IBM

Monitoring System z Cryptographic Services

- System z software tools to monitor hardware cryptography activity
- Special focus on z/OS ICSF process flow and RMF reporting
- Expert considerations for RMF value interpretation

Patrick Kappeler
Guillaume Hoareau
Gérard Laumay
Dominique Richard
Jean-Marc Darees
Joël Porterie

ibm.com/redbooks

Redpaper

International Technical Support Organization

Monitoring System z Cryptographic Services

February 2008

REDP-4358-00

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (February 2008)

This edition applies to Version 1, Release 9 of z/OS (product number 5694-A01).

Contents

Notices

This information was developed for products and services offered in the U.S.A.

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 U.S.A.

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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo)  ®
eServer™
z/Architecture™
z/OS®
z/VM®
z/VSE™
zSeries®
z9™
FICON®
GDPS®
IBM®
MVS™
OMEGAMON®
Parallel Sysplex®
Processor Resource/Systems Manager™
Redbooks™
RACF®
RMF™
S/390®
System z™
System z9™
Tivoli Enterprise™
Tivoli®
VTAM®

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This IBM® Redpaper is intended to provide IBM System z™ hardware cryptography users with an overview of software tools that they can use to monitor and assess the workload that is being driven to cryptographic coprocessors. It also provides information about IBM published measurements that pertain to the performance of System z hardware cryptography. A specific chapter is dedicated to the reports that are generated by the z/OS® Reporting and Management Facility (RFM) and how you can use and interpret them.

RMF is the IBM strategic product for z/OS performance measurement and management. It collects performance data for z/OS base and sysplex environments and issues reports that can be used to monitor system performance so that users can optimally tune and configure their systems to meet business needs.

The team that wrote this paper

This paper was produced by a team of specialists from the Montpellier Products and Solutions Support Center (PSSC), France, on behalf of the International Technical Support Organization (ITSO), Poughkeepsie Center.

Patrick Kappeler is a lead consulting IT Specialist in the Montpellier PSSC. During his 37-year career with IBM, he has held many international positions, all dealing with mainframe hardware and software technical support and education. He extensively presents, writes, and provides advanced technical support and consulting on this topic worldwide. He is also the co-author and leader of many other ITSO projects on z/OS and business security.

Guillaume Hoareau is an IT Specialist at the New Technology Center of the PSSC in Montpellier. He is responsible for ISV Sizing Support on the System z platform for the Virtual International Competency Center at Montpellier. He has worked on several z/VM® and Linux® on System z projects.

Gérard Laumay is a System z IBM certified IT Specialist at the Montpellier PSSC. He has more than 21 years of experience in the large systems field as a consultant for IBM clients. His areas of expertise include IBM System z9™ hardware, z/OS, z/VM, Linux operating systems, and new workloads on System z. A member of the zChampion worldwide technical team, he teaches at numerous conferences. A frequent participant in international projects, he has written several IBM Redbooks™.

Dominique Richard is an IT Specialist for IBM France. He joined IBM in 1982 as a System Engineer supporting MVS™ Customers in France. Since 2005, he has been involved in benchmarks with the Montpellier PSSC. He specializes in host system security.

Jean Marc Darees joined IBM in 1984 as an MVS system engineer. He has held several specialist and architect positions dealing with mainframes and other technologies. He joined the Montpellier PSSC in 1997, where he now provides consulting and pre-sales technical support in the area of IT infrastructure for enterprise clients.

Joël Porterie is a Senior IT Specialist who has been with IBM France for 30 years. He works for Network and Channel Connectivity Services in the EMEA Product Support Group. His areas of expertise include z/OS, TCP/IP, VTAM®, OSA-Express, and Parallel Sysplex® for zSeries®. He has taught OSA-Express and FICON® problem determination classes and

provided on-site assistance in these areas in numerous countries. He also co-authored *Using the IBM S/390 Application StarterPak*, SG24-2095; *OSA-Express Gigabit Ethernet Implementation Guide*, SG24-5443; *OSA-Express Implementation Guide*, SG24-5948; *Introduction to the New Mainframe: Networking*, SG24-6772; and *Communications Server for z/OS V1R7 TCP/IP Implementation, Volume 4 Policy-Based Network Security*, SG24-7172.

Thanks to the following people for their contributions to this project:

Paola Bari
ITSO Poughkeepsie

Richard Conway
ITSO Poughkeepsie

Roberto Calderon
IBM Tivoli® OMEGAMON® Technical Enablement - Worldwide

John Fiedler
IBM Tivoli Development - USA

Ingo Franzki
IBM z/VSE™ Development - Boeblingen Laboratory

Matthias Gubitz
IBM RMF™ Development - Boeblingen Laboratory

Michael Kelly
ICSF Development - Poughkeepsie Laboratory

Mark Bidwell
z/OS Middleware Performance - Endicott Laboratory

Greg Boyd
Washington System Center

Bruno Lahousse
IBM GDPS® Solution Test - PSSC Montpellier

Pascal Marachian
IBM Data Centers Network support - PSSC Montpellier

Pascal Tillard
IBM System z9 Benchmark Center - PSSC Montpellier

Klaus Werner
IBM Processor Firmware Development - Boeblingen Laboratory

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

System z hardware cryptography implementation: A brief review

IBM System z9 implements enhanced functions for the cryptographic facilities that are already available for z990 and z890: Crypto Express 2 (CEX2C) and Central Processor Assist for Cryptographic Functions (CPACF). This chapter is intended to help readers that are familiar with integrated mainframe hardware cryptography by:

- Covering the positioning of the facilities in the overall cryptography infrastructure of System z
- Reviewing their exploitation environments
- Providing general information about the hardware cryptographic services available in System z9 as of the writing of this book

Note: The z990 CPACF and PCIX Cryptographic Coprocessor (PCIXCC) card are described in *IBM eServer zSeries 990 (z990) Cryptography Implementation*, SG24-7070.

As is the case with previous implementations of hardware cryptography in zSeries systems, the Integrated Cryptographic Service Facility (ICSF) component of z/OS provides the IBM Common Cryptographic Architecture (CCA) for invoking the hardware cryptographic services.

1.1 Cryptographic function support in System z9

System z9 includes both standard cryptographic hardware and optional cryptographic features for flexibility and growth capability.

The System z secure coprocessors (CEX2Cs) implement all the protection mechanisms that are required by the FIPS 140-2 standard and have been certified at the highest possible level of FIPS 140-2 level 4. Information on FIPS 140-2 certification can be found at:

http://csrc.nist.gov/cryptval/140-2.htm

System z9 cryptographic functions include the full range of cryptographic operations needed for e-business, e-commerce, and financial institution applications. In addition, custom cryptographic functions can be added to the set of functions that System z9 offers.

1.2 Overview of the cryptographic devices in System z9

Two types of cryptographic hardware devices are available in System z9. However, the cryptographic hardware facilities are usable only when they have been explicitly enabled using the Feature Code 3863, except for the CPACF SHA-1 and SHA-256 functions, which are always enabled regardless of whether the feature code is installed or not.

1.2.1 CPACF

Each system central processor has an assisting processor on the chip to support cryptography. The CPACF is actually considered part of the system processing unit (PU) hardware and, as such, is not really a coprocessor or an accelerator. Strictly speaking, it is a a set of instructions that are implemented in the PU, or the so-called Message Security Assist (MSA) instructions.

CPACF offers a set of symmetric cryptographic functions that enhance the encryption and decryption performance of clear key operations for SSL, VPN, and data storing applications that do not require FIPS 140-2 level 4 security. The MSA instructions provide for Data Encryption Standard (DES), T-DES, AES data encryption and decryption, message authentication codes (MACs), and SHA-1 and SHA-256 hashing. These functions are directly available to application programs because they are provided as problem state z/Architecture™ instructions, diminishing programming overhead. Alternatively, these functions can also be called though the Integrated Cryptographic Service Facility (ICSF) component of z/OS by an ICSF-aware application. The following problem-state instructions were introduced with the z/990 and z/890 cryptographic assist functions:

- KMAC: Compute Message Authentic Code
- KM: Cipher message

  The KM instruction has been extended in System z9 to support the AES-128 encryption and decryption function.
- KMC: Cipher message with chaining

  The KMC instruction has been extended in System z9 to support the AES-128 and Pseudo Random Number Generator (PRNG) functions.
- KIMD: Compute Intermediate Message Digest

  The KIMD instruction has been extended in System z9 to support the SHA-256 function.
- KLMD: Compute Last Message Digest

The KLMD instruction has been extended in System z9 to support the SHA-256 function.

CPACF runs at System z9 processor speed. Because the facility is available for every CP in the system, there are no affinity issues such as those for earlier CMOS processors.

As an alternative to direct calls to the CPACF, an application can call the ICSF services CSNB One-Way-Hash (CSNBOWH), CSNB Symmetric Key Encrypt (CSNBSYE), and CSNB Symmetric Key Decrypt (CSNBSKD). ICSF routes these requests to CPACF for execution.

Note: CPACF is physically local to each PU, so all CPACF operations are synchronous.

1.2.2 CEX2C

CEX2C is a Peripheral Component Interconnect Extended (PCI-X) pluggable feature that provides a high-performance and secure cryptographic environment. CEX2C implements the master key concept. A master key in the coprocessor hardware enclosure protects applications by encrypting secure keys, or keys that cannot be compromised.

CEX2C consolidates the functions that were previously offered on the z900 by the Cryptographic Coprocessor (CCF), the PCI Cryptographic Coprocessor (PCICC), and the PCI Cryptographic Accelerator (PCICA) features. (As a result, these functions are no longer available for System z9). CEX2C performs the following functions:

- Data encryption and decryption algorithms using secure keys (that is, keys encrypted with the symmetric master key or key-encrypting-key):
  - DES
  - Double length-key DES
  - Triple length-key DES
- Generation and distribution of DES, double and triple-DES keys
- PIN generation, verification, and translation functions, using secure keys

It also features the PRNG and the Public Key Algorithm (PKA) Facility, using secure or clear keys. The PKA Facility is intended for application programs that exploit public key algorithms, including importing RSA public-private key pairs in clear and encrypted forms. Other PKA Facility features include:

- RSA

  The following operations are executed with RSA keys of up to 2048 bits in length, or up to 4096 bits for System z9 with ICSF FMID HCR7750 and above, and the related CEX2C firmware update:

  - Key generation
  - Signature generation and verification
  - Import and export of DES keys under an RSA key
- Public Key Encrypt (CSNDPKE)

  This service assists with the SSL/TLS handshake. With the Mod_Raised_to Power (MRP) function, it can off load computing intensive portions of the Diffie-Hellman protocol.
- Public Key Decrypt (CSNDPKD)

  Public Key Decrypt supports a zero-pad option for clear RSA private keys. CSNDPKD performs raw hardware RSA private operations such as those required by the SSL/TLS handshake and digital signature generation. Linux exploits the zero-pad option so that it can use the System z9 CEX2C features for improved performance of the SSL/TLS handshake and digital signature generation.

- Derived Unique Key Per Transaction (DUKPT)

  This service writes applications that implement the DUKPT algorithms defined by the ANSI X9.24 standard. DUKPT provides additional security for point-of-sale transactions that are standard in the retail industry. CEX2C supports DUKPT algorithms for triple-DES with double-length keys.

- Europay Mastercard VISA (EMV) 2000 standard

  Applications may be written to comply with the EMV 2000 standard for financial transactions between heterogeneous hardware and software. Support for EMV 2000 applies only to the CEX2C feature of the System z9.

Other functions of CEX2C serve to enhance the security of public/private key encryption processing:

- Retained key support (the generation of RSA private keys that are stored within the secure hardware boundary), which is currently not recommended for System z
- Support for 4753 Network Security Processor migration
- User Defined Extensions

  User-Defined Extensions (UDX) to CCA support custom algorithms that run in CEX2C. UDX is added as specific coprocessor code built by IBM or by an approved third party. Building a UDX is an IBM service offering performed under contract.

Note: Because of the physical implementation of CEX2C in System z, which involves delays that are created by cable length, access to hardware storage areas, and more, all ICSF operations that invoke CEX2C are asynchronous. ICSF starts the operation and then periodically polls CEX2C for completion of the operation. The calling application is held until after the coprocessor operation completes.

The coprocessors in CEX2C

CEX2C contains two coprocessors, or *cards*, of the IBM 4764-001 cryptographic coprocessor model. The 4764-001 is also called PCIXCC. The PCIXCC feature was a predecessor to the CEX2C feature for the early z990 systems. The CEX2C can therefore be considered to provide all the functions that are available in the z990/z890 PCIXCC with a doubled throughput.

Attention: A *card* in the IBM context is a coprocessor; what is actually being plugged in the system is a *feature* (which contains one or two cards). Cards are also designated as *Adjunct Processors* (AP) in System z hardware terminology.

Setup, controls, and status of hardware cryptographic coprocessors in System z are always performed or given at the coprocessor, or card, level.

The CEX2C feature is designed for the FIPS 140-2 Level 4 compliance rating for secure cryptographic hardware modules. Among the many protective functions that the standard requires is that an unauthorized removal of the card or feature should *zeroize* its contents, including the master keys, to preserve secrecy.

With the CEX2C coprocessor in System z9, a user can perform these actions using application keys protected by a master key (*secure key*s):

- Encrypt and decrypt data using secret-key algorithms. Triple-length key DES and double-length key DES algorithms are supported.

- Generate, install, and distribute cryptographic keys securely using both public and secret key cryptographic methods.
- Generate, verify, and translate personal identification numbers (PINs).
- Ensure the integrity of data using MACs, hashing algorithms, and the RSA PKA digital signatures.

The security-relevant portion of the cryptographic functions is performed inside the secure physical boundary of a tamper-resistant card. Master keys and other security-relevant information are also maintained inside this secure boundary.

1.2.3 Crypto Express 2 Accelerator (System z9 only)

The Crypto Express 2 Accelerator (CEX2A) is actually a CEX2C that has been re-configured by the user so that it only provides a subset of the CEX2C functions at enhanced speed. This re-configuration is a manual process that is performed by the System z9 Support Element or HMC and is intended to increase the throughput of hardware assisted SSL/TLS handshakes. It has the following characteristics, benefits, and limitations:

- The re-configuration is done at the coprocessor level. A CEX2C feature can host a CEX2C coprocessor and a CEX2A, or two CEX2C coprocessors, or two CEX2As.
- The re-configuration works both ways, from CEX2C to CEX2A and CEX2A to CEX2C. Master keys in the CEX2C domains can be optionally preserved when re-configuring from CEX2C to CEX2A.
- The re-configuration process disrupts the operations of the coprocessor/accelerator involved. The coprocessor/accelerator must be deactivated at all ICSF instances (that is, all logical partitions) that use it before engaging the manual re-configuration process.
- The FIPS 140-2 certification is not relevant to CEX2A because it is operating with clear keys only.
- The extension capability through UDX is not available to CEX2A.

Actually, the only CEX2C functions that remain available when re-configured into a CEX2A are the former PCICA functions. These functions are used for the acceleration of modular arithmetic operations (that is, the RSA cryptographic operations used with the SSL/TLS protocol):

- CSNDPKD with PKCS-1.2 formatting
- CSNDPKE with ZERO-PAD formatting
- Digital signature verify

The encrypt and decrypt functions support key lengths of 512 to 2048 bits, or up to 4096 bits on System z9 with the November 2007 ICSF and CEX2C firmware updates, in the Modulus Exponent (ME) and Chinese Remainder Theorem (CRT) formats.

The maximum number of SSL transactions per second that can be supported in System z9 by any combination of CPACF and CEX2A coprocessors is limited by the amount of cycles that are available to perform the software portion of the SSL/TLS transactions. When both PCI-X coprocessors on a CEX2C feature are configured as accelerators, the CEX2C feature is designed to perform up to 6000 SSL handshakes per second and per feature. This represents, approximately, a performance improvement of three times that of the z990 when it is using either a PCICA feature, or the current CEX2C feature.

Note: These figures indicate throughput, which, in this case, is what is required to initiate several threads of parallel requests to the CEX2A to achieve this level of performance.

In System z9, there can be a maximum of eight CEX2C features, and all coprocessors in these eight features can be re-configured as CEX2As.

1.3 Hardware coprocessors and logical partitioning review

Each CEX2C coprocessor hosts 16 domains, and each domain can be thought of as a set of hardware registers dedicated to an active logical partition. The domain that a logical partition can have access to is specified in the image profiles of the partition. A domain is intended to hold the master keys that have been set by the cryptographic support software running in the logical partition (for example, ICSF for z/OS), or, when using the CEX2C coprocessor in accelerator mode, the domain refers to a queue number that the coprocessor dedicates for requests arriving from this logical partition.

The user also specifies, in the logical partition image profile, which coprocessor (or AP, in system hardware configuration terminology) the partition has access to. This means, therefore, that the real entity that the logical partition has access to is determined by the combination <*AP number*>.<*domain number*>.

Important: The combination <AP number>.<domain number> is unique in the whole set of active logical partitions in the system, which means that no more than one active logical partition can obtain access to a specific <AP number>.<domain> combination.

Note that the coprocessors that a logical partition has access to are specified in two lists in the partition image profile:

- The online list, where the coprocessors are made available to the logical partition as soon as it has been activated.
- The candidate list, which specifies which coprocessors the partition can have access to; however, these coprocessors must previously be varied manually online to the logical partition. The candidate line can specify coprocessors that are not yet physically available in the system.

Note: The CPACF facility is implicitly shared because any logical partition dispatched on a PU will systematically get access to its CPACF.

Refer to “System z9 Processor Resource/Systems Manager™ Planning Guide”, SB10-7041 for further details on the setup of logical partitions access to the cryptographic coprocessors.

1.3.1 Configuration data summary

Table 1-1 on page 7 summarizes the support of partitions for CEX2C and CEX2A on System z9.

Table 1-1 PCI -X Cryptographic features

	Maximum number of features per System z9 server	Number of cryptographic coprocessors or accelerators per feature	Maximum number of cryptographic coprocessors or accelerators per System z9 server	Number of cryptographic domains per accelerator[a] or coprocessor	Number of logical partitions per System z9 server (Defined/Active)
CEX2C	8	2	16	16	60/60

a. Although an accelerator is not using a cryptographic domain to protect a master key, the notion of domain still exists and refers to a request/response queue that the device maintains with a specific logical partition.

1.3.2 System z9 hardware cryptography feature codes

Table 1-2 lists the main cryptographic features available with System z9.

Table 1-2 System z9 cryptographic features

Feature code	Description
3863	Crypto enablement CD, a prerequisite for the use of the CPACF (except for SHA-1 and SHA-256) and of the CEX2C/CEX2A hardware features. The feature is installed once and applies to the whole system.
0863	CEX2C feature with two coprocessors. Each coprocessor is also re-configurable as a CEX2A.
0870	Crypto Express 2-1P (CEX2-1P) feature with one coprocessor. The coprocessor is also re-configurable as CEX2A. The CEX2-1P feature is available only for the System z9 BC.
0859	TKE V5 hardware with Ethernet connection for up to three features per System z9.

Important: Clients must use the TKE V5 workstations to control the System z9 CEX2Cs. The TKE V5 workstations can also be used to control the cryptographic coprocessors for z990, z890, z900, and z800 servers. However, previous TKE versions cannot be upgraded to TKE V5 hardware.

TKE workstation feature

A TKE workstation is part of a customized solution for using ICSF to manage the cryptographic keys of a System z9 system with CEX2C features that have been installed for the purpose of using of DES and PKA with secure cryptographic keys. The TKE workstation provides secure control of the CEX2C features, including the loading of master keys.

If one or more logical partitions are customized for using CEX2C cards, the TKE workstation can be used to manage DES master keys and PKA master keys for all cryptographic domains of each CEX2C coprocessor that is defined to the TKE workstation.

Each logical partition in the same physical system that is using a domain managed through a TKE workstation connection is either a TKE host or a TKE target. A logical partition with TCP/IP connection to the TKE is referred to as TKE host; all other partitions are TKE targets.

The cryptographic controls that set for a logical partition, through the System z9 Support Element, determine whether it can be a TKE host or TKE target.

1.4 System z9 cryptographic features comparison

Table 1-3 summarizes the functions and attributes of System z9 cryptographic hardware features. Note that:

- The latest ICSF release available as of the writing of this book is FMID HCR7750.
- You can add CEX2C/CEX2A features without being disruptive by predefining the logical partition with the appropriate PCI-X processor number in the partition image profile.
- Linux does not require CPACF enablement if only the RSA clear key operations of CEX2C are being used. DES or T-DES encryption requires CPACF, even when invoked from Linux.
- CEX2C/CEX2A is assigned two PCHIDs per feature (one per coprocessor or accelerator).

Table 1-3 System z9 cryptographic features comparison

Functions or attributes	CPACF	CEX2C	CEX2A
Supports z/OS applications using ICSF	X		X
Encryption and decryption using secret-key algorithms		X	
Highest SSL handshake performance			X
Highest symmetric (clear key) encryption performance	X		
Highest symmetric (clear key) encryption performance			X
Highest symmetric (encrypted key) encryption performance		X	
Process for adding features is disruptive		X	X
Requires IOCDS definition			
Uses CHPID numbers			
Is assigned PCHIDs		X	X
Physically embedded on each PU	X		
Requires CPACF Enablement FC 3863	X	X	X
Requires ICSF to be active		X	X
Offers user programming function support (UDX)		X	
Usable for data privacy (encryption and decryption processing)	X	X	
Usable for data integrity (hashing and message authentication)	X	X	
Usable for financial processes and key management operations		X	
Crypto performance RMF monitoring		X	X
Requires system master keys to be loaded		X	
System (master) key storage		X	
Retained key storage		X	
Tamper-resistant hardware packaging		X	
Designed for FIPS 140-2 Level 4 certification		X	
Supports SSL functions	X	X	X
Supports Linux applications performing SSL handshakes			X

Functions or attributes	CPACF	CEX2C	CEX2A
RSA functions		X	X
High performance SHA-1 and SHA-256 Hash function	X		
Clear key DES/T-DES	X		
AES 128-bit key	X		
Pseudo Random Number generator	X	X	
Clear key RSA		X	X
Double length DUKPT support		X	
Europay Mastercard VISA (EMV) support		X	
Public Key Decrypt (PKD) support for Zero-Pad option for clear RSA private keys)		X	X
Public Key Encrypt (PKE) support for MRP function		X	X

1.5 z/OS hardware cryptography infrastructure review

Figure 1-1 shows the overall hardware and software layout of System z and z/OS hardware cryptography.

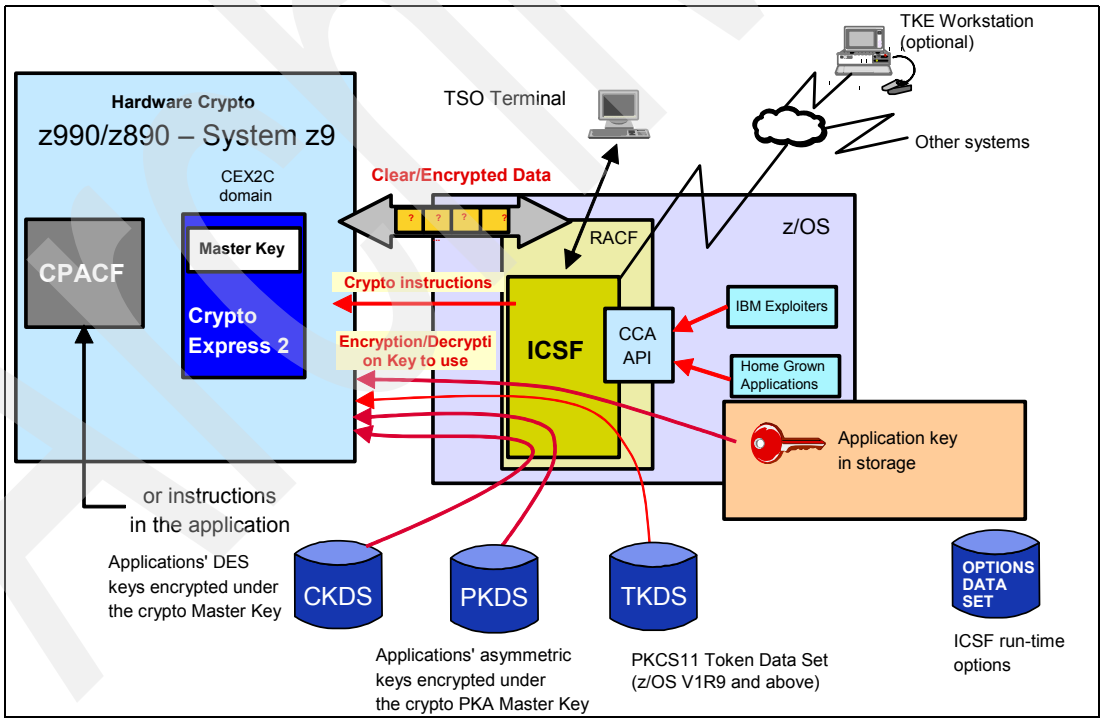


Figure 1-1 Overall hardware and software layout

As you can see, CPACF, CEX2C, and CEX2A have specific software requirements. ICSF is the support program for the cryptographic features of CPACF (as an alternative replacement of the direct use of machine instructions), CEX2C, and CEX2A.

As of the writing of this book, the ICSF release that is delivered with z/OS V1R9 is FMID HCR7740. A newer ICSF release (FMID HCR7750) has been made available, as an SMP/E installable Web deliverable (“Cryptographic Support for z/OS V1R7-V1R9 and z/OS.e V1R7-V1R8”) and can be found at:

`http://www.ibm.com/eserver/zseries/zos/downloads`

The exploiters of the cryptographic services call the ICSF API. Some functions are performed by the ICSF software without invoking the cryptographic coprocessor; other functions require ICSF to execute routines that contain the crypto instructions. The crypto instructions to drive the CEX2C are IBM proprietary and are not disclosed; the crypto instructions to interface with the CPACF are published in the z/Architecture Principles of Operation. These instructions are executed by a CPU engine and, if not addressing the CPACF functions, result in the generation of a work request for a cryptographic coprocessor.

The crypto coprocessor includes:

- Data to encrypt or decrypt from the system memory.
- The keys used to encrypt or decrypt provided by ICSF at the request of the exploiter. These encryption/decryption keys are themselves encrypted and, therefore, unusable when they are outside the crypto coprocessor. Physically, these keys can be stored in ICSF-managed VSAM data sets and pointed to by the application that is using the label they are stored under.
- The Cryptographic Key Data Set (CKDS) used to store the symmetric keys in their encrypted form.
- The Public Key Data Set (PKDS) used to store the asymmetric keys. The application can also provide an encrypted encryption key or a clear encryption key directly to the memory (that is, to use *as is*) of the coprocessor.
- The Token Key Data Set (TKDS), an optional data set for z/OS V1R9 and above. ICSF uses this data set for the storage and management of the z/OS PKCS#11 tokens.

For high-speed access to symmetric cryptographic keys, the keys in the CKDS are duplicated into an ICSF-owned data space.

Note: Although the software runs some services for the sake of performance, ICSF tasks will not to start if there is no hardware crypto facility available in the system (that, is at least one CPACF in operational status).

2

Hardware cryptography activity assessment on System z

In this chapter, we provide an overview of the System z infrastructures that exploit hardware cryptography and we introduce the methods that you can use to assess whether and how heavily the System z hardware cryptographic services are invoked. The hardware cryptography “commodity” in System z is today an option for many applications or middleware that otherwise use their own software cryptographic code. Because of the push to make operations as “transparent” as possible to users, these users are sometimes uncertain as to whether the application or the middleware is really exploiting the System z integrated hardware cryptography instead of using its own software implementation of the services.

In this chapter and those that follow, we describe ways of obtaining confirmation, if needed, that the hardware cryptographic devices are being used and how to assess, whenever possible, their utilization ratio.

2.1 Who is exploiting the System z hardware cryptography?

The System z cryptographic devices, CPACF and CEX2C (either in coprocessor or accelerator mode), are exploited by applications or middleware in these operating systems:

- z/OS
- z/VSE
- Linux for System z

Note: We did not include z/VM because it acts as an invisible link between the guest virtual machine and the cryptographic hardware. However, it can provide configuration data pertaining to hardware cryptography resources (for more information, see 2.5, “Setting up the hardware cryptography configuration of z/VM” on page 28).

2.1.1 Hardware cryptography exploitation on z/OS

Figure 2-1 summarizes, at a high level, the z/OS cryptographic APIs as of z/OS V1R9, in addition to the direct access they might have to the CPACF hardware using the MSA instructions. (For more details, see *IBM eServer zSeries 990 (z990) Cryptography Implementation*, SG24-7070.)

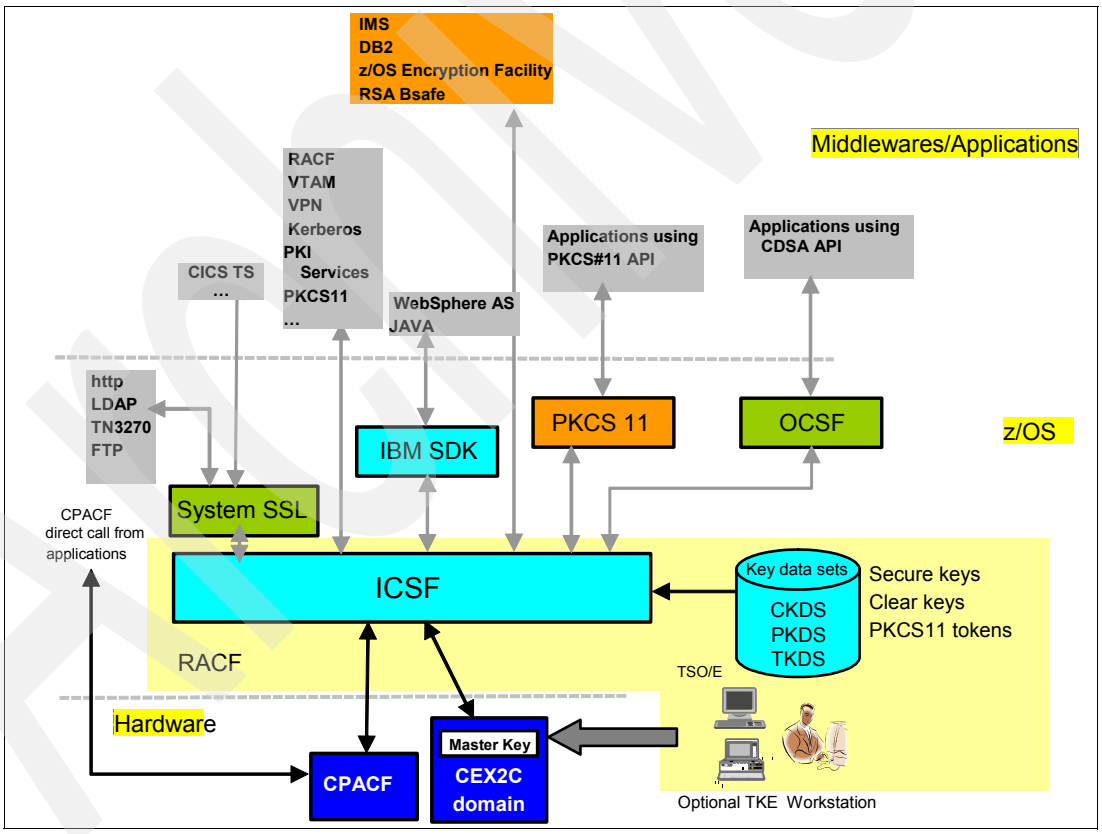


Figure 2-1 z/OS Hardware Cryptography infrastructure

The ICSF APIs are the lowest level of cryptographic APIs that z/OS provides. They are:

- The IBM CCA API
- A subset of the RSA PKCS#11 API, which was introduced in z/OS V1R9 and provided as C/C++ libraries

Note: The ICSF PKCS#11 API implementation relies on already existing ICSF functions with the consequence that ICSF and hardware cryptographic activities of PKCS#11 API are actually the execution of CCA services and are reported as such from the utilization standpoint.

The System SSL API is at a higher level than ICSF. System SSL converts the cryptographic service requests it receives from applications into ICSF CCA service calls, or direct invocations of the CPACF. System SSL is intended to provide applications with SSL/TLS runtime support. System SSL does not rely exclusively on the System z hardware cryptographic devices to provide the requested services because it also has a software implementation of these services in case hardware cryptography is unavailable on the system. z/OS System SSL is a typical example of where it might be uncertain, at first glance, for the user to discern whether hardware or software cryptography is currently being used.

The OCSF API is another high level API, and it is the z/OS implementation of the Intel® Common Data Security Architecture (CDSA) set of cryptographic services. The exploiting application or middleware must select the cryptographic service provider it wants to "attach" to. One of these service providers is actually ICSF, invoked by the CCA API. When hardware cryptography is not available in the system, then the ICSF cryptographic provider fails.

The IBM SDK also provides access, for Java™ exploiters, to the ICSF CCA API; however, this implies that the Java cryptographic service providers have been properly specified. Note that, even with the Java hardware cryptography provider specified, some cryptographic services are still provided by software.

Hardware cryptography utilization measurement in z/OS

You can assess hardware cryptography exploitation in z/OS using messages or trace information that is built into the exploiting application or middleware. See "Assessing the use of hardware cryptography on z/OS" on page 16 for examples of such messages and traces.

The primary tool for measuring z/OS hardware cryptography utilization is the z/OS RMF product, which relies on data that is provided by the z/OS System Management Facility (SMF) and hardware data provided by CEX2C. The way RMF operates and how it can be used for measuring hardware cryptography activity is explained in "Using RMF to measure z/OS hardware cryptography activity" on page 34.

Alternatively, or in addition to using RMF, the hardware cryptography activity can be measured using the IBM Tivoli OMEGAMON XE for z/OS reporting product. For an example of how to use of OMEGAMON, see 4.1, "OMEGAMON XE for z/OS support for cryptographic coprocessors" on page 44.

2.1.2 Hardware cryptography exploitation on z/VSE

The main use of hardware cryptography in z/VSE stemmed originally from the need to provide hardware assistance to the SSL/TLS protocol. To meet this objective, specific cryptographic services were developed to be used internally in z/VSE, and a decision was made to externalize these services as the set of APIs in Figure 2-2 on page 14.

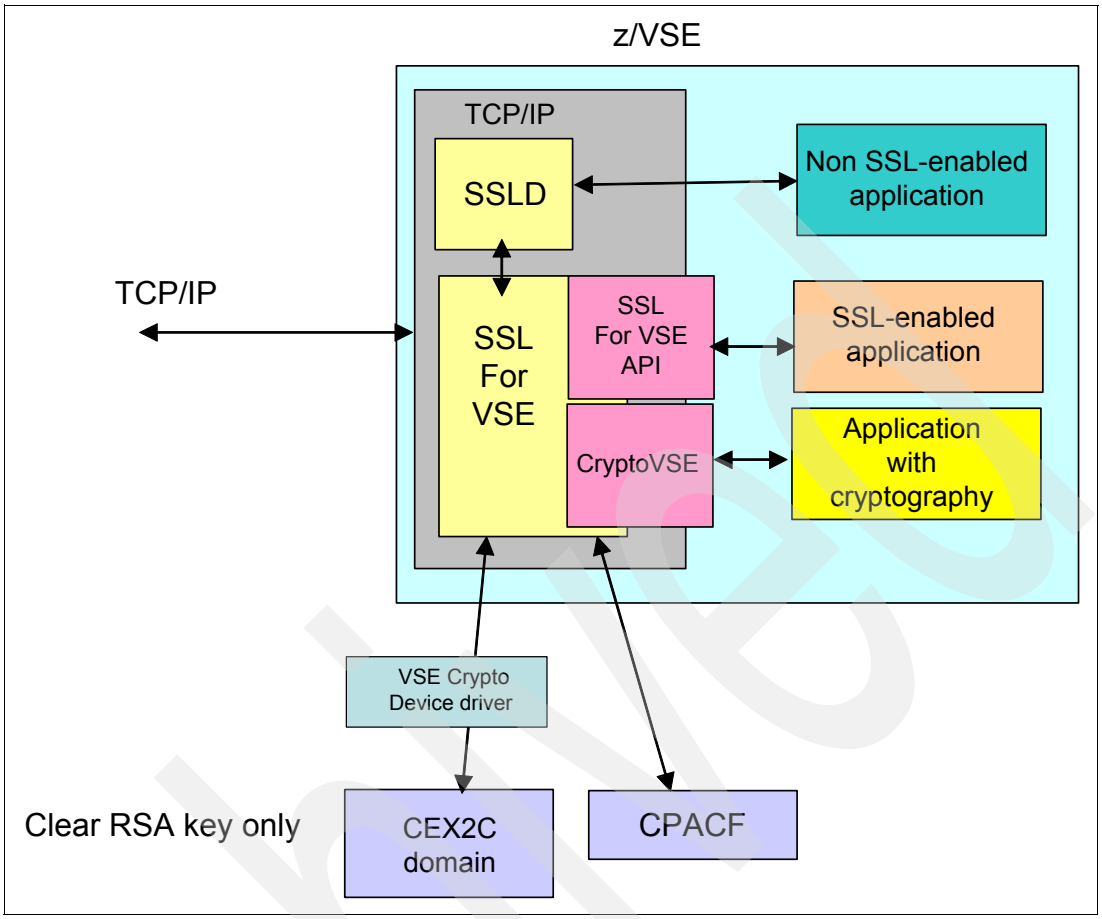


Figure 2-2 z/VSE Hardware Cryptography infrastructure

Note that, because of the initial intent of providing cryptographic support specifically for SSL/TLS, the z/VSE cryptographic APIs are hosted by the TCP/IP stack component. It provides the cryptographic services and invokes hardware cryptography when relevant and available. Hardware cryptography using CEX2C/CEX2A and CPACF is supported starting with TCP/IP for VSE 1.5D running on z/VSE 3.1.

RSA acceleration

z/VSE exploits the CEX2C or CEX2A for RSA acceleration only. That is, it uses clear keys for:

- RSA decryption or encryption operations during initiation of the SSL/TLS session
- RSA signature of certificates built by the z/VSE certificate utility

CPACF exploitation

CPACF is used for the SSL/TLS data transfer when one of the following symmetric algorithms has been agreed upon between the client and server: DES, Triple DES, or AES-128 (the latter on System z9 only). CPACF is also used when the SHA-1 algorithm has been selected for checking data integrity, during data transfer or when building a certificate.

Note: Whether hardware cryptography is used is not visible to the z/VSE TCP/IP applications. When it cannot be used, the software implementation of the cryptographic services in the TCP/IP stack is used. The use of hardware cryptography can be disabled by a setting in the so-called *$SOCKOPT Phase* for the TCP/IP stack.

Infrastructure

The lowest-level hardware cryptography API that is provided in z/VSE is the CryptoVSE API, where applications or middleware can call specific cryptographic services that are eventually performed by CPACF or CEX2C devices.

The SSL for VSE API is a higher level API, similar to the z/OS System SSL API, that provides SSL-enabled applications with runtime support for handling the SSL/TLS protocol. This API invokes the CryptoVSE functions.

The SSL Daemon (SSLD) is not an API. It manages the SSL/TLS protocol on behalf of non-SSL-enabled applications, much like z/OS Application- Transparent TLS (AT-TLS) does.

Hardware cryptography utilization measurement in z/VSE

z/VSE does not provide a measurement utility similar to z/OS RMF for reporting hardware cryptography activity; however, it is possible to obtain some statistical information (see 2.3, “Assessing the use of hardware cryptography on z/VSE” on page 24).

2.1.3 Hardware cryptography exploitation in Linux for System z

The hardware cryptography infrastructure for Linux for System z is shown in Figure 2-3.

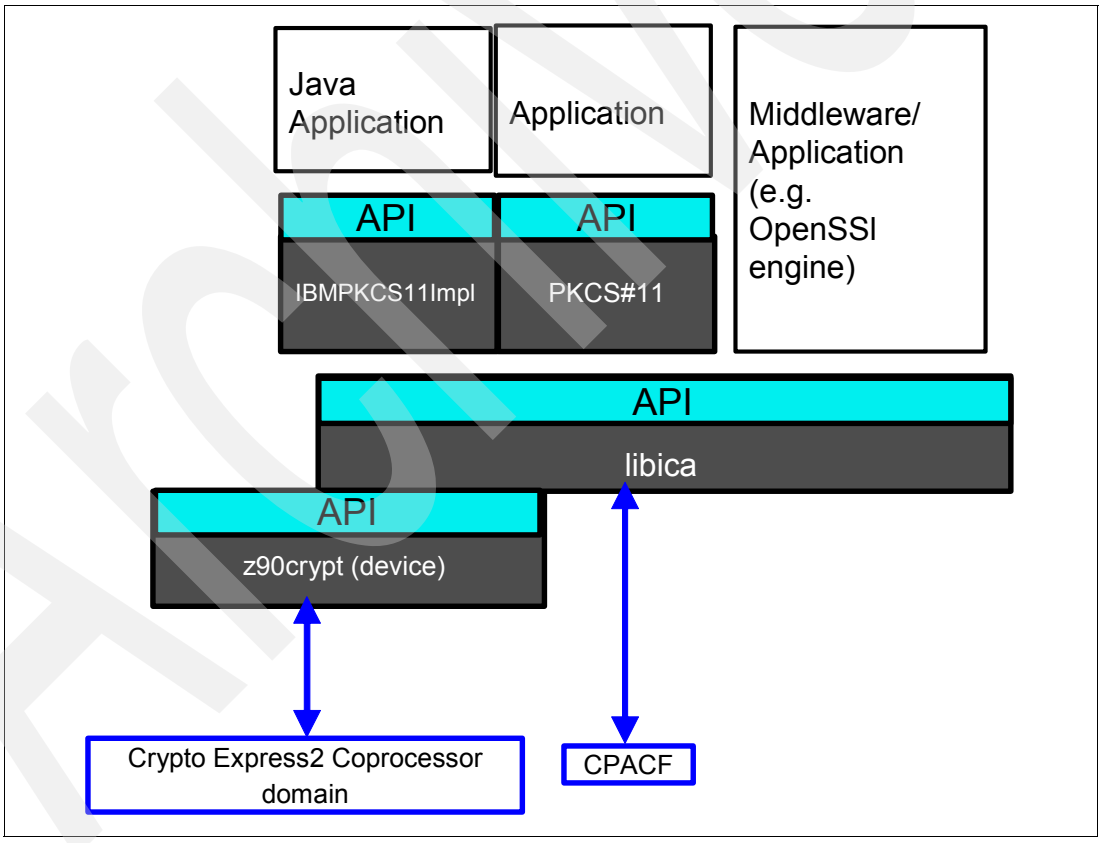


Figure 2-3 Linux for System z hardware cryptography infrastructure

There are globally three layers of APIs. At the lowest level, the *z90crypt*, or the more recently available *zcrypt*, the device driver provides an API that is usually not exploited directly by applications but instead is intended for intermediate software layers, which in turn provide more sophisticated cryptographic functions to the next upper level of code.

The libica Linux library is an intermediate cryptographic functions library that offers a wide range of cryptographic functions, some of them being performed by the hardware cryptographic devices under control of the device driver. As it stands today libica, in the majority of cases, is still not directly called by applications or middleware. They still rely on a higher level cryptographic API such as the PKCS#11 API, or its JAVA version, the IBMPKCS11Impl cryptographic API.

Hardware cryptography utilization measurement in Linux for System z

The hardware cryptography utilization assessment in Linux for System z can be achieved by using messages or traces that are issued by the exploiting application or middleware. z90crypt or zcrypt also maintains operations counters that can be used for an assessment of hardware cryptography activity (for more information, see 2.4, “Assessing the use of hardware cryptography on Linux for System z” on page 25).

2.2 Assessing the use of hardware cryptography on z/OS

This section describes three of the methods for assessing whether z/OS applications are actually calling the system hardware cryptography, as opposed to using the software implementation of the cryptographic services. These methods are:

- Detecting whether there are actual attempts to access cryptographic resources by protecting them with profiles in RACF®, namely profiles in the CSFSERV class
- Exploiting the tracing capabilities built in an application or middleware such as in the System SSL component of z/OS
- Exploiting the ICSF component trace

2.2.1 Detecting the use of RACF protected cryptographic resources

ICSF services can be protected by RACF profiles in the CSFSERV class of profiles (see the *z/OS Cryptographic Services Integrated Cryptographic Service Facility Administrator’s Guide*, SA22-7521).

The idea here is to detect access attempts by applications to RACF-protected ICSF services. We assume that the user does not want to prevent the request from proceeding and is therefore looking for a warning message rather than hard stopping the request with a RACF violation exception. One way to do this is to follow this process:

1. Activate the CSFSERV class of profiles:

   ```
   SETROPTS CLASSACT(CSFSERV) RACLIST(CSFSERV) GENERIC(CSFSERV)
   ```

2. Create the RACF profile ** in the CSFSERV class, with no access by default and the WARNING attribute:

   ```
   RDEF CSFSERV ** UACC(NONE) WARNING
   ```

3. Issue a REFRESH command to modify the ‘in-storage’ RACF profile

   ```
   SETROPTS RACLIST(CSFSERV) REFRESH
   ```

4. To verify this installation, issue the following command:

   ```
   RLIST CSFSERV ** ALL
   ```

Note: Another option is to audit all accesses to specific ICSF services by specifying AUDIT(ALL) in the relevant profiles.

The response to this command should resemble that in Figure 2-4.

```
CLASS      NAME
-----      ----
CSFSERV    ** (G)

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    WELLIE2         NONE              NONE     YES

INSTALLATION DATA
-----------------
NONE

APPLICATION DATA
----------------
NONE

SECLEVEL
--------
NO SECLEVEL

CATEGORIES
----------
NO CATEGORIES

SECLABEL
--------
NO SECLABEL

AUDITING
--------
FAILURES(READ)

NOTIFY
------
NO USER TO BE NOTIFIED

CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
 (DAY) (YEAR)       (DAY) (YEAR)      (DAY) (YEAR)
-------------  -------------------  ----------------
  089    98          089    98         089    98

ALTER COUNT   CONTROL COUNT   UPDATE COUNT   READ COUNT
-----------   -------------   ------------   ----------
NOT APPLICABLE FOR GENERIC PROFILE

USER      ACCESS
----      ------
WELLIE2   ALTER
```

Figure 2-4 Listing RACF profiles in the CSFSERV class

After this setup has been done, all users calling ICSF services that are not permitted to this generic profile trigger the RACF warning message (Figure 2-5).

```
ICH408I USER(MOP005  ) GROUP(SYS1    ) NAME(MOP USER
   CSFKGN CL(CSFSERV )
   WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
   FROM ** (G)
   ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

Figure 2-5 RACF warning message

Note that the name of the service that is called (CSFKGN) appears in the message with the user ID that called the service (MOP005). Because of the WARNING attribute in the resource profile, RACF temporarily grants access to the protected ICSF service.

Attention: It is very important that you ask your installation security staff whether running with RACF profiles in WARNING mode is acceptable to them.

Already defined discrete profiles in the CSFSERV class have precedence over the generic profile that we defined in the example.

Do not forget to suppress the WARNING attribute in the profile once you are finished by issuing the RALTER CSFSERV ** NOWARNING command.

The ICSF services that call CPACF for encryption and decryption of data (CSNBSYE and CSNBSYD) are not protected by RACF profiles and their invocation cannot be detected by this method.

2.2.2 Exploiting the tracing capability in z/OS System SSL

z/OS System SSL can provide information to confirm and assess the use of the System z hardware cryptographic facilities. System SSL also provides its own software implementation of the CCA cryptographic algorithms, and it might be unclear to the user which one of the two implementations is currently being used.

z/OS System SSL infrastructure

Figure 2-6 shows the z/OS System SSL infrastructure.

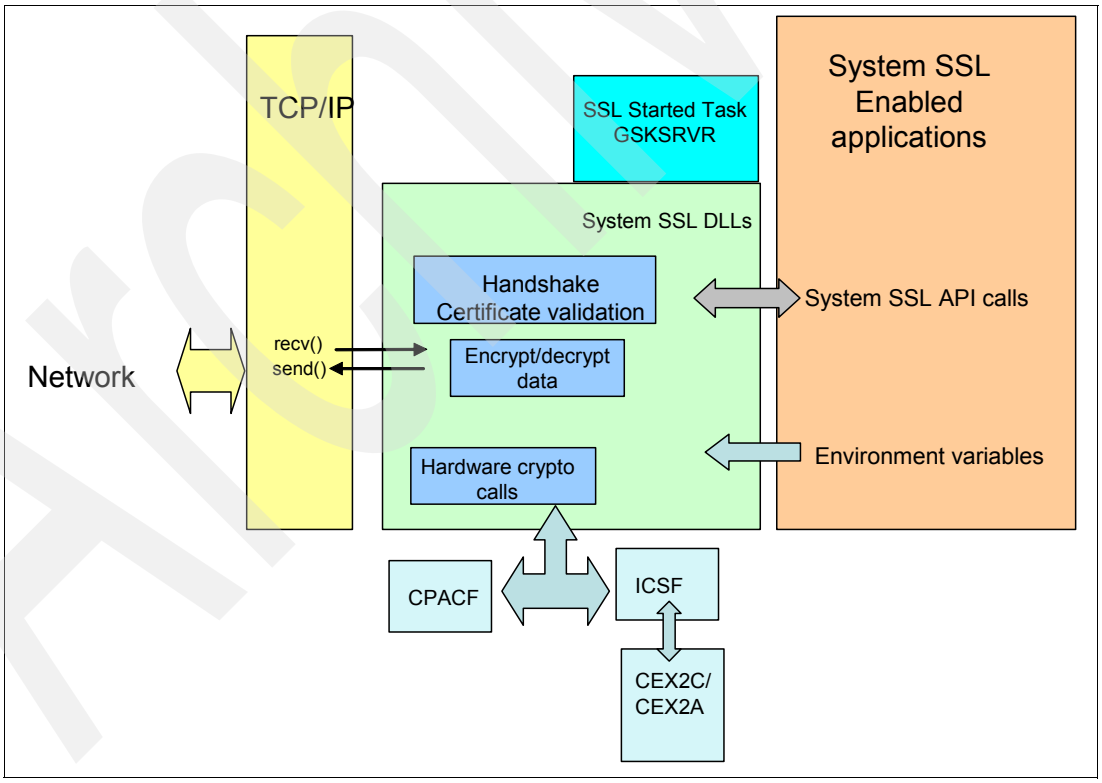


Figure 2-6 z/OS System SSL infrastructure

The System SSL-enabled application, such as the z/OS HTTP server, calls the System SSL high-level API to trigger the complex actions that are required by the protocol. Examples of these actions are conducting SSL handshakes and encrypting and decrypting that is data being exchanged with the partner application, and they are performed by the System SSL

software. System SSL requests, without visibility and whenever appropriate and available, services from the hardware cryptographic coprocessors or accelerators to ICSF or calls CPACF directly using the MSA instructions. System SSL also recognizes a set of environment variables that can be passed by the calling applications and that affect its behavior.

Finally, the *SSL Started Task* provides additional services that are not required to support the SSL/TLS communication, but instead help manage the System SSL runtime environment, including providing information about the cryptographic facilities or taking component traces. The activation of the SSL Started Task is an option left to the user. For more information about the z/OS System SSL environment variables and the SSL Started Task, see *z/OS Cryptographic Services System Secure Sockets Layer Programming*, SC24-5901.

Querying the available cryptographic services

The optional SSL Started Task element of system SSL can provide information about the cryptographic services that are identified as being offered by the system. The z/OS console command, MODIFY GSKSRVR,D CRYPTO, invokes the SSL Started Task for querying the cryptographic configuration and providing a status (Figure 2-7).

```
F GSKSRVR,D CRYPTO

GSK01009I Cryptographic status 464
Algorithm       Hardware    Software
DES                 56          56
3DES               168         168
AES                128         256
RC2                 --         128
RC4                 --         128
RSA Encrypt       2048        4096
RSA Sign          2048        4096
DSS                 --        1024
SHA-1              160         160
SHA-256            256         256
```

Figure 2-7 Display crypto by the SSL Started Task

The right column indicates which cryptographic algorithms System SSL can provide by software implementation. The center column shows the hardware cryptographic support in the system as it is sensed by the SSL Started Task and that System SSL is able to exploit. Note that this display does not constitute in itself evidence of hardware cryptography use, but it shows at least that hardware cryptographic services are available in the system.

Getting informative messages about the use of hardware cryptography

Among the set of environment variables that System SSL recognizes, the GSK_SSL_HW_DETECT_MESSAGE variable, when it has been exported with a value of “1” by the calling application, makes system SSL issue explicit messages about the hardware cryptography configuration that it recognizes during its initialization. These messages can be retrieved in the STDERR file (Figure 2-8 on page 20) of the application.

```
..... This is IBM HTTP Server V5R3M0
..... Built on Jan 19 2007 at 14:59:51

System SSL: SHA-1 crypto assist is available
System SSL: SHA-256 crypto assist is available
System SSL: DES crypto assist is available
System SSL: DES3 crypto assist is available
System SSL: AES 128-bit crypto assist is available
System SSL: AES 256-bit crypto assist is not available
System SSL: ICSF FMID is HCR7731
System SSL: PCI cryptographic accelerator is available
System SSL: PCIX cryptographic coprocessor is available
System SSL: Public key hardware support is available
```

Figure 2-8 System SSL reported cryptographic configuration

Note that System SSL has been called by the z/OS HTTP server, which can pass the GSK_SSL_HW_DETECT_MESSAGE environment variable value in one of two ways:

- By editing the HTTP Server started task procedure to add:

  `LEPARM='ENVAR("GSK_SSL_HW_DETECT_MESSAGE=1")'`

- By adding the line in the httpd.envvar file:

  `GSK_SSL_HW_DETECT_MESSAGE=1`

Switching from hardware to software cryptography

The decision as to whether to utilize hardware or software is made internally by System SSL. If a problem is encountered when using hardware cryptography, System SSL switches off hardware support for that particular function and falls back to its software implementation. Before z/OS V1R9, this was done so that it was not visible to the user and an analysis of System SSL traces was needed to detect this event.

As of z/OS V1R9, System SSL provides a notification, if the SSL Started Task is active, that such a switch occurred by displaying the GSK01052W message on the system console. One way to obtain this message is to stop ICSF while SSL is using the hardware cryptography (Figure 2-9). This new notification is intended to help you quickly identify conditions that led to a burst in CPU activity, such as switching to software cryptography.

```
GSK01051E HTTPSRV/00B7 Hardware encryption error. ICSF hardware
encryption processing is unavailable.
GSK01052W HTTPSRV/00B7 Hardware encryption error. PKE encryption
processing switched to software.
```

Figure 2-9 System SSL notification of fallback to software encryption

Tracing the System SSL calls to the hardware cryptographic devices

System SSL offers two tracing options that show calls to ICSF or direct invocation of the CPACF. These options are:

- A “simple” System SSL trace, which is controlled using the GSK_TRACE_FILE and GSK_TRACE environment variables
- A System SSL component trace, which requires activating the SSL Started Task prior to starting the trace

SSL Trace

Experience shows that the first “simple” trace usually provides enough information to pinpoint which hardware cryptographic services are actually being used. The trace excerpt in Figure 2-10 indicates which hardware cryptographic services and devices are sensed.

```
04/04/2007-12:14:35 Thd-0 INFO gsk_svc_init(): System SSL Version 3, Release 18, Service
level OA18836
04/04/2007-12:14:35 Thd-0 INFO gsk_svc_init(): LE runtime level 0x41080000, 31-bit
addressing mode
04/04/2007-12:14:35 Thd-0 INFO gsk_svc_init(): STDOUT handle=-1, STDERR handle=-1, TRACE
handle=4
04/04/2007-12:14:35 Thd-0 INFO gsk_dll_init_once(): Using variant character table for
code set IBM-1047
04/04/2007-12:14:35 Thd-0 INFO gsk_dll_init_once(): Using local code page IBM-1047
04/04/2007-12:14:35 Thd-0 INFO gsk_dll_init_once(): Using ISO8859-1 for TELETEX string
04/04/2007-12:14:35 Thd-0 INFO gsk_dll_init_once(): 64-bit encryption enabled
04/04/2007-12:14:35 Thd-0 INFO gsk_dll_init_once(): 128-bit encryption enabled
04/04/2007-12:14:35 Thd-0 INFO gsk_dll_init_once(): 168-bit encryption enabled
04/04/2007-12:14:35 Thd-0 INFO gsk_dll_init_once(): 256-bit encryption enabled
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): SHA-1 crypto assist is available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): SHA-256 crypto assist is available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): DES crypto assist is available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): DES3 crypto assist is available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): AES 128-bit crypto assist is available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): AES 256-bit crypto assist is not available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): ICSF FMID is HCR7731
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): PCI cryptographic accelerator is available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): PCIX cryptographic coprocessor is
available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): Public key hardware support is available
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): Maximum sign key size 2048, Maximum
management key size 2048
04/04/2007-12:14:35 Thd-0 INFO crypto_init(): Maximum RSA token size 2500
04/04/2007-12:14:35 Thd-0 INFO gsk_dll_init_once(): Job name HTTPSRV, Process 0100034D
......
04/04/2007-12:14:35 Thd-0 ENTRY gsk_open_keyring(): ---> Keyring webring
04/04/2007-12:14:35 Thd-0 INFO gsk_open_keyring(): Identifier 1 assigned to
'WebSphereCA'
04/04/2007-12:14:35 Thd-0 ENTRY gsk_decode_certificate(): --->
04/04/2007-12:14:35 Thd-0 ASCII gsk_decode_certificate(): Encoded X.509 certificate
        00000000:  30820289 308201f2 a0030201 02020100  *0...0...........*
        ......
```

Figure 2-10 System SSL trace (1/2)

Another piece of the trace (Figure 2-11 on page 22) then clearly indicates the use of CPACF (“*Clear key DES3 decryption performed*”) and the cryptographic coprocessor (“*Hardware RSA private key decryption performed*”). Likewise, software encryption is be clearly indicated if the hardware cryptographic services are not used by System SSL.

```
04/04/2007-12:14:35 Thd-0 EXIT gsk_generate_random_bytes(): <--- Exit status 0x00000000 (0)
04/04/2007-12:14:35 Thd-0 INFO crypto_des_encrypt(): Clear key DES encryption performed for 640
bytes
....

04/04/2007-12:15:23 Thd-16 INFO gsk_secure_socket_init(): SSL V2 cipher specs: 713624
04/04/2007-12:15:23 Thd-16 INFO gsk_secure_socket_init(): SSL V3 cipher specs: 0A0504090306
04/04/2007-12:15:23 Thd-16 INFO default_setsocketoptions(): TCP_NODELAY set for socket 12

04/04/2007-12:15:23 Thd-16 INFO crypto_des_decrypt(): Clear key DES decryption performed for 640
bytes
04/04/2007-12:15:23 Thd-16 EXIT gsk_get_record_by_label(): <--- Exit status 0x00000000 (0)
......
04/04/2007-12:15:23 Thd-16 INFO crypto_rsa_private_decrypt(): Using PKCS private key
04/04/2007-12:15:23 Thd-16 INFO crypto_rsa_private_decrypt(): RSA modulus is 1024 bits
04/04/2007-12:15:23 Thd-16 INFO crypto_rsa_private_decrypt(): Generating external CRT key token
04/04/2007-12:15:23 Thd-16 INFO crypto_rsa_private_decrypt(): Hardware RSA private key decryption
performed
.......
04/04/2007-12:15:23 Thd-16 INFO crypto_des3_decrypt_ctx(): Clear key DES3 decryption performed for
64 bytes

04/04/2007-12:15:23 Thd-16 INFO crypto_des3_encrypt_ctx(): Clear key DES3 encryption performed for
240 bytes
04/04/2007-12:15:23 Thd-16 INFO gsk_write_v3_record(): Calling write routine for 245 bytes
04/04/2007-12:15:23 Thd-16 INFO gsk_write_v3_record(): 245 bytes written
......
04/04/2007-12:15:23 Thd-16 INFO crypto_des3_decrypt_ctx(): Clear key DES3 decryption performed for
464 bytes
```

Figure 2-11 System SSL trace (2/2)

Attention: You can disable selected hardware cryptographic functions by setting the appropriate bits to zero in the GSK_HW_CRYPTO environment variable value, per the following bit position assignments:

- 1 = SHA-1 digest generation
- 2 = 56-bit DES encryption/decryption
- 4 = 168-bit Triple DES encryption/decryption
- 8 = Public key encryption/decryption
- 16 = AES 128-bit encryption/decryption
- 32 = SHA-256 digest generation

The corresponding software algorithms are used when a hardware function is disabled. This is reflected in the System SSL trace as: *Thd-0 INFO crypto_des_encrypt(): Clear key DES encryption performed for 640 bytes.*

This becomes, when DES has been disabled using GSK_HW_CRYPTO: *Thd-0 INFO crypto_des_encrypt(): **Software** DES encryption performed for 640 bytes*.

System SSL component trace

The System SSL Started Task provides component trace support for any SSL thread that has been started by applications running in the same system as the GSKSRVR. The trace records can be in a trace external writer or they can be kept in an in-storage trace buffer that is part of the GSKSRVR address space. IPCS is used to format and display the trace records from either a trace data set or an SVC dump of the GSKSRVR address space.

The System SSL component trace produces outputs such as that in Figure 2-12.

```
SC60      MESSAGE   00000002  15:36:19.236028  SSL_EXIT

  Job HTTPSRV   Process 02000CA0  Thread 00000000  crypto_generate_random_bytes
  Exit status 00000000 (0)

SC60      MESSAGE   00000008  15:36:19.236104  SSL_INFO

  Job HTTPSRV   Process 02000CA0  Thread 00000000  crypto_des_encrypt
  Clear key DES encryption performed for 640 bytes

SC60      MESSAGE   00000008  15:36:19.236373  SSL_INFO

  Job HTTPSRV   Process 02000CA0  Thread 00000000  gsk_open_keyring
  Identifier 3 assigned to 'pssc pki ca'

SC60      MESSAGE   00000001  15:36:19.236476  SSL_ENTRY

  Job HTTPSRV   Process 02000CA0  Thread 00000000  gsk_decode_certificate
```

Figure 2-12 System SSL component trace

Measuring hardware cryptography activity with z/OS System SSL

z/OS System SSL does not provide measurement data per se regarding the use of the System z hardware cryptography. Instead, the user has to rely on additional performance reporting products, such as IBM RMF or IBM Tivoli Omegamon XE, for measurement data. Figure 2-13 is a high level description of the measurement data collection infrastructure as built around RMF.

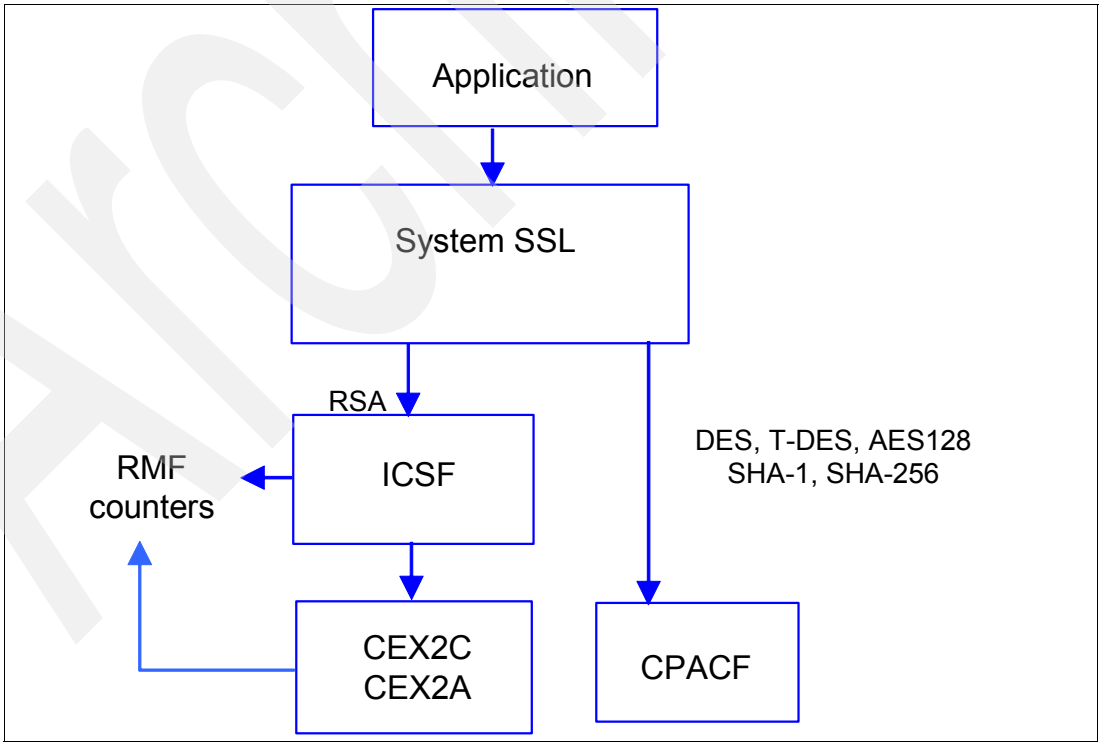


Figure 2-13 Collection of hardware cryptography measurement data with System SSL

The way RMF operates for data collection and how it is used is further developed in “Using RMF to measure z/OS hardware cryptography activity” on page 34.

Important: There is no z/OS-based measurement capability available today that can provide utilization data for CPACF, if it is not built in the application itself. Because this facility is actually a part of the system PU, the additional machine cycles created are simply reported as CPU utilization by RMF or equivalent products. As an example, the only activity reported as hardware cryptography activity in Figure 2-13 on page 23 is the RSA functions performed by the CEX2C in coprocessor or accelerator mode.

2.2.3 The ICSF component trace

ICSF offers a system component trace capability, which is described in *z/OS Cryptographic Services Integrated Cryptographic Service Facility System Programmer’s Guide*, SA22-7520.

Among other data, IPCS can extract counters from the ICSF Component trace indicating what ICSF services have been called, how many times during tracing, and how many failed (Figure 2-14).

```
COMPONENT TRACE SHORT FORMAT
 COMP(CSF)
 OPTIONS((COUNTS))
ICSF COUNTS FROM CTRACE:
SERVICE CALLS_FOUND = 00000185
FAILING SERVICES    = 00000055
SERVICE   #SUCCESS   #FAILED
CSFS1TRL  00000017   00000055
CSFN1GKP  00000007   00000000
CSFS1GAV  00000028   00000000
CSFS1TRD  00000014   00000000
CSFNPKI   00000005   00000000
CSFNDSG   00000005   00000000
CSFNDSV   00000005   00000000
CSFNPKD   00000042   00000000
CSFNIQF   00000007   00000000
```

Figure 2-14 ICSF component trace

2.3 Assessing the use of hardware cryptography on z/VSE

z/VSE does not offer a performance reporting facility such as the z/OS RMF; however, it can provide statistical information about the use of hardware cryptography with the VSE Security Server STATUS=CR command. Figure 2-15 on page 25 shows an example of the command output. It contains information about the total number of AP requests executed, the type of the cryptographic coprocessor actually available, and the status, and which algorithms are supported by the CPACF in the system.

```
MSG FB,DATA=STATUS=CR
AR 0015 1I40I  READY
FB 0011 BST223I CURRENT STATUS OF THE SECURITY TRANSACTION SERVER:
FB 0011 ADJUNCT PROCESSOR CRYPTO SUBTASK STATUS:
FB 0011   AP CRYPTO SUBTASK STARTED ..... : YES
FB 0011   MAX REQUEST QUEUE SIZE ........... : 1
FB 0011   MAX PENDING QUEUE SIZE ........... : 1
FB 0011   TOTAL NO. OF AP REQUESTS ......... : 16
FB 0011   NO. OF POSTED CALLERS ............ : 16
FB 0011   AP CRYPTO POLLING TIME (1/300 SEC).. : 1
FB 0011   AP CRYPTO WAIT ON BUSY (1/300 SEC).. : 75
FB 0011   AP CRYPTO RETRY COUNT ............ : 5
FB 0011   AP CRYPTO TRACE LEVEL ............ : 3
FB 0011   TOTAL NO. OF WAITS ON BUSY ....... : 0
FB 0011   CURRENT REQUEST QUEUE SIZE ....... : 0
FB 0011   CURRENT PENDING QUEUE SIZE ....... : 0
FB 0011   ASSIGNED APS : PCICC / PCICA ..... : 0 / 0
FB 0011                 CEX2C / CEX2A ..... : 2 / 0
FB 0011                 PCIXCC ............ : 0
FB 0011     AP 0 : CEX2C   - ONLINE
FB 0011     AP 1 : CEX2C   - ONLINE
FB 0011   ASSIGNED AP QUEUE (CRYPTO DOMAIN)... : 4
FB 0011 CPU CRYPTOGRAPHIC ASSIST FEATURE:
FB 0011   CPACF AVAILABLE .................. : YES
FB 0011   INSTALLED CPACF FUNCTIONS:
FB 0011     DES, TDES-128, TDES-192
FB 0011     AES-128
FB 0011     SHA-1, SHA-256
FB 0011 END OF CPACF STATUS
```

Figure 2-15 The z/VSE Security Server STATUS command

2.4 Assessing the use of hardware cryptography on Linux for System z

There are two ways of establishing evidence of a hardware cryptography exploitation by applications that execute in Linux for System z. One way is to analyze the status of the z90crypt device driver that executes inside the Linux for System z instance. This is what we explain in this section.

You can also exploit the RMF reporting in a z/OS logical partition that is located in the same physical machine as the Linux for System z logical partition. This is demonstrated in 3.4.3, “Crypto Hardware Activity report without local activity” on page 40.

Important: There is no facility available in Linux to track and assess the use of CPACF. Such tracking has to be provided by the exploiting application.

2.4.1 Status of the z90crypt device driver

The z90crypt device driver routes the application cryptographic workload to the supported cryptographic hardware devices installed in the machine.

As a first step, ensure that the device driver is installed by issuing the **Ismod** command, which displays the list of the module loaded in the kernel. In the example in Figure 2-16, the z90crypt device driver is present.

```
vpnsrv:~ # lsmod
Module                  Size  Used by
sha1_z990              20224  0
des_z990               21760  0
des_check_key          18944  1 des_z990
sg                     68936  0
st                     68920  0
sd_mod                 43272  0
sr_mod                 39980  0
scsi_mod              206712  4 sg,st,sd_mod,sr_mod
cdrom                  65320  1 sr_mod
ipv6                  426664  136
tun                    29184  1
qeth                  243904  0
qdio                   75088  3 qeth
ccwgroup               27648  1 qeth
z90crypt               77992  2
dm_mod                100120  0
dasd_eckd_mod          89344  4
dasd_mod              103528  5 dasd_eckd_mod
ext3                  184512  1
jbd                   118856  1 ext3
```

Figure 2-16 lsmod command output

For more information about Linux for System z cryptography, see "*Using Cryptographic Adapters for Web Servers with Linux on IBM System z9 and zSeries,*" REDP-4131.

2.4.2 Collecting information about hardware cryptography activity

The **cat /proc/driver/z90crypt** command collects information about the device driver status. Figure 2-17 on page 27 shows the status for a Linux system executing in a logical partition, or in a VM guest with access to two running PCICA coprocessors on a z990..

```
vpnsrv:~ # cat /proc/driver/z90crypt

z90crypt version: 1.3.3
Cryptographic domain: 2
Total device count: 2
PCICA count: 2
PCICC count: 0
PCIXCC MCL2 count: 0
PCIXCC MCL3 count: 0
CEX2C count: 0
CEX2A count: 0
requestq count: 0
pendingq count: 0
Total open handles: 2


Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
           1100000000000000 0000000000000000 0000000000000000 0000000000000000


Waiting work element counts
           0000000000000000 0000000000000000 0000000000000000 0000000000000000


Per-device successfully completed request counts
    0000040E 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Figure 2-17 z90crypt device driver status

The information that is displayed is composed of four parts:

- General information
- Online devices
- Waiting work elements counts
- Per-device count of successfully completed requests

In the *general information* section, the total device count indicates how many cryptographic hardware devices are detected to be online to Linux and therefore potentially available for applications that are requesting RSA cryptographic services. The general information also includes other information like the device driver version number and the cryptographic domain that the Linux applications have access to.

The *online devices section* reports the physical arrangement of the detected cryptographic coprocessors in the system. Each hexadecimal digit position is the AP number in the system. There is a maximum of 16 cryptographic devices, or APs, that can be detected from 64 possible AP positions. When the digit has a value of zero, there is no device detected for the AP number. Otherwise, the digit value indicates which type of cryptographic device is occupying the AP position as follows:

- 1 = PCICA
- 2 = PCICC
- 3 = PCIXCC(Machine Level Code 2)
- 4 = PCIXCC(Machine Level Code 3)
- 5 = CEX2C
- 6 = CEX2A

Based on the information in Figure 2-17 on page 27, we concluded that this Linux system instance has detected two PCICA online devices as AP0 and 1.

The *waiting work element counts section* reports the number of outstanding units of work for each detected AP at the time of execution of the cat command.

The *per-device successfully completed request counts section* reports the successful hardware cryptography activity for each AP that has been detected online. Each device has an eight hexadecimal digit count which represents the cumulative count of work units that were successfully performed by the device. A count equal to “00000000” can indicate either that:

- There is simply no invocation of the hardware cryptographic coprocessor because no application is requesting cryptographic services, or the applications are requesting services that are not provided by the coprocessor or accelerator.
- All requests to the AP have been failing.

In the example in Figure 2-17 on page 27, the first online device has successfully performed hexadecimal 40E units of work. As the count for the second online device is zero, then we can conclude that there is no activity or all requests have failed for AP1.

2.4.3 Programs that invoked hardware cryptography

The programs that invoked hardware cryptography can be retrieved with the **lsof** command., which lists open files and which program opened them. Figure 2-18 shows which programs opened the z90crypt device driver.

```
vpnsrv:~ # lsof |grep z90
openvpn     581 nobody    6u   CHR             10,62            285122 /dev/z90crypt
sshd      12737   root    3u   CHR             10,62            285122 /dev/z90crypt
sshd      12740 guigui    3u   CHR             10,62            285122 /dev/z90crypt
```

Figure 2-18 Programs that opened the z90crypt device driver

2.5 Setting up the hardware cryptography configuration of z/VM

In z/VM, real cryptographic coprocessors are assigned to guest virtual machines by specifying the CRYPTO parameter in the VM guest machine directory with the DOMAIN and APDEDICATED or APVIRTUAL operands. Refer to *z/VM CP Planning and Administration*, SC24-6083, for further explanation of these parameters.

APDEDICATED specifies the numbers of APs that the virtual machine can use for dedicated access to the AP cryptographic facility. The DOMAIN operand also must be specified to indicate the coprocessor domain to access in the APs. The AP queues that are equivalent to the domains that are specified by all of the DOMAIN operands are then assigned to the guest for each AP specified on the APDED operand on all CRYPTO statements. The APs must be selected from the set in the PCI Cryptographic Online List on the Crypto Image Profile Page for the z/VM logical partition, or from the candidate list of APs that have been brought online to the z/VM logical partition. The DOMAINs must be part of the set selected in the Usage Domain Index list in the Crypto Image Profile Page for the Logical Partition.

APVIRTUAL tells the z/VM Control Program that this virtual machine can share access to the clear-key functions of only the AP cryptographic facility with other virtual machines. z/VM drives the requests to whatever coprocessor is available. The DOMAIN operand is not necessary when specifying the APVIRT operand because z/VM rejects all requests for services that would involve the use of secure keys.

Figure 2-19 is an example of a VM guest machine setup. The z/VM system logical partition has access to APs 5 and 6 in coprocessor mode and AP 2 in accelerator mode. The directory of the z/OS VM guest machine has been set up to specify a dedicated access to domain 1 of AP 5 and 6, and the Linux VM guest machines are sharing access to whatever accelerator is available to the z/VM system. There is no setup required to share the CPACF because the facility is available to whichever guest program is dispatched on the PU.

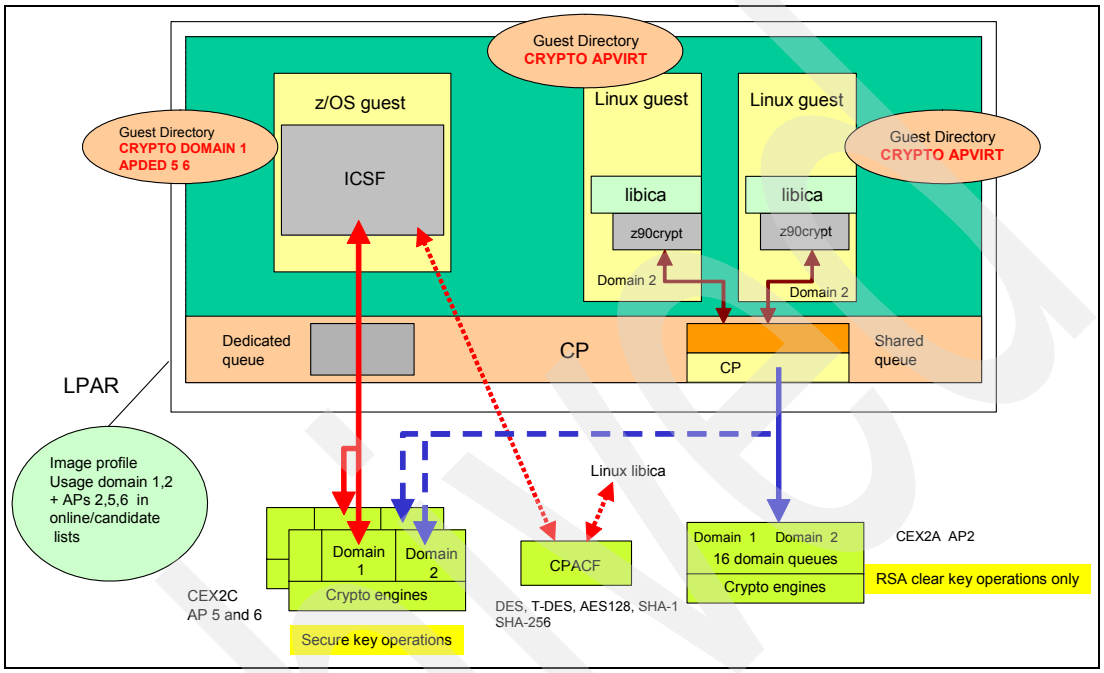


Figure 2-19 Hardware cryptographic coprocessors assignment to VM guest machines

2.5.1 Checking the hardware cryptography configuration with z/VM

There is no z/VM facility to monitor hardware cryptography activity; however, you can obtain configuration information that is related to the physical or virtual hardware cryptographic devices. For example, the CEX2C status appears as an output of the z/VM CP QUERY CRYPTO command that can be used to display the z/VM crypto system configuration. Again, this does not provide any cryptographic activity information but it at least shows which VM guests are entitled to use hardware cryptographic coprocessors.

Physical hardware cryptographic devices

The CP QUERY CRYPTO command can be used to query for the cryptographic coprocessors physical configuration. For example, in Figure 2-20, one PCICA accelerator feature has been detected.

.

```
q crypto
Crypto Adjunct Processor Instructions are installed

q crypto ap
AP 00 PCICA Queue 11 is installed
AP 01 PCICA Queue 11 is installed
AP 02 PCIXCC Queue 11 is dedicated to LNXSU2
```

Figure 2-20 z/VM query crypto command

This example of a QUERY CRYPTO command output shows that the z/VM system has access to three cryptographic coprocessors (for this example, we issued the command on a z990 system, with PCICA and PCIXCC features installed). Each coprocessor has domain 11 assigned to handle the requests coming from the logical partition where the z/VM instance resides.

Virtual hardware cryptographic devices

The QUERY VIRTUAL CRYPTO command can be used to display what virtual coprocessors are actually available to a guest machine (Figure 2-21).

```
q v crypto
No CAM or DAC Crypto Facilities defined
AP 06 PCICA Queue 00 dedicated
AP 06 PCICA Queue 01 dedicated
AP 06 PCICA Queue 10 dedicated
AP 12 PCIXCC Queue 00 dedicated
AP 12 PCIXCC Queue 01 dedicated
AP 12 PCIXCC Queue 10 dedicated
```

Figure 2-21 The QUERY VIRTUAL CRYPTO command

3

Measuring the hardware cryptography activity on z/OS with RMF

This chapter describes how you can use RMF to measure the hardware cryptography activity in a z/OS system. It addresses the way the product operates, shows the reporting data that can be issued, and indicates how that data can be interpreted.

As an introduction, we also explain how the ICSF workload is balanced and review which SMF records are used to report hardware cryptography activity.

3.1 A brief overview of ICSF cryptographic workload balancing

Note: ICSF performs the workload balancing of the cryptographic service requests. z/OS Workload Manager (WLM) does not perform any direct load balancing or prioritization of requests arriving at ICSF.

When operating in a logical partition, an ICSF instance is given access to a unique domain that belongs to as many coprocessors specified as online to the logical partition.

For each coprocessor (actually, the domain in the coprocessor) that ICSF has access to, ICSF can drive an architecturally-fixed maximum number of N requests to be executed concurrently. ICSF maintains a structure in storage for each coprocessor that keeps track of these N requests. If more requests come in to ICSF than can be driven to the collection of coprocessors (that is, would exceed the N number for each accessible coprocessor), ICSF queues that work behind one of these coprocessors based on a workload balancing scheme.

Note: N today equals 8. That is, the structure has 8 entries. It has not always been this way, and the value of N might again change in the future to adapt to new cryptographic coprocessor technologies. The authors refer to either “N” or “8”, depending on the relevance or the context.

The coprocessor workload balancing scheme is essentially as follows: When a new request comes into ICSF, ICSF starts scanning the coprocessor structures, beginning with the lowest numbered coprocessor (that is, the AP number) to find the processor that is least used. If it can drive this work out to the coprocessor, because there are less than N requests for it, it does so; if not, it queues the work to be executed on that coprocessor later. Note that, after it is queued for one coprocessor, the request does not move to another coprocessor even if it becomes available in the meanwhile.

When the number of requests on a coprocessor drops below the architectural limit of N, queued requests to this coprocessor are dequeued and sent to the coprocessor. The consequence of this design is that the workload is usually not evenly distributed between available coprocessors. For example, if the workload is very light, almost all work is directed only to the lowest numbered coprocessor because its structure is empty when new work comes into ICSF.

3.2 SMF reporting of hardware cryptography activity

In this section, we describe the SMF records that are used to report hardware cryptography activity in z/OS. To learn more about these SMF records and how to collect them, see *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

3.2.1 Type 82 (ICSF record)

This SMF record mainly reports ICSF administrative and environmental events, except for subtypes 17, 19, and 20, which are intended to provide performance data. Table 3-1 on page 33 summarizes the record subtypes.

Table 3-1 Type 82 subtypes

Name	Description
Subtype 1	Written whenever ICSF is started
Subtype 3	Written whenever there is a change in the number of available processors with the cryptographic feature
Subtype 4	Written whenever ICSF handles error conditions for cryptographic feature failure (CC3, Reason Code 1) or cryptographic tampering (CC3 Reason Code 3)
Subtype 5	Written whenever a change to a special security mode is detected
Subtype 6 and 7	Written whenever a key part is entered with the key entry unit (KEU)
Subtype 8	Written whenever the in-storage copy of the CKDS is refreshed
Subtype 9	Written whenever the CKDS is updated by a dynamic CKDS update service
Subtype 10	Written when a clear key part is entered for one of the PKA master keys
Subtype 11	Written when a clear key part is entered for the DES master key
Subtype 12	Written for each request and reply from calls to the CSFSPKSC service by TKE
Subtype 13	Written whenever the PKDS is updated by a dynamic PKDS update service
Subtype 14	Written when a clear key part is entered for any of the PCICC master keys
Subtype 15	Written whenever a PCICC-retained key is created or deleted
Subtype 16	Written for each request and reply from calls to the CSFPCI service by TKE
Subtype 17	Written periodically to provide some indication of PCICC usage
Subtype 18	Written when a PCICC, PCIC, PCIXCC, CEX2C, or CEX2A comes online or offline
Subtype 19	Written when a PCIXCC operation begins or ends
Subtype 20	Written by ICSF to record processing times for PCIXCCs and CEX2Cs
Subtype 21	Written when ICSF issues IXCJOIN to join the ICSF sysplex group or issues IXCLEAVE to leave the sysplex group
Subtype 22	Written when the Trusted Block Create Callable services are invoked

Note: A sample job is supplied in z/OS to format the ICSF Type 82 SMF records, which gives a report of ICSF activity. The job is available in SYS1.SAMPLIB(CSFSMFJ), and invokes a REXX exec SYS1.SAMPLIB(CSFSMFR). *IBM eServer zSeries 990 (z990) Cryptography Implementation*, SG24-7070, provides examples of the use of these utilities.

3.2.2 Type 70 - Subtype 2 (RMF Processor Activity)

This record contains measurement data for cryptographic coprocessors and accelerators located in the following sections:

- The Cryptographic Coprocessor Data section, which contains measurement data for cryptographic coprocessors as provided by their CEX2C features
- The Cryptographic Accelerator Data section, which contains measurement data for cryptographic accelerators as provided by their CEX2C features

- The ICSF Services Data section, which contains measurement data for the set of ICSF services to be reported by RMF

This record is written for each measurement interval and when the session terminates.

3.2.3 Type 30 (Common Address Space Work)

This record contains the SMF30CSC field, which is described as the Integrated Cryptographic Service Facility/MVS (ICSF/MVS) service count. This is the number of cryptographic instructions that are executed on behalf of the caller: each time ICSF issues a command to a hardware coprocessor, this count is incremented by one. However, this means that in some cases, such as the bulk encryption of data, the count would be incremented several times, because ICSF might loop on commands that are being issued to the coprocessor even though they are performing a single service call at the ICSF API level. For other operations, like a PIN verification, the count would be incremented by one for a single service call to ICSF. There is, therefore, no guaranteed correlation between the “number of cryptographic instructions” and the actual number of service calls to ICSF.

3.2.4 Type 72 - Subtype 3 (Workload Activity)

Two fields are used to track, by sampling, the status of tasks that are waiting to get access to an available hardware cryptographic coprocessor (or AP):

- R723APUAP, which is the count of “crypto using” samples (that is, a TCB was found executing on a cryptographic coprocessor)
- R723APD, which is the count of “AP crypto delay samples” (that is, a TCB was found waiting for a cryptographic coprocessor)

These counters directly reflect, on a sampling basis, the number of the requests being queued by ICSF, because they would exceed the N limit and the number of requests that are currently being executed by the coprocessor. These counters are further discussed in 3.4.4, “The Workload Activity report” on page 40.

3.3 Using RMF to measure z/OS hardware cryptography activity

RMF is the IBM strategic product for z/OS performance measurement and management. It collects performance data for z/OS base and sysplex environments and issues reports that can be used to monitor system performance so that users can optimally tune and configure their systems to meet business needs.

The hardware cryptographic coprocessors are among the resources for which RMF can provide activity and performance data. 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.

Note: The RMF Spreadsheet Reporter Tool V5.2.3 for Windows® is a tool that complements RMF, and it is available for download from the IBM Web site. However, as of this writing, it does not support cryptographic hardware feature performance analysis.

3.3.1 RMF data collection infrastructure for hardware cryptography

Figure 3-1 shows the RMF data collection infrastructure. Hardware counters that count the requests arriving at the CEX2C or CEX2A regardless of their domains and the time it takes to complete these requests are maintained. RMF uses control blocks to collect these values.

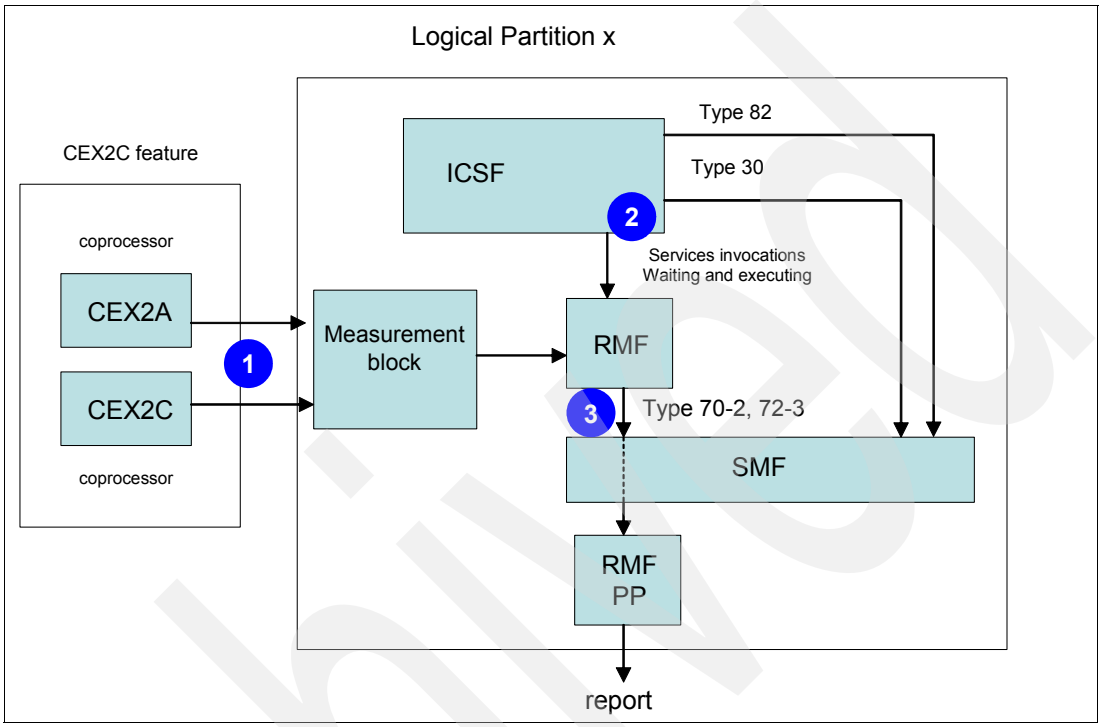


Figure 3-1 RMF data collection infrastructure

Important: The reports identify whether the requests are going to a CEX2C or a CEX2A and the length of the RSA key. No differentiation is made between the domains involved, however, so RMF can collect data about hardware cryptography activities that are initiated by other logical partitions in the system.

ICSF maintains its own software counters, pertaining to a set of services intentionally designed for RMF reports, and provides the data to RMF. ICSF reports activities for:

- Encipher and decipher (that is DES, double-DES or triple-DES encryption and decryption) performed with secure keys in CEX2C
- MAC generate and MAC verify (still performed by CEX2C)
- One Way Hash SHA-1 and SHA-256, either performed by CEX2C (SHA-1) or the CPACF (SHA-1 or SHA-256)
- PIN translate and PIN verify performed by the CEX2C

Important: SHA-1 and SHA-256 are the *only* CPACF activities that are explicitly reported by RMF. However, they must be invoked with the CSNBOWH service because they are not reported if they are directly invoked by the application using the MSA instructions. No other CPACF activities are reported, even those invoked by the CSNBYE or CSNBYD services.

RMF, in turn, makes a request to SMF to wrap the collected data into type 70 and 72 records.

3.4 The RMF post-processor reports

The RMF post-processor processes the activity data that the RFM data gatherer collects. For hardware cryptography activity, RMF Monitor 1 gathers the data based on the specification in the ERBRMFxx member of SYS1.PARMLIB. By default, hardware cryptography activity gathering is enabled unless the NOCRYPTO keyword is specified. To generate reports about hardware cryptography activity (the Crypto Hardware Activity or the Workload Activity report), the RMF post-processor program operates based on the option (CRYPTO or NOCRYPTO) specified in the REPORTS control statement, which is provided as input in the post-processor JCL. Figure 3-2 shows the control statements for a Crypto Hardware Activity report and a Workload Activity report.

```
//SYSIN    DD   *
  SUMMARY(INT,TOT)
  DINTV(0001)
  RTOD(0900,1000)
  STOD(0900,1000)
  REPORTS(CRYPTO)
  SYSRPTS(WLMGL(SCPER))
  SYSOUT(A)
```

Figure 3-2 Sample RMF Postprocessor SYSIN DD Statement

3.4.1 The Crypto Hardware Activity RMF Report

The RMF post-processor report data is arranged in three sections (Figure 3-3).

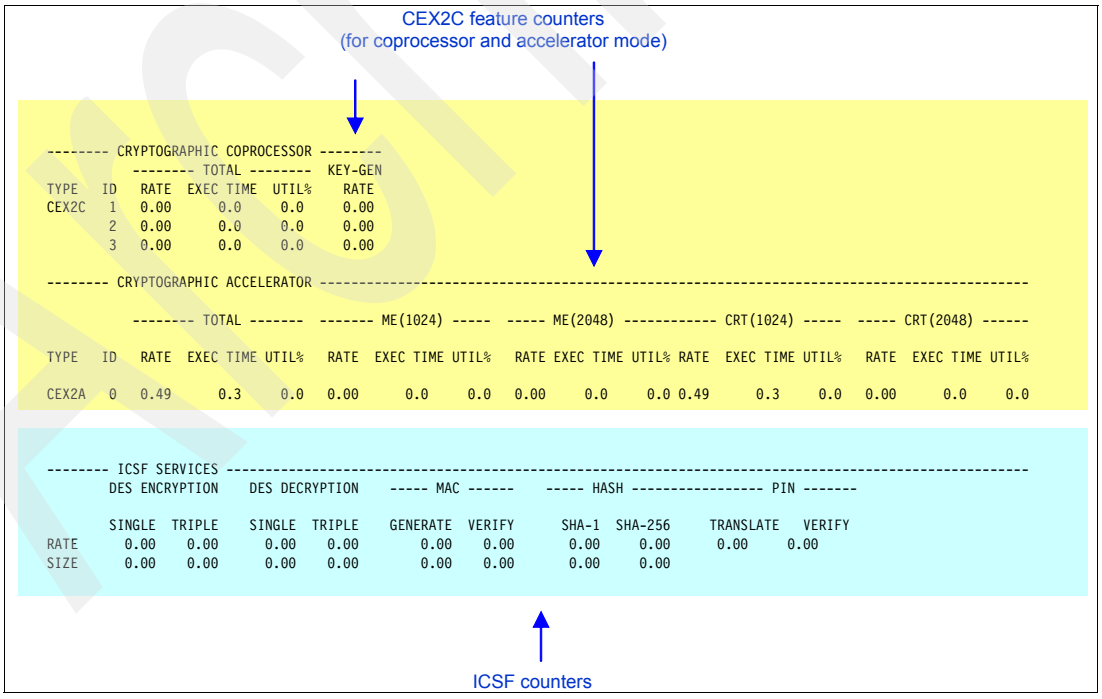


Figure 3-3 Structure of the Crypto Hardware Activity report

The cryptographic coprocessor section

The recorded activity relates to secure key symmetric cryptographic functions, clear and secure key asymmetric operations, and the calls to the user defined extension (UDX) functions, if any, installed in the coprocessor. Note that the RSA key-generation is given special attention because it consumes a significant amount of computing capacity.

The fields in the cryptographic coprocessor section are:

- TYPE. This fields specifies the type of cryptographic coprocessor, which must be CEX2C for System z9.
- ID. This field specifies the index number of the coprocessor (also known as the AP number).
- TOTAL RATE. This field indicates the global arrival rate, that is, the average number of requests per second, for all operations that were executed by the cryptographic coprocessor in the RMF time interval.
- TOTAL EXEC TIME. This field indicates the global average execution time in milliseconds of elapsed time for all operations performed in this cryptographic coprocessor.
- TOTAL UTIL%. This field gives the total utilization percentage of this processor, that is, the hundredths of seconds of time spent per elapsed second by the coprocessor executing requests, on average, in this interval.

  Note: The following formula calculates the coprocessor utilization percentage and the result is always produced with only one single digit after the decimal point:

  ```
  TOTAL UTIL% = TOTAL RATE x TOTAL EXEC TIME /10
  ```

- KEY-GEN RATE. This field gives the occurrence rate of RSA key generation operations.

The cryptographic accelerator section

CEX2A performs the public key cryptography operations that are used with the SSL or TLS protocols. These protocols are widely used to help secure e-business applications. The activity data for the cryptographic accelerators provides details about the RSA key format used: ME and CRT. They also indicate the key lengths (1024 and 2048 bit) used with each format. For considerations for both formats, see 5.4.5, “Cryptographic accelerator key length and format” on page 61.

The fields in this section are:

- TYPE. This field specifies the type of cryptographic accelerator, which should be CEX2A for System z9.
- ID. This field specifies the index number, or AP number, of the cryptographic accelerator.
- TOTAL RATE. This field gives the global average arrival rate, that is, the number of requests received per second, for all the operations performed by this cryptographic accelerator during the RMF time interval.
- ME (1024) RATE, EXEC TIME and UTIL%. These fields indicate the total rate, average execution time in milliseconds of elapsed time, and utilization percentage of the cryptographic accelerator for all RSA operations with 1024-bit keys, or less, in the ME format.
- ME (2048) RATE, EXEC TIME and UTIL%. These fields indicate the total rate, average execution time in milliseconds of elapsed time, and utilization percentage of the cryptographic accelerator for all RSA operations with 2048-bit keys in the ME format.

- CRT(1024) RATE, EXEC TIME and UTIL%. These fields give the total rate, average execution time in milliseconds of elapsed time, and utilization percentage of the cryptographic accelerator for all RSA operations with 1024-bit keys in the CRT format.
- CRT(2048) RATE, EXEC TIME and UTIL%. These fields give the total rate, average execution time in milliseconds of elapsed time and utilization percentage of the cryptographic accelerator for all RSA operations with 2048-bit keys in the CRT format.

Note: The following formula applies and the result is always produced with only one single digit after the decimal point:

```
TOTAL UTIL% = TOTAL RATE x TOTAL EXEC TIME /10
```

ICSF services

This is the section where the activities of the pre-determined set of ICSF services are displayed.

Important: The RMF is not reporting any activity data for the CPACF other than what is embedded in the CPU time, except, implicitly, when the CPACF is invoked by the CSNBOWH ICSF callable service for SHA-1 or SHA-256 computations. Even in that case the CPACF utilization cycles are reported in CPU time.

The fields in this section are:

- DES ENCRYPTION, SINGLE and TRIPLE, RATE. These fields indicate the rate of requests to the CEX2C to encrypt data with the single-DES or triple-DES algorithm. These are secure key operations only.
- DES ENCRYPTION, SINGLE and TRIPLE, SIZE. These fields indicate the average number of bytes per request that have been encrypted, using secure keys, with the single-DES or triple-DES algorithm.
- DES DECRYPTION, SINGLE and TRIPLE, RATE. These fields indicate the rate of requests to the CEX2C to decrypt data with the single-DES or triple-DES algorithm. These are secure key operations only.
- DES DECRYPTION, SINGLE and TRIPLE, SIZE. These fields indicate the average number of bytes per request that have been decrypted, using secure keys, with the single-DES or triple-DES algorithm.
- MAC GENERATE and VERIFY RATE. These fields indicate the rate of requests to respectively generate or verify Message Authentication Codes.
- MAC GENERATE and VERIFY SIZE. These fields indicate the average number of bytes per request for which MACs have been respectively generated or verified.
- HASH SHA-1 and SHA-256 RATE. These fields indicate the rate of requests to hash data, using the SHA-1 or SHA-256 hash algorithms.
- HASH SHA-1 and SHA-256 SIZE. These fields indicate the average number of bytes that were hashed per request using the SHA-1 or SHA-256 hash algorithms.
- PIN TRANSLATE and VERIFY RATE. These fields indicate the rate of requests to translate or verify PINs.

3.4.2 An example of Crypto Hardware Activity report

Figure 3-4 shows an example of a Crypto Hardware Activity report.

```
                                C R Y P T O  H A R D W A R E  A C T I V I T Y
                                                                                                                PAGE    1
           z/OS V1R8               SYSTEM ID SC60            START 04/03/2007-12.42.00  INTERVAL 000.01.00
                                   RPT VERSION V1R8 RMF      END   04/03/2007-12.43.00  CYCLE 1.000 SECONDS
 -------- CRYPTOGRAPHIC COPROCESSOR --------
           -------- TOTAL --------  KEY-GEN
 TYPE   ID   RATE  EXEC TIME  UTIL%    RATE
OCEX2C   1 763.0      1.3    99.9     0.70

 -------- CRYPTOGRAPHIC ACCELERATOR ----------------------------------------------------------------------------------------------------
           -------- TOTAL -------  ------- ME(1024) -----  ----- ME(2048) ------  ------ CRT(1024) -----  ----- CRT(2048) ------
 TYPE   ID   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%
 CEX2A   0  575.9      0.3    18.2  575.9      0.3    18.2   0.00     0.0     0.0   0.00      0.0     0.0   0.00      0.0     0.0
-
 -------- ICSF SERVICES ----------------------------------------------------------------------------
         DES ENCRYPTION    DES DECRYPTION    ----- MAC ------    ----- HASH -----    ------ PIN -------
         SINGLE  TRIPLE    SINGLE  TRIPLE    GENERATE  VERIFY      SHA-1  SHA-256    TRANSLATE   VERIFY
 RATE     70.90   63.45     70.92   63.43       72.85   72.85      16263    15415         0.00     0.00
 SIZE     32.00   32.00     32.00   32.00       32.00   32.00       1257     1256
```

Figure 3-4 An example of Crypto Hardware Activity report.

This report shows a configuration of two CEX2C *cards*, one card in coprocessor mode (CEX2C) with index number 1 and the other one in accelerator mode (CEX2A) with index number 0. The RMF time interval is 1 minute.

The CEX2C in coprocessor mode has been utilized at an average of 99.9% during the time interval, having received, on average, 763 requests per second, each of them consuming on average 1.3 msec of elapsed time. Among these requests, RSA key pair generation requests were received at a rate of 0.70 requests per second, or 42 RSA key generation requests during the interval. Remember that this data pertains to the coprocessor utilization of all logical partitions that are active during the collection of hardware cryptography activity data.

The CEX2A has been used for 18.2% of the interval to serve, on average, 575.9 requests per second, each request taking an average of 0.3 msec of elapsed time. The CEX2A can perform the following clear key services:

- Digital signature verify
- PKA decrypt
- PKA encrypt (ZERO-PAD and MRP only)

It is not possible to tell, from the report, how the CEX2A utilization breaks down into each one of these services; however, we can see that all operations were involving 1024-bit RSA keys in the ME format. Remember that this data pertains to the accelerator utilization of all the logical partitions active during the RMF interval.

The ICSF services section of the report shows a CEX2C (in coprocessor mode) activity with DES and Triple-DES encryption and decryption, each request processing, on average, 32 bytes, and MAC Generate and Verify with an average length of processed data of 32 bytes.

The report also shows SHA-1 and SHA-256 activity.

Note: The RATE column shows a value of “<0,01” when the crypto activity in the RMF interval is too low to yield a significant measure (see Figure 3-5 on page 40).

3.4.3 Crypto Hardware Activity report without local activity

Figure 3-5 illustrates a case where the CEX2C feature reported an activity that was recorded by RMF but no requests are shown for the local ICSF.

```
            z/OS V1R7               SYSTEM ID MVO6            START 04/06/2007-14.20.00  INTERVAL 000.20.00
                                 CONVERTED TO z/OS V1R8 RMF       END   04/06/2007-14.40.00  CYCLE 1.000 SECONDS
0-------- CRYPTOGRAPHIC ACCELERATOR ----------------------------------------------------------------------------------------------
           -------- TOTAL -------  ------- ME(1024) -----  ----- ME(2048) ------  ------ CRT(1024) -----  ----- CRT(2048) ------
 TYPE   ID   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%
OPCICA   0   0.01      6.5     0.0   0.01      3.3     0.0  <0.01    15.6     0.0  <0.01      4.5     0.0   0.00      0.0     0.0

 -------- ICSF SERVICES EXECUTED ON CCF ---------------------------------------------------------------
        DES ENCRYPTION    DES DECRYPTION    ----- MAC ------    ----- HASH -----    ------ PIN -------
        SINGLE  TRIPLE    SINGLE  TRIPLE    GENERATE  VERIFY      SHA-1  SHA-256    TRANSLATE   VERIFY
 RATE      0.00    0.00      0.00    0.00        0.00    0.00       0.00     0.00         0.00     0.00
 SIZE      0.00    0.00      0.00    0.00        0.00    0.00       0.00     0.00
******************************** BOTTOM OF DATA ********************************************************************
```

Figure 3-5 Crypto Hardware Activity report without local activity.

The CEX2C activity is therefore caused by other logical partitions in the system that share this coprocessor, or it might be ICSF calls that are not part of the set of callable services reported to RMF. This might be a way for tracking, from z/OS, hardware cryptography activity initiated from a Linux logical partition in the same physical system (this assumes, however, a somehow ideal environment where it is proven that only the Linux logical partition issues requests to the hardware coprocessors).

3.4.4 The Workload Activity report

The Workload Activity report provides information about how the requests sent to the CEX2C or CEX2A flow, that is, how often, on a sampling basis, they are either immediately served or put in a queue by ICSF. The fields to look at are CRYPTO% USG and DLY:

- USG reports the percentage of time a TCB or SRB was found to be using a cryptographic processor, that is, the application request to ICSF has been sent out to the coprocessor and occupies one entry among N in the structure maintained by ICSF.
- DLY reports the percentage of time a TCB or SRB was found to be waiting for a cryptographic processor queue, that is, the application request is being kept in a wait queue by ICSF.

Note: The USG and DLY values are based on state samples provided by WLM that pertain to work units running in service or report class periods. If the work unit holds an entry in the structure of N operations in process, the using state count is increased for the service/report class period of the work unit. Likewise, if a work unit is found to be waiting in a queue behind a coprocessor, it is reported as an additional point in a delay count.

The CRYPTO %USG and %DLY values in the WLM report are the percentages of the TOTAL using or delay samples, TOTAL being the sum of all using and delay state samples counted by WLM for all work units in the system. Refer to *z/OS Resource Measurement Facility Report Analysis*, SC33-7991, for a complete list of sampled states, in the WLMGL report chapter.

To provide examples, the authors generated reports pertaining to the service class BATHI, which was defined with two periods. Note that in these examples, assembler loops calling the cryptographic services were almost the only activity in the system. Therefore, the hardware

cryptography reported figures are relatively exaggerated when compared to the figures that you can obtain in a real production environment.

Figure 3-6 shows that, on average, during the interval, the local instance of ICSF was participating for 10.2% of the work units held in wait in the system while it was contributing to 40% of the work units that were executing in period 1.

```
1                                          W O R K L O A D   A C T I V I T Y
                                                                                                                PAGE   1
        z/OS V1R8              SYSPLEX PLEX60            START 04/05/2007-07.43.00 INTERVAL 000.01.00   MODE = GOAL
                               RPT VERSION V1R8 RMF      END   04/05/2007-07.44.00

                                        POLICY ACTIVATION DATE/TIME 09/06/2006 23.03.35

   ------------------------------------------------------------------------------------------------------------------- SERVICE CLASS PERIODS

   REPORT BY: POLICY=WLMPOL     WORKLOAD=BAT_WKL    SERVICE CLASS=BATHI      RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=3
                                                    CRITICAL     =NONE

   TRANSACTIONS   TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE----  SERVICE TIMES  ---APPL %---  PAGE-IN RATES   ---STORAGE---
   AVG      9.84  ACTUAL                 0  SSCHRT   0.0  IOC         0   CPU      2.3   CP      3.85  SINGLE     0.0  AVG    296.73
   MPL      9.84  EXECUTION              0  RESP     0.0  CPU     65483   SRB      0.0   AAPCP   0.00  BLOCK      0.0  TOT   2921.19
   ENDED       0  QUEUED                 0  CONN     0.0  MSO         0   RCT      0.0   IIPCP   0.00  SHARED     0.0  CEN   2921.19
   END/S    0.00  R/S AFFIN              0  DISC     0.0  SRB         8   IIT      0.0                 HSP        0.0  EXP      0.00
   #SWAPS      0  INELIGIBLE             0  Q+PEND   0.0  TOT     65491   HST      0.0   AAP      N/A  HSP MISS   0.0
   EXCTD       0  CONVERSION             0  IOSQ     0.0  /SEC     1092   AAP      N/A   IIP      N/A  EXP SNGL   0.0  SHR      9.84
   AVG ENC  0.00  STD DEV                0                                 IIP      N/A                 EXP BLK    0.0
   REM ENC  0.00                                          ABSRPTN   111                                 EXP SHR    0.0
   MS ENC   0.00                                          TRX SERV  111

   GOAL: EXECUTION VELOCITY 30.0%     VELOCITY MIGRATION:   I/O MGMT  46.7%     INIT MGMT 46.7%

            RESPONSE TIME EX   PERF  AVG   ------ USING% ----- ------------ EXECUTION DELAYS % ------------ ---DLY%-- -CRYPTO%-    %
   SYSTEM                  VEL% INDX ADRSP   CPU  AAP  IIP  I/O  TOT  CPU                                UNKN IDLE  USG  DLY QUIE

   SC60          --N/A--    46.7  0.6   9.8   0.9  N/A  N/A  0.0  1.1  1.1                                98.0  0.0 40.0 10.2  0.0
```

Figure 3-6 Workload Activity report - period 1

The second part (Figure 3-7) shows that ICSF was holding 9.7% of the work units in wait while 40.3% were being executed by the coprocessors in period 2.

```
REPORT BY: POLICY=WLMPOL     WORKLOAD=BAT_WKL    SERVICE CLASS=BATHI      RESOURCE GROUP=*NONE      PERIOD=2 IMPORTANCE=4
                                                    CRITICAL     =NONE

   TRANSACTIONS   TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE----  SERVICE TIMES  ---APPL %---  PAGE-IN RATES   ---STORAGE---
   AVG      0.16  ACTUAL                 0  SSCHRT   0.0  IOC         0   CPU      0.1   CP      0.09  SINGLE     0.0  AVG    327.16
   MPL      0.16  EXECUTION              0  RESP     0.0  CPU      1427   SRB      0.0   AAPCP   0.00  BLOCK      0.0  TOT     50.77
   ENDED       0  QUEUED                 0  CONN     0.0  MSO         0   RCT      0.0   IIPCP   0.00  SHARED     0.0  CEN     50.77
   END/S    0.00  R/S AFFIN              0  DISC     0.0  SRB        26   IIT      0.0                 HSP        0.0  EXP      0.00
   #SWAPS      0  INELIGIBLE             0  Q+PEND   0.0  TOT      1453   HST      0.0   AAP      N/A  HSP MISS   0.0
   EXCTD       0  CONVERSION             0  IOSQ     0.0  /SEC       24   AAP      N/A   IIP      N/A  EXP SNGL   0.0  SHR      0.16
   AVG ENC  0.00  STD DEV                0                                 IIP      N/A                 EXP BLK    0.0
   REM ENC  0.00                                          ABSRPTN   156                                 EXP SHR    0.0
   MS ENC   0.00                                          TRX SERV  156

   GOAL: EXECUTION VELOCITY 20.0%     VELOCITY MIGRATION:   I/O MGMT   0.0%     INIT MGMT  0.0%

            RESPONSE TIME EX   PERF  AVG   ------ USING% ----- ------------ EXECUTION DELAYS % ------------ ---DLY%-- -CRYPTO%-    %
   SYSTEM                  VEL% INDX ADRSP   CPU  AAP  IIP  I/O  TOT  CPU                                UNKN IDLE  USG  DLY QUIE

   SC60          --N/A--     0.0  0.0   0.2   0.0  N/A  N/A  0.0 16.7 16.7                               77.8  0.0 40.3  9.7  0.0
```

Figure 3-7 Workload Activity report - period 2

3.4.5 The Overview report

The RMF Overview report can also track hardware cryptography activity. Figure 3-8 shows the control statements used for such a report and the resulting reported data. The control statements to collect are:

- The using percentage for period 1 in service class BATHI
- The delay percentage for period 1 in service class BATHI
- The same as above for period 2
- The utilization percentage for the CEX2C coprocessor (ID=1)

```
OVW(APUSG1   (APUSGP(S.BATHI.1)))
OVW(APDLY1   (APDLYP(S.BATHI.1)))
OVW(APUSG2   (APUSGP(S.BATHI.2)))
OVW(APDLY2   (APDLYP(S.BATHI.2)))
OVW(CCUTIL   (CRYCTU(1)))
....

DATE    TIME     INT        APUSG1    APDLY1    APUSG2    APDLY2    CCUTIL
MM/DD HH.MM.SS HH.MM.SS
04/05 07.41.00 00.01.00       0.0       0.0       0.0       0.0       0.0
04/05 07.42.00 00.00.59      40.5      10.0       0.0       0.0      50.7
04/05 07.43.00 00.01.00      40.0      10.2      40.3       9.7      98.6
04/05 07.44.00 00.01.00      42.5       7.5      40.1      10.0      98.7
04/05 07.45.00 00.00.59       0.0       0.0      40.3       9.7      72.5
```

Figure 3-8 RMF Overview report

The output data of this Overview report can be interpreted as:

- Interval ending at 07.41: No crypto load
- Interval ending at 07.42: Some crypto load with 50% utilization of coprocessor USING and DELAY samples, but only in period 1. No work spills over into period 2.
- Interval ending at 07.43: A heavy crypto load; the coprocessor is almost 100% utilized. We can now see USING and DELAY samples in period 1 and 2.
- Interval ending at 07.44: Similar to the previous interval.
- Interval ending at 07.45: The cryptographic workload is decreasing. Actually, the only workload left is the one that already spilled over into period 2.

4

Assessing hardware cryptography activity with Tivoli OMEGAMON XE for z/OS and RMF

This chapter provides an overview of how IBM Tivoli OMEGAMON XE for z/OS can be used monitor and assess the hardware cryptography activity in z/OS. The documentation to refer to for more information about Tivoli OMEGAMON XE for z/OS is:

- *Tivoli OMEGAMON XE on z/OS: Planning and Configuration Guid*e, SC32-1822
- *Tivoli OMEGAMON XE on z/OS: User's Guid*e, SC32-1821

4.1 OMEGAMON XE for z/OS support for cryptographic coprocessors

Tivoli OMEGAMON XE for z/OS support for IBM cryptographic coprocessors provides an infrastructure (Figure 4-1) that gives users insight into the initial installation and configuration of the cryptographic coprocessors and ongoing visibility into their performance.

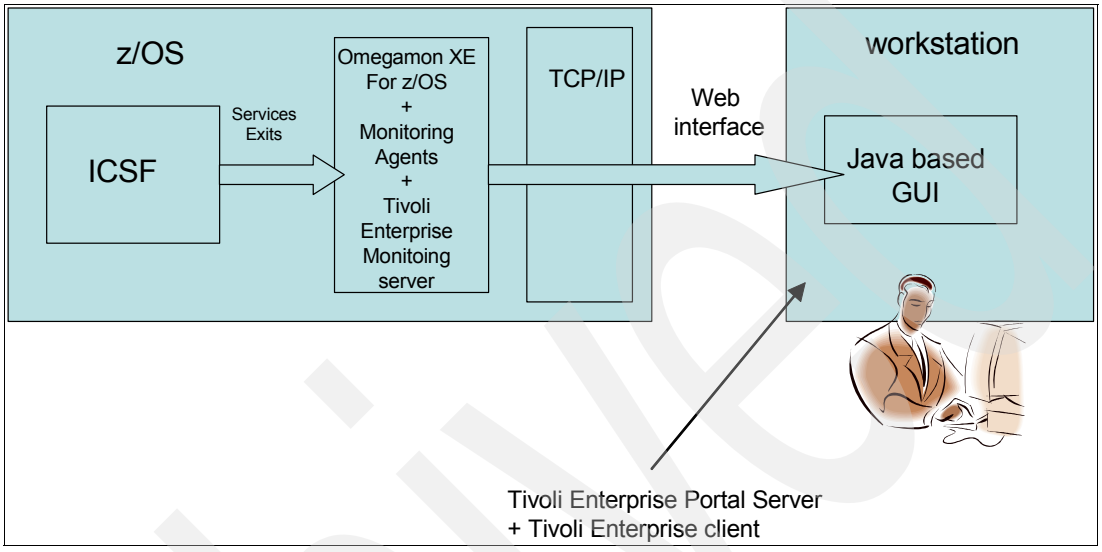


Figure 4-1 The OMEGAMON XE for z/OS Infrastructure we used

This infrastructure provides the functions required to:

- Monitor the ICSF subsystem of z/OS for coprocessor status, configuration errors, and service call performance.
- Monitor the arrival rates of requests, queue lengths, and service calls.
- Scan for enabled or disabled public keys, active or inactive PKDS, and valid or invalid master keys, and issue an alert for any discrepancies.
- Speed problem identification and resolution with Situation Editor, Expert Advice, Take Action, automated alert notification, and customizable workspaces.

In our example, the infrastructure was compacted; functional participants, such as Tivoli Enterprise™ Monitoring Server and Tivoli Enterprise Portal Server share hosts with other elements of the infrastructure, when usually they have distributed hosts of their own.

Note: OMEGAMON for z/OS reports the hardware cryptography activity initiated by the ICSF callable services. ICSF service exits are part of the process of installing OMEGAMON for z/OS. The reporting therefore pertains only to the local hardware cryptography activity as initiated by the local instance of ICSF.

4.2 OMEGAMON XE for z/OS graphical interface

OMEGAMON XE for z/OS can display information about the ICSF status and environment in a graphical format (Figure 4-2).

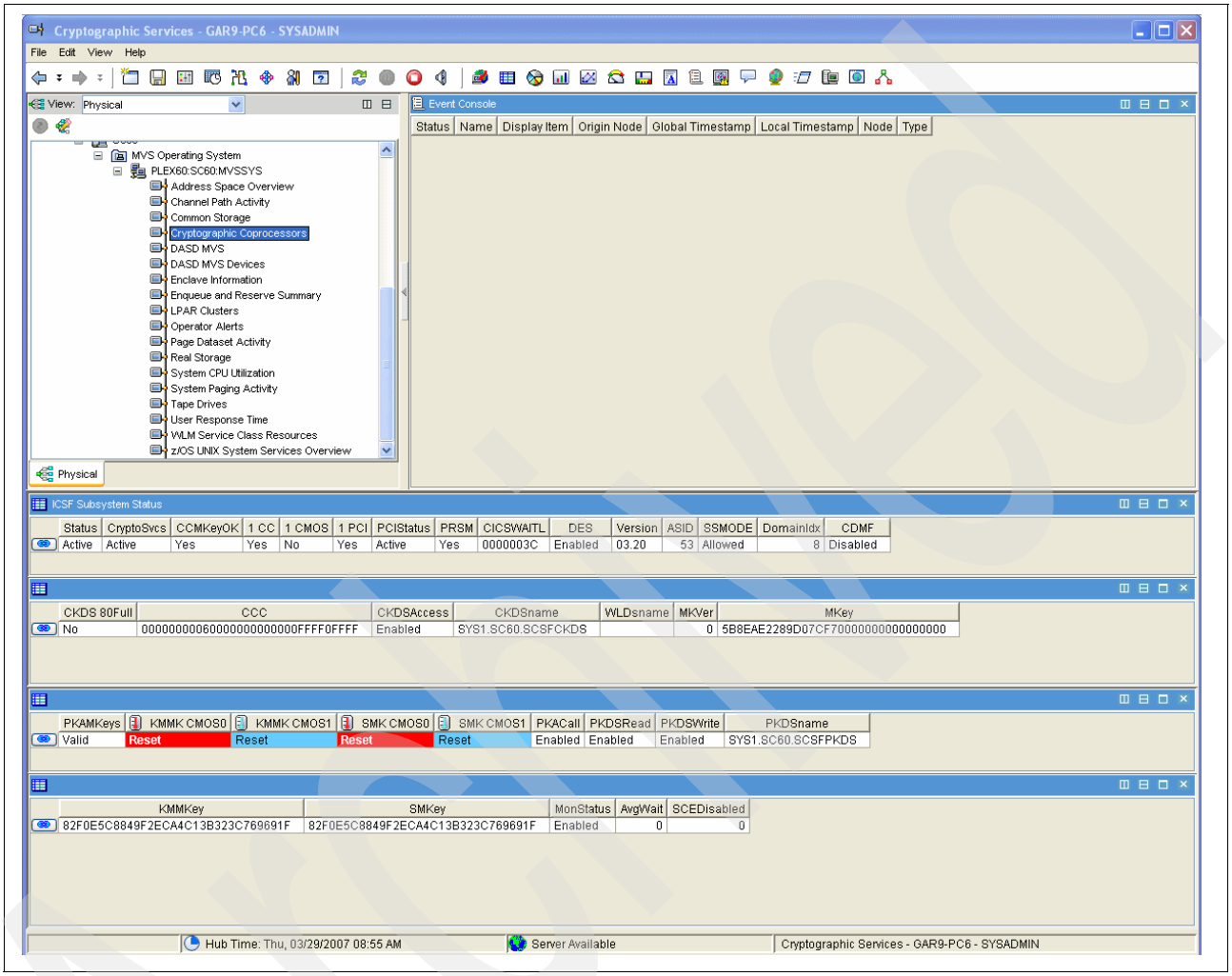
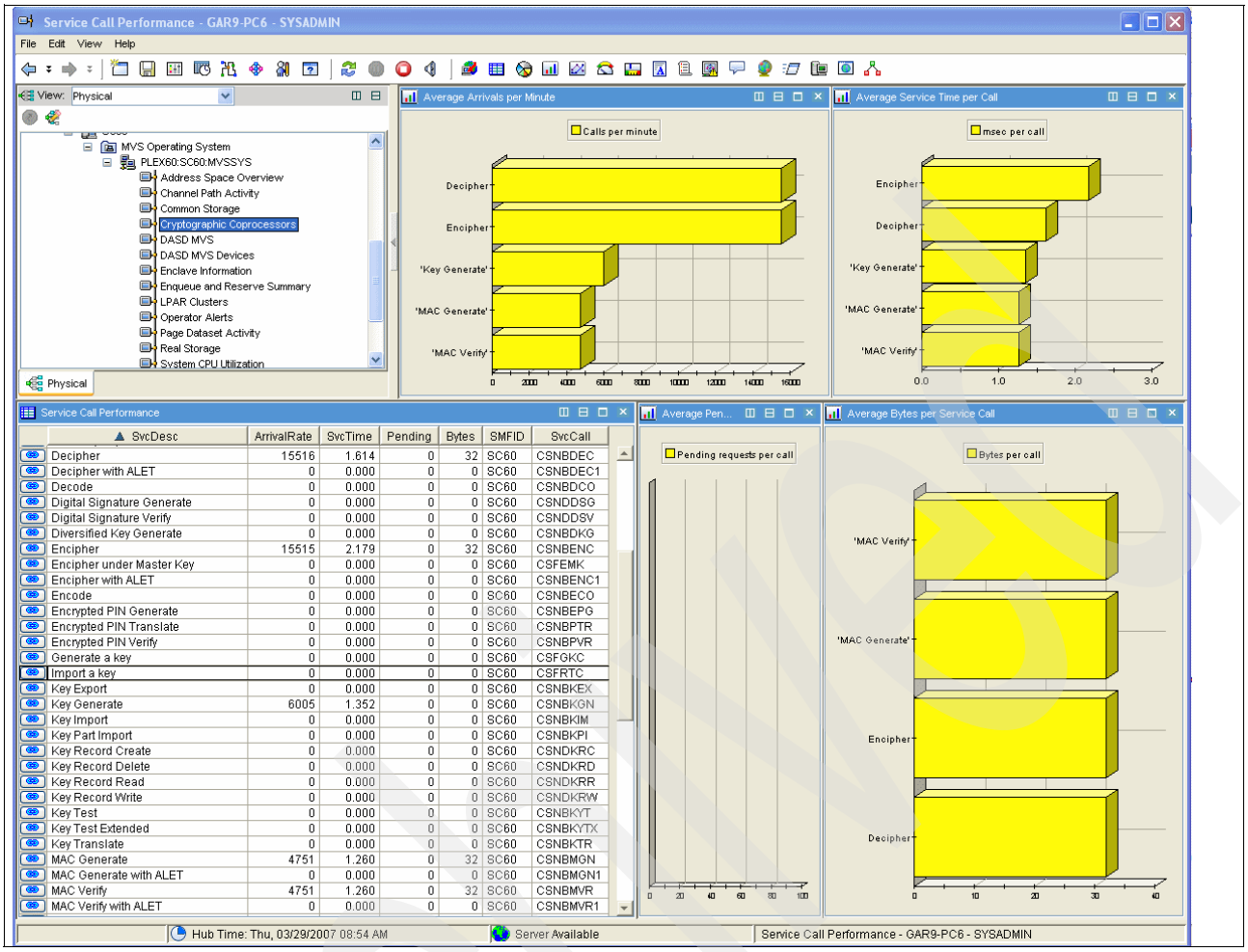


Figure 4-2 OMEGAMON XE - ICSF status display

It can also provide a view of the cryptographic workload performance, per callable service (Figure 4-3 on page 46).



SvcDesc	ArrivalRate	SvcTime	Pending	Bytes	SMFID	SvcCall
Decipher	15516	1.614	0	32	SC60	CSNBDEC
Decipher with ALET	0	0.000	0	0	SC60	CSNBDEC1
Decode	0	0.000	0	0	SC60	CSNBDCO
Digital Signature Generate	0	0.000	0	0	SC60	CSNDDSG
Digital Signature Verify	0	0.000	0	0	SC60	CSNDDSV
Diversified Key Generate	0	0.000	0	0	SC60	CSNBDKG
Encipher	15515	2.179	0	32	SC60	CSNBENC
Encipher under Master Key	0	0.000	0	0	SC60	CSFEMK
Encipher with ALET	0	0.000	0	0	SC60	CSNBENC1
Encode	0	0.000	0	0	SC60	CSNBECO
Encrypted PIN Generate	0	0.000	0	0	SC60	CSNBEPG
Encrypted PIN Translate	0	0.000	0	0	SC60	CSNBPTR
Encrypted PIN Verify	0	0.000	0	0	SC60	CSNBPVR
Generate a key	0	0.000	0	0	SC60	CSFGKC
Import a key	0	0.000	0	0	SC60	CSFRTC
Key Export	0	0.000	0	0	SC60	CSNBKEX
Key Generate	6005	1.352	0	0	SC60	CSNBKGN
Key Import	0	0.000	0	0	SC60	CSNBKIM
Key Part Import	0	0.000	0	0	SC60	CSNBKPI
Key Record Create	0	0.000	0	0	SC60	CSNDKRC
Key Record Delete	0	0.000	0	0	SC60	CSNDKRD
Key Record Read	0	0.000	0	0	SC60	CSNDKRR
Key Record Write	0	0.000	0	0	SC60	CSNDKRW
Key Test	0	0.000	0	0	SC60	CSNBKYT
Key Test Extended	0	0.000	0	0	SC60	CSNBKYTX
Key Translate	0	0.000	0	0	SC60	CSNBKTR
MAC Generate	4751	1.260	0	32	SC60	CSNBMGN
MAC Generate with ALET	0	0.000	0	0	SC60	CSNBMGN1
MAC Verify	4751	1.260	0	32	SC60	CSNBMVR
MAC Verify with ALET	0	0.000	0	0	SC60	CSNBMVR1

Figure 4-3 OMEGAMON XE for z/OS - Cryptographic workload performance

4.3 Measuring hardware cryptography activity with RMF and OMEGAMON XE for z/OS

In this section, we provide examples of hardware cryptography activities that have been measured both with RMF and OMEGAMON XE for z/OS.

4.3.1 SHA-1 activity (CPACF activity)

SHA-1 on System z9 is performed by the CPACF facility in the PU and there is no way to differentiate the CPACF activity from other PU activity in the activity reports. This is also true for the reporting done by OMEGAMON XE for z/OS. However, when the SHA-1 or SHA-256 functions are called with the ICSF CSNBOWH callable service, then ICSF provides accounting information.

For example, in Figure 4-4 on page 47, the SHA-1 activity is reported by RMF in the ICSF Services section without showing any activity in the Cryptographic Coprocessor section.

```
C R Y P T O   H A R D W A R E   A C T I V I T Y                                                    PAGE    1

           z/OS V1R8                SYSTEM ID SC60            START 03/28/2007-09.56.00  INTERVAL 000.00.59
                                    RPT VERSION V1R8 RMF      END   03/28/2007-09.57.00  CYCLE 1.000 SECONDS

-------- CRYPTOGRAPHIC COPROCESSOR --------
           -------- TOTAL --------  KEY-GEN
TYPE   ID   RATE  EXEC TIME  UTIL%    RATE
CEX2C   1   0.00      0.0     0.0     0.00
        2   0.00      0.0     0.0     0.00
        3   0.00      0.0     0.0     0.00

-------- CRYPTOGRAPHIC ACCELERATOR ----------------------------------------------------------------------------------------------------
           -------- TOTAL -------  ------- ME(1024) -----  ----- ME(2048) ------  ------ CRT(1024) -----  ----- CRT(2048) ------
TYPE   ID   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%
CEX2A   0   0.00      0.0     0.0   0.00      0.0     0.0   0.00     0.0     0.0   0.00      0.0     0.0   0.00      0.0     0.0

-------- ICSF SERVICES ----------------------------------------------------------------------------
        DES ENCRYPTION    DES DECRYPTION    ----- MAC ------    ----- HASH -----    ------ PIN -------
        SINGLE  TRIPLE    SINGLE  TRIPLE    GENERATE  VERIFY     SHA-1  SHA-256    TRANSLATE   VERIFY
RATE      0.00    0.00      0.00    0.00        0.00    0.00     52795     0.00         0.00     0.00
SIZE      0.00    0.00      0.00    0.00        0.00    0.00      1256     0.00
******************************** BOTTOM OF DATA ********************************************************************************
```

Figure 4-4 RMF reporting of SHA-1 requests

OMEGAMON XE Online Monitor can provide a snapshot (Figure 4-5) for the same activity.

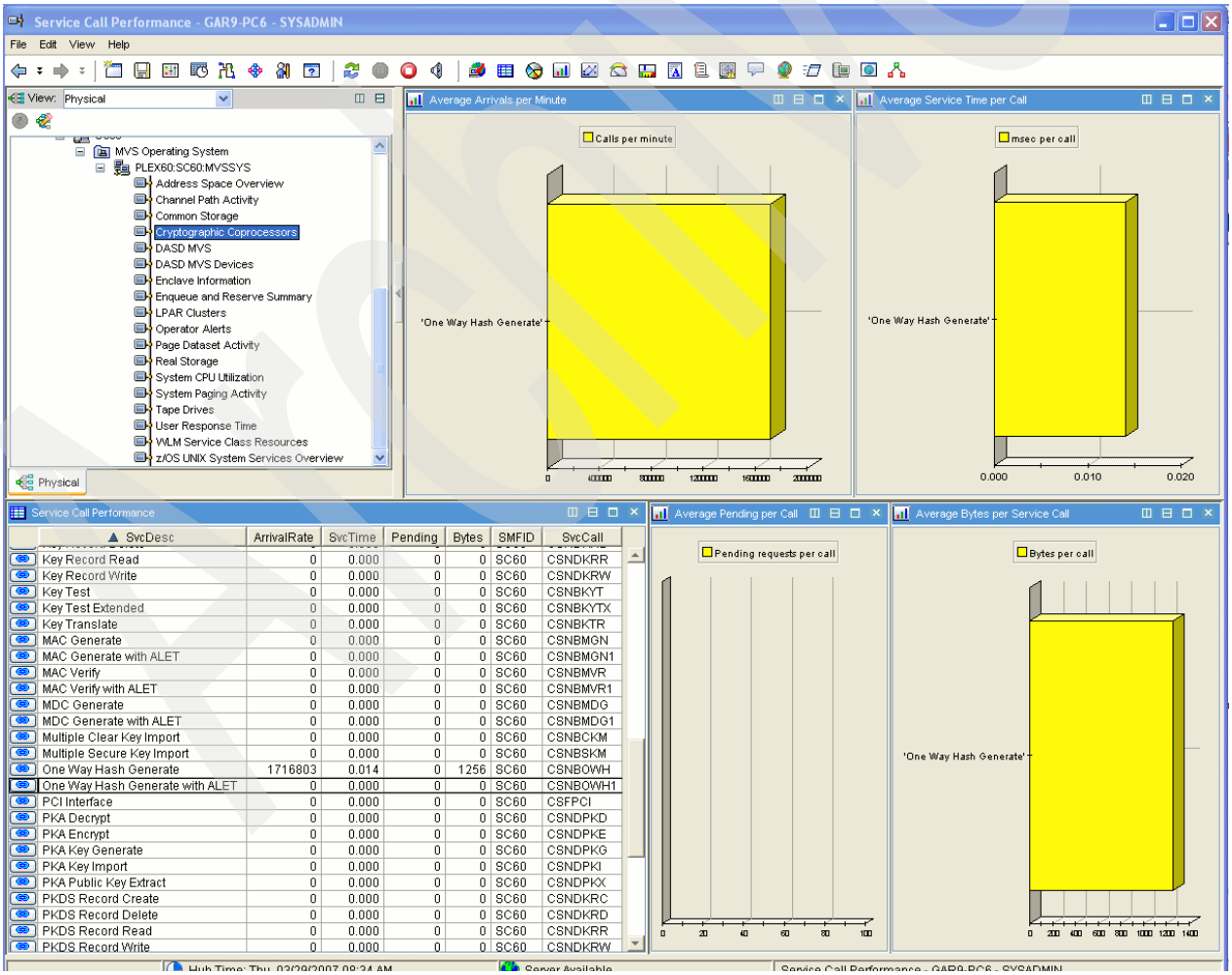


Figure 4-5 OMEGAMON XE graphical display of SHA-1 activity

4.3.2 CEX2C activity

In this example, an application is requesting DES encryption and decryption using secure key. RMF displays the coprocessor activity in a post-processor report (Figure 4-6).

```
C R Y P T O   H A R D W A R E   A C T I V I T Y                                 PAGE    1

           z/OS V1R8               SYSTEM ID SC60           START 03/28/2007-07.33.00  INTERVAL 000.00.59
                                   RPT VERSION V1R8 RMF     END   03/28/2007-07.34.00  CYCLE 1.000 SECONDS
-------- CRYPTOGRAPHIC COPROCESSOR --------
          -------- TOTAL --------  KEY-GEN
TYPE    ID   RATE  EXEC TIME  UTIL%    RATE
CEX2C 1 767.8       0.9      69.8      0.00

-------- CRYPTOGRAPHIC ACCELERATOR ----------------------------------------------------------------------------------------
          -------- TOTAL -------  ------- ME(1024) -----  ----- ME(2048) ------  ------ CRT(1024) -----  ----- CRT(2048) ------
TYPE    ID   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE EXEC TIME UTIL%   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%
CEX2A    0   0.00      0.0     0.0   0.00      0.0     0.0   0.00     0.0     0.0   0.00      0.0     0.0   0.00      0.0     0.0

-------- ICSF SERVICES ----------------------------------------------------------------------
        DES ENCRYPTION    DES DECRYPTION    ----- MAC ------    ----- HASH -----    ------ PIN -------
        SINGLE  TRIPLE    SINGLE  TRIPLE    GENERATE  VERIFY      SHA-1  SHA-256    TRANSLATE   VERIFY
RATE     255.9    0.00     255.9    0.00        0.00    0.00       0.00     0.00         0.00     0.00
SIZE      32.00   0.00      32.00   0.00        0.00    0.00       0.00     0.00
```

Figure 4-6 RMF reporting of CEX2C activity.

The OMEGAMON XE for z/OS graphic also displays this activity (Figure 4-7).

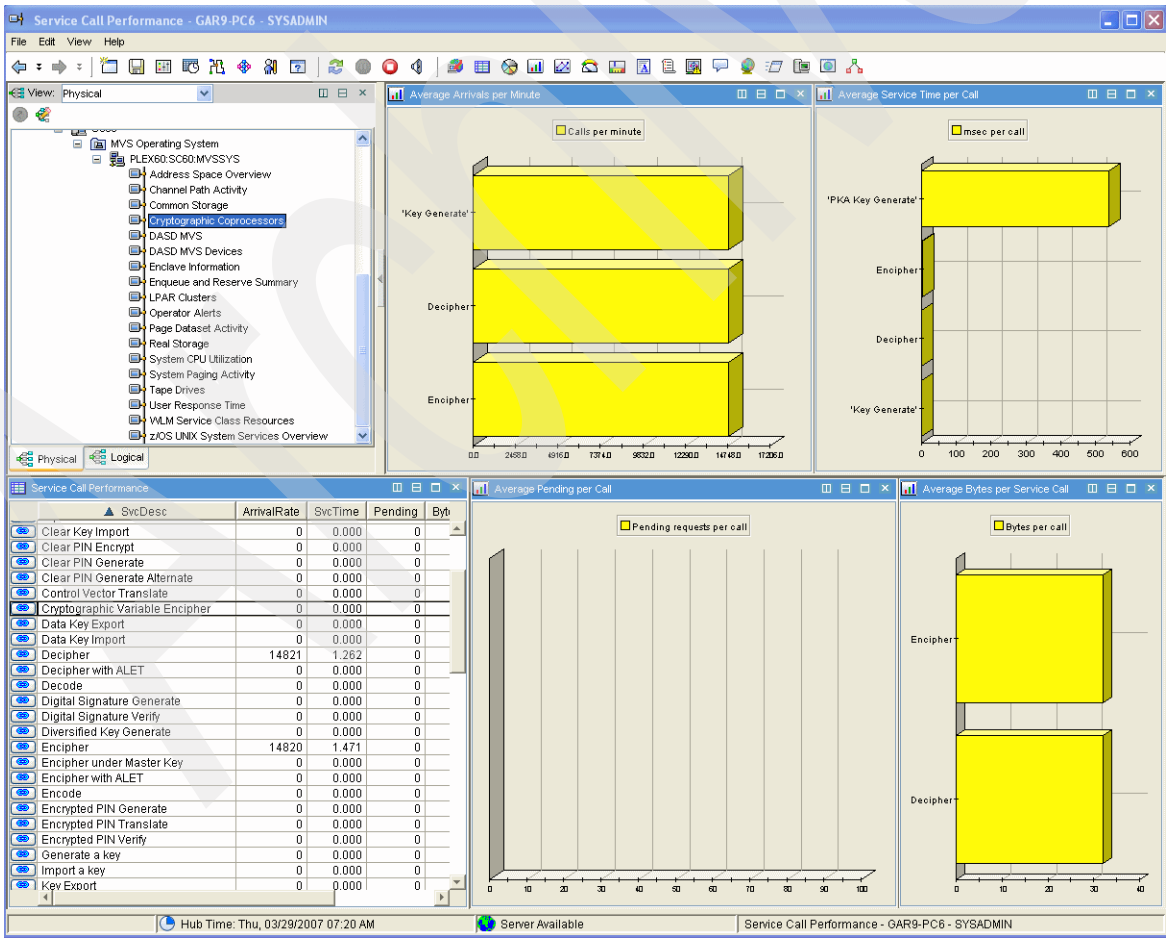


Figure 4-7 OMEGAMON XE graphical display of CEX2C activity

4.3.3 CEX2A activity

CEX2A supports a very limited set of RSA operations using clear key only. Digital Signature Verify is one of the functions supported by a CEX2A. In this example, the application is calling this service and the CEX2A activity is recorded by RMF (Figure 4-8) and displayed by OMEGAMON XE for z/OS (Figure 4-9).

```
C R Y P T O   H A R D W A R E   A C T I V I T Y                                                    PAGE    1

           z/OS V1R8                SYSTEM ID SC60            START 03/30/2007-08.51.00  INTERVAL 000.01.00
                                    RPT VERSION V1R8 RMF      END   03/30/2007-08.52.00  CYCLE 1.000 SECONDS
-------- CRYPTOGRAPHIC COPROCESSOR --------
          -------- TOTAL --------  KEY-GEN
TYPE   ID   RATE  EXEC TIME  UTIL%    RATE
CEX2C   1   0.00      0.0     0.0     0.00


-------- CRYPTOGRAPHIC ACCELERATOR ----------------------------------------------------------------------------------------------------
          -------- TOTAL -------  ------- ME(1024) -----  ----- ME(2048) ------  ------ CRT(1024) -----  ----- CRT(2048) ------
TYPE   ID   RATE  EXEC TIME UTIL%   RATE  EXEC TIME UTIL%  RATE EXEC TIME UTIL%  RATE  EXEC TIME UTIL%  RATE  EXEC TIME UTIL%
CEX2A   0   1466      0.3    46.1   1466      0.3    46.1  0.00     0.0     0.0  0.00      0.0     0.0  0.00      0.0     0.0

-------- ICSF SERVICES ----------------------------------------------------------------------
        DES ENCRYPTION    DES DECRYPTION     ----- MAC ------     ----- HASH -----    ------ PIN -------
        SINGLE  TRIPLE    SINGLE  TRIPLE     GENERATE  VERIFY     SHA-1  SHA-256    TRANSLATE   VERIFY
RATE      0.00    0.00      0.00    0.00         0.00    0.00      0.00     0.00         0.00     0.00
SIZE      0.00    0.00      0.00    0.00         0.00    0.00      0.00     0.00
```

Figure 4-8 RMF reporting of CEX2A activity

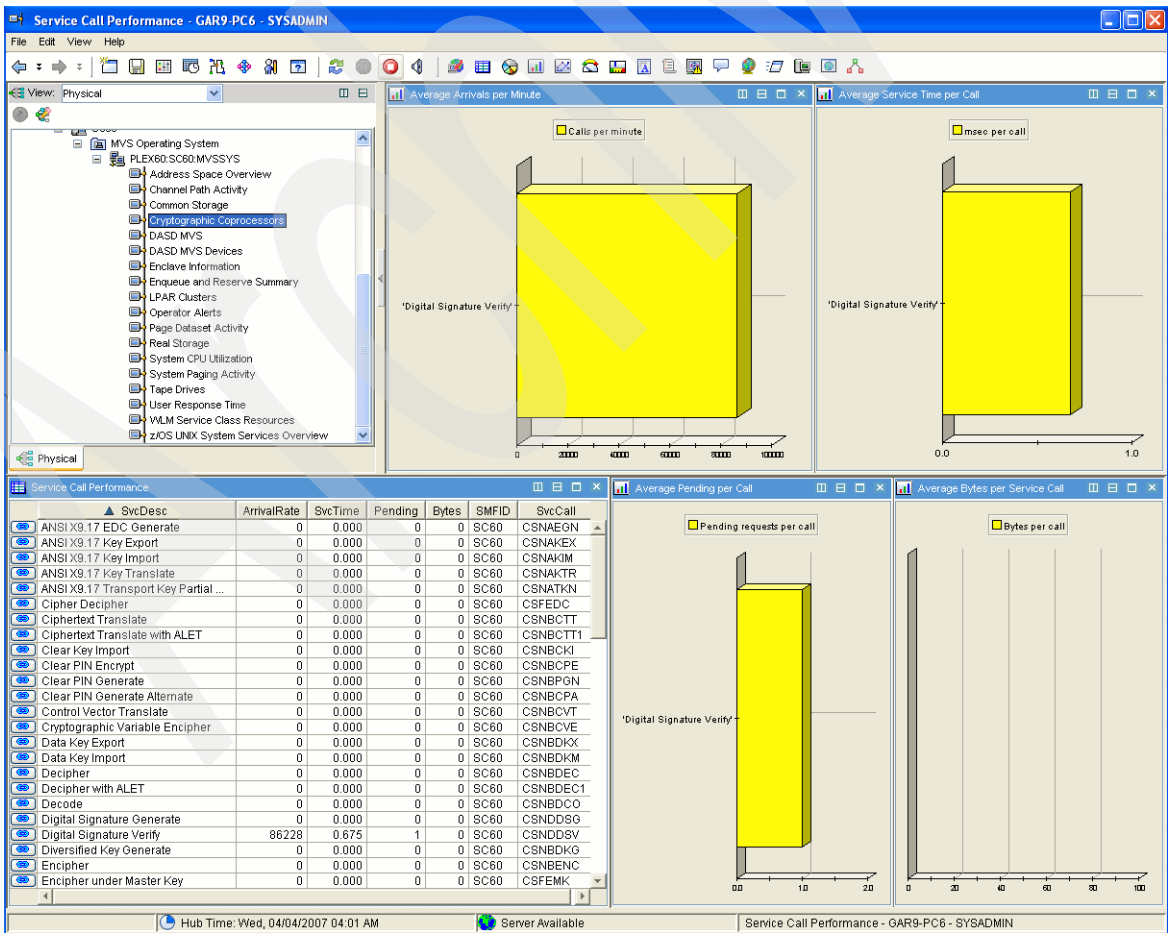
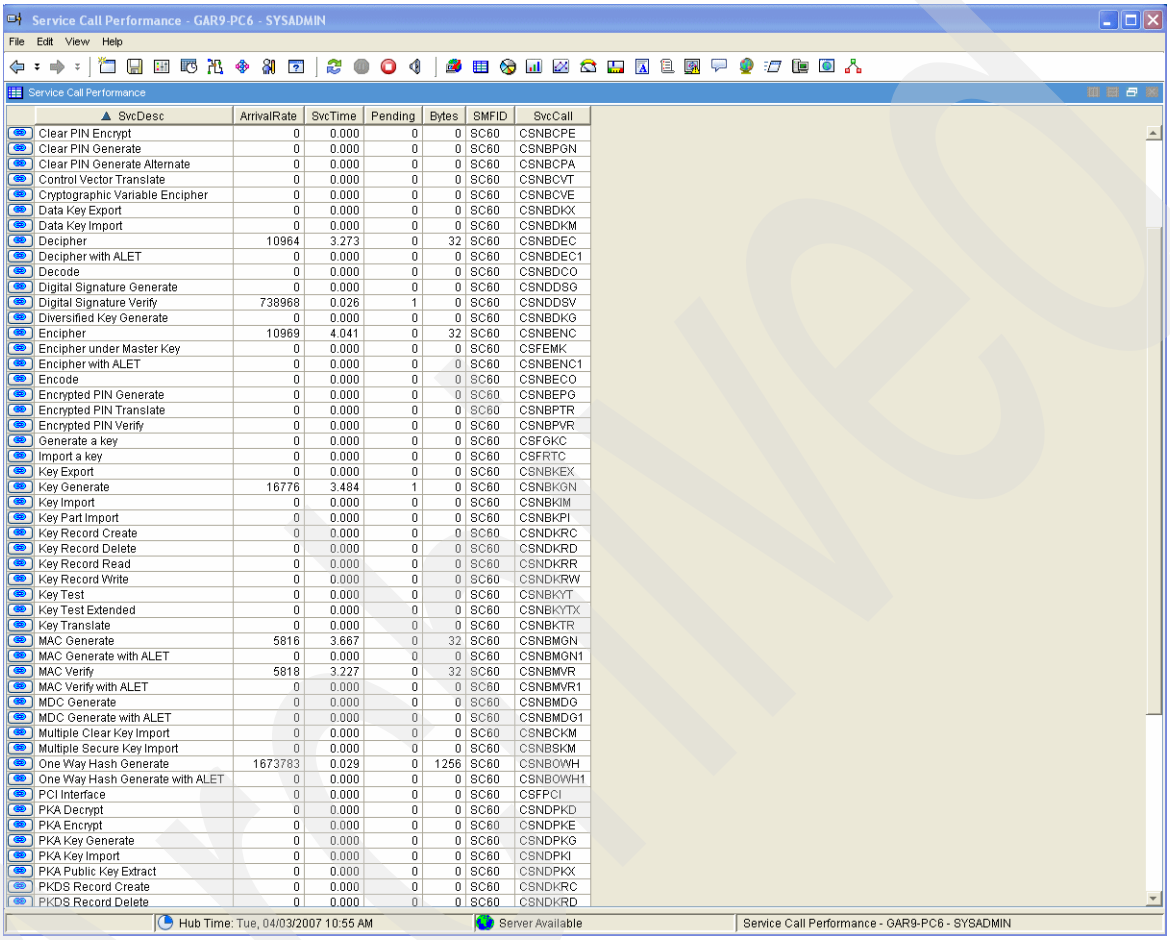


Figure 4-9 OMEGAMON XE graphical display of CEX2A activity

4.4 Using the OMEGAMON XE Service Call Performance workspace

The Service Call Performance workspace displays bar charts of the top ten service calls by arrival rate, service time, pending, and bytes processed. The lower right table shows details of all 78 service calls monitored (Figure 4-10).



SvcDesc	ArrivalRate	SvcTime	Pending	Bytes	SMFID	SvcCall
Clear PIN Encrypt	0	0.000	0	0	SC60	CSNBCPE
Clear PIN Generate	0	0.000	0	0	SC60	CSNBPGN
Clear PIN Generate Alternate	0	0.000	0	0	SC60	CSNBCPA
Control Vector Translate	0	0.000	0	0	SC60	CSNBCVT
Cryptographic Variable Encipher	0	0.000	0	0	SC60	CSNBCVE
Data Key Export	0	0.000	0	0	SC60	CSNBDKX
Data Key Import	0	0.000	0	0	SC60	CSNBDKM
Decipher	10964	3.273	0	32	SC60	CSNBDEC
Decipher with ALET	0	0.000	0	0	SC60	CSNBDEC1
Decode	0	0.000	0	0	SC60	CSNBDCO
Digital Signature Generate	0	0.000	0	0	SC60	CSNDDSG
Digital Signature Verify	738968	0.026	1	0	SC60	CSNDDSV
Diversified Key Generate	0	0.000	0	0	SC60	CSNBDKG
Encipher	10969	4.041	0	32	SC60	CSNBENC
Encipher under Master Key	0	0.000	0	0	SC60	CSFEMK
Encipher with ALET	0	0.000	0	0	SC60	CSNBENC1
Encode	0	0.000	0	0	SC60	CSNBECO
Encrypted PIN Generate	0	0.000	0	0	SC60	CSNBEPG
Encrypted PIN Translate	0	0.000	0	0	SC60	CSNBPTR
Encrypted PIN Verify	0	0.000	0	0	SC60	CSNBPVR
Generate a key	0	0.000	0	0	SC60	CSFGKC
Import a key	0	0.000	0	0	SC60	CSFRTC
Key Export	0	0.000	0	0	SC60	CSNBKEX
Key Generate	16776	3.484	1	0	SC60	CSNBKGN
Key Import	0	0.000	0	0	SC60	CSNBKIM
Key Part Import	0	0.000	0	0	SC60	CSNBKPI
Key Record Create	0	0.000	0	0	SC60	CSNDKRC
Key Record Delete	0	0.000	0	0	SC60	CSNDKRD
Key Record Read	0	0.000	0	0	SC60	CSNDKRR
Key Record Write	0	0.000	0	0	SC60	CSNDKRW
Key Test	0	0.000	0	0	SC60	CSNBKYT
Key Test Extended	0	0.000	0	0	SC60	CSNBKYTX
Key Translate	0	0.000	0	0	SC60	CSNBKTR
MAC Generate	5816	3.667	0	32	SC60	CSNBMGN
MAC Generate with ALET	0	0.000	0	0	SC60	CSNBMGN1
MAC Verify	5818	3.227	0	32	SC60	CSNBMVR
MAC Verify with ALET	0	0.000	0	0	SC60	CSNBMVR1
MDC Generate	0	0.000	0	0	SC60	CSNBMDG
MDC Generate with ALET	0	0.000	0	0	SC60	CSNBMDG1
Multiple Clear Key Import	0	0.000	0	0	SC60	CSNBCKM
Multiple Secure Key Import	0	0.000	0	0	SC60	CSNBSKM
One Way Hash Generate	1673783	0.029	0	1256	SC60	CSNBOWH
One Way Hash Generate with ALET	0	0.000	0	0	SC60	CSNBOWH1
PCI Interface	0	0.000	0	0	SC60	CSFPCI
PKA Decrypt	0	0.000	0	0	SC60	CSNDPKD
PKA Encrypt	0	0.000	0	0	SC60	CSNDPKE
PKA Key Generate	0	0.000	0	0	SC60	CSNDPKG
PKA Key Import	0	0.000	0	0	SC60	CSNDPKI
PKA Public Key Extract	0	0.000	0	0	SC60	CSNDPKX
PKDS Record Create	0	0.000	0	0	SC60	CSNDKRC
PKDS Record Delete	0	0.000	0	0	SC60	CSNDKRD

Figure 4-10 The Service Call Performance workspace

Residual activity

We used several assembler loops of ICSF calls in parallel to obtain this display. However, we found out that, even after stopping all our loops, the Service Call Performance workspace was still showing hardware cryptography activity. This is because this report shows the rolling averages of the last 10 minutes of collected data, and historical sampling by OMEGAMON XE occurs at 5-minute intervals. Therefore, the figures for arrivals are computed over the last 10 minutes and continue to be computed as rolling averages after the job completes. The previously collected minutes of activity continue to be included in the rolling average until they are “old” enough. Therefore, the activity numbers go to zero only after a period where no cryptographic activity is issued.

In practice, it is highly unlikely that a cryptographic coprocessor workload is submitted in such high rates. A more realistic workload issues cryptographic service requests followed by some interleaving I/O operations and the rolling average method more closely resembles the actual arrival of hardware cryptography requests.

5

Synthesis of the available measurements data and their interpretation

In this chapter, we summarize the measurement data available today to users of System z hardware cryptography with z/OS, and we attempt to provide the readers with a better understanding of the actual meaning of the figures that they might themselves collect.

In this chapter, we use IBM published measurements that were released in a Web document available at:

`www.ibm.com/servers/eserver/zseries/security/cryptography.html`

Important: In this chapter, we do not address observations that are related to the performance of CPACF, because we are not in a position to measure precisely the PU cycles it consumes and then make assumptions to the CPU time dedicated to the CPACF operations. However, CPACF measurement data is also available in the IBM Web document. It demonstrates how fast the CPACF operates and it is therefore expected, in the vast majority of use cases, not to have a significant negative effect on PU performance.

5.1 A review of the z/OS cryptographic services flow and the RMF reporting infrastructure

In this section, we follow the path of a cryptographic service from ICSF to the coprocessor and back with a response to ICSF. Eventually, the level of performance for this process is globally reflected in the RMF Crypto Activity Report. Figure 5-1 is a graphical representation.

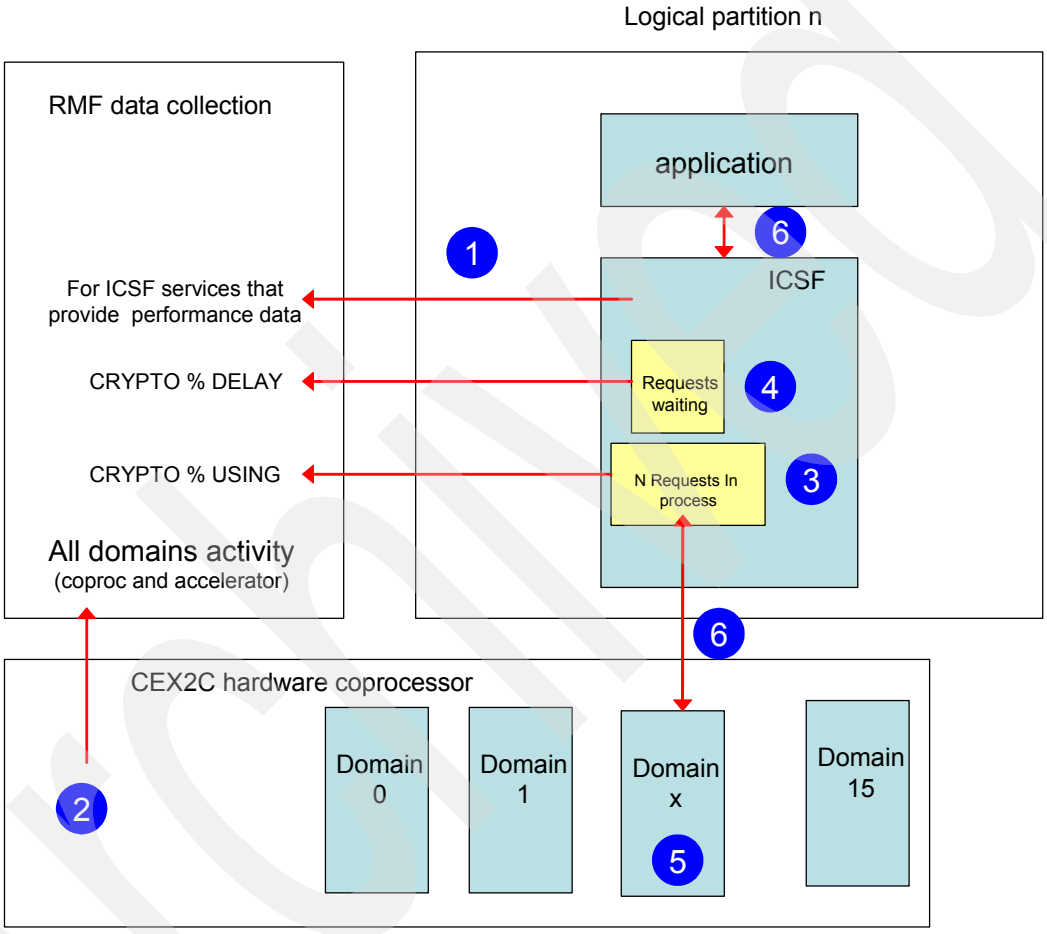


Figure 5-1 RMF reporting infrastructure and cryptographic service requests flow.

The numbered items in Figure 5-1 are:

- Item 1: RMF ICSF data collection. This is performed at the logical partition level, that is, implicitly at the coprocessor cryptographic domain level. Not all service calls are reported by ICSF, but these are:
  - DES and triple-DES encryption and decryption, as performed in the accessible CEX2C domain. These are always secure key operations, and the ICSF callable services are CSNBENC and CSNBDEC. RMF exploits this information to produce a report with an arrival rate of requests and an average length of data.

Note: The CSNBYE and CSBNSYD ICSF services, which can invoke CPACF, are not reported to RMF.

- MAC Generate and Verify as performed in the CEX2C domain. These are always secure key operations, and the ICSF services are CSNBMGM and CSNBMVR. RMF exploits this information to produce a report with an arrival rate for requests and an average length of data.
- SHA-1 and SHA-256 generation.In System z990, z890, and z9, these operations are performed by CPACF (SHA-256 is available in System z9 only), which is invoked by the ICSF CSNBOWH callable service. RMF exploits this information to produce a report with an arrival rate for requests and an average length of data.
- Encrypted PIN Translate and Verify as performed in the CEX2C domain. These are always secure key operations, and the ICSF services are CSNBPTR and CSNBPVR. RMF exploits this information to produce a report with an arrival rate for requests.

- Item 2: RMF CEX2C feature data collection. CEX2C reports the total amount of operations performed, regardless of domain. The operations are differentiated only by:
  - CEX2C coprocessor mode of operations (actual coprocessor or accelerator mode)
  - RSA key length for RSA operations executed in accelerator mode
  - RSA key generation operations

  CEX2C also provides the elapsed execution time, which is the time between receiving the request and responding for all operations. RMF uses it to report execution time, request arrival rate, and coprocessor utilization percentage, which refers to 1 sec of elapsed time.
- Item 3: The structure at N entries, N being “8” as of the writing of this book for z/OS V1R9 with ICSF FMID HCR7740, that ICSF uses to keep track of the concurrent requests that it submits to the coprocessor. The occupation of one entry by an on-going request is sampled by RMF as a unit of work in the system that has the *using* status.

  Note: These are the operations that an ICSF instance drives at the cryptographic domain that it has access to. Each ICSF instance in the system ignores what other logical partitions are doing in terms of submitting their own workload to the coprocessor.

- Item 4: This is the queue, maintained by ICSF, of requests waiting for co-processor available, that is, the requests waiting for an entry to be freed in the N structure.
- Item 5: The multithreading capability of the CEX2C card, that is, the ability for more than one request to progress at the same time in the coprocessor. The multithreading ratio depends both on the amount and mix of requests.
- Item 6: The delay created by ICSF with respect to the real end of the operation at the coprocessor. After the CEX2C completes a request, it queues the results to ICSF. ICSF obtains these results by periodically polling the completion queue or checking the queue when it sends new work to CEX2C. This affects the time it takes to send the coprocessor completion status back to the requestor. It can also prevent the coprocessor from being fully busy (especially when the arrival period of new requests exceeds the polling interval).

5.2 IBM measurements of CEX2C throughput and scalability

The CEX2C feature contains two separate PCIXCC (or 4764-001) cards, which means that ICSF can balance the workload at the application request level. This creates an opportunity to increase the global throughput of the feature if many applications are concurrently sending ICSF requests for CEX2C services. The throughput, in the case of concurrent requests to the coprocessor, can also be augmented by the PCIXCC design, which lends itself to some

degree of parallel processing, under control of its internal operating system, to optimize the use of the various cryptographic engines.

In this section, we report excerpts of the measurements available in the IBM Web document that demonstrate this behavior of the CEX2C coprocessor.

5.2.1 CEX2C throughput increase with concurrent z/OS application requests

The throughput capability of the CEX2C is demonstrated in the performance measurements that IBM provides at:

`www.ibm.com/servers/eserver/zseries/security/cryptography.html`

Attention: The examples in this section were collected using repetitive and simple assembler language loops that called the different ICSF services. The throughput variations and scalability factor that are shown depend entirely on the coprocessor and ICSF intrinsic performance and are not affected by the contention that can occur in a real customer production environment.

Symmetric algorithms

Table 5-1 shows measurements for single-DES encryption performed by one CEX2C coprocessor in a z9 BC model S04. These measurements were taken with a single job continuously calling ICSF.

Table 5-1 One CEX2 Coprocessor single-DES CBC Encipher - One single flow of requests

Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	908.60	58.15
256	908.20	232.5
1024	899.40	921.0
4096	612.10	2507.2
64K	60.77	3982.7
1M	3.94	4130.1

The execution of the cryptographic operation in the CEX2C card is asynchronous to the z9 CP execution. Only one job is run on the CP, so the next cryptographic service request is sent to the coprocessor only when ICSF receives the result of the previous cryptographic operation and it is reflected in the application. Therefore, there is a considerable delay before the application can initiate the next cryptographic operation. This inefficiency is removed when the host program consists of several jobs requesting the same cryptographic operation repetitively.

ICSF and the CEX2C multitasking capability allows for enqueueing and dequeueing of requests in parallel with cryptographic operations being performed. This yields, for our example, the figures in Figure 5-2 on page 55, where we have added the throughput improvement computed as the ratio of the operations performed per second in the seven-job and the single-job case.

Table 5-2 One CEX2C coprocessor Single DES CBC Encipher - 7 concurrent flows of execution

Data length (bytes)	Operations/sec	x 10**3 bytes/sec	Throughput improvement ration vs. single job
64	1388.0	88.85	1.52
256	1293.0	331.1	1.42
1024	1043.0	1068.7	1.16
4096	812.1	3326.5	1.33
64K	78.31	5132.1	1.29
1M	5.06	5306.5	1.28

The throughput increase between one single job and seven jobs demonstrates some degree of interleaving between the concurrent flows of cryptographic service requests.

Note: The throughput here is not multiplied by a factor of 7 (the amount of concurrent requests flows), so there is an extension of the time it takes to perform these requests. This is not addressed in the IBM Web document; however, we provide our own observations on this effect in 5.3.1, “Extension of the average elapsed execution time” on page 57.

The IBM document also provides the measurements shown in Table 5-3 for other CEX2C cryptographic services that use symmetric encryption, where we have also added the throughput improvement ratio.

Table 5-3 Other examples of symmetric throughput improvement

CEX2C symmetric key operations - Examples	Operations/sec 1 job	Operations/sec 7 concurrent jobs	Throughput improvement ratio
Key Generate (operational DES KEYGENKY key)	617	932	1.51
Clear PIN Generate Alternate (DES OPINENC + DES PINGEN keys)	671	990	1.48
Clear PIN Generate (16 digits) (DES PINGEN key)	910	1,394	1.53
Encrypted PIN Translation (DES IPINENC key + DES OPINENC key)	909	1,101	1.21
Encrypted PIN Translation (2 UKPT enabled KEYGENKY keys)	313	332	1.06
Encrypted PIN Verification (UKPT enabl.KEYGENKY+DES PINVER keys)	458	483	1.05

Asymmetric algorithms

The IBM Web document provides the values (Figure 5-4 on page 56) that demonstrate a CEX2C throughput improvement when submitting several flows of concurrent requests. We have again added the throughput improvement ratio.

Table 5-4 Digital Signature Generate

Operations on 2096-S04 with 1 CP	Operations/sec 1 job	Operations/sec 7 concurrent jobs	Throughput improvement ratio
DSG-CRT 512 bit	865	1115	1.29
DSG-CRT 1024 bit	612	997	1.63
DSG-CRT 2048 bit	268	466	1.74

5.2.2 Throughput scalability with additional coprocessors

In this section, we report excerpts of the measurements available in the IBM Web document that demonstrate the throughput scalability when you add CEX2C coprocessors.

Symmetric algorithms

The IBM Web document presents measurements performed with the same jobs repetitively requesting the same cryptographic operation for Single DES, Triple DES, and Single DES MAC with four CEX2C coprocessors and shows a throughput close to 4 times the throughput obtained when running 7 jobs with one CEX2C.

Asymmetric algorithms

Table 5-5 shows the measurements performed when running repetitive loops of the ICSF Digital Signature Generate service. These measurements demonstrate the significant improvement in and scalability of throughput when there are parallel threads at the coprocessor level and additional coprocessors are running.

Table 5-5 Digital Signature Generate on 2096-S04 with up to 4 CPs

Digital Signature Generate service	Operations/sec 1 job on 2096-S04 with 1 CP	Operations/sec 7 concurrent jobs on 2096-S04 with 1 CP	Operations/sec 14 concurrent jobs on 2096-S04 with 2 CPs	Operations/sec2 28 concurrent jobs on 2096-S04 with 4 CPs
DSG-CRT 512 bit	865	1115	2233	4329
DSG-CRT 1024 bit	612	997	2000	4034
DSG-CRT 2048 bit	268	466	932	1860

5.3 Our additional observations

In this section, we provide our own observations of the IBM published measurements, consolidated with some simple measurements that we performed during our residency, and the interpretation of this data.

5.3.1 Extension of the average elapsed execution time

We used the measurements that we collected for throughput and scalability to determine the effect that symmetric and asymmetric algorithms have on the average elapsed execution time.

Symmetric algorithms

Using Table 5-1 on page 54 and Table 5-3 on page 55, we built the table in Table 5-6. The observed throughput indicates some level of "parallelization" of the requests. However, because the throughput is not multiplied by 7 for 7 concurrent flows of requests, there is implicitly an extension of the elapsed execution time for each flow of requests. We computed an average ratio of extension as follows:

```
(number of operations per second of a single job x 7)/(number of operations per
second when running the 7 jobs concurrently)
```

Table 5-6 Comparison of one flow of execution vs. seven flows.

Length of data blocks (bytes)	Operations per second with one flow of execution	Operations per second with 7 flows of execution	Throughput improvement ratio	Computed average elapsed time extension
64	908.60	1388.0	1.52	4.58
256	908.20	1293.0	1.42	4.92
1024	899.40	1043.0	1.16	6.03
4096	612.10	812.1	1.33	5.27
64K	60.77	78.31	1.29	5.43
1M	3.94	5.06	1.28	5.45

Asymmetric algorithms

We used the same reasoning for the asymmetric algorithm measurements in Table 5-4 on page 56. Table 5-7 shows the results.

Table 5-7 Digital Signature Generate on 2096-S04 with 4 CPs.

Digital Signature Generate operations on 2096-S04 with 1 CP	Operations/sec 1 job	Operations/sec 7 concurrent jobs	Throughput improvement ratio	Computed average elapsed time extension
DSG-CRT 512 bit	865	1115	1.29	5.43
DSG-CRT 1024 bit	612	997	1.63	4.3
DSG-CRT 2048 bit	268	466	1.74	4.02

Our own measurements and observations

We exercised our CEX2C feature on a System z9-109 (model S18, with two CPs) with assembler language loops calling basic services of ICSF. The CEX2C feature had a coprocessor in coprocessor mode and another in accelerator mode. Therefore, our observations pertain to the behavior of a single coprocessor when serving one or multiple concurrent flows of requests. We were the only users of the coprocessor in the system.

Table 5-8 shows the measurements we took to address more precisely the extension of the average elapsed execution time and the coprocessor utilization percentage from the RMF Crypto Hardware Activity report. The T-DES flow of requests is a repetitive loop of calls for Triple-DES key generation, followed by Triple-DES encryption and its subsequent decryption of 32 bytes of data.

Table 5-8 Average elongation time for our test jobs.

Concurrent flows of requests	Coprocessor utilization (from RMF)	Coprocessor execution time (from RMF)	Real elapsed time	Average elapsed time extension ratio	Comments
1	76.9 %	1.3 msec	189 sec	1	This is a single flow of T-DES calls used as a reference flow.
2	95.0 %	1.1 msec	304 sec	1.6	This a mix of the T-DES call reference flow with one flow of single DES calls.
5	100 %	1.3 msec	788 sec	4.2	These are 5 concurrent executions of the T-DES reference flow.

For the two flows of concurrent requests, we mixed a reference flow (T-DES) with a flow of single-DES requests.

5.3.2 Coprocessor utilization percentage reported by RMF

We had to start at least five concurrent executions of the reference loop (see Table 5-8) to achieve 100% of coprocessor utilization as reported by RMF, based on the exploitation of the data that is provided by the CEX2C feature.

Note: Adding workload to the coprocessor resulted in a very noticeable extension of the average job execution elapsed time.

5.3.3 The CRYPTO% DLY in the Workload Activity RMF report

We used the Workload Activity RMF report to correlate the CRYPTO% DLY with the volume of our test workload (see 3.4.4, “The Workload Activity report” on page 40, for more information). Our test workload was the only significant user workload in the system, so the CRYPTO% DLY remained at zero up to the point where the amount of concurrent request flows “overflowed” the structure at N entries (N=8), where ICSF keeps track of requests in process in the coprocessor. We achieved this with 10 concurrent executions of our T-DES reference flow. RMF then reported CRYPTO% USG = 40 and CRYPTO% DLY = 10.2.

5.4 Our considerations of the RMF indicators of coprocessor load and their interpretation

We are not in a position to deliver a capacity planning methodology for System z hardware cryptography in this document. However, based on the available measurement data and our observations, you might be able detect actual performance data or trends indicating that adding one or more coprocessors to your systems would contribute to improving the global hardware cryptography throughput.

In this section, we review a few measurements that are provided in the Crypto Hardware Activity RMF report (Figure 5-2) along with considerations that we think to be helpful for a deeper interpretation of this data.

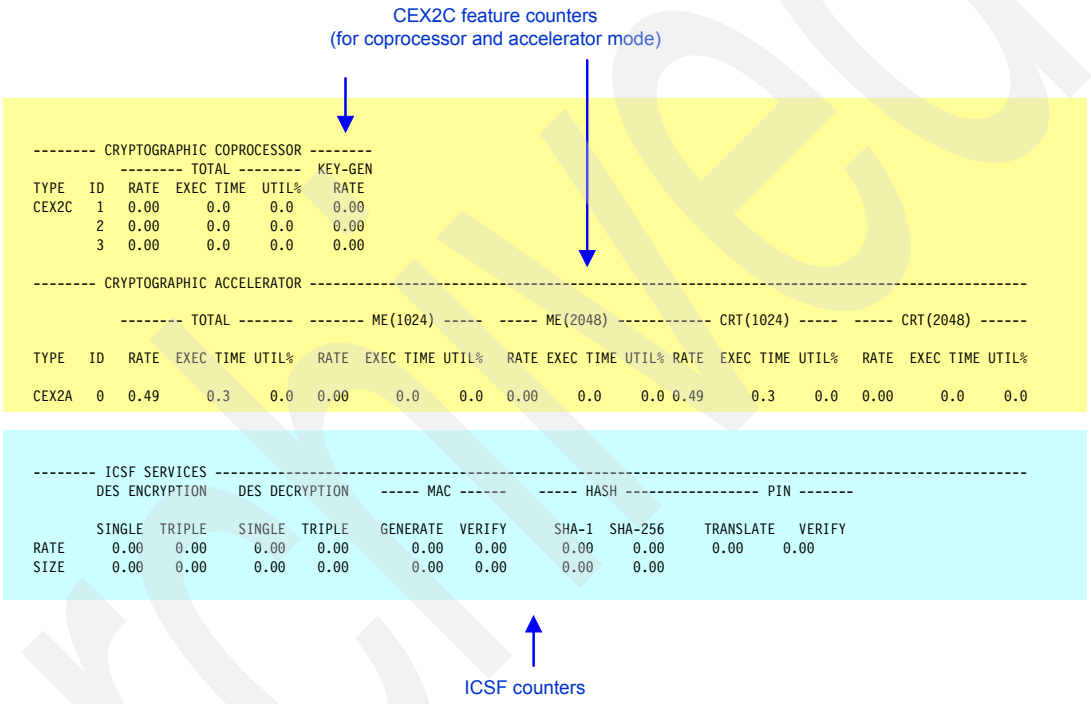


Figure 5-2 Crypto Hardware Activity RMF report.

5.4.1 Cryptographic coprocessor or accelerator utilization

This value indicates how close a system is to exploiting the coprocessor (or implicitly the accelerator) capacity fully in a second of elapsed time. The points to consider are:

- This reports global coprocessor utilization, that is, the value integrates all the requests that the coprocessor receives from any partitions it is accessible from.
- These requests are serialized at the application level; the application instruction flow resumes execution when it receives a response from ICSF. Because there are unavoidable delays in the process flow between the application and the coprocessor, a single flow of requests in the system cannot drive a a coprocessor to 100% utilization as reported by RMF, except in the case of RSA key generation (see in 5.4.3, “Cryptographic coprocessor key generation rate” on page 60).
- Performance degradation related to the elapsed job time for the application might appear before reaching the 100% utilization of the coprocessor.

An increase in the coprocessor utilization ratio can have two effects:

- The cryptographic application execution elapsed time increases as requests compete for service at the coprocessor level.
- A CRYPTO% DLY appears when the application requests arrival rate; at the ICSF level, it cannot accommodate the increased elapsed time to execute the coprocessor requests.

A major difficulty in performing a finer analysis in an actual production environment is the interference from other logical partitions that send requests to the same physical cryptographic coprocessor. It would require a temporary shutdown of these other cryptographic workloads while collecting proper data.

Another difficulty is that most of the obtained results relate to unit of work, and the actual workload that it represents can vary widely with the nature of the operation requested. For instance, the units of work for an RSA operation create a far greater workload in the coprocessor than DES or PIN operations.

Note: There is a limited set of services available in accelerator mode that can be used for SSL or TLS handshakes. Therefore, when we ran the CEX2C in accelerator mode, we used a workload with known and fixed contents. More precise capacity planning assumptions about anticipated accelerated utilization increases can be made this way.

5.4.2 Cryptographic coprocessor or accelerator average execution time

This is an average value of the coprocessor execution time, in elapsed milliseconds, that it takes to serve a request. This, again, is an average that is computed from all requests arriving from all logical partitions that have access to the coprocessor.

This value is expected to vary with the mix of requests that the coprocessor served during the RMF interval.

5.4.3 Cryptographic coprocessor key generation rate

This value indicates how many RSA key generations have been executed on average per second during the RMF period. RSA key generation is a highly computing intensive and very long-duration operation from the coprocessor standpoint. Our measurements showed 100% of coprocessor utilization with a single flow of requests, as reported by RMF. It is likely, therefore, that this operation will have a negative effect on hardware cryptography performance when called concurrently with other cryptographic requests in the same or different logical partitions.

5.4.4 Cryptographic coprocessor or accelerator arrival rate of requests

This information pertains to all requests that are issued by all logical partitions with access to the coprocessor; it shows how many requests arrived on average per second at the coprocessor during the RMF interval. This information can prove useful for detecting changes in the configuration of the cryptographic workloads of the installation and for correlating these changes accordingly with modifications of other indicators, such as the coprocessor utilization or the average execution time.

The correlation of the reported data for the cryptographic accelerator is made easier because of the known and fixed contents of the workload that is processed in accelerator mode.

5.4.5 Cryptographic accelerator key length and format

Based on the data collected from the CEX2C feature, RMF reports what RSA key lengths and formats have been used during the RMF interval. There are two key formats that the cryptographic coprocessor can handle:

- The ME format, which is intended for key lengths that are less than or equal to 1024 bits. This key format is also produced by some z/OS components (for example, RACF) and some cryptographic vendor products for z/OS when it comes to producing keys that can be directly exploited by ICSF.
- The CRT format, which is intended for key lengths equal to or above 1024 bits. The CRT format performs better in terms of required computing resources and speed than the ME format.

The two key formats, as kept in the ICSF key tokens, are illustrated in the appendix of *z/OS ICSF Application Programmer's Guide*, SA22-7522.

Note that the data reported under a key length of 1024 bits in ME format actually pertains to RSA keys with a length that is less or equal to 1024 bits.

5.4.6 The ICSF services arrival rate

This figure indicates how many requests have been received by the local instance of ICSF on average per second for the specific services indicated in the report. This can be used to map the "density" of requests to specific ICSF services that are issued by a specific logical partition and can be helpful for a correlation of detected modifications of other indicators. Remember, however, that not all ICSF services are reported.

5.4.7 The ICSF services data size

This is the average amount of date processed per request to the specific service. From the application standpoint, the length of the block of data that is submitted to ICSF affects the throughput of the cryptographic service. This is demonstrated particularly by Table 5-1 on page 54, where the amount of bytes processed per second greatly varies with the data length and shows the coprocessor throughput increasing with the length of the data blocks.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 64. Note that some of the documents referenced here may be available in softcopy only.

- *S/390 Crypto PCI Implementation Guide, SG24-5942*
- *zSeries Crypto Guide Update, SG24-6870*
- *IBM eServer zSeries 990 (z990) Cryptography Implementation, SG24-7070*

Other publications

These publications are also relevant as further information sources:

- *z/OS ICSF Overview, SA22-7519*
- *z/OS ICSF System Programmer’s Guide, SA22-7520*
- *z/OS Cryptographic Services Integrated Facility Administrator’s Guide, SA22-7521*
- *z/OS ICSF Application programmer’s Guide, SA22-7522*

Online resources

These Web sites are also relevant as further information sources:

- IBM System z9 Performance of Cryptographic Operations (Cryptographic Hardware: CPACF, CEX2C, CEX2A):

  `http://www.ibm.com/servers/eserver/zseries/security/cryptography.html`

- The IBM Technical Sales Library (techdocs):

  `http://www.ibm.com/support/techdocs`

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z

Monitoring System z Cryptographic Services

Redpaper™

System z software tools to monitor hardware cryptography activity

Special focus on z/OS ICSF process flow and RMF reporting

Expert considerations for RMF value interpretation

This IBM Redpaper is intended to provide System z hardware cryptography users with an overview of software tools that they can use to monitor and assess the workload that is being driven to cryptographic coprocessors. It also provides information about IBM published measurements that pertain to the performance of System z hardware cryptography.

A specific chapter is dedicated to the reports that are generated by the z/OS Reporting and Management Facility (RFM) and how you can use and interpret them. RMF is the IBM strategic product for z/OS performance measurement and management. It collects performance data for z/OS base and sysplex environments and issues reports that can be used to monitor system performance so that users can optimally tune and configure their systems to meet business needs.

REDP-4358-00