IBM z15 Performance of Cryptographic Operations

(Cryptographic Hardware: CPACF, CEX7S)

IBM Corporation

Marketing Communications, Server Group

Route 100

Somers, NY 10589

U.S.A.

Produced in the United States of America

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 equivalent to the performance rates stated here.

Contents

Preface

The performance information presented in this publication was measured on IBM™ z15™ 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 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 z15. z15 supports the following cryptographic hardware features:

1. Central Processor Assist for Cryptographic Function (CPACF).
2. Crypto Express5S (CEX5S) feature.
3. Crypto Express6S (CEX6S) feature.
4. Crypto Express7S (CEX7S) feature.

The CPACF delivers cryptographic support for Advanced Encryption Standard (AES), Triple DES (TDES) and Data Encryption Standard (DES) data encryption/decryption, Secure Hash Algorithm (SHA) and some PKCS#11 Public Key Algorithms.

The Crypto Express5S, CryptoExpress6S and Crypto Express7S features are supported on z15, however this document does not present performance information for CEX5S or CEX6S. Performance information for CEX5S on z13™ and Crypto Express6S on z14 can be found at IBM z13 Performance of Cryptographic Operations and IBM z14 Performance of Cryptographic Operations respectively. The Crypto Express5S and Crypto Express6S are the same features which are available in z13 and z14 and are expected to exhibit similar performance characteristics when installed in z15.

CEX7S 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). CEX7S is a follow-on to CEX6S with improved performance and addresses CEX7S end of life components.

Using the HMC console, the CEX7S 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) clear key acceleration).

All CEX7S data presented in this document is from actual measurements with one or more CEX7S features configured as denoted in each section.

2. Cryptographic Hardware Supported on z15

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, Secure Hash Algorithm (SHA) and some PKCS#11 Public Key Algorithms. z15 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 z15 architecture called Message Security Assist (MSA). Support is also available for z/OS® via Cryptographic Support for z/OS V2R1 – z/OS V2R4 (ICSF FMID HCR77D1) web deliverable and for Linux on Z via the ICA IBM Z hardware cryptographic library (libica).

z15 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), SHA functions and PKCS#11 algorithms 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 Express7S (CEX7S) Feature

The Crypto Express7S 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 clear key acceleration) modes in a single feature. Using the HMC console, the CEX7S feature can be configured to function as a CCA Coprocessor, a PKCS #11 Coprocessor, or an Accelerator. The Crypto Express7S feature is a follow-on to the Crypto Express6S feature with updates to provide additional function and improved performance.

Up to 16 Crypto Express7S features can be installed in a z15.

When configured in CCA Coprocessor mode (CEX7C), the CEX7S 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 (CEX7P), the CEX7S 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 CEX7S 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 CEX7S 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 CEX7S. The Master Keys are always kept in battery backed-up memory within the tamper-protected boundary of the CEX7S Coprocessor and are destroyed if physical tampering is detected.

A CEX7S 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 CEX7S will support the use of compliant tagged keys in cryptographic operations. A Trusted Key Entry (TKE) workstation is required to configure the CEX7S in PCI-HSM compliance mode.

The CEX7S 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 CEX7S 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 load master keys when the CEX7S is configured in Enterprise PKCS #11 mode.

When configured in Accelerator mode (CEX7A), the CEX7S 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 CEX7S Accelerator and thus increase system capacity. The CEX7S in Accelerator mode works in 'clear key' mode only.

The Crypto Express7S executes its cryptographic operations asynchronously to a Central Processor (CP) operation in z15. When a cryptographic operation completes on the CEX7S, an interrupt will be presented to the host (z/OS or Linux on Z), which will then dequeue the result from the CEX7S and return it to the requesting application. For each CEX7S, 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 Express7S, several operations can be worked on in parallel.

For z15, the Crypto Express7S works with ICSF FMID HCR77D1 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 Express7S. For unique customer cryptographic application requirements, the Crypto Express7S 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 z15 using the z/OS System Control Program, CEX7S 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 must 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, CEX7S 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 CEX7S via the zcrypt device driver. The openCryptoki interface and CCA library is used for all Linux on Z 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 4 (z/OS V2.4) and ICSF FMID HCR77D1.

Linux on Z performance measurements were performed with Red Hat Enterprise Linux 7.8.

All measurements were performed using a single LPAR defined on an IBM z15 Model 8561-770 or IBM z15 Model 8561-703. 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 z15 Model 8561 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 various 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’).

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 z15 Model 8561-770, 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. z15 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	17708834	1133
256	10424401	2668
1024	4252303	4354
4096	1258452	5154
64K	78564	5148
1M	4846	5081

AES 128-bit CBC decipher throughput was 11% higher than the encipher throughput for the smallest data size measured (64 bytes) and was 184% 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	16313183	1044
256	8717697	2231
1024	3228818	3306
4096	915458	3749
64K	57115	3743
1M	3555	3728

AES 256-bit CBC decipher throughput was 15% higher than the encipher throughput for the smallest data size measured (64 bytes) and was 259% 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	19565263	1252
256	15996125	4095
1024	10575272	10829
4096	4277684	17521
64K	265902	17426
1M	14463	15165

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	18566450	1188
256	14802502	3789
1024	9669557	9901
4096	3978854	16297
64K	240816	15782
1M	13333	13981

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-AES clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	5264352	337
256	4807554	1230
1024	4177891	4278
4096	2059384	8435
64K	209106	13704
1M	11928	12507

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-AES clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4890433	313
256	4701131	1203
1024	3985306	4080
4096	2032641	8325
64K	210073	13767
1M	12142	12732

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	8277942	530
256	2796243	716
1024	777475	796
4096	199909	819
64K	12469	817
1M	779.4	817

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-DEA clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	8421966	539
256	2788148	714
1024	768592	787
4096	197489	809
64K	12378	811
1M	770.2	808

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	11865313	759
256	6603474	1690
1024	2843771	2912
4096	867311	3552
64K	55516	3638
1M	3465	3633

Compute Message Digest SHA3-256

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

Native: Compute Message Digest SHA3-256 bit (KLMD-SHA3-256-bit clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	9765874	625
256	8496363	2175
1024	4649040	4760
4096	1560500	6391
64K	112252	7356
1M	7019	7360

Compute Message Digest SHA3-512

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

Native: Compute Message Digest SHA3-512 bit (KLMD-SHA3-512-bit clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	10461129	670
256	7445715	1906
1024	3609822	3696
4096	1138151	4661
64K	77659	5089
1M	4860	5096

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	6387184	409
256	5097744	1305
1024	2979180	3050
4096	1078084	4415
64K	77456	5076
1M	4806	5040

AES-128 CBC decipher throughput was 3% higher than encipher with the smallest measured data length (64 bytes) and was 195% higher 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	6140117	393
256	4618038	1182
1024	2426863	2485
4096	812418	3327
64K	56448	3699
1M	3515	3685

AES-128 CBC decipher throughput was 5% higher than encipher with the smallest measured data length (64 bytes) and was 268% higher 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	6637515	425
256	6191951	1585
1024	5193188	5317
4096	2763490	11319
64K	253891	16639
1M	14306	15001

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	6434627	412
256	5934329	1519
1024	4887946	5005
4096	2600093	10649
64K	231114	15146
1M	13190	13831

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-AES protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	3514007	225
256	3364787	861
1024	2992182	3063
4096	1725195	7066
64K	205670	13478
1M	11954	12535

GCM-AES 128 bit decipher throughput was 42% lower than encipher with the smallest measured data length (64 bytes) and was 7% lower with the largest measured data length (1 MB).

GCM–AES 256 bit

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

Native: GCM-AES 256 bit Encipher (KMA-AES protected key)

Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	3446233	221
256	3306024	846
1024	2933611	3004
4096	1715489	7026
64K	205559	13471
1M	11931	12511

GCM-AES 256 bit decipher throughput was 42% lower than encipher with the smallest measured data length (64 bytes) and was 5% lower with the largest measured data length (1 MB).

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	4478110	287
256	2160048	553
1024	718037	735
4096	195005	799
64K	12393	812
1M	774.7	812

TDEA CBC 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	4545592	291
256	2161882	553
1024	715552	733
4096	193197	791
64K	12199	799
1M	766.7	804

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 z15 Model 770, 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 2.9 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: AES 128 bit Encipher (KMC-AES clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	766095	49.0
256	765377	196
1024	706262	723
4096	479659	1964
64K	71564	4690
1M	4816	5050

AES 128 bit CBC decipher throughput was equivalent to encipher for the smallest data size measured (64 bytes) and was 154% 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: AES 256 bit Encipher (KMC-AES clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	862256	55.1
256	855043	219
1024	712733	730
4096	456059	1868
64K	53557	3509
1M	3490	3660

AES 256 bit CBC decipher throughput was equivalent to encipher for the smallest data size measured (64 bytes) and was 232% 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: GCM-AES 128 bit Encipher (KMA-AES clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	766355	49.0
256	758533	194
1024	714700	732
4096	645926	2645
64K	163699	10728
1M	12069	12655

GCM-AES 128 bit decipher throughput was 8% higher than encipher 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: GCM-AES 256 bit Encipher (KMA-AES clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	823192	52.6
256	816326	209
1024	785822	805
4096	692016	2834
64K	167912	11004
1M	11810	12384

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

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: Triple DES CBC Encipher (KMC-TDEA clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	960378	61.4
256	772558	198
1024	444240	455
4096	166592	682.3
64K	12281	804.8
1M	778.2	816

TDEA decipher with triple length key has similar throughput 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-512	SHA3-256	SHA3-512
64	387854	382715	381869
256	377284	381823	376074
1024	351485	367781	356010

4096	274693	317874	294374
64K	48810	87781	64930
1M	3452	6856	4803

PKCS#11 Clear Key Elliptic Curve Operations supported on CPACF
(IBM Z Message Security Assist architecture instruction: ECSA)

Generate Key Pair (GKP), Derive Key (DVK), Private Key Sign (PKS), Public Key Verify (PKV)	
	Operations/sec (One job)
GKP - Ed448	15028
GKP - Ed25519	24856
GKP - X448	20010
GKP - X25519	31670
GKP - Prime 256 bit	22122
GKP - Prime 384 bit	13024
GKP - Prime 521 bit	8444
DVK - X448	20931
DVK - X25519	34016
DVK - Prime 256 bit	23540
DVK - Prime 384 bit	13396
DVK - Prime 521 bit	8549
PKS - Ed448	16278
PKS - Ed25519	25389
PKS - Prime 256 bit	48660
PKS - Prime 384 bit	25804
PKS - Prime 521 bit	18156
PKV - Ed448	10610
PKV - Ed25519	18877
PKV - Prime 256 bit	19365
PKV - Prime 384 bit	9913
PKV - Prime 521 bit	6446

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 CEX7S 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 CEX7S 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 CEX7S 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: AES 128 bit Encipher (KMC-AES protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	289915	18
256	282775	72
1024	273339	279
4096	235650	965
64K	61095	4003
1M	4744	4975

AES 128 bit decipher has similar performance characteristics as the encipher operation for small data lengths. At data lengths of 1024 bytes and above, decipher throughput begins to exceed encipher, reaching 152% higher 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: AES 256 bit Encipher (KMC-AES protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	285436	18
256	280783	71
1024	267744	274
4096	218987	896

64K	46757	3064
1M	3442	3609

AES 256 bit decipher has similar performance characteristics as the encipher operation for small data lengths. At data lengths of 1024 bytes and above, decipher throughput begins to exceed encipher, reaching 224% higher 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 protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	262880	16.8
256	261782	67
1024	259221	265
4096	248760	1018
64K	113911	7465
1M	11985	12567

GCM AES 128 bit decipher has similar performance characteristics as the encipher operation for all data lengths.

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: GCM-AES 256 bit Encipher (KMA-AES protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	271827	17.3
256	270403	69.2
1024	268589	275
4096	252388	1033
64K	114257	7487
1M	12026	12610

GCM AES 256 bit decipher has similar performance characteristics as the encipher operation for all data lengths.

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: Triple DES CBC Encipher (KMC-TDEA protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	273586	17.5
256	257721	65.9
1024	208213	213
4096	117092	479
64K	11901	779
1M	777	815

Decipher with triple length key has similar performance characteristics as the encipher operation for all data lengths.

3.3 Crypto Express7S Performance (z/OS)

The Crypto Express7S feature is designed to satisfy high-end server security requirements. The Crypto Express7S 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/TLS acceleration (Accelerator). Like its predecessors, the Crypto Express7S 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 z15 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 CEX7S feature via the PCIe bus to the z15 Central Processors (CPs) incurs latency and data transmission time. Because of this connection to the z15 CPs, the CEX7S operates asynchronously to the z15 CPs.

There can be a maximum of 16 CEX7S features in a z15, each CEX7S feature containing one PCIe adapter. The CCA 7.0.68z firmware version was used for all measurements in this

section.

3.3.1 CEX7S CCA Coprocessor (CEX7C) - Encryption/Decryption and MAC Operations

This section deals with CEX7S 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 CEX7S 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 CEX7S 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 CEX7S CCA Coprocessor feature should be used only when the security requirements for the application require it.

The data quoted is from test cases run on a z15 Model 8561-770 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 CEX7C is asynchronous to the z15 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 CEX7C adapter's multitasking capability allows for enqueuing 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 CEX7C 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 CEX7C. 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 V2.4, ICSF FMID HCR77D1.

CEX7S CCA Coprocessor AES 128-bit Cipher Block Chaining Encipher

CEX7C: AES 128-bit CBC Encipher (CSNBSAE) Operations / second		
Data Length (Bytes)	1 job	8 jobs
64	10145	27156
256	10084	26963
1024	9936	25995
4096	9458	24410

64K	1253	2994
1M	83.8	198.8

The throughput of eight jobs for CEX7C AES 128-bit CBC encryption is on the order of 2.3 times to 2.6 times higher than for one job.

CEX7S CCA Coprocessor AES 256-bit Cipher Block Chaining Encipher

CEX7C: AES 256-bit CBC Encipher (CSNBSAE) Operations / second		
Data Length (Bytes)	1 job	8 jobs
64	10130	27090
256	10069	26880
1024	9899	25899
4096	9365	24449
64K	1191	2984
1M	79.29	198.6

The throughput of eight jobs for CEX7C AES 256-bit CBC encryption is on the order of 2.1 to 2.5 times higher than for one job.

CEX7S CCA Coprocessor TDEA Cipher Block Chaining Encipher with Triple Length Key (192 Bits)

CEX7C: Triple DES CBC Encipher (CSNBENC) Operations / second		
Data Length (Bytes)	1 job	8 jobs
64	10533	27542
256	10429	27276
1024	10055	26849
4096	8882	24543
64K	1042.0	2919
1M	68.63	197.3

The throughput of eight jobs for CEX7C Triple DES CBC Encipher is on the order of 2.5 times to 2.7 times higher than for one job.

CEX7S CCA Coprocessor Message Authentication Code with TDEA Double Length Key (112 Bits)

CEX7C: MAC with double length DES key (CSNBMGN) Operations / second		
Data Length (Bytes)	1 job	8 jobs
64	10552	28177
256	10468	27987
1024	10138	27398
4096	9110	22212
64K	999.8	2517

1M	65.4	163.9

The throughput of eight jobs for CEX7C MAC is on the order of 2.3 to 2.6 times higher than for one job.

CEX7S CCA Coprocessor Hash Message Authentication Code (HMAC)

CEX7C (one job): HMAC Generate (CSNBHMG) Operations per Second		
Data Length (Bytes)	SHA-256	SHA-512
64	9298	9267
256	9265	9243
1024	9134	9135
4096	8778	8889
64K	1052	1078
1M	71.6	72.8

CEX7C HMAC generate using SHA-256 and SHA-512 exhibit similar throughput.

CEX7C (eight jobs): HMAC Generate (CSNBHMG) Operations per Second		
Data Length (Bytes)	SHA-256	SHA-512
64	22282	22079
256	22194	22019
1024	22055	21884
4096	21586	21201
64K	2524	2578
1M	170	175

The throughput of eight jobs for CEX7C HMAC generate is on the order of 2.2 to 2.3 times higher than for one job.

CEX7C (one job): HMAC Verify (CSNBHMV) Operations per Second		
Data Length (Bytes)	SHA-256	SHA-512
64	9293	9252
256	9240	9217
1024	9125	9112
4096	8775	8873
64K	1053	1080
1M	70.8	73.0

CEX7C HMAC verify using SHA-256 and SHA-512 exhibit similar throughput.

CEX7C (eight jobs): HMAC Verify (CSNBHMV) Operations per Second		
Data Length (Bytes)	SHA-256	SHA-512
64	22289	22149
256	22157	22060
1024	22019	21924
4096	21525	21182
64K	2539	2578
1M	171	175

The throughput of eight jobs for CEX7C HMAC verify is on the order of 2.2 to 2.3 times higher than for one job.

3.3.2 CEX7S CCA Coprocessor – Key Management

CEX7C Symmetric Key Management	Ops/s	Ops/s
	(1 job)	(8 jobs)
Key Generate2 (AES 256 bit Operational Key)	8388	20546
Key Import (PIN Verification Key)	7609	18628
Key Export (Cipher Key)	9790	24158
Key Test (Generate 256 bit AES)	11021	29477
Key Translate (MAC key)	8650	21309
Symmetric Key Import (256 bit AES Key enciphered under 2048 bit RSA key)	1350	8004
Symmetric Key Export (256 bit AES Key)	8233	20150
Secure Key Import (256 bit AES Key)	9248	22294
Unique Key Derive (TR-31 Key Block)	5352	12178

3.3.3 CEX7S CCA Coprocessor - Financial Services

The following table gives the performance in terms of operations per second for one CEX7S CCA Coprocessor for some selected financial related services.

CEX7C Financial Services	Ops/s	Ops/s
	(1 job)	(8 jobs)
Clear PIN Generate Alternate (TDES OPINENC + TDES PINGEN keys)	9226	22347
Clear PIN Generate (16 digits) (TDES PINGEN key)	10317	28021
Clear PIN Encrypt	9936	24554
Encrypted PIN Generate	9283	22632
Encrypted PIN Translation (TDES IPINENC key and TDES OPINENC key)	9503	22955
Encrypted PIN Translation (2 DUKPT enabled KEYGENKY keys)	5549	12517

Encrypted PIN Verification (DUKPT enabled KEYGENKY + TDES PINVER key)	6795	15766

3.3.4 CEX7S 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 CEX7C.

CEX7C 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	4660	10596
VFPE Decipher	4689	10698
VFPE Translate	4310	9707
CBC Encipher	9461	23123
CBC Decipher	9833	24134
CBC Translate	9108	21877

3.3.5 CEX7S 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 HCR77D1 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 CEX7S 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 CEX7S. 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 HCR77D1 and one CEX7S 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)	451592
CSNBRNGL (8)	443144
CSNBRNGL (64)	442218
CSNBRNGL (1024)	266408
CSNBRNGL (8192)	44680

3.3.6 CEX7S CCA Coprocessor - PKA Operations

The CEX7S 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 HCR77D1 invoking the operation via the ICSF API according to the PKCS-1.2 Standard. Measurements were performed on a z15 Model 770.

CEX7S CCA Coprocessor PKA Performance

CEX7C on 8561-770 with z/OS V2.4; ICSF FMID HCR77D1			
Public Key Decrypt (PKD), Public Key Encrypt (PKE) Digital Signature Generate (DSG), Digital Signature Verify (DSV) Symmetric Key Import (encrypted with RSA key) (SYI)			
CEX7C	1	1	2
Jobs	1	8	16
	Operations/sec	Operations/sec	Operations/sec
PKD-CRT 1024 bit	4529	18215	37085
PKD-CRT 2048 bit	1615	12082	23533
PKD-CRT 4096 bit	270	1684	3360
PKD-ME 512 bit	4924	19271	39695
PKD-ME 1024 bit	1676	12316	24197

PKE 1024 bit	7904	18080	37602
PKE 2048 bit	6470	17166	34896
PKE 4096 bit	4002	15542	31234
DSG-CRT 1024 bit	4498	17654	36672
DSG-CRT 2048 bit	1617	12104	23812
DSG-CRT 4096 bit	269	1684	3344
DSG-EC BP-192 bit	3298	19561	38540
DSG-EC BP-256 bit	2581	17591	35104
DSG-EC BP-512 bit	974	6547	12909
DSG-EC PC-192 bit	4050	20134	40353
DSG-EC PC-256 bit	3182	19331	38856
DSG-EC PC-521 bit	1450	10333	20091
DSV-CRT 1024 bit	7963	19079	39481
DSV-CRT 2048 bit	6553	18113	37041
DSV-CRT 4096 bit	4056	16571	33273
DSV-EC BP-192 bit	2012	14838	28794
DSV-EC BP-256 bit	1497	10622	21190
DSV-EC BP-512 bit	513	3254	6505
DSV-EC PC-192 bit	2581	18220	36216
DSV-EC PC-256 bit	1885	13999	25720
DSV-EC PC-521 bit	788	5152	10284
SYI-CRT 512 bit	5307	14814	29967
SYI-CRT 1024 bit	4161	14464	27473
SYI-CRT 4096 bit	270	1699	3283
CRT: Chinese Remainder Theorem; ME: Modulus Exponent; EC-BP: Elliptic Curve – BrainPool; EC-PC: Elliptic Curve – Prime Curve			

For the examples in the table above, the PKA cryptographic operation throughput with 2 CEX7C adapters and 16 jobs repetitively requesting the same cryptographic operation is close to 2 times the throughput of one CEX7C adapter and 8 jobs.

PKA Key Generation

The CEX7S 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) and Prime Curve (PC) 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 CEX7S CCA Coprocessor.

CEX7S CCA Coprocessor PKA Key Generation Performance

CEX7C PKA Key Generate	
	Operations/sec (1 job)
RSA CRT 1024 bit	96
RSA CRT 2048 bit	26
RSA CRT 4096 bit	2.4
RSA ME 1024 bit	91
EC BP-192 bit	1039
EC BP-256 bit	769
EC BP-512	259
EC PC-192 bit	1356
EC PC-256 bit	993
EC PC-521 bit	398

3.3.7 CEX7S CCA Coprocessor - PCI-HSM mode

Beginning with the CEX6S 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 CEX7S 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 CEX7S CCA Coprocessor in PCI-HSM mode for some selected symmetric key operations when compliant tagged keys are used.

CEX7C PCI-HSM Financial Services – Examples with Compliant Tagged keys	Ops/s (1 job)	Ops/s (8 jobs)
Key Generate (operational Key Generating Key)	8500	20904

Clear PIN Generate Alternate (TDES OPINENC + TDES PINGEN keys)	7501	17374
Clear PIN Generate (16 digits) (TDES PINGEN key)	8999	21982
Encrypted PIN Translation (TDES IPINENC key and TDES OPINENC key)	7807	17689
Encrypted PIN Translation (2 DUKPT enabled KEY GENKY keys)	3775	8167

3.3.8 CEX7S Enterprise PKCS #11 Coprocessor (CEX7P) – Encryption / Decryption and HMAC operations

z15 with CEX7S cryptographic feature provides the ability to configure the CEX7S in Enterprise PKCS #11 (EP11) Coprocessor mode. ICSF FMID HCR77D1 supports the use of CEX7S 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 CEX7S EP11 Coprocessor. EP11 firmware version 4.7.15 was used for all measurements in this section.

CEX7S Enterprise PKCS #11 Coprocessor AES 128-bit Cipher Block Chaining Encipher

CEX7P: AES 128-bit CBC Encipher		
Data Length (Bytes)	Operations/sec (1 job)	Operations/sec (8 jobs)
64	10034	14720
256	9833	14993
1024	9102	14036
4096	7007	11124
64K	531	882
1M	38.2	63.3

The throughput of eight jobs for CEX7C AES 128-bit CBC encipher is in the range of 1.4 to 1.6 times higher than for one job.

CEX7S Enterprise PKCS #11 Coprocessor TDEA Cipher Block Chaining Encipher with Triple Length Key (168 Bits)

CEX7P: Triple DES CBC Encipher		
Data Length (Bytes)	Operations/sec (1 job)	Operations/sec (8 jobs)
64	10120	15100
256	9827	15019
1024	8978	13922
4096	6591	10185
64K	492	848
1M	35.0	59.5

The throughput of eight jobs for Triple DES CBC encipher is in the range of 1.4 to 1.7 times higher than for one job.

CEX7S Enterprise PKCS #11 Coprocessor Secure Key HMAC Operations

CEX7P (one job): HMAC Generate Operations per Second				
Data Length (Bytes)	SHA-256	SHA-512	SHA3-256	SHA3-512
64	9049	9056	8924	8853
256	8953	8942	8828	8737
1024	8554	8568	8494	8462
4096	7257	7357	7386	7358
64K	596	586	508	518
1M	45.2	45.3	38.9	39.9

CEX7P (eight jobs): HMAC Generate Operations per Second				
Data Length (Bytes)	SHA-256	SHA-512	SHA3-256	SHA3-512
64	13599	13512	12946	12990
256	13597	13341	12871	12855
1024	12949	12993	12543	12339
4096	11313	11476	10976	11011
64K	968.7	943.2	790.7	809.4
1M	72.44	71.97	60.11	61.52

CEX7P HMAC verify using SHA and SHA3 algorithms exhibit similar throughput.

CEX7P (one job): HMAC Verify Operations per Second				
Data Length (Bytes)	SHA-256	SHA-512	SHA3-256	SHA3-512
64	9109	9134	8997	8980
256	8922	9047	8927	8999
1024	8549	8623	8543	8551
4096	7196	7252	7291	7291
64K	591	584	507	518
1M	45.1	45.2	38.8	40.0

CEX7P (eight jobs): HMAC Verify Operations per Second				
Data Length (Bytes)	SHA-256	SHA-512	SHA3-256	SHA3-512
64	13718	13682	13199	13235
256	13695	13507	13162	13018

1024	13179	13046	12542	12626
4096	11293	11222	10993	10979
64K	960	944.9	790.9	805.3
1M	72.6	71.77	59.65	61.5

CEX7P HMAC verify using SHA and SHA3 algorithms exhibit similar throughput.

3.3.9 CEX7S Enterprise PKCS #11 Coprocessor (CEX7P) - PKA Operations

CEX7P Enterprise PKCS #11 Coprocessor PKA Performance

Private Key Decrypt (PKD), Private Key Encrypt (PKE), Private Key Sign (PKS), Private Key Verify (PKV), Wrap Private Key (WPK), Unwrap Private Key (UPK)		
CEX7P	1	1
Jobs	1	8
	Operations/sec	Operations/sec
PKD RSA 1024 bit	3941	12531
PKD RSA 2048 bit	1502	10121
PKD RSA 4096 bit	264	1658
PKE RSA 1024 bit	4832	9585
PKE RSA 2048 bit	3352	7215
PKE RSA 4096 bit	1636	3917
PKS-RSA 1024 bit	3811	11711
PKS-RSA 2048 bit	1466	9746
PKS-RSA 4096 bit	262	1656
PKS BrainPool 192 bit	2818	12190
PKS BrainPool 256 bit	2273	11687
PKS BrainPool 512 bit	921	6484
PKS Prime Curve 192 bit	3374	12374
PKS Prime Curve 256 bit	2737	12207
PKS Prime Curve 521 bit	1333	9556
PKS Dilithium	120	235
PKS Ed448	20.1	39.9
PKS Ed25519	60.2	120

PKV-RSA 1024 bit	3944	7084
PKV-RSA 2048 bit	2556	5018
PKV-RSA 4096 bit	1141	2442
PKV BrainPool 192 bit	1668	8962
PKV BrainPool 256 bit	1339	8288
PKV BrainPool 512 bit	474	3255
PKV Prime Curve 192 bit	2133	9629
PKV Prime Curve 256 bit	1617	8734
PKV Prime Curve 521 bit	714	5138
PKV Dilithium	219	424
PKV Ed448	20.5	40.7
PKV Ed25519	59.0	116
WPK-RSA 1024 bit	4405	8247
WPK-RSA 2048 bit	3135	6249
WPK-RSA 4096 bit	1606	3569
WPK AES	4407	8449
WPK Ed448	6394	8855
WPK Ed25519	6518	9176
UPK-RSA 1024 bit	3124	7838
UPK-RSA 2048 bit	1364	7016
UPK-RSA 4096 bit	257	1642
UPK AES	3222	4601
UPK Ed448	3665	4756
UPK Ed25519	4067	5197

CEX7P IBM PKCS #11 Coprocessor PKA Key Generate Performance

CEX7P PKA Key Generate	
	Operations/sec
RSA CRT 1024 bit	143
RSA CRT 2048 bit	40.8
RSA CRT 4096 bit	3.9

EC Brainpool 192 bit	2246
EC Brainpool 256 bit	1828
EC Brainpool 512 bit	833
EC Prime Curve 192 bit	2547
EC Prime Curve 256 bit	2179
EC Prime Curve 521 bit	1176
EC Dilithium	190
EC Ed448	41.1
EC Ed25519	116

3.3.10 CEX7S Accelerator Performance

The CEX7S 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 V2.4 and ICSF FMID HCR77D1 invoking the operation via the ICSF API according to the PKCS-1.2 Standard. The Accelerator throughput rates in this section were achieved using a single z15 LPAR to send requests to the CEX7S. Higher rates can be achieved by sharing the CEX7S Accelerator between multiple LPARs.

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)

CEX7S Accelerator PKA Performance

CEX7A PKA Key Decrypt (PKD), Public Key Encrypt (PKE), and Digital Signature Verify (DSV)		
8561-770 CPs	4	4
CEX7A Adapters	1	1
Jobs	1	8
	Operations/sec	Operations/sec
PKD CRT 1024 bit	8154	63676

PKD CRT 2048 bit	1912	15232
PKD CRT 4096 bit	278	2222
PKD ME 1024 bit	1927	15363
PKE 1024 bit	29961	206562
PKE 2048 bit	17797	133757
PKE 4096 bit	7117	55572
DSV CRT 1024 bit	29939	200298
DSV CRT 2048 bit	17867	132677
DSV CRT 4096 bit	7118	55729

The first result column of the above table is for measurements where one job was continuously executing the cryptographic operation using one CEX7S Accelerator. As mentioned, the execution of the cryptographic operation in the CEX7S Accelerator is asynchronous to the z15 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 application. 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 CEX7S Accelerator. The increased throughput is because tasks are always available for execution in the CEX7S Accelerator due to the parallel threads that run in the z15 CPs. Thus, the capability of the CEX7S Accelerator for parallel execution of the cryptographic operation can be utilized.

3.4 Crypto Express7S Performance (Linux on Z)

3.4.1 CEX7S CCA Coprocessor (CEX7C) - Encryption/Decryption

This section deals with CEX7S 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 CEX7S 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 z15 Model 8561-703 using either 1 thread that initiates the cryptographic operation or 8 concurrent threads initiating the cryptographic operation. The execution of the cryptographic operation in the CEX7C is asynchronous to

the z15 Central Processor (CP) execution. If only one thread 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 threads requesting cryptographic operations at the same time. The CEX7C adapter's multitasking capability allows for enqueuing and dequeuing of requests in parallel with cryptographic operations being performed. A measurement environment using several parallel threads highlights better the throughput capacity of the CEX7C adapter whereas the 'single thread' measurement environment is better suited to highlight the delay an application may experience waiting for the result of the cryptographic operation performed in the CEX7C. For each cryptographic operation type quoted there is a statement on scalability of the results when multiple threads are used to initiate operations.

The performance numbers are from measurements using Red Hat Enterprise Linux 7.8 (RHEL 7.8) and CEX7S CCA firmware version 7.0.62.

CEX7S CCA Coprocessor AES 128-bit Cipher Block Chaining Encipher

CEX7C: AES 128-bit CBC Encipher		
Data Length (Bytes)	Operations/sec (1 thread)	Operations/sec (8 threads)
64	10402	26911
256	10346	26080
1024	10185	25548
4096	9655	24879
65536	1416	3269
1024000	92.98	213.5

The throughput of eight threads for CEX7C AES 128-bit CBC encryption is on the order of 2.3 times to 2.5 times higher than for one thread.

CEX7S CCA Coprocessor AES 256-bit Cipher Block Chaining Encipher

CEX7C: AES 256-bit CBC Encipher		
Data Length (Bytes)	Operations/sec (1 thread)	Operations/sec (8 threads)
64	10366	26757
256	10298	25899
1024	10124	25251
4096	9520	24636
65536	1381	3243
1024000	90.67	213.8

The throughput of eight threads for CEX7C AES 256-bit CBC encryption is on the order of 2.3 to 2.5 times higher than for one thread.

CEX7S CCA Coprocessor TDEA Cipher Block Chaining Encipher with Triple Length

Key (192 Bits)

CEX7C: Triple DES CBC Encipher		
Data Length (Bytes)	Operations/sec (1 thread)	Operations/sec (8 threads)
64	10943	27787
256	10816	27946
1024	10481	26307
4096	9225	24418
65536	1189	3277
1024000	77.68	213.9

The throughput of eight threads for Triple DES CBC Encipher is 2.5 to 2.7 times higher than the throughput for one thread.

CEX7S CCA Coprocessor Message Authentication Code with Double Length TDES Key (112 Bits)

CEX7C: MAC with double length TDES key		
Data Length (Bytes)	Operations/sec (1 thread)	Operations/sec (8 threads)
64	9489	22034
256	9426	21994
1024	9310	21704
4096	8971	20631
65536	1345	3134
1024000	88.19	207.5

The throughput of eight threads for CEX7C MAC with double length TDES key is on the order of 2.3 times higher than for one thread.

3.4.2 CEX7S CCA Coprocessor – Financial Services Examples

The following table gives the performance in number of operations per second for one CEX7S CCA Coprocessor for some selected financial services.

CEX7C Financial Services Examples	Ops/s (1 thread)	Ops/s (8 threads)
Key Generate2 (AES 256-bit key type CIPHER)	7277	16588
Clear PIN Generate Alternate	9505	22245
Clear PIN Generate	10830	27991
Clear PIN Encrypt	10803	27987
Encrypted PIN Generate	9433	22481
Encrypted PIN Translation (TDES IPINENC key and TDES OPINENC key)	10031	23094
Encrypted PIN Translation (2 DUKPT enabled KEYGENKY keys)	5593	12353

Encrypted PIN Verification (DUKPT enabled KEYGENKY + TDES PINVER keys)	6671	15389

3.4.3 CEX7S 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 CEX7C.

CEX7C VISA Format Preserving Encryption – Personal Access Number (PAN) using VFPE mode or CBC mode		
Operation	Operations/sec (1 thread)	Operations/sec (8 threads)
VFPE Encipher	4531	10127
VFPE Decipher	4541	10213
VFPE Translate	3975	8824
CBC Encipher	9627	22755
CBC Decipher	10290	23893
CBC Translate	9561	22200

3.4.4 CEX7S 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 thread)
CSNBRNG (8)	12907
CSNBRNGL (8)	12822
CSNBRNGL (64)	12679
CSNBRNGL (1024)	11180
CSNBRNGL (8192)	6176

3.4.5 CEX7S CCA Coprocessor - PKA Operations

CEX7C on 8561-703 with RHEL 8.1
Public Key Decrypt (PKD), Public Key Encrypt (PKE)
Digital Signature Generate (DSG), Digital Signature Verify (DSV)

Threads	1	8
	Operations/sec	Operations/sec
PKD-CRT 1024 bit	4735	20470
PKD-CRT 2048 bit	1634	11054
PKD-CRT 4096 bit	269	1677
PKD-ME 512 bit	4931	22195
PKD-ME 1024 bit	1699	11615
PKE-CRT 1024 bit	6073	14399
PKE-CRT 2048 bit	5071	13291
PKE-CRT 4096 bit	3307	11695
DSG-CRT 1024 bit	4730	20323
DSG-CRT 2048 bit	1642	11128
DSG-CRT 4096 bit	270	1688
DSG-EC BrainPool 192 bit	3337	20094
DSG-EC BrainPool 256 bit	2603	16940
DSG-EC BrainPool 512 bit	976	6518
DSG Prime Curve 192 bit	4102	21837
DSG Prime Curve 256 bit	3217	19728
DSG Prime Curve 521 bit	1457	9997
DSV-CRT 1024 bit	6368	15257
DSV-CRT 2048 bit	5289	13856
DSV-CRT 4096 bit	3418	12399
DSV BrainPool 192 bit	2051	13967
DSV BrainPool 256 bit	1522	10426
DSV BrainPool 512 bit	505	3274
DSV Prime Curve 192 bit	2686	16814
DSV Prime Curve 256 bit	1927	13072
DSV Prime Curve 521 bit	782	5180
CRT: Chinese Remainder Theorem; ME: Modulus Exponent		

PKA Key Generation

The CEX7S 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. 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 CEX7S CCA Coprocessor.

CEX7S CCA Coprocessor PKA Key Generation Performance

CEX7C on 8561-703 with RHEL 8.1		
Key Generate operations		
CEX7C	1	1
Threads	1	8
RSA CRT 1024-bit	95.5	204
RSA CRT 2048-bit	25.5	54
RSA CRT 4096-bit	2.38	5.13
EC BP 192	1078	7185
EC BP 256	791	5184
EC BP 512	261	1631
EC PC 192	1422	9169
EC PC 256	1028	6847
EC PC 521	404	2568

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 z15 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 z15 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 z15 to act as a Web Server providing SSL/TLS compliant communication to many 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 z15 the PKD operation will be routed for execution to the CEX7S CCA Coprocessor or CEX7S 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 z15 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 009D TLS_RSA_WITH_AES_256_GCM_SHA384.
- 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 must 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 must be performed the compute intensive PKD operation, then necessary on the server, can be performed in System SSL software or with hardware on a CEX7S Accelerator or CEX7S 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 z15 Central Processors during the measurement interval from the RMF CPU Activity report.

Crypto Utilization %: Average utilization of the CEX7S Accelerator or CEX7S CCA Coprocessor features during the measurement interval from the RMF Crypto report.

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 transaction throughput capacity 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 CEX7S in Accelerator mode increases the SSL/TLS handshake throughput capacity of the CEX7S. Thus, for high SSL/TLS transaction rate environments, Accelerator is the preferred configuration mode for a CEX7S feature.

3.5.2 System SSL with z/OS V2R4 and Cryptographic Support for z/OS V2R1-V2R4 (ICSF FMID HCR77D1)

z15 Model 8561-770 (4 Central Processors)

Caching SID	RSA Key Length	Handshake	Client Auth.	ETR	CPU Util. %	Crypto Util. %

100%	2048	Avoided	no	38098	99.01	NA
no	2048	Software	no	255	100.00	NA
no	2048	1 CEX7C	no	10847	44.40	99.0
no	2048	1 CEX7A	no	13456	55.32	98.6
no	2048	2 CEX7A	yes	14471	91.50	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 most 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 CEX7C 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 42.5 times and reduced the CP utilization by 55%. The CEX7 Coprocessor was 99.0% 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 CEX7 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 CEX7S adapter in Accelerator mode. In this measurement the utilization of the CEX7S Accelerator was 98.6%. Adding additional CEX7 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 CEX7 Accelerator was added to the system configuration for this measurement. The average utilization of the 2 Accelerators was 100%.