IBM z14 Performance of Cryptographic Operations

(Cryptographic Hardware: CPACF, CEX6S)

IBM Corporation

Marketing Communications, Server Group

Route 100

Somers, NY 10589

U.S.A.

Produced in the United States of America

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 Contents

Preface

The performance information presented in this publication was measured on IBM™ z14™ 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 features described herein are presented for informational purposes; actual performance and security characteristics may vary depending on individual customer configurations and conditions.

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 is 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 z14. z14 supports the following cryptographic hardware features:

1. Central Processor Assist for Cryptographic Function (CPACF).
2. Crypto Express5S (CEX5S) feature.
3. Crypto Express6S (CEX6S) feature.

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

The Crypto Express5S feature is supported on z14, however this document does not present performance information for CEX5S. Performance information for CEX5S on z13™ can be found at *IBM z13 Performance of Cryptographic Operations*. The Crypto Express5S is the same feature which is available in z13, and is expected to exhibit similar performance characteristics when installed in z14.

CEX6S is a PCIe adapter card that contains a cryptographic coprocessor subsystem housed

within a FIPS 140-2 Level 4 physically secure enclosure (security module). It is planned for use in IBM Z, Power Systems and as a Machine Type Model (MTM) in X86 servers to provide secure cryptographic functions to banking, finance and high data security customers. The primary customer application within the card is CCA (Common Cryptographic Architecture). CEX6S is a follow-on to CEX5S with improved performance, PCI-HSM compliance mode, concurrent upgrade, improved RAS, and addresses CEX5S end of life components.

Using the HMC console, the CEX6S feature can be configured to function as a CCA Coprocessor (for secure key encrypted operations), Enterprise PKCS #11 Coprocessor (for PKCS #11 secure key operations), or Accelerator (for Secure Sockets Layer / Transport Layer Security (SSL/TLS) acceleration).

All CEX6S data presented in this document is from actual measurements with one or more CEX6S features configured as denoted in each section.

2. Cryptographic Hardware Supported on z14

2.1 Central Processor Assist for Cryptographic Function (CPACF)

CPACF delivers cryptographic support for Advanced Encryption Standard (AES), Triple DES (TDES) and Data Encryption Standard (DES) encryption/decryption, as well as Secure Hash Algorithm (SHA). z14 has one CPACF for every Central Processor (CP), therefore, CPACF encryption throughput scales with the number of CPs in the system.

The SHA functions are shipped enabled. The AES, TDES and DES functions require enablement of the CPACF for export control. CPACF functions for AES, TDES, DES and SHA can be invoked by problem state instructions defined by an extension of the z14 architecture called Message Security Assist (MSA). Support is also available for z/OS® via Cryptographic Support for z/OS V2R1 – z/OS V2R3 (ICSF FMID HCR77C1) web deliverable and for Linux on Z via the ICA IBM Z hardware cryptographic library (libica).

z14 continues support introduced with System z10 EC GA3 for the capability to invoke CPACF functions with protected keys. CPACF protected keys are wrapped with a CPACF wrapping key and are never in operating system addressable memory in an unwrapped (unencrypted) state. Using CPACF functions with protected keys leverages the encryption performance benefits of CPACF hardware while providing added protection required by security sensitive applications. Support for CPACF functions with clear key values remains unchanged.

The CPACF hardware that performs the symmetric key operations (AES; TDES; DES) and SHA functions operates synchronously to CP operations. The CP cannot perform any other

instruction execution while a CPACF cryptographic operation is being executed. The hardware has a fixed set up time per request and a fixed operation speed for the unit of operation. Therefore maximum throughput can be achieved for larger blocks of data (up to a hardware defined limit).

2.2 Crypto Express6S (CEX6S) Feature

The Crypto Express6S feature combines the functions of CCA Coprocessor (for secure key encrypted transactions), Enterprise PKCS #11 Coprocessor (for PKCS #11 secure key operations), and Accelerator (for SSL/TLS acceleration) modes in a single feature. Using the HMC console, the CEX6S feature can be configured to function as a CCA Coprocessor, a PKCS #11 Coprocessor, or an Accelerator. The Crypto Express6S feature is a follow-on to the Crypto Express5S feature with updates to provide additional function and improved performance.

Up to 16 Crypto Express6S features can be installed in a z14.

When configured in CCA Coprocessor mode (CEX6C), the CEX6S feature supports:
- Use of secure encrypted key values
- A wide variety of symmetric key, public key, hashing, and other cryptographic functions
- Specialized cryptographic functions required for banking and payment card applications
- Support for user defined extensions (UDX)
- Support for Payment Card Industry Hardware Security Module (PCI-HSM) security requirements

When configured in Enterprise PKCS #11 Coprocessor mode (CEX6P), the CEX6S feature supports:
- Use of secure encrypted key values
- A wide variety of symmetric key, public key, hashing, and other cryptographic functions
- Industry standard PKCS #11 cryptographic API

The CEX6S in Coprocessor mode (either CCA or Enterprise PKCS #11) 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 AES, Elliptic Curve, RSA, TDES, DES and SHA cryptographic operations in a secure environment. The CEX6S Coprocessor is designed to protect the cryptographic keys used by security sensitive applications. Secure cryptographic keys are encrypted under the Master Key when outside the boundary of the CEX6S. The Master Keys are always kept in battery backed-up memory within the tamper-protected boundary of the CEX6S Coprocessor and are destroyed if physical tampering is detected.

A CEX6S configured in CCA Coprocessor mode can also be configured in Payment Card Industry Hardware Secure Module (PCI-HSM) compliance mode. When configured as PCI-HSM compliant, the CEX6S will support the use of compliant tagged keys in cryptographic operations. A Trusted Key Entry (TKE) workstation is required to configure the CEX6S in PCI-HSM compliance mode.

The CEX6S in CCA Coprocessor mode also supports the 'clear key' PKA operations that currently are predominantly used to support SSL/TLS protocol communications.

When configured in Enterprise PKCS #11 Coprocessor mode, the CEX6S feature implements an IBM version of the PKCS #11 standard and provides hardware support for PKCS #11 operations utilizing secure keys. A Trusted Key Entry (TKE) workstation is required to configure the CEX6S in Enterprise PKCS #11 mode.

When configured in Accelerator mode (CEX6A), the CEX6S 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 off-loaded from the CP to the CEX6S Accelerator and thus increase system capacity. The CEX6S in Accelerator mode works in 'clear key' mode only.

The Crypto Express6S executes its cryptographic operations asynchronously to a Central Processor (CP) operation in z14. When a cryptographic operation completes on the CEX6S, an interrupt will be presented to the host (z/OS or Linux on Z), which will then dequeue the result from the CEX6S and return it to the requesting application. For each CEX6S, 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. Within the Cryptographic Express6S, several operations can be worked on in parallel.

For z14, the Crypto Express6S works with ICSF FMID HCR77C1 and the IBM Resource Access Control Facility (RACF®) in a z/OS operating environment to provide cryptographic services with the IBM Common Cryptographic Architecture (CCA) or the IBM Enterprise PKCS #11 (EP11) protocol.

The CCA and EP11 implementations provide a base on which customer programs can request cryptographic services from the Crypto Express6S. For unique customer cryptographic application requirements the Crypto Express6S in CCA Coprocessor mode provides for user-defined extensions (UDX) to the CCA interface.

In a IBM Z environment an application will not have direct access to the Crypto Express cards. The application requiring a cryptographic service will call a programming interface which is interpreted by some services of the System Control Program.

In the z14 using the z/OS System Control Program, CEX6S cryptographic hardware can only

be used through ICSF. ICSF is a standard component of z/OS that provides the callable services 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.

The CPACF hardware can be accessed either via ICSF callable services or by Message Security Assist instructions provided by the system architecture. The performance of both modes of operation will be presented in this publication.

When using a Linux on Z Control Program, CEX6S cryptographic hardware can be used through either openSSL (with the ibmca engine) or openCryptoki cryptographic interfaces. The ICA (libica), EP11 and CCA (libcsulcca) libraries pass the request to the CEX6S via the zcrypt device driver. The openCryptoki interface and CCA library is used for all data presented in this publication.

3. Performance Information

3.1 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 2 Release 3 (z/OS V2.3) and ICSF FMID HCR77C1.

Linux on Z performance measurements were performed with SUSE™ Linux Enterprise Server 12 Service Pack 3 (SLES 12 SP3), openCryptoki-64bit-3.1-7.35.s390x, and csulcca-5.2.23-11.s390x.

All measurements were performed on an IBM z14 Model 3906-750. 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 the measurement were run on a z14 Model 3906-750 with only one dedicated CP.

For the cryptographic operations that can be used with a variable length of data such as Advanced Encryption Standard (AES) 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 may scale with usage of multiple features.

3.2 CP Assist for Cryptographic Function (CPACF)

3.2.1 CPACF Performance - MSA Architecture Interface

Prior to System z10 EC GA3, all CPACF functions required the use of clear keys. With z10 EC GA3 and beyond the CPACF MSA architecture interface was extended to support the use of CPACF protected keys. CPACF protected keys are wrapped with a CPACF wrapping key and are never in operating system addressable memory in an unwrapped (unencrypted) state. Using CPACF functions with protected keys leverages the performance benefits of CPACF hardware while providing added key protection required by security sensitive applications. This section presents CPACF encryption rates using the MSA architecture instructions for both clear key and protected key modes of operation.

The results show that protected key operations have lower encryption rates than the equivalent clear key operation. This is expected because the protected key needs to first be unwrapped within the CPACF (using a CPACF wrapping key) before the requested instruction can be processed. As the data length increases, the key manipulation is a less dominant factor and the protected key rate approaches the clear key rate.

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

The data quoted is from test cases run on a z14 Model 3906-750, however, using only one of the CPACFs. Scalability measurements were also taken using 4 CPACFs (not quoted) and in all cases the throughput with 4 CPACFs was four times the throughput of 1 CPACF. z14 has one CPACF for every Central Processor (CP), therefore, CPACF encryption throughput is expected to scale with the number of CPs in the system. Scalability measurements had 4 dedicated CPs and 4 concurrent jobs that initiated the cryptographic operation.

Terminology Explanation: The term AES stands for Advanced Encryption Standard according to NIST FIPS 197 and related standards. The term DEA stands for Data Encryption Algorithm which is a block cipher according to the Data Encryption Standard (DES).

3.2.1.1 CPACF MSA Architecture Interface - Clear Key Mode

AES Cipher Block Chaining Encipher with 128 Bit Key
(IBM Z Message Security Assist architecture instruction: KMC-AES clear key)

Native: AES 128 bit CBC Encipher (KMC-AES clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7703900	493
256	7807448	1998
1024	4185826	4286
4096	1147194	4698
64K	76620	5021
1M	4762	4993

AES 128 bit cipher block chaining decipher performance was 12% lower than the encipher operation for the smallest data size measured (64 bytes) and was 175% higher for the largest data sized measured (1 M bytes).

AES Cipher Block Chaining Encipher with 256 Bit Key
(IBM Z Message Security Assist architecture instruction: KMC-AES clear key)

Native: AES 256 bit CBC Encipher (KMC-AES clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7802060	499.3
256	7804191	1997
1024	3143680	3219
4096	891702	3652
64K	55998	3669
1M	3497	3667

AES 256 bit cipher block chaining decipher performance was 17% lower than the encipher operation for the smallest data size measured (64 bytes) and was 250% higher for the largest data sized measured (1 M bytes).

AES XTS Encipher with 128 Bit Key
(IBM Z Message Security Assist architecture instruction: KM-XTS clear key)

Native: AES 128 bit XTS Encipher (KM-XTS clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7455228	477.1
256	7792475	1994
1024	7804423	7991
4096	3439030	14086
64K	249896	16377
1M	13622	14283

AES 128 bit XTS decipher has similar performance characteristics as the encipher operation.

AES XTS Encipher with 256 Bit Key
(IBM Z Message Security Assist architecture instruction: KM-XTS clear key)

Native: AES 256 bit XTS Encipher (KM-XTS clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	6375991	408.0
256	7798505	1996
1024	7802986	7990
4096	3346849	13708
64K	234842	15390
1M	12534	13142

AES 256 bit XTS decipher has similar performance characteristics as the encipher operation.

GCM–AES 128 bit
(IBM Z Message Security Assist architecture instruction: KMA-GCM-AES-128 clear key)

Native: GCM-AES 128 bit Encipher (KMA-GCM-AES-128 clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4826058	308.8
256	4507339	1153
1024	3832069	3924
4096	1895442	7763
64K	206998	13565
1M	12007	12590

GCM-AES 128 bit decipher has similar performance characteristics as the encipher operation.

GCM–AES 256 bit
(IBM Z Message Security Assist architecture instruction: KMA-GCM-AES-256 clear key)

Native: GCM-AES 256 bit Encipher (KMA-GCM-AES-256 clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4868499	311.5
256	4515334	1155
1024	3876711	3969
4096	1908251	7816
64K	210055	13766
1M	12144	12734

GCM-AES 256 bit decipher has similar performance characteristics as the encipher operation.

TDEA Cipher Block Chaining Encipher with Triple Length Key (192 Bits)
(IBM Z Message Security Assist architecture instruction: KMC-TDEA clear key)

Native: Triple DES CBC Encipher (KMC-TDEA clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7803585	499.4
256	2706098	692.7
1024	763479	781.8
4096	196679	805.5
64K	12281	804.8
1M	767.8	805.1

TDEA cipher block chaining decipher with triple length key has similar performance characteristics as the encipher operation.

Compute Message Authentication Code (MAC) with TDEA Triple Length Key (192 Bits)
(IBM Z Message Security Assist architecture instruction: KMAC-TDEA clear key)

Native: Compute MAC with triple DES (KMAC-TDEA clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7830712	501.1
256	2757630	705.9
1024	769470	787.9
4096	197705	809.8
64K	12323	807.6
1M	769.8	807.2

Compute Message Digest SHA-1
(IBM Z Message Security Assist architecture instruction: KLMD-SHA-1 clear key)

Native: Compute Message Digest SHA-1(KLMD-SHA-1 clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7826899	500.9
256	4520570	1157
1024	1602089	1640
4096	445708	1825
64K	28134	1843
1M	1756	1841

Compute Message Digest SHA-512
(IBM Z Message Security Assist architecture instruction: KLMD-SHA-512 clear key)

Native: Compute Message Digest SHA-512(KLMD-SHA-512 clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7823181	500.6
256	6551108	1677
1024	2810787	2878
4096	856958	3510
64K	54750	3588
1M	3412	3578

Compute Message Digest SHA3-256
(IBM Z Message Security Assist architecture instruction: KLMD-SHA3-256 clear key)

Native: Compute Message Digest SHA3-256(KLMD-SHA3-256 clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7826190	500.8
256	7827079	2003
1024	4553906	4663
4096	1546558	6334
64K	110630	7250
1M	6924	7261

Compute Message Digest SHA3-512
(IBM Z Message Security Assist architecture instruction: KLMD-SHA3-512 clear key)

Native: Compute Message Digest SHA3-512(KLMD-SHA3-512 clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7810004	499.8
256	7286488	1865
1024	3570657	3656
4096	1128644	4622
64K	76385	5006
1M	4787	5020

3.2.1.2 CPACF MSA Architecture Interface - Protected Key Mode

This section presents the results from test cases using protected keys. CPACF protected keys are keys wrapped with a CPACF wrapping key and are never in operating system

addressable memory in an unwrapped (unencrypted) state. In our testing, the PCKMO instruction was used to wrap the appropriate key type as specified with each test case. The wrapped key was then used in the KMC, KMA or KMAC instruction. The PCKMO instruction execution is not included in the results.

AES Cipher Block Chaining Encipher with 128 Bit Key
(IBM Z Message Security Assist architecture instruction: KMC-AES protected key)

Native: AES 128 bit CBC Encipher (KMC-AES protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7268883	465.2
256	5629912	1441
1024	3126550	3201
4096	1048191	4293
64K	76005	4981
1M	4782	5014

AES-128 CBC decipher performance was 5% better than encipher with the smallest measured data length (64 bytes) and was 175% better with the largest measured data length (1 MB).

AES Cipher Block Chaining Encipher with 256 Bit Key
(IBM Z Message Security Assist architecture instruction: KMC-AES protected key)

Native: AES 256 bit CBC Encipher (KMC-AES protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	6996928	447.8
256	5071525	1298
1024	2523283	2583
4096	834955	3419
64K	55505	3637
1M	3491	3660

AES-128 CBC decipher performance was 5% better than encipher with the smallest measured data length (64 bytes) and was 250% better with the largest measured data length

(1 MB).

AES XTS Encipher with 128 Bit Key
(IBM Z Message Security Assist architecture instruction: KM-XTS protected key)

Native: AES 128 bit XTS Encipher (KM-XTS protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7612012	487.1
256	6970385	1784
1024	5530438	5663
4096	2683771	10992
64K	243684	15970
1M	13687	14352

AES 128 bit XTS decipher has similar performance as the encipher operation.

AES XTS Encipher with 256 Bit Key
(IBM Z Message Security Assist architecture instruction: KM-XTS protected key)

Native: AES 256 bit XTS Encipher (KM-XTS protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	7415811	474.6
256	6716361	1719
1024	5345780	5474
4096	2629450	10770
64K	229491	15039
1M	12749	13085

AES 256 bit XTS decipher has similar performance as the encipher operation.

GCM–AES 128 bit
(IBM Z Message Security Assist architecture instruction: KMA-GCM-AES-128 protected key)

Native: GCM-AES 128 bit Encipher (KMA-GCM-AES-128 protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	3573035	228.6
256	3386128	866.8
1024	3003423	3075
4096	1668188	6832
64K	205060	13438
1M	12046	12631

GCM-AES 128 bit decipher has similar performance as the encipher operation.

GCM–AES 256 bit
(IBM Z Message Security Assist Architecture instruction: KMA-GCM-AES-256 protected key)

Native: GCM-AES 256 bit Encipher (KMA-GCM-AES-256 protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	3533623	226.1
256	3346982	856.8
1024	2964473	3035
4096	1654711	6777
64K	206024	13502
1M	12132	12722

GCM-AES 256 bit decipher has similar performance as the encipher operation.

TDEA Cipher Block Chaining Encipher with Triple Length Key (192 Bits)
(IBM Z Message Security Assist architecture instruction: KMC-TDEA protected key)

Native: Triple DES CBC Encipher (KMC-TDEA protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4756741	304.4
256	2202628	563.8
1024	716852	734.0
4096	193280	791.6
64K	12252	802.9
1M	766.5	803.7

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

Compute Message Authentication Code (MAC) with TDEA Triple Length Key (192 Bits)
(IBM Z Message Security Assist Architecture instruction: KMAC-TDEA protected key)

Native: Compute MAC with triple DES (KMAC-TDEA protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4856592	310.8
256	2227384	570.2
1024	720875	738.1
4096	190355	779.6
64K	12244	802.4
1M	769.8	807.2

3.2.2 CPACF Performance - ICSF API

Prior to Cryptographic Support for z/OS V1.9 through z/OS V1.11 Web deliverable (ICSF

FMID HCR7770) all CPACF functions available via ICSF required the use of clear keys. In ICSF FMID HCR7770 and beyond the ICSF APIs were extended to leverage CPACF support for protected keys. CPACF protected keys are keys wrapped with a CPACF wrapping key and are never in operating system addressable memory in an unwrapped state. Using CPACF functions with protected keys leverages the performance benefits of CPACF hardware while providing added key protection required by security sensitive applications. This section presents CPACF encryption rates using the ICSF API for both clear key and protected key modes of operation.

All test cases are written in IBM 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 IBM Z Message Security Assist (MSA) Architecture as indicated with each group.

The data quoted is from test cases run on a z14 Model 750, however, using only one of the CPACFs. Scalability measurements were also taken using 4 CPACFs (not quoted). Scalability measurements had 4 dedicated CPs and 4 concurrent jobs that initiated the cryptographic operation. The throughput with 4 CPACFs was 2.6 times (for operations with small data lengths) to 4 times (for operations with large data lengths) the throughput with 1 CPACF.

As the performance measurement results show, all ICSF API test cases have lower throughput than the equivalent MSA architecture test cases. This is expected because of the additional instruction path length involved when calling the ICSF API rather than executing the MSA instruction directly. As the data length increases, the ICSF path length is a less dominant factor and the throughput for large data lengths is nearly the same as when the MSA instruction is executed directly.

3.2.2.1 CPACF ICSF API - Clear Key Operations

AES Cipher Block Chaining Encipher with 128 Bit Key - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMC-AES clear key)

ICSF API: CSNBSYE AES 128 bit Encipher (KMC-AES clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	787648	50.4
256	786470	201.3
1024	714994	732.1
4096	484806	1985
64K	70008	4588
1M	4827	5062

AES 128 bit CBC decipher performance was equivalent to the encipher operation for the smallest data size measured (64 bytes) and was 150% higher for the largest data sized measured (1 M bytes).

AES Cipher Block Chaining Encipher with 256 Bit Key - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMC-AES clear key)

ICSF API: CSNBSYE AES 256 bit Encipher (KMC-AES clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	875252	56.0
256	871629	223.1
1024	714697	731.8
4096	462037	1892
64K	52915	3467
1M	3521	3693

AES-256 CBC decipher performance was equivalent to the encipher operation for the smallest data size measured (64 bytes) and was 250% higher for the largest data sized measured (1 M bytes).

AES Galois Counter Mode Encipher with 128 Bit Key - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMA-GCM-AES-128 clear key)

ICSF API: CSNBSYE GCM-AES 128 bit Encipher (KMA-AES-128 clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	688252	44.0
256	688590	176.2
1024	685039	701.4
4096	580786	2378
64K	159078	10425
1M	12225	12819

GCM-AES 128 bit decipher performance was 15% higher than the encipher operation for the smallest data size measured (64 bytes) and was equivalent for the largest data sized measured (1 M bytes).

AES Galois Counter Mode Encipher with 256 Bit Key - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMA-GCM-AES-256 clear key)

ICSF API: CSNBSYE GCM-AES 256 bit Encipher (KMA-AES-256 clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	757705	48.4
256	750479	192.1
1024	700842	717.6
4096	637253	2610
64K	159329	10441
1M	12481	13087

GCM-AES 256 bit decipher performance was 5% higher than the encipher operation for the smallest data size measured (64 bytes) and was equivalent for the largest data sized measured (1 M bytes).

TDEA Cipher Block Chaining Encipher with Triple Length Key (192 Bits) - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMC-TDEA clear key)

ICSF API: CSNBSYE Triple DES CBC Encipher (KMC-TDEA clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	892037	57.0
256	750461	192.1
1024	439449	449.9
4096	168290	689.3
64K	12278	804.6
1M	777.9	815.7

TDEA decipher with triple length key has similar performance characteristics as the encipher operation.

Compute Message Digest - ICSF API CSNBOWH
(IBM Z Message Security Assist architecture instruction: KLMD)

Compute Message Digest ICSF API: CSNBOWH (KLMD clear key) 1 job				
	Operations/sec			
Data Length (Bytes)	SHA-1	SHA-512	SHA3-256	SHA3-512
64	406667	405446	398601	397974
256	389347	396524	395045	390092
1024	337122	361529	380412	374058
4096	218412	282229	327825	294653
64K	26584	48762	87745	64932
1M	1767	3404	6920	4791

3.2.2.2 CPACF ICSF API - Protected Key Operations

As previously mentioned, ICSF FMID HCR7770 and beyond support the use of protected keys with CPACF encryption. CPACF protected keys are keys wrapped with a CPACF wrapping key and are never in operating system addressable memory in an unwrapped (unencrypted) state. The application uses the ICSF API for a desired CPACF encryption

operation and supplies a secure key as input. The secure key is decrypted from the master key in the CEX6S and then encrypted with a CPACF wrapping key prior to being passed back to ICSF and subsequently to the CPACF. This section presents CPACF protected key encryption rates using the ICSF API.

The results show that CPACF protected key operations have lower throughput rates than the equivalent clear key operation (Section 3.2.2.1). The rates are expected to be lower than clear key rates because the CPACF wrapped key needs to first be decrypted with the CPACF wrapping key prior to the requested operation being performed. As the data length increases, the key manipulation is a less dominant factor and the protected key rate approaches the clear key rate.

The results also show that CPACF protected key operations have higher throughput rates than the equivalent secure key operation executed on a CEX6S feature (Section 3.3.1). The first time a secure key is used for CPACF encryption, ICSF caches the CPACF wrapped key, avoiding the need to decrypt the secure key from the master key in the CEX6S and encrypt the key with the CPACF wrapping key for subsequent encryption requests using the same secure key. Using CPACF functions with protected keys leverages the performance benefits of CPACF hardware while helping to maintain key protection required by security sensitive applications.

AES Cipher Block Chaining Encipher with 128 Bit Key - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMC-AES protected key)

ICSF API: CSNBSYE AES 128 bit Encipher (KMC-AES protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	329314	21.0
256	326044	83.4
1024	303696	310.9
4096	260877	1068
64K	61731	4045
1M	4825	5059

AES 128 bit decipher has similar performance characteristics as the encipher operation for small data lengths. At data lenghts of 4096 bytes and above, decipher performance begins to exceed encipher, reaching 150% better at the 1M bytes data point.

AES Cipher Block Chaining Encipher with 256 Bit Key - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMC-AES protected key)

ICSF API CSNBSYE: AES 256 bit Encipher (KMC-AES protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	327891	20.9
256	319173	81.7
1024	286919	293.8
4096	244672	1002
64K	47945	3142
1M	3530	3702

AES 256 bit decipher has similar performance characteristics as the encipher operation for small data lengths. At data lenghts of 4096 bytes and above, decipher performance begins to exceed encipher, reaching 250% better at the 1M bytes data point.

AES Galois Counter Mode Encipher with 128 Bit Key - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMA-GCM-AES-128 protected key)

ICSF API: CSNBSYE GCM-AES 128 bit Encipher (KMA-AES-128 protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	313153	20.0
256	311613	79.7
1024	303312	310.5
4096	285177	1168
64K	121922	7990
1M	12716	13333

AES Galois Counter Mode Encipher with 256 Bit Key - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMA-GCM-AES-256 protected key)

ICSF API: CSNBSYE GCM-AES 256 bit Encipher (KMA-AES-256 protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	312933	20.0
256	309207	79.1
1024	301952	309.1
4096	285019	1167
64K	121704	7976
1M	12690	13307

TDEA Cipher Block Chaining Encipher with Triple Length Key (192 Bits) - ICSF API CSNBSYE
(IBM Z Message Security Assist architecture instruction: KMC-TDEA protected key)

ICSF API: CSNBSYE Triple DES CBC Encipher (KMC-TDEA protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	334023	21.3
256	304208	77.8
1024	240694	246.4
4096	127368	521.7
64K	12031	788.5
1M	780.2	818.1

Decipher with triple length key has similar performance characteristics as the encipher operation.

3.3 Crypto Express6S Performance (z/OS)

The Crypto Express6S feature is designed to satisfy high-end server security requirements. The Crypto Express6S feature is configurable and can be defined for secure key encrypted transactions (CCA Coprocessor – the default, or Enterprise PKCS #11 Coprocessor) or clear key SSL acceleration (Accelerator). Like its predecessors, the Crypto Express6S feature has been designed to satisfy the security requirements of an enterprise server.

When configured as a Coprocessor (either CCA or Enterprise PKCS #11), the PCIe adapter is designed to provide security-rich cryptographic operations to be used by z14 host application programs. The Coprocessor mode offers security for symmetric keys and private keys. In this case the cryptographic keys are encrypted under the corresponding Master Keys when outside the boundary of the HSM.

When configured as an Accelerator, the PCIe adapter is designed to provide high speed acceleration of Elliptic Curve and 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 or TLS session, in 'clear key' mode.

The connection of the CEX6S feature via the PCIe bus to the z14 Central Processors (CPs) incurs latency and data transmission time. Because of this connection to the z14 CPs, the CEX6S operates asynchronously to the z14 CPs.

There can be a maximum of 16 CEX6S features in a z14, each CEX6S feature containing one PCIe adapter.

3.3.1 CEX6S CCA Coprocessor (CEX6C) - Encryption/Decryption and MAC Operations

This section deals with CEX6S CCA Coprocessor cryptographic operations with a user supplied length of data as, e.g., AES or TDES operations.

All test cases are written in IBM Assembler Language issuing an API call to ICSF for the cryptographic operation. ICSF will resolve the API call and handle the communication with the CEX6S CCA 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 PCIe adapter.

The throughput for symmetric key operations using the CEX6S CCA Coprocessor is

considerably less than the throughput for the corresponding operations using CP Assist for Cryptographic Function (CPACF) hardware. For this type of cryptographic operation the CEX6S CCA 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 is from test cases run on a z14 Model 3906-750 using either 1 job that initiates the cryptographic operation or 8 concurrent jobs initiating the cryptographic operation. The execution of the cryptographic operation in the CEX6C is asynchronous to the z14 Central Processor (CP) execution. If 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 CEX6C adapter's multitasking capability allows for enqueing and dequeing of requests in parallel with cryptographic operations being performed. A measurement environment using several parallel jobs highlights better the throughput capacity of the CEX6C adapter whereas the 'single job' measurement environment is better suited to highlight the delay an application may experience waiting for the result of the cryptographic operation performed in the CEX6C. For each cryptographic operation type quoted there is a statement on scalability of the results when multiple jobs are used to initiate operations.

The performance numbers are from measurements using z/OS V2.3 and ICSF FMID HCR77C1.

CEX6S CCA Coprocessor AES 128 bit Cipher Block Chaining Encipher

CEX6C: AES 128 bit CBC Encipher (CSNBSAE)		
Data Length (Bytes)	Operations/sec(1 job)	Operations/sec(8 jobs)
64	10577	15352
256	9152	12604
1024	9028	12553
4096	8406	12342
64K	1063	1725
1M	71.39	117.4

The throughput for eight jobs for CEX6C AES 128 bit CBC encryption is on the order of 1.4 times to 1.6 times higher than for one job.

CEX6S CCA Coprocessor AES 256 bit Cipher Block Chaining Encipher

CEX6C: AES 256 bit CBC Encipher (CSNBSAE)		
Data Length (Bytes)	Operations/sec(1 job)	Operations/sec(8 jobs)
64	10569	15348
256	9126	12593
1024	9019	12526
4096	8225	12313
64K	1017	1727
1M	67.64	117.0

The throughput for eight jobs for CEX6C AES 256 bit CBC encryption is on the order of 1.3 to 1.7 times higher than for one job.

CEX6S CCA Coprocessor TDEA Cipher Block Chaining Encipher with Triple Length Key (192 Bits)

CEX6C: Triple DES CBC Encipher (CSNBENC)		
Data Length (Bytes)	Operations/sec(1 job)	Operations/sec(8 jobs)
64	11844	14558
256	10039	13540
1024	9905	13489
4096	8400	13177
64K	988.1	1846
1M	65.04	124.9

The throughput for eight jobs for CEX6C Triple DES CBC Encipher is on the order of 1.2 times to 1.9 times higher than for one job.

CEX6S CCA Coprocessor Message Authentication Code with TDEA Double Length Key (112 Bits)

CEX6C: MAC with double length DES key (CSNBMGN)		
Data Length (Bytes)	Operations/sec(1 job)	Operations/sec(8 jobs)
64	12355	15247
256	10242	14219
1024	9970	14168
4096	8431	13878
64K	875.8	1340
1M	56.98	85.35

The throughput for eight jobs for CEX6C MAC is on the order of 1.2 to 1.6 times higher than for one job.

CEX6S CCA Coprocessor Hash Message Authentication Code (HMAC)

CEX6C (one job): HMAC Generate Operations per Second (CSNBHMG)			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	7664	7340	7271
256	7648	7335	7261
1024	7609	7294	7228
4096	6898	6751	6921
64K	834	818	856
1M	58.0	57.1	60.1

CEX6C (eight jobs): HMAC Generate Operations per Second (CSNBHMG)			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	9358	8928	8904
256	9339	8926	8901
1024	9307	8908	8883
4096	9211	8822	8818
64K	1128	1065	1068
1M	77.3	73.9	74.9

The throughput for eight jobs for CEX6C HMAC generate is on the order of 1.2 to 1.3 times

higher than for one job.

CEX6C (one job): HMAC Verify Operations per Second (CSNBHMV)			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	7601	7319	7258
256	7591	7311	7241
1024	7547	7274	7199
4096	6835	6722	6881
64K	835	818	854
1M	58.1	56.9	60.0

CEX6C (eight jobs): HMAC Verify Operations per Second (CSNBHMV)			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	9244	8879	8842
256	9230	8875	8825
1024	9198	8851	8814
4096	9092	8759	8740
64K	1125	1064	1066
1M	77.1	73.8	74.9

The throughput for eight jobs for CEX6C HMAC verify is on the order of 1.2 to 1.3 times higher than for one job.

3.3.2 CEX6S CCA Coprocessor - VISA Format Preserving Encryption (FPE)

Format Preserving Encryption (FPE) refers to a method of encryption where the resulting cipher text has the same form as the input clear text. The following table depicts the rates at which a 16 digit Personal Access Number (PAN) can be enciphered, deciphered and translated with one CEX6C.

CEX6C VISA Format Preserving Encryption – 16 digit Personal Access Number (PAN) using VFPE mode or CBC mode		
Operation	Operations/sec (1 job)	Operations/sec (8 jobs)
VFPE Encipher	5784	6868
VFPE Decipher	5661	6762
VFPE Translate	4787	5464
CBC Encipher	10543	14331
CBC Decipher	10799	15122
CBC Translate	10142	13927

3.3.3 CEX6S CCA Coprocessor - Financial Services Examples

The following table gives the performance in maximum number of operations per second for one CEX6S CCA Coprocessor for some selected financial related services.

CEX6C Financial Services - Examples	Ops/s (1 job)	Ops/s (8 jobs)
Key Generate (operational TDES KEY GENKY key)	8600	10860
Clear PIN Generate Alternate (TDES OPINENC + TDES PINGEN keys)	10614	14277
Clear PIN Generate (16 digits) (TDES PINGEN key)	12281	15409
Encrypted PIN Translation (TDES IPINENC key and TDES OPINENC key)	11518	15914
Encrypted PIN Translation (2 DUKPT enabled KEY GENKY keys)	7155	8702
Encrypted PIN Verification (DUKPT enabled KEY GENKY + TDES PINVER keys)	7959	10077

3.3.4 CEX6S CCA Coprocessor - Random Number Generation

Random number generation is commonly exploited by security related applications such as Secure Sockets Layer / Transport Layer Security (SSL/TLS) and Java Secure Socket Extension (JSSE). ICSF FMID HCR77C1 utilizes a random number data cache to enhance performance. The random number data cache resides in private storage within the ICSF address space. The cache is allocated and filled by a random number generate request to the

CEX6S when the ICSF address space is initialized. This support allows ICSF to satisfy random number requests from an internal private cache, eliminating the delay associated with sending each request to the CEX6S. When the cache depletion threshold is reached, ICSF refills the cache in the background while continuing to service incoming requests. Separate random number caches are implemented for non-FIPS and FIPS certified environments. The following table gives the throughput as the number of operations per second for random number generation of various sizes when ICSF FMID HCR77C1 and one CEX6S CCA Coprocessor are used to maintain the cache in a non-FIPS certified environment.

ICSF Service (random bytes requested)	Operations/sec (1 job)
CSNBRNG (8)	1255730
CSNBRNGL (1)	1138384
CSNBRNGL (64)	1128283
CSNBRNGL (1024)	186257
CSNBRNGL (8192)	24222

3.3.5 CEX6S CCA Coprocessor - PKA Operations

The CEX6S CCA 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 lengths of 1024, 2048 and 4096 bits. Throughput rates for Elliptic Curve cryptography (EC) Brainpool (BP) for 192, 256 and 512 bits and Prime Curve (PC) for 192, 256 and 521 bits are also included.

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 as noted in the table. 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 V2.3 and ICSF FMID HCR77C1 invoking the operation via the ICSF API according to the PKCS-1.2 Standard. Measurements were performed on a z14 Model 750.

CEX6S CCA Coprocessor PKA Performance

CEX6C on 3906-750 with z/OS V2.3; ICSF FMID HCR77C1			
Public Key Decrypt (PKD), Public Key Encrypt (PKE) Digital Signature Generate (DSG), Digital Signature Verify (DSV) Symmetric Key Import (encrypted with RSA key) (SYI)			
CEX6C	1	1	2
Jobs	1	8	16
	Operations/sec	Operations/sec	Operations/sec
PKD-CRT 1024 bit	4136	10004	20414
PKD-CRT 2048 bit	1372	9065	18285
PKD-CRT 4096 bit	222	1381	2761
PKD-ME 512 bit	4375	11200	23268
PKD-ME 1024 bit	1411	10101	18750
PKE 1024 bit	7585	9708	21005
PKE 2048 bit	6010	9130	19660
PKE 4096 bit	3536	8214	17558
DSG-CRT 1024 bit	4113	9893	19974
DSG-CRT 2048 bit	1380	8952	18103
DSG-CRT 4096 bit	222	1381	2761
DSG-EC BP-192 bit	3237	10800	23028

DSG-EC BP-256 bit	2463	10725	22855
DSG-EC BP-512 bit	847	5644	11128
DSG-EC PC-192 bit	3768	10808	22818
DSG-EC PC-256 bit	2800	10780	22726
DSG-EC PC-521 bit	1152	7968	15931
DSV-CRT 1024 bit	7558	9910	21663
DSV-CRT 2048 bit	6015	9259	20303
DSV-CRT 4096 bit	3551	8269	17947
DSV-EC BP-192 bit	1854	11312	24477
DSV-EC BP-256 bit	1357	9537	18773
DSV-EC BP-512 bit	434	2741	5479
DSV-EC PC-192 bit	2242	11377	24483
DSV-EC PC-256 bit	1549	11129	21368
DSV-EC PC-521 bit	611	3937	7868
SYI-CRT 512 bit	4781	7852	15575
SYI-CRT 1024 bit	3645	7515	14928
SYI-CRT 4096 bit	222	1393	2758
CRT: Chinese Remainder Theorem; ME: Modulus Exponent; EC-BP: Elliptic Curve – BrainPool; EC-PC: Elliptic Curve – Prime Curve			

The PKA cryptographic operation throughput with 2 CEX6C adapters with a sufficient number of jobs repetitively requesting the same cryptographic operation for the examples in the table above is close to 2 times the throughput of one CEX6C adapter with 8 jobs.

PKA Key Generation

The CEX6S CCA Coprocessor also offers services to generate PKA Keys. The PKA Key Generate performance is listed for RSA key modulus lengths of 512, 1024, 2048 and 4096 bits dependent on the format, either the Chinese Remainder Theorem (CRT) Format or the

Modulus Exponent (ME) Format. Throughput rates for Elliptic Curve cryptography (EC) Brainpool (BP) for 192, 256 and 512 bits and Prime Curve (PC) for 192, 256 and 521 bits are also included.

PKA Key Generation is a compute intensive operation. The table below specifies the number of Key generations per second provided by one CEX6S CCA Coprocessor.

CEX6S CCA Coprocessor PKA Key Generation Performance

CEX6C PKA Key Generate	
	Operations/sec (1 job)
RSA CRT 512 bit	106
RSA CRT 1024 bit	79.1
RSA CRT 2048 bit	20.5
RSA CRT 4096 bit	1.89
RSA ME 512 bit	104.5
RSA ME 1024 bit	75.3
EC BP-192 bit	959
EC BP-256 bit	699
EC BP-512	220
EC PC-192 bit	1158
EC PC-256 bit	809
EC PC-521 bit	307

3.3.6 CEX6S CCA Coprocessor - PCI-HSM mode

Beginning with the CEX6 feature, the CCA coprocessor supports PCI-HSM mode (PHM) which is designed to meet the PCI-HSM standard. The PCI-HSM standard defines a set of operational and technical requirements to help protect the safety of data when processing payment transactions. When configured in PCI-HSM mode, the CEX6S simultaneously supports PCI-HSM compliant operations and non-compliant operations. This section provides

performance data for PCI-HSM compliant operations which utilize PCI-HSM complaint tagged keys.

The following table gives the throughput in number of operations per second for one CEX6S CCA Coprocessor in PCI-HSM mode for some selected symmetric key operations when compliant tagged keys are used.

CEX6C PCI-HSM Financial Services – Examples with Compliant Tagged keys	Ops/s (1 job)	Ops/s (8 jobs)
Key Generate (operational TDES KEY GENKY key)	7950	9964
Clear PIN Generate Alternate (TDES OPINENC + TDES PINGEN keys)	7719	9657
Clear PIN Generate (16 digits) (TDES PINGEN key)	9374	12495
Encrypted PIN Translation (TDES IPINENC key and TDES OPINENC key)	8102	10209
Encrypted PIN Translation (2 DUKPT enabled KEY GENKY keys)	4499	5077

3.3.7 CEX6S Enterprise PKCS #11 Coprocessor (CEX6P) – Encryption / Decryption and HMAC operations

z14 with CEX6S cryptographic feature provides the ability to configure the CEX6S in Enterprise PKCS #11 (EP11) Coprocessor mode. ICSF FMID HCR77C1 supports the use of CEX6S in EP11 Coprocessor mode with secure key PKCS #11 APIs. 'Secure key' means that the key material is always in wrapped form whenever it is outside of the Hardware Security Module (HSM). When configured in EP11 Coprocessor mode none of the legacy CCA Coprocessor function is available. The following tables provide throughput rates for various PKCS #11 secure key operations with a CEX6S EP11 Coprocessor.

CEX6S Enterprise PKCS #11 Coprocessor AES 128-bit Cipher Block Chaining Encipher

CEX6P: AES 128 bit CBC Encipher		
Data Length (Bytes)	Operations/sec(1 job)	Operations/sec(8 jobs)
64	8246	9466
256	8059	9255
1024	7211	8062
4096	5081	5697
64K	355.4	378.8
1M	25.0	26.97

The throughput for eight jobs for CEX6C AES 128 bit CBC encipher is approximately 1.1 times higher than for one job.

CEX6S Enterprise PKCS #11 Coprocessor TDEA Cipher Block Chaining Encipher with Triple Length Key (168 Bits)

CEX6P: Triple DES CBC Encipher		
Data Length (Bytes)	Operations/sec(1 job)	Operations/sec(8 jobs)
64	8361	9567
256	8096	9288
1024	7047	8066
4096	4895	5729
64K	340.6	379.8
1M	23.9	27.2

The throughput for eight jobs for Triple DES CBC encipher is approximately 1.1 times higher than for one job.

CEX6S Enterprise PKCS #11 Coprocessor Secure Key HMAC Operations

CEX6P (one job): HMAC Generate Operations per Second			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	5867	5684	4325
256	5716	5552	3947
1024	4692	4721	3696
4096	4015	4009	3328
64K	387	396	387
1M	28.6	29.3	29.2

CEX6P (1 job): HMAC Verify Operations per Second			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	5850	5715	4292
256	5783	5625	3995
1024	4768	4721	3707
4096	4043	3997	3291
64K	387	396	386
1M	28.6	29.3	29.2

3.3.8 CEX6S Enterprise PKCS #11 Coprocessor (CEX6P) - PKA Operations

CEX6P Enterprise PKCS #11 Coprocessor PKA Performance

Private Key Sign (PKS), Private Key Verify (PKV), Wrap Private Key (WPK), Unwrap Private Key (UPK)		
CEX6P	1	1
Jobs	1	8
	Operations/sec	Operations/sec
PKS-RSA 1024 bit	3308	5908

PKS-RSA 2048 bit	1228	4488
PKS-RSA 4096 bit	217	1108
PKS BrainPool 192 bit	2503	5673
PKS BrainPool 256 bit	1992	5501
PKS BrainPool 512 bit	772	4974
PKS Prime Curve 192 bit	2849	5898
PKS Prime Curve 256 bit	2237	5607
PKS Prime Curve 521 bit	1036	5050
PKV-RSA 1024 bit	3584	3718
PKV-RSA 2048 bit	2379	2579
PKV-RSA 4096 bit	1089	874
PKV BrainPool 192 bit	1549	4581
PKV BrainPool 256 bit	1161	4356
PKV BrainPool 512 bit	404	2706
PKV Prime Curve 192 bit	1849	4895
PKV Prime Curve 256 bit	1337	4579
PKV Prime Curve 521 bit	564	3834
WPK-RSA 1024 bit	3775	4123
WPK-RSA 2048 bit	2673	3037
WPK-RSA 4096 bit	1358	1567
UPK-RSA 1024 bit	2429	3635
UPK-RSA 2048 bit	1097	3100
UPK-RSA 4096 bit	210	1042

CEX6P IBM PKCS #11 Coprocessor PKA Key Generate Performance

CEX6P PKA Key Generate	
	Operations/sec
RSA CRT 1024 bit	10.87
RSA CRT 2048 bit	2.40
RSA CRT 4096 bit	0.32
EC Brainpool 192 bit	62.49
EC Brainpool 256 bit	49.99
EC Brainpool 512 bit	14.70
EC Prime Curve 192 bit	83.32
EC Prime Curve 256 bit	62.49
EC Prime Curve 521 bit	31.24

3.3.9 CEX6S Accelerator Performance

The CEX6S Accelerator mode is designed to offer fast RSA algorithm cryptographic operations. The performance is listed for RSA key modulus lengths of 1024, 2048 and 4096 bits. The performance numbers are from measurements with z/OS V2.3 and ICSF FMID HCR77C1 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)

CEX6S Accelerator PKA Performance

CEX6A PKA Key Decrypt (PKD), Public Key Encrypt (PKE), and Digital Signature Verify (DSV)		
3906-750 CPs	4	4
CEX6A Adapters	1	1
Jobs	1	8
	Operations/sec	Operations/sec
PKD CRT 1024 bit	7067	50670
PKD CRT 2048 bit	1588	9899
PKD CRT 4096 bit	228	1377
PKD ME 1024 bit	1602	9985
PKE 1024 bit	30341	194636
PKE 2048 bit	16630	122006
PKE 4096 bit	6097	42289
DSV CRT 1024 bit	29942	190790
DSV CRT 2048 bit	16379	121361
DSV CRT 4096 bit	6094	42300

The first result column of the above table is for measurements where one job was continuously executing the cryptographic operation using one CEX6S Accelerator. As mentioned, the execution of the cryptographic operation in the CEX6S Accelerator is asynchronous to the z14 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 may 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 CEX6S Accelerator. The increased throughput is due to the fact that tasks are always available for execution in the CEX6S Accelerator due to the parallel threads that run in the z14 CPs. Thus the capability of the CEX6S Accelerator for parallel execution of the cryptographic operation can be utilized.

3.4 Crypto Express6S Performance (Linux on Z)

3.4.1 CEX6S CCA Coprocessor (CEX6C) - Encryption/Decryption

This section deals with CEX6S CCA Coprocessor cryptographic operations with a user supplied length of data as, e.g., AES or TDES operations.

All test cases are written in C language and issue an API call to the ICA (libica) library of cryptographic services for a cryptographic operation. ICA will resolve the API call and handle the communication with the CEX6S CCA 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 PCIe adapter.

The data quoted is from test cases run on a z14 Model 3906-750 using either 1 job that initiates the cryptographic operation or 8 concurrent jobs initiating the cryptographic operation. The execution of the cryptographic operation in the CEX6C is asynchronous to the z14 Central Processor (CP) execution. If 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 CEX6C adapter's multitasking capability allows for enqueing and dequeing of requests in parallel with cryptographic operations being performed. A measurement environment using several parallel jobs highlights better the throughput capacity of the CEX6C adapter whereas the 'single job' measurement environment is better suited to highlight the delay an application may experience waiting for the result of the cryptographic operation performed in the CEX6C. For each cryptographic operation type quoted there is a statement on scalability of the results when multiple jobs are used to initiate operations.

The performance numbers are from measurements using SUSE Linux Enterprise Server 12 SP3 and the following software versions:

- libica-2_3_0-2.3.0-15.2.s390x
- openCryptoki-3.1-7.35.s390x
- csulcca-5.2.23-11.s390x

CEX6S CCA Coprocessor AES 128 bit Cipher Block Chaining Encipher

CEX6C: AES 128 bit CBC Encipher		
Data Length (Bytes)	Operations/sec (1 job)	Operations/sec (8 jobs)
64	11889	16123
256	9882	13549
1024	9723	13539
4096	8904	13228
65536	1204	2124
1024000	76.73	138.8

The throughput for eight jobs for CEX6C AES 128 bit CBC encryption is on the order of 1.3 times to 1.8 times higher than for one job.

CEX6S CCA Coprocessor AES 256 bit Cipher Block Chaining Encipher

CEX6C: AES 256 bit CBC Encipher		
Data Length (Bytes)	Operations/sec (1 job)	Operations/sec (8 jobs)
64	11917	16091
256	9792	13321
1024	9717	13268
4096	8692	13013
65536	1151	2065
1024000	71.32	135.5

The throughput for eight jobs for CEX6C AES 256-bit CBC encryption is on the order of 1.3 to 1.9 times higher than for one job.

CEX6S CCA Coprocessor TDEA Cipher Block Chaining Encipher with Triple Length Key (192 Bits)

CEX6C: Triple DES CBC Encipher		
Data Length (Bytes)	Operations/sec (1 job)	Operations/sec (8 jobs)
64	12503	14208
256	10355	14758
1024	10169	14670
4096	8576	14172
65536	1074	2208
1024000	69.99	143.8

The throughput of one CEX6C adapter with eight jobs is 1.1 to 2.0 times higher than the throughput with 1 job, depending on the data length.

CEX6S CCA Coprocessor Message Authentication Code with Double Length TDES Key (112 Bits)

CEX6C: MAC with double length TDES key		
Data Length (Bytes)	Operations/sec (1 job)	Operations/sec (8 jobs)
64	11051	15951
256	8983	12100
1024	8922	12068
4096	8127	11652
65536	1060	1579
1024000	69.22	103.9

The throughput for eight jobs for CEX6C MAC with double length TDES key is on the order of 1.4 to 1.5 times higher than for one job.

3.4.2 CEX6S CCA Coprocessor - VISA Format Preserving Encryption (FPE)

Format Preserving Encryption (FPE) refers to a method of encryption where the resulting cipher text has the same form as the input clear text. The following table depicts the rates at which a Personal Access Number (PAN) can be enciphered, deciphered and translated with

one CEX6C.

CEX6C VISA Format Preserving Encryption – Personal Access Number (PAN) using VFPE mode or CBC mode		
Operation	Operations/sec (1 job)	Operations/sec (8 jobs)
VFPE Encipher	5692	6769
VFPE Decipher	5707	6699
VFPE Translate	4943	5532
CBC Encipher	10278	13390
CBC Decipher	10820	15070
CBC Translate	10483	14439

3.4.3 CEX6S CCA Coprocessor – Financial Services Examples

The following table gives the performance in maximum number of operations per second for one CEX6S CCA Coprocessor for some selected financial services.

CEX6C Financial Services Examples	Ops/s (1 job)	Ops/s (8 jobs)
Key Generate2 (AES 256-bit key type CIPHER)	7012	8333
Clear PIN Generate Alternate	10978	15242
Clear PIN Generate	12674	15608
Encrypted PIN Translation	11870	15512
Encrypted PIN Translation	7066	8626
Encrypted PIN Verification	8111	10260

3.4.4 CEX6S CCA Coprocessor - Random Number Generation

Random number generation is commonly exploited by security related applications such as Secure Sockets Layer (SSL) and Java Secure Socket Extension (JSSE). The following table gives the throughput in number of operations per second for random number generation of various sizes.

CCA Service (random bytes requested)	Operations/sec (1 job)
CSNBRNG (8)	15604
CSNBRNGL (1)	15884
CSNBRNGL (64)	15783
CSNBRNGL (1024)	13275
CSNBRNGL (8192)	6302

3.4.5 CEX6S CCA Coprocessor - PKA Operations

CEX6C on 3906-750 with SLES 12 SP3		
Public Key Decrypt (PKD), Public Key Encrypt (PKE)		
Digital Signature Generate (DSG), Digital Signature Verify (DSV)		
Jobs	1	8
	Operations/sec	Operations/sec
PKD-CRT 1024 bit	4231	10453
PKD-CRT 2048 bit	1378	8903
PKD-CRT 4096 bit	221	1376
PKD-ME 512 bit	4448	11578
PKD-ME 1024 bit	1438	9547
PKE-CRT 1024 bit	7695	9080
PKE-CRT 2048 bit	5704	8495
PKE-CRT 4096 bit	3357	7315
DSG-CRT 1024 bit	4193	10062
DSG-CRT 2048 bit	1381	8863

DSG-CRT 4096 bit	223	1388
DSG-EC BrainPool 192 bit	3234	11344
DSG-EC BrainPool 256 bit	2460	11278
DSG-EC BrainPool 512 bit	849	5625
DSG Prime Curve 192 bit	3759	11438
DSG Prime Curve 256 bit	2797	11393
DSG Prime Curve 521 bit	1153	7890
DSV-CRT 1024 bit	7599	10543
DSV-CRT 2048 bit	5958	9769
DSV-CRT 4096 bit	3502	9490
DSV BrainPool 192 bit	1881	12000
DSV BrainPool 256 bit	1379	9528
DSV BrainPool 512 bit	434	2709
DSV Prime Curve 192 bit	2323	12778
DSV Prime Curve 256 bit	1589	10690
DSV Prime Curve 521 bit	608	3868
CRT: Chinese Remainder Theorem; ME: Modulus Exponent		

PKA Key Generation

The CEX6S CCA Coprocessor also offers services to generate PKA Keys. The PKA Key Generate performance is listed for RSA key modulus lengths of 1024, 2048 and 4096 bits dependent on the format, either the Chinese Remainder Theorem (CRT) Format or the Modulus Exponent (ME) Format. Throughput rates for Elliptic Curve cryptography (EC) Brainpool (BP) for 192, 256 and 512 bits and Prime Curve (PC) for 192, 256 and 521 bits are also included.

PKA Key Generation is a compute intensive operation. The table below specifies the number

of Key generations per second provided by one CEX6S CCA Coprocessor.

CEX6S CCA Coprocessor PKA Key Generation Performance

CEX6C on 3906-750 with SLES 12 SP3		
Key Generate operations		
CEX6C	1	1
Jobs	1	8
RSA CRT 1024-bit	78.4	80.9
RSA CRT 2048-bit	20.5	21.8
RSA CRT 4096-bit	1.7	1.8
EC BP 192	1011	5035
EC BP 256	725	4467
EC BP 512	223	1376
EC PC 192	1233	5567
EC PC 256	847	4576
EC PC 521	313	1949

3.5 SSL Handshake Performance

3.5.1 SSL / TLS Protocol based Communication

Secure Sockets Layer / Transport Layer Security (SSL/TLS) is a communication protocol that was designed to facilitate secure communication over an open communication network, such as the Internet. The SSL/TLS protocol is a layered protocol that is intended to be used on top of a reliable transport, e.g. Transmission Control Protocol / Internet Protocol (TCP/IP). SSL/TLS 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/TLS 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 / TLS 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 z14 will result in a series of cryptographic operations. In the z/OS environment, System 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 cryptographic function. The SSL/TLS protocol will result in an increase in transaction execution time compared to an unsecured 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 or in software on a CP). This publication will state the performance in the SSL/TLS environment as the maximum number of SSL handshakes the z14 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/TLS environment is to demonstrate the capabilities of a z14 to act as a Web Server providing SSL/TLS compliant communication to a large number of clients. For this purpose the maximum number of SSL/TLS connects and data exchanges per second made between the server and all clients are provided for different configurations. There is no intention to provide a more detailed performance analysis for this environment.

As this performance publication primarily deals with performance of cryptographic operations and Web based communication, the measurements for the SSL/TLS environments include only the processing required for the SSL/TLS 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.

The SSL/TLS handshake 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 a PKA Private Key operation on the server side. 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 z14 the PKD operation will be routed for execution to the CEX6S CCA Coprocessor or CEX6S Accelerator, if available in the configuration. For all presented measurements the PKD operation is in 'clear key' mode which is currently the predominant usage for SSL/TLS protected communications.

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

- Measurements were performed on a z14 with 4 CPs as a server.
- The performance data is for the server side of the transaction only.
- The clients used to drive the workload were running on separate systems. Performance data from the client systems is not included.
- The TLS 1.2 protocol was used.
- The RSA key length for the Public Key operation was 2048 bits as noted in the table. The cipher was TLS_RSA_WITH_AES_128_CBC_SHA except when stated otherwise. The symmetric key data encryption for AES and SHA was executed in CPACF hardware.
- One packet of 2048 bytes was exchanged with each transaction.
- The SSL/TLS 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 CEX6S Accelerator or CEX6S CCA Coprocessor feature.

Client Authentication: Additional processing is required if authentication of the Client is requested. Authentication of the Client is optional.

External Throughput Rate (ETR): Number of transactions performed per second. A transaction is defined as the establishment of a TLS session between the Client and Server, the exchange of 2048 bytes of random data, and session disconnect.

CPU Utilization %: Average utilization of the z14 Central Processors during the measurement interval.

Crypto Utilization %: Average utilization of the CEX6S Accelerator or CEX6S CCA Coprocessor features during the measurement interval.

As mentioned, the measurements for the SSL/TLS handshake include the ‘pure’ handshake and the exchange 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 an unsecured transaction protocol with an SSL/TLS 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/TLS protected transaction. Furthermore, the results show that configuring the CEX6S in Accelerator mode increases the SSL/TLS handshake throughput capacity of the CEX6S. Thus for high SSL/TLS transaction rate environments, Accelerator is the preferred configuration mode for a CEX6S feature.

3.5.2 System SSL with z/OS V2R3 and Cryptographic Support for z/OS V2R1-V2R3 (ICSF FMID HCR77C1)

z14 Model 3906-750 (4 Central Processors)

Caching SID	RSA Key Length	Handshake	Client Auth.	ETR	CPU Util. %	Crypto Util. %
100%	2048	Avoided	no	42,391	80.29	NA
no	2048	Software	no	254	100	NA
no	2048	1 CEX6C	no	8,923	31.74	99.3
no	2048	1 CEX6A	no	9965	34.88	99.2
no	2048	2 CEX6A	yes	10,470	51.27	100

The first row of the table shows the transaction rate when the client SSL/TLS session identifier was cached in the server resulting in the majority of the SSL/TLS handshake processing being avoided.

The next four rows show the transaction rates when the client SSL/TLS session identifier was not cached in the server resulting in a full SSL/TLS handshake for each client connection.

Using the CEX6C cryptographic hardware compared to using System SSL software (second and third rows in the above table) produced an increase in throughput (number of SSL/TLS handshakes per second) of 35.1 times and reduced the CP utilization by 68%. The CEX6 Coprocessor was 99.3% utilized. This demonstrates how off-loading the compute intensive processing associated with an SSL/TLS protocol handshake increased system capacity and reduced CP Utilization. Adding additional CEX6 Coprocessors to this environment would allow for a higher ETR as there was plenty of CPU available to handle additional workload.

The fourth row shows that a higher ETR can be achieved by configuring the CEX6S adapter in Accelerator mode. In this measurement the utilization of the CEX6S Accelerator was 99.2%. Adding additional CEX6 Accelerators to this environment would allow for a higher ETR as there was plenty of CPU available to handle additional workload.

If client authentication is required, the additional cryptographic operations necessary to authenticate the client reduced the throughput capacity of the server, as shown in row 5 of the

table. A second CEX6 Accelerator was added to the system configuration for this measurement. The average utilization of the 2 Accelerators was 100%.