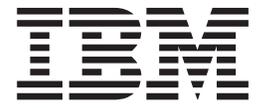


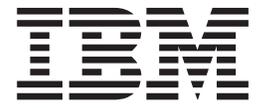
z/OS  
Cryptographic Services  
Integrated Cryptographic Service Facility



# Overview



z/OS  
Cryptographic Services  
Integrated Cryptographic Service Facility



# Overview

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 85.

**Fifth Edition (September 2003)**

This is a major revision of SA22-7519-03.

This edition applies to Version 1 Release 4 of z/OS (5694-A01) and Version 1 Release 4 of z/OS.e (5655-G52), and all subsequent releases and modifications until otherwise indicated in new editions.

IBM® welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation  
Department 55JA, Mail Station P384  
2455 South Road  
Poughkeepsie, NY 12601-5400  
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)

Internet e-mail: mhvrfs@us.ibm.com

World Wide Web: <http://www.ibm.com/servers/eserver/zseries/zos/webqs.html>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1996, 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Figures</b> . . . . .	vii
<b>Tables</b> . . . . .	ix
<b>About This document</b> . . . . .	xi
ICSF Features . . . . .	xi
Who Should Use This document . . . . .	xii
How to Use This document . . . . .	xii
Where to Find More Information . . . . .	xii
The ICSF Library . . . . .	xiii
Related Publications . . . . .	xv
Information on Other IBM Cryptographic Products . . . . .	xv
Using LookAt to look up message explanations . . . . .	xv
Accessing z/OS™ licensed documents on the Internet . . . . .	xvi
<b>Summary of changes</b> . . . . .	xvii
<b>Chapter 1. Introducing Cryptography and ICSF</b> . . . . .	1
What Is Cryptography? . . . . .	1
The Basic Elements of a Cryptographic System . . . . .	2
How Does ICSF Support Cryptography? . . . . .	4
How Does ICSF Extend the Uses of Cryptography? . . . . .	5
Key Generation and Distribution . . . . .	5
Personal Identification Numbers (PINs) . . . . .	5
Message Authentication Codes (MACs) . . . . .	5
Hashing Algorithms . . . . .	6
Digital Signatures . . . . .	6
Card-Verification Values . . . . .	7
Translation of Data and PINs in Networks . . . . .	7
SET Secure Electronic Transaction . . . . .	7
Secure Sockets Layer (SSL) . . . . .	8
EMV Integrated Circuit Card Specifications . . . . .	8
<b>Chapter 2. Solving Your Business Needs with ICSF</b> . . . . .	9
Keeping Your Data Private . . . . .	9
Transporting Data Securely Across a Network . . . . .	10
Supporting the Internet Secure Sockets Layer Protocol . . . . .	11
Transacting Commerce on the Internet . . . . .	11
Exchanging Keys Safely Between Networks . . . . .	12
Exchanging Keys Using DES Callable Services . . . . .	12
Exchanging DES Data-encrypting Keys Using an RSA Key Scheme . . . . .	13
Managing Keys Using a Trusted Key Entry Workstation . . . . .	14
Using Personal Identification Numbers (PINs) for Personal Authentication . . . . .	15
Verifying Data Integrity and Authenticity . . . . .	15
Using Message Authentication Codes . . . . .	16
Generating and Verifying Digital Signatures . . . . .	16
Using Modification Detection Codes and Message Hashing . . . . .	16
Verifying Payment Card Data. . . . .	17
Maintaining Continuous Operations . . . . .	17
Reducing Costs by Improving Productivity . . . . .	18
Improving Cryptographic Performance . . . . .	18
Using RMF and SMF to Monitor z/OS ICSF Events . . . . .	19
Improving Performance in a CICS Environment . . . . .	19

Customizing ICSF to Meet Your Installation's Needs . . . . .	19
Using ICSF Exits to Meet Special Needs . . . . .	20
Creating Installation-Defined Callable Services . . . . .	20
Using Options to Tailor ICSF . . . . .	20
Isolating and Protecting PR/SM Partitions . . . . .	21
Enabling Growth . . . . .	21
Protecting Your Investment . . . . .	22
<b>Chapter 3. Callable Services and Key Management . . . . .</b>	<b>23</b>
Callable Services . . . . .	23
Protecting and Controlling DES Keys . . . . .	24
DES Master Key Variant . . . . .	25
DES Transport Key Variant . . . . .	25
DES Key Forms . . . . .	25
Control Vectors . . . . .	25
Types of DES Keys . . . . .	25
Protecting and Controlling PKA Keys . . . . .	27
PKA Master Keys . . . . .	28
RSA Private and Public Keys. . . . .	28
DSA Private and Public Keys. . . . .	29
Exchanging Encrypted Keys and PINs on a DES System . . . . .	29
Exchanging RSA-Encrypted Data Keys . . . . .	30
Using Multiple Encipherment to Protect Keys and Data . . . . .	30
Running in Special Secure Mode . . . . .	31
Cryptographic Key Data Set (CKDS) . . . . .	32
Dynamic CKDS Update Callable Services . . . . .	33
PKA Cryptographic Key Data Set . . . . .	33
Dynamic PKDS Update Callable Services . . . . .	34
Key Generator Utility Program and Key Generate Callable Service . . . . .	34
ANSI X9.17 Key Management Callable Services . . . . .	34
Composing and Decomposing SET Blocks. . . . .	35
Exchanging Secure Sockets Layer Session Key Seed . . . . .	35
<b>Chapter 4. Using ICSF with Other Cryptographic Products . . . . .</b>	<b>37</b>
Using IBM's Common Cryptographic Architecture . . . . .	37
Coexisting with Other IBM Cryptographic Products. . . . .	37
Running PCF Applications under ICSF . . . . .	37
Running 4753-HSP Applications under ICSF . . . . .	38
Managing Keys with the Distributed Key Management System (DKMS) . . . . .	39
Encrypting and Decrypting Information from Other Products . . . . .	39
Virtual Telecommunications Access Method (VTAM) Session-Level Encryption	40
Access Method Services Cryptographic Option . . . . .	40
Using ICSF with BSAFE . . . . .	40
<b>Chapter 5. Planning for the Integrated Cryptographic Service Facility . . . . .</b>	<b>43</b>
System Requirements . . . . .	43
Operating System . . . . .	43
Cryptographic Hardware Features . . . . .	43
Servers. . . . .	45
Configurations by Server . . . . .	46
Configuring the IBM @server zSeries 990 . . . . .	46
Configuring the S/390 Enterprise Servers, the S/390 Multiprise Server, the	
IBM @server zSeries 800 and the IBM @server zSeries 900 . . . . .	48
Hardware Features by Server . . . . .	50
Security . . . . .	51
Operating Considerations . . . . .	53

	ICSF Initialization Options . . . . .	54
	Effect of Multiple Records on Performance. . . . .	54
	LPAR Considerations . . . . .	54
	Migration Activities . . . . .	54
	Callable Services . . . . .	55
	CICS Attachment Facility . . . . .	56
	CKDS . . . . .	57
	Installation Options Data Set . . . . .	57
	Key Tokens . . . . .	58
	PCI Cryptographic Accelerator . . . . .	58
	PKDS . . . . .	58
	Resource Manager Interface (RMF) . . . . .	59
	Special Secure Mode . . . . .	59
	TKE Workstation . . . . .	59
	Migrating from 4753-HSP . . . . .	61
	<b>Appendix A. Standards . . . . .</b>	<b>65</b>
	<b>Appendix B. Summary of Callable Service Support by Hardware Configuration . . . . .</b>	<b>67</b>
	<b>Appendix C. IBM @server zSeries 990 with a PCI X Cryptographic Coprocessor . . . . .</b>	<b>73</b>
	Operating System Requirements . . . . .	73
	Applications and programs . . . . .	73
	Callable services . . . . .	73
	CKDS and PKDS (PCI X Cryptographic Coprocessor) . . . . .	76
	ICSF Setup and Initialization . . . . .	76
	Migration . . . . .	77
	Functions Not Supported . . . . .	77
	Setup Considerations . . . . .	77
	Programming Considerations. . . . .	77
	TKE workstation . . . . .	78
	Access Control Points . . . . .	78
	TSO panels . . . . .	79
	<b>Appendix D. IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor . . . . .</b>	<b>81</b>
	Operating System Requirements . . . . .	81
	Applications and programs . . . . .	81
	Callable services . . . . .	81
	ICSF Setup and Initialization . . . . .	82
	Secure Sockets Layer (SSL) . . . . .	82
	TKE workstation . . . . .	82
	<b>Appendix E. Accessibility . . . . .</b>	<b>83</b>
	Using assistive technologies . . . . .	83
	Keyboard navigation of the user interface . . . . .	83
	<b>Notices . . . . .</b>	<b>85</b>
	Trademarks . . . . .	86
	<b>Glossary . . . . .</b>	<b>89</b>
	<b>Index . . . . .</b>	<b>99</b>



---

## Figures

1.	The z/OS ICSF Library . . . . .	xiv
2.	Enciphering and Deciphering Data in a Secret Key System. . . . .	3
3.	An Example of Nonrepudiation Using Digital Signatures . . . . .	4
4.	Creating and Verifying Digital Signatures in a Public Key System . . . . .	7
5.	DES Encrypted Data Protected When Sent on Intermediate Systems . . . . .	10
6.	PKA Encrypted Data Protected When Sent on Intermediate Systems. . . . .	11
7.	Key Exchange in a DES Cryptographic System . . . . .	13
8.	Distributing a DES Data-encrypting Key Using an RSA Cryptographic Scheme . . . . .	14
9.	Using Transport Keys to Exchange Keys . . . . .	30
10.	An Example of Multiple Encipherment . . . . .	31
11.	How the Cryptographic Key Data Set Is Maintained and Used . . . . .	32
I 12.	Two Crypto PCICAs on a Processor Complex Running in LPAR Mode . . . . .	47
I 13.	Multiple Crypto Coprocessors on a Complex Running in LPAR Mode . . . . .	48
I 14.	Two Crypto CPs on a Processor Complex Running in Single Image Mode. . . . .	49
I 15.	Two Crypto Coprocessors and one PCICC on a Processor Complex Running in LPAR Mode . . . . .	50



---

## Tables

1.	Summary of new and changed ICSF callable services . . . . .	55
2.	Summary of ICSF Callable Services Support . . . . .	68
3.	Summary of new and changed ICSF callable services . . . . .	74



---

## About This document

This document supports z/OS (5694-A01) and z/OS.e (5655-G52). It contains overview and planning information for the z/OS Integrated Cryptographic Service Facility (ICSF). The z/OS Cryptographic Services includes the following components:

- z/OS Integrated Cryptographic Service Facility (ICSF)
- z/OS Open Cryptographic Services Facility (OCSF)
- z/OS System Secure Socket Level Programming (SSL)
- z/OS Public Key Infrastructure Services (PKI)

ICSF is a software element of z/OS that works with hardware cryptographic features and the Security Server (RACF) to provide secure, high-speed cryptographic services in the z/OS environment. ICSF provides the application programming interfaces by which applications request the cryptographic services. The cryptographic feature is secure, high-speed hardware that performs the actual cryptographic functions.

The cryptographic hardware features available to your applications depend on the server.

---

## ICSF Features

ICSF supports the ANSI Data Encryption Algorithm (DEA) for the encryption and decryption of data. DEA is also known as the U.S. National Institute of Science and Technology (NIST) Data Encryption Standard (DES) algorithm. In this publication, we use the term DES when referring to this algorithm.

ICSF also provides support for:

- Triple DES encryption for privacy.
- The transport of data keys through the use of the Rivest-Shamir-Adelman (RSA) public key algorithm.
- The generation and verification of digital signatures through the use of both the RSA and the Digital Signature (DSA) algorithm.
- The generation of RSA and DSA keys.
- The Secure Electronic Transaction (SET) standard which was created by VISA International and MasterCard.
- The PKA Encrypt and PKA Decrypt callable services that can be used to enhance the security and performance of Secure Sockets Layer (SSL) security protocol applications.
- Advanced Encryption Standard (AES) encryption and decryption.
- The Derived Unique Key Per Transaction (DUKPT or UKPT) algorithm.
- The EMV (Europay, MasterCard, VISA) Integrated Circuit Card Specifications

ICSF enhances security as follows:

- It ensures data privacy by encrypting and decrypting the data.
- It manages personal identification numbers (PINs).
- It ensures the integrity of data through the use of message authentication codes (MACs), modification detection codes (MDCs), hash functions, digital signatures, or VISA Card Verification Value.

- It ensures the privacy of cryptographic keys themselves by encrypting them under a master key or another key-encrypting key.
- It enforces DES key separation, which ensures that cryptographic keys are used only for their intended purposes.
- It enhances system availability by providing continuous operation.
- It enables the use of Rivest-Shamir-Adelman (RSA) and Digital Signature (DSA) public and private keys on a multi-user, multi-application platform.

---

## Who Should Use This document

This document is for chief information officers, information system executives, and information security professionals and auditors. Installation managers and security administrators who are responsible for planning the data security strategy for their installation will also find this document to be helpful. This publication applies to installations that have z/OS with ICSF and a hardware cryptographic feature installed.

---

## How to Use This document

The major topics by chapter are:

- Chapter 1, “Introducing Cryptography and ICSF” introduces the general subject of cryptography, and describes why ICSF may be right for your installation.
- Chapter 2, “Solving Your Business Needs with ICSF” describes how ICSF can help your business.
- Chapter 3, “Callable Services and Key Management” describes the cryptographic callable services available with ICSF and the basic concepts of managing cryptographic keys.
- Chapter 4, “Using ICSF with Other Cryptographic Products” describes how ICSF relates to other cryptographic products.
- Chapter 5, “Planning for the Integrated Cryptographic Service Facility” identifies the system facilities and system resources that ICSF requires and presents guidelines and suggestions to help you when you plan for installing, operating, and migrating ICSF.
- Appendix A, “Standards,” on page 65 provides a list of International and USA standards for the Cryptographic Coprocessor Feature, PCI Cryptographic Coprocessor, and ICSF.
- Appendix B, “Summary of Callable Service Support by Hardware Configuration,” on page 67 summarizes ICSF callable services by configuration.
- Appendix C, “IBM @server zSeries 990 with a PCI X Cryptographic Coprocessor,” on page 73 describes processing and functionality support in this environment.
- Appendix D, “IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor,” on page 81 describes processing and functionality support in this environment.
- Appendix E, “Accessibility,” on page 83 contains information on accessibility features in z/OS.
- “Notices” on page 85 contains notices and trademarks.

---

## Where to Find More Information

This section describes documents that contain ICSF information.

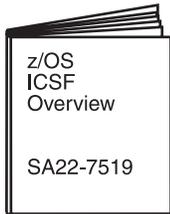
## The ICSF Library

Figure 1 on page xiv shows the ICSF library, which includes the following publications:

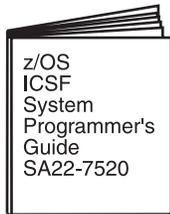
- *z/OS Cryptographic Services ICSF Overview, SA22-7519*  
This publication provides an introduction to ICSF, an overview of cryptography, and planning information.
- *z/OS Cryptographic Services ICSF Administrator's Guide, SA22-7521*  
See this document for information on managing cryptographic keys. It describes the tasks of entering DES and PKA master keys, changing a DES master key, using the key generator utility program, and viewing the status of the cryptographic feature.
- *z/OS Cryptographic Services ICSF System Programmer's Guide, SA22-7520*  
See this document for information on initialization, customization, migration, and problem diagnosis.
- *z/OS Cryptographic Services ICSF Application Programmer's Guide, SA22-7522*  
See this document for information on writing application programs that use the callable services that are provided by ICSF to access cryptographic functions. These callable services can be used in high-level languages such as C, COBOL, FORTRAN, and PL/I, as well as in Assembler.
- *z/OS Cryptographic Services ICSF Messages, SA22-7523*  
See this document for explanations of messages that are produced by ICSF, and for the routing and descriptor codes for those messages.
- *z/OS Cryptographic Services ICSF TKE Workstation User's Guide, SA22-7524*  
This document is available with the optional Trusted Key Entry (TKE) workstation and explains how to install and run the TKE workstation for key distribution (Version 3 or higher).

---

## Tasks



Evaluating  
Planning



Customizing  
Diagnosis  
Installing  
Operating



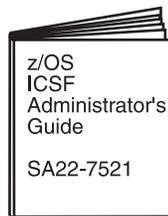
Application  
Programming

---

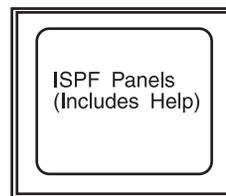
## Tasks



Administrating  
Application Programming  
Diagnosis  
Operating



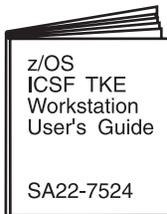
Administrating



Administrating

---

## Optional Features



Available with the  
Trusted Key Entry  
Workstation  
(TKE Version 4)



The ICSF Library and  
the Trusted Key Entry  
Workstation User's  
Guide are included on  
the IBM Online Library:  
z/OS Collection Kit  
SK3T-4269

Figure 1. The z/OS ICSF Library

The following publications contain additional ICSF information:

- *z/OS MVS System Codes*, SA22-7626  
This document describes reason codes for ICSF X'18F' abend code.
- *z/OS MVS System Management Facilities (SMF)*, SA22-7630  
This document describes SMF record type 82, where ICSF records events.
- *z/OS MVS Initialization and Tuning Guide*, SA22-7591
- *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- *z/OS MVS Programming: Callable Services for HLL*, SA22-7613
- *z/OS MVS Programming: Authorized Assembler Services Guide*, SA22-7608

- *z/OS MVS Programming: Extended Addressability Guide*, SA22-7614
- *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*, SA22-7609
- *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG*, SA22-7610
- *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU*, SA22-7611
- *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*, SA22-7612

## Related Publications

- *IBM Common Cryptographic Architecture: Cryptographic Application Programming Interface Reference*, SC40-1675
- *IBM ES/9000 and ES/3090 Processor Complex PR/SM Planning Guide*, GA22-7123
- *IBM Security Architecture: Securing the Open Client/Server Distributed Enterprise*, SC28-8135
- *VTAM Programming for LU 6.2*, SC31-6551
- *RSA's Frequently Asked Questions About Today's Cryptography*, available on the World Wide Web. See RSA's home page at <http://www.rsa.com>.
- *BSAFE User's Manual*
- *BSAFE Library Reference Manual*
- *Applied Cryptography, Second Edition*, Second Edition, by Bruce Schneier

## Information on Other IBM Cryptographic Products

- *IBM Transaction Security System: General Information Manual and Planning Guide*, GA34-2137
- *IBM Transaction Security System: Concepts and Programming Guide: Volume I, Access Controls and DES Cryptography*, GC31-3937
- *IBM Transaction Security System: Basic CCA Cryptographic Services*, SA34-2362
- *IBM Transaction Security System: Concepts and Programming Guide: Volume II, Public-Key Cryptography*, GC31-2889
- *IBM Distributed Key Management System, Installation and Customization Guide*, GG24-4406

## Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

<http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/> or from anywhere in z/OS or z/OS.e where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS).

The LookAt Web site also features a mobile edition of LookAt for devices such as Pocket PCs, Palm OS, or Linux-based handhelds. So, if you have a handheld device with wireless access and an Internet browser, you can now access LookAt message information from almost anywhere.

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the LookAt Web site's **Download** link.

## Accessing z/OS™ licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

<http://www.ibm.com/servers/resourceLink>

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.<sup>1</sup>

To obtain your IBM Resource Link user ID and password, log on to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

**Note:** You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

---

1. z/OS.e™ customers received a Memo to Licensees, (GI10-0684) that includes this key code.

---

# Summary of changes

## Summary of changes for SA22-7519-04 z/OS Version 1 Release 4

This document contains information previously presented in *z/OS ICSF Overview*, SA22-7519-03, which supports z/OS Version 1 Release 4.

### New information

- Support for the PCI X Cryptographic Coprocessor (PCIXCC) has been added. The new support includes:
  - changes to many callable services
  - new and changed TSO panels
  - additional services added to the default CICS wait list
  - reason code changes - errors formerly detected by ICSF are now being detected in the PCIXCC
  - new access control points
  - refer to Appendix C, “IBM @server zSeries 990 with a PCI X Cryptographic Coprocessor,” on page 73 for complete information
- Installation Options Data Set
  - CKTAUTH, an installation option, decides if authentication will be performed for every CKDS record read from DASD.

### Changed information

- Pass Phrase Initialization has been enhanced to initialize a PKDS and support the PCI X Cryptographic Coprocessor
- LPAR panel setup allows the same domain to be assigned to different LPARs if the cards are different
- CDMF keyword is no longer supported on KGUP control statements and panels
- DSS keys are not supported on the PCI X Cryptographic Coprocessor
- Cryptographic hardware, server, and configuration information (moved from the *z/OS Cryptographic Services ICSF System Programmer's Guide*) is now consolidated in chapter 5.

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

## Summary of changes for SA22-7519-03 z/OS Version 1 Release 4

This document contains information previously presented in *z/OS ICSF Overview*, SA22-7519-02, which supports z/OS Version 1 Release 3.

### **New information**

- Information is added to indicate this document supports z/OS.e and the IBM @server zSeries 800.
- Support for the IBM @server zSeries 990 server has been added. If you are running ICSF in this environment, refer to Appendix D, “IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor,” on page 81.
- Callable services
  - Symmetric Key Decipher (CSNBSYD1) - ALET support
  - Symmetric Key Encipher (CSNBSYE1) - ALET support
- Access Control Points
  - Data Key Export - Unrestricted
  - Data Key Import - Unrestricted
  - Key Export - Unrestricted
  - Key Import - Unrestricted
  - Key Part Import - Unrestricted

### **Changed information**

- Callable services
  - Callable services (Usage notes section) have been enhanced to include a table which lists the required hardware (by server) and restrictions for the callable service.
  - Encrypted PIN Verify (CSNBPVR) - *rule\_array* enhanced to support the VISAPVV4 keyword.
  - MAC Generate (CSNBMGN) and MAC Verify (CSNBMVR) has been enhanced to support longer text on a PCI Cryptographic Coprocessor.
  - Symmetric Key Decipher (CSNBSYD) has been enhanced to support the DES and TDES algorithms, but requires CP Assist for Cryptographic Functions. *Rule\_array* key processing rules CUSP, IPS, and X9.23 have been added.
  - Symmetric Key Encipher (CSNBSYE) has been enhanced to support the DES and TDES algorithms, but requires CP Assist for Cryptographic Functions. *Rule\_array* key processing rules CUSP, IPS, and X9.23 have been added.
- Additional bit definitions (Crypto assist instructions and DES and TDES enablement) have been added to the Cryptographic Communication Vector Table (CCVT).

### **Deleted information**

References to DATAC have been removed. The services affected are CV Generate, Key Export, Key Import, Key Generate, and Key Token Build. Double-length DATA keys should be used instead of DATAC.

References to Cryptographic Unit Support Product (CUSP) have been removed as the product is no longer supported.

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial

words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

### **Summary of changes for SA22-7519-02 z/OS Version 1 Release 3**

This document contains information previously presented in *z/OS ICSF Overview*, SA22-7519-01, which supports z/OS Version 1 Release 2.

#### **New information**

- Access Control Points
  - UKPT - PIN Verify, PIN Translate
- Callable services - The following new callable services perform encryption using the AES algorithm. AES encryption is only allowed if the CCC is enabled for triple DES. Only clear key support is provided.
  - Symmetric Key Decipher (CSNBSYD) - Deciphers data in an address space using the cipher block chaining or electronic code book modes.
  - Symmetric Key Encipher (CSNBSYE) - Enciphers data in an address space using the cipher block chaining or electronic code book modes.
- ICSF Setup
  - ICSF setup for E-Delivery delivery has been added. A sample ICSF options dataset, CSFPRM01, has been added to SYS1.SAMPLIB for the purpose of setting master keys by means of batch processing.
  - A sample CKDS allocation job (member CSFCKDS) has been added to SYS1.SAMPLIB.
  - A sample PKDS allocation job (member CSFPKDS) has been added to SYS1.SAMPLIB.
  - Samples for CSFSTART (ICSF Startup Procedures) has been added.
  - Sample JCL (CSFSETMK) for E-Delivery default passphrase has been added.
- Support to enable RMF to provide performance measurements on selected ICSF services and functions that use Direct Access Crypto (DAC) CCF instructions has been added.
- An appendix with z/OS product accessibility information has been added.

#### **Changed information**

- Callable services
  - Control Vector Generate (CSNBCVG) - *rule\_array* enhanced to support the UKPT keyword.
  - Key Token Build (CSNBKTB) - *rule\_array* enhanced to support the UKPT keyword.
  - Encrypted PIN Translate (CSNBPTR) - *rule\_array* enhanced to support UKPT keywords UKPTIPIN, UKPTOPIN, and UKPTBOTH.
  - Encrypted PIN Verify (CSNBPVR) - *rule\_array* enhanced to support UKPT keyword UKPTIPIN.
  - Symmetric Key Export (CSNDSYX) - a new *rule\_array* keyword, PKCSOAEP, has been added. This keyword specifies the method found in RSA PKCS #1V2 OAEP.

- Symmetric Key Generate (CSNDSYG) - a new *rule\_array* keyword, PKCSOAEP, has been added. This keyword specifies the method found in RSA PKCS #1V2 OAEP.
- Symmetric Key Import (CSNDSYI) - a new *rule\_array* keyword, PKCSOAEP, has been added. This keyword specifies the method found in RSA PKCS #1V2 OAEP.
- The ICSF TSO panels have been updated to enhance usability:
  - Coprocessor management functions have been combined onto one panel
  - Master key management/CKDS functions combined onto one panel
  - TKE TSO utilities combined onto one panel
  - Primary panel simplified
  - New utility added to generate master key values from a pass phrase

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

**Summary of changes  
for SA22-7519-01  
z/OS Version 1 Release 2**

This document contains information previously presented in *z/OS ICSF Overview*, SA22-7519-00, which supports z/OS Version 1 Release 1.

**New information**

- Callable services
  - PKA Key Token Change (CSNDKTC) callable service - This service changes PKA internal key tokens (RSA and DSS) from encipherment with the old PCI Cryptographic Coprocessor asymmetric-keys master key to encipherment with the current PCI Cryptographic Coprocessor asymmetric-keys master key.
  - Secure Messaging for Keys (CSNBSKY) callable service - This service encrypts a text block, including a clear key value decrypted from an internal or external DES token.
  - Secure Messaging for PINs (CSNBSPN) callable service - This service encrypts a text block, including a clear PIN block recovered from an encrypted PIN block.
- Installation Options Data Set
  - PKDSCACHE, an installation option, defines the size of the PKDS Cache in records. The PKDS cache improves performance as it facilitates access to frequently used records. Specify *n* as a decimal value from 0 to 256. If *n* is zero, no cache will be implemented. If PKDSCACHE is not specified, the default value is 64. PKDSCACHE can be implemented on OS/390 V2 R10 and z/OS V1 R1 by installing APAR OW48568.
  - When specifying parameter values within parentheses, leading and trailing blanks are ignored. Embedded blanks may cause unpredictable results.
- PCI Cryptographic Accelerator (PCICA) support has been added. If a PCI Cryptographic Accelerator is available, clear RSA key processing in the CSNDPKD service will be routed to the PCI Cryptographic Accelerator. If you have a PCI Cryptographic Accelerator online, toleration APAR OW49402 is required on lower levels of ICSF (OS/390 V2 R9, OS/390 V2 R10 and z/OS V1 R1).
- Support to REENCIPHER PKDS and ACTIVATE PKDS has been added to the Master Key Management Panels. The new utility, CSFPUTIL, can also be used to

reencrypt the PKDS from the old asymmetric-keys master key to the current master key and to activate the reencrypted PKDS. Toleration APAR OW49386 is required on the following systems in order to activate the re-encrypted PKDS:

- HCRP210 (standalone), HCRP220(OS/390 V2 R6, OS/390 V2 R7, OS/390 V2 R8), HCRP230 (OS/390 V2 R9), and HCR7703 (OS/390 V2 R10 and z/OS V1 R1)
- UDX support - Support for writing your own UDX has been added.

### Changed information

- Beginning in z/OS V1 R2, the DOMAIN parameter is an optional parameter in the installation options data set. It is, however, required if more than one domain is specified as the usage domain on the PR/SM panels or if running in native mode. If specified in the options data set, it will be used and it must be one of the usage domains for the LPAR. If DOMAIN is not specified in the options data set, ICSF determines which domains are available in this LPAR. If only one domain is defined for the LPAR, ICSF will use it. If more than one is available, ICSF will issue error message "CSFM409E MULTIPLE DOMAINS AVAILABLE. SELECT ONE IN THE OPTIONS DATA SET."
- Callable services
  - MAXLEN parameter checking has been eliminated for the following services:
    - Encipher (CSNBENC and CSNBENC1)
    - Decipher (CSNBDEC and CSNBDEC1)
    - MAC generate (CSNBMGN and CSNBMGN1)
    - MAC verify (CSNBMVR and CSNBMVR1)
    - Ciphertext translate (CSNBCTT and CSNBCTT1)
    - MDC generate (CSNBMDG and CSNBMDG1)

The MAXLEN parameter is also no longer enforced in the CUSP compatibility CIPHER service. The MAXLEN parameter may still be specified in the options data set, but only the maximum value limit will be enforced (2147483647). If a value greater than this is specified, an error will result and ICSF will not start.

- Pass Phrase Initialization now allows uninitialized PCI Cryptographic Coprocessors to be initialized without processing all Cryptographic Coprocessors. A new panel option (Initialize new PCICC Only) has been added to the Pass Phrase Initialization panel to allow the initialization of the new PCI Cryptographic Coprocessors.

### Deleted information

- Message IEC161I has been eliminated during the first time startup of ICSF.
- The following reason codes for ICSF/MVS X'18F' are being eliminated and will be replaced with operator messages.
  - Reason Code X'3C' - replaced by message CSFM105E
  - Reason Code X'48' - replaced by message CSFM120E
  - Reason Code X'1B' - replaced by message CSFM410E
  - Reason Code X'4B' - replaced by message CSFM107E
  - Reason Code X'106' - If the CCC is all zeroes, abend X'18F' reason code 4A will occur. If the CCC does not exist, message CSFM113E will be displayed.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.



---

## Chapter 1. Introducing Cryptography and ICSF

The Internet is rapidly becoming the basis for electronic commerce. More businesses are automating their data processing operations. Online databases are becoming increasingly large and complex. Many businesses transmit sensitive data on open communication networks and store confidential data offline. Every day the potential for unauthorized persons to access sensitive data increases.

To achieve security in a distributed computing environment, a combination of elements must work together. A security policy should be based on an appraisal of the value of data and the potential threats to that data. This provides the foundation for a secure environment.

IBM has categorized the following security functions according to International Organization for Standardization (ISO) standard 7498-2:

- Identification and authentication — includes the ability to identify users to the system and provide proof that they are who they claim to be.
- Access control — determines which users can access which resources.
- Data confidentiality — protects an organization's sensitive data from being disclosed to unauthorized persons.
- Data integrity — ensures that data is in its original form and that nothing has altered it.
- Security management — administers, controls, and reviews a business security policy.
- Nonrepudiation — assures that the appropriate individual sent the message.

Only cryptographic services can provide the data confidentiality and the identity authentication that is required to protect business commerce on the Internet.

---

### What Is Cryptography?

Cryptography includes a set of techniques for scrambling or disguising data. The scrambled data is available only to someone who can restore the data to its original form. The purpose is to make data unintelligible to unauthorized persons, but readily decipherable to authorized persons. Cryptography deals with several processes:

- **Enciphering** is converting *plaintext*, which is intelligible, into *ciphertext*, which is not intelligible. Enciphering is also called encrypting.
- **Deciphering** is converting ciphertext back into plaintext. Deciphering is also called decrypting.
- **Hashing** involves using a one-way calculation to condense a long message into a compact bit string, or message digest.
- **Generating and verifying digital signatures** involves encrypting a message digest with a private key to create the electronic equivalent of a handwritten signature. Both a handwritten signature and a digital signature verify the identity of the signer and cannot be forged.

Digital signatures also serve to ensure that nothing has altered the signed document since it was signed.

The growth of distributed systems and the increasing use of the Internet have resulted in the need for increased data security. Cryptography provides a strong, economical basis for keeping data confidential and for verifying data integrity.

Cryptography is already playing a critical and expanding role in electronic commerce and electronic mail services. Emerging markets that require secure data transmission and the authentication of the sender are already relying on cryptography.

## The Basic Elements of a Cryptographic System

Most practical cryptographic systems combine two elements:

- A process or algorithm which is a set of rules that specify the mathematical steps needed to encipher or decipher data.
- A cryptographic key (a string of numbers or characters), or keys. The algorithm uses the key to select one relationship between plaintext and ciphertext out of the many possible relationships the algorithm provides. The selected relationship determines the composition of the algorithm's result.

ICSF supports two main types of cryptographic processes:

- Symmetric, or secret key, algorithms, in which the same key value is used in both the encryption and decryption calculations.
- Asymmetric, or public key, algorithms, in which a different key is used in the decryption calculation than was used in the encryption calculation.

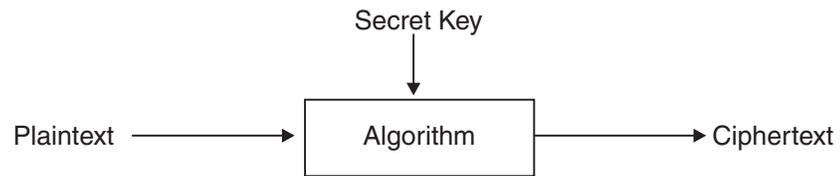
### Secret Key Cryptography

Secret key cryptography uses a conventional algorithm such as the Data Encryption Standard (DES) algorithm that is supported by ICSF. Another term for secret key cryptography is symmetric cryptography. To have intelligent cryptographic communications between two parties who are using a conventional algorithm, the following criteria must be satisfied:

- Both parties must use the same cryptographic algorithm.
- The cryptographic key that the sending party uses to encipher the data must be available to the receiving party to decipher the data.

Figure 2 on page 3 is a simplified illustration of the cryptographic components that are needed to encipher and decipher data in a secret key cryptographic system. In this system, Tom and Linda have established a secure communications channel by sharing a secret key. Tom enciphers the plaintext by using the algorithm and the secret key before sending it to Linda. When she receives the ciphertext, Linda deciphers it using the same algorithm and the same secret key. In a secret key system, it is critically important to maintain the secrecy of the shared key.

Tom enciphers a message to send to Linda



Linda decipheres the message from Tom

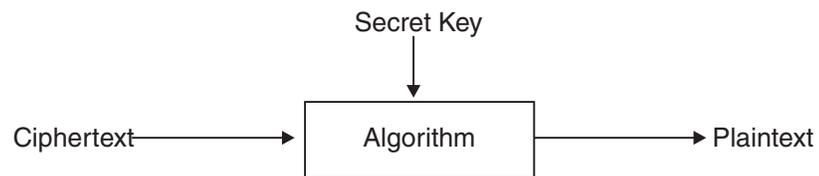


Figure 2. Enciphering and Deciphering Data in a Secret Key System

### Public Key Cryptography

Each party in a public key cryptography system has a pair of keys. One key is public and is published, and the other key is private. Another term for public key cryptography is asymmetric cryptography because the public key and private key are not identical. The sending party looks up the receiving party's public key and uses it to encipher the data. The receiving party then uses its private key to decipher the data. In a public key system, it is critically important to maintain the secrecy of the private key.

Public key cryptography requires complex mathematical calculations. For this reason, these types of systems are not used for enciphering messages or large amounts of data. They are, however, used to encipher and decipher DES keys that are transported between two systems.

Public key cryptography systems are often used to generate and verify digital signatures on electronic documents. The sender uses his or her private key to generate the digital signature. The receiver then uses the sender's public key to verify the identity of the sender. On the emerging information highway, the digital signature replaces the handwritten signature as a legal proof of authenticity. Digital signatures are the principal mechanism in any system of nonrepudiation.

Figure 3 on page 4 shows an example of a nonrepudiation system that uses digital signatures. Linda sends her broker Tom an electronic order to buy 100 shares of IBM stock. The electronic transmission application on Linda's system attaches Linda's digital signature to the order before sending the order to Tom. Linda's digital signature provides Tom with proof that Linda sent the order. When Tom receives the purchase order, an acknowledgment of his receipt, including his own digital signature, is returned to Linda. This receipt serves as proof that Tom received the order. Nonrepudiation is critical for the security of electronic data interchange (EDI).

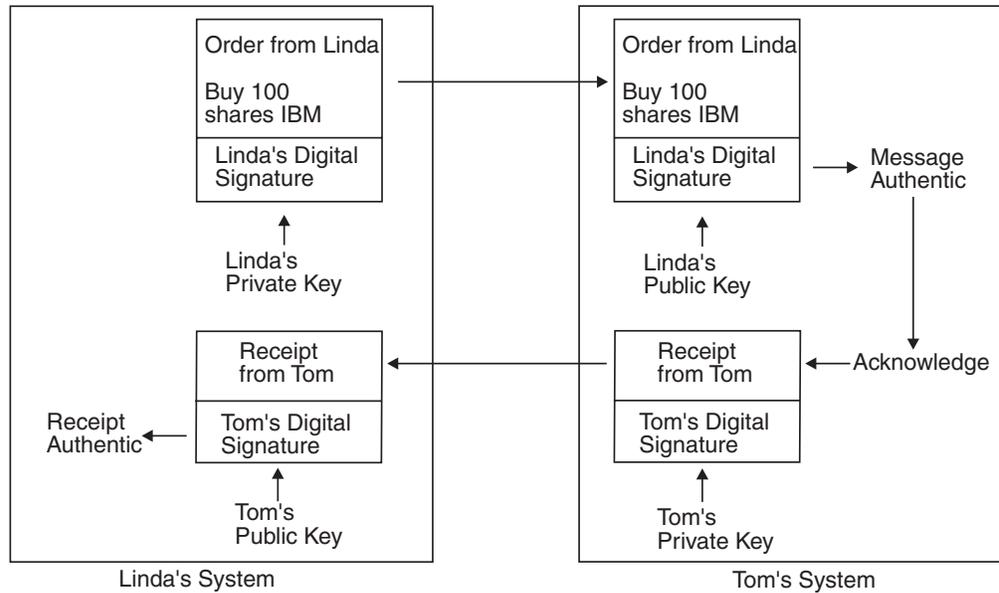


Figure 3. An Example of Nonrepudiation Using Digital Signatures

## How Does ICSF Support Cryptography?

ICSF supports IBM's Common Cryptographic Architecture (CCA). The CCA is based on the ANSI Data Encryption Algorithm (DEA). DEA is also known as the U.S. National Institute of Science and Technology Data Encryption Standard (DES) algorithm. In this secret key cryptography system, two parties share secret keys that are used to protect data and keys that are exchanged on the network. Sharing secret keys establishes a secure communications channel. The only way to protect the security of the data in a shared secret key cryptographic system is to protect the secrecy of the secret key.

On some models of the S/390 G5 Enterprise Servers, and above, ICSF also supports triple DES encryption for data privacy. Triple DES uses three, single-length keys to encipher and decipher the data. This results in a stronger form of cryptography than that available with single DES encipherment.

ICSF also supports the Advanced Encryption Standard (AES). Data can be encrypted and decrypted using 128-bit, 192-bit, and 256-bit clear keys. CBC and ECB encryption is also supported. The availability of this support is the same as triple-DES.

For public key cryptography, ICSF supports both the Rivest-Shamir-Adelman (RSA) algorithm<sup>2</sup>, and the NIST Digital Signature Standard algorithm. RSA and DSS are the most widely used public key encryption algorithms. In this system, each party establishes a pair of cryptographic keys, which includes a public key and a private key. Both parties publish their public keys in a reliable information source, and maintain their private keys in secure storage.

2. Invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adelman

**Note:** Public key cryptography is available only on S/390 G6 Enterprise Servers, and above, and S/390 Multiprisers.

---

## How Does ICSF Extend the Uses of Cryptography?

In addition to the encryption and decryption of data, ICSF provides application programs with a callable interface to perform the following tasks:

- Generate, install, and distribute DES cryptographic keys securely using both public and secret key cryptographic methods
- Generate, verify, and translate personal identification numbers (PINs)
- Ensure the integrity of data by using message authentication codes (MACs), hashing algorithms, digital signatures, or the VISA Card Verification Value/MasterCard Card Verification Code
- Develop Secure Electronic Transaction (SET) applications at the merchant and acquirer payment gateway
- PKA-encrypt and PKA-decrypt symmetric key data that Secure Sockets Layer (SSL) applications can use to generate session keys
- Develop EMV ICC applications using CSNBKDG, CSNBSPN and CSNBKY services

## Key Generation and Distribution

With ICSF callable services, you can generate a variety of cryptographic keys for use on your system or distribution to other systems. You can develop key distribution protocols by using both secret key and public key cryptographic methods. With a secret key distribution system, you must first share a secret key with the system to which you intend to distribute keys. This is a major drawback with secret key distribution systems. With public key cryptography, however, you encrypt the keys you are distributing under the receiver's public key. The receiver decrypts the keys by using the receiving system's private key. Public key encryption provides methods for key distribution and authentication.

## Personal Identification Numbers (PINs)

Many people are familiar with PINs, which enable them to use an automated teller machine (ATM). Financial networks use PINs primarily to authenticate users. Typically, the financial institution assigns a PIN. The user enters the PIN at automated teller machines (ATMs) to gain access to his or her accounts. It is extremely important to keep the PIN private, so that no one other than the account owner can use it.

ICSF enables your applications to generate PINs, to verify supplied PINs, to translate PINs from one format to another, and to store and transmit PINs in encrypted PIN blocks.

## Message Authentication Codes (MACs)

MACs are used to authenticate and verify data that is transmitted over a network, stored on the system, or stored outside the system (for example, on removable media such as tape). The MAC is a 32-bit value that is generated by using the data itself and a DES key, as specified in the ANSI X9.9 and ANSI X9.19 standards. The MAC is sent or stored with the data. The MAC is verified when the data is received, or retrieved from storage. The MAC verification process uses the data and either a MAC verification key, a MAC generation key, or a DATA key. If the verification is successful, the data has not been altered. ICSF enables your application to generate and verify MACs.

MACs give you the following benefits:

- You can validate the authenticity of data that is transmitted over a network. You can also ensure that nothing has altered the data during transmission. For example, an active eavesdropper might tap into a transmission line, and either interject bogus messages or alter sensitive data that is being transmitted. Since the sender and the receiver share a secret key, the receiver can use a callable service to calculate a MAC on the received message. The application then compares the MAC it calculates to the MAC that was transmitted with the message. The message is accepted as genuine and unaltered only if the two MACs are identical.
- Similarly, you can store a MAC with data on tape or DASD. Then, when the system retrieves the data, an application can generate a MAC and compare it with the original MAC to detect alterations.
- In either data transmission or storage, you can use MACs in an anti-virus campaign. MACs help ensure that no unauthorized executable code has been inserted into your system.

## Hashing Algorithms

The use of a hashing algorithm is another means of verifying that data has not been altered during transmission or storage. A hash, or message digest, is calculated with a public, one-way function, rather than with a secret key like a MAC. A hash, therefore, cannot be used to verify the authenticity of a message. Hashes are used in situations where it is impractical to share a secret key. For example, you can use a hash as part of a software delivery process to uncover deliberate or inadvertent modifications to software.

The originator of the data calculates the hash using the data itself and the hashing algorithm. The originator then ensures that the hash is transmitted with integrity to the intended receiver of the data. One way to ensure this is to publish the hash in a reliable source of public information. When the receiver gets the data, an application can generate a hash and compare it to the original one. If the two are equal, the data can be accepted as genuine; if they differ, the data is assumed to be bogus.

You can use the ICSF hashing algorithms to generate modification detection codes (MDCs), support the Public Key Cryptographic Standard (PKCS), and create hashes for digital signatures.

## Digital Signatures

The RSA and DSS public key cryptography systems authenticate messages and their senders through the use of *digital signatures*. A digital signature on an electronically distributed document is similar to a handwritten signature on a paper document. It is not easy to forge either type of signature. Both types of signatures authenticate that the signing party either agreed to, or generated and/ agreed to, the signed document.

The originator of the data uses a hash of the data and the originator's private key to create the digital signature. The digital signature is then attached to the message. The receiver uses the originator's public key and the signed message to verify that the message was signed by the originator.

Figure 4 on page 7 is an example of using digital signatures. The sender uses a hash of the message and the private key to create the digital signature and attach it to the message before sending it to the receiving system. The receiver uses the

sender's public key to regenerate the hash value from the digital signature and compares this hash value to a hash calculated on the received message. If the two hash values match, the message is considered to be authentic.

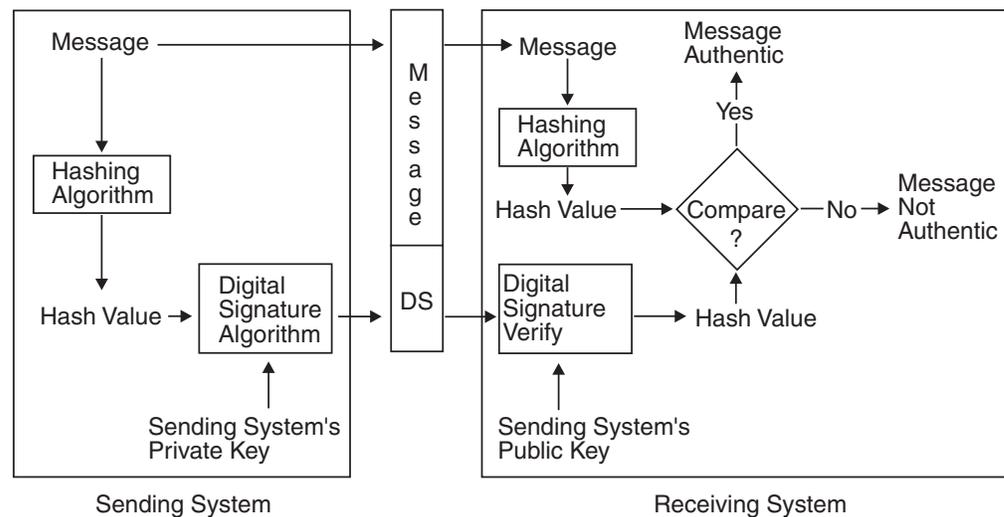


Figure 4. Creating and Verifying Digital Signatures in a Public Key System

## Card-Verification Values

The Visa International Service Association (VISA) and MasterCard International, Incorporated have specified a cryptographic method to calculate a card verification value or code. This value relates to the personal account number (PAN), the card expiration date, and the service code. ICSF provides callable services to generate and verify the VISA card-verification value (CVV) and the MasterCard card-verification code (CVC) by the track-2 method.

## Translation of Data and PINs in Networks

Increasingly data is being transmitted across networks in which, for various reasons, the keys that are used on one network cannot be used on another network. Encrypted data and PINs that are transmitted across these boundaries must be “translated” securely from encryption under one key to encryption under another key. For example, a traveler visiting a foreign city might wish to use an ATM to access an account at home. The PIN that is entered at the ATM might need to be encrypted there and sent over one or more financial networks to the traveler's home bank. The home bank must verify the PIN before the ATM in the foreign city allows access. On intermediate systems (between networks), applications can use the Encrypted PIN translate callable service to reencrypt a PIN block from one key to another. These applications can use ICSF to ensure that PINs never appear in the clear and that the keys for encrypting the PIN are isolated on their own networks. Further, applications that use ICSF can increase transaction rates.

## SET Secure Electronic Transaction

The SET Secure Electronic Transaction standard is a global industry specification that was developed jointly by Visa International, MasterCard, and other companies. The SET protocol uses digital certificates to protect credit card transactions that are conducted over the Internet. The SET standard is a major step toward securing Internet transactions, paving the way for more merchants, financial institutions, and consumers to participate in electronic commerce.

ICSF provides callable services that support the development of SET applications that run at the merchant and acquirer payment gateway.

## Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) is an industry-standard protocol that the Netscape Development Corporation designed to provide a data security layer between application protocols and TCP/IP. The SSL security protocol is widely deployed in applications on both the Internet and private intranets. SSL defines methods for data encryption, server authentication, message integrity, and client authentication for a TCP/IP connection. SSL uses public key and symmetric techniques to protect information.

SSL requires the decryption of a 48-byte SSL seed and the manipulation of this seed in the clear to produce symmetric session keys. The Common Cryptographic Architecture (CCA), however, does not permit even privacy session keys to appear in the clear in host storage. The new ICSF SSL support services permit the RSA encryption and decryption of any PKCS 1.2-formatted symmetric key data. The PKA encrypt callable service encrypts a supplied clear key under an RSA public key. Using the PKA decrypt callable service makes it possible to unwrap the RSA-encrypted SSL seed and return it to the application in the clear. The application can then use this clear key to generate session encryption keys.

## EMV Integrated Circuit Card Specifications

EMV (Europay, MasterCard and VISA) have worked together in the creation of one 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.

With ICSF, you can develop EMV ICC applications using diversified key generate (CSNBDKG), secure messaging for PINs (CSNBSPN) and secure messaging for keys (CSNBSKY) services.

---

## Chapter 2. Solving Your Business Needs with ICSF

As more businesses automate their operations and start conducting electronic commerce over the Internet, the increased use of workstations and automated teller machines generates high transaction loads. Attacks on security are becoming more sophisticated. Criminals can gain tremendous payoffs from wiretapping and theft of data from storage.

Electronic commerce, electronic funds transfer (EFT), and electronic data interchange (EDI) applications can use ICSF callable services to secure Internet transactions. These applications can make use of cryptography to protect funds transfers, purchase orders, letters of intent, contracts, credit card information, and other sensitive data from the risks of theft, fraud, or sabotage. A business can also decrease the amount of sensitive material that is exchanged by couriers. This allows a business to provide better service, become more competitive, and potentially reduce its expenses.

ICSF provides a high level of security and integrity for your data. It can help you meet many of the current needs and the future needs of your business by solving many of the information system security problems you face. This chapter explains how you can use ICSF for data security, key exchange, and personal authentication.

---

### Keeping Your Data Private

ICSF cryptographic functions are specifically designed for high security. ICSF uses the DES algorithm, which is widely regarded as highly secure, to encipher and decipher data. In addition, ICSF also provides the following security precautions:

- The master keys are stored in highly secure hardware.
- DES keys and PKA private keys are encrypted under the master keys for protection.
- You can use cryptographic keys only for their intended function. For example, a program that uses a key to verify a MAC cannot use the same key to generate MACs.
- CP Assist for Cryptographic Functions  
Clear keys are used to provide improved performance for the DES, TDES, and SHA-1 algorithms. The instructions are available on all CP's. This is only available on the IBM @server zSeries 990.
- You can use IBM Resource Access Control Facility (RACF) to control access to specific ICSF callable services, to specific keys that are stored in a CKDS, or to both.
- With the optional Trusted Key Entry (TKE) workstation, you can create a logical secure channel. You can then use this channel to distribute master keys and operational keys to remote systems. The TKE workstation is particularly suited to the distributed computing environment that requires remote key management of one or more systems. For added security, you can require that multiple security officers perform critical operations. The TKE workstation is available on:
  - IBM S/390 Multiprise
  - S/390 G6 Enterprise Server
  - IBM @server zSeries 800
  - IBM @server zSeries 900
  - IBM @server zSeries 990 - TKE Version 4 only

**Note:** TKE Version 4 will continue to support operational keys on legacy machines as it does today. Operational key support on the PCIXCC will be available at a later date.

## Transporting Data Securely Across a Network

You may need to protect data that is sent between two applications when the data must pass through one or more intermediate systems.

In a DES cryptographic system, if the two applications cannot share a key, you must set up an application on one or more of the intermediate systems to translate the ciphertext from encryption under the sending system's key. Translation re-encrypts the ciphertext under a new key for which the receiving system has a complementary key.

An application can use the ICSF ciphertext translate callable service to do this. ICSF prevents the recovery of plaintext on intermediate systems, because you cannot decrypt the data with the same key that is used to translate the ciphertext on the intermediate system. Figure 5 illustrates the use of the ciphertext translate callable service.

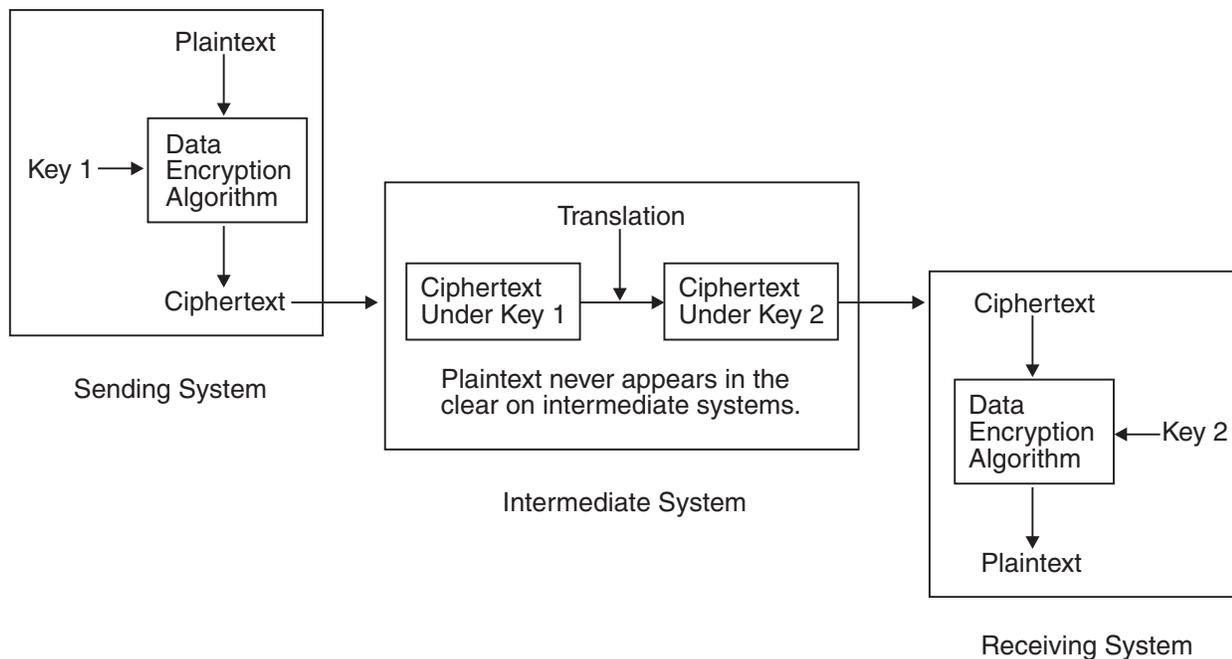


Figure 5. DES Encrypted Data Protected When Sent on Intermediate Systems

In a PKA cryptographic system, you can develop an application that does not require translation of ciphertext by the intermediate systems. The sender enciphers the message by using a DES data-encrypting key. The sender then uses the receiver's PKA public key to encipher the DES data-encrypting key. The intermediate system merely transfers the ciphertext and the enciphered key to the receiving system. The intermediate system does not have the receiver's PKA private key and, therefore, cannot decipher the enciphered data-encrypting key. Without the deciphered data-encrypting key, the intermediate system cannot decipher the message. The receiving system uses its PKA private key to decipher

the DES data-encrypting key, which it then uses to decipher the message Figure 6.

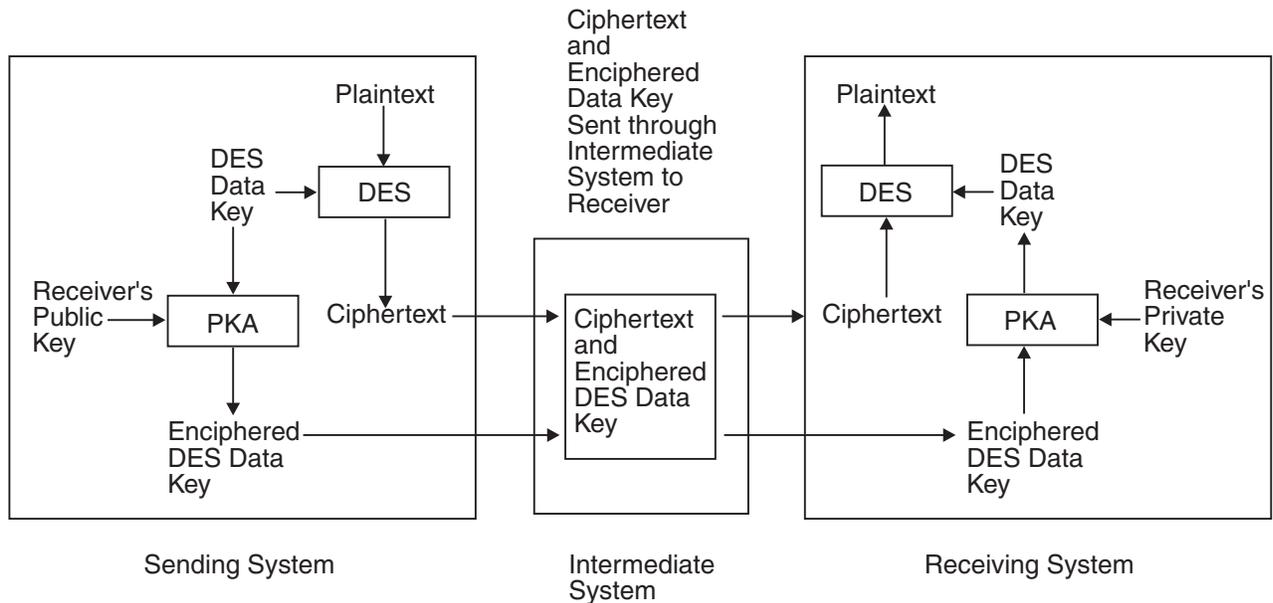


Figure 6. PKA Encrypted Data Protected When Sent on Intermediate Systems

## Supporting the Internet Secure Sockets Layer Protocol

The Secure Sockets Layer (SSL) provides a data security layer between the network layer and various internet transfer protocol applications. For example, SSL can provide a secure session between the transmission control protocol/internet protocol (TCP/IP) network layer and the hypertext transfer protocol (HTTP) or file transfer protocol (FTP) application. SSL provides data encryption, message integrity, and server authentication for TCP/IP connections between clients and servers. SSL ensures that credit card numbers and other sensitive information can be sent over the Internet without fear of interception.

To begin a secure session, the server and client exchange a handshake. In this digital handshake, the client and server are authenticated and also agree on the SSL version, data compression method, and cryptographic algorithm they will use when exchanging data. They also exchange an RSA-encrypted seed key that SSL manipulates to create symmetric session keys that are used to encrypt the data that the client and server exchange. The ICSF PKA encrypt and PKA decrypt callable services provide a secure method for SSL applications to exchange this seed key.

You can exploit PCI cryptographic accelerators (since z/OS V1R2) without entering master keys if SSL uses clear keys and APAR OW46430 is installed. This will enhance performance.

---

## Transacting Commerce on the Internet

The Internet is rapidly becoming a major arena of commerce. For electronic commerce to grow to its full potential, however, we need to resolve several barriers to buying and selling over the Internet. Consumers are reluctant to send their bank card data over the Internet without assurances that this information is secure. Merchants need to be able to determine the clear identities of their online customers. The SET Secure Electronic Transaction protocol can help to break down

these major barriers to electronic commerce. MasterCard and Visa, with the assistance of IBM and a number of technology industry partners, cooperatively developed the SET protocol.

SET is an industry-wide, open standard for online credit card transactions. The SET protocol addresses the transaction payment phase of a transaction from the individual, to the merchant, to the acquirer (the merchant's current credit card processor). The SET protocol ensures the privacy and integrity of real time bank card payments over the Internet. In addition, with SET in place, everyone in the payment process knows the identity of everyone else. The core protocol of SET is the use of digital certificates to fully authenticate the card holder, the merchant, and the acquirer. Each participant in the payment transaction holds a certificate that validates his or her identity. Public key cryptography makes it possible to exchange, check, and validate these digital certificates for every Internet transaction. The mechanics of this operation are transparent to the application.

Under the SET protocol, a digital certificate which identifies the card-holder to the merchant must accompany every online purchase. The buyer's digital certificate serves as an electronic representation of the buyer's bank card but does not actually show the credit card number to the merchant. The merchant's SET application authenticates the buyer's identity. The application then decrypts the order information, processes the order, and forwards the still-encrypted payment information to the acquirer for processing. The acquirer's SET application authenticates the buyer's credit card information, identifies the merchant, and arranges settlement. With SET, the Internet becomes a safer, more secure environment for the use of payment cards.

---

## Exchanging Keys Safely Between Networks

The practice of transmitting clear keys between networks can be a security exposure. Persons that obtain the clear keys can use them to decrypt transmitted data. ICSF offers several ways to eliminate this problem and ensure that keys are transmitted safely.

## Exchanging Keys Using DES Callable Services

ICSF provides the following security measures for DES key exchange:

- Encrypting the keys to be sent between systems, so that they are not in the clear.
- Requiring that specialized transport keys protect the data-encrypting keys or key-encrypting keys. Transport keys can be used only to protect other keys; they cannot be used for other cryptographic operations.
- Requiring that the sending (exporting) and receiving (importing) of a key be by two different, complementary forms of the same transport key (for example, export and import). These two forms are complements of each other. You cannot use a key in place of its complement.
- Requiring that a key protected under a transport key be made no longer operational—that is, not usable for other cryptographic functions such as encryption, MAC verification, and PIN verification. Only the receiving system can make a protected key operational.

An “exported” key is a key that leaves your system. The transport key that is used to protect it is called an exporter key-encrypting key. When another system receives the key, the key is still protected under the same key-encrypting key. This key-encrypting key must be installed as an importer key-encrypting key on the

receiving system. Before two systems can exchange keys, they must establish pairs of transport keys. The exporter key-encrypting key and the importer key-encrypting key are a complementary pair. You can set up pairs of transport keys, using the key generator utility program (KGUP) or callable services. To exchange keys in only one direction, you need a single pair of transport keys. To exchange keys in both directions, you need two pairs of transport keys. The illustration in Figure 7 shows an example of using DES transport keys to exchange keys between systems.

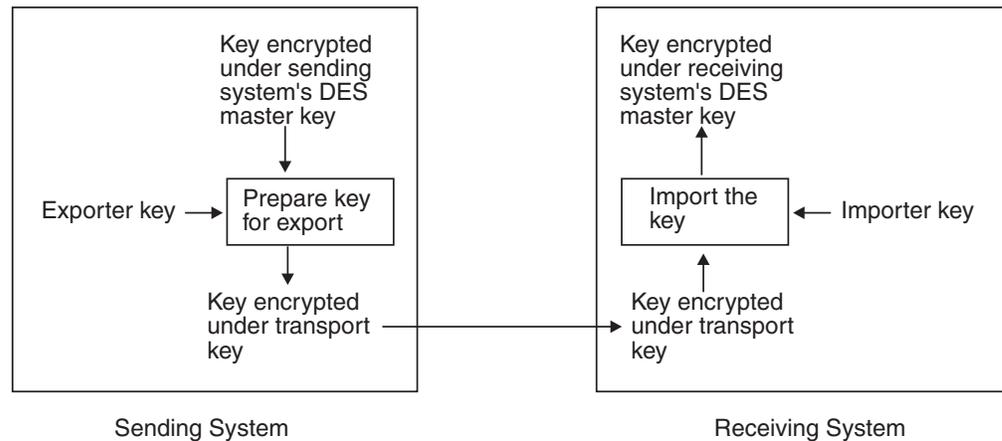


Figure 7. Key Exchange in a DES Cryptographic System

**Note:** In Program Cryptographic Facility (PCF) applications, transport key could only protect data-encrypting keys. In ICSF, all DES keys can be protected and securely distributed through the use of transport keys.

## Exchanging DES Data-encrypting Keys Using an RSA Key Scheme

The ability to create secure key-exchange systems is one of the advantages of combining both DES and PKA support in the same cryptographic system. Because PKA cryptography uses more intensive computations than DES cryptography, it is not the method of choice for all cryptographic functions. PKA cryptography enhances the security of DES key exchange. DES data-encrypting keys that are encrypted using an RSA public key can be exchanged safely between two systems. The sending system and the receiving system do not need to share a secret key to be able to exchange RSA-encrypted DES data-encrypting keys. Figure 8 shows an example of this. The sending system enciphers the DES data-encrypting key under the receiver's RSA public key and sends the enciphered data-encrypting key to the receiver. The receiver deciphers the data-encrypting key by using the receiving system's RSA private key.

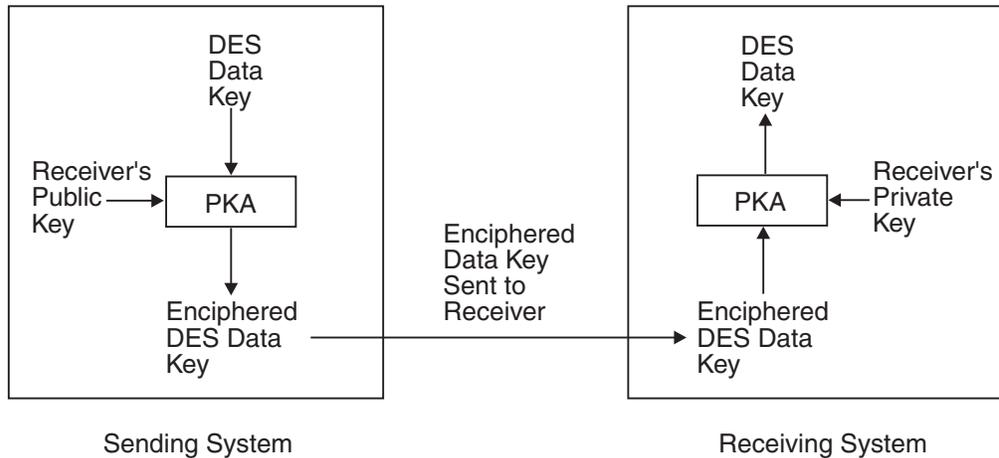


Figure 8. Distributing a DES Data-encrypting Key Using an RSA Cryptographic Scheme

**Note:** Only DES and CDMF data-encrypting keys can be encrypted under RSA public keys.

## Managing Keys Using a Trusted Key Entry Workstation

ICSF supports a Trusted Key Entry (TKE) workstation (TKE Version 4), which is available as an optional feature on:

- IBM @server zSeries 990
- IBM @server zSeries 800
- IBM @server zSeries 900
- S/390 G6 Enterprise Server

To manage master keys in the optional PCI X Cryptographic Coprocessor, you must use TKE Version 4. The TKE workstation enables the creation of a logically secure channel for master key entry and key distribution. TKE Version 4 workstations are TCP/IP attached.

ICSF supports TKE Version 3, which is available as an optional feature on:

- IBM S/390 Multiprise
- S/390 G6 Enterprise Server

To manage master keys in the optional PCI Cryptographic Coprocessor, you must use TKE Version 3 or 3.1. The TKE workstation enables the creation of a logically secure channel for master key entry and key distribution. All versions of the TKE workstation are secure. TKE Version 2 workstations (supported only on a G3 or G4) are APPC attached. TKE Version 3 workstations are TCP/IP attached.

The TKE workstation uses a variety of public key cryptographic techniques to ensure both the integrity and the privacy of the master key transfer channel. In addition, you can use a single TKE workstation to set up master keys in all the cryptographic coprocessors to which it is APPC (TKE Version 2) or TCP/IP (TKE Version 3 and Version 4) attached without manual intervention. The TKE workstation also provides support for loading operational transport and PIN keys.

**Note:** Operational key entry with TKE Version 4 is not currently available on a PCIXCC.

---

## Using Personal Identification Numbers (PINs) for Personal Authentication

*Personal authentication* is the process of validating personal identities. The personal identification number (PIN) is the basis for verifying the identity of a customer across financial industry networks. ICSF provides callable services to generate and verify PINs, and translate PIN blocks. You can use the callable services to prevent unauthorized disclosures when organizations handle PINs. Except for the Clear PIN generate callable service, PINs never appear in the clear.

ICSF provides services for handling a wide variety of PIN block formats, including:

- ISO Format 0 (same as ANSI X9.8, ECI Format 1, and VISA Format 1)
- ISO Format 1 (same as ECI Format 4)
- ISO Format 2
- VISA Format 2
- VISA Format 3
- VISA Format 4
- IBM 4704 Encrypting PINPAD Format
- IBM 3624 Format
- IBM 3621 Format (same as IBM 5906)
- ECI Format 2
- ECI Format 3

ICSF also supports the following Clear PIN generate and verification algorithms:

- IBM 3624 Institution-Assigned PIN
- IBM 3624 Customer-Selected PIN (through a PIN offset)
- IBM German Bank Pool PIN (verify through an institution key)
- IBM German Bank Pool PIN (verify through a pool key and a PIN offset)
- VISA PIN (through a VISA PIN validation value)
- Interbank PIN

For more information about PIN block formats and the ICSF callable services that support PINs, refer to *z/OS Cryptographic Services ICSF Application Programmer's Guide*.

---

## Verifying Data Integrity and Authenticity

ICSF provides several processes for verifying the integrity of transmitted messages and stored data:

- Message authentication codes (MAC)
- Modification detection codes (MDC) or hashes
- Digital signatures
- VISA card-verification value or MasterCard Card Verification Code

These processes enable your applications to verify that a message you have received has not been altered. The message itself can be in clear or encrypted form. In addition, digital signatures also authenticate the message sender's identity. VISA card-verification values ensure the safe transmission of credit card information over a computer network.

Your choice of callable service depends on the security requirements of your environment. If the sender and receiver share a secret key, use MAC processing to ensure both the authenticity of the sender and the integrity of the data. If the sender and receiver do not share a secret key, use a digital signature to ensure both the authenticity of the sender and the integrity of the data. If the sender and the receiver do not share a secret cryptographic key and you need to ensure only the integrity of transmitted data, use a hashing process.

## Using Message Authentication Codes

To use message authentication when sending a message, an application generates a MAC for it using the MAC generate callable service and one of the following methods:

- The ANSI standard X9.9, option 1 with either a single-length MAC key or a single-length DATA key
- The X9.19 optional double key MAC procedure with using a double-length MAC key.
- The EMV padding rules with using either a single-length or double-length MAC key

The originator of the message then sends the MAC with the message text.

When the receiver gets the message, an application program calls the MAC verification callable service. The service again encrypts the message text by using the same method that was used to compute the original MAC. The callable service then notifies the receiver whether the MAC has been verified or not. The callable service does not allow the receiver to have access to the MAC it generates. Because the sender and the receiver share secret cryptographic keys that are used in the MAC calculation, the MAC comparison also ensures the authenticity of the message.

**Note:** ICSF provides support for the use of a DATA key in the MAC generate and MAC verify callable services and the use of a MAC key in the MAC verify service. Double key MAC generation requires a double-length MAC key. Double key MAC verification requires either a double-length MAC key or a double-length MACVER key. Use of double length DATAM and DATAMV keys instead of double length MAC and MACVER keys is also allowed.

## Generating and Verifying Digital Signatures

An application generates a digital signature for a message by first supplying a hash of the message to the digital signature generate callable service. The callable service then uses the signer's private key to create the signature. ICSF supports the use of both RSA and DSA RSA digital signatures. To verify the digital signature, the receiver's application supplies a hash of the message and the digital signature to the digital signature verify callable service. The callable service then uses the sender's public key to verify the signature. A return code indicates that the verification either succeeded or failed. Figure 4 on page 7 provides an example of using digital signatures.

## Using Modification Detection Codes and Message Hashing

When you are sending a message, use either the MDC generate callable service, or the one-way hash generate callable service to generate a message hash. The choice depends on the cryptographic standard you are using.

The MDC is a 128-bit value that is generated by a one-way cryptographic calculation. The originator of the message transmits the MDC with integrity to the intended receiver of the file. For instance, the originator could publish the MDC in a reliable source of public information. The receiver of the message can use an application program and the same callable service to generate another MDC. If the two MDCs are identical, the receiver assumes that the message is genuine. If they differ, the receiver assumes that someone or some event altered the message.

A hash is a message digest that is generated by a one-way cryptographic calculation. ICSF supports the MD5 hash algorithm, which produces a 128-bit hash value that can be used to generate RSA digital signatures. ICSF also supports the SHA-1 hash algorithm, which generates a 160-bit hash value. Both the RSA and DSA applications can use the SHA-1 hash value and the originator's private key to generate a digital signature and attach it to the message. The receiver of the message uses the originator's public key to authenticate the digital signature.

Both MACs and hashes can be used similarly to ensure the integrity of data that is stored on the system or on removable media such as tape.

## Verifying Payment Card Data

The Visa International Service Association (VISA) and MasterCard International, Incorporated have specified a cryptographic method to calculate the VISA card-verification value (CVV) and the MasterCard card-verification code (CVC). This value relates to the personal account number (PAN), the card expiration date, and the service code and is used to detect forged cards. The CVC can be encoded on either track 1 or track 2 of a magnetic-striped card. Because most online transactions use track-2, the ICSF callable services generate and verify the CVV<sup>3</sup> by the track-2 method.

The VISA CVV service generate callable service calculates a 1- to 5-byte CVV. This value results from using two data-encrypting keys to DES-encrypt the PAN, the card expiration date, and the service code. The VISA CVV service verify callable service calculates the CVV by the same method. The service compares the CVV it calculates to the CVV supplied by the application (which reads the credit card's magnetic stripe). The service then issues a return code that indicates whether the card is authentic.

---

## Maintaining Continuous Operations

ICSF provides continuous cryptographic operations. Cryptographic keys stored in a cryptographic key data set (CKDS) can be reenciphered under a new master key or updated by using either the key generator utility program or the dynamic CKDS update callable services. ICSF performs these updates without disrupting applications in process. With PCF, you need to stop cryptographic functions before changing the master key or updating the CKDS. You do not need to stop ICSF or interrupt cryptographic applications before changing the DES master key, refreshing the CKDS, or dynamically updating either the CKDS or the public key data set (PKDS).

**Note:** The ability to change the DES master key or update the CKDS without interruption requires that ICSF be running in noncompatibility mode. That is,

---

3. The VISA CVV and the MasterCard CVC refer to the same value. This information uses CVV to mean both CVV and CVC.

you must convert all existing PCF applications to the new callable services. For a description of noncompatibility mode, see “Running PCF Applications under ICSF” on page 37.

The following features and actions enhance the security of cryptographic functions:

- Performing cryptographic calculations and storing master keys within tamper-resistant hardware
- Enforcing separation of DES keys
- Controlling access to functions and keys through the use of RACF
- Generating system management facility (SMF) audit records

---

## Reducing Costs by Improving Productivity

ICSF improves productivity by simplifying routine operations and providing interfaces and callable services that help you manage your enterprise’s cryptographic environment.

ICSF simplifies the job of the security administrator by providing ISPF dialogs for key management and distribution. ICSF also provides a pass phrase initialization procedure that generates and loads all needed master keys. Use pass phrase initialization to fully enable your cryptographic system in a minimum of steps. In addition, a series of Clear Master Key Entry panels simplifies the master key entry procedure. These panels permit the administrator to change the DES master key without interrupting application programs that use cryptographic functions.

In enterprises that require enhanced key-entry security, a Trusted Key Entry (TKE) workstation is available as an optional feature. The TKE workstation allows the security administrator to securely load DES and PKA master keys and operational keys (PIN keys and Transport keys). The security administrator can use the TKE workstation to load keys into multiple Cryptographic Coprocessor Features, PCI Cryptographic Coprocessors, and PCI X Cryptographic Coprocessors from a remote location.

**Restriction:** Operational key entry with TKE Version 4 is not currently available on PCIXCCs.

ICSF provides the application programmer with a set of callable services that support cryptographic functions and key management protocols. Applications written in Assembler and several high-level programming languages can use these callable services.

ICSF provides the systems programmer with an easy method of setting and changing the ICSF installation options. The systems programmer needs only to edit an options data set rather than altering an object module. ICSF provides a sample installation options data set in member CSFPRM00 of SYS1.SAMPLIB.

---

## Improving Cryptographic Performance

ICSF uses state-of-the-art hardware to improve performance of both DES and PKA calculations. This can remove limitations on the growth of your installation and enable it to use cryptography in high-transaction-rate applications. ICSF also improves performance by exploiting z/OS and by using an in-storage copy of the CKDS. Maintaining DES cryptographic keys in a protected data space (in addition to a data set) improves performance and availability by reducing requirements for read access to cryptographic keys.

## Using RMF and SMF to Monitor z/OS ICSF Events

You can run ICSF in different configurations and use installation options to affect ICSF performance. While ICSF is running, you can use the Resource Management Facilities (RMF) and System Management Facilities (SMF) to monitor certain events. For example, ICSF records information in the ICSF SMF data set when ICSF status changes in a processor or when you enter or change the master key. ICSF also sends information and diagnostic messages to data sets and consoles.

With the availability of cryptographic hardware (PCI X Cryptographic Coprocessor, PCI Cryptographic Coprocessor and PCI Cryptographic Accelerator) on an LPAR basis, RMF provides performance monitoring in the Postprocessor Crypto Hardware Activity report. This report is based on SMF record type 70, subtype 2. The Monitor I gathering options on the REPORTS control statement are CRYPTO and NOCRYPTO. Specify CRYPTO to measure cryptographic hardware activity and NOCRYPTO to suppress the gathering. In addition, overview criteria is shown for the Postprocessor in the Postprocessor Workload Activity Report - Goal Mode (WLMGL) report. Refer to *z/OS RMF Programmer's Guide*, SC33-7994, *z/OS RMF User's Guide*, SC33-7990, and *z/OS RMF Report Analysis*, SC33-7991 for additional information.

ICSF also supports enabling RMF to provide performance measurements on ICSF services (Encipher, Decipher, MAC Generate, MAC Verify, One Way Hash, PIN Translate, and PIN Verify) and functions that use Direct Access Crypto (DAC) CCF instructions.

For diagnosis monitoring, use Interactive Problem Control System (IPCS) to access the trace buffer and to format control blocks.

## Improving Performance in a CICS Environment

ICSF supports a CICS-ICSF attachment facility that improves the performance of applications in the CICS regions when an application in the region requests a long-running ICSF service. The attachment facility consists, in part, of a CICS Task Related User Exit (TRUE) that attaches a task control block that does the actual call to the ICSF services. The CICS Resource Manager Interface allows a CICS application program to invoke code that is not written expressly for use under CICS, using the application programming interface that is native to that code. Code that is accessed in this manner is called a resource manager. In the case of the CICS-ICSF attachment facility, ICSF becomes a resource manager for CICS. This means that a CICS application desiring to use long-running ICSF services (such as PKA operations) can be placed in a CICS WAIT rather than an OS WAIT for the duration of the operation. This results in improved performance for other applications that are running in the same CICS region.

For additional information about installing the CICS-ICSF attachment facility or creating a modifiable CICS Wait List, refer to *z/OS Cryptographic Services ICSF System Programmer's Guide*. The WAITLIST parameter, an option in the Installation Options data set, points to a modifiable data set which contains the names of services that are placed in the CICS Wait List. If this option is not specified, the default ICSF CICS Wait List will be utilized by ICSF when a CICS application invokes an ICSF callable service.

---

## Customizing ICSF to Meet Your Installation's Needs

ICSF provides the flexibility your installation needs to customize your cryptographic system.

## Using ICSF Exits to Meet Special Needs

Exits are programs that your system programmer writes to meet your installation's particular needs. These exits (and installation-defined callable services) perform tasks such as tailoring, monitoring, changing, or diagnosing ICSF. Use of such interfaces can create dependencies on the detailed design or implementation of ICSF. For this reason, use installation exits only for these specialized purposes.

ICSF exits include the following:

- Exits called when an operator command starts, stops, or changes ICSF
- An exit for each of the callable services
- Exits that are called when you access the disk copy of the CKDS
- An exit that is called when an application accesses the in-storage CKDS

For more information about ICSF exits, refer to *z/OS Cryptographic Services ICSF System Programmer's Guide*.

## Creating Installation-Defined Callable Services

Your installation can define a callable service that will run in the ICSF address space and have access to selected ICSF control blocks.

The UDX function is invoked by an "installation-defined" or generic callable service. The callable service is defined in the Installation Options data set (UDX parameter) and the service stub is link-edited with the application. The application program calls the service stub which accesses the UDX installation-defined service.

There is a one-to-one correspondence between a specific generic service in ICSF and a specific UDX command processor in the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor. The administrator, through ICSF panels, performs UDX authorization processing on each PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor. Authorization is not LPAR specific. See *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521, for additional information on authorizing User Defined Extensions.

z/OS V1 R2 added support for writing your own UDX for a PCI Cryptographic Coprocessor. (Development of a UDX for a PCIX Cryptographic Coprocessor requires a special contract with IBM.) See the *UDX Reference and Guide* and the *4758 Custom Software Developer's Toolkit Guide* for additional information. These, and other publications related to the IBM 4758 Coprocessor can be obtained in PDF format from the Library page located at <http://www.ibm.com/security/cryptocards>.

## Using Options to Tailor ICSF

ICSF lets your installation use different sets of options at different times in the operation of your system. Your installation can specify which options are in each set. The following are some of your choices:

- You can choose which of three migration options to use when migrating from or coexisting with PCF: noncompatibility mode, compatibility mode, or coexistence mode.

For more information on running PCF applications with ICSF, refer to "Running PCF Applications under ICSF" on page 37.

- You can allow processing in special secure mode, in which you can work with clear keys and clear PINs. Alternatively, you can disallow processing in that mode.

- For each exit point, you can specify the name of the exit routine and operating information.
- You can alter the REASONCODES options parameter in the Installation Options data set to determine which set of reason codes (ICSF or TSS values) are returned to application service calls. If the REASONCODES option is not specified, the default of REASONCODES(ICSF) is used. The codes will only be converted if there is a 1-to-1 correspondence.
- You can use the WAITLIST (data\_set\_name) options parameter in the Installation Options Data Set to point to a modifiable data set that contains the names of services that are placed into the CICS Wait List. If the WAITLIST option is not specified, the default ICSF CICS Wait List will be utilized by ICSF when a CICS application invokes an ICSF callable service.
- You can use the UDX(UDX-id,service-number,load-module name,'comment\_text',FAIL(fail-option)) parameter to define a User Defined Extension (UDX) service to ICSF.
- You can use the PKDSCACHE option to define the size of the PKDS cache in records. The PKDS cache improves performance as it facilitates access to frequently used records. Specify *n* as a decimal value from 0 to 256. If *n* is zero, no cache will be implemented. If PKDSCACHE is not specified, the default value is 64. PKDSCACHE can be implemented on OS/390 V2 R10 and z/OS V1 R1 by installing APAR OW48568.

---

## Isolating and Protecting PR/SM Partitions

If you are using the Processor Resource/Systems Manager (PR/SM) feature to run in logically partitioned mode, each PR/SM partition is able to use its own DES and PKA master keys on the Cryptographic Coprocessor Feature and symmetric-keys master key (SYM-MK) and asymmetric-keys master key (ASYM-MK) on the PCI X Cryptographic Coprocessor or PCI Cryptographic Coprocessor. This allows your installation to have multiple independent cryptographic systems running on the same processor with the same degree of isolation and protection as if they were running on physically separate processors.

---

## Enabling Growth

For applications that need to protect critical data against disclosure or modification, ICSF provides callable services that enable high-level language applications to easily access the system's underlying cryptographic functions.

By providing callable services that comply with IBM's Common Cryptographic Architecture (CCA), ICSF enables application designers and programmers to extend the uses of their current applications. Most of the callable services provided by ICSF are also provided by the IBM 4753 Network Security Processor MVS Support Program (4753-HSP), program number 5706-028. This allows the development of significant applications for both CCA and ANSI X9.17 key management that will run without change on both systems.

On the PCI X Cryptographic Coprocessor and the PCI Cryptographic Coprocessor, you can generate RSA keys.

ICSF's callable services enable installations to add cryptographic functions (such as MAC generation and verification) to current applications without redesign.

The combination of the hardware cryptographic features and ICSF provides high-performance cryptography, which removes bottlenecks on high-volume

transaction applications and gives them needed protection. ICSF can support various combinations of cryptographic hardware.

An installation can use ICSF installation exits to change or extend the callable services.

---

## Protecting Your Investment

The use of an enterprise's computing resources is improved and protected by built-in product features.

An ICSF/MVS Version 1 Release 2, or later, Cryptographic Key Data Set (CKDS) can be used in OS/390 ICSF or z/OS without change. If you are upgrading from ICSF/MVS Version 1 Release 1, use the conversion program that is provided to migrate an ICSF/MVS Version 1 Release 1 CKDS for use with later releases of ICSF. The migration program, which can run under either product version, leaves an appropriate audit trail.

ICSF also ensures that existing Program Cryptographic Facility (PCF) cryptographic applications, skills, and equipment can continue to be used effectively. This facilitates the earlier implementation of desired security applications while minimizing the disruption of existing applications.

- Existing PCF applications can run without change and without reassembly on ICSF in compatibility mode.
- A PCF conversion program has been provided to convert a PCF cryptographic key data set to an ICSF format.
- ICSF applications can run concurrently on the same processor with PCF applications.

---

## Chapter 3. Callable Services and Key Management

This chapter describes the ICSF callable services and some of the concepts of cryptographic key management.

---

### Callable Services

ICSF provides access to cryptographic functions through callable services. A callable service is a routine that receives control from a CALL statement in an application language. Each callable service performs one or more cryptographic functions or a utility function. Many of these callable services comply with IBM's Common Cryptographic Architecture (CCA), while others are extensions to the CCA.

The callable services available to your applications depend on your processor or server. For a list of the callable services available with each configuration, refer to Appendix B, "Summary of Callable Service Support by Hardware Configuration," on page 67.

The application programs can be written in high-level languages such as C, COBOL, FORTRAN, and PL/I, as well as in Assembler. ICSF callable services allow applications to perform the following tasks:

- Enciphering and deciphering data by using the cipher block chaining (CBC) form of the DES or TDES algorithms, with or without record chaining using a single (64-bit), double (128-bit), or triple (192) length key. If using the AES algorithm, key lengths are 128, 196 or 256 bits.
- Translating ciphertext from encryption under one key to encryption under another key by use of the Ciphertext translate callable service.

This service securely decipheres the text that was enciphered under one key, and then enciphers it under another key. The service uses the cipher block chaining (CBC) form of the DES algorithm.

- Generating DES cryptographic keys of all types for use by application programs.
- Importing and exporting keys.
- Generating RSA public and private key pairs.

Application programs can use the PKA key generate callable service to generate RSA public and private keys on a PCI Cryptographic Coprocessor, if available. To support the SET Certificate Authority requirements, the RSA private key may be retained within the secure hardware boundaries of the PCI Cryptographic Coprocessor.

- Listing and deleting retained RSA private keys.

Application programs can list and delete RSA private keys retained within the secure boundaries of a PCI Cryptographic Coprocessor.

- Generating random numbers.

Application programs can use a callable service to generate a random number for use in cryptography or for other general use. The callable service uses the cryptographic feature to generate a random number for use in encryption. The foundation for the random number generator is a time-variant input with a very low probability of recycling.

- Encoding and decoding data through the use of clear keys and the electronic code book (ECB) form of the DES algorithm.
- Generating and verifying PINs and translating PIN blocks.

An application program can use the callable services in generating and verifying PINs. In addition, use the Encrypted PIN translate callable service to reencrypt a PIN block from one PIN-encrypting key to another, or to reformat a PIN block.

- Generating and verifying MACs.

An application can use single-length or double-length MAC or MACVER keys, or single-length DATA keys to generate and verify message authentication codes.

- Generating MDCs and other hash patterns.
- Generating and verifying Visa CVVs.
- Develop EMV ICC applications
- Updating the CKDS and PKDS dynamically.

ICSF provides callable services that application programs can use to create, read, write, and delete records in the CKDS and PKDS.

- Supporting the ANSI X9.17 key management standard.

ICSF provides callable services for generating, exporting, importing, translating, and notarizing ANSI X9.17 keys.

- Distributing DATA keys enciphered under an RSA key.
- Generating and verifying digital signatures.
- Composing and decomposing SET blocks.
- PKA-encrypting and PKA-decrypting any PKCS 1.2-formatted symmetric key data.

---

## Protecting and Controlling DES Keys

DES keys are protected by encryption under a DES master key. The DES master key always remains within the secure boundary of the cryptographic feature on the server. There is only one DES master key and it is used only to encrypt and decrypt other DES keys.

**Note:** On a IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor, there is no encrypted key support.

On the PCI X Cryptographic Coprocessor and PCI Cryptographic Coprocessor, DES keys are protected by the SYM-MK which always remains within the secure hardware boundary of the card. The verification pattern produced by each type of master key must be the same in order for the cryptographic coprocessors to be active.

The cryptographic hardware controls the use of DES keys by separating them into unique types. A unique key type can be used only for a specific purpose. For example, you cannot protect a key with a key that is intended to protect data. This hardware-enforced key separation provides better key protection than software key separation techniques. To enforce key separation, the cryptographic hardware automatically encrypts each type of key under a unique variation of the DES or SYM-MK. Each variation encrypts a different type of key. Although you enter only one DES or SYM-MK, in effect you have a unique master key to encrypt each DES key type.

**Note:** In PCF, key separation applies only to keys that are encrypted under the master key. In ICSF, key separation also applies to keys that are encrypted under transport keys or key-encrypting keys. This enables the creator of a key to transmit the key to another system and to enforce its use at the other system.

## DES Master Key Variant

Each key must be enciphered under the DES master key or SYM-MK before it can be used in any cryptographic function. Each key type is enciphered with a unique variation of the master key called a *Master key variant*. ICSF creates a master key variant by exclusive ORing a fixed pattern, called a *control vector*, onto the master key. For information about control vectors, refer to “Control Vectors.”

Each master key variant protects a different type of key. The effect is similar to having a unique master key to protect all the keys of a certain type. The master key, in its variants, protects keys that operate on the system. When systems want to share keys, they use transport keys to protect keys sent outside of systems.

## DES Transport Key Variant

As with the master key, ICSF also creates variations of a DES transport key to encrypt a key according to its type. This allows for key separation when transporting keys off the system. A *transport key variant*, or *key-encrypting key variant*, is created in the same way as a master key variant. The transport key is exclusive ORed with a control vector that is associated with the key type of the key it protects.

**Note:** To exchange keys with systems that do not recognize transport key variants, ICSF allows you to encrypt selected keys under a transport key itself, not under the transport key variant.

## DES Key Forms

A key that is protected under the DES or SYM-MK is in *operational form*, which means that ICSF can use it in cryptographic functions on the system.

When you store a key with a file or send it to another system, the key is enciphered under a transport key rather than the master key. When ICSF enciphers a key under a transport key, the key is not in operational form and cannot be used to perform cryptographic functions.

When a key is enciphered under a transport key, the sending system considers the key to be in the *exportable form*. The receiving system considers the key to be in the *importable form*. When a key is re-enciphered from under a transport key to under a system’s master key, it is in operational form again.

## Control Vectors

For each type of DES key the master key enciphers, there is a unique control vector. The cryptographic feature exclusive ORs the master key with the control vector associated with the type of key the master key will encipher. For example, all the different types of DATA, PIN, MAC, and transport keys are each exclusive ORed with a unique control vector. The control vector ensures that an operational key can be used in only cryptographic functions for which it is intended. For example, the control vector for an input PIN-encrypting key ensures that such a key can be used only in the PIN translation and PIN verification functions. “Types of DES Keys” describes the different DES key types.

## Types of DES Keys

ICSF groups DES cryptographic keys into the following categories according to the functions they perform.

- DES Master key

The DES master key is a double-length (128-bit) key that is used only to encrypt other DES keys. The ICSF administrator installs and changes the DES master key using the ICSF panels. Alternatively, you can use the optional TKE workstation. The master key always remains within the secure boundary of the cryptographic feature. The DES master key is installed in the CCFs.

The DES master key is used only to encipher and decipher operational keys. Cryptographic keys that are in exportable or importable form are not enciphered under the master key. They are enciphered under the appropriate transport key, which has itself been enciphered under the master key.

- Symmetric-Keys Master Key (SYM-MK)

The SYM-MK is a double-length (128-bit) key that is used to encrypt other DES keys on the PCI X Cryptographic Coprocessor and PCI Cryptographic Coprocessor. The ICSF administrator installs and changes the SYM-MK using either the ICSF panels or the optional Trusted Key Entry (TKE) workstation. The master key always remains within the secure boundary of the PCI X Cryptographic Coprocessor or PCI Cryptographic Coprocessor.

As with the DES master key, the SYM-MK is used only to encipher and decipher operational keys. On a system with both CCFs and PCICCs, the SYM-MK must have the same value as the DES master key.

- Transport keys (or key-encrypting keys)

Transport keys are also known as key-encrypting keys. They are double-length (128-bit) keys that are used to protect keys when you distribute them from one system to another. For installations that do not support double-length 128-bit keys, ICSF supports the use of effective single-length keys. In an effective single-length key, the left half equals the right half.

There are five types of transport keys:

- *Exporter or OKEYXLAT key-encrypting keys* protect keys of any type that are sent from your system to another. The exporter key at the originator is the same as the importer key of the receiver. An exporter key is paired with an importer key or a IKEYXLAT key.
- *Importer or IKEYXLAT key-encrypting keys* protect keys of any type that are sent from another system to yours. It also protects keys that you store externally in a file that you can import to your system later. The importer key at the receiver is the same as the exporter key at the originator. An importer key is paired with an exporter key or a OKEYXLAT key.
- *ANSI X9.17 key-encrypting keys (AKEKs)* are used exclusively with the ANSI X9.17 key management callable services. AKEKs are used to transport DATA keys, AKEKs, and CCA key-encrypting keys.

**Note:** Transport keys replace the local, remote, and cross keys that PCF uses.

- Data-encrypting keys

Data-encrypting (DATA) keys are single-length (64-bit), double-length (128-bit), or triple-length (192-bit) keys. DATA keys are used to encipher and decipher data. ICSF provides support for the use of single-length data-encrypting keys in the callable services that generate and verify MACs.

DATAc, a double length key for enciphering and deciphering data, is available with a PCI X Cryptographic Coprocessor.

- CIPHER Keys

These consist of CIPHER, ENCIPHER and DECIPHER keys. They are single and double length keys for enciphering and deciphering data

**Note:** Double length CIPHER, ENCIPHER, and DECIPHER keys are only supported on the IBM @server zSeries 990.

- Data-translation keys

These single-length (64-bit) keys are used for the ciphertext translate callable service as either the input or the output data-translation (DATAXLAT) key.

- MAC keys

These single-length (64-bit) (MAC and MACVER) and double-length (128-bit) (DATAM and DATAMV) keys are used for the callable services that generate and verify MACs.

With a PCI X Cryptographic Coprocessor, MAC and MACVER can be single or double-length keys.

- PIN keys

The personal identification number (PIN) is a basis for verifying the identity of a customer across financial industry networks. PIN keys are double-length (128-bit) keys. The callable services that generate, verify, and translate PINs use PIN keys.

For installations that do not support double-length 128-bit keys ICSF provides effective single-length keys. In an effective single-length key, the left key half of the key equals the right key half.

- Cryptographic variable encrypting keys

These single-length keys are used to encrypt special control values in CCA DES key management. The Control Vector Translate and Cryptographic Variable Encipher callable services use cryptographic variable encrypting keys.

---

## Protecting and Controlling PKA Keys

In a public key cryptographic system, it is a priority to maintain the security of the private key. It is vital that only the intended user or application have access to the private key.

On supported IBM servers, ICSF and the cryptographic hardware features (CCFs, PCICCs and PCIXCCs) ensure this by enciphering PKA private keys under a unique PKA object protection key. The PKA object protection key has itself been enciphered under a PKA master key. Each PKA private key also has a name that is cryptographically bound to the private key and cannot be altered. ICSF uses the private key name or the PKDS key label to control access to the private key. This combination of hardware-enforced coupling of cryptographic protection and access control, through the use of the Security Server (RACF), is unique to ICSF. It provides a significant level of security and integrity for PKA applications.

The PCI Cryptographic Coprocessor, which is available as an option on the S/390 G6 Enterprise Server, IBM @server zSeries 800 and the IBM @server zSeries 990, provides additional security for PKA applications. You can generate RSA public and private key pairs within the secure hardware boundary of the PCI Cryptographic Coprocessor. In addition, you can retain the RSA private key within the PCI card where it is generated, a requirement to be a SET Certificate Authority. The RSA private key is protected by the ASYM-MK on the PCI Cryptographic Coprocessor.

The PCI X Cryptographic Coprocessor, which is available as an option on the IBM @server zSeries 990, provides additional security for PKA applications. You can generate RSA public and private key pairs within the secure hardware boundary of the PCI X Cryptographic Coprocessor. In addition, you can retain the RSA private

key within the PCIX card where it is generated, a requirement to be a SET Certificate Authority. The RSA private key is protected by the ASYM-MK on the PCI X Cryptographic Coprocessor.

## PKA Master Keys

The PKA master keys on the Cryptographic Coprocessor Feature are triple-length (192-bit) keys. As with the DES master key, the PKA master keys are used only to encipher and decipher PKA keys. There are two PKA master keys on the Cryptographic Coprocessor Feature. One PKA master key, the signature master key (SMK), protects private keys that are intended for creating digital signatures. The other PKA master key, the key management master key (KMMK), protects private keys that are used in DES key distribution. Private keys that are protected by the KMMK can also be designated to be usable to generate digital signatures. It is highly recommended that the KMMK have the same value as the SMK.

The asymmetric keys master key (ASYM-MK) on the PCI Cryptographic Coprocessor and PCI X Cryptographic Coprocessor is a triple-length key used to encipher and decipher PKA keys. In order for the PCI Cryptographic Coprocessor to function, the hash pattern of the ASYM-MK must have the same value as the hash pattern of the SMK on the Cryptographic Coprocessor Feature. There is no such dependency with the PCI X Cryptographic Coprocessor.

The ICSF administrator installs the PKA master keys on the Cryptographic Coprocessor Feature and the ASYM-MK on the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor by using either the ICSF pass phrase initialization panel, the clear master key entry panels, or the optional TKE workstation.

## RSA Private and Public Keys

An RSA key pair includes a private and a public key. The RSA private key is used to generate digital signatures, and the RSA public key is used to verify digital signatures. The RSA public key is also used for key encryption of DES DATA keys and the RSA private key for key recovery.

The RSA public key algorithm is based on the difficulty of the factorization problem. The factorization problem is to find all prime numbers of a given number,  $n$ . When  $n$  is sufficiently large and is the product of a few large prime numbers, this problem is believed to be difficult to solve. For RSA,  $n$  is typically at least 512 bits, and  $n$  is the product of two large prime numbers. For more information about the RSA public key algorithm, refer to the ISO 9796 standard and *RSA's Frequently Asked Questions About Today's Cryptography*.

### Generating RSA Keys on a Cryptographic Coprocessor Feature

The Cryptographic Coprocessor Feature does not provide the ability to generate RSA public and private keys within the secure hardware boundary. There are several ways to generate RSA key pairs and load them.

- You can use the optional TKE Workstation with Version 2.0 or higher of the TKE Workstation code to generate the RSA key pair and load them directly into the ICSF public key data set (PKDS) on the server.
- You can generate RSA key pairs in the encrypted form on a workstation with a 4755 cryptographic adapter installed. A workstation with a 4758 PCI Cryptographic Coprocessor can also be used. Use the PKA key import callable service to import the RSA key pairs in the form of an external PKA private key token.

- You can generate RSA keys in the clear on another platform or by using any suitable software program. Use the PKA Key Token Build callable service to build an external token with clear key values. Then use the PKA key import callable service to import the RSA keys into the PKA key token in the internal format.

### **Generating RSA Keys on a PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor**

With the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor you can use the PKA key generate callable service to generate RSA public and private key pairs within the secure boundary of the PCI card. The cryptographic coprocessors can generate RSA keys with a modulus size of 512 to 2048 bits. The RSA private key may be retained and used within the secure boundary of the PCI card. This capability is a requirement to be a SET Certificate Authority. The public key and the key name for the private key are stored in the ICSF public key data set (PKDS), but the value of a retained private key never appears in any form outside the PCI card.

## **DSA Private and Public Keys**

A DSA key pair also includes a private and a public key. The DSA private key is used to generate digital signatures, and the DSA public key is used to verify digital signatures.

The difficulty of the discrete logarithm problem is the basis for the NIST Digital Signature Standard (DSS) public key algorithm. The discrete logarithm problem is to find  $x$  given a large prime  $p$ , a generator  $g$  and a value  $y=(g^{**x}) \bmod p$ , where  $**$  represents exponentiation. This problem is believed to be very hard when  $p$  is sufficiently large and  $x$  is a sufficiently large random number. For DSA,  $p$  is at least 512 bits, and  $x$  is 160 bits. The NIST FIPS 186 Digital Signature Standard defines DSA.

ICSF provides a callable service to generate PKA internal key tokens for use with the DSA algorithm in digital signature services.

**Restriction:** The IBM @server zSeries 990 does not support DSS.

---

## **Exchanging Encrypted Keys and PINs on a DES System**

When a system sends a DATA key to another system, the sending system encrypts the DATA key under an *exporter* key-encrypting key. The receiving system re-encrypts the DATA key from encryption under an *importer* key-encrypting key to encryption under its master key. The importer and exporter key-encrypting keys at these systems complement each other and have the same clear value.

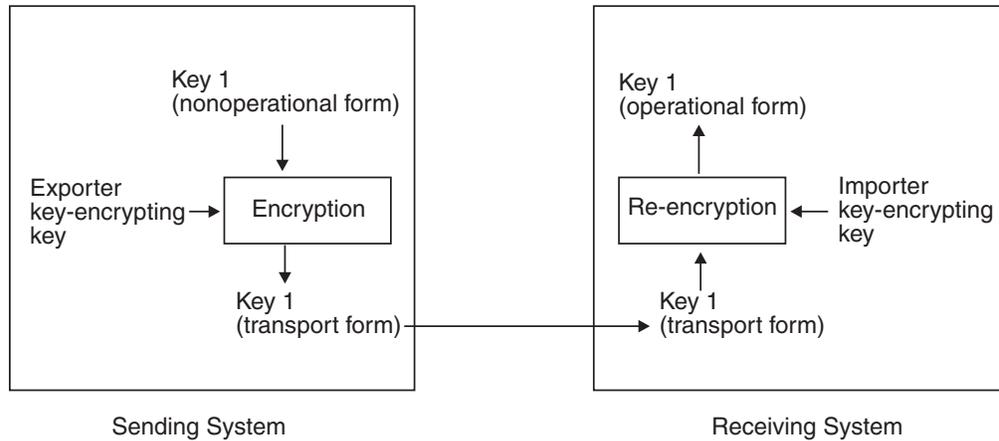


Figure 9. Using Transport Keys to Exchange Keys

In ICSF, you work with the following complementary keys:

- Importer key-encrypting key and exporter key-encrypting key
- Importer key-encrypting key and OKEYXLAT key-encrypting key
- Exporter key-encrypting key and IKEYXLAT key-encrypting key
- Input PIN-encrypting key and output PIN-encrypting key
- PIN-generation key and PIN-verification key
- MAC-generation key and MAC-verification key

Your installation can use the key generator utility program (KGUP) or the callable services to generate and maintain complementary pairs of keys.

When KGUP generates a key, it also generates a KGUP control statement to create the complement of that key. You can send the control statement to the system with which you are exchanging keys or PINs.

---

## Exchanging RSA-Encrypted Data Keys

In an RSA cryptographic system, the sending system and the receiving system do not need to share complementary importer and exporter key pairs to exchange DATA keys. The sender enciphers the DATA key by using the receiver's public key. The receiver decipheres the DATA key by using his or her own private key. Refer to "Exchanging DES Data-encrypting Keys Using an RSA Key Scheme" on page 13 for a more detailed explanation.

---

## Using Multiple Encipherment to Protect Keys and Data

The PCI X Cryptographic Coprocessor, Cryptographic Coprocessor Feature and PCI Cryptographic Coprocessor use multiple encipherment whenever they encipher a key under a key-encrypting key like the master key or a transport key. In addition to protecting and retrieving cryptographic keys, the Cryptographic Coprocessor Feature uses multiple encipherment and decipherment to protect or retrieve 64-bit PIN blocks in the area of PIN applications. Multiple encipherment is superior to single encipherment because it is much harder to break. The actual process to encipher a key depends on the type of key that is being enciphered and the type of key-encrypting key that is being used to encipher it. Figure 10 on page 31 shows an example of multiple encipherment. In this example, the left half of the enciphering key is used to encrypt the key in the first step. The result is then decrypted under

the right half of the enciphering key. Finally, this result is encrypted under the left half of the enciphering key again.

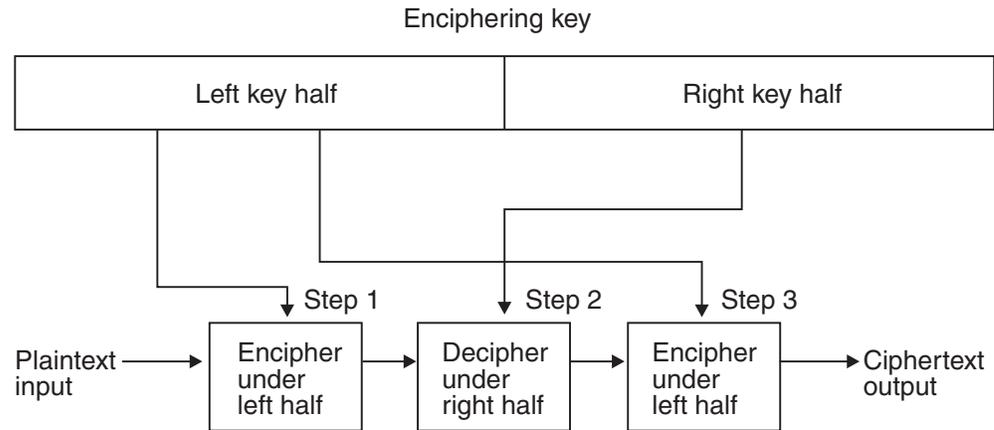


Figure 10. An Example of Multiple Encipherment

Triple DES data encryption uses multiple encipherment with either double-length or triple-length DATA keys to protect data. For this procedure the data is first enciphered using the first DATA key. The result is then deciphered using the second DATA key. When using a triple-length key, this second result is then enciphered using the third DATA key. When using a double-length key, the first DATA key is reused to encrytp the second result.

**Note:** Multiple decipherment is the inverse of multiple encipherment (decipher-encipher-decipher).

---

## Running in Special Secure Mode

Special secure mode is a special processing mode for the entry of clear keys. To perform the following tasks, you must enable Special Secure Mode:

- Use the secure key import or multiple secure key import callable service, which work with clear keys.
- Use the Clear PIN generate service which works with clear PINs.
- Use the symmetric key generate callable service with the IM keyword. Special Secure Mode is not required if a PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor is available and the modulus bit length is greater than or equal to 512 bits.
- Use KGUP to enter clear keys into the CKDS.

On the S/390 G6 Enterprise Server, S/390 Multiprise, IBM @server zSeries 800, and the IBM @server zSeries 900, special secure mode can be entered only if the hardware is enabled for special secure mode, and if the installation options data set allows it. With the optional TKE workstation, you can access the environmental control mask to enable or disable special secure mode.

On the IBM @server zSeries 990, special secure mode can be entered only if the installation options data set allows it. Additional hardware control for these callable services can be enforced with the optional TKE workstation. All access control points will be enabled on the PCI X Cryptographic Coprocessor, even those customers converting from TKE 3.1 to TKE 4.0. This is the first communication with a PCIXCC and there is no indication that the PCIX was being controlled by a

previous TKE. The default role would be in affect. Only DKYGENKY-DALL and DSG ZERO-PAD unrestricted hash length will be disabled.

For customers without a TKE, the hardware control for services is always enabled, with the exception of DKYGENKY-DALL and DSG ZERO-PAD unrestricted hash length. The same is true for new TKE customers.

## Cryptographic Key Data Set (CKDS)

ICSF stores DES keys in a specialized data set called a cryptographic key data set (CKDS). ICSF maintains both a disk copy and an in-storage copy of the CKDS. This makes it possible to refresh the cryptographic keys without interrupting the application programs. ICSF provides a sample CKDS allocation job (member CSFCKDS) in SYS1.SAMPLIB. For information on managing and sharing the CKDS in a sysplex environment, see *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521.

ICSF updates the CKDS at the following times:

- When you use KGUP to generate keys or enter keys into the system, ICSF updates the disk copy, rather than the in-storage copy. ICSF does not require that you stop cryptographic functions before updating the CKDS, unlike PCF.
- When you change the master key, ICSF enables you to reencipher the disk copy of the CKDS. ICSF then automatically refreshes the in-storage copy of the CKDS with the re-enciphered keys.
- When you convert a PCF CKDS to an ICSF CKDS, the PCF conversion program updates the disk copy of the ICSF CKDS.
- When an application uses the dynamic CKDS update callable services, both the disk copy and in-storage copy of the CKDS are dynamically updated.
- When a key or key part is imported from the TSO panel, the key part is entered using the TKE operational key entry (on G6, IBM @server zSeries 800, and IBM @server zSeries 900 servers).

ICSF allows these operations without interrupting cryptographic functions that are used by application programs.

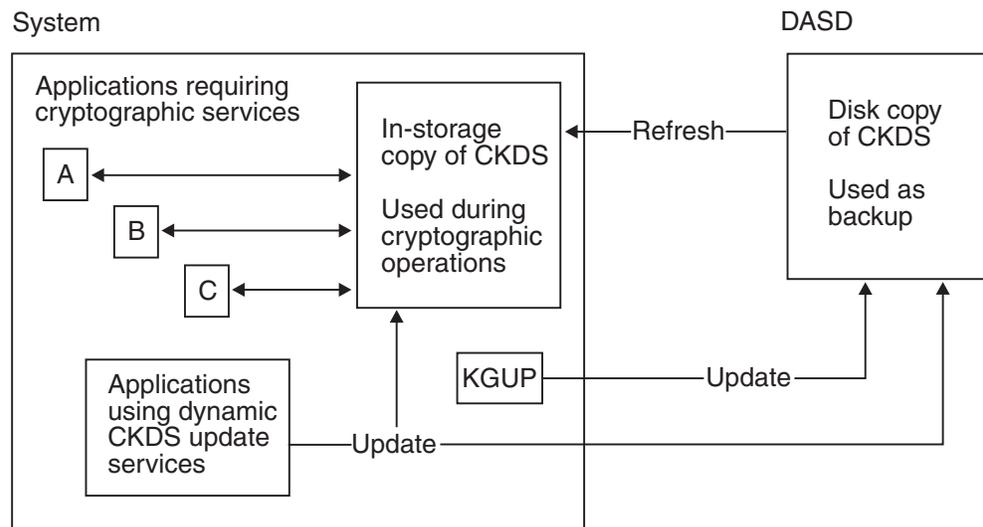


Figure 11. How the Cryptographic Key Data Set Is Maintained and Used

Callable services use the in-storage copy of the CKDS. For example, in Figure 11 on page 32 applications A, B, and C might make many calls for services that require the CKDS. Having the CKDS in storage avoids time-consuming I/O to a data set that is stored on DASD.

KGUP updates the disk copy rather than the in-storage copy. The ICSF administrator can then use the ICSF panel dialog or a batch job to refresh the in-storage CKDS with the updated disk copy of the CKDS. Cryptographic functions do not have to stop while KGUP updates the CKDS.

The dynamic CKDS update callable services permit an application to perform dynamic update of both the disk copy and the in-storage copy of the CKDS.

## Dynamic CKDS Update Callable Services

The dynamic CKDS update callable services allow applications to directly manipulate both the in-storage copy and the DASD copy of the CKDS. These callable services have the identical syntax as the 4753-HSP *verbs* of the same name. Key management applications that use these common callable services, or verbs, can be run on either system without change. Cryptographic functions do not have to stop while the dynamic CKDS update callable services update the CKDS.

---

## PKA Cryptographic Key Data Set

You can store RSA and DSA public and private keys in a specialized external VSAM data set that is called a public key data set (PKDS). ICSF provides a sample PKDS allocation job (member CSFPKDS) in SYS1.SAMPLIB. For information on managing and sharing the PKDS in a sysplex environment, see *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521.

PKDS initialization support is available on the Master Key Management panel, CSFPUTIL utility, and PassPhrase Initialization. It is a requirement if installing HCR770A or later.

Beginning in z/OS V1 R2, support to REENCIPHER PKDS and ACTIVATE PKDS has been added to the Master Key Management Panels and to the new CSFPUTIL utility. CSFPUTIL is a new utility that performs the same functions as REENCIPHER PKDS and ACTIVATE PKDS. These functions allow you to reencipher the PKDS from the old asymmetric-keys master key to the current master key and activate the reenciphered PKDS. Other systems with lower levels of ICSF which are sharing the PKDS would disable PKDS read and PKDS write and activate the reenciphered PKDS. For information on managing and sharing the PKDS in a sysplex environment, see *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521.

Toleration APAR OW49386 is required on the following systems in order to activate the re-enciphered PKDS:

- HCRP210 (standalone), HCRP220 (OS/390 V2 R6, OS/390 V2 R7, OS/390 V2 R8), HCRP230 (OS/390 V2 R9), and HCR7703 (OS/390 V2 R10 and z/OS V1 R1)

PKDSCACHE, an installation option, defines the size of the PKDS Cache in records. The PKDS cache improves performance as it facilitates access to frequently used records. Specify *n* as a decimal value from 0 to 256. If *n* is zero, no

cache will be implemented. If PKDSCACHE is not specified, the default value is 64. PKDSCACHE can be implemented on OS/390 V2 R10 and z/OS V1 R1 by installing APAR OW48568.

## Dynamic PKDS Update Callable Services

ICSF provides dynamic PKDS update callable services that permit an application to create, read, write, and delete PKDS records. You do not need to stop cryptographic functions while applications use these services to update the PKDS.

---

## Key Generator Utility Program and Key Generate Callable Service

With ICSF, you can use either the key generator utility program (KGUP) or the key generate callable service to generate DES keys.

With KGUP, you can generate key-encrypting keys, PIN keys, data-encrypting keys, data-translation keys, and MAC keys. A master key variant enciphers each type of key that KGUP creates. After this program generates a key, it stores it in the CKDS where it can be saved and maintained.

The key generate callable service creates all types of DES keys. It generates a single key or a pair of keys. Unlike KGUP, however, the key generate service does not store DES keys in the CKDS but returns them to the application program that called it.

---

## ANSI X9.17 Key Management Callable Services

ICSF supports the ANSI X9.17 key management standard, which defines a process for protecting and exchanging DES keys. The ANSI X9.17 standard uses the processes of notarization and offset to create a key identifier for both the sender and the receiver of a key. The key identifier includes a sequence number, and an ASCII-coded origin and destination identifier. The key identifier is cryptographically coupled with the key. In addition to providing callable services that support these functions, ICSF also defines and permits a process of partial notarization. You can use these callable services to develop ANSI X9.17 key management applications that exploit the offset, notarization, and partial notarization processes.

Offsetting combines an ANSI key-encrypting key (AKEK) with a counter by exclusive ORing the two values. The application initializes the counter the first time the key is used and increments the counter with each use. By checking the counter, applications can detect if a message has been transmitted out of sequence and take the appropriate action.

Notarization involves the coupling of an AKEK with ASCII character strings that contain identifiers for both origin and destination, and then offsetting the AKEK with a counter.

Partial notarization involves coupling the AKEK with the origin and destination ASCII character strings without offset. A partially notarized AKEK may be offset at a later time to form a fully notarized key.

**Restriction:** ANSI X9.17 services are not available on the IBM @server zSeries 990.

---

## Composing and Decomposing SET Blocks

ICSF provides callable services for developing SET applications that make use of the cryptographic hardware at the merchant and acquirer payment gateway. The SET Block Compose callable service performs DES encryption of data, OAEP-formatting through a series of SHA-1 hashing operations, and the RSA-encryption of the Optimal Asymmetric Encryption Padding (OAEP) block. The SET Block Decompose callable service decrypts both the RSA-encrypted and the DES-encrypted data.

---

## Exchanging Secure Sockets Layer Session Key Seed

Beginning in OS/390 V2 R3, ICSF provides two callable services that make it possible to exchange the seed key that the SSL application needs to generate session keys. The PKA encrypt callable service encrypts a supplied clear key value under an RSA public key. Currently, this service supports the PKCS 1.2 and ZERO-PAD formats. The PKA decrypt callable service decrypts the supplied key value using the corresponding RSA private key and returns the seed key value to the application in the clear. Currently, this service supports only the PKCS 1.2 format. The SSL application can then use the clear key value to generate symmetric session keys.



---

## Chapter 4. Using ICSF with Other Cryptographic Products

This chapter describes how ICSF works with other cryptographic products.

---

### Using IBM's Common Cryptographic Architecture

ICSF provides callable services that comply with IBM's Common Cryptographic Architecture (CCA). This allows application programs written in high-level languages such as C, COBOL, FORTRAN, and PL/I, as well as in Assembler, to be used under more than one cryptographic product.

Another family of products that provide these services is the IBM Transaction Security System. The Transaction Security System includes the IBM 4753 Network Security Processor MVS Support Program (4753-HSP), the IBM 4755 Cryptographic Adapter, and the IBM 4754 Security Interface Unit.

The PCI Cryptographic Coprocessor adds a high security environment to your OS/2, Windows NT, Windows 2000, OS/400, z/OS and OS/390 server systems for DES and RSA cryptographic functions and sensitive custom applications.

The PCI X Cryptographic Coprocessor adds a high security environment to your z/OS server systems for DES and RSA cryptographic functions and sensitive custom applications.

---

### Coexisting with Other IBM Cryptographic Products

ICSF can coexist simultaneously with other IBM cryptographic products within the same operating system image. This protects your installation's investment in programming skills and user applications, and provides a framework for migrating to ICSF.

### Running PCF Applications under ICSF

If your installation uses PCF, you can run PCF applications on ICSF. Your applications can benefit from the enhanced performance and availability of ICSF. Running PCF applications on ICSF allows you to test ICSF. ICSF also helps you migrate PCF applications. As soon as you can, you should convert these applications to use ICSF callable services rather than the PCF macros. This will permit you to change the master key without interrupting the converted applications.

**Restriction:** If you are using a PCF application, a PCI X Cryptographic Coprocessor is required if running on an IBM @server zSeries 990.

ICSF continues to support the PCF macros (GENKEY, RETKEY, EMK, and CIPHER). If an application uses these PCF macros, you can run the application on ICSF. The CIPHER macro will use the DES algorithm on a Cryptographic Coprocessor Feature that is configured for DES. The CIPHER macro will use the CDMF algorithm on a Cryptographic Coprocessor Feature that is configured for CDMF. If exits exist for either the GENKEY or the RETKEY macro, you should evaluate their applicability to ICSF. If your applications still need these exits, you must rewrite them for ICSF.

You can run PCF applications on systems with ICSF installed. How they run depends on the mode in which ICSF is running. You can run ICSF in any of the following modes:

- In **compatibility mode**, you can run PCF applications on ICSF without reassembling them, because ICSF supports the PCF macros.  
You cannot start PCF at the same time as ICSF on the same operating system.
- In **coexistence mode**, you can run a PCF application on PCF, or you can reassemble it to run on ICSF. ICSF provides coexistence macros for this purpose.  
You can start PCF at the same time as ICSF on the same operating system.
- In **noncompatibility mode**, you can run PCF applications only on PCF, and you can run ICSF applications only on ICSF. You cannot run PCF applications on ICSF, because ICSF does not support the PCF macros in this mode.  
You can start PCF at the same time as ICSF on the same operating system.

An application that is running under PCF may use a key in a CKDS managed by PCF. Before you run such an application on ICSF, you should convert the key to an ICSF format. ICSF provides a program to do this conversion.

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

## Running 4753-HSP Applications under ICSF

You can run both 4753-HSP and ICSF applications on the same z/OS operating system. This allows your installation to take advantage of the flexibility in PIN and key management systems provided by 4753-HSP and the high-volume throughput and bulk data-encryption available with ICSF.

**Restriction:** If you are using 4753-HSP applications, a PCI X Cryptographic Coprocessor is required if running on an IBM @server zSeries 990.

The key management callable services available in ICSF are identical to the key management *verbs* that are supported by the 4753-HSP. These common extensions beyond the CCA make it possible to develop significant key management applications that run without change on both systems. If your installation currently uses the 4753-HSP, you may be able to run your key storage management applications on ICSF without change if you have used these common verbs.

With the optional PCI Cryptographic Coprocessor feature, ICSF has the flexibility to route cryptographic functions to the PCI Cryptographic Coprocessor for processing. In OS/390 V2 R10 ICSF, ICSF provides support for new callable services which are supported by 4753-HSP (CSNBCPE, CSNBCVG, CSNBCVT, CSNBCVE, CSNBDKM, CSNBEPG, CSNBKTR, CSNBPEX) and enhances existing ICSF callable services to provide support for additional key types which are supported by 4753-HSP. This ICSF support will allow many 4753-HSP applications to run without change on ICSF.

There are some restrictions when running both ICSF and the 4753-HSP in the same operating system environment. Although both applications are capable of running in PCF compatibility mode, only one system can provide this service at any time. Because both ICSF and the 4753-HSP support the CCA callable services, applications need to be linked with the appropriate library routines to access the intended service. Note that internal key tokens are not interchangeable between the two products.

There are also some differences between the PKA implementations on z/OS ICSF and the 4753-HSP. The Transaction Security System family of products has two implementations of PKA: PKA92 and PKA96. Applications that are written to one

PKA implementation will not run on the other, and techniques that use RSA keys for DEA key distribution are incompatible between the two PKA versions. ICSF supports only the PKA96 and APIs are the same for the services that ICSF and the 4753-HSP have in common, with the following exceptions:

- The 4753-HSP does not support the Digital Signature Standard (DSS)

You can exchange RSA digital signatures between the two products if the digital signatures use ISO9796 formatting.

---

## Managing Keys with the Distributed Key Management System (DKMS)

The Distributed Key Management System provides a general key management facility to ICSF as well as to the Transaction Security System. DKMS automates the key management process, and exchanges and replaces keys on demand. Further, DKMS enforces key separation and maintains backup copies of all critical keys. DKMS provides key management support for a broad range of external devices with which cryptographic applications communicate, such as automated teller machines (ATMs) and point-of-sale terminals.

---

## Encrypting and Decrypting Information from Other Products

ICSF can exchange encrypted information with other cryptographic products. The only limitation is the form of DES encryption used. Some examples:

- MACs: ICSF supports both the ANSI standard X9.9, option 1, and the X9.19 optional double-MAC procedure for generating message authentication codes (MACs). Therefore, if a MAC has been generated with another product that uses either of these standards, ICSF can verify that MAC. ICSF provides support for the use of data-encrypting keys in both the MAC generating and verifying services. This support allows these services to interface more smoothly with non-CCA key distribution systems, including those that follow the ANSI X9.17 protocol.
- PINs: ICSF supports a wide variety of PIN block formats, as is shown in “Using Personal Identification Numbers (PINs) for Personal Authentication” on page 15.
- Data: ICSF can exchange encrypted data with other products that use the cipher block chaining (CBC) form of the DES algorithm.
- Keys: ICSF can exchange encrypted keys with other products that conform to the IBM's Common Cryptographic Architecture. If you need to exchange keys with systems that do not recognize transport key variants, ICSF enables you to encrypt selected keys under the transport key rather than under the transport key variant. You can use either an application program or KGUP to do this. ICSF can exchange RSA-encrypted data-encrypting keys with systems that format the key according to the PKCS 1.2 Standard.
- Digital Signatures: ICSF can exchange digital signatures with systems that support any of the following standards:
  - DSS
  - RSA signatures with MDC, SHA-1, or MD5 hashes and ISO-9796 formatting
  - RSA signatures with MD5 hashes and PKCS 1.0 or PKCS 1.1 formatting

---

## Virtual Telecommunications Access Method (VTAM) Session-Level Encryption

ICSF supports VTAM session-level encryption, which provides protection for messages within SNA sessions—that is, between pairs of logical units. Note that VTAM session level encryption will not work with HCR7708 on an IBM @server zSeries 990.

When VTAM session-level encryption is in effect, only the originating logical unit can encipher the data, and only the destination logical unit can decipher the data. Thus, the data never appears in the clear while passing through the network.

ICSF places no restrictions on the addressing mode of calling programs. In particular, when VTAM session-level encryption is used with ICSF, VTAM can use storage above 16 megabytes.

For information on setting up VTAM session-level encryption, refer to *VTAM Programming for LU 6.2*.

---

## Access Method Services Cryptographic Option

ICSF supports the Access Method Services Cryptographic Option. The option enables the user of the Access Method Services REPRO command to encipher data by using the Data Encryption Algorithm. The Access Method Services user can use REPRO to encipher data, write it to a data set, and then store the enciphered data set offline. When the user needs the enciphered data set, he or she can retrieve it and use REPRO to decipher it. The user can decipher the data either on the host processor where it was enciphered or on another host processor that contains the Access Method Services Cryptographic Option and the cryptographic key needed.

With the exception of catalogs, all data set organizations that are supported for input by REPRO are eligible as input for enciphering. Similarly, and with the same exception, all data set organizations supported for output by REPRO are eligible as output for deciphering. The resulting enciphered data sets are always sequentially organized (SAM or VSAM entry-sequenced data sets).

Cryptographic keys can either be created by ICSF or be supplied by the Access Method Services user.

---

## Using ICSF with BSAFE

ICSF works in conjunction with RSA Security, Inc.'s BSAFE toolkit (BSAFE 3.1 or later). If you are currently using applications developed with BSAFE, you may want to take advantage of the increased security and performance available with the zSeries and ICSF.

For increased security, you can use ICSF to generate and protect DES cryptographic keys. The keys are encrypted under the DES master key and stored in the Cryptographic Key Data Set. The DES encryption and decryption processes take place within the secure hardware boundary of the Cryptographic Coprocessor Feature.

You can also take advantage of the high transaction rates available with the cryptographic solution to increase the performance of DES encryption and decryption and hashing operations.



---

## Chapter 5. Planning for the Integrated Cryptographic Service Facility

This chapter contains guidelines and suggestions to help you plan the installation and operation of ICSF.

---

### System Requirements

The following sections describe the environment you need to use ICSF.

#### Operating System

ICSF is an element of z/OS. HCR770A is available for z/OS V1R4 and z/OS V1R2. HCR770A will be available for z/OS V1R3 and OS/390 V2R10 at a future date.

#### Cryptographic Hardware Features

This section describes the cryptographic hardware features available. Information on adding and removing cryptographic coprocessors can be found in *z/OS Cryptographic Services ICSF System Programmer's Guide*.

##### PCI X Cryptographic Coprocessor (PCIXCC)

The PCI X Cryptographic Coprocessor is an asynchronous cryptographic coprocessor. It is a replacement for the Cryptographic Coprocessor Feature and PCI Cryptographic Coprocessor. It is only available on a IBM @server zSeries 990.

The PCIXCC symmetric keys-master key is used in place of the CCF DES master key. The asymmetric keys-master key is used in place of the CCF signature and key management master keys. The PCIXCC supports 512 to 2048 bit RSA keys in all PKA services except SET services (Set Block Compose and Set Block Decompose).

This feature is in the process of being certified for Federal Information Processing Standard (FIPS) 140-2. This includes algorithmic certification under FIPS 46-2 (DES) and FIPS 180-1 (Secure Hash Standard).

##### CP Assist for Cryptographic Functions (CPACF)

CPACF is a set of new cryptographic instructions available on all CPs and improving performance. The SHA-1 algorithm is always available.

CP Assist for Cryptographic Functions (CPACF) DES/TDES Enablement, feature 3863, provides for clear key DES and TDES instructions. Software AES is supported by ICSF.

If you want to include a PCI X Cryptographic Coprocessor and/or a PCI Cryptographic Accelerator, then feature 3863 is required.

CPACF is only available on a IBM @server zSeries 990.

##### PCI Cryptographic Accelerator (PCICA)

On all systems, the PCI Cryptographic Accelerator provides support for clear keys in the CSNDPKD callable services for better performance. On z990, it also supports CSNDDSV and CSNDPKE.

A PCI Cryptographic Accelerator enables maximum SSL performance.

### **Cryptographic Coprocessor Feature (CCF)**

The Cryptographic Coprocessor Feature (CCF) can have up to two cryptographic coprocessors as high-speed extensions of the central processor. Each CCF contains both DES and PKA cryptographic processing units. You can configure the processor complex to run in either single-image mode or logical partition mode.

If the Cryptographic Coprocessor Feature is in single-image mode, the same master keys must be installed on both CCFs. If you bring a second coprocessor online, ICSF verifies that the master keys are the same. If the DES master keys are different, ICSF will not use the second coprocessor. The PKA master keys must be the same on both Coprocessors in order to enable the PKA services.

This feature is currently certified for Federal Information Processing Standard (FIPS) 140-1 level 4. This includes algorithmic certification under FIPS 46-2 (DES), FIPS 180-1 (Secure Hash Standard), and FIPS 186 (Digital Signature Standard).

The possible configurations include:

- DES with PKA  
These servers are configured for full 64-bit DES keys (effective length 56 bits), 1024-bit PKA keys for DES key distribution, and 1024-bit PKA signature keys.
- Triple DES with PKA  
These servers are configured for 192-bit DES keys (effective length 169 bits), 1024-bit PKA keys for DES key distribution, and 1024-bit PKA signature keys.  
This configuration is available on S/390 G5 Enterprise Servers and higher.

### **PCI Cryptographic Coprocessor**

The PCI Cryptographic Coprocessor, which works in conjunction with the Cryptographic Coprocessor Feature, provides the capability of generating and retaining RSA keys in secure hardware. This capability meets a requirement to become a SET Certificate Authority. A PCI Cryptographic Coprocessor is required for:

- Customer-written UDX capability
- Generating RSA public and private keys
- The retained key list and retain key delete callable service.
- See “Callable Services” on page 55 for other functions requiring a PCICC

The PCI cards are in addition to the Cryptographic Coprocessor Feature. In order for the PCI Cryptographic Coprocessor to operate, the verification pattern for the SYM-MK master key must match the verification pattern of the DES master key on the server's Cryptographic Coprocessor Feature. Before you can use the PKA services of the PCI Cryptographic Coprocessor, you must install both the KMMK and the SMK on the Cryptographic Coprocessor Feature and the ASYM-MK master key on the PCI Cryptographic Coprocessor. The hash pattern on the ASYM-MK master key must match the hash pattern on the SMK in order to use the PCI Cryptographic Coprocessor.

**Note:** For new installations, it is recommended that the installation enter the KMMK equal to the SMK master key. Existing customers on z/OS V1 R2 and above should reencipher their PKDS and migrate to a system with KMMK equal to SMK.

## Servers

This section describes the servers that the cryptographic hardware features are available on.

### IBM @server zSeries 990 (z990)

The IBM @server zSeries 990 provides constraint relief and addresses various customer demands. It has several cryptographic features.

- **CP Assist for Cryptographic Functions** are implemented on every processor. SHA-1 secure hashing is directly available to application programs.
- Feature code 3863, **CP Assist for Cryptographic Functions (CPACF) DES/TDES Enablement** – enables clear key DES and TDES instructions on all CPs. In addition, ICSF supports software implementation of AES.
- Feature code 0862, **PCI Cryptographic Accelerator** – optional, and only available if you have feature 3863, CPACF DES/TDES Enablement installed. The IBM @server zSeries 990 can support a maximum of 12 PCI Cryptographic Accelerators. Each feature code has two coprocessors.
- Feature code 0868, **PCI X Cryptographic Coprocessor** – optional, and only available if you have feature 3863, CPACF DES/TDES Enablement installed. The IBM @server zSeries 990 can support a maximum of 4 PCI X Cryptographic Coprocessors. Each feature code has one coprocessor.

**Note:** You can have a maximum of 6 PCICA features (12 cards) and 4 PCIXCC features (4 cards), with maximum of 8 features total.

### IBM @server zSeries 900 (z900) — Feature Code 800

You can enable the following features on this server:

- **Cryptographic Coprocessor Feature** – one or two cryptographic coprocessors protected by tamper-detection circuitry and a cryptographic battery unit.
- Feature code 0861, **PCI Cryptographic Coprocessor (PCICC)** – based on the 4758 model 2 standard PCI-bus card package. You must have at least one Cryptographic Coprocessor Feature on your system with a PCICC. Note that each feature has two coprocessors.
- Feature code 0862, **PCI Cryptographic Accelerator (PCICA)**. You must have at least one Cryptographic Coprocessor Feature on your system with a PCICA. Note that each feature has two coprocessors.

**Note:** The IBM @server zSeries 900 can support a combination of PCI Cryptographic Coprocessors (maximum of 16) or PCI Cryptographic Accelerators (maximum of 12), but the total must not exceed 16.

### IBM @server zSeries 800 (z800) — Feature Code 800

The following features are available on this server:

- **Cryptographic Coprocessor Feature (CCF)** – one or two cryptographic coprocessors protected by tamper-detection circuitry and a cryptographic battery unit.

- Feature code 0861, **PCI Cryptographic Coprocessor (PCICC)** – based on the 4758 model 2 standard PCI-bus card package. You must have at least one Cryptographic Coprocessor Feature on your system with a PCICC. Note that each feature has two coprocessors.
- Feature code 0862, **PCI Cryptographic Accelerator (PCICA)**. You must have at least one Cryptographic Coprocessor Feature on your system with a PCICA. Note that each feature code has two coprocessors.

**Note:** The IBM @server zSeries 800 can support a combination of PCI Cryptographic Coprocessors (maximum of 16) or PCI Cryptographic Accelerators (maximum of 12), but the total must not exceed 16.

### **IBM S/390 G6 Enterprise Server (G6) — Feature Code 800**

The following features are available on this server:

- **Cryptographic Coprocessor Feature** – one or two cryptographic coprocessors protected by tamper-detection circuitry and a cryptographic battery unit.
- **PCI Cryptographic Coprocessor (PCICC)** – based on the 4758 model 2 standard PCI-bus card package. You must have at least one Cryptographic Coprocessor Feature on your system with a PCICC. Feature code 0860 is needed for each PCI Cryptographic Coprocessor. Note that each feature code has one coprocessor.

---

## **Configurations by Server**

A central processor complex can have multiple cryptographic features of various types. This section describes some of the different configurations available with the various servers.

### **Configuring the IBM @server zSeries 990**

There is only LPAR mode on an IBM @server zSeries 990. You can divide your processor complex into PR/SM logical partitions. When you create logical partitions on your processor complex, you use the usage domain index on the Support Element Customize Image Profile page only if you have PCICAs and/or PCIXCCs or plan to add PCICAs/PCIXCCs to your system.

The DOMAIN parameter is optional. The number that is specified for the usage domain index must correspond to the domain number you specified with the DOMAIN(n) keyword in the installation options data set – if you specified one. The DOMAIN keyword is required if more than one domain is specified as the usage domain on the PR/SM panels.

The PCI Cryptographic Accelerator can be configured and shared across multiple partitions.

With the IBM @server zSeries 990, there is support for up to 30 LPARs. On previous systems, where the maximum number of LPARs was 16, a domain was unique to an LPAR. With more than 16 LPARs to support, the domain may not be unique across LPARs but the same domain may be assigned to different LPARs if they are accessing different PCICAs. This is illustrated by LPAR 1 and LPAR 3 in Figure 12 on page 47. They are both assigned to usage domain 0 but on two different PCICAs.

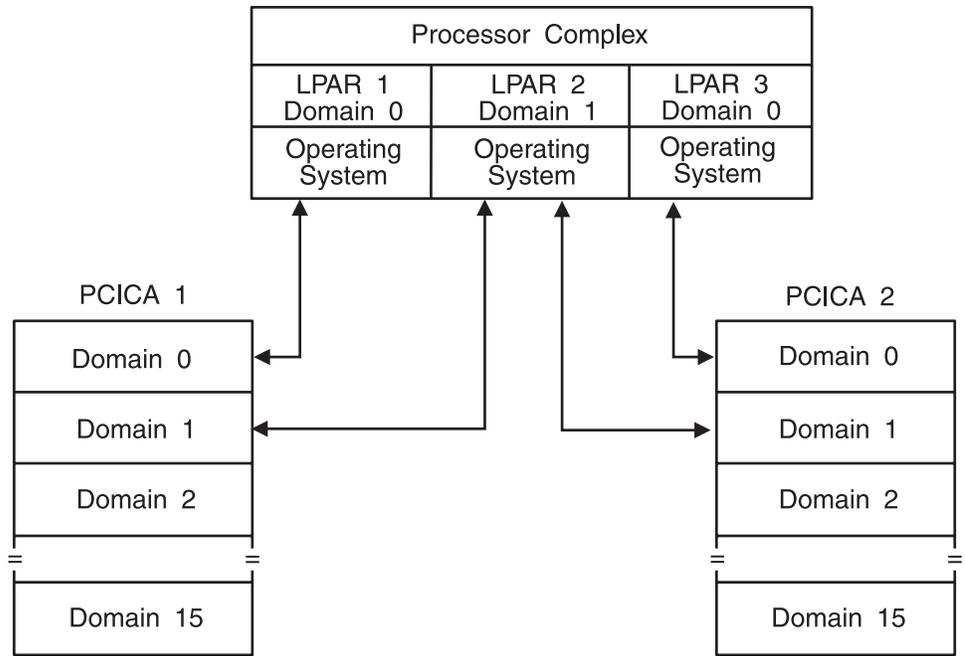


Figure 12. Two Crypto PCICAs on a Processor Complex Running in LPAR Mode

The example in Figure 12 shows that LPAR 2 has assigned access to Domain 1 on both PCICA 1 and PCICA 2. If you were to add another PCICA or PCIXCC to LPAR 2, Domain 1 on the new PCICA/PCIXCC would also be assigned.

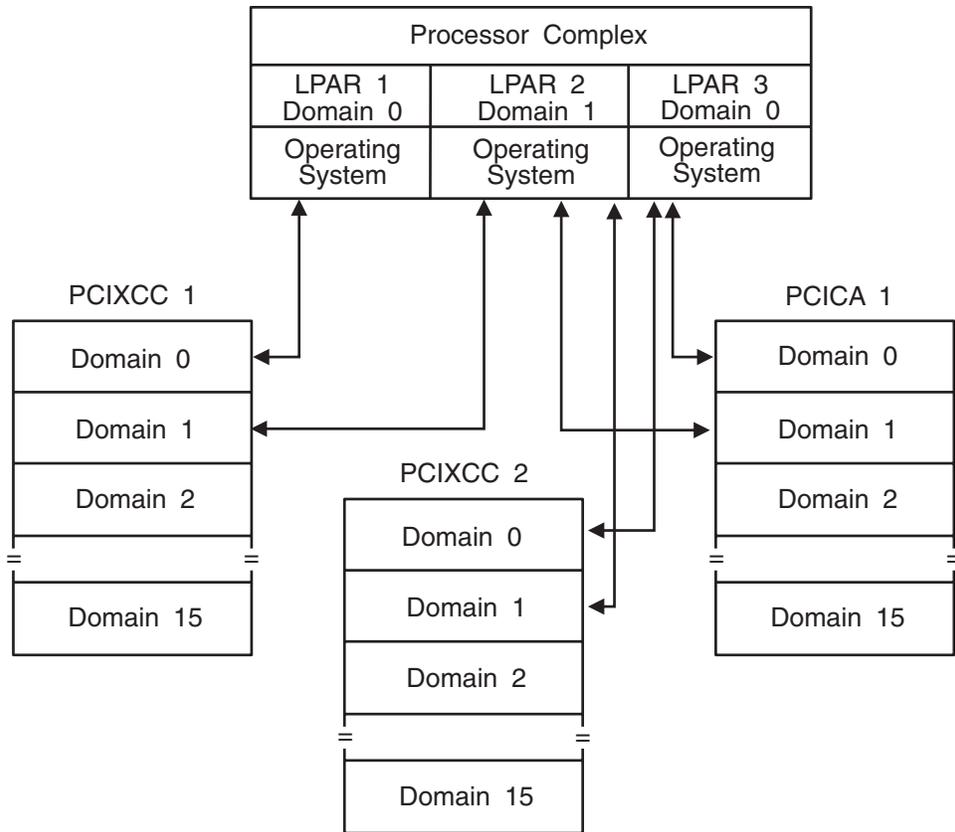


Figure 13. Multiple Crypto Coprocessors on a Complex Running in LPAR Mode

The example in Figure 13 shows that LPAR 2 has assigned access to Domain 1 on PCIXCC 1, PCIXCC 2, and PCICA 1. LPAR 3 has assigned access to Domain 0 on PCIXCC2 and PCICA 1.

## Configuring the S/390 Enterprise Servers, the S/390 Multiprise Server, the IBM @server zSeries 800 and the IBM @server zSeries 900

The Cryptographic Coprocessor Feature can include up to two cryptographic coprocessors, each of which is attached to a central processor within the central processor complex. Each cryptographic coprocessor has sixteen master key register sets, referred to as domains. ICSF uses the domain usage index to access each domain. You can configure the complex to run in one of the following modes:

- Single image mode
- Logical partition mode

### Single Image Mode

In single image mode, the processor complex has access to the same domain on both Crypto CP 0 and Crypto CP 4. The domain is specified in the installation options data set. Beginning in z/OS V1 R2, the DOMAIN parameter is optional. It is required if more than one domain is specified as the usage domain on the PR/SM panels or if running in native mode. See *z/OS Cryptographic Services ICSF System Programmer's Guide* for additional information on the DOMAIN parameter. The accessed domain on both coprocessors must have the same master key. Figure 14 on page 49 shows an example single image mode configuration.

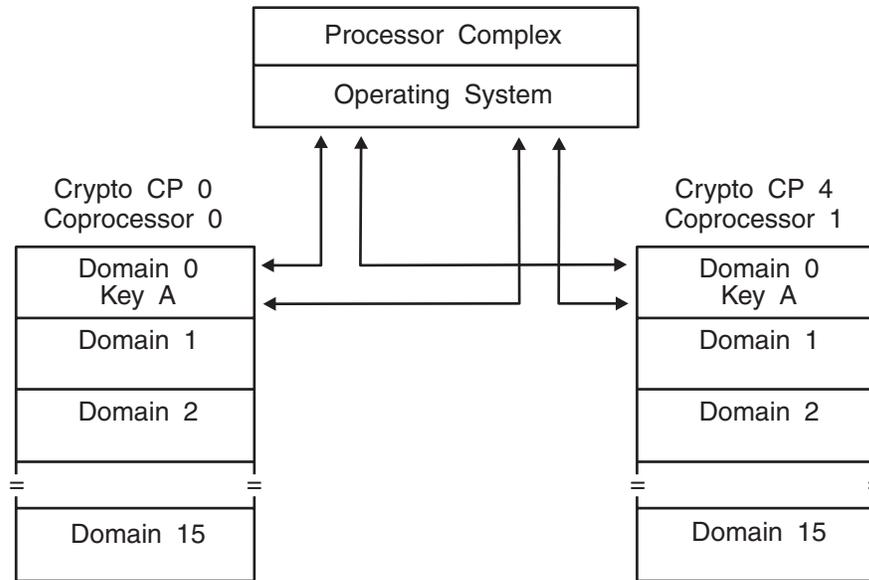


Figure 14. Two Crypto CPs on a Processor Complex Running in Single Image Mode

### Logical Partition (LPAR) Mode

You can divide your processor complex into PR/SM logical partitions (LPARs). When you create logical partitions on your processor complex, you use the usage domain index on the Support Element Customize Image Profile page to enable access to a Crypto CP domain. The number that is specified for the usage domain index must correspond to the domain number you specify with the DOMAIN(n) keyword in the installation options data set. Beginning in z/OS V1 R2, the DOMAIN parameter is optional. It is required if more than one domain is specified as the usage domain on the PR/SM panels or if running in native mode. See *z/OS Cryptographic Services ICSF System Programmer's Guide* for additional information on the DOMAIN parameter.

You can assign more than one domain to an LPAR, but you must use a unique installation options data set to define each domain. Assigning more than one domain to an LPAR makes it possible to have separate master keys for different purposes. For example, you might have one master key for production operations and a different master key for test operations.

The PCI Cryptographic Coprocessor can be configured like a Cryptographic Coprocessor Feature. It can be dedicated or shared across multiple partitions with each card supporting up to 16 domains.

The PCI Cryptographic Accelerator can be configured like a Cryptographic Coprocessor Feature. It can be dedicated or shared across multiple partitions with each card supporting up to 16 domains.

When using logical partitions, there is no domain sharing unless TKE is being used.

**Note:** The domain assigned to the TKE Host LPAR must be unique if TKE is to control all the PCIX cards in the environment. No other LPAR can use the domain assigned to the TKE Host.

The 'HOST' LPAR can control the domains of the other LPARS if the control domain for the first LPAR is setup for it. The example in Figure 15 on page 50 shows that

LPAR 1 has access to Domain 0 on Crypto CP 0, Crypto CP 1, and PCICC. LPAR 2 has access to Domain 1 and Domain 2 on both Crypto CPs and on the PCICC. LPAR 1 cannot access Domain 1 or Domain 2 on the PCICC or on either of the Crypto CPs. Likewise, LPAR 2 cannot access Domain 0 on either Crypto CP or the PCICC. The ICSF system running on the operating system in LPAR 2 has access to only one domain at any particular time, as specified in the installation options data set.

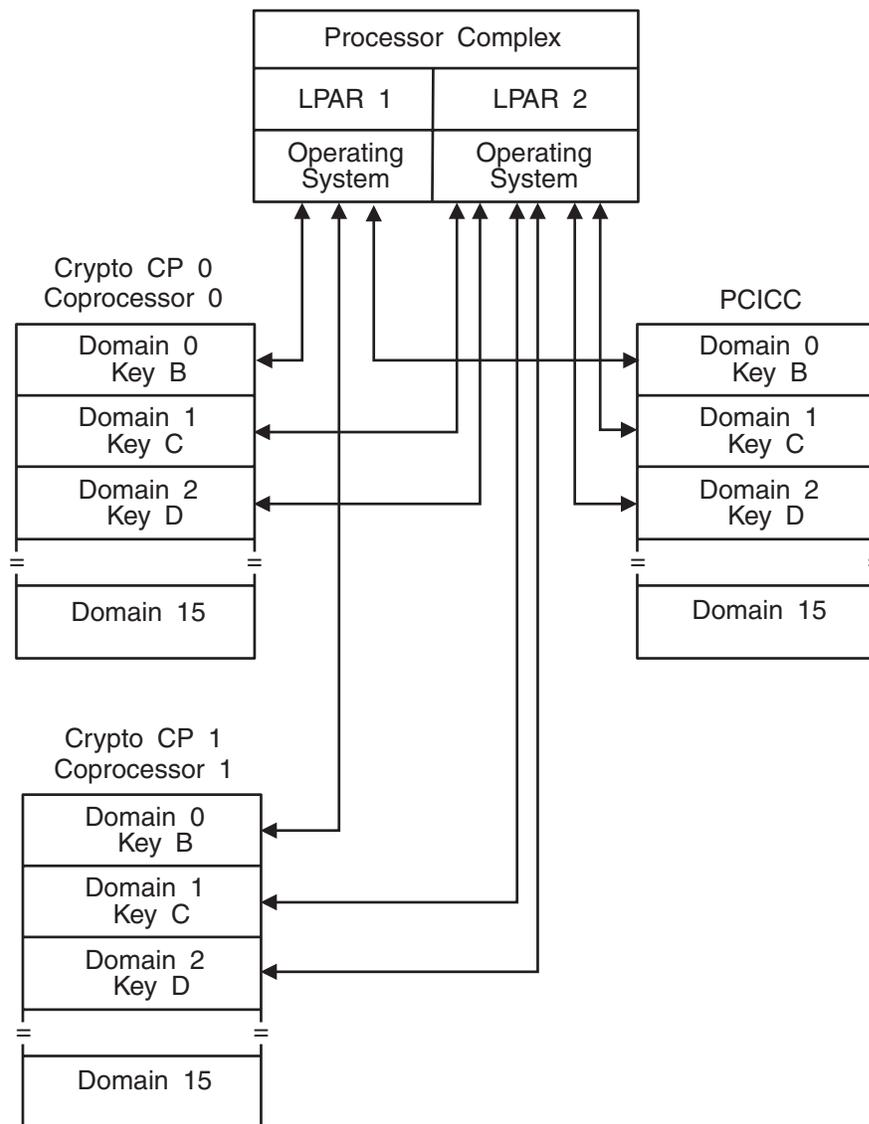


Figure 15. Two Crypto Coprocessors and one PCICC on a Processor Complex Running in LPAR Mode

## Hardware Features by Server

Configuration of the cryptographic features is dependent on U.S. Export controls. Common configurations are:

- **A. CCF only**

S/390 G6 Enterprise Server or S/390 Multiprise configured for full DES with PKA or G6 Enterprise Server or S/390 Multiprise configured for DES with exportable PKA.

- **B. CCF with PCICC**

S/390 G6 Enterprise Server, IBM @server zSeries 800, and IBM @server zSeries 900 configured for full DES with PCI Cryptographic Coprocessor.

- **C. PCIXCC**

IBM @server zSeries 990 with CP Assist for Cryptographic Functions (feature 3863) and PCI X Cryptographic Coprocessor (feature 0868).

For a complete list of callable services and the hardware configurations that support them, refer to Appendix B, “Summary of Callable Service Support by Hardware Configuration,” on page 67.

---

## Security

In reviewing your installation security plan before installing ICSF, consider the following points:

- **Controlling Access to Disk Copies of the CKDS**

You should determine which users and applications should have access to each copy of the CKDS on your system.

**Note:** The in-storage copy of the CKDS can be accessed only through ICSF functions such as callable services, KGUP, or the ICSF panels. To protect the in-storage copy of the CKDS, control who can use these services.

- **Controlling Access to the PKDS**

You should determine which users and applications should have access to the PKDS on your system.

- **Controlling Access to the Key Generator Utility Program (KGUP)**

- **Controlling Access to Services and Keys**

Anyone who is running the KGUP can read and change an unprotected CKDS. To prevent unauthorized persons from using the KGUP, store the program in an APF-authorized library that is protected by the Security Server (RACF).

Users of the Security Server (RACF) can use the CSFSERV and CSFKEYS classes to perform access checking and auditing of services and keys, respectively. The CSFSERV class also protects some critical administrative TSO panels, such as changing the master key and refreshing the CKDS. The audit records that are produced by these routines are SMF type 80 records.

- **Scheduling Changes for Cryptographic Keys**

To reduce the possibility of exposing a key value, you may want to change the value of cryptographic keys, including the DES master key, from time to time:

- You can use the ICSF panels to change the DES master key.
- If you have an optional Trusted Key Entry (TKE) workstation installed, you can use it to change DES and PKA master keys for both the Cryptographic Coprocessor Feature and the PCI Cryptographic Coprocessors.
- If you have an optional Trusted Key Entry (TKE) workstation installed and running TKE V4, you can change the SYM-MK and ASYM-MK on the PCIXCC.
- You can use KGUP or the ICSF panel to change the CKDS.
- You can develop applications that use the dynamic CKDS update callable services to change both the in-storage and DASD copies of the CKDS.

You can perform all of these operations without interrupting cryptographic functions.

- **Allowing or Preventing Clear Cryptographic Keys**

With ICSF, keys exist in the clear only in the following cases:

- If you specifically allow special secure mode *and* actually set special secure mode during operations, applications can use the Secure Key Import callable service, the Clear PIN Generate callable service, the Clear PIN Generate Alternate callable service and the Multiple Secure Key Import and Symmetric Key Generate with the IM keyword.

On a IBM @server zSeries 990 with a PCIXCC, access control points for these services must be enabled.

- If ICSF is not in special secure mode, all keys in the system are encrypted except DATA keys that a user may enter through the use of the clear key import callable service.

**Note:** The clear key import callable service is equivalent to the PCF EMK macro.

- The encode callable service can use a clear key to encipher data.
- If you use the Clear Master Key Entry panels to enter the key parts of a DES master key or a PKA master key on the S/390 G6 Enterprise Server, or higher, and the S/390 Multiprise, the key parts appear briefly in the clear in host storage.
- When an application calls the symmetric key generate callable service to generate a DATA key, the DATA key appears briefly in the clear in host storage. The DATA key is then quickly encrypted under the DES master key and the RSA public key.
- When an application calls the symmetric key import callable service to transfer a DATA key from encryption under an RSA public key to encryption under the host DES master key, the DATA key appears briefly in the clear in host storage.
- When an application calls the symmetric key export callable service to transfer a DATA key from encryption under the host DES master key to encryption under an RSA public key, the DATA key appears briefly in the clear in host storage.
- When an application calls the SET block compose and SET block decompose callable services, the DATA key exists briefly in the clear in host storage.
- On the IBM @server zSeries 990, clear keys are used to provide improved performance for the DES, TDES and SHA-1 algorithms. Symmetric key encrypt and decrypt services (CSNBSYD and CSNBSYE) are available on all CP's.

With PCI Cryptographic Coprocessor, the following services will be routed to the PCI Cryptographic Coprocessor if one is available: CSNDSYG, CSNDSYX, CSNDSBC, CSNDSYI, CSNDSBD, CSNBSKY, and CSNBSPN. If no PCI Cryptographic Coprocessor is available, then keys will appear briefly in the clear as stated above.

- **Sending Cryptographic Keys to Other Installations**

To eliminate the need to have a courier deliver clear keys between installations, you can use either or both of the following options:

- DES transport keys to encrypt keys for network distribution
- The receiving installation's RSA public key to encrypt a DES DATA key prior to electronic distribution

Both of these methods make key distribution more secure.

- **SMF Records Generated by ICSF**

ICSF generates type 82 records in the SMF data set when the following conditions occur:

- ICSF starts
- ICSF status changes on a processor
- When you enable or disable special secure mode
- When you enter either a DES or PKA clear master key part through the use of the ICSF panels on an S/390 G3 Enterprise Server, or higher, or an S/390 Multiprise
- When you enter a key part
- When the in-storage CKDS is refreshed
- When an application uses any of the dynamic services that write to the CKDS
- When ICSF handles error conditions or tampering
- When you issue a command from the TKE workstation to the Cryptographic Coprocessor Feature
- When an application uses any of the dynamic services that write to the PKDS
- When you use the Clear Master Key Entry panels to enter a master key in the PCI Cryptographic Coprocessor and PCI X Cryptographic Coprocessor
- When you create or delete a retained key on a PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor
- When you use the TKE workstation to communicate with the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor
- To capture measurements of timing and configuration for the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor

You can also use the Security Server (RACF) or an equivalent product to record attempts to use protected cryptographic keys or functions.

- **SMF Records Generated by PCF Macros**

When you use the GENKEY and RETKEY macros on PCF, the System Management Facilities (SMF) create SMF type 82 records. However, ICSF does not record SMF type 82 records during GENKEY and RETKEY macro processing. If you need to audit the use of these services, specify auditing in the resource profiles that protect them. If you run the PCF macros on ICSF, SMF records are *not* recorded, because ICSF does not record those SMF records for any macro or callable service.

Profiles can protect the GENKEY and RETKEY macros.

- **Recording and Formatting type 82 SMF Records in a Report**

Beginning in z/OS V1 R3, the following sample jobs are available (in SYS1.SAMPLIB) to assist in the recording and formatting of type 82 SMF data:

- **CSFSMFJ** - JCL that executes the code to dump and format SMF type 82 records for ICSF. Before executing the JOB step, you need to make modifications to the JCL (see the prologue in the sample for specific instructions). After the JCL has been modified, terminate SMF recording of the currently active dump dataset (by issuing I SMF) to allow for the unloading of SMF records. After SMF recording has been terminated, execute the JCL. The output goes into the held queue.
- **CSFSMFR** - An EXEC that formats the SMF records into a readable report.

---

## Operating Considerations

Before operating a computing system that has ICSF installed, you should consider certain items.

## ICSF Initialization Options

Your system operator can use the START and STOP operator commands to start and stop ICSF. Also, your system programmer can set up different sets of options such as a PARMLIB member that the operator can specify on the START command. This enables you to set ICSF up to run differently at different times. For more information, see “Using Options to Tailor ICSF” on page 20.

**Note:** To make such changes, PCF requires system programming skills to alter bits in an object module. ICSF is much easier. First, the system programmer creates alternative sets of options in data sets (such as PARMLIB members). The system operator can then use the STOP and START operator commands and these option data sets to change the ICSF operating mode.

## Effect of Multiple Records on Performance

If you add more than 10,000 records to a CKDS, and Local Shared Resource (LSR) is installed on your system, you should consider using the batch LSR subsystem with the VSAM deferred-write option. You should also plan to do sequential additions rather than insertions. This can greatly improve the performance of this operation.

## LPAR Considerations

You can divide your processor complex into PR/SM logical partitions (LPARs) by assigning crypto CP master key registers or domains to each LPAR. When running in LPAR mode, use system symbols in the installation options data set to define different domains. You can assign one or more domains to an LPAR.

Beginning in z/OS V1 R2, the DOMAIN parameter is an optional parameter in the installation options data set. It is required if more than one domain is specified as the usage domain on the PR/SM panels or if running in native mode. If you assign multiple domains to an LPAR, you can have separate master keys for different purposes. For instance, you might have one master key for production operations and another master key for test operations.

With an IBM @server zSeries 990 server, you are able to use the same domain in more than one LPAR as long as you're not sharing the same PCI X Cryptographic Coprocessor cards.

---

## Migration Activities

The following sections describe common activities and considerations that should be considered when you migrate from:

- z/OS V1 R1 and higher
- OS/390 V2 R10

Initially, HCR770A is only available on z/OS V1R2 and z/OS V1R4. HCR770A will be available for z/OS V1R3 and OS/390 V2R10 at a later date.

The PCI X Cryptographic Coprocessor is available on the following server:

- IBM @server zSeries 990

The Cryptographic Coprocessor Feature and PCI Cryptographic Coprocessor are available on the following servers:

- IBM @server zSeries 900

- IBM @server zSeries 800
- S/390 G6 Enterprise Server
- S/390 G5 Enterprise Server

If you are installing on a z990 server, see Appendix C, “IBM @server zSeries 990 with a PCI X Cryptographic Coprocessor,” on page 73 or Appendix D, “IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor,” on page 81.

## Callable Services

If you are running HCR770A on an IBM @server zSeries 990 with a PCI X Cryptographic Coprocessor, see “Callable services” on page 73 for the callable services that are available to you.

If you are running HCR770A on an IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor, see “Callable services” on page 81 for the callable services that are available to you.

Table 1. Summary of new and changed ICSF callable services

Callable service	Release	Description
PKA Encrypt (CSNDPKE)	HCR770A	<b>Changed:</b> Supports 2048-bit modulus keys for ZERO-PAD keyword. APAR OA03153 provides this function on z/OS V1.2, V1.3 and V1.4.
Encrypted PIN Verify (CSNBPVR)	HCR7708	<b>Changed:</b> <i>rule_array</i> keyword enhanced to support the VISAPVV4 keyword. APAR OA02575 provides this function on z/OS V1 R3 and z/OS V1 R4.
MAC generate (CSNBMGN and CSNBMGN1)	HCR7708	<b>Changed:</b> Supports longer text on a PCI Cryptographic Coprocessor
MAC verify (CSNBMVR and CSNBMVR1)	HCR7708	<b>Changed:</b> Supports longer text on a PCI Cryptographic Coprocessor
Symmetric Key Decipher (CSNBSYD)	HCR7708	<b>Changed:</b> Supports DES and TDES algorithms, but requires CP Assist for Cryptographic Functions on z990. <i>rule_array</i> key processing rules CUSP, IPS, and X9.23 have been added.
Symmetric Key Decipher (CSNBSYD1)	HCR7708	<b>New:</b> This callable service deciphers data in an address space or a data space using the cipher block chaining or electronic code book modes.
Symmetric Key Encipher (CSNBSYE)	HCR7708	<b>Changed:</b> Supports DES and TDES algorithms, but requires CP Assist for Cryptographic Functions on z990. <i>rule_array</i> key processing rules CUSP, IPS, and X9.23 have been added.
Symmetric Key Encipher (CSNBSYE1)	HCR7708	<b>New:</b> This callable service enciphers data in an address space or a data space using the cipher block chaining or electronic code book modes.
Control Vector Generate (CSNBCVG)	z/OS V1R3	<b>Changed:</b> <i>rule_array</i> keyword enhanced to support the UKPT keyword.
Encrypted PIN Translate (CSNBPTR)	z/OS V1R3	<b>Changed:</b> <i>rule_array</i> keyword enhanced to support the UKPT keywords UKPTIPIN, UKPTOPIN, UKPTBOTH
Encrypted PIN Verify (CSNBPVR)	z/OS V1R3	<b>Changed:</b> <i>rule_array</i> keyword enhanced to support the UKPT keywords UKPTIPIN
Key Token Build (CSNBKTB)	z/OS V1R3	<b>Changed:</b> <i>rule_array</i> keyword enhanced to support the UKPT keyword
Symmetric Key Decrypt (CSNBSYD)	z/OS V1R3	<b>New:</b> Provides support for AES. APAR OW52813 is available on HCR7704 (z/OS V1R2)

Table 1. Summary of new and changed ICSF callable services (continued)

Callable service	Release	Description
Symmetric Key Encrypt (CSNBSYE)	z/OS V1R3	<b>New:</b> Provides support for AES. APAR OW52813 is available on HCR7704 (z/OS V1R2)
Symmetric Key Export (CSNDSYX)	z/OS V1R3	<b>Changed:</b> <i>rule_array</i> keyword, PKCSOAEP, has been added. This keyword specifies the method found in RSA PKCS #1V2 OAEP. APAR OW50507 is available on HCR7703 (OS/390 V2R10 and z/OS V1 R1) and HCR7704 (z/OS V1 R2).
Symmetric Key Generate (CSNDSYG)	z/OS V1R3	<b>Changed:</b> <i>rule_array</i> keyword, PKCSOAEP, has been added. This keyword specifies the method found in RSA PKCS #1V2 OAEP. APAR OW50507 is available on HCR7703 (OS/390 V2R10 and z/OS V1 R1) and HCR7704 (z/OS V1 R2).
Symmetric Key Import (CSNDSYI)	z/OS V1R2	<b>Changed:</b> <i>rule_array</i> keyword, PKCSOAEP, has been added. This keyword specifies the method found in RSA PKCS #1V2 OAEP. APAR OW50507 is available on HCR7703 (OS/390 V2R10 and z/OS V1 R1) and HCR7704 (z/OS V1 R2).
Encipher (CSNBENC and CSNBENC1)	z/OS V1R2	<b>Changed:</b> MAXLEN parameter checking eliminated.
Decipher (CSNDBEC and CSNBDEC1)	z/OS V1R2	<b>Changed:</b> MAXLEN parameter checking eliminated.
MAC generate (CSNBMGN and CSNBMGN1)	z/OS V1R2	<b>Changed:</b> MAXLEN parameter checking eliminated.
MAC verify (CSNBMVR and CSNBMVR1)	z/OS V1R2	<b>Changed:</b> MAXLEN parameter checking eliminated.
Ciphertext translate (CSNBCTT and CSNBCTT1)	z/OS V1R2	<b>Changed:</b> MAXLEN parameter checking eliminated.
Diversified Key Generate (CSNBDBG)	z/OS V1R2	<b>New:</b> Generates a key based on the key-generating-key, the processing method, and the parameter supplied.
PKA Key Token Change (CSNDKTC)	z/OS V1R2	<b>New:</b> Changes the PKA internal key tokens (RSA and DSS) from encipherment with the old PCI Cryptographic Coprocessor ASYM-MK to encipherment with the current PCI Cryptographic Coprocessor ASYM-MK.
Secure Messaging for Keys (CSNBSKY)	z/OS V1R2	<b>New:</b> Encrypts a text block, including a clear key value decrypted from an internal or external DES token.
Secure Messaging for PINs (CSNBSPN)	z/OS V1R2	<b>New:</b> It encrypts a text block, including a clear PIN block recovered from an encrypted PIN block.

## 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.

With a PCI Cryptographic Coprocessor, modify and include:

- z/OS V1 R2: CSNBSKY, CSNBSPN, CSNDKTC, CSNBDBG

With a PCI X Cryptographic Coprocessor, modify and include:

- z/OS V1 R4: CSNBSKY, CSNBSPN, CSNDKTC, CSNBCKI, CSNBCKM, CSNBDEC, CSNBENC, CSNBKYTX, CSNBMGN, CSNBMVR, CSNBPEXX, CSNBDBG and CSNBRNG

**Note:** If no Wait List is specified on a IBM @server zSeries 990, sample CSFWTL01 will be used.

## CKDS

When sharing a CKDS with CCF systems and the IBM @server zSeries 990, the CKDS must be created on a CCF system.

Once new key types are added to the CKDS, the following considerations apply when sharing the CKDS with non-OS/390 R10 or non-z/OS systems:

- once keys with non-CCF control vectors are added to the CKDS, a CKDS reencipher operation must be invoked from a system which has a PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessors installed.
- once keys of type IMPORTER, EXPORTER, PINGEN, PINVER, IPINENC, or OPINENC which have non-CCF control vectors are added to the CKDS, a toleration APAR OW43926 must be installed on the pre-OS/390 V2 R10 ICSF systems. The APAR ensures that ICSF services will fail a request to use a key which contains a non-CCF control vector.
- after you change a CKDS to contain the ANSI X9.17 enablement keys, all instances of ICSF/MVS Version 1 Release 2 that share that CKDS must have PTF UW90181 installed. This PTF was shipped against ICSF/MVS Version 1 Release 2 and was rolled up into Version 2 Release 1.
- once a CKDS is modified to contain the ANSI X9.17 enablement keys, all instances of ICSF/MVS that share the CKDS must have APAR OW13633 installed.
- you can share a CKDS that contains a limited authority importer key. However, OS/390 ICSF or ICSF/MVS 2.1 running on S/390 Enterprise Servers and S/390 Multiprise servers must perform any CKDS reencipherment.
- once new system keys, double-length MAC keys, IMP-PKA keys, etc. (introduced in ICSF/MVS 2.1) are added to the CKDS, it is sharable with instances of ICSF/MVS which do not support these keys. A CKDS reencipher operation must be performed on a system which supports these key types.

## Installation Options Data Set

- CKTAUTH - decides if authentication will be performed for every CKDS record read from DASD. With a large CKDS, CKTAUTH(YES) could cause an impact to performance.
- DOMAIN - this parameter is optional with z/OS V1.2, V1.3 and V1.4. It is required if more than one domain is specified as the usage domain on the PR/SM panels or if running in native mode.  

On an IBM @server zSeries 990, if a PCI Cryptographic Accelerator or PCI X Cryptographic Coprocessors is not available, the DOMAIN parameter is not required. If a PCICA or PCIXCC is available, the DOMAIN parameter is required if more than one domain is specified as the usage domain on the PR/SM panels. If only one usage domain is assigned to the LPAR, the DOMAIN parameter is optional.
- PKDSCACHE - defines the size of the PKDS Cache in records. The PKDS cache improves performance as it facilitates access to frequently used records. Specify *n* as a decimal value from 0 to 256. If *n* is zero, no cache will be implemented. If PKDSCACHE is not specified, the default value is 64.

## Key Tokens

- On the z990, all RSA internal tokens are usable on the PCI X Cryptographic Coprocessor if they are encrypted under the current ASYM-MK.
- Existing DES internal key tokens can be used on either the Cryptographic Coprocessor Feature or the PCI Cryptographic Coprocessor.
- An existing PKA internal token created for the Cryptographic Coprocessor Feature cannot be used on the PCI Cryptographic Coprocessor unless you recreate it by reimporting the key. Since the Cryptographic Coprocessor Feature cannot generate PKA keys (these tokens were all generated on another platform and imported for use with the Cryptographic Coprocessor Feature), you'll need to reimport them to use them on the PCI Cryptographic Coprocessor. For maximum flexibility, you should install the SMK to be equal to the KMMK. Existing PKA tokens should then be reimported.

## PCI Cryptographic Accelerator

If you have a PCI Cryptographic Accelerator online, toleration APAR OW49402 is required on lower levels of ICSF (OS/390 V2 R9, OS/390 V2 R10 and z/OS V1 R1). Without this APAR, ICSF will abend with an X'18F' reason code 50.

## PKDS

Beginning with HCR770A, the PKDS must be initialized.

Beginning in z/OS V1 R2, support to REENCIPHER PKDS and ACTIVATE PKDS has been added to the Master Key Management Panels and to the new CSFPUTIL utility. CSFPUTIL is a new utility that performs the same functions as REENCIPHER PKDS and ACTIVATE PKDS. These functions allow you to reencipher the PKDS from the old asymmetric-keys master key to the current master key and activate the reenciphered PKDS. Other systems with lower levels of ICSF which are sharing the PKDS would disable PKDS read and PKDS write and activate the reenciphered PKDS. For information on managing and sharing the PKDS in a sysplex environment, see *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521. Toleration APAR OW49386 is required on the following systems in order to activate the re-enciphered PKDS:

- HCRP210 (standalone) and HCR7703 (OS/390 V2 R10 and z/OS V1 R1)

With OS/390 V2 R10 ICSF and above (including z/OS), if you share the PKDS with HCRP210 (ICSF/MVS V2 R1 – standalone), the following APARS must be installed:

- APARS OW33234 and OW37623. Tokens on previous releases of ICSF will be handled as follows.
  - Retained key tokens contain a public key token. These public key tokens may be used in public key services such as Digital Signature Verify (CSNDDSV). These tokens may not be updated or deleted through the PKDS Record Write (CSNDKRW) or PKDS Record Delete (CSNDKRD) callable services.
  - All modulus-exponent form RSA internal key tokens imported or created on an OS/390 V2 R9 ICSF or OS/390 V2 R10 ICSF system and above (including z/OS) with PCI Cryptographic Coprocessor will have a private section identifier of X'06'. These tokens will be converted where possible to internal tokens with a private section of X'02' for use on previous levels of ICSF without a PCI Cryptographic Coprocessor. Modulus-exponent tokens with a private section identifier of X'06' which are signature-use only tokens can be converted since these tokens are encrypted under the ASYM-MK of the PCI Cryptographic Coprocessor (which is the same as the SMK of the Cryptographic Coprocessor Feature). Modulus-exponent tokens with a private

section identifier of X'06' which are designated as key-management usage can only be converted for use on previous levels of ICSF if the KMMK on the Cryptographic Coprocessor Feature is the same as the SMK.

- CRT tokens are not supported on previous levels of ICSF.

For additional information on ME and CRT tokens, see diagnosis reference information in *z/OS Cryptographic Services ICSF System Programmer's Guide*.

## Resource Manager Interface (RMF)

Beginning in z/OS V1 R3, support to enable RMF to provide performance measurements on the following selected ICSF services and functions that use Direct Access Crypto (DAC) CCF instructions has been added. Support to enable RMF is also available on z/OS V1 R2 through APAR OW51003.

- Encipher (CSNBENC)
- Decipher (CSNBDEC)
- MAC Generate (CSNBMGN)
- MAC Verify (CSNBMVR)
- One-Way Hash (CSNBOWH)
- PIN Translate (CSNBPTR)
- PIN Verify (CSNBPVR)

## Special Secure Mode

Use of some ICSF services (CSNBSKI, CSNBSKM, CSNBPGN, CSNDSYG with the IM keyword) requires that ICSF be in special secure mode.

**Note:** If a PCI X Cryptographic Coprocessor or PCI Cryptographic Coprocessor is available and the modulus bit length of the RSA public key is greater than or equal to 512 bits, then special secure mode is not required for CSNDSYG IM form.

## TKE Workstation

The TKE workstation (Version 3 or higher) uses the IBM 4758 card. If you have a TKE workstation (Version 3 or higher) that connects to a host system running z/OS V1 R1 or lower, install APAR OW46381 on the system(s) running releases prior to z/OS V1 R2 to ensure proper communications between the TKE workstation and ICSF. For detailed information on these changes, refer to *z/OS Cryptographic Services ICSF TKE Workstation User's Guide*.

### TKE Version 3.1 and Access to Callable Services

Access to services that are executed on the PCI Cryptographic Coprocessor is through Access Control Points in the DEFAULT Role. To execute callable services on the PCI Cryptographic Coprocessor, access control points must be enabled for each service in the DEFAULT Role. The ability to enable/disable access control points in the DEFAULT Role was introduced on OS/390 V2 R10 through APAR OW46381 for the Trusted Key Entry Workstation. For systems that do not use the optional TKE Workstation, all access control points (current and new) are enabled in the DEFAULT Role with the appropriate microcode level on the PCI Cryptographic Coprocessor.

New TKE users and non-TKE users have all access control points enabled. This is also true for brand new TKE V3.1 users (not converting from TKE V3.0).

**Note:** Access control point DKYGENKY-DALL is always disabled in the DEFAULT Role for all customers (TKE and Non-TKE). A TKE Workstation is required to enable this access control point for the Diversified Key Generate service.

For existing TKE V3.0 users, upgrading to TKE V3.1 (APAR OW46381 and its corresponding ECA), current access control points in the DEFAULT Role are enabled. Any new access control points are disabled in the DEFAULT Role and must be enabled through TKE if the service is required.

**Notes:**

1. APAR OW46381 will update the TKE Host Code (z/OS V1 R1 and below)
2. ECA 186 will update the TKE Workstation Code
3. The latest or most current driver is required for the PCI Cryptographic Coprocessor microcode for the S/390 G5 Enterprise Server, S/390 G6 Enterprise Server, IBM @server zSeries 800, and IBM @server zSeries 900

All of the above components are required for complete access control point support.

Access to services which execute on the Cryptographic Coprocessor Feature is through SAF. Disablement through SAF is sufficient to prevent execution of a service by the Cryptographic Coprocessor Feature, the PCI Cryptographic Coprocessor or the PCI X Cryptographic Coprocessor. For functions which can be executed on the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor, enablement of the function requires that the function be enabled through SAF and through the access control point in the DEFAULT Role. For additional details, see “TKE Version 4 and Access to Callable Services” on page 61.

If you are on OS/390 V2 R10, using a TKE V3.0 workstation, access control points for services requiring APARs OW46380 and OW46382 will be disabled. Existing access control points will be enabled in the DEFAULT Role. APAR OW46381 must be installed to enable the OS/390 V2 R10 interface. This will allow the TKE Administrator to enable any new access control points for ICSF services that execute in the PCI Cryptographic Coprocessor under the DEFAULT Role.

Access Control Points (requiring APARs OW46380 and OW46382) for OS/390 V2 R10 are:

- DATAM Key Management Control

**Note:** For existing TKE installations (upgrading to TKE V3.1), it is required that this access control point be enabled. Failure to do so will result in processing errors for Double MAC keys in Key Import, Key Export, and Key Generate.

- Diversified Key Generate - Single length or same halves
- Diversified Key Generate - CLR8-ENC
- Diversified Key Generate - TDES-ENC
- Diversified Key Generate - TDES-DEC
- Diversified Key Generate - SESS-XOR
- Diversified Key Generate - DKYGENKY-DALL

**Note:** This access control point is always disabled in the DEFAULT Role for all customers (TKE and Non-TKE). A TKE Workstation is required to enable the function.

- MAC Generate - For existing TKE installations, it is recommended that this access control point be enabled.

- MAC Verify - For existing TKE installations, it is recommended that this access control point be enabled.

Access Control Points for z/OS V1 R2 are:

- PKA Key Token Change
- Secure Messaging for Keys
- Secure Messaging for PINs

Access Control Points for z/OS V1 R3 are:

- UKPT - PIN Verify, PIN Translate

Access Control Points for z/OS V1 R4 are:

- Data Key Export - Unrestricted
- Data Key Import - Unrestricted
- Key Export - Unrestricted
- Key Import - Unrestricted
- Key Part Import - Unrestricted

These access control points were added with APAR OW53666 for HCR7703, HCR7704, and HCR7706. They are in the base for HCR7708 so the APAR doesn't need to be installed. If the customer has TKE 3.0 or 3.1 and did not have the APAR installed on one of the previous levels, the access control points would need to be enabled to allow the services to work as they did in the previous release. If the APAR was installed, the state of the access control points would be saved and the customer would not need to do anything.

### **TKE Version 4 and Access to Callable Services**

Access to services that are executed on the PCI X Cryptographic Coprocessor is through Access Control Points in the DEFAULT Role. To execute callable services on the PCI X Cryptographic Coprocessor, access control points must be enabled for each service in the DEFAULT Role. For systems that do not use the optional TKE Workstation, all access control points (current and new) are enabled in the DEFAULT Role with the appropriate microcode level on the PCI X Cryptographic Coprocessor.

New TKE users and non-TKE users have all access control points enabled. This is also true for brand new TKE V4 users. For TKE V4.0, all access control points will be enabled on the PCIXCC cards, even if you're converting from TKE 3.1.

**Note:** Access control points DKYGENKY-DALL and DSG ZERO-PAD unrestricted hash length are always disabled in the DEFAULT Role for all customers (TKE and Non-TKE). A TKE Workstation is required to enable these access control points.

---

## **Migrating from 4753-HSP**

ICSF provides key management callable services that are identical to the 4753-HSP verbs of the same name. Key management applications that are developed for the 4753-HSP and use these common verbs can be run on OS/390 ICSF or z/OS ICSF without reassembly. You will, however, need to relink them.

If your installation is currently using the 4753-HSP and you are migrating to OS/390 ICSF or z/OS ICSF, consider the following:

- **4753-HSP cryptographic key storage**

Internal key tokens for ICSF and the 4753-HSP are not interchangeable. Key token migration for the 4753 exists through the optional TKE Version 3 Workstation for the S/390 G5, G6, z800 and z900 servers and TKE Version 4 for zSeries servers. TKE Version 3 and Version 4 supply a 4753 Migration Utility. It allows you to migrate internal DES key tokens from the 4753 to ICSF. Key exchange between the two systems is through the external key token. To migrate keys from the 4753-HSP to ICSF, you must first establish an exporter/importer key relationship between the 4753-HSP and ICSF. You can then write an application to export keys from the 4753-HSP key storage and import them into the ICSF CKDS. You can perform this type of key exchange only with CCA-defined keys, which have the same control vectors on key-encrypting keys. If your 4753-HSP installation includes non-CCA key types in key storage, you need to generate a special exporter/importer key-encrypting key pair on the 4753-HSP. The exporter key-encrypting key nullifies the CV value that is used on the 4753-HSP, and the importer key-encrypting key includes the CV value that is needed at ICSF.

- **Key labels**

ICSF/MVS Version 1 Release 2 and above supports an extended key label of up to 64 bytes. Although the 4753-HSP also supports a 64-byte key label, there are additional key label formatting restrictions that do not apply to ICSF. The 4753-HSP key label consists of one to five name tokens that are separated by periods. Each name token includes one to eight alphanumeric or national string characters. ICSF, therefore, can accept all 4753-HSP key labels, but the 4753-HSP cannot accept all ICSF key labels. For more information on key label formatting restrictions, refer to *IBM Transaction Security System: Concepts and Programming Guide: Volume I, Access Controls and DES Cryptography*.

ICSF/MVS Version 1 Release 2 and above, like the 4753-HSP, requires unique key labels for data-encrypting keys, data-translation keys, and MAC keys. To maintain compatibility with ICSF/MVS Version 1 Release 1, however, KGUP will continue to allow multiple key types per label for importer, exporter, and PIN keys under the following conditions. Use either KGUP or the KEU to enter the keys, and ensure that the key labels do not conflict with other unique label restrictions.

- **UDX (User Defined Extension) support**

Beginning with OS/390 V2 R10 ICSF, ICSF support is provided for UDX capabilities. UDX routines are developed by special contract with IBM and are only distributed to authorized customers.

The UDX function is invoked by an "installation-defined" or generic callable service. The callable service is defined in the Installation Options data set (UDX parameter) and the service stub is link-edited with the application. The application program calls the service stub which accesses the UDX installation-defined service.

There is a one-to-one correspondence between a specific generic service in ICSF and a specific UDX command processor in the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor. The administrator, through ICSF panels, performs UDX authorization processing on each PCI Cryptographic Coprocessor. Authorization is not LPAR specific. See *z/OS Cryptographic Services ICSF Administrator's Guide*, SA22-7521, for additional information on authorizing User Defined Extensions.

Beginning in z/OS V1 R2, support for writing your own UDX for a PCI Cryptographic Coprocessor has been added. (Development of a UDX for a PCIX Cryptographic Coprocessor requires a special contract with IBM.) See the *UDX Reference and Guide* and the *4758 Custom Software Developer's Toolkit Guide*

for additional information. These, and other publications related to the IBM 4758 Coprocessor can be obtained in PDF format from the Library page located at <http://www.ibm.com/security/cryptocards>.

See *z/OS Cryptographic Services ICSF System Programmer's Guide*, SA22-7520, for details on installation-defined callable services and a description of the UDX parameter in the installation options data set.



---

## Appendix A. Standards

The Cryptographic Coprocessor Feature, PCI Cryptographic Coprocessor, and ICSF provide support for the following International and USA standards (at least in part):

**ISO 8730**

*Banking—Requirements for Standard Message Authentication (Wholesale)*

**ISO 8731**

*Banking—Approved Algorithms for Message Authentication—Part 1: DES-1 algorithm*

**ISO 8732**

*Information Processing: Modes of Operation for a 64-bit Cipher Algorithm*

**ISO 9564**

*Personal Identification Number Management and Security Part 1—PIN Protection Principles and Technique*

**ISO 9796**

*Information Technology — Security Techniques — Digital Signature Scheme Giving Message Recovery*

**FIPS 46-2**

*Data Encryption Standard*

**FIPS 180-1**

*Secure Hash Standard*

**FIPS 186**

*Digital Signature Standard*

**FIPS 197**

*Advanced Encryption Standard*

**ANSI X3.92 - 1981**

*Data Encryption Algorithm*

**ANSI X3.106 - 1983**

*Modes of DEA Operation*

Two modes specified in this standard are supported:

1. Electronic Code Book (ECB) mode
2. Cipher Block Chaining (CBC) mode

**ANSI X9.8 - 1982**

*Personal Identification Number (PIN) Management and Security*

**ANSI X9.9 - 1986**

*Financial Institution Message Authentication (Wholesale)*

**ANSI X9.17 - 1985 (Reaffirmed 1991)**

*Financial Institution Key Management (Wholesale)*

**ANSI X9.19**

*Optional Double-MAC Procedure*

**ANSI X9.23 - 1988**

*Encryption of Wholesale Financial Messages*



---

## Appendix B. Summary of Callable Service Support by Hardware Configuration

The callable services available to your applications depend on the configuration of your server and cryptographic features. The configuration of the cryptographic features is dependent on U.S. Export Regulations. For information on the configurations available in your country, contact your IBM Marketing Representative.

In Table 2, letters represent various configurations according to the following table:

**Note:** Feature code 0800 is required of all the hardware except for the IBM @server zSeries 990.

- **A (CCF only)** — S/390 G6 Enterprise Server or S/390 Multiprise configured for full DES with PKA or G6 Enterprise Server or S/390 Multiprise configured for DES with exportable PKA
- **B (CCF with PCICC)** — S/390 G6 Enterprise Server, IBM @server zSeries 800, and IBM @server zSeries 900 configured for full DES with PCI Cryptographic Coprocessor
- **C (PCIXCC)** — IBM @server zSeries 990 with CP Assist for Cryptographic Functions and PCI X Cryptographic Coprocessor.

### Notes:

1. Support for the PCI Cryptographic Accelerator has been added in z/OS V1 R2 on the IBM @server zSeries 900. ICSF services exploiting the PCI Cryptographic Accelerator are PKA Decrypt (CSNDPKD) and PKA Encrypt (CSNDPKE) - ZERO-PAD formatting only.
2. CSNDDSV with clear keys will be routed to the PCICA on the IBM @server zSeries 990.
3. The following services are available on the IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor:
  - Character/Nibble Conversion (CSNBXBC and CSNBXCB)
  - Code Conversion (CSNBXEA and CSNBXAE)
  - Control Vector Generate (CSNBCVG)
  - Decode (CSNBDCO) - This service requires CP Assist for Cryptographic Functions.
  - Encode (CSNBECO) - This service requires CP Assist for Cryptographic Functions.
  - MDC Generate (CSNBMDG and CSNBMDG1) - This service requires CP Assist for Cryptographic Functions.
  - One-Way Hash Generate (CSNBOWH and CSNBOWH1)
  - PKA Decrypt (CSNDPKD) - This service requires a PCI Cryptographic Accelerator
  - PKA Encrypt (CSNDPKE) ZERO-PAD formatting only - This service requires a PCI Cryptographic Accelerator
  - PKA Key Token Build (CSNDPKB)
  - PKA Public Key Extract (CSNDPKX)
  - Symmetric Key Decipher (CSNBSYD and CSNBSYD1) - These services requires CP Assist for Cryptographic Functions for the DES, TDES and AES algorithms.

- Symmetric Key Encipher (CSNBSYE and CSNBSYE1) - These services requires CP Assist for Cryptographic Functions for the DES, TDES and AES algorithms.
- X9.9 Data Editing (CSNB9ED)

Table 2. Summary of ICSF Callable Services Support

Service Name	Function	A	B	C
ANSI X9.17 EDC generate	Generates an ANSI X9.17 error detection code on an arbitrary length string using the special MAC key (x'0123456789ABCDEF').	X	X	
ANSI X9.17 key export	Uses the ANSI X9.17 protocol to export a DATA key or a pair of DATA keys with or without an AKEK. Supports the export of a CCA IMPORTER or EXPORTER KEK. Converts a single DATA key or combines two DATA keys into a single MAC key.	X	X	
ANSI X9.17 key import	Uses the ANSI X9.17 protocol to import a DATA key or a pair of DATA keys with or without an AKEK. Supports the import of a CCA IMPORTER or EXPORTER KEK. Converts a single DATA key or combines two DATA keys into a single MAC key.	X	X	
ANSI X9.17 key translate	Uses the ANSI X9.17 protocol to translate, in a single service call, either one or two DATA keys or a single KEK from encryption under one AKEK to encryption under another AKEK. Converts a single DATA key or combines two DATA keys into a single MAC key.	X	X	
ANSI X9.17 transport key partial notarize	Permits the preprocessing of an AKEK with origin and destination identifiers to create a partially notarized AKEK.	X	X	
Character/nibble conversion	Converts a binary string to a character string or vice versa.	X	X	X
Ciphertext translate	Translates the user-supplied ciphertext from one key and enciphers the ciphertext to another key.	X	X	
Clear key import	Imports a clear DATA key, enciphers it under the master key, and places the result into an internal key token.	X	X	X
Clear PIN encrypt	Formats a PIN into a PIN block format (IBM 3621, IBM3624, ISO-0, ISO-1, ISO-2, IBM 4704 encrypting PINPAD, VISA 2, VISA 3, VISA 4, ECI 2, ECI 3) and encrypts the results.		X	X
Clear PIN generate	Generates a clear personal identification number (PIN), a PIN verification value (PVV), or an offset using one of the following algorithms: Interbank PIN (INBK-PIN) IBM 3624 (IBM-PIN or IBM-PINO) IBM German Bank Pool (GBP-PIN or GBP-PINO) VISA PIN validation value (VISA-PVV)	X	X	X
Clear PIN generate alternate	Generates a clear VISA PIN validation value (PVV) from an input encrypted PIN block.	X	X	X
Code conversion	Converts EBCDIC data to ASCII data or vice versa.	X	X	X
Control vector generate	Builds a control vector from keywords specified as input to the service.	X	X	X

Table 2. Summary of ICSF Callable Services Support (continued)

Service Name	Function	A	B	C
Control vector translate	Changes the control vector used to encipher an external key.		X	X
Cryptographic variable encipher	Encrypts plaintext using the Cipher Block Chaining (CBC) method.		X	X
Data key export	Converts a DATA key from operational form into exportable form.	X	X	X
Data key import	Imports an encrypted single-length or double-length DES data key and creates or updates a target internal key token with the master key-enciphered source key.		X	X
Decipher	Deciphers data using the cipher block chaining mode of the DES. <b>Note:</b> Triple DES decipherment is available only on the S/390 G4 Enterprise Server (with LIC level driver 98), or higher (with feature codes 0832, 0833, 0834, or 0835).	X	X	X
Decipher	Deciphers data using the CDMF mode of the DES.	X	X	
Decode	Decodes an 8-byte string of data using the electronic code book mode of the DES.	X	X	X
Digital signature generate	Generate a digital signature using a supplied hash and a private key.	X	X	X
Digital signature verify	Verifies a digital signature using the same supplied hash that was used to generate the signature and the public key that corresponds to the private key used to generate the signature.	X	X	X
Diversified key generate	Generates a key based on the key-generating key, the processing method, and the parameter supplied. The control vector of the key-generating key also determines the type of target key that can be generated.		X	X
Encipher	Enciphers data using the cipher block chaining mode of the DES. <b>Note:</b> Triple DES encipherment is available only on the S/390 G4 Enterprise Server (with LIC level driver 98), or higher (with feature codes 0832, 0833, 0834, or 0835).	X	X	X
Encipher	Enciphers data using the CDMF mode of the DES.	X	X	
Encode	Encodes an 8-byte string of data using the electronic code book mode of the DES.	X	X	X
Encrypted PIN generate	Generates and formats a PIN and encrypts the PIN block.		X	X
Encrypted PIN translate	Reenciphers a PIN block from one PIN-encrypting key to another and, optionally, changes the PIN block format.	X	X	X
Encrypted PIN verify	Verifies a supplied PIN using one of the following algorithms: Interbank PIN (INBK-PIN) IBM 3624 (IBM-PIN or IBM-PINO) IBM German Bank Pool (GBP-PIN or GBP-PINO) VISA PIN validation value (VISA-PVV)	X	X	X

Table 2. Summary of ICSF Callable Services Support (continued)

Service Name	Function	A	B	C
Key export	Converts any key from operational form into exportable form.	X	X	X
Key generate	Generates a 64-bit or 128-bit odd parity key, or a pair of keys, and returns them in encrypted forms.	X	X	X
Key import	Converts any key from importable form into operational form.	X	X	X
Key part import	Combines the clear key parts of an AKEK and returns the combined key value in an internal key token or an update to the CKDS.	X	X	X
Key record create	Adds a key record containing a key token set to binary zeros to both the in-storage and DASD copies of the CKDS.	X	X	X
Key record delete	Deletes a key record from both the in-storage and DASD copies of the CKDS.	X	X	X
Key record read	Copies an internal key token from the in-storage copy of the CKDS to application storage.	X	X	X
Key record write	Writes an internal key token to the CKDS record specified in the key label parameter. Updates both the in-storage and DASD copies of the CKDS currently in use.	X	X	X
Key test	Generates or verifies a secure verification pattern for keys. CSNBKYT requires the tested key to be in the clear or encrypted under the master key. CSNBKYTX also allows the tested key to be encrypted under a key-encrypting key.	X	X	X
Key token build	Builds an internal token from the supplied parameters.	X	X	X
Key translate	Uses one key-encrypting key to decipher an input key and then enciphers this using another key-encrypting key.		X	X
MAC generation	Generates a 4-, 6-, or 8-byte message authentication code (MAC) for a text string that the application program supplies. The MAC can be computed using either the ANSI X9.9-1 algorithm, the ANSI X9.19 optional double-MAC algorithm, or the EMV padding rules.	X	X	X
MAC verification	Verifies a 4-, 6-, or 8-byte message authentication code (MAC) for a text string that the application program supplies. The MAC is computed using either the ANSI X9.9-1 algorithm, the ANSI X 9.19 optional double-MAC algorithm, or the EMV padding rules and is compared with a user-supplied MAC.	X	X	X
MDC generation	Generates a 128-bit modification detection code (MDC) for a text string that the application program supplies.	X	X	X
Multiple Clear Key Import	Imports a clear DATA key of one, two, or three parts, enciphers it under the master key, and places the result into an internal key token.	X	X	X

Table 2. Summary of ICSF Callable Services Support (continued)

Service Name	Function	A	B	C
Multiple Secure Key Import	Enciphers a clear key under the master key or an IMPORTER KEK, and places the result into an internal or external key token as any key type. Permits the import of double-length DATA, MAC and MACVER keys and triple-length DATA keys.	X	X	X
One-way hash generate	Generates a one-way hash on specified text using the SHA-1 or MD5 method.	X	X	X
PCI Interface	Trusted Key Entry (TKE) workstation interface to the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor.		X	X
PKA decrypt	Decrypts an RSA-encrypted key value and returns it to the application in the clear.	X	X	X
PKA encrypt	Encrypts a PKCS 1.2 or ZERO-PAD formatted clear key value under an RSA public key to support Secure Sockets Layer (SSL) applications.	X	X	X
PKA key generate (DSS)	Generate a PKA internal token for use with the DSS algorithm in digital signature services.	X	X	
PKA key generate (RSA)	Generate a PKA internal token for use with the DSS algorithm in digital signature services.		X	X
PKA key import	Import a PKA key token.	X	X	X
PKA key token build	Create an external PKA key token containing an unenciphered private key.	X	X	X
PKA key token change	Changes PKA key tokens (RSA and DSS) from encipherment with the old PCI Cryptographic Coprocessor asymmetric-key master key to encipherment with the current signature/PCI Cryptographic Coprocessor asymmetric-key master key.		X	X
PKA public key extract	Extract a PKA public key from a supplied PKA internal or external private key token.	X	X	X
PKDS Record Create	Writes a new record to the PKDS	X	X	X
PKDS Record Write	Writes over an existing record in the PKDS	X	X	X
PKDS Record Read	Reads a record from the PKDS and returns the content of the record.	X	X	X
PKDS Record Delete	Deletes an existing record from the PKDS	X	X	X
PKSC interface	Trusted Key Entry (TKE) workstation interface.	X	X	
Prohibit export	Modifies an operational key so that it cannot be exported.		X	X
Prohibit export extended	Changes the external token of a key in exportable form so that it can be imported at the receiver node but not exported from that node.	X	X	X
Random number generate	Generates an 8-byte random number. The output can be specified in three forms of parity: RANDOM, ODD, and EVEN.	X	X	X
Retained key delete	Deletes a key that has been retained within a PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor.		X	X

Table 2. Summary of ICSF Callable Services Support (continued)

Service Name	Function	A	B	C
Retained key list	Lists the key labels of keys that have been retained within the PCI Cryptographic Coprocessor or PCI X Cryptographic Coprocessor.		X	X
Secure key import	Enciphers a clear key under the master key or an IMPORTER KEK, and places the result into an internal or external key token as any key type.	X	X	X
Secure messaging for keys	Encrypts a text block, including a clear key value decrypted from an internal or external DES token.		X	X
Secure messaging for PINs	Encrypts a text block, including a clear PIN block recovered from an encrypted PIN block.		X	X
SET block decompose	Compose the RSA-OAEP block and the DES-encrypted data block in support of the SET protocol.	X	X	X
SET block compose	Decompose the RSA-OAEP block and the DES-encrypted data block in support of the SET protocol.	X	X	X
Symmetric key decipher	Deciphers data in an address space or a data space using the cipher block chaining or electronic code book modes. This service is available on machines with triple-DES feature codes.	X	X	X
Symmetric key encipher	Enciphers data in an address space or a data space using the cipher block chaining or electronic code book modes. This service is available on machines with triple-DES feature codes.	X	X	X
Symmetric key generate	Generates a symmetric (DATA) key and returns it in two forms: encrypted under the DES master key and encrypted under a PKA public key.	X	X	X
Symmetric key import	Imports a symmetric (DATA) key enciphered under an RSA public key and enciphers it under the DES master key.	X	X	X
Symmetric key export	Transfers a symmetric (DATA) key from encryption under the DES host master key to encryption under an RSA public key.	X	X	X
Transform CDMF key	Changes a CDMF DATA key in an internal or external token to a transformed shortened DES key.	X	X	
User Derived Key	Generates a single- or double-length SESSION MAC key or updates an existing user derived key.	X	X	
VISA CVV generate	Generates a Card Verification Value (CVV) or Card Verification Code (CVC).	X	X	X
VISA CVV verify	Verifies a Card Verification Value (CVV) or Card Verification Code (CVC).	X	X	X
X9.9 data editing	Edits an ASCII text string according to the editing rules of ANSI X9.9-4.	X	X	X

---

## Appendix C. IBM @server zSeries 990 with a PCI X Cryptographic Coprocessor

For secure key cryptography, the IBM @server zSeries 990 server requires the optional feature 0868, PCI X Cryptographic Coprocessor (PCIXCC). Feature code 3863, CP Assist for Cryptographic Functions (CPACF) DES/TDES Enablement, must also be installed. The PCIXCC replaces the Cryptographic Coprocessor Feature (CCF) and the PCI Cryptographic Coprocessor (PCICC).

CP Assist for Cryptographic Functions and the optional PCI Cryptographic Accelerator (feature code 0862) are also available on the z990 server.

The PCIXCC symmetric-keys master key is used in place of the CCF DES master key. The asymmetric-keys master key is used in place of the CCF signature and key management master keys.

**Restriction:** The PCI X Cryptographic Coprocessor is not available on the S/390 G6 Enterprise Server, IBM @server zSeries 800, or IBM @server zSeries 900.

---

### Operating System Requirements

ICSF support for the PCI X Cryptographic Coprocessor is available in the z990 Cryptographic Support web deliverable (FMID HCR770A). It can be installed on z/OS V1.2 or z/OS V1.4. The web deliverable will support OS/390 V2.10 and z/OS V1.3 in the future.

---

### Applications and programs

Applications requiring secure cryptography using encrypted keys will be able to execute on the IBM @server zSeries 990 as long as the optional PCI X Cryptographic Coprocessor is also installed.

---

### Callable services

The following services are not available with a PCI X Cryptographic Coprocessor:

- ANSI X9.17 EDC Generate (CSNAEGN)
- ANSI X9.17 Key Export (CSNAKEX)
- ANSI X9.17 Key Import (CSNAKIM)
- ANSI X9.17 Key Translate (CSNAKTR)
- ANSI X9.17 Transport Key Partial Notarize (CSNAKTR)
- Ciphertext Translate (CSNBCTT)
- PKSC Interface Service (CSFPKSC)
- Transform CDMF Key (CSNBTK)
- User Derived Key (CSFUDK)

The following services have changed and are available with a PCI X Cryptographic Coprocessor:

| Table 3. Summary of new and changed ICSF callable services

Callable service	Release	Description
Clear Key Import (CSNBCKI)	HCR770A	<b>Changed:</b> No internal token markings for CDMF or DES; no token copying.
Clear PIN Generate (CSNBPGN)	HCR770A	<b>Changed:</b> <i>rule_array</i> keyword GBP-PINO is no longer supported. Format control in the PIN profile parameter must specify NONE.
Clear PIN Generate Alternate (CSNBCPA)	HCR770A	<b>Changed:</b> Format control in the PIN profile parameter must specify NONE.
Control Vector Generate (CSNBCVG)	HCR770A	<b>Changed:</b> Single- and double-length control vectors can be generated for MAC, MACVER, CIPHER, ENCIPHER and DECIPHER class keys.
Data Key Export (CSNBKX)	HCR770A	<b>Changed:</b> Token marking for DES/CDMF and key-encrypting keys are ignored.
Data Key Import (CSNBKIM)	HCR770A	<b>Changed:</b> Supports triple-length DATA keys. Token marking for DES/CDMF and key-encrypting keys are ignored.
Decipher (CSNBDEC and CSNBDEC1)	HCR770A	<b>Changed:</b> If keyword CDMF is specified or if the token is marked as CDMF, the service fails. Single- and double-length CIPHER and DECIPHER class keys are supported.
Digital Signature Generate (CSNDDSG)	HCR770A	<b>Changed:</b> Retained keys are supported. DSS tokens are not supported. The hash length limit for ZERO-PAD formatting is controlled by an access control point in the PCIXCC.
Digital Signature Verify (CSNDSV)	HCR770A	<b>Changed:</b> DSS tokens are not supported. It may execute on a PCICA, if available.
Encipher (CSNBENC and CSNBENC1)	HCR770A	<b>Changed:</b> If keyword CDMF is specified or if the token is marked as CDMF, the service fails. Single- and double-length CIPHER and ENCIPHER class keys are supported.
Encrypted PIN Translate (CSNBPTR)	HCR770A	<b>Changed:</b> Format control in the PIN profile parameter must specify NONE.
Encrypted PIN Verify (CSNBPVR)	HCR770A	<b>Changed:</b> <i>rule_array</i> keyword GBP-PINO is no longer supported. Format control in the PIN profile parameter must specify NONE.
Key Export (CSNBKEX)	HCR770A	<b>Changed:</b> DATA LAT and MACD keytypes are no longer supported. Token markings for DES/CDMF on DATA and KEKs are ignored. NOCV KEKs are supported by this service and the NOCV Exporter is controlled by a new access control point. Existing internal tokens with a MACIIMAC CV must be exported with either a TOKEN or DATAM key type. The external CV will be DATAM CV.
Key Generate (CSNBKGN)	HCR770A	<b>Changed:</b> DATA LAT key type not supported. Single and double length MAC, MACVER, CIPHER, ENCIPHER and DECIPHER keys can now be created.
Key Import (CSNBKIM)	HCR770A	<b>Changed:</b> DES and CDMF token markings are not made on DATA and key-encrypting keys, and are ignored on the IMPORTER key-encrypting key. Use of NOCV keys are controlled by an access control point in the PCIXCC. Creation of NOCV key-encrypting keys is only available for standard IMPORTERS and EXPORTERS. DATA LAT key type is no longer supported. Imported DATA C tokens will now have the same CV as external DATA C tokens. The export prohibited bit in the flag byte of the internal token is no longer used. The internal token will have the appropriate CV for export prohibit.
Key Part Import (CSNBKPI)	HCR770A	<b>Changed:</b> <i>rule_array</i> keywords ADD-PART and COMPLETE are added. New access control points are added for control of the new keywords.

Table 3. Summary of new and changed ICSF callable services (continued)

Callable service	Release	Description
Key Record Write (CSNBKRW)	HCR770A	<b>Changed:</b> DES and CDMF token markings are ignored. You can write NOCV keys to the CKDS without being in supervisor state.
Key Test (CSNBKYT)	HCR770A	<b>Changed:</b> Support added for generation and verification of triple length keys for the ENC-ZERO verification process. KEY-ENC and KEY-ENCD keywords can be used for triple length key tokens. No support for clear triple length keys.
Key Test Extended (CSNBKYTX)	HCR770A	<b>Changed:</b> Support added for generation and verification of single, double, and triple length keys for the ENC-ZERO verification process.
Key Token Build (CSNBKTB)	HCR770A	<b>Changed:</b> CDMF keyword not supported. AKEK and DATAXLAT keytype not supported.
MAC Generate (CSNBMGN and CSNBMGN1)	HCR770A	<b>Changed:</b> Text length greater than 4K is supported.
MAC Verify (CSNBMVR and CSNBMVR1)	HCR770A	<b>Changed:</b> Text length greater than 4K is supported.
Multiple Clear Key Import (CSNBCKM)	HCR770A	<b>Changed:</b> CDMF keyword will fail.
Multiple Secure Key Import (CSNBSKM)	HCR770A	<b>Changed:</b> DATAXLAT keytype is no longer supported. For DATAC keytype, the internal tokens will have the CCA compliant control vectors.  Creation of NOCV key-encrypting keys is only available for standard IMPORTERS and EXPORTERS. The NOCV IMPORTER access control point must be enabled to use the function.
PCI Interface (CSFPCI)	HCR770A	<b>Changed:</b> <i>rule_array</i> keyword XCPMASK will return online and active PCIXCCs on the system. Results are returned in the <i>masks_data</i> parameter and only for XCPMASK. PCIMASKS will return counts and masks of 0 on a z990 system.
PKA Encrypt (CSNDPKE)	HCR770A	<b>Changed:</b> ZERO-PAD requests are routed to a PCICA, if available. Execution on a PCIXCC is controlled by new access control points.
PKA Decrypt (CSNDPKD)	HCR770A	<b>Changed:</b> For clear RSA private keys, this service will be routed to the PCICA, if available, to provide optimal performance for SSL.
PKA Key Generate (CSNDPKG)	HCR770A	<b>Changed:</b> DSS keys will no longer be generated.
PKA Key Import (CSNDPKI)	HCR770A	<b>Changed:</b> DSS keys will no longer be imported.
PKA Key Token Build (CSNDPKB)	HCR770A	<b>Changed:</b> DSS key tokens can be created, but cannot be used in any other service.
PKA Key Token Change (CSNDKTC)	HCR770A	<b>Changed:</b> DSS key tokens are supported. In a shared PKDS environment, it may be necessary to reencipher on one system, rather than requiring the reencipher of the DSS token on a CCF system.
PKA Public Key Extract (CSNDPKX)	HCR770A	<b>Changed:</b> DSS key tokens are supported by this service, but cannot be used in any other service. Internal and external RSA tokens and PKDS labelnames are supported.
Prohibit Export (CSNBPEX)	HCR770A	<b>Changed:</b> MAC and MACVER keys are supported. Old internal DATAM and DATAMV are not supported. DATA keys are not supported.
Prohibit Export Extended (CSNBPEXX)	HCR770A	<b>Changed:</b> External MACD keys are not supported.

Table 3. Summary of new and changed ICSF callable services (continued)

Callable service	Release	Description
Secure Key Import (CSNBSKI)	HCR770A	<b>Changed:</b> DATAXLAT keytype is no longer supported. Special Secure Mode in the Options Data Set must be enabled. To create NOCV key-encrypting keys, token copying for standard IMPORTERs and EXPORTERs. Token copying is not supported for DES or CDMF flags. The NOCV IMPORTER access control point must be enabled to use the function.
Set Block Decompose (CSNDSBD)	HCR770A	<b>Changed:</b> The RSA private key used by this service does not need to be generated as a signature-only key.
Symmetric Key Generate (CSNDSYG)	HCR770A	<b>Changed:</b> The generated internal DATA key will not have any algorithm markings.
Symmetric Key Import (CSNDSYI)	HCR770A	<b>Changed:</b> Retained keys are supported. The imported internal DATA key will not have any algorithm markings.

Reason codes may be different when running on a PCIXCC (rather than a CCF). All the reason codes have been merged into one table in the *z/OS Cryptographic Services ICSF Application Programmer's Guide*.

## CKDS and PKDS (PCI X Cryptographic Coprocessor)

The PCI X Cryptographic Coprocessor eliminates the need for many of the system keys in the CKDS – namely the SYSTEM IMPORTER and EXPORTER keys, the NOCV dummy keys, the ANSI keys, and the ESYS keys. These system keys are not created on a z990 initialized CKDS.

If your CKDS was initialized on an IBM @server zSeries 990, it can not be used on a CCF system.

The PKDS must be initialized first before it can be used by callable services.

## ICSF Setup and Initialization

It is normal to see the following messages during the startup of ICSF on a IBM @server zSeries 990:

- Starting ICSF (HCR770A) on an IBM @server zSeries 990 without a PCI Cryptographic Accelerator, you will see the following messages:  

```
CSFM508I CRYPTOGRAPHY - THERE ARE NO CRYPTOGRAPHIC ACCELERATORS ONLINE
CSFM001I ICSF INITIALIZATION COMPLETE
```
- Starting ICSF (HCR770A) on an IBM @server zSeries 990 without any PCI X Cryptographic Coprocessors online, you will see the following messages:  

```
CSFM507I: "CRYPTOGRAPHY - THERE ARE NO PCI X CRYPTOGRAPHIC COPROCESSORS ONLINE
CSFM001I ICSF INITIALIZATION COMPLETE
```
- Starting ICSF on a IBM @server zSeries 990 with a PCI Cryptographic Accelerator. You'll receive message CSFM411I for each PCI Cryptographic Accelerator you have online.  

```
CSFM411I PCI CRYPTOGRAPHIC ACCELERATOR Ann IS ACTIVE
CSFM001I ICSF INITIALIZATION COMPLETE
```
- Starting ICSF on a IBM @server zSeries 990 with a PCI X Cryptographic Coprocessor. You'll receive message CSFM416I for each PCI X Cryptographic Coprocessor which has valid master keys.

```
CSFM416I BOTH MASTER KEYS CORRECT ON PCI X CRYPTOGRAPHIC COPROCESSOR Xnn,  
SERIAL NUMBER nnnnnnnnn.  
CSFM001I ICSF INITIALIZATION COMPLETE
```

---

## Migration

If you are migrating from HCR7708 to HCR770A and you have a PCI X Cryptographic Coprocessor, almost all the functionality previously available with z/OS V1 R3 is now supported.

## Functions Not Supported

The following section lists functions not supported by HCR770A with a PCI X Cryptographic Coprocessor installed.

1. There is no KMMK (key management master key).
2. The Commercial Data Masking Facility (CDMF) is no longer supported. The CDMF keyword on KGUP control statements and panels is no longer supported.
3. The Public Key Algorithm Digital Signature Standard is not supported. This affects callable services CSNDPKG, CSNDPKI, CSNDDSG, and CSNDDSV.
4. The PBVC keyword is not supported on a PCI X Cryptographic Coprocessor. This affects callable services Clear PIN Generate Alternate (CSNBCPE), Clear PIN Encrypt (CSNBCPE), PIN Translate (CSNBPTR) and PIN Verify (CSNBPVR).
5. RSA keys of modulus less than 512 bits are not supported on a PCI X Cryptographic Coprocessor.

## Setup Considerations

The following section lists setup changes that should be considered when installing HCR770A with a PCI X Cryptographic Coprocessor installed.

Consideration should be given to:

1. The PCIXCC has only one PKA master key, the asymmetric-keys master key (ASYM-MK). Users of CCF systems where the SMK value is not equal to the KMMK value should change the PKA master key values to be the same, and reencipher their PKDS. (You must have a PCICC to do the reencipher.) Otherwise, their private keys encrypted under the KMMK will not be usable on a PCIXCC system.
2. CICS wait list should be updated for services now executing on PCIXCCs. The sample CICS wait list, CSFWTL01, supplied by IBM includes these services and can be used as a reference.
3. PKDS initialization is required.
4. New options data set keyword CKTAUTH.
5. A CKDS initialized on an IBM @server zSeries 990 can not be used on CCF systems.
6. If sharing a PKDS with a PCICC and PCIXCC, delete the PKDS records for labelnames of retained keys on PCICCs no longer in use.
7. Customers who run CSFEUTIL to setup ICSF for automated electronic delivery process no longer need to execute CSFEUTIL on a z990 system. SHA-1 is available on z990 without entering ICSF master keys.

## Programming Considerations

The following section lists programming changes that should be considered when installing HCR770A with a PCI X Cryptographic Coprocessor installed.

Consideration should be given to:

1. The DATAC key type should only be used with a PCI X Cryptographic Coprocessor on the IBM @server zSeries 990.
2. The PIN block format checking on PCIXCC is more rigorous. For CSNBPVR, CSNBPTR and CSNBCPA services, the input PIN block must have the correct format as specified in the PIN Profile parameter. On a CCF system, the PIN block format checking is incomplete. In particular, the TRANSLAT processing mode of PIN Translate (CSNBPTR) may now fail on PCIXCC while it was successful on CCF.
3. 512 to 2048 bit modulus for RSA keys is supported in all PKA services except SET services (Set Block Compose and Set Block Decompose).
4. All CCF functions are now executed on the PCI X Cryptographic Coprocessor. This may cause some impact on the performance of customer applications.
5. Reason codes from the PCI X Cryptographic Coprocessor may be different from previous cryptographic hardware.
6. With PCIXCCs, the requirement that caller must be in supervisor state to use NOCV tokens is lifted for the Key Record Write (CSNBKRW) service.
7. The z/OS SCHEDULE and IEAMSCHD macros are used to schedule SRBs. On the IBM @server zSeries 990, since there are no CCFs on the system, applications should delete FEATURE=CRYPTO on the SCHEDULE and IEAMSCHD macros or the SRB being scheduled will not run.
8. External tokens that are export prohibited are imported differently on a z990 system with PCIXCCs. The imported internal token will have the same control vector as the external with export prohibited. These tokens will only be usable on a z990 system with PCIXCC or on CCF systems with PCICCs. On previous hardware (CCF systems) the imported internal token had a control vector that allowed export, and export prohibition was enforced by the export flag in the token.
9. Prohibit Export service can now be used for MAC and MACVER keys.

---

## TKE workstation

The Trusted Key Entry (TKE) workstation (Version 4.0) is available on the IBM @server zSeries 990. It can also be used to provide data key management on the IBM @server zSeries 900, IBM @server zSeries 800, S/390 G6 Enterprise Server, and S/390 G5 Enterprise Server.

Previous versions of TKE cannot be used with the PCI X Cryptographic Coprocessor.

## Access Control Points

Access to services that are executed on the PCI X Cryptographic Coprocessor is through Access Control Points in the DEFAULT Role. To execute callable services on the PCI X Cryptographic Coprocessor, access control points must be enabled for each service in the DEFAULT Role. For systems that do not use the optional TKE Workstation, all access control points (current and new) are enabled in the DEFAULT Role with the appropriate microcode level on the PCI X Cryptographic Coprocessor.

New TKE users and non-TKE users have all access control points enabled. This is also true for brand new TKE V4 users. For TKE V4.0, all access control points will

| be enabled on the PCIXCC cards, even if you're converting from TKE 3.1. For a  
| complete list of the access control points, see *z/OS Cryptographic Services ICSF*  
| *Application Programmer's Guide*.

| **Note:** Access control points DKYGENKY-DALL and CSNDDSG ZERO-PAD  
| (unrestricted hash length) are always disabled in the DEFAULT Role for all  
| customers (TKE and Non-TKE). A TKE Workstation is required to enable  
| these access control points.

---

## | TSO panels

| There are new panels and changes to panels to support the PCI X Cryptographic  
| Coprocessor.



---

## Appendix D. IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor

This section describes the processing of the IBM @server zSeries 990 environment, without a PCI X Cryptographic Coprocessor. Note that this server does not support the Cryptographic Coprocessor Feature or the PCI Cryptographic Coprocessor.

---

### Operating System Requirements

HCR770A can only be installed on z/OS V1.2 and R1.4. The web deliverable will support OS/390 V2.10 and z/OS V1.3 in the future.

---

### Applications and programs

Applications requiring secure cryptography using encrypted keys will not be able to execute on the IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor. All cryptographic keys must be clear keys.

The following applications and programs are not supported:

- Access Method Services Cryptographic option
- CICS attachment facility
- CKDS Conversion program
- CSFEUTIL program for CKDS reencipher, refresh, change master key, and passphrase initialization functions
- CSFPUTIL program for PKDS activate, cache refresh, reencipher, and initialization functions
- Distributed Key Management System (DKMS)
- Key Generation Utility Program (KGUP)
- PCF applications
- UDX (User Defined Extension) support
- VTAM Session Level Encryption
- 4753-HSP applications
- Applications that access ICSF services through the BSAFE interfaces

---

### Callable services

The following services are available when running on a IBM @server zSeries 990 without a PCI X Cryptographic Coprocessor:

- Character/Nibble Conversion (CSNBXBC and CSNBXCB)
- Code Conversion (CSNBXEA and CSNBXAE)
- Control Vector Generate (CSNBCVG)
- Decode (CSNBDCO) - This service requires CP Assist for Cryptographic Functions.
- Digital Signature Verify (CSNDDSV) - This service requires a PCI Cryptographic Accelerator.
- Encode (CSNBECO) - This service requires CP Assist for Cryptographic Functions.

- MDC Generate (CSNBMDG and CSNBMDG1) - This service requires CP Assist for Cryptographic Functions.
- One-Way Hash Generate (CSNBOWH and CSNBOWH1)
- PKA Decrypt (CSNDPKD) - This service requires a PCI Cryptographic Accelerator.
- PKA Encrypt (CSNDPKE) ZERO-PAD formatting only - This service requires a PCI Cryptographic Accelerator.
- PKA Key Token Build (CSNDPKB)
- PKA Public Key Extract (CSNDPKX)
- Symmetric Key Decipher (CSNBSYD and CSNBSYD1) - This service requires CP Assist for Cryptographic Functions.
- Symmetric Key Encipher (CSNBSYE and CSNBSYE1) - This service requires CP Assist for Cryptographic Functions.
- X9.9 Data Editing (CSNB9ED)

Installation defined callable services are supported only if you're using clear keys and using one of the above supported callable services.

---

## ICSF Setup and Initialization

It is normal to see the following messages during the startup of ICSF on a IBM @server zSeries 990:

- Starting ICSF on a IBM @server zSeries 990 without a PCI Cryptographic Accelerator or PCI X Cryptographic Coprocessor:  

```
CSFM507I CRYPTOGRAPHY - THERE ARE NO PCI X CRYPTOGRAPHIC COPROCESSORS ONLINE.
CSFM508I CRYPTOGRAPHY - THERE ARE NO CRYPTOGRAPHIC ACCELERATORS ONLINE.
CSFM001I ICSF INITIALIZATION COMPLETE
```
- Starting ICSF on a IBM @server zSeries 990 with a PCI Cryptographic Accelerator and without a PCI X Cryptographic Coprocessor. You 'll receive message CSFM411I for each PCI Cryptographic Accelerator you have online.  

```
CSFM411I PCI CRYPTOGRAPHIC ACCELERATOR Ann IS ACTIVE
CSFM507I CRYPTOGRAPHY - THERE ARE NO PCI X CRYPTOGRAPHIC COPROCESSORS ONLINE
CSFM001I ICSF INITIALIZATION COMPLETE
```

---

## Secure Sockets Layer (SSL)

System SSL applications are supported on the IBM @server zSeries 990. SSL defines methods for data encryption, server authentication, message integrity, and client authentication for a TCP/IP connection. Security is provided on the link and callable services have been enhanced for DES, TDES and SHA-1 services.

---

## TKE workstation

The Trusted Key Entry (TKE) workstation is not available with this hardware configuration.

---

## Appendix E. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

---

### Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

---

### Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.



---

## Notices

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Mail Station P300  
2455 South Road  
Poughkeepsie, NY 12601-5400  
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX  
CICS  
ES/3090  
ES/9000  
eServer  
IBM  
IBMLink  
Multiprise  
MVS  
MVS/ESA  
MVS/SP  
OS/390  
Parallel Sysplex  
Personal Security  
Processor Resource/Systems Manager  
PR/SM  
RACF  
Resource Link  
RMF  
S/390  
S/390 Parallel Enterprise Server  
System/390  
VTAM  
3090  
zSeries  
z/OS  
z/OS.e

The following terms are trademarks or registered trademarks of other companies:

**BSAFE**          RSA Data Security, Inc.

<b>MasterCard</b>	MasterCard International, Incorporated
<b>Netscape</b>	Netscape Communications Corporation
<b>SET</b>	SET Secure Electronic Transaction, LLC
<b>UNIX</b>	The Open Group
<b>VISA</b>	VISA International Service Association

Other company, product, and service names may be trademarks or service marks of others.



---

# Glossary

This glossary defines terms and abbreviations used in Integrated Cryptographic Service Facility (ICSF). If you do not find the term you are looking for, refer to the index of the appropriate Integrated Cryptographic Service Facility document or view *IBM Glossary of Computing Terms* located at:

<http://www.ibm.com/ibm/terminology>

This glossary includes terms and definitions from:

- *IBM Glossary of Computing Terms*. Definitions are identified by the symbol (D) after the definition.
- *The American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- *The Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

Definitions specific to the Integrated Cryptographic Services Facility are labeled "In ICSF."

## A

**access method services (AMS).** The facility used to define and reproduce VSAM key-sequenced data sets (KSDS). (D)

**Advanced Encryption Standard (AES).** In computer security, the National Institute of Standards and Technology (NIST) Advanced Encryption Standard (AES) algorithm. The AES algorithm is documented in a draft Federal Information Processing Standard.

**AES.** Advanced Encryption Standard.

**American National Standard Code for Information Interchange (ASCII).** The standard code using a coded character set consisting of 7-bit characters (8 bits including parity check) that is used for information exchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

**ANSI key-encrypting key (AKEK).** A 64- or 128-bit key used exclusively in ANSI X9.17 key management applications to protect data keys exchanged between systems.

**ANSI X9.17.** An ANSI standard that specifies algorithms and messages for DES key distribution.

**ANSI X9.19.** An ANSI standard that specifies an optional double-MAC procedure which requires a double-length MAC key.

**application program.** (1) A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities. (D)

**application program interface (API).** (1) A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program. (D) (2) In ICSF, a callable service.

**asymmetric cryptography.** Synonym for public key cryptography. (D)

**authentication pattern.** An 8-byte pattern that ICSF calculates from the master key when initializing the cryptographic key data set. ICSF places the value of the authentication pattern in the header record of the cryptographic key data set.

**authorized program facility (APF).** A facility that permits identification of programs authorized to use restricted functions. (D)

## C

**callable service.** A predefined sequence of instructions invoked from an application program, using a CALL instruction. In ICSF, callable services perform cryptographic functions and utilities.

**CBC.** Cipher block chaining.

**CCA.** Common Cryptographic Architecture.

**CCF.** Cryptographic Coprocessor Feature.

**CDMF.** Commercial Data Masking Facility.

**CEDA.** A CICS transaction that defines resources online. Using CEDA, you can update both the CICS system definition data set (CSD) and the running CICS system.

**checksum.** (1) The sum of a group of data associated with the group and used for checking purposes. (T) (2) In ICSF, the data used is a key part. The resulting checksum is a two-digit value you enter when you use the key-entry unit to enter a master key part or a clear key part into the key-storage unit.

**Chinese Remainder Theorem (CRT).** A mathematical theorem that defines a format for the RSA private key that improves performance.

**CICS.** Customer Information Control System.

**cipher block chaining (CBC).** A mode of encryption that uses the data encryption algorithm and requires an initial chaining vector. For encipher, it exclusively ORs the initial block of data with the initial control vector and then enciphers it. This process results in the encryption both of the input block and of the initial control vector that it uses on the next input block as the process repeats. A comparable chaining process works for decipher.

**ciphertext.** (1) In computer security, text produced by encryption. (2) Synonym for enciphered data. (D)

**CKDS.** Cryptographic Key Data Set.

**clear key.** Any type of encryption key not protected by encryption under another key.

**CMOS.** Complementary metal oxide semiconductor.

**coexistence mode.** An ICSF method of operation during which CUSP or PCF can run independently and simultaneously on the same ICSF system. A CUSP or PCF application program can run on ICSF in this mode if the application program has been reassembled.

**Commercial Data Masking Facility (CDMF).** A data-masking algorithm using a DES-based kernel and a key that is shortened to an effective key length of 40 DES key-bits. Because CDMF is not as strong as DES, it is called a masking algorithm rather than an encryption algorithm. Implementations of CDMF, when used for data confidentiality, are generally exportable from the USA and Canada.

**Common Cryptographic Architecture: Cryptographic Application Programming Interface.** Defines a set of cryptographic functions, external interfaces, and a set of key management rules that provide a consistent, end-to-end cryptographic architecture across different IBM platforms.

**compatibility mode.** An ICSF method of operation during which a CUSP or PCF application program can run on ICSF without recompiling it. In this mode, ICSF cannot run simultaneously with CUSP or PCF.

**complementary keys.** A pair of keys that have the same clear key value, are different but complementary types, and usually exist on different systems.

**console.** A part of a computer used for communication between the operator or maintenance engineer and the computer. (A)

**control-area split.** In systems with VSAM, the movement of the contents of some of the control intervals in a control area to a newly created control area in order to facilitate insertion or lengthening of a data record when there are no remaining free control intervals in the original control area. (D)

**control block.** (1) A storage area used by a computer program to hold control information. (I) Synonymous with control area. (2) The circuitry that performs the control functions such as decoding microinstructions and generating the internal control signals that perform the operations requested. (A)

**control interval.** A fixed-length area of direct-access storage in which VSAM stores records and creates distributed free space. Also, in a key-sequenced data set or file, the set of records pointed to by an entry in the sequence-set index record. The control interval is the unit of information that VSAM transmits to or from direct access storage. A control interval always comprises an integral number of physical records. (D)

**control interval split.** In systems with VSAM, the movement of some of the stored records in a control interval to a free control interval to facilitate insertion or lengthening of a record that does not fit in the original control interval. (D)

**control statement input data set.** A key generator utility program data set containing control statements that a particular key generator utility program job will process.

**control statement output data set.** A key generator utility program data set containing control statements to create the complements of keys created by the key generator utility program.

**control vector.** In ICSF, a mask that is exclusive ORed with a master key or a transport key before ICSF uses that key to encrypt another key. Control vectors ensure that keys used on the system and keys distributed to other systems are used for only the cryptographic functions for which they were intended.

**cross memory mode.** Synchronous communication between programs in different address spaces that permits a program residing in one address space to access the same or other address spaces. This

synchronous transfer of control is accomplished by a calling linkage and a return linkage.

**CRT.** Chinese Remainder Theorem.

**cryptographic adapter (4755 or 4758).** An expansion board that provides a comprehensive set of cryptographic functions for the network security processor and the workstation in the TSS family of products.

**cryptographic coprocessor.** A microprocessor that adds cryptographic processing functions to specific OS/390 Enterprise Servers, the OS/390 Multiprise and higher processors, and the IBM @server zSeries. The Cryptographic Coprocessor Feature is a tamper-resistant chip built into the processor board. The combination of the Cryptographic Coprocessor Feature and ICSF/MVS Version 2 Release 1, or higher, provides secure high-speed cryptographic services in the OS/390 and z/OS environment.

**cryptographic key data set (CKDS).** (1) A data set that contains the encrypting keys used by an installation. (D) (2) In ICSF, a VSAM data set that contains all the cryptographic keys. Besides the encrypted key value, an entry in the cryptographic key data set contains information about the key.

**cryptography.** (1) The transformation of data to conceal its meaning. (2) In computer security, the principles, means, and methods for encrypting plaintext and decrypting ciphertext. (D) (3) In ICSF, the use of cryptography is extended to include the generation and verification of MACs, the generation of MDCs and other one-way hashes, the generation and verification of PINs, and the generation and verification of digital signatures.

**CUSP (Cryptographic Unit Support Program).** The IBM cryptographic offering, program product 5740-XY6, using the channel-attached 3848.

**CUSP/PCF conversion program.** A program, for use during migration from CUSP or PCF to ICSF, that converts a CUSP or PCF cryptographic key data set into a ICSF cryptographic key data set.

**Customer Information Control System (CICS).** An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user written application programs. It includes facilities for building, using, and maintaining databases.

**CVC.** Card verification code used by MasterCard.

**CVV.** Card verification value used by VISA.

## D

**data encryption algorithm (DEA).** In computer security, a 64-bit block cipher that uses a 64-bit key, of

which 56 bits are used to control the cryptographic process and 8 bits are used for parity checking to ensure that the key is transmitted properly. (D)

**data encryption standard (DES).** In computer security, the National Institute of Standards and Technology (NIST) Data Encryption Standard, adopted by the U.S. government as Federal Information Processing Standard (FIPS) Publication 46, which allows only hardware implementations of the data encryption algorithm. (D)

**data key or data-encrypting key.** (1) A key used to encipher, decipher, or authenticate data. (D) (2) In ICSF, a 64-bit encryption key used to protect data privacy using the DES algorithm or the CDMF algorithm.

**data set.** The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. (D)

**data-translation key.** A 64-bit key that protects data transmitted through intermediate systems when the originator and receiver do not share the same key.

**DEA.** Data encryption algorithm.

**decipher.** (1) To convert enciphered data in order to restore the original data. (T) (2) In computer security, to convert ciphertext into plaintext by means of a cipher system. (3) To convert enciphered data into clear data. Contrast with encipher. Synonymous with decrypt. (D)

**decode.** (1) To convert data by reversing the effect of some previous encoding. (I) (A) (2) In ICSF, to decipher data by use of a clear key.

**decrypt.** See decipher.

**DES.** Data Encryption Standard.

**diagnostics data set.** A key generator utility program data set containing a copy of each input control statement followed by a diagnostic message generated for each control statement.

**digital signature.** In public key cryptography, information created by using a private key and verified by using a public key. A digital signature provides data integrity and source nonrepudiation.

**Digital Signature Standard (DSS).** A standard describing the use of algorithms for digital signature purposes. One of the algorithms specified is DSA (Digital Signature Algorithm).

**domain.** (1) That part of a network in which the data processing resources are under common control. (T) (2) In ICSF, an index into a set of master key registers.

**double-length key.** A key that is 128 bits long. A key can be either double- or single-length. A single-length key is 64 bits long.

**DSA.** Digital Signature Algorithm.

**DSS.** Digital Signature Standard.

## E

**ECB.** Electronic codebook.

**ECI.** Eurochèque International S.C., a financial institution consortium that has defined three PIN block formats.

**EID.** Environment Identification.

**electronic codebook (ECB) operation.** (1) A mode of operation used with block cipher cryptographic algorithms in which plaintext or ciphertext is placed in the input to the algorithm and the result is contained in the output of the algorithm. (D) (2) A mode of encryption using the data encryption algorithm, in which each block of data is enciphered or deciphered without an initial chaining vector. It is used for key management functions and the encode and decode callable services.

**electronic funds transfer system (EFTS).** A computerized payment and withdrawal system used to transfer funds from one account to another and to obtain related financial data. (D)

**encipher.** (1) To scramble data or to convert data to a secret code that masks the meaning of the data to any unauthorized recipient. Synonymous with encrypt. (2) Contrast with decipher. (D)

**enciphered data.** Data whose meaning is concealed from unauthorized users or observers. (D)

**encode.** (1) To convert data by the use of a code in such a manner that reconversion to the original form is possible. (T) (2) In computer security, to convert plaintext into an unintelligible form by means of a code system. (D) (3) In ICSF, to encipher data by use of a clear key.

**encrypt.** See encipher.

**exit.** (1) To execute an instruction within a portion of a computer program in order to terminate the execution of that portion. Such portions of computer programs include loops, subroutines, modules, and so on. (T) (2) In ICSF, a user-written routine that receives control from the system during a certain point in processing—for example, after an operator issues the START command.

**exportable form.** A condition a key is in when enciphered under an exporter key-encrypting key. In this form, a key can be sent outside the system to another system. A key in exportable form cannot be used in a cryptographic function.

**exporter key-encrypting key.** A 128-bit key used to protect keys sent to another system. A type of transport key.

## F

**file.** A named set of records stored or processed as a unit. (T)

## G

**GBP.** German Bank Pool.

**German Bank Pool (GBP).** A German financial institution consortium that defines specific methods of PIN calculation.

## H

**hashing.** An operation that uses a one-way (irreversible) function on data, usually to reduce the length of the data and to provide a verifiable authentication value (checksum) for the hashed data.

**header record.** A record containing common, constant, or identifying information for a group of records that follows. (D)

## I

**ICSF.** Integrated Cryptographic Service Facility.

**importable form.** A condition a key is in when it is enciphered under an importer key-encrypting key. A key is received from another system in this form. A key in importable form cannot be used in a cryptographic function.

**importer key-encrypting key.** A 128-bit key used to protect keys received from another system. A type of transport key.

**initial chaining vector (ICV).** A 64-bit random or pseudo-random value used in the cipher block chaining mode of encryption with the data encryption algorithm.

**initial program load (IPL).** (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs. (D)

**input PIN-encrypting key.** A 128-bit key used to protect a PIN block sent to another system or to translate a PIN block from one format to another.

**installation exit.** See exit.

**Integrated Cryptographic Service Facility (ICSF).** A licensed program that runs under MVS/System Product 3.1.3, or higher, or OS/390 Release 1, or higher, or z/OS, and provides access to the hardware cryptographic feature for programming applications. The

combination of the hardware cryptographic feature and ICSF provides secure high-speed cryptographic services.

**International Organization for Standardization.** An organization of national standards bodies from many countries, established to promote the development of standards to facilitate the international exchange of goods and services and to develop cooperation in intellectual, scientific, technological, and economic activity. ISO has defined certain standards relating to cryptography and has defined two PIN block formats.

**ISO.** International Organization for Standardization.

## J

**job control language (JCL).** A control language used to identify a job to an operating system and to describe the job's requirements. (D)

## K

**key-encrypting key (KEK).** (1) In computer security, a key used for encryption and decryption of other keys. (D) (2) In ICSF, a master key or transport key.

**key generator utility program (KGUP).** A program that processes control statements for generating and maintaining keys in the cryptographic key data set.

**key output data set.** A key generator utility program data set containing information about each key that the key generator utility program generates except an importer key for file encryption.

**key part.** A 32-digit hexadecimal value that you enter for ICSF to combine with other values to create a master key or clear key.

**key part register.** A register in the key storage unit that stores a key part while you enter the key part.

## L

**linkage.** The coding that passes control and parameters between two routines.

**load module.** All or part of a computer program in a form suitable for loading into main storage for execution. A load module is usually the output of a linkage editor. (T)

**LPAR mode.** The central processor mode that enables the operator to allocate the hardware resources among several logical partitions.

## M

**MAC generation key.** A 64-bit or 128-bit key used by a message originator to generate a message authentication code sent with the message to the message receiver.

**MAC verification key.** A 64-bit or 128-bit key used by a message receiver to verify a message authentication code received with a message.

**magnetic tape.** A tape with a magnetizable layer on which data can be stored. (T)

**master key.** (1) In computer security, the top-level key in a hierarchy of key-encrypting keys. (2) In ICSF, there are three types of master keys on the Cryptographic Coprocessor Feature: the 128-bit DES master key, the 192-bit signature master key, and the 192-bit key management master key. On the PCI Cryptographic Coprocessor there are two types of master keys: the 192-bit Symmetric master key and the 192-bit Asymmetric master key. Master keys are known only to the ICSF hardware and maintained in the cryptographic enclosure in a secure fashion. All keys in operational form in the system are enciphered under a master key. Master keys are used only to encrypt other keys.

**master key concept.** The idea of using a single cryptographic key, the master key, to encrypt all other keys on the system.

**master key register.** A register in the Cryptographic Coprocessor Feature that stores the master key that is active on the system.

**master key variant.** A key derived from the master key by use of a control vector. It is used to force separation by type of keys on the system.

**MD4.** Message Digest 4. A hash algorithm.

**MD5.** Message Digest 5. A hash algorithm.

**message authentication code (MAC).** (1) The cryptographic result of block cipher operations on text or data using the cipher block chain (CBC) mode of operation. (D) (2) In ICSF, a MAC is used to authenticate the source of the message, and verify that the message was not altered during transmission or storage.

**modification detection code (MDC).** (1) A 128-bit value that interrelates all bits of a data stream so that the modification of any bit in the data stream results in a new MDC. (2) In ICSF, an MDC is used to verify that a message or stored data has not been altered.

**multiple encipherment.** The method of encrypting a key under a double-length key-encrypting key.

## N

**new master key register.** A register in the key storage unit that stores a master key before you make it active on the system.

**NIST.** U.S. National Institute of Science and Technology.

**NOCV processing.** Process by which the key generator utility program or an application program encrypts a key under a transport key itself rather than a transport key variant.

**noncompatibility mode.** An ICSF method of operation during which CUSP or PCF can run independently and simultaneously on the same z/OS, OS/390 or MVS system. You cannot run a CUSP or PCF application program on ICSF in this mode.

**nonrepudiation.** A method of ensuring that a message was sent by the appropriate individual.

**notarization.** The ANSI X9.17 process involving the coupling of an ANSI key-encrypting key (AKEK) with ASCII character strings containing origin and destination identifiers and then exclusive ORing (or offsetting) the result with a binary counter.

## O

**OAEP.** Optimal asymmetric encryption padding.

**offset.** The process of exclusively ORing a counter to a key.

**old master key register.** A register in the key storage unit that stores a master key that you replaced with a new master key.

**operational form.** The condition of a key when it is encrypted under the master key so that it is active on the system.

**output PIN-encrypting key.** A 128-bit key used to protect a PIN block received from another system or to translate a PIN block from one format to another.

## P

**PAN.** Personal Account Number.

**parameter.** Data passed between programs or procedures. (D)

**parmlib.** A system parameter library, either SYS1.PARMLIB or an installation-supplied library.

**partial notarization.** The ANSI X9.17 standard does not use the term partial notarization. IBM has divided the notarization process into two steps and defined the term partial notarization as a process during which only

the first step of the two-step ANSI X9.17 notarization process is performed. This step involves the coupling of an ANSI key-encrypting key (AKEK) with ASCII character strings containing origin and destination identifiers.

**partitioned data set (PDS).** A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. (D)

**Personal Account Number (PAN).** A Personal Account Number identifies an individual and relates that individual to an account at a financial institution. It consists of an issuer identification number, customer account number, and one check digit.

**PCI Cryptographic Coprocessor.** The 4758 model 2 standard PCI-bus card supported on the field upgraded IBM S/390 Parallel Enterprise Server - Generation 5, the IBM S/390 Parallel Enterprise Server - Generation 6 and the IBM @server zSeries.

**PCICA.** PCI Cryptographic Accelerator.

**PCICC.** PCI Cryptographic Coprocessor.

| **PCI X Cryptographic Coprocessor.** An asynchronous  
| cryptographic coprocessor available on the IBM  
| @server zSeries 990.

**PCIXCC.** PCI X Cryptographic Coprocessor.

**personal identification number (PIN).** The 4- to 12-digit number entered at an automatic teller machine to identify and validate the requester of an automatic teller machine service. Personal identification numbers are always enciphered at the device where they are entered, and are manipulated in a secure fashion.

**Personal Security card.** An ISO-standard "smart card" with a microprocessor that enables it to perform a variety of functions such as identifying and verifying users, and determining which functions each user can perform.

**PIN block.** A 64-bit block of data in a certain PIN block format. A PIN block contains both a PIN and other data.

**PIN generation key.** A 128-bit key used to generate PINs or PIN offsets algorithmically.

**PIN key.** A 128-bit key used in cryptographic functions to generate, transform, and verify the personal identification numbers.

**PIN offset.** For 3624, the difference between a customer-selected PIN and an institution-assigned PIN. For German Bank Pool, the difference between an institution PIN (generated with an institution PIN key) and a pool PIN (generated with a pool PIN key).

**PIN verification key.** A 128-bit key used to verify PINs algorithmically.

**PKA.** Public Key Algorithm.

**PKCS.** Public Key Cryptographic Standards (RSA Data Security, Inc.)

**PKDS.** Public key data set (PKA cryptographic key data set).

**plaintext.** Data in normal, readable form.

**primary space allocation.** An area of direct access storage space initially allocated to a particular data set or file when the data set or file is defined. See also secondary space allocation. (D)

**private key.** In computer security, a key that is known only to the owner and used with a public key algorithm to decrypt data or generate digital signatures. The data is encrypted and the digital signature is verified using the related public key.

**processor complex.** A configuration that consists of all the machines required for operation.

**Processor Resource/Systems Manager.** Enables logical partitioning of the processor complex, may provide additional byte-multiplexer channel capability, and supports the VM/XA System Product enhancement for Multiple Preferred Guests.

**Programmed Cryptographic Facility (PCF).** (1) An IBM licensed program that provides facilities for enciphering and deciphering data and for creating, maintaining, and managing cryptographic keys. (D) (2) The IBM cryptographic offering, program product 5740-XY5, using software only for encryption and decryption.

**PR/SM.** Processor Resource/Systems Manager.

**public key.** In computer security, a key made available to anyone who wants to encrypt information using the public key algorithm or verify a digital signature generated with the related private key. The encrypted data can be decrypted only by use of the related private key.

**public key algorithm (PKA).** In computer security, an asymmetric cryptographic process in which a public key is used for encryption and digital signature verification and a private key is used for decryption and digital signature generation.

**public key cryptography.** In computer security, cryptography in which a public key is used for encryption and a private key is used for decryption. Synonymous with asymmetric cryptography.

## R

**RACE Integrity Primitives Evaluation Message Digest.** A hash algorithm.

**RDO.** Resource definition online.

**record chaining.** When there are multiple cipher requests and the output chaining vector (OCV) from the previous encipher request is used as the input chaining vector (ICV) for the next encipher request.

**Resource Access Control Facility (RACF).** An IBM licensed program that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources. (D)

**retained key.** A private key that is generated and retained within the secure boundary of the PCI Cryptographic Coprocessor.

**return code.** (1) A code used to influence the execution of succeeding instructions. (A) (2) A value returned to a program to indicate the results of an operation requested by that program. (D)

**Rivest-Shamir-Adleman (RSA) algorithm.** A process for public key cryptography that was developed by R. Rivest, A. Shamir, and L. Adleman.

**RMF.** Resource Manager Interface.

**RMI.** Resource Measurement Facility.

**RSA.** Rivest-Shamir-Adleman.

## S

**SAF.** Security Authorization Facility.

**save area.** Area of main storage in which contents of registers are saved. (A)

**secondary space allocation.** In systems with VSAM, area of direct access storage space allocated after primary space originally allocated is exhausted. See also primary space allocation. (D)

**Secure Electronic Transaction.** A standard created by Visa International and MasterCard for safe-guarding payment card purchases made over open networks.

**Secure Sockets Layer.** A security protocol that provides communications privacy over the Internet by allowing client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

**sequential data set.** A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. (D)

**SET.** Secure Electronic Transaction.

**SHA-1.** Secure Hash Algorithm 1, a hash algorithm required for use with the Digital Signature Standard.

**single-length key.** A key that is 64 bits long. A key can be single- or double-length. A double-length key is 128 bits long.

**special secure mode.** An alternative form of security that allows you to enter clear keys with the key generator utility program or generate clear PINs.

**SSL.** Secure Sockets Layer.

**supervisor state.** A state during which a processing unit can execute input/output and other privileged instructions. (D)

**System Authorization Facility (SAF).** An interface to a system security system like the Resource Access Control Facility (RACF).

**system key.** A key that ICSF creates and uses for internal processing.

**System Management Facilities (SMF).** An optional control program feature of OS/VS that provides the means for gathering and recording information that can be used to evaluate system usage. (D)

## T

**TDEA.** Triple Data Encryption Algorithm.

**TKE.** Trusted key entry.

**Transaction Security System.** An IBM product offering including both hardware and supporting software that provides access control and basic cryptographic key-management functions in a network environment. In the workstation environment, this includes the 4755 Cryptographic Adapter, the Personal Security Card, the 4754 Security Interface Unit, the Signature Verification feature, the Workstation Security Services Program, and the AIX Security Services Program/6000. In the host environment, this includes the 4753 Network Security Processor and the 4753 Network Security Processor MVS Support Program.

**transport key.** A 128-bit key used to protect keys distributed from one system to another. A transport key can either be an exporter key-encrypting key, an importer key-encrypting key, or an ANSI key-encrypting key.

**transport key variant.** A key derived from a transport key by use of a control vector. It is used to force separation by type for keys sent between systems.

**TRUE.** Task-related User Exit (CICS). The CICS-ICSF Attachment Facility provides a CSFATRUE and CSFATREN routine.

## U

**UAT.** UDX Authority Table.

**UDF.** User-defined function.

**UDK.** User-derived key.

**UDP.** User Developed Program.

**UDX.** User Defined Extension.

## V

**verification pattern.** An 8-byte pattern that ICSF calculates from the key parts you enter when you enter a master key or clear key. You can use the verification pattern to verify that you have entered the key parts correctly and specified a certain type of key.

**Virtual Storage Access Method (VSAM).** (1) An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. (D) (2) An access method for indexed or sequential processing of fixed and variable-length records on direct-access devices. The records in a VSAM data set or file can be organized in logical sequence by means of a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by means of relative-record number.

**VISA.** A financial institution consortium that has defined four PIN block formats and a method for PIN verification.

**VISA PIN Verification Value (VISA PVV).** An input to the VISA PIN verification process that, in practice, works similarly to a PIN offset.

## Numerics

**3621.** A model of an IBM Automatic Teller Machine that has a defined PIN block format.

**3624.** A model of an IBM Automatic Teller Machine that has a defined PIN block format and methods of PIN calculation.

**4753.** The Network Security processor. The IBM 4753 is a processor that uses the Data Encryption Algorithm and the RSA algorithm to provide cryptographic support for systems requiring secure transaction processing (and other cryptographic services) at the host computer. The NSP includes a 4755 cryptographic adapter in a workstation which is channel attached to a S/390 host computer.

**4758.** The IBM PCI Cryptographic processor provides a secure programming and hardware environment where DES and RSA processes are performed.



# Index

## Numerics

- 3624
  - Customer-Selected PIN 15
  - PIN generation and verification algorithms 15
- 4753
  - cryptographic services 37
  - HSP applications running under ICSF 38
  - key tokens 61
- 4754
  - Security Interface Unit 37
- 4755
  - cryptographic services 37
- 4758
  - cryptographic services 37

## A

- access control 1
- Access Method Services Cryptographic Option and ICSF 40
- accessibility 83
- addressing mode
  - no restrictions on ICSF's caller 40
- AKEK 26
  - notarization 34
  - partial notarization 34
- ANSI Data Encryption Algorithm 2, 4
- ANSI X9.17
  - key management callable services 24, 34
  - key-encrypting key 26
- Assembler
  - callable services 23
- asymmetric cryptographic system 3
- auditing 53
- authenticity of data
  - using digital signature 6
- authorization 1

## B

- BSAFE
  - using with ICSF 40

## C

- C high-level language
  - callable services 23
- callable service
  - ANSI X9.17 key management 34
  - compliant with IBM's Common Cryptographic Architecture 37
  - dynamic CKDS update 32, 33
  - dynamic PKDS update 34
  - exits 20
  - hardware configuration support 67
  - improved productivity 18
  - installation-defined 20

- callable service (*continued*)
  - PIN generation in special secure mode 31
  - secure key import in special secure mode 31
  - summary 23
- callable services
  - ICSF 55
- changing ICSF
  - exits 20
- CICS wait list 77
- cipher 26
- cipher block chaining 23
- CIPHER macro
  - exit considerations 37
- ciphertext 1
- ciphertext translate callable service 10, 23
- CKDS
  - dynamic update callable services 24
  - updating 17, 32
- clear keys
  - allowing or preventing 52
  - description of callable service 23
  - entering into the CKDS in special secure mode 31
- clear master key entry panels 28
- Clear PIN generate callable service
  - can be used only in special secure mode 52
- COBOL high-level language
  - callable services 23
- coexistence mode
  - description of PCF 38
  - installation option 20
- Common Cryptographic Architecture 4, 37
- compatibility mode
  - installation option 20
  - running ICSF and 4753-HSP 38
  - with PCF, description of 38
- complementary key forms 12
- complementary key pairs
  - list 30
  - maintaining using KGUP 30
- confidentiality of data 1
- configurations
  - Cryptographic Coprocessor Feature 23
- continuous operations
  - maintaining 17
- control vectors
  - description of 25
  - selectively avoiding use 25, 39
- controlling access
  - to PKDS 51
  - to services and keys 51
  - to the disk copy of the CKDS 51
  - to the key generator utility program 51
- conversion considerations
  - 4753-HSP to OS/390 ICSF 61
- CP Assist for Cryptographic Functions
  - description 43
- cross key
  - replaced by transport key 26

- crypto CP
  - more than one available for use by ICSF 51
- Cryptographic Coprocessor Feature
  - configurations 23
  - description 44
  - export control level 23
- cryptographic key data set (CKDS)
  - controlling access to 51
  - description 32
  - disk copy 32
  - dynamic update using callable services 32
  - dynamically updating 33
  - exit called when in-storage copy is accessed 20
  - exits called when disk copy is accessed 20
  - how maintained and used 32, 34
  - in-storage copy 32
  - performance considerations 54
  - performance improvement because kept in storage 18
  - storing keys 34
- cryptographic keys
  - generating and distributing 5
  - generation
    - description of callable service 23
- cryptology
  - basic elements 2
  - description 1
  - introduction 1
  - using a public key 3
  - using a secret key 2
- customizing ICSF to meet your installation's needs 19

## D

- data
  - confidentiality 1
  - exchanging with other systems 39
  - integrity 1
  - translation across networks 7
- data encipher/decipher
  - description of callable services 23
- Data Encryption Standard 2
- DATA keys 26
- data security policy
  - functions of 1
- data-encrypting key 26
- data-translation key 27
- DATAXLAT keys 27
- decipher 1
  - description of callable services 23
- decoding data 23
- DES 2, 25
  - key exchange using RSA key scheme 13
  - keys, protecting 24
  - master key 25
  - with PKA 44
- DES with PKA 44
- digital signatures
  - description 1
  - how used 6
- disability 83

- distributing cryptographic keys 5
- documents, licensed xvi
- double-length key
  - using 26
- DSA
  - key pair generation 29
- DSS
  - algorithm 4
- dynamic CKDS update callable services 32
  - overview 33
- dynamic PKDS update callable services 34

## E

- ECI Format 2 15
- ECI Format 3 15
- EDI 9
- EFT 9
- electronic commerce 9
  - on the Internet 2
- electronic data interchange (EDI) 9
- electronic funds transfer (EFT) 9
- EMK macro
  - exit considerations 37
  - replaced by the clear key import callable service 52
- EMV (Europay, MasterCard and VISA) 8
- encipher 1
  - description of callable services 23
- encode callable service
  - using a clear key 52
- encoding data 23
- encrypted keys
  - exchanging 29
- exchanging keys between systems 12
- exits 20
  - installation option 21
  - migration considerations for PCF macros 37
- exportable key form 25
- exporter key-encrypting key 26, 29
- exporting DES keys 12, 23

## F

- factorization problem 28
- FORTTRAN high-level language
  - callable services 23

## G

- generating
  - clear PINs 31
  - cryptographic keys 5, 23
  - MACs 24
  - MDCs 24
  - PINs 23
  - random numbers 23
  - RSA public and private key pairs 23
- GENKEY macro
  - exit considerations 37
  - SMF records 53

German Bank Pool PIN generation and verification algorithms 15

## H

hardware

- generating random number 23
- improved performance 18
- support for callable services 67

hardware features

- IBM @server zSeries 800 45
- IBM @server zSeries 900 45
- IBM @server zSeries 990 45
- IBM S/390 G6 Enterprise Server 46

hashes

- generating and verifying 15

hashing 24

- description 1, 15

hashing algorithms

- how used 6

high-level languages

- callable services 23

## I

IBM @server zSeries 800

- configurations 51
- hardware features 45

IBM @server zSeries 900

- configurations 51
- hardware features 45

IBM @server zSeries 990

- configurations 51
- functions not supported 77
- hardware features 45
- with PCI X Cryptographic Coprocessor 73
- without PCI X Cryptographic Coprocessor 81

IBM 3621 Format 15

IBM 3624 Format 15

IBM Encrypting PINPAD Format 15

IBM S/390 G6 Enterprise Server

- hardware features 46

IBM Transaction Security System

- callable services 37

IBM's Common Cryptographic Architecture

- using 37

ICSF

- callable services 23
- V1R2 and 4753-HSP key label considerations 62

ICSF interface changes

- callable services 55

identification 1

importable key form 25

importer key-encrypting key 26, 29

importing DES keys 12, 23

improving cryptographic performance 18

installation option

- to enable special secure mode 31

installation requirements 43

installation-defined callable services 20

integrity of data 1

integrity of data (*continued*)

methods of verifying

- hashing 6, 15
- message authentication codes (MACs) 5, 15
- modification detection codes (MDCs) 15

Interbank PIN 15

Internet

- electronic commerce on 2

ISO Format 0 15

ISO Format 1 15

## K

keeping your data private 9

key form

- definition 25
- exportable 25
- importable 25
- operational 25

key generate callable service

- overview 34

key generator utility program (KGUP)

- controlling access to 51
- description 34
- in special secure mode 31

key import and key export

- description of callable service 23

key labels

- differences between ICSF/MVS Version 1 Release 2 and 4753-HSP 62

key management master key (KMMK) 28

key separation 24

key storage unit (KSU)

- stores the master key 9

key types 25

key-encrypting key 26

- definition 25
- description 26

keyboard 83

keys

- AKEK 26
- allowing or preventing clear keys 52
- ANSI X9.17 key-encrypting 26
- CIPHER 26
- control vector 25
- controlling 24
- data-encrypting 26
- data-translation 27
- DES master 25
- exchanging with other systems 39
- exporter key-encrypting 26
- importer key-encrypting 26
- key-encrypting 26
- MAC 27
- master key variant 25
- PIN 27
- PKA master 28
- PKA, controlling access to 27
- scheduled changes 51
- sending to other installations 52
- SYM-MK master 26

- keys (*continued*)
  - transport 26
  - transport key variant 25
  - types of DES 25

## L

- licensed documents xvi
- listing and deleting
  - retained RSA private keys 23
- local key
  - replaced by transport key 26
- LookAt message retrieval tool xv
- LPAR mode 54

## M

- MAC
  - keys 27
- macro
  - PCF 37
- maintaining complementary key pairs 30
- maintaining continuous operations 17
- master key
  - DES 25
  - PKA 28
  - separate master keys in PR/SM partitions 21
  - SYMMK 26
  - variant 25
- MasterCard card-verification code (CVC) 7
- message authentication codes (MACs)
  - benefits 5
  - description 15
  - description of callable services 24
  - exchanging with other systems 39
  - generating and verifying 15
  - how used 5
- message retrieval tool, LookAt xv
- migrating to HCR770A 77
- migration considerations
  - 4753-HSP to OS/390 ICSF 61
- mode
  - special secure 31
- modification detection codes (MDCs)
  - benefits 6
  - description 15
  - description of callable services 24
  - generating and verifying 15
  - how used 6
- multiple encipherment 30

## N

- networks
  - translation of data and PINs across 7
- NIST Data Encryption Standard (DES) 4
- noncompatibility mode
  - installation option 20
  - with PCF, description 38
- nonrepudiation 1
  - using digital signatures 3

- notarization 34
- Notices 85

## O

- operating system requirement 43
- operational key form 25
- options
  - setting alternative 54

## P

- PARMLIB member
  - and ICSF options 54
- partial notarization 34
- pass phrase initialization 28
- PCF
  - applications 38
  - changing operating mode 54
  - compatibility with ICSF 23
  - macros 38
  - migrating macro exits 37
  - SMF records 53
- PCI Cryptographic Accelerator
  - description 43
- PCI Cryptographic Coprocessor
  - description 45
- PCI X Cryptographic Coprocessor
  - description 43
- performance
  - considerations when adding records to CKDS 54
  - consistent with hardware 18
- personal identification number (PIN)
  - description 15
  - description of callable services 23
  - exchanging 29
  - exchanging with other systems 39
  - how used 5
  - keys 27
  - translation across networks 7
- PIN block format
  - ECI Format 2 15
  - ECI Format 3 15
  - IBM 3621 format 15
  - IBM 3624 format 15
  - IBM Encrypting PINPAD format 15
  - ISO Format 0 15
  - ISO Format 1 15
  - VISA Format 2 15
  - VISA Format 3 15
  - VISA Format 4 15
- PIN generation and verification algorithm
  - 3624 Institution-Assigned PIN 15
  - 3624 PIN offset 15
  - IBM German Bank Pool PIN 15
  - Interbank PIN 15
  - VISA PIN 15
- PIN keys 27
- PKA cryptographic key data set (PKDS)
  - controlling access to 51
  - description 33

- PKA cryptographic key data set (PKDS) *(continued)*
  - disk copy 34
  - dynamic update using callable services 34
- PKA keys
  - description 27
- PKA master keys
  - Cryptographic Coprocessor Feature 28
  - key management master key (KMMK) 28
  - PCI Cryptographic Coprocessor 28
  - PCI X Cryptographic Coprocessor 28
  - signature master key (SMK) 28
- PKDS
  - dynamic update callable services 24
  - updating 17
- PKDSCACHE option 21
- PL/I high-level language
  - callable services 23
- plaintext 1
- planning considerations 43
- PR/SM partitions
  - separate master key in each PR/SM partition 21
- productivity
  - reducing costs by improving 18
- programming interface
  - improved productivity 18
  - summary 23
- protecting DES keys 24
- public key algorithms 4
- public key cryptography 3

## R

- random numbers
  - description of callable service 23
- reducing costs by improving productivity 18
- remote key
  - replaced by transport key 26
- restrictions
  - running ICSF and 4753-HSP 38
- retained RSA private keys
  - listing and deleting
    - description of callable service 23
- RETKEY macro
  - exit considerations 37
  - SMF records 53
- RSA
  - algorithm 4
  - BSAFE Toolkit 40
- RSA encrypted DATA keys
  - exchanging 30
  - key exchange 30
- RSA key pair
  - generation 28
- RSA protected DES key exchange 13
- RSA public and private key pairs
  - generation
    - description of callable service 23
- running 4753-HSP applications under ICSF 38
- running PCF applications under ICSF 37

## S

- S/390 G6 Enterprise Server
  - configurations 51
- scheduled changes for cryptographic keys 51
- secret key cryptography 2
- secure key import callable service
  - can be used only in special secure mode 52
- Secure Sockets Layer (SSL) 8
- security management 1
- sending cryptographic keys to other installations 52
- services
  - controlling access to 51
- SET Certificate Authority 45
- SET Secure Electronic Transaction 7
- SET Software Development Kit 8
- shortcut keys 83
- signature master key (SMK) 28
- single-length key
  - using 26
- SMF records
  - generated by ICSF 53
  - generated by PCF macros 53
- special secure mode
  - allows clear keys and PINs 52
  - description 31
  - enabling 31
  - IBM @server zSeries 900 31
  - IBM @server zSeries 990 32
  - installation option 20
  - PCI Cryptographic Coprocessor 31
  - PCI X Cryptographic Coprocessor 32
  - using for clear key entry 31
- starting ICSF 54
  - exits 20
  - START operator command options 54
- stopping ICSF 54
  - exits 20
  - STOP operator command options 54
- SYM-MK
  - master key 26
- symmetric cryptographic system 2

## T

- tampering
  - SMF records generated 53
- TKE workstation 10, 14, 18
- translating ciphertext 23
- transport key 26
  - example of use 29
  - how used 12
  - used to send cryptographic keys to other installations 52
  - variant 25, 39
- transporting data across a network 10
- triple DES
  - for data privacy 44
- triple-length key 28
- trusted key entry 10, 14
- types of DES keys 25

## U

- UDX option 21
- UDX support 62
- updating the CKDS 17, 32
- updating the PKDS 17
- User Defined Extension 62
- using different configurations 46, 51
- using ICSF exits to meet special needs 20
- using RSA encryption 30

## V

- variant
  - master key 25
  - transport key 25
- verifying
  - customer identity 27
  - PINs 23
- virtual storage constraint relief
  - for the caller of ICSF 40
- VISA card-verification value (CVV) 7
- VISA Format 2 15
- VISA Format 3 15
- VISA Format 4 15
- VISA PIN, through a VISA PIN validation value (VISA PVV) 15
- VTAM session-level encryption
  - and ICSF 40

## W

- WAITLIST option 21

## Z

- z/OS ICSF
  - operating system requirement 43

---

# Readers' Comments — We'd Like to Hear from You

z/OS  
Cryptographic Services  
Integrated Cryptographic Service Facility  
Overview

Publication No. SA22-7519-04

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



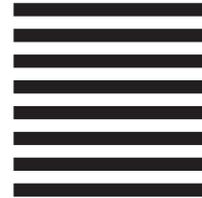
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
Department 55JA, Mail Station P384  
2455 South Road  
Poughkeepsie, NY  
12601-5400



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5694-A01, 5655-G52

Printed in USA

SA22-7519-04

