M/Chip 2.1 session key derivation scheme.

The data in this field will not be reformatted before use.

This parameter must be 4 bytes.

data_length

Direction	Type
Input/Output	Integer

Specifies the length of the DATA parameter, in bytes. The value must be 2 for function DACVER and 8 for functions DYNVER and DECCNT.

data

Direction	Type
Input/Output	String

For function DACVER:

The two leftmost bytes are used from the input as the DAC to be verified. On output, the field contains the correct DAC in the case of a mismatch.

For function DYNVER:

All 8 bytes are used. On input, this is the dynamic number to be verified. On output, it contains the correct dynamic number in the case of a mismatch.

For function DECCNT:

All 8 bytes are used. On input, this is the 8-byte encrypted counters. On output, the decrypted counters are returned.

reserved1_length

Direction	Type
Input	Integer

Length in bytes of the *reserved1* parameter. The value must be 0.

reserved1

Direction	Type
Input	String

This field is ignored.

reserved2_length

Direction	Type
Input	Integer

Length in bytes of the *reserved2* parameter. The value must be 0.

reserved2

Direction	Type
Input	String

This field is ignored.

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.

Cryptographic services used by EMV Verification Functions

The following CCA cryptographic services are used by EMV Verification Functions:

For action DACVER:

- CSNBENC - Encipher

For action DYNVER:

- CSNBKTB - Key Token Build
- CSNBDKG - Diversified Key Generate
- CSNBENC - Encipher

For action DECCNT:

- CSNBKTB - Key Token Build
- CSNBDKG - Diversified Key Generate
- CSNBDEC - Decipher

The caller does not require authorization to each of these services, only to the EMV Verification Functions. Additionally, the caller must have the required access control points enabled (see information below about required access control points).

Access control points

The following access control points must be enabled to use the EMV Verification Functions:

For action DACVER:

- Encipher - DES

For action DYNVER:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - SESS-XOR
- Encipher - DES

For action DECCNT:

- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - SESS-XOR
- Diversified Key Generate - TDESEMV2/TDESEMV4
- Decipher - DES

Required hardware

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

Table 24. EMV Verification Functions required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890	PCI X Cryptographic Coprocessor Crypto Express2 Coprocessor	
IBM System z9 EC IBM System z9 BC	Crypto Express2 Coprocessor	
IBM System z10 EC IBM System z10 BC	Crypto Express2 Coprocessor Crypto Express3 Coprocessor	
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	
IBM z13	Crypto Express5 CCA Coprocessor	

Reason codes for return code 4 (4)

Table 25. Reason codes for return code 4 (4)

Reason Code Hex (Decimal)	Description
D41 (3393)	The ARQC could not be verified. **User action**: Ensure that the correct cryptogram information was passed, the correct key mode was specified, and the correct issuer master key was used.
D45 (3397)	Failure to verify the data authentication code. **User action**: Ensure that the correct PAN and PAN sequence number were passed and the correct issuer master key was used.
D46 (3398)	Failure to verify the dynamic number. **User action**: Ensure that the correct ATC was passed and the correct issuer master key was used.

Reason codes for return code 8 (8)

Table 26. Reason codes for return code 8 (8)

Reason Code Hex (Decimal)	Description
D42 (3394)	ARPC generation failed. **User action:** Ensure that the correct key mode was specified and the issuer master key being used is valid for ARPC generation.
D43 (3395)	Secure messaging with integrity failure. **User action:** Ensure that the correct key mode was specified and the issuer master key being used is valid for EMV scripting with integrity.
D44 (3396)	Secure messaging with confidentiality failure. **User action:** Ensure that the correct key mode was specified and the issuer master key being used is valid for EMV scripting with confidentiality.
D47 (3399)	Failure to decrypt the encrypted counter. **User action:** Ensure that a valid encrypted counter was passed and the correct issuer master key was used.

Visa, MasterCard, and EMV-related smart card formats and processes

The VISA, MasterCard, and EMV specifications for performing secure messaging with an EMV compliant smart card are covered in these documents:

- *EMV 2000 Integrated Circuit Card Specification for Payment Systems Version 4.0 (EMV4.0) Book 2*
- *Design Visa Integrated Circuit Card Specification Manual*
- *Integrated Circuit Card Specification (VIS) 1.4.0 Corrections*
- *MasterCard International: M/Chip 4 Security & Key Management Version 1.0*

M/Chip 4 Security & Key Management describes how a smart-card, card-specific session key is derived from a card-issuer-supplied master key.

Access control points and callable services

If an access control point is disabled, the corresponding ICSF callable service will fail during execution with an access denied error.

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.

Table 27. Access control points affecting multiple services or requiring special consideration

Name	Callable Services	Notes	Usage
NOCV KEK usage for export-related functions	CSNBGIM / CSNEGIM, CSNBKEX / CSNEKEX, CSNBSKM / CSNESKM, and CSNBKGN / CSNEKGN	When enabled, NOCV key-encrypting keys can be used by the listed services.	ED, SC

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

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

Table 28. Access control points – Callable Services

Name	Callable Service	Usage
Decipher - DES	CSNBDEC / CSNEDEC and CSNBEVF / CSNEEVF	ED
Diversified Key Generate - SESS-XOR	CSNBDKG / CSNEDKG, CSNBESC / CSNEESC, and CSNBEVF / CSNEEVF	ED
Diversified Key Generate - TDES-ENC	CSNBDCM / CSNEDCM, CSNBDKG / CSNEDKG, CSNBDSK / CSNEDSK, CSNBEAC / CSNEEAC, CSNBESC / CSNEESC, and CSNBEVF / CSNEEVF	ED
Diversified Key Generate - TDES-XOR	CSNBDCM / CSNEDCM, CSNBDKG / CSNEDKG, CSNBDSK / CSNEDSK, CSNBEAC / CSNEEAC, and CSNBESC / CSNEESC	ED
Diversified Key Generate - TDESEMV2/TDESEMV4	CSNBDCM / CSNEDCM, CSNBDKG / CSNEDKG, CSNBDSK / CSNEDSK, CSNBEAC / CSNEEAC, CSNBESC / CSNEESC, and CSNBEVF / CSNEEVF	ED
Encipher - DES	CSNBENC / CSNEENC, CSNBESC / CSNEESC, and CSNBEVF / CSNEEVF	ED
Key Generate – OP	CSNBGIM / CSNEGIM and CSNBKGN / CSNEKGN	ED
MAC Generate	CSNBEAC / CSNEEAC, CSNBESC / CSNEESC, and CSNBMGN / CSNEMGN	ED
MAC Verify	CSNBEAC / CSNEEAC and CSNBMVR / CSNEMVR	ED
PIN Change/Unblock - change EMV PIN with OPINENC	CSNBESC / CSNEESC and CSNBPCU / CSNEPCU	ED

Table 28. Access control points – Callable Services (continued)

Name	Callable Service	Usage
PIN Change/Unblock - change EMV PIN with IPINENC	CSNBESC / CSNEESC and CSNBPCU / CSNEPCU	ED
Secure Messaging for PINs	CSNBESC / CSNEESC and CSNBSPN / CSNESPN	ED

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

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

Setting up profiles in the CSFSERV general resource class

Table 29. Resource names for ICSF callable services

Resource name	Callable service names	Callable service description
CSFDCM	CSNBDCM CSNEDCM	Derive ICC MK
CSFDSK	CSNBDSK CSNEDSK	Derive Session Key
CSFEAC	CSNBEAC CSNEEAC	EMV Transaction Service
CSFESC	CSNBESC CSNEESC	EMV Scripting Service
CSFEVF	CSNBEVF CSNEEVF	EMV Verification Functions
CSFGIM	CSNBGIM CSNEGIM	Generate Issuer MK

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 30. Callable services and parameters affected by key store policy

ICSF callable service	31-bit name	Parameter checked
Derive ICC MK	CSNBDCM	issuer_master_key_identifier transport_key_identifier
Derive Session Key	CSNBDSK	master_key_identifier **Note:** ICC master keys derived from issuer master keys are affected by key store policy before they are used to derive session keys.

Table 30. Callable services and parameters affected by key store policy (continued)

ICSF callable service	31-bit name	Parameter checked
EMV Scripting Service	CSNBESC	issuer_integrity_master_key_identifier issuer_confidentiality_master_key_identifier new_PIN_encrypting_key_identifier current_PIN_encrypting_key_identifier **Note:** ICC master keys derived from issuer master keys are affected by key store policy before they are used to derive session keys.
EMV Transaction Service	CSNBEAC	issuer_master_key_identifier issuer_ARPC_master_key_identifier **Note:** ICC master keys derived from issuer master keys are affected by key store policy before they are used to derive session keys.
EMV Verification Functions	CSNBEVF	key_identifier **Note:** ICC master keys derived from issuer master keys are affected by key store policy before they are used to derive session keys.
Generate Issuer MK	CSNBGIM	transport_key_identifier

Chapter 5. Update of z/OS Cryptographic Services ICSF System Programmer's Guide, SC14-7507-03, information

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

Parameters in the installation options data set

The installation options data set is an intended programming interface.

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 31 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 31. Exit identifiers and exit invocations

Exit identifiers	Exit invocations
CSFDCM	Gets control during the Derive ICC MK callable service.
CSFDSK	Gets control during the Derive Session Key callable service.
CSFEAC	Gets control during the EMV Transaction Service callable service.
CSFESC	Gets control during the EMV Scripting Service callable service.

Table 31. Exit identifiers and exit invocations (continued)

Exit identifiers	Exit invocations
CSFEVF	Gets control during the EMV Verification Functions callable service.
CSFGIM	Gets control during the Generate Issuer MK callable service.

Callable services

Table 32. Summary of new and changed ICSF callable services

Callable service	Release	Description
Derive ICC MK	HCR77A0	**New:** This service generates an ICC master key from an issuer master key.
Derive Session Key	HCR77A0	**New:** Use this callable service to derive a session key from either an issuer master key or an ICC master key.
EMV Scripting Service	HCR77A0	**New:** This service simplifies EMV Scripting.
EMV Transaction Service	HCR77A0	**New:** This service simplifies EMV Authorization Request Cryptogram (ARQC) and Authorization Response Cryptogram (ARPC) transaction processing.
EMV Verification Functions	HCR77A0	**New:** This service provides additional functions used by MasterCard for their EMV cards in addition to application cryptograms and scripting.
Generate Issuer MK	HCR77A0	**New:** This callable service helps with the initial steps of EMV setup by generating and storing the issuer master keys.

CICS attachment facility

If you have the CICS Attachment Facility 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.

Modify and include:

- HCR77A0: CSFDCM, CSFDSK, CSFEAC, CSFESC, CSFGIM, CSFEVF

Printed in USA