IBM zEnterprise EC12 Performance of Cryptographic Operations

(Cryptographic Hardware: CPACF, CEX4S)

IBM Corporation
Marketing Communications, Server Group
Route 100
Somers, NY 10589
U.S.A.
Produced in the United States of America
All Rights Reserved

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

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY, 10504-1785 USA.

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

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

Table of Contents

Preface

The performance information presented in this publication was measured on IBM™ zEnterprise EC12™ 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 are adequate and can therefore be used safely.

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

1. Introduction

The purpose of this publication is to provide performance information to the user of cryptographic services on zEnterprise EC12 (zEC12). zEC12 supports the following cryptographic hardware features:

1. Central Processor Assist for Cryptographic Function (CPACF).
2. Crypto Express3 (CEX3) feature.
3. Crypto Express4S (CEX4S) feature.

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

The Crypto Express3 feature is supported on zEC12, however this document does not present performance information for CEX3. Performance information for CEX3 on z196 can be found at IBM System z196 Enterprise Class Performance of Cryptographic Operations The Crypto Express3 is the same feature which is available in System z196™, and is expected to exhibit similar performance characteristics when installed in zEC12.

The CEX4S feature differs from the CEX3 feature in that it contains only one PCIe adapter per feature. Using the HMC console, the CEX4S 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 (SSL) acceleration).

All CEX4S data presented in this document is from actual measurements with one or more CEX4S features configured as denoted in each section.

2. Cryptographic Hardware Supported on zEC12

2.1 Central Processor Assist for Cryptographic Function (CPACF)

The CPACF delivers cryptographic support for Data Encryption Standard (DES), Triple DES (TDES), and Advanced Encryption Standard (AES) encryption/decryption, as well as Secure Hash Algorithm (SHA). zEC12 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 DES, TDES and AES functions require enablement of the CPACF for export control. The CPACF functions for DES, TDES, AES and SHA can be invoked by problem state instructions defined by an extension of the zEC12 architecture called Message Security Assist (MSA). Support is also available for z/OS via Cryptographic Support for z/OS V1R12-R13 (ICSF FMID HCR77A0) web deliverable.

zEC12 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 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 hardware of the CPACF that performs the symmetric key operations (DES; TDES; AES) and SHA functions operates synchronously to the 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 Express3 (CEX3) Feature

The Crypto Express3 feature is supported on zEC12, however this document does not present performance information for CEX3. Performance information for CEX3 on z196 can be found at IBM System z196 Enterprise Class Performance of Cryptographic Operations The Crypto Express3 is the same feature which is available in System z196^{TM}, and is expected to exhibit similar performance characteristics when installed in zEC12.

2.3 Crypto Express4S (CEX4S) Feature

The Crypto Express4S 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 Secure Sockets Layer (SSL) acceleration) modes in a single feature. Using the HMC console, the CEX4S feature can be configured to function as a CCA Coprocessor, a PKCS#11 Coprocessor, or an Accelerator. The Crypto Express4S feature is a follow-on to the Crypto Express3 feature with updates to provide additional function and improved performance.

Like the z196, z114 and z10 machines, up to 16 cryptographic engines can be installed on a zEC12. On z196, z114 and z10 machines, each CEX2 and CEX3 feature provided two cryptographic engines, so it took just eight features to fully populate the CEC. On the zEC12, each CEX4S provides a single cryptographic engine, therefore up to 16 features can be installed.

When configured in CCA Coprocessor mode (CEX4C), the CEX4S feature supports:

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

When configured in Enterprise PKCS#11 Coprocessor mode (CEX4P), the CEX4S feature supports:

- Secure key PKCS#11 operations

The CEX4S 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 DES, TDES, AES, RSA, and SHA cryptographic operations in a secure environment. The CEX4S 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 CEX4S. The Master Keys are always kept in battery backed-up memory within the tamper-protected boundary of the CEX4S Coprocessor and are destroyed if physical tampering is detected.

The CEX4S in CCA Coprocessor mode also supports the ‘clear key’ PKA operations that currently are predominantly used to support SSL protocol communications.

When configured in Enterprise PKCS#11 Coprocessor mode, the CEX4S feature implements an IBM version of the PKCS#11 standard and provides hardware support for PKCS#11 operations utilizing secure keys.

When configured in Accelerator mode (CEX4A), the CEX4S 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 CEX4S Accelerator and thus increase system throughput. The CEX4S in Accelerator mode works in ‘clear key’ mode only.

The Crypto Express4S executes its cryptographic operations asynchronously to a Central Processor (CP) operation in the zEC12. A CP requesting a cryptographic operation from the CEX4S uses the message queuing protocol to communicate with the CEX4S. After enqueuing a request to the CEX4S, the host operating system will dispense the task that has enqueued the cryptographic operation and dispatch another task. Thus, processing of the cryptographic operation in the CEX4S will work in parallel to other tasks being executed in a zEC12 CP. With z10 GA2 architecture level and beyond, support was added for Cryptographic AP-Queue I/O interrupts. This function is exploited by z/OS V1R13 and ICSF FMID HCR77A0. With this support, when a cryptographic operation completes on the CEX4S, an interrupt will be presented to ICSF. ICSF will then dequeue the result from the CEX4S and return it to the requesting application. All CEX4S measurement results presented in this paper are from systems utilizing the new Cryptographic AP-Queue I/O interrupt support. For each CEX4S, 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 Express4S, several operations can be worked on in parallel.

For zEC12, the Crypto Express4S works with ICSF FMID HCR77A0 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 Express4S. For unique customer cryptographic application requirements the Crypto Express4S in CCA Coprocessor mode provides for user-defined extensions (UDX) to the CCA interface.

In a System 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 zEC12 using the z/OS System Control Program, CEX4S 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.

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 1 Release 13 (z/OS V1.13) and ICSF FMID HCR77A0 except when stated otherwise.

All measurements were performed on an IBM zEC12 Model 2827-HA1. 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 zEC12 Model 2827-HA1 with only one dedicated CP.

For the cryptographic operations that can be used with a variable length of data such as Data Encryption Algorithm (DEA) and 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 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 System z Assembler Language issuing the System z Message Security Assist (MSA) Architecture cryptographic operation instructions as indicated with each group.

The data quoted is from test cases run on a zEC12 Model 2827-HA1, however, using only one of the CPACFs. Scalability measurements were also taken using 2 CPACFs (not quoted) and in all cases the throughput with 2 CPACFs was two times the throughput of 1 CPACF. zEC12 has one CPACF for every Central Processor (CP), therefore, CPACF encryption throughput scales with the number of CPs in the system. Scalability measurements had 2 dedicated CPs and 2 jobs that initiated the cryptographic operation.

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

3.2.1.1 CPACF MSA Architecture Interface - Clear Key Mode

DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits)
(System z Message Security Assist Architecture instruction: KMC-DEA clear key)

Native: Single DES CBC Encipher (KMC-DEA clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	8570163	548.4
256	3228369	826.4
1024	914931	936.8
4096	239104	979.3
64K	14880	975.1
1M	927.7	972.8

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

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

Native: Triple DES CBC Encipher (KMC-TDEA clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4542165	290.6
256	1390915	356.0
1024	370517	379.4
4096	94468	386.9
64K	5899	386.6
1M	368.2	386.1

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

AES Cipher Block Chaining Encipher with 128 Bit Key

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

Native: AES - 128 bit CBC Encipher (KMC-AES clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	10067527	644.3
256	4583934	1173
1024	1424870	1459
4096	385303	1578
64K	23960	1570
1M	1491	1563

AES-128 Cipher Block Chaining Decipher has similar performance characteristics as the Encipher operation.

AES Cipher Block Chaining Encipher with 256 Bit Key

(System 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	8884305	568.5
256	3750053	960.0
1024	1122332	1149
4096	298597	1223
64K	18583	1217
1M	1156	1212

AES-256 Cipher Block Chaining Decipher has similar performance characteristics as the Encipher operation.

Compute Message Authentication Code (MAC) with DEA Single Length Key (56 Bits)
(System z Message Security Assist Architecture instruction: KMAC-DEA clear key)

Native: MAC with single DES (KMAC-DEA clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	9849848	630.3
256	3390275	867.9
1024	935602	958.0
4096	240247	984.0
64K	15005	983.3
1M	935.1	980.6

Compute Message Digest SHA-1
(System z Message Security Assist Architecture instruction: KLMD-SHA-1 clear key)

Native: SHA-1(KLMD-SHA-1 clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	5132656	328.4
256	2528028	647.1
1024	833215	853.2
4096	226249	926.7
64K	14354	940.7
1M	895.9	939.4

Compute Message Digest SHA-512
(System z Message Security Assist Architecture instruction: KLMD-SHA-512 clear key)

Native: SHA-512(KLMD-SHA-512 clear key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4428458	283.4
256	2082182	533.0
1024	793906	812.9
4096	228553	936.1
64K	14706	963.7
1M	918.5	963.1

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 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 or KMAC instruction. The PCKMO instruction execution is not included in the results.

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

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

Native: Single DES CBC Encipher (KMC-DEA protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4567857	292.3
256	2452431	627.8
1024	838372	858.4
4096	233210	955.2
64K	14524	951.8
1M	905.2	949.1

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

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

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

Native: Triple DES CBC Encipher (KMC-TDEA protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	3124009	199.9
256	1218835	312.0
1024	356309	364.8
4096	93381	382.4
64K	5834	382.3
1M	364.1	381.8

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

AES Cipher Block Chaining Encipher with 128 Bit Key

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

Native: AES - 128 bit CBC Encipher (KMC-AES protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	5005353	320.3
256	3140023	803.8
1024	1242898	1272
4096	370539	1517
64K	23041	1510
1M	1436	1506

AES-128 Cipher Block Chaining Decipher has similar performance characteristics as the Encipher operation.

AES Cipher Block Chaining Encipher with 256 Bit Key

(System 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	4700328	300.8
256	2701862	691.6
1024	1008131	1032
4096	289741	1186
64K	18028	1181
1M	1123	1178

AES-256 Cipher Block Chaining Decipher has similar performance characteristics as the Encipher operation.

Compute Message Authentication Code (MAC) with DEA Single Length Key (56 Bits)
(System z Message Security Assist Architecture instruction: KMAC-DEA protected key)

Native: MAC with single DES (KMAC-DEA protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	4967758	317.9
256	2531931	648.1
1024	854550	875.0
4096	234118	958.9
64K	14629	958.7
1M	912.5	956.8

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

The data quoted is from test cases run on a zEC12 Model HA1, 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 jobs that initiated the cryptographic operation. The throughput with 4 CPACFs was 3 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 ‘Native’ test cases. This is expected because of the additional ICSF path length. As the data length increases, the ICSF path length is a less dominant factor and the throughput for large data lengths is nearly the same as for the ‘Native’ test case.

3.2.2.1 CPACF ICSF API - Clear Key Operations

DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits) - ICSF API
(System z Message Security Assist Architecture instruction: KMC-DEA clear key)

ICSF API: Single DES CBC Encipher (KMC-DEA clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	807352	51.6
256	698265	178.7
1024	451348	462.1
4096	188019	770.1
64K	14599	956.8
1M	926.5	971.5

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

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

ICSF API: Triple DES CBC Encipher (KMC-TDEA clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	745691	47.7
256	542922	138.9
1024	260821	267.0
4096	85165	348.8
64K	5853	383.6
1M	367.9	385.8

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

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

ICSF API: AES-128 Encipher (KMC-AES clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	782738	50.0
256	715261	183.1
1024	530294	543.0
4096	263940	1081
64K	23229	1522
1M	1487	1560

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

AES Cipher Block Chaining Encipher with 256 Bit Key - ICSF API (System z Message Security Assist Architecture instruction: KMC-AES clear key)

ICSF API: AES-256 Encipher (KMC-AES clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	792134	50.6
256	704826	180.4
1024	488828	500.5
4096	221501	907.2
64K	18144	1189
1M	1155	1211

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

Compute Message Digest SHA-1 - ICSF API

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

ICSF API: SHA-1(KLMD-SHA-1 clear key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	519647	33.2
256	470619	120.4
1024	340620	348.7
4096	162066	663.8
64K	13941	913.6
1M	892.7	936.0

Compute Message Digest SHA-512 - ICSF API
(System z Message Security Assist Architecture instruction: KLMD-SHA-512)

ICSF API: SHA-512(KLMD-SHA-512 clear key) one job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	507703	32.4
256	447686	114.6
1024	331598	339.5
4096	162801	666.8
64K	14272	935.3
1M	915.1	959.5

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 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 CEX4S 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 4.2.1.2). 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 CEX4S feature (Section 4.3.1). 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 CEX4S 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.

DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits) - ICSF API
(System z Message Security Assist Architecture instruction: KMC-DEA protected key)

ICSF API: Single DES CBC Encipher (KMC-DEA protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	376011	24.0
256	350657	89.7
1024	275855	282.4
4096	147818	605.4
64K	13953	914.4
1M	902.7	946.6

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

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

ICSF API: Triple DES CBC Encipher (KMC-TDEA protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	367009	23.4
256	309034	79.1
1024	191528	196.1
4096	75781	310.4
64K	5750	376.8
1M	364.4	382.1

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

AES Cipher Block Chaining Encipher with 128 Bit Key - ICSF API

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

ICSF API: AES-128 Encipher (KMC-AES protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	385387	24.6
256	367602	94.1
1024	311475	318.9
4096	195370	800.2
64K	21714	1423
1M	1426	1495

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

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

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

ICSF API: AES-256 Encipher (KMC-AES protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	383752	24.5
256	362131	92.7
1024	295411	302.5
4096	170212	697.1
64K	17186	1126
1M	1118	1172

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

3.3 Crypto Express4S Performance

The Crypto Express4S feature is designed to satisfy high-end server security requirements. The Crypto Express4S feature is configurable and can be defined for secure key encrypted transactions (CCA Coprocessor – the default, or Enterprise PKCS#11 Coprocessor) or SSL acceleration (Accelerator). Like its predecessors, the Crypto Express4S 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 zEC12 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 RSA operations in ‘clear key’ mode, providing security rich communication for Web site-based applications which utilize the SSL or TLS protocol. It is current practice to execute the public key operation, incurred during set up of an SSL session, in ‘clear key’ mode.

The connection of the CEX4S feature via the PCIe bus to the zEC12 Central Processors (CPs) incurs latency and data transmission time. Because of this connection to the zEC12 CPs, the CEX4S operates asynchronously to the zEC12 CPs.

There can be a maximum of 16 CEX4S features in a zEC12, each CEX4S feature containing one PCIe adapter.

3.3.1 CEX4S CCA Coprocessor (CEX4C) Symmetric Key Performance - Encryption/Decryption and MAC Operations

This chapter deals with CEX4S CCA Coprocessor cryptographic operations with a user supplied length of data as, e.g., DES or AES operations.

All test cases are written in System z Assembler Language issuing an API call to ICSF for the cryptographic operation. ICSF will resolve the API call and handle the communication with the CEX4S 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 CEX4S CCA Coprocessor is considerably less than the throughput for the corresponding operations using the CP Assist for Cryptographic Function (CPACF) hardware. For this type of cryptographic operation the CEX4S 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 zEC12 Model 2827-HA1 using 1 job that initiates the cryptographic operation. For each cryptographic operation type quoted there is a statement on scalability of the results if multiple jobs are used to initiate operations. The increase of measured throughput using 5 jobs is exemplified for the Single DES CBC Encipher operation.

The performance numbers are from measurements using z/OS V1.13 and ICSF FMID HCR77A0.

All CEX4S results presented in this paper are from measurements utilizing the Cryptographic AP-Queue I/O interrupt support.

CEX4S CCA Coprocessor DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits)

CEX4C (one job): Single DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	2575	164.8
256	2535	649.0
1024	2382	2439
4096	2049	8393
64K	234.8	15388
1M	15.45	16204

The above table provides measurement results for an environment where one job was continuously executing the cryptographic operation using one CEX4S CCA Coprocessor card. As mentioned, the execution of the cryptographic operation in the CEX4C card is asynchronous to the zEC12 Central Processor (CP) execution. As only one job is run on the CP the next cryptographic operation is started only when the result of the previous cryptographic operation has been received by the CP. Thus there is a considerable delay before the next cryptographic operation can be initiated by the host CP. This inefficiency is removed when the host program consists of several jobs requesting cryptographic operations at the same time. The CEX4C 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 CEX4C adapter whereas the ‘single job’ measurement environment is better suited to highlight the delay an application experiences waiting for the result of the cryptographic operation performed in the CEX4C.

CEX4C (five jobs): Single DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	3364	215.3
256	3355	859.0
1024	3255	3333
4096	3011	12336
64K	360.6	23638
1M	23.89	25054

The throughput with N CEX4C adapters with a sufficient number of jobs repetitively requesting the same cryptographic operation such as Single DES, Triple DES, AES-128, AES-256 and Single DES Message Authentication (MAC) (see the following tables) is close to N times the throughput of one CEX4C adapter with five jobs (as exemplified above).

CEX4S CCA Coprocessor DEA Cipher Block Chaining Encipher with Triple Length Key (168 Bits)

CEX4C (one job): Triple DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	2513	160.8
256	2473	633.3
1024	2316	2372
4096	1878	7692
64K	204.0	13372
1M	13.35	14000

The throughput for five jobs for CEX4C TDES is on the order of 1.2 times to 1.7 times higher than for one job.

CEX4S CCA Coprocessor AES 128-bit Cipher Block Chaining Encipher

CEX4C (one job): AES 128-bit CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	2313	148.0
256	2258	578.2
1024	2007	2055
4096	1499	6142
64K	149.8	9822
1M	9.71	10191

The throughput for five jobs for CEX4C AES 128-bit CBC encryption is on the order of 1.2 times to 1.3 times higher than for one job.

CEX4S CCA Coprocessor AES 256-bit Cipher Block Chaining Encipher

CEX4C (one job): AES 256-bit CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	2323	148.6
256	2265	580.0
1024	2014	2062
4096	1483	6074
64K	147.1	9645
1M	9.53	9995

The throughput for five jobs for CEX4C AES 256-bit CBC encryption is on the order of 1.3 times higher than for one job.

CEX4S CCA Coprocessor Message Authentication Code with DEA Single Length Key (56 Bits)

CEX4C (one job): MAC with single DES		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	2612	167.1
256	2595	664.4
1024	2515	2576
4096	2263	9269
64K	198.4	13008
1M	12.71	13337

The throughput for five jobs for CEX4C MAC is on the order of 1.2 to 1.4 times higher than for one job.

3.3.2 CEX4S CCA Coprocessor Symmetric Key Performance - Diverse Operations

The following table gives the performance in maximum number of operations per second for one CEX4S CCA Coprocessor for some selected symmetric key operations.

CEX4C Symmetric Key Operations - Examples	Ops/s (1 job)	Ops/s (5 jobs)
Key Generate (operational DES KEY GENKY key)	1650	1869
Clear PIN Generate Alternate (DES OPINENC + DES PINGEN keys)	1935	2270
Clear PIN Generate (16 digits) (DES PINGEN key)	2499	3185
Encrypted PIN Translation (DES IPINENC key and DES OPINENC key)	2136	2552
Encrypted PIN Translation (2 UKPT enabled KEY GENKY keys)	890.9	942.8
Encrypted PIN Verification (UKPT enabled KEY GENKY + DES PINVER keys)	1175	1270

The throughput with N CEX4C adapters with a sufficient number of jobs repetitively requesting the same cryptographic operation for the examples in the table above is close to N times the throughput of one CEX4C adapter with 5 jobs.

3.3.3 CEX4S 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 performance of random number generation was substantially improved in ICSF FMID HCR77A0 with the implementation of a random number data cache. The random number data cache resides in private storage within the ICSF address space. The cache is allocated and filled when the ICSF address space is initialized. This support allows ICSF FMID HCR77A0 to satisfy random number requests from an internal private cache, eliminating the delay associated with sending the request to the CEX4 adapter. When the cache depletion threshold is reached, ICSF FMID HCR77A0 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 performance in maximum number of operations per second for random number generation of various sizes when ICSF FMID HCR77A0 and one CEX4S CCA Coprocessor are used to maintain the cache in a non-FIPS certified environment.

Random Data Request Size (bytes)	Operations/sec (1 job)
RNG-8	640283
RNGL-1	639404
RNGL-64	599553
RNGL-1K	241337
RNGL-8K	32938

3.3.4 CEX4S CCA Coprocessor PKA Performance

The CEX4S 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 512, 1024, 2048 and 4096 bits.

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

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

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

The performance numbers are from measurements with z/OS V1.13 including ICSF FMID HCR77A0 invoking the operation via the ICSF API according to the PKCS-1.2 Standard. Measurements were performed on a zEC12 Model HA1.

CEX4S CCA Coprocessor PKA Performance

CEX4C on 2827-HA1 with z/OS V1.13; ICSF FMID HCR77A0				
Public Key Decrypt (PKD), Public Key Encrypt (PKE)				
Digital Signature Generate (DSG), Digital Sign Verify (DSV)				
Symmetric Key Import (encrypted with RSA key) (SYI)				
CEX4C	1	1	2	4
Jobs	1	5	10	20
	Operations/sec	Operations/sec	Operations/sec	Operations/sec
PKD-CRT 1024 bit	1216	2281	4475	8876
PKD-CRT 2048 bit	362	923	1846	3683
PKD-CRT 4096 bit	45	83	164	329
PKD-ME 512 bit	1326	2565	5041	9973
PKD-ME 1024 bit	642	1853	3733	7442

PKE 512 bit	1796	2112	4138	8136
PKE 1024 bit	1685	2005	3944	7783
PKE 2048 bit	1331	1663	3247	6427
PKE 4096 bit	309	422	845	1689
DSG-CRT 1024 bit	1209	2254	4431	8777
DSG-CRT 2048 bit	366	932	1864	3724
DSG-CRT 4096 bit	48	97	194	389
DSV-CRT 1024 bit	2023	2485	4895	9691
DSV-CRT 2048 bit	1722	2287	4497	8922
DSV-CRT 4096 bit	340	422	845	1689
SYI-CRT 512 bit	1330	1653	3271	6499
SYI-CRT 1024 bit	1004	1584	3135	6229
SYI-CRT 4096 bit	45	88	181	327

The PKA cryptographic operation throughput with N CEX4C adapters with a sufficient number of jobs repetitively requesting the same cryptographic operation for the examples in the table above is close to N times the throughput of one CEX4C adapter with 5 jobs (as stated above).

PKA Key Generation

The CEX4S 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 CEX4S CCA Coprocessor.

CEX4S CCA Coprocessor PKA Key Generation Performance

CEX4C PKA Key Generate	
	Operations/sec
RSA CRT 512 bit	6.71
RSA CRT 1024 bit	4.04
RSA CRT 2048 bit	1.74
RSA CRT 4096 bit	0.28
RSA ME 512 bit	8.61
RSA ME 1024 bit	5.67
EC BP-192 bit	35.88
EC BP-256 bit	19.39
EC BP-512	3.74
EC PC-192 bit	104.2
EC PC-256 bit	41.69
EC PC-512 bit	9.65

3.3.5 CEX4S Enterprise PKCS#11 Coprocessor (CEX4P)

New on zEC12 with CEX4S cryptographic cards is the ability to configure the CEX4S in Enterprise PKCS#11 (EP11) Coprocessor mode. ICSF FMID HCR77A0 supports the use of CEX4S 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 CEX4 EP11 Coprocessor.

3.3.5.1 CEX4S Enterprise PKCS#11 Coprocessor (CEX4P) Secure Key HMAC Operations

CEX4P (1 job): HMAC Generate Operations per Second			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	2254	2051	1622
256	2148	1920	1424
1024	1771	1540	1020
4096	1048	853	477
64K	63.9	54.1	31.8
1M	4.36	3.65	2.08

CEX4P (1 job): HMAC Verify Operations per Second			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	2382	2134	1672
256	2289	2016	1473
1024	1940	1652	1069
4096	1213	959	508
64K	63.8	54.0	31.7
1M	4.36	3.65	2.08

3.3.5.2 CEX4S Enterprise PKCS#11 Coprocessor (CEX4P) Secure Key PKA Operations

CEX4P Enterprise PKCS#11 Coprocessor Secure Key PKA Performance

	Operations/sec
Private Key Sign 1024 bit	575
Private Key Sign 2048 bit	271
Private Key Verify 1024 bit	885
Private Key Verify 2048 bit	586
Wrap Private Key 1024 bit	3306
Wrap Private Key 2048 bit	909
Unwrap Private Key 1024 bit	471
Unwrap Private Key 2048 bit	246

CEX4P IBM PKCS#11 Coprocessor PKA Key Generate Performance

CEX4P PKA Key Generate	
	Operations/sec
RSA CRT 1024 bit	1.296
RSA CRT 2048 bit	0.418
EC Brainpool 192 bit	36.85
EC Brainpool 256 bit	19.65
EC Brainpool 384 bit	7.478
EC Prime Curve 192 bit	104.1
EC Prime Curve 256 bit	46.10
EC Prime Curve 521 bit	9.713

3.3.6 CEX4S Accelerator Performance

The CEX4S Accelerator mode is designed to offer fast RSA algorithm cryptographic operations. The performance is listed for RSA key modulus lengths of 512, 1024 and 2048 bits. The performance numbers are from measurements with z/OS V1.13 including ICSF FMID HCR77A0 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)

CEX4S Accelerator PKA Performance

CEX4A PKA Key Decrypt (PKD), Public Key Encrypt (PKE), and Digital Signature Verify (DSV)				
2827 CPs	4	4	4	4
CEX4A Adapters	1	1	2	4
Jobs	1	8	16	32
	Operations/sec	Operations/sec	Operations/sec	Operations/sec
PKD CRT 512 bit	5328	15354	30825	57163
PKD CRT 1024 bit	2545	6236	12325	24420
PKD CRT 2048 bit	436	902	1804	3605
PKD ME 512 bit	2565	6296	12492	22662
PKD ME 1024 bit	848	1805	3615	7213
PKE 512 bit	9208	18937	40283	77325
PKE 1024 bit	8701	19557	41643	82661
PKE 2048 bit	5882	16472	35208	64613
DSV CRT 512 bit	9186	18944	40255	79828

DSV CRT 1024 bit	8664	19530	41488	82563
DSV CRT 2048 bit	5860	16368	35504	69694

The first result column of the above table is for measurements where one job was continuously executing the cryptographic operation using one CEX4S Accelerator card. As mentioned, the execution of the cryptographic operation in the CEX4S Accelerator is asynchronous to the zEC12 Central Processor (CP) execution. As only one job is run on the CP the next cryptographic operation is started only when the result of the previous cryptographic operation has been received by the CP. The single job measurement indicates the delay an application would experience waiting for the result of the cryptographic operation.

The second result column of the above table is for measurements where eight jobs were continuously executing the same cryptographic operation using one CEX4S Accelerator card. The increased throughput is due to the fact that tasks are always available for execution in the CEX4S Accelerator card due to the parallel threads that run in the zEC12 CPs. Thus the capability of the CEX4S Accelerator card for parallel execution of the cryptographic operation can be utilized.

The third and fourth columns show the scalability of the throughput when multiple CEX4A adapters are used in one zEC12.

3.4 SSL Handshake Performance

3.4.1 SSL Protocol based Communication

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

Executing the SSL/TLS protocols for a Server (or Client) on a zEC12 will result in a series of cryptographic operations. In the z/OS environment, SSL will either invoke the available cryptographic hardware directly (via the MSA instructions), or use the hardware via ICSF (for the PKA operations) or use its own software routines to perform the 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 environment as the maximum number of SSL handshakes the zEC12 can provide as a server within the given system constraints and assess the utilization of the measured system.

The intent for providing capacity information in the SSL environment is to demonstrate the capabilities of a zEC12 to act as a Web Server providing SSL-compliant communication to a large number of clients. For this purpose the maximum number of SSL connects and data exchanges per second made between the server and all clients are provided for different 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 environments include only the processing required for the SSL protocol handshake and some data exchange. Explicitly excluded is the processing for the 'business transaction' that in a normal environment would be initiated in the server on behalf of the client's request. As most SSL protocol-based measurements in this report are limited by the processing capacity of the server, in a 'real life' environment the processing for the business transaction would reduce the number of handshakes considerably.

The SSL 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 on the Server side a PKA Private Key operation. This Public Key Decrypt (PKD) on the Server can be performed either in software or may be assisted by cryptographic hardware. In the presented measurements on the zEC12 the PKD operation will be routed for execution to the CEX4S CCA Coprocessor or CEX4S Accelerator adapter, 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 protected communications.

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

- Measurements were performed on a zEC12 with 4 CPs as a Server.
- The performance data is for the server only. The server was driven to near maximum utilization by increasing the number of client systems (on separate systems) or until some system resource became limited.

- The TLS 1.0 protocol was used
- The key length for the Public Key operation is 1024 bits. The SSL data encryption is AES-128 bit and SHA-1 cipher except when stated otherwise. The symmetric key data encryption for AES-128 and SHA-1 is executed in CPACF hardware.
- One packet of 2048 Bytes is exchanged with each transaction.
- The SSL 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 CEX4S Accelerator or CEX4S CCA Coprocessor feature.
Client Authentication: The authentication of the Client is optional.
External Throughput Rate (ETR): Number of transactions performed per second.
CPU Utilization %: Average utilization of the zEC12 Central Processors during the measurement interval.
Crypto Utilization %: Average utilization of the CEX4S Accelerator or CEX4S CCA Coprocessor features during the measurement interval.

As mentioned, the measurements for the SSL 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 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 protected transaction. Furthermore, the measurement results show the throughput with one CEX4S Accelerator adapter being on the order of 2.2 times the throughput with one CEX4S CCA Coprocessor adapter in the measured SSL environment. Thus for high SSL transaction rate environments, Accelerator is the preferred configuration mode for a CEX4S feature.

The resource consumption in system processing power for one SSL protocol handshake is on the order of 1/17,000 of the system (see table below) in the z/OS environment for a zEC12

Model HA1 with 4 Central Processors and 4 CEX4S features (4 CEX4S Accelerators).

If a currently unsecured transaction is changed to be ‘secured’ by an SSL protocol then the maximum transaction rate which can be achieved on a given system would be reduced by the amount of processing that is required by the secure protocol.

3.4.2 SSL Performance - System SSL with z/OS V1.13 and Cryptographic Support for z/OS V1R12-R13 (ICSF FMID HCR77A0)

zEC12 Model 2827-HA1 (4 Central Processors)

Caching SID	Handshake	Client Auth.	ETR	CPU Util. %	Crypto Util. %
100%	Avoided	no	24,808	98.44	NA
no	Software	no	1,378	100	NA
no	4 CEX4C	no	9,003	56.29	99.4
no	4 CEX4A	no	17,493	98.34	87.8
no	4 CEX4A	yes	11,477	98.61	79.1

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

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

Using the CEX4C cryptographic hardware compared to using System SSL Software (second and third rows in the above table) produces an increase in throughput (number of SSL handshakes per second) of 6.5 times and reduces the CP utilization by 44%. The CP utilization of this measurement only reached 56.29% because the 4 CEX4C cards were fully utilized at 99.4% and limited the throughput capacity of this configuration. Adding additional CEX4C cards to this environment would allow for a higher ETR.

The fourth row shows that a higher ETR can be achieved with the same 4 CEX4S adapters configured in Accelerator mode. In this measurement the average utilization of the CEX4A adapters was 87.8%, indicating that the 4 CEX4A adapters could process more than 19,000 SSL handshakes before reaching 100% utilization. The 17,493 ETR represents close to the maximum number of SSL handshakes that can be supported with this configuration because the 4 Central Processors are 98% utilized.

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