

## **IBM System z10 Enterprise Class Performance of Cryptographic Operations**

**(Cryptographic Hardware: CPACF, CEX2C, CEX2A)**

© Copyright IBM Corporation 2008  
IBM Corporation  
New Orchard Rd.  
Armonk, NY 10504  
U.S.A.

Produced in the United States of America  
2/08  
All Rights Reserved

IBM, IBM @server, IBM eServer, the IBM logo, the e-business logo, HiperSockets, OS/390, RACF, S/390, z/OS, z/VM, z9 and System z10 Enterprise Class are trademarks or registered trademarks of International Business Machines Corporation of the United States, other countries or both.

Intel is a registered trademark of the Intel Corporation in the United States, other countries or both.

Linux is a registered trademark of Linux Torvalds.

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

IBM may not offer the products, services or features discussed in this document in other all countries in which IBM operates, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

**Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.**

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY, 10504-1785 USA.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Performance is in External Throughput Rate (ETR) based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance rates stated here.

## Table of Content

<b>IBM System z10 Enterprise Class Performance of Cryptographic Operations</b> .....	1
(Cryptographic Hardware: CPACF, CEX2C, CEX2A) .....	1
<b>Preface</b> .....	4
<b>1. Introduction</b> .....	4
<b>2. Cryptographic Hardware supported on System z10 EC</b> .....	5
2.1. CP Assist for Cryptographic Function (CPACF) .....	5
2.2. Crypto Express2 (CEX2) Feature .....	5
<b>3. Exploitation of Cryptographic Hardware on System z10 EC</b> .....	7
3.1. SSL Protocol based Communication .....	7
<b>4. Performance Information</b> .....	9
4.1. Definitions .....	9
4.2. CP Assist for Cryptographic Function (CPACF) .....	9
4.2.1. CP Assist for Cryptographic Function (CPACF) Performance - Architecture Instruction Interface ('Native') .....	9
4.2.2. CP Assist for Cryptographic Function (CPACF) Performance - ICSF API Interface .....	12
4.3. Crypto Express2 Performance - z/OS .....	15
4.3.1. CEX2 Coprocessor Symmetric Key Performance - Encryption/Decryption and MAC Operations .....	15
4.3.2. CEX2 Coprocessor Symmetric Key Performance - Diverse Operations .....	18
4.3.3. CEX2 Coprocessor PKA Performance .....	18
4.3.4 CEX2 Accelerator Performance .....	20
4.4. Crypto Express2 Performance - Linux on System z .....	22
4.4.1. CEX2 Coprocessor Symmetric Key Performance - Encryption/Decryption and MAC Operations .....	22
4.4.2. CEX2 Coprocessor PKA Performance .....	25
4.5. SSL Protocol Handshake Performance .....	27
4.5.1. Applicability of SSL Performance Results to a Customer Environment .....	28
4.5.2. SSL Protocol Performance - System SSL .....	29
with z/OS V1.9 / Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF) .....	29
4.5.3. SSL Protocol Performance - Linux on System z OpenSSL .....	30

## Preface

The performance information presented in this publication was measured on IBM System z10 Enterprise Class (System z10 EC) in an unconstrained environment for the specific benchmark with a system control program (operating system) as specified. Many factors may result in variances between the presented information and the information a customer may obtain by trying to reproduce the data. IBM does not guarantee that your results will correspond to the measurement results herein. This information is provided 'as is' without warranty, express or implied.

The performance numbers stated for some of the operations are only for demonstration purposes. When quoting some key length or cryptographic algorithms one may not conclude that IBM implies the key length or cryptographic algorithm are adequate and can therefore be used safely.

The cryptographic functions described here may not be available in all countries and may require special enablement subject to export regulations.

## 1. Introduction

The purpose of this publication is to provide performance information to the user of cryptographic services on IBM System z10 EC. System z10 EC supports the following cryptographic hardware functions:

1. CP Assist for Cryptographic Function (CPACF).
2. Crypto Express2 (CEX2) feature.

The CP Assist for Cryptographic Function delivers cryptographic support for Data Encryption Standard (DES), Triple DES (TDES), and Advanced Encryption Standard (AES) data encryption/decryption, as well as Secure Hash Algorithm (SHA).

The Crypto Express2 (CEX2) feature combines the functions of Coprocessor (for secure key encrypted transactions) and Accelerator (for Secure Sockets Layer (SSL) acceleration) modes in a single feature with two PCI-X adapters. Using the HMC console, the CEX2 feature can be configured to have either two Coprocessors, two Accelerators or one of each. The Crypto Express2 feature is the same feature which is available on System z9, however installation on System z10 EC provides new microcode which enables additional functions and improved performance for some operations.

## 2. Cryptographic Hardware supported on System z10 EC

### 2.1. CP Assist for Cryptographic Function (CPACF)

System z10 EC supports the Message Security Assist (MSA) Architecture along with the CP Assist for Cryptographic Function (CPACF). The CP Assist for Cryptographic Function delivers cryptographic support for Data Encryption Standard (DES), Triple DES (TDES), and Advanced Encryption Standard (AES) data encryption/decryption, as well as Secure Hash Algorithm (SHA) hashing. System z10 EC has one CPACF for every 2 CPs, therefore, CPACF encryption throughput scales with the number of CPACFs in the system rather than the number of CPs as with z9 systems.

The DES, TDES and AES functions of the CPACF use clear key values. The SHA functions are shipped enabled. The DES, TDES and AES functions require enablement of the CPACF for export control. The CPACF functions for DES, TDES, AES and SHA can be invoked by problem state instructions defined by an extension of the System z architecture. Support is also available via Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF) in z/OS.

The hardware of the CPACF that performs the symmetric key operations (DES; TDES; AES) and SHA functions operates synchronous to the CP operations. The CP cannot perform any other instruction execution while a CPACF cryptographic operation is being executed. The CP internal code performs data fetches and stores resultant data while cryptographic operations are executed in the CPACF hardware on a unit basis as defined by the hardware. The hardware has a fixed set up time per request and a fixed operation speed for the unit of operation. Thus maximum throughput can be achieved for larger blocks of data (up to a hardware defined limit).

### 2.2. Crypto Express2 (CEX2) Feature

The Crypto Express2 (CEX2) feature combines the functions of Coprocessor (for secure key encrypted transactions) and Accelerator (for secure sockets layer SSL acceleration) modes in a single feature with two PCI-X adapters. Using the HMC console, the CEX2 feature can be configured to have either two Coprocessors, two Accelerators or one of each. The Crypto Express2 feature is the same feature which is available on System z9 with updates to provide additional function and improved performance.

There can be a maximum of 8 CEX2 features in a System z10 EC for a total of 16 PCI-X adapters.

When configured in Coprocessor mode, the CEX2 feature supports:

- Secure cryptographic functions
- Use of secure encrypted key values
- Clear key and secure PKA operations
- User defined Extensions (UDX)

The CEX2 in Coprocessor mode provides a security-rich cryptographic subsystem. The tamper-responding hardware is designed to qualify at the highest level under the FIPS 140-2

standard. Specialized hardware performs DES, TDES, RSA, and SHA cryptographic operations in a secure environment. The CEX2 Coprocessor is designed to protect the cryptographic keys and sensitive custom applications. Security relevant cryptographic keys are encrypted under the Master Key when outside the secure boundary of the CEX2 card. The Master Keys are always kept in battery backed-up memory within the tamper-protected secure boundary of the CEX2 Coprocessor.

The CEX2 Coprocessor also supports the 'clear key' PKA operations that currently are predominantly used to provide SSL protocol communications.

When configured in Accelerator mode, the CEX2 feature provides hardware support to accelerate certain cryptographic operations that occur in the e-business environment. Compute intensive public key operations as used by SSL/TLS protocols can be offloaded from the CP to the CEX2 Accelerator and thus increase system throughput. The CEX2 in Accelerator mode works in 'clear key' mode only.

The operations in the CEX2 are controlled by an on-board microprocessor with memory to hold the controlling program. A security-rich code-loading process enables control program and application program loading into the CEX2. The Linux based control program together with the application program provide for the IBM Common Cryptographic Architecture (CCA) interface for applications using the CEX2 feature.

The Crypto Express2 executes its cryptographic operations asynchronously to a Central Processor (CP) operation in the System z10 EC. A CP requesting a cryptographic operation from the CEX2 uses the message queuing protocol to communicate with the CEX2. After enqueueing a request to the CEX2, the host operating system will dispense the task that has enqueue the cryptographic operation and dispatches another task. Thus, processing of the cryptographic operation in the CEX2 will work in parallel to other tasks being executed in a System z10 EC CP. A special CP task will poll at fixed time intervals for finished operations of the Cryptographic Express2, dequeue them, and execute the Release function to cause the redispach of the application waiting for the result of the cryptographic operation. For each PCI-X adapter in the CEX2, up to 8 requests can be waiting in the queue either for execution or waiting with the result of the cryptographic operation to be dequeued by a CP. In the Cryptographic Express2, several operations can be worked on in parallel.

For System z10 EC, the Crypto Express2 works with Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF) and the IBM Resource Access Control Facility (RACF®) in a z/OS or OS/390® operating environment to provide cryptographic services with the IBM Common Cryptographic Architecture (CCA) secure key management.

The IBM Common Cryptographic Architecture implementation provides a base on which customer programs can request cryptographic services from the Crypto Express2. For unique customer cryptographic application requirements the Crypto Express2 in Coprocessor mode provides for user-defined extensions (UDX) to the Common Cryptographic Architecture interface.

### 3. Exploitation of Cryptographic Hardware on System z10 EC

In the cryptographic application environment it is quite common that an application will not have direct access to the cryptographic hardware. The application requiring a cryptographic service will call a Programming Interface (API) which is interpreted by some services of the System Control Program.

In the System z10 EC using the z/OS System Control Program, most cryptographic hardware can only be used through Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF). ICSF is a standard component of z/OS. It provides cryptographic services in the z/OS environment. ICSF provides the application programming interfaces (APIs) by which applications request cryptographic services. Thus ICSF relieves the application from dealing with the complexity of the cryptographic hardware communication. However, these ICSF services are operating software path lengths which have to be added (from an application's point of view) to the execution time of the cryptographic hardware.

As mentioned in the description of the CPACF cryptographic hardware, an application program can use this hardware by invoking the MSA machine instructions. However, there is also an API provided by ICSF. The performance of both modes of operation will be presented in this publication.

When running Linux for System z Control Program, access to the System z10 EC cryptographic hardware is provided by the `xcryptolinz rpm` which can be downloaded from [www.ibm.com/security/cryptocards/pcixcc/ordersoftware.shtml](http://www.ibm.com/security/cryptocards/pcixcc/ordersoftware.shtml)

The `xcryptolinz rpm` provides the APIs by which applications request cryptographic services, relieving the application from dealing with the complexity of the cryptographic hardware communication.

#### 3.1. SSL Protocol based Communication

Secure Sockets Layer (SSL) is a communication protocol that was designed to facilitate secure communication over an open communication network, such as the Internet. The SSL protocol is a layered protocol that is intended to be used on top of a reliable transport, e.g. Transmission Control Protocol (TCP/IP). SSL is designed to provide data privacy and integrity by using cryptographic operations and optionally Server and Client authentication based on public key certificates. Once an SSL connection is established between a Client and Server, data communications between Client and Server are transparent to the encryption and integrity added by the SSL protocol. Transport Layer Security (TLS) is the newer version of the SSL protocol.

Executing the SSL/TLS protocols for a Server (or Client) on a System z10 EC will result in a series of cryptographic operations. In the z/OS environment, SSL will either invoke the available cryptographic hardware directly (via the MSA instructions), or use the hardware via ICSF (for the PKA operations) or use its own software routines to perform the crypto function. The SSL/TLS protocol will result in an increase in transaction execution time compared to an unsecure protocol. Some factors contributing to the increase are 1)CP path length (due to the protocol itself and due to operating system support); 2) the symmetric key operation's execution time (either hardware assisted or in software executed on a CP); and 3) the execution time of the

public key operations (either hardware assisted (operating in parallel to the CP instruction execution) or in software on a CP). This publication will state the performance in the SSL environment as the maximum number of SSL handshakes the System z10 EC can provide as a server within the given system constraints and assess the utilization of the measured system.

The intent for providing capacity information in the SSL environment is to demonstrate the capabilities of a System z10 EC to act as a Web Server providing SSL-compliant communication to a large number of clients. For this purpose the maximum number of SSL connects and data exchanges per second made between the server and all clients are provided for different environments. There is no intention to provide a more detailed performance analysis for this environment.

In this publication, performance/capacity information will be given for running SSL protocol based communication in the following environments:

- z/OS
- Linux

As this performance publication primarily deals with performance of cryptographic operations and Web based communication, the measurements for the SSL environments include only the processing required for the SSL protocol handshake and some data exchange. Explicitly excluded is the processing for the 'business transaction' that in a normal environment would be initiated in the server on behalf of the client's request. As most SSL protocol-based measurements in this report are limited by the processing capacity of the server, in a 'real life' environment the processing for the business transaction would reduce the number of necessary handshakes considerably.

## 4. Performance Information

### 4.1. Definitions

z/OS performance information stated in this publication is normally provided on the ICSF API level except when stated otherwise. Measurements were performed with the control program z/OS Version 1 Release 9 (z/OS V1.9) and Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF), except when stated otherwise.

Linux for System z information stated in this publication was provided on the xcryptolinz API level. Measurements were performed with SLES 10 SP0 and xcryptolinz Version 3.28.

All measurements were performed on an IBM System z10 EC Model E26. Most of the measurements were run with 4 dedicated Central Processors assigned to the LPAR. If, however, the measurement invokes only one single job or thread the performance behavior is the same as if this measurement were run on a System z10 EC Model E26 with only one dedicated CP.

For the cryptographic operations that can be used with a variable length of data such as Data Encryption Algorithm (DEA) Standard encryption, the performance is stated for test cases using different data lengths. The length is specified in Bytes ('K' equals 1024, 'M' equals 1,048,576). The resulting data rate is specified in multiples of 1,000,000 Bytes (not 'M').

In order to keep this performance publication at a reasonable length results of measurements are generally presented using a single cryptographic feature. In some cases, a statement is made how the performance results scale with usage of multiple features.

### 4.2. CP Assist for Cryptographic Function (CPACF)

#### 4.2.1. CP Assist for Cryptographic Function (CPACF) Performance - Architecture Instruction Interface ('Native')

All test cases are written in System z Assembler Language issuing the System z Message Security Assist (MSA) Architecture cryptographic operation instructions as indicated with each group.

The data quoted was from test cases run on a System z10 EC Model E26, however, using only one of the CPACFs. For each cryptographic operation type quoted, there is a statement on scalability of the results if up to 3 CPACFs are being used. The throughput using N CPACFs performing the same cryptographic operation is close to N times the throughput of using one CPACF. The reduction of the measured throughput from N times the throughput of one CPACF is stated with each measurement.

Terminology Explanation: The term DEA stands for Data Encryption Algorithm which is a block cipher according to the Data Encryption Standard (DES).

**DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits)**

(System z Message Security Assist Architecture instruction: KMC-DEA)

Native: Single DES CBC Encipher (KMC-DEA)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	6242115	399.4
256	2399886	614.3
1024	730227	747.7
4096	189736	777.1
64K	11850	776.6
1M	738.4	774.3

The KMC-DEA operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 1 percent with 3 CPACFs.

DEA Cipher Block Chaining Decipher with Single Length Key (not shown) has similar performance characteristics as the Encipher operation.

**DEA Cipher Block Chaining Encipher with Triple Length Key (168 Bits)**

(System z Message Security Assist Architecture instruction: KMC-TDEA)

Native: Triple DES CBC Encipher (KMC-TDEA)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	3473534	222.3
256	1117487	286.0
1024	296350	303.4
4096	75451	309.0
64K	4711	308.7
1M	294.0	308.3

The KMC-TDEA operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 1 percent (with 3 CPACFs).

DEA Cipher Block Chaining Decipher with Triple Length Key has similar performance characteristics as the Encipher operation.

**AES Cipher Block Chaining Encipher with 128 Bit Key**

(System z Message Security Assist Architecture instruction: KMC-AES)

Native: AES - 128 bit CBC Encipher (KMC-AES)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	6963038	445.6
256	2902959	743.1
1024	946020	968.7
4096	250207	1024.8
64K	15617	1023.5
1M	973.4	1020.7

The KMC-AES operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 1 percent (with 3 CPACFs).

**AES Cipher Block Chaining Encipher with 256 Bit Key**

(System z Message Security Assist Architecture instruction: KMC-AES)

Native: AES - 256 bit CBC Encipher (KMC-AES)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	6368283	407.5
256	2489758	637.3
1024	770171	788.6
4096	202448	829.2
64K	12645	828.7
1M	788.1	826.4

The KMC-AES operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 1 percent (with 3 CPACFs).

**Compute Message Authentication Code with DEA Single Length Key (56 Bits)**

(System z Message Security Assist Architecture instruction: KMAC-DEA)

Native: MAC with single DES (KMAC-DEA)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	6593857	422.0
256	2540661	650.4
1024	730843	748.3
4096	190516	780.3
64K	11923	781.4
1M	743.3	779.4

The KMAC-DEA operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 1 percent (with 3 CPACFs).

**Compute Message Digest SHA-1**

(System z Message Security Assist Architecture instruction: KLMD-SHA-1)

Native: SHA-1(KLMD-SHA-1)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	3903535	249.8
256	1997169	511.2
1024	648729	664.2
4096	178194	729.8
64K	11351	743.9
1M	708.4	742.8

The KLMD-SHA-1 operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 1 percent (with 3 CPACFs).

**Compute Message Digest SHA-512**

(System z Message Security Assist Architecture instruction: KLMD-SHA-512)

Native: SHA-512(KLMD-SHA-512)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	3861916	247.1
256	1744084	446.4
1024	632678	647.8
4096	181648	744.0
64K	11733	768.9
1M	733.4	769.1

The KLMD-SHA-512 operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 1 percent (with 3 CPACFs).

#### 4.2.2. CP Assist for Cryptographic Function (CPACF) Performance - ICSF API Interface

All test cases are written in System z Assembler Language issuing an API call to ICSF for the cryptographic operation. ICSF will resolve the API call and issue instructions for the cryptographic operation according to the System z Message Security Assist (MSA) Architecture as indicated with each group.

The data quoted is from test cases run on a z10 EC Model E26, however, using only one of the CPACFs. For each cryptographic operation type quoted there is a statement on scalability of the results if multiple CPACFs are being used. The throughput using N CPACFs performing the same cryptographic operation is close to N times the throughput of using one CPACF. The reduction of the measured throughput from N times the throughput of one CPACF is stated with each measurement.

As the performance measurement results show, all ICSF API interface test cases have lower throughput than the equivalent 'Native' test cases. This is expected because of the additional ICSF path length. As the data length increases, the ICSF path length is a less dominant factor and the throughput is nearly the same as for the 'Native' test cases for large data lengths.

**DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits) - ICSF API**

(System z Message Security Assist Architecture instruction: KMC-DEA)

ICSF API: Single DES CBC Encipher (KMC-DEA) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	267454	17.1
256	250436	64.1
1024	200014	204.8
4096	112655	461.4
64K	11176	732.4
1M	725.1	760.3

The DEA Encipher with Single Length Key operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 16 % for 3 CPACFs and 64 byte data length and decreases to 0% for 1MB data length.

DEA Decipher with Single Length Key has similar performance characteristics as the Encipher operation.

**DEA Cipher Block Chaining Encipher with Triple Length Key (168 Bits) - ICSF API**  
(System z Message Security Assist Architecture instruction: KMC-TDEA)

ICSF API: Triple DES CBC Encipher (KMC-TDEA) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	258448	16.5
256	221448	56.6
1024	142959	146.3
4096	58984	241.6
64K	4592	300.9
1M	290.8	304.9

The DEA Encipher with Triple Length Key operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 15 % for 3 CPACFs and 64 byte data length and decreases to 0% for 1MB data length.

DEA Decipher with Triple Length Key has similar performance characteristics as the Encipher operation.

**AES Cipher Block Chaining Encipher with 128 Bit Key - ICSF API**  
(System z Message Security Assist Architecture instruction: KMC-AES)

ICSF API: AES128 Encipher (128 bit key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	261133	16.7
256	249028	63.7
1024	210900	215.9
4096	130389	534.0
64K	14457	947.5
1M	951.6	997.8

The AES Encipher with 128 bit key operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 16 % for 3 CPACFs and 64 byte data length and decreases to 0% for 1MB data length.

AES Decipher with 128 bit key has similar performance characteristics as the Encipher operation.

### AES Cipher Block Chaining Encipher with 256 Bit Key - ICSF API

(System z Message Security Assist Architecture instruction: KMC-AES)

AES256 Encipher (256 bit key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	261779	16.7
256	246237	63.0
1024	202166	207.0
4096	116082	475.4
64K	11864	777.5
1M	772.7	810.3

The AES Encipher with 256 bit key operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 17 % for 3 CPACFs and 64 byte data length and decreases to 0% for 1MB data length.

AES Decipher with 256 bit key has similar performance characteristics as the Encipher operation.

### Compute Message Digest SHA-1 - ICSF API

(System z Message Security Assist Architecture instruction: KLMD-SHA-1)

ICSF API: SHA-1(KLMD-SHA-1) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	197610	12.6
256	187954	48.1
1024	156813	160.5
4096	95408	390.7
64K	10640	697.3
1M	697.9	731.8

The Compute message Digest SHA-1 operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 15 % for 3 CPACFs and 64 byte data length and decreases to 0% for 1MB data length.

### Compute Message Digest SHA-512 - ICSF API

(System z Message Security Assist Architecture instruction: KLMD-SHA-5)

ICSF API: SHA-512(KLMD-SHA-512)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	197079	12.6
256	185294	47.4
1024	155649	159.3
4096	96169	393.9
64K	10962	718.4
1M	720.1	755.0

The Compute message Digest SHA-512 operation scales with the number of CPACFs executing multiple jobs with the same operation. The reduction is less than 14 % for 3 CPACFs and 64 byte data length and decreases to 0% for 1MB data length.

### **4.3. Crypto Express2 Performance - z/OS**

The Crypto Express2 feature is designed to satisfy high-end server security requirements. The Crypto Express2 feature, with two PCI-X adapters, is configurable and can be defined for secure key encrypted transactions (Coprocessor – the default) or SSL acceleration (Accelerator). Crypto Express2 executes the functions that were previously offered by the PCICA and PCIXCC features, performing hardware acceleration for SSL transactions and clear key RSA operations. Like its predecessors, the Crypto Express2 feature has been designed to satisfy the security requirements of an enterprise server. The PCIXCC, PCICC, and PCICA features are not supported on System z10 EC.

When configured as a Coprocessor, the PCI-X adapter is designed to provide security-rich cryptographic operations to be used by System z10 EC host application programs. The Coprocessor mode offers security for symmetric key and public key operations. In this case the cryptographic keys are encrypted under the corresponding Master Keys when outside the secure boundary of the PCI-X adapter.

When configured as an Accelerator, the PCI-X adapter is designed to provide high speed acceleration of RSA operations in ‘clear key’ mode, providing security rich communication for Web site-based applications which utilize the SSL or TLS protocol. It is current practice to execute the public key operation, incurred during set up of an SSL session, in ‘clear key’ mode.

The connection of the CEX2 feature via the PCIX bus to the System z10 EC Central Processors (CPs) incurs latency and data transmission time. Because of this connection to the System z10 EC CPs, the CEX2 operates asynchronous to the System z10 EC CPs.

There can be a maximum of 8 CEX2 features in a System z10 EC, each CEX2 feature containing two PCI-X adapters.

#### **4.3.1. CEX2 Coprocessor Symmetric Key Performance - Encryption/Decryption and MAC Operations**

This chapter deals with CEX2 Coprocessor cryptographic operations with a user supplied length of data as, e.g., DES operations.

All test cases are written in System z Assembler Language issuing an API call to ICSF for the cryptographic operation. ICSF will resolve the API call and handle the communication with the CEX2 Coprocessor feature which does the actual cryptographic processing. The symmetric key that is used for the cryptographic operation is encrypted under the corresponding Master Key which in turn is kept in the secure boundary of the PCI-X adapter.

The throughput for symmetric key operations using the CEX2 Coprocessor is considerably less than the throughput for the corresponding operations using the CP Assist for Cryptographic Function (CPACF) hardware. For this type of cryptographic operation the CEX2 Coprocessor feature should be used only when the security requirements for the application require it. Be

aware that in the tables of this chapter the rates are quoted in thousands of Bytes, not in millions of bytes as in previous tables.

The data quoted was from test cases run on a System z10 EC Model E26 using 1 job that performs the cryptographic operation. For each cryptographic operation type quoted there is a statement on scalability of the results if multiple jobs are being used. The increase of measured throughput using 7 jobs is exemplified for the Single DES CBC Encipher operation.

The performance numbers are from measurements with z/OS V1.9 including Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF).

### **CEX2 Coprocessor DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits)**

CEX2C (one job): Single DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	941.9	60.28
256	942.0	241.1
1024	683.8	700.2
4096	632.1	2589.4
64K	87.9	5760.9
1M	5.94	6233.4

The above table provides measurement results for an environment where one job was continuously executing the cryptographic operation using one CEX2 Coprocessor card. As mentioned, the execution of the cryptographic operation in the CEX2C card is asynchronous to the System z10 EC Central Processor (CP) execution. As only one job is run on the CP the next cryptographic operation is started only when the result of the previous cryptographic operation has been received by the CP. Thus there is a considerable delay before the next cryptographic operation can be initiated by the host CP. This inefficiency is removed when the host program consists of several jobs requesting cryptographic operations at the same time. The CEX2C adapter's multitasking capability allows for enqueueing and dequeuing of requests in parallel with cryptographic operations being performed. A measurement environment using several parallel jobs highlights better the throughput capacity of the CEX2C adapter whereas the 'single job' measurement environment is better suited to highlight the delay an application experiences waiting for the result of the cryptographic operation performed in the CEX2C.

CEX2C (seven jobs): Single DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	1539	98.51
256	1428	365.5
1024	1148	1176.4
4096	1129	4628.4
64K	123.0	8063.5
1M	8.07	8457.5

The throughput with N CEX2C adapters with a sufficient number of jobs repetitively requesting the same cryptographic operation for Single DES, Triple DES, and Single DES Message Authentication (MAC) (see the following tables) is close to N times the throughput of one CEX2C adapter with 7 jobs (as exemplified above).

### CEX2 Coprocessor DEA Cipher Block Chaining Encipher with Triple Length Key (168 Bits)

CEX2C (one job): Triple DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	941.9	60.28
256	941.8	241.1
1024	633.0	648.2
4096	632.2	2589.6
64K	69.72	4569.5
1M	4.57	4794.1

The throughput for seven jobs for CEX2C TDES is on the order of 1.3 times to 1.6 times higher than for one job.

### CEX2 Coprocessor Message Authentication Code with DEA Single Length Key (56 Bits)

CEX2C (one job): MAC with single DES		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	942.2	60.30
256	942.2	241.2
1024	942.8	965.5
4096	926.0	3793.2
64K	88.48	5798.7
1M	5.99	6283.4

The throughput for seven jobs for CEX2C MAC is on the order of 1.3 to 1.6 times higher than for one job, the lower number applying to large data lengths and the higher to small data lengths.

### 4.3.2. CEX2 Coprocessor Symmetric Key Performance - Diverse Operations

The following table gives the performance in maximum number of operations per second for one CEX2 Coprocessor for some selected symmetric key operations.

CEX2C Symmetric Key Operations - Examples	Ops/s	Ops/s
	1 job	7 jobs
Key Generate (operational DES KEYGENKY key)	633	1,063
Clear PIN Generate Alternate (DES OPINENC + DES PINGEN keys)	633	1,033
Clear PIN Generate (16 digits) ( DES PINGEN key)	942	1,572
Encrypted PIN Translation (DES IPINENC key + DES OPINENC key)	941	1,191
Encrypted PIN Translation (2 UKPT enabled KEYGENKY keys)	319	360
Encryp.PIN Verificat. (UKPT enabl.KEYGENKY+DES PINVER keys)	476	532

The throughput with N CEX2C adapters with a sufficient number of jobs repetitively requesting the same cryptographic operation for the examples in the table above is close to N times the throughput of one CEX2C adapter with 7 jobs.

### 4.3.3. CEX2 Coprocessor PKA Performance

The CEX2 Coprocessor is designed to offer good Public Key Algorithm (PKA) cryptographic operation performance in addition to the high-security environment. The PKA performance is listed for RSA key modulus length of 512 bits, 1024 bits (1K bits), 2048 bits (2K bits) and 4096 bits (4K bits).

The numbers quoted for performing the Public Key Decrypt (PKD) cryptographic operation (using the Private Exponent) are either through the Chinese Remainder Theorem (CRT) Format or the Modulus Exponent (ME) Format. The PKD operation uses the private key in 'clear key' mode.

For the Public Key Encrypt (PKE) cryptographic operation ICSF always uses an RSA public key with the Modulus Exponent (ME) Format. The modulus is according to the length specified and the (Public) Exponent has the value of 65537 which in hexadecimal notation is X'10001' (with leading zeros up to the length of the modulus).

For the Digital Signature Generate (DSG) and the Symmetric Key Import (SYI) cryptographic operations the PKA keys (signature key or encryption key) are encrypted under the corresponding master key.

The performance numbers are from measurements with z/OS V1.9 including Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF) invoking the operation via the ICSF API according to the PKCS-1.2. Standard. Measurements were performed on a System z10 EC Model E26.

**CEX2C Coprocessor PKA Performance**

<b>CEX2C on z/OS V1.9 (ICSF level: HCR7750)</b>				
Public Key Decrypt (PKD), Public Key Encrypt (PKE)				
Digital Signature Generate (DSG), Digital Sign. Verify (DSV)				
Symmetric Key Import (encrypted with RSA key) (SYI)				
	2097 E26	2097 E26	2097 E26	2097 E26
CEX2C	1	1	2	4
Jobs	1	7	14	28
	Operations/sec	Operations/sec	Operations/sec	Operations/sec
PKD--CRT, 1024 bit	631	1130	2261	4525
PKD--CRT, 2048 bit	273	466	932	1861
PKD--CRT, 4096 bit	41	44	88	175
PKD--ME, 512 bit	631	1248	2494	4994
PKD--ME, 1024 bit	472	933	1865	3725
PKE, 512 bit	938	1379	2769	5554
PKE, 1024 bit	899	1190	2384	4762
PKE, 2048 bit	631	918	1835	3665
PKE, 4096 bit	8	8	17	34
DSG--CRT, 1024 bit	631	1124	2253	4502
DSG--CRT, 2048 bit	273	466	932	1861
DSG--CRT, 4096 bit	45	49	97	194
DSV--CRT, 1024 bit	939	1555	3115	6231
DSV--CRT, 2048 bit	856	1471	2946	5895
DSV--CRT, 4096 bit	8	8	17	34
SYI--CRT, 512 bit	631	904	1814	3625
SYI--CRT, 1024 bit	476	857	1720	3438
SYI--CRT, 4096 bit	41	44	88	176

The PKA cryptographic operation throughput with N CEX2C adapters with a sufficient number of jobs repetitively requesting the same cryptographic operation for the examples in the table above is close to N times the throughput of one CEX2C adapter with 7 jobs (as stated above).

## PKA RSA Key Generate

The CEX2 Coprocessor also offers services to generate PKA RSA Keys. The PKA RSA Key Generate performance is listed for RSA key modulus length of 512 bits, 1024 bits (1K bits), 2048 bits (2K bits) and 4096 bits (4K bits) dependent on the format, either the Chinese Remainder Theorem (CRT) Format or the Modulus Exponent (ME) Format.

PKA Key Generation is a compute intensive operation. The table below specifies the number of Key generations per second provided by one CEX2 Coprocessor.

### CEX2 Coprocessor PKA RSA Key Generation Performance

<b>CEX2C PKA RSA Key Generate</b>	
CEX2C PKA RSA Key Generate	Operations/sec
External CRT, 512bit	3.95
External CRT, 1024bit	1.94
External CRT, 2048bit	0.91
External CRT, 4096bit	0.22
Internal ME, 512bit	4.78
Internal ME, 1024bit	2.57

## 4.3.4 CEX2 Accelerator Performance

The CEX2 Accelerator configuration mode is designed to offer fast Public Key Algorithm cryptographic (PKA) operations. The PKA performance is listed for RSA key modulus length of 512 bits, 1024 bits (1K bits) and 2048 bits (2K bits). The performance numbers are from measurements with z/OS V1.9 including Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF) invoking the operation via the ICSF API according to the PKCS-1.2 Standard.

Quoted are the numbers performing the Public Key Decrypt (PKD) cryptographic operation which uses the Private Exponent either through the Chinese Remainder Theorem (CRT) Format or the Modulus Exponent (ME) Format.

For the Public Key Encrypt (PKE) cryptographic operation ICSF always uses an RSA public key with the Modulus Exponent (ME) Format. The modulus is according to the length specified and the (Public) Exponent has the value of 65537 which in hexadecimal notation is X'10001' (with leading zeros up to the length of the modulus)

**CEX2 Accelerator PKA Performance**

<b>CEX2A Public Key Decrypt (PKD) and Public Key Encrypt (PKE) (z/OS V1.9, ICSF HCR7750)</b>			
2097 CPs	4	4	4
CEX2A Adapters	1	1	4
Jobs	1	8	32
	Operations/sec	Operations/sec	Operations/sec
PKD-CRT, 512 bit	1841	8359	33111
PKD--CRT, 1024 bit	1841	3334	13302
PKD--CRT, 2048 bit	382	456	1821
PKD--ME, 512 bit	1842	3370	13439
PKD--ME, 1024 bit	633	920	3672
PKE, 512 bit	1851	13255	43254
PKE, 1024 bit	1851	13452	44384
PKE, 2048 bit	1850	9959	37065

The first result column of the above table is for measurements where one job was continuously executing the cryptographic operation using one CEX2 Accelerator card. As mentioned, the execution of the cryptographic operation in the CEX2 Accelerator is asynchronous to the System z10 EC Central Processor (CP) execution. As only one job is run on the CP the next cryptographic operation is started only when the result of the previous cryptographic operation has been received by the CP. The single job measurement indicates the delay an application would experience waiting for the result of the cryptographic operation.

The second result column of the above table is for measurements where eight jobs were continuously executing the same cryptographic operation using one CEX2 Accelerator card. The increased throughput is due to the fact that tasks are always available for execution in the CEX2 Accelerator card due to the parallel threads that run in the System z10 EC CPs. Thus the full capability of the CEX2 Accelerator card for parallel execution of the cryptographic operation can be utilized.

The third column of the above table is for measurements where 32 jobs were continuously executing the same cryptographic operation using 4 CEX2 Accelerator cards. The results show the scalability of the throughput when multiple CEX2A adapters are used in one System z10 EC.

## 4.4. Crypto Express2 Performance - Linux on System z

The Crypto Express2 feature is also supported by Linux on System z. Support is provided via the xcryptolinz rpm which can be downloaded from [www.ibm.com/security/cryptocards/pcixcc/ordersoftware.shtml](http://www.ibm.com/security/cryptocards/pcixcc/ordersoftware.shtml)

The zcrypt device driver is included in Novell SUSE Linux Enterprise Server 10 Service Pack 1 (SLES10 SP1) and RedHat Enterprise Linux 5.2 (RHEL 5.2).

The device driver can be loaded in either of two modes : Polling On (default) or Polling Off. This document presents data for both modes of operation.

In Polling On mode, the zcrypt device driver queues the requested cryptographic operation on the CEX2 Coprocessor and then dispatches a polling thread with the minimum scheduling priority. If there is an otherwise idle CP it will run the polling thread which goes in to a spin-wait loop waiting for the result. The spin-wait loop has microsecond granularity, allowing the result of the operation to be polled out close to when it completes, resulting in higher single threaded throughput than Polling Off mode. However, the otherwise idle CP on which the spin-wait is running is causing higher CP Utilization when compared to Polling Off mode.

In Polling Off mode, the zcrypt device driver queues the requested cryptographic operation on the CEX2 Coprocessor and checks for the result every 10 milliseconds (ms) until it is returned. The polling interval of 10 ms limits the throughput of a single threaded application to a maximum of 100 operations per second. Higher throughput can be achieved with multi-threaded applications. The advantage of Polling Off mode is that it does not use additional CP cycles while waiting for the result of the crypto operation.

Polling mode is controlled by a 'poll\_thread' option when loading the zcrypt module. The default of poll\_thread=1 will result in Polling On mode. Setting poll\_thread=0 will result in Polling Off mode. (example: modprobe zcrypt poll\_thread=0)

For all Linux on System z CEX2 data presented here the following applies:

- Linux System Level: SLES10 SP0
- Linux Kernel Level: 2.6.16
- xcryptolinz Level: 3.28-rc05

### 4.4.1. CEX2 Coprocessor Symmetric Key Performance - Encryption/Decryption and MAC Operations

This chapter deals with CEX2 Coprocessor cryptographic operations with a user supplied length of data as, e.g., DES operations.

The test case issues an API call to the xcryptolinz device driver for the cryptographic operation. The xcryptolinz device driver will resolve the API call and handle the communication with the CEX2 Coprocessor feature which does the actual cryptographic processing. The symmetric key that is used for the cryptographic operation is encrypted under the corresponding Master Key which in turn is kept in the secure boundary of the PCI-X adapter.

The throughput for symmetric key operations using the CEX2 Coprocessor is considerably less than the throughput for corresponding operations using the CP Assist for Cryptographic Function (CPACF) hardware. For this type of cryptographic operation the CEX2 Coprocessor feature should be used only when the security requirements for the application require it. Be aware that in the tables of this chapter the rates are quoted in thousands of Bytes, not in millions of bytes as in Chapter 4.2.

The data quoted was from test cases run on a System z10 EC Model E26 using 1 job that performs the cryptographic operation. For each cryptographic operation type quoted there is a statement on scalability of the results if multiple jobs are being used. The increase of measured throughput using 9 jobs is exemplified for the Single DES CBC Encipher operation.

### **CEX2 Coprocessor DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits)**

CEX2C (one thread): Single DES CBC Encipher					
Data Length (Bytes)	Polling Off		Polling On		
	Operations/sec	x10**3 Bytes/sec	Operations/sec	x10**3 Bytes/sec	
64	99.98	6.40	1190.10	76.17	
256	99.98	25.60	1129.08	289.04	
1024	99.98	102.38	924.68	946.87	
4096	99.98	409.53	866.86	3550.66	
64K	14.28	936.05	93.49	6126.90	
1M	1.00	1048.58	6.28	6587.15	

The above table provides measurement results for an environment where one job was continuously executing the cryptographic operation using one CEX2 Coprocessor card. As mentioned, the execution of the cryptographic operation in the CEX2C card is asynchronous to the System z10 EC Central Processor (CP) execution. As only one job is run on the CP the next cryptographic operation is started only when the result of the previous cryptographic operation has been received by the CP. Thus there is a considerable delay before the next cryptographic operation can be initiated by the host CP. This inefficiency is removed when the host program consists of several jobs requesting cryptographic operations at the same time. The CEX2C adapter's multitasking capability allows for enqueueing and dequeuing of requests in parallel with cryptographic operations being performed. A measurement environment using several parallel jobs highlights better the throughput capacity of the CEX2C adapter whereas the 'single job' measurement environment is better suited to highlight the delay an application experiences waiting for the result of the cryptographic operation performed in the CEX2C.

CEX2C (nine threads): Single DES CBC Encipher					
Data Length (Bytes)	Polling Off		Polling On		x10**3 Bytes/sec
	Operations/sec	x10**3 Bytes/sec	Operations/sec	x10**3 Bytes/sec	
64	799.73	51.18	1569.03	100.42	
256	799.73	204.73	1468.52	375.94	
1024	799.73	818.93	1165.74	1193.72	
4096	799.73	3275.71	1148.90	4705.88	
64K	111.52	7308.57	123.91	8120.30	
1M	7.58	7952.40	8.34	8748.27	

The throughput with N CEX2C adapters with a sufficient number of threads repetitively requesting the same cryptographic operation for Single DES, Triple DES, and Single DES Message Authentication (MAC) (see the following tables) is close to N times the throughput of one CEX2C adapter with 9 threads (as exemplified above).

### CEX2 Coprocessor DEA Cipher Block Chaining Encipher with Double Length Key (112 Bits)

CEX2C (one thread): Triple DES CBC Encipher with Double Length Key					
Data Length (Bytes)	Polling Off		Polling On		x10**3 Bytes/sec
	Operations/sec	x10**3 Bytes/sec	Operations/sec	x10**3 Bytes/sec	
64	99.98	6.40	1136.37	72.73	
256	99.98	25.60	1079.65	276.39	
1024	99.98	102.38	891.30	912.69	
4096	99.98	409.53	837.82	3431.69	
64K	14.28	936.05	90.68	5942.48	
1M	1.00	1048.58	6.10	6392.12	

With polling off, the throughput for nine threads for CEX2C TDES is approximately 8 times higher than for one thread. With polling on, the throughput for nine threads is on the order of 1.2 times higher than for 1 thread.

### CEX2 Coprocessor Message Authentication Code with DEA Single Length Key (56 Bits)

CEX2C (one thread): MAC with Single DES					
Data Length (Bytes)	Polling Off		Polling On		x10**3 Bytes/sec
	Operations/sec	x10**3 Bytes/sec	Operations/sec	x10**3 Bytes/sec	
64	99.98	6.40	1290.65	82.60	
256	99.98	25.60	1249.03	319.75	
1024	99.98	102.38	1099.83	1126.22	
4096	99.98	409.53	1152.14	4719.15	
64K	14.28	936.05	102.60	6723.73	
1M	1.00	1048.58	6.78	7106.20	

With polling off, the throughput for nine threads for CEX2C MAC is approximately 8 times higher than for one thread. With polling on, the throughput for nine threads is on the order of 1.2 times higher than for 1 thread.

## 4.4.2. CEX2 Coprocessor PKA Performance

The CEX2 Coprocessor is designed to offer good Public Key Algorithm (PKA) cryptographic operation performance in addition to the high-security environment. The PKA performance is listed for RSA key modulus length of 512 bits, 1024 bits (1K bits) and 2048 bits (2K bits).

The numbers quoted for performing the Public Key Decrypt (PKD) cryptographic operation (using the Private Exponent) are either through the Chinese Remainder Theorem (CRT) Format or the Modulus Exponent (ME) Format. The PKD operation uses the private key in 'clear key' mode.

For the Digital Signature Generate (DSG) cryptographic operations the PKA keys (signature key or encryption key) are encrypted under the corresponding master key.

### CEX2 Coprocessor PKA Performance

<b>CEX2C on SLES 10 SP 0 with Polling On</b>				
Public Key Decrypt (PKD), Public Key Encrypt (PKE)				
Digital Signature Generate (DSG), Digital Sign. Verify (DSV)				
	2097 E26	2097 E26	2097 E26	2097 E26
CEX2C	1	1	2	4
Threads	1	9	18	36
	Operations/sec	Operations/sec	Operations/sec	Operations/sec
PKD--CRT, 1024 bit	791	1258	2524	5043
PKD--CRT, 2048 bit	307	464	925	1833
PKD--ME, 512 bit	859	1398	2801	5620
PKD--ME, 1024 bit	509	937	1877	3741
PKE--CRT, 1024 bit	1035	1287	2580	5154
PKE--CRT, 2048 bit	900	1136	2274	4538
PKE--ME, 512 bit	1096	1365	2729	5459
PKE--ME, 1024 bit	1062	1324	2647	5292
DSG--CRT, 1024 bit	781	1207	2408	4820
DSG--CRT, 2048 bit	306	460	922	1858
DSV--ME, 512 bit	1113	1391	2781	5567
DSV--ME, 1024 bit	1073	1334	2678	5355

The PKA cryptographic operation throughput with N CEX2C adapters with a sufficient number of threads repetitively requesting the same cryptographic operation for the examples in the table above is close to N times the throughput of one CEX2C adapter with 9 threads (as stated above).

### PKA RSA Key Generate

The CEX2 Coprocessor also offers services to generate PKA RSA Keys. The PKA RSA Key Generate performance is listed for RSA key modulus length of 512 bits, 1024 bits (1K bits), and 2048 bits (2K bits) dependent on the format, either the Chinese Remainder Theorem (CRT) Format or the Modulus Exponent (ME) Format.

PKA Key Generation is a compute intensive operation. The table below specifies the number of key generations per second provided by one thread and one CEX2 Coprocessor with polling off. However, due to the length of the key generation operation, turning on the polling thread or increasing the number of threads requesting the same operation does not increase the throughput achievable with one CEX2 Coprocessor.

### CEX2 Coprocessor PKA RSA Key Generation Performance

<b>CEX2C PKA RSA Key Generate with Polling Off</b>		
CEX2C PKA RSA Key Generate		
		Operations/sec
CRT-CLEAR, 512bit		3.968
CRT-CLEAR, 1024bit		1.922
CRT-CLEAR, 2048bit		0.841
ME-CLEAR, 512bit		3.316
ME-CLEAR, 1024bit		1.588

## 4.5. SSL Protocol Handshake Performance

The SSL handshake protocol is used to negotiate the secure attributes of a session between Client and Server. This process establishes Protocol Version, Session Identification (SID), Authentication (authentication of the Client is optional), and a symmetric key to help protect the data transmitted between Server and Client. The attributes of an established session can be kept as Session Identification in a Client and/or Server cache for later reuse. This may be of interest as establishing a session is a compute intensive process and requires on the Server side a PKA Private Key operation. This Public Key Decrypt (PKD) on the Server can be performed either in software or may be assisted by cryptographic hardware. In the presented measurements on the System z10 EC the PKD operation will be routed for execution to the CEX2 Coprocessor or CEX2 Accelerator adapter, if available in the configuration. For all presented measurements the PKD operation is in 'clear key' mode which is currently the predominate usage for SSL protocol communications.

For all SSL protocol performance measurements in this publication the following applies:

- Measurements were performed on a System z10 EC with 4 CPs as a Server.
- The performance data is for the server only. The server was driven to a maximum utilization by increasing the number of client systems (on separate systems) until some system resource came to its limits.
- The key length for the Public Key operation is 1024 bits. The SSL data encryption is Triple DES (168 bits) and SHA cipher except when stated otherwise. This SSL data symmetric key encryption for TDES and SHA is executed in CPACF hardware.
- One packet of 2048 Bytes is used as Send Bytes and Receive Bytes.
- The SSL protocol handshake is the pure handshake with the transfer of one 2048 Bytes data packet.

### Legend for all SSL Performance Tables:

**Caching Session ID:** If the SID is cached the initial handshake process is avoided. If the SID is not cached the initial handshake has to be performed for every new connection between Client and Server.

**Handshake:** If the Session ID is 100 % cached the initial handshake is always avoided. If the handshake has to be performed the compute intensive PKD operation, then necessary on the server, can be performed in System SSL software or with hardware on a CEX2 Accelerator or CEX2 Coprocessor adapter.

**Client Authentication:** The authentication of the Client is optional in the SSL protocol.

**External Throughput Rate (ETR):** Number of handshakes performed per second.

**CPU Utilization %:** Average utilization of the System z10 EC Central Processors during the measurement interval.

**Crypto Utilization %:** Average utilization of the CEX2 Accelerator or CEX2 Coprocessor adapters during the measurement interval.

### 4.5.1. Applicability of SSL Performance Results to a Customer Environment

As mentioned, the measurements for the SSL protocol handshake include the 'pure' handshake and the transfer of one 2048 Bytes encrypted data packet. There is no instruction processing for the application which means there is no instruction processing that results from a 'business transaction' with e.g. a query and potential update of a data base. The performance numbers provided give guidelines only on the additional system resources required if an existing On-line transaction environment were converted by replacing the 'unchecked' transaction protocol by an SSL protocol for the communication between Client and Server.

The performance measurement results clearly suggest using cryptographic hardware for improved throughput in the transaction rate if more than a few transactions per second are expected to be handled using an SSL protocol transaction. Furthermore, the measurement results show the throughput with one CEX2 Accelerator adapter being on the order of three times the throughput as with one CEX2 Coprocessor adapter in the SSL environment. Thus for high SSL protocol transaction rate environments, CEX2 Accelerator is the preferred configuration mode for a System z10 EC.

The resource consumption in system processing power for one SSL protocol handshake is on the order of 1/6000 of the system (see table below) in the z/OS environment for a System z10 EC Model E26 with 4 Central Processors and 2 CEX2 features (4 CEX2 Accelerator cards).

If the transaction were to be 'secured' by an SSL protocol and the server portion were run on a System z10 EC the maximum transaction rate achieved on that server without the SSL protocol would be reduced by the portion of processing capacity that is required for the Server SSL protocol path length.

#### 4.5.2. SSL Protocol Performance - System SSL with z/OS V1.9 / Cryptographic Support for z/OS V1.7, V1.8, V1.9 and z/OS.e 1 V1.7 and V1.8 Web deliverable (ICSF)

##### System z10 EC Model E26 (4 Central Processors)

Caching SID	Handshake	Client Auth.	ETR	CPU Util. %	Crypto Util. %
100%	Avoided	no	13,197	92.6	NA
no	Software	no	912	99.5	NA
no	8 CEX2C	no	9,760	97.1	97.7
no	4 CEX2A	no	9,618	95.1	75.4
no	4 CEX2A	yes	6,525	94.7	63.6

Using the CEX2C cryptographic hardware compared to using System SSL Software (second and third rows in the above table) produces an increase in throughput (number of SSL protocol handshakes per second) of 10.7 times.

The 9,760 ETR (third row) represents close to the maximum number of SSL handshakes that can be supported with this system because the 4 Central Processors are 97% utilized. The average utilization of the 8 CEX2C adapters is 97.7%, indicating that the 8 CEX2C adapters could process more than 9900 SSL handshakes before reaching 100% utilization.

The fourth row shows that a similar throughput rate can be achieved with CEX2A adapters. With the CEX2A configuration mode, only 4 adapters were used and the average utilization of the CEX2A adapters was 75.4%, indicating that the 4 CEX2A adapters could process more than 12,500 SSL handshakes before reaching 100% utilization.

If Client authentication is required the throughput of the server is considerably reduced, as shown in row 5 of the above table.

### 4.5.3. SSL Protocol Performance - Linux on System z OpenSSL

For all Linux Open SSL measurements the following applies:

- Linux System Level: SLES10 SP0
- Linux Kernel Level: 2.6.16
- Open SSL Code Level: 0.9.7d
- z90crypt version: 1.3.3
- No Client Authentication

#### Linux Open SSL - Native Measurements

Caching SID	Handshake	ETR	CPU Utilization %
no	Software	1,183	91.9
no	8 CEX2C Cards	9,029	84.2
no	4 CEX2A Cards	10,064	99.1

Using the CEX2 Coprocessor hardware provides an increase in throughput (number of SSL protocol handshakes) of 7.6 times the throughput of using Open SSL Software (first and second line in the above table). The CPU Utilization was only 84.2% because the 8 CEX2 Coprocessors had reached their maximum throughput capacity.

Using the CEX2 Accelerator hardware provides an even higher ETR and fewer adapters are required to support the transaction rate.