IBM z13 Performance of Cryptographic Operations

(Cryptographic Hardware: CPACF, CEX5S)

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 improvements equivalent to the performance rates stated here.

Table of Contents

Preface

The performance information presented in this publication was measured on IBM™ z13™ 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 z13. z13 supports the following cryptographic hardware features:

1. Central Processor Assist for Cryptographic Function (CPACF).
2. Crypto Express5S (CEX5S) 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).

CEX5S is a PCIe adapter card that contains a Cryptographic Coprocessor Subsystem housed within a FIPS Level 4 physically secure enclosure (security module). It is planned for use in System z, Power Systems and as a Machine Type Model (MTM) in System x 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). CEX5S is a follow-on to CEX4S with up to 2X performance, new crypto algorithms, concurrent upgrade, improved RAS, and addresses CEX4S end of life components.

Using the HMC console, the CEX5S 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 CEX5S data presented in this document is from actual measurements with one or more CEX5S features configured as denoted in each section.

2. Cryptographic Hardware Supported on z13

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). z13 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 z13 architecture called Message Security Assist (MSA). Support is also available for z/OS® via Enhanced Cryptographic Support for z/OS V1.13-V2.1 (ICSF FMID HCR77B0) web deliverable.

z13 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).

2.3 Crypto Express5S (CEX5S) Feature

The Crypto Express5S 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 CEX5S feature can be configured to function as a CCA Coprocessor, a PKCS #11 Coprocessor, or an Accelerator. The Crypto Express5S feature is a follow-on to the Crypto Express4S feature with updates to provide additional function and improved performance.

Up to 16 Crypto Express5S features can be installed on a z13.

When configured in CCA Coprocessor mode (CEX5C), the CEX5S 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 (CEX5P), the CEX5S feature supports:
- Secure key PKCS #11 operations

The CEX5S 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 CEX5S 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 CEX5S. The Master Keys are always kept in battery backed-up memory within the tamper-protected boundary of the CEX5S Coprocessor and are destroyed if physical tampering is detected.

The CEX5S 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 CEX5S 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 (CEX5A), the CEX5S 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 CEX5S Accelerator and thus increase system throughput. The CEX5S in Accelerator mode works in ‘clear key’ mode only.

The Crypto Express5S executes its cryptographic operations asynchronously to a Central Processor (CP) operation in the z13. A CP requesting a cryptographic operation from the CEX5S uses the message queuing protocol to communicate with the CEX5S. After enqueuing a request to the CEX5S, 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 CEX5S will work in parallel to other tasks being executed in a z13 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 V2.1 and ICSF FMID HCR77B0. With this support, when a cryptographic operation completes on the CEX5S, an interrupt will be presented to ICSF. ICSF will then dequeue the result from the CEX5S and return it to the requesting application. All CEX5S measurement results presented in this paper are from systems utilizing the Cryptographic AP-Queue I/O interrupt support. For each CEX5S, 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 Express5S, several operations can be worked on in parallel.

For z13, the Crypto Express5S works with ICSF FMID HCR77B0 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 Express5S. For unique customer cryptographic application requirements the Crypto Express5S 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 z13 using the z/OS System Control Program, CEX5S 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 2 Release 1 (z/OS V2.1) and ICSF FMID HCR77B0.

All measurements were performed on an IBM z13 Model 2964-N96. 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 z13 Model 2964-N96 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 z13 Model 2964-N96, 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. z13 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 2 dedicated CPs and 2 concurrent 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	9449230	604.7
256	5187701	1328.0
1024	1846515	1890.8
4096	516673	2116.2
64K	32214	2111.2
1M	2019	2118.0

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	6434447	411.8
256	2460542	629.8
1024	732401	749.9
4096	190946	782.1
64K	11956	783.5
1M	747.1	783.4

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	9795690	626.9
256	6275398	1606
1024	2549222	2610
4096	756453	3098
64K	47591	3118
1M	2961	3105

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	9291255	594.6
256	5376558	1376
1024	2012023	2060
4096	574147	2351
64K	36090	2365
1M	2251	2360

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	9926718	635.3
256	5297466	1356
1024	1859005	1903
4096	517249	2118
64K	32517	2131
1M	2032	2131

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	6980990	446.7
256	4045051	1035
1024	1499265	1535
4096	428352	1754
64K	27207	1783
1M	1701	1783

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	7517956	481.1
256	5222970	1337
1024	2500917	2560
4096	810165	3318
64K	52456	3437
1M	3287	3446

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	5768698	369.1
256	3845493	984.4
1024	1640179	1679
4096	498622	2042
64K	32269	2114
1M	2016	2114

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	4513154	288.8
256	2112310	540.7
1024	697899	714.6
4096	188329	771.3
64K	11924	781.5
1M	745.8	782.1

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 protected key)

Native: AES - 128 bit CBC Encipher (KMC-AES protected key)		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	5884378	376.6
256	4390316	1123
1024	2173599	2225
4096	718971	2944
64K	47321	3101
1M	2959	3103

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	5332784	341.2
256	3775899	966.6
1024	1740520	1782
4096	550699	2255
64K	36046	2362
1M	2254	2363

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	5961977	381.5
256	3921503	1003
1024	1652072	1691
4096	499471	2045
64K	32421	2124
1M	2030	2129

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 z13 Model N96, 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 2.7 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	835766	53.4
256	781000	199.9
1024	610713	625.3
4096	328712	1346
64K	31185	2043
1M	2017	2115

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	806921	51.6
256	669691	171.4
1024	406864	416.6
4096	157354	644.5
64K	11789	772.6
1M	746.5	782.8

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	743354	47.5
256	712301	182.3
1024	609677	624.3
4096	386995	1585
64K	44707	2929
1M	2946	3089

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	780501	49.9
256	732443	187.5
1024	597294	611.6
4096	341774	1399
64K	34623	2269
1M	2251	2360

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	534903	34.2
256	504958	129.2
1024	420701	430.7
4096	247437	1013
64K	26001	1704
1M	1698	1781

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	536382	34.3
256	518796	132.8
1024	466807	478.0
4096	335337	1373
64K	47940	3141
1M	3272	3431

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 CEX5S 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 CEX5S 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 CEX5S 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	333938	21.3
256	322853	82.6
1024	290058	297.0
4096	206545	846.0
64K	29347	1923
1M	2007	2104

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	335068	21.4
256	308288	78.9
1024	237767	243.4
4096	123539	506.0
64K	11522	755.1
1M	743.7	779.8

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 protected key)

ICSF API: AES-128 Encipher (KMC-AES protected key) 1 job		
Data Length (Bytes)	Operations/sec	x10**6 Bytes/sec
64	340312	21.7
256	331985	84.9
1024	309827	317.2
4096	239510	981.0
64K	41426	2714
1M	2927	3070

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	338923	21.6
256	329504	84.3
1024	299476	306.6
4096	217721	891.7
64K	32573	2134
1M	2241	2350

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

3.3 Crypto Express5S Performance

The Crypto Express5S feature is designed to satisfy high-end server security requirements. The Crypto Express5S 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 Express5S 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 z13 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 CEX5S feature via the PCIe bus to the z13 Central Processors (CPs) incurs latency and data transmission time. Because of this connection to the z13 CPs, the CEX5S operates asynchronously to the z13 CPs.

There can be a maximum of 16 CEX5S features in a z13, each CEX5S feature containing one PCIe adapter.

3.3.1 CEX5S CCA Coprocessor (CEX5C) Symmetric Key Performance - Encryption/Decryption and MAC Operations

This chapter deals with CEX5S 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 CEX5S 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 CEX5S 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 CEX5S 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 z13 Model 2964-N96 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. An example of the increase of measured throughput using 8 jobs is shown for the Single DES CBC Encipher operation.

The performance numbers are from measurements using z/OS V2.1 and ICSF FMID HCR77B0.

All CEX5S results presented in this paper are from measurements utilizing the Cryptographic AP-Queue I/O interrupt support.

CEX5S CCA Coprocessor DEA Cipher Block Chaining Encipher with Single Length Key (56 Bits)

CEX5C (one job): Single DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	8176	523.2
256	8154	2087
1024	8038	8231
4096	7493	30691
64K	920.2	60310
1M	61.06	64036

The above table provides measurement results for an environment where one job was continuously executing the cryptographic operation using one CEX5S CCA Coprocessor card. As mentioned, the execution of the cryptographic operation in the CEX5C card is asynchronous to the z13 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 CEX5C 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 CEX5C 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 CEX5C.

CEX5C (eight jobs): Single DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	11398	729.5
256	11374	2911
1024	11331	11603
4096	11202	45887
64K	1589	104193
1M	107.5	112783

The throughput with 4 CEX5C adapters with 32 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 4 times the throughput of one CEX5C adapter with eight jobs (as shown above).

CEX5S CCA Coprocessor DEA Cipher Block Chaining Encipher with Triple Length Key (168 Bits)

CEX5C (one job): Triple DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	7731	494.8
256	7685	1967
1024	7561	7742
4096	6842	28026
64K	791.8	51894
1M	52.17	54711

The throughput for eight jobs for CEX5C TDES is on the order of 1.4 times to 1.9 times higher than for one job.

CEX5S CCA Coprocessor AES 128-bit Cipher Block Chaining Encipher

CEX5C (one job): AES 128-bit CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	8050	515.2
256	7986	2044
1024	7889	8079
4096	7374	30207
64K	918.5	60197
1M	61.06	64031

The throughput for eight jobs for CEX5C AES 128-bit CBC encryption is on the order of 1.4 times to 1.7 times higher than for one job.

CEX5S CCA Coprocessor AES 256-bit Cipher Block Chaining Encipher

CEX5C (one job): AES 256-bit CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	8065	516.1
256	8020	2053
1024	7916	8106
4096	7125	29186
64K	876.1	57421
1M	58.26	61099

The throughput for eight jobs for CEX5C AES 256-bit CBC encryption is on the order of 1.4 to 1.8 times higher than for one job.

CEX5S CCA Coprocessor Message Authentication Code with DEA Single Length Key (56 Bits)

CEX5C (one job): MAC with single DES		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	8176	167.1
256	8085	664.4
1024	7991	2576
4096	7493	9269
64K	654.2	13008
1M	41.81	13337

The throughput for eight jobs for CEX5C MAC is on the order of 1.4 to 1.5 times higher than for one job.

3.3.2 CEX5S CCA Coprocessor VISA Format Preserving Encryption (FPE)

New with CEX5S CCA Coprocessor and ICSF FMID HCR77B0 is support for VISA Format Preserving Encryption encipher, decipher and translate services. 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 CEX5C.

CEX5C 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	2260	2403
VFPE Decipher	2260	2401
VFPE Translate	1909	2010
CBC Encipher	6209	8273
CBC Decipher	6726	9129
CBC Translate	5590	6899

3.3.3 CEX5S CCA Coprocessor Symmetric Key Performance - Diverse Operations

The following table gives the performance in maximum number of operations per second for one CEX5S CCA Coprocessor for some selected symmetric key operations.

CEX5C Symmetric Key Operations - Examples	Ops/s (1 job)	Ops/s (8 jobs)
Key Generate (operational DES KEY GENKY key)	4972	5823
Clear PIN Generate Alternate (DES OPINENC + DES PINGEN keys)	5663	7164
Clear PIN Generate (16 digits) (DES PINGEN key)	7515	10655
Encrypted PIN Translation (DES IPINENC key and DES OPINENC key)	6085	7504
Encrypted PIN Translation (2 UKPT enabled KEY GENKY keys)	2240	2316
Encrypted PIN Verification (UKPT enabled KEY GENKY + DES PINVER keys)	2973	3265

The throughput with 4 CEX5C adapters with 32 jobs repetitively requesting the same cryptographic operation for the examples in the table above is close to 4 times the throughput of one CEX5C adapter with 8 jobs.

3.3.4 CEX5S 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). ICSF FMID HCR77B0 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 when the ICSF address space is initialized. This support allows ICSF FMID HCR77B0 to satisfy random number requests from an internal private cache, eliminating the delay associated with sending the request to the CEX5 adapter. When the cache depletion threshold is reached, ICSF FMID HCR77B0 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 HCR77B0 and one CEX5S 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	650214
RNGL-1	642746
RNGL-64	645882
RNGL-1K	256231
RNGL-8K	32515

3.3.5 CEX5S CCA Coprocessor PKA Performance

The CEX5S 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 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 V2.1 and ICSF FMID HCR77B0 invoking the operation via the ICSF API according to the PKCS-1.2 Standard. Measurements were performed on a z13 Model N96.

CEX5S CCA Coprocessor PKA Performance

CEX5C on 2964-N96 with z/OS V2.1; ICSF FMID HCR77B0				
Public Key Decrypt (PKD), Public Key Encrypt (PKE)				
Digital Signature Generate (DSG), Digital Sign Verify (DSV)				
Symmetric Key Import (encrypted with RSA key) (SYI)				
CEX5C	1	1	2	4
Jobs	1	8	16	32
	Operations/sec	Operations/sec	Operations/sec	Operations/sec
PKD-CRT 1024 bit	2921	5293	10402	20822
PKD-CRT 2048 bit	1113	3733	7360	14761
PKD-CRT 4096 bit	208	690	1381	2761
PKD-ME 512 bit	3317	6431	12645	24257
PKD-ME 1024 bit	1246	5068	10134	20013
PKE 512 bit	5682	7209	13810	28153

PKE 1024 bit	4983	6148	11896	24173
PKE 2048 bit	3652	4813	9371	18972
PKE 4096 bit	2134	3368	6612	13107
DSG-CRT 1024 bit	2965	5361	10469	19161
DSG-CRT 2048 bit	1129	3807	7453	14895
DSG-CRT 4096 bit	208	690	1381	2761
DSG-EC BP-192 bit	2877	8253	16318	29246
DSG-EC BP-256 bit	2244	8116	16085	31727
DSG-EC BP-512 bit	818	2824	5648	11288
DSG-EC PC-192 bit	3294	8229	16111	31808
DSG-EC PC-256 bit	2514	8066	16079	31884
DSG-EC PC-521 bit	1089	3993	7983	15951
DSV-CRT 1024 bit	5116	6433	12550	25104
DSV-CRT 2048 bit	3732	4930	9656	19343
DSV-CRT 4096 bit	2149	3377	6603	13359
DSV-EC BP-192 bit	1729	6956	13904	27797
DSV-EC BP-256 bit	1284	4778	9553	19093
DSV-EC BP-512 bit	424	1370	2740	5478
DSV-EC PC-192 bit	2064	8517	17032	33958
DSV-EC PC-256 bit	1457	5592	11182	22301
DSV-EC PC-521 bit	590	1969	3937	7871
SYI-CRT 512 bit	3169	4471	8961	17663
SYI-CRT 1024 bit	2428	3864	7646	15298
SYI-CRT 4096 bit	207	696	1393	2785

The PKA cryptographic operation throughput with 4 CEX5C adapters with a sufficient number of jobs repetitively requesting the same cryptographic operation for the examples in the table above is close to 4 times the throughput of one CEX5C adapter with 8 jobs.

PKA Key Generation

The CEX5S 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 CEX5S CCA Coprocessor.

CEX5S CCA Coprocessor PKA Key Generation Performance

CEX5C PKA Key Generate	
	Operations/sec
RSA CRT 512 bit	42.9
RSA CRT 1024 bit	30.9
RSA CRT 2048 bit	7.59
RSA CRT 4096 bit	0.64
RSA ME 512 bit	42.7
RSA ME 1024 bit	30
EC BP-192 bit	777
EC BP-256 bit	591
EC BP-512	206
EC PC-192 bit	905
EC PC-256 bit	671
EC PC-512 bit	279

3.3.6 CEX5S Enterprise PKCS #11 Coprocessor (CEX5P) Symmetric Key Performance – Encryption / Decryption and HMAC operations

z13 with CEX5S cryptographic feature provides the ability to configure the CEX5S in Enterprise PKCS #11 (EP11) Coprocessor mode. ICSF FMID HCR77B0 supports the use of CEX5S 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 CEX5S EP11 Coprocessor.

CEX5S Enterprise PKCS #11 Coprocessor DEA Cipher Block Chaining Encipher with Triple Length Key (168 Bits)

CEX5P (one job): Triple DES CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	6658	426.1
256	6450	1651
1024	5692	5829
4096	3882	15902
64K	268.5	17596
1M	19.3	20251

CEX5S Enterprise PKCS #11 Coprocessor AES 128-bit Cipher Block Chaining Encipher

CEX5P (one job): AES 128-bit CBC Encipher		
Data Length (Bytes)	Operations/sec	x10**3 Bytes/sec
64	6643	425.1
256	6339	1623
1024	6021	6166
4096	3968	16253
64K	274.5	17995
1M	20.1	21084

CEX5S Enterprise PKCS #11 Coprocessor Secure Key HMAC Operations

CEX5P (one job): HMAC Generate Operations per Second			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	4676	4582	3426
256	4629	4421	3218
1024	3937	3775	3012
4096	3186	3109	2615
64K	313	310	302
1M	23.2	22.9	22.9

CEX5P (1 job): HMAC Verify Operations per Second			
Data Length (Bytes)	SHA-1	SHA-256	SHA-512
64	4832	4634	3588
256	4649	4444	3247
1024	3964	3840	3123
4096	3313	3182	2682
64K	307	308	304
1M	22.8	23.0	22.7

3.3.7 CEX5S Enterprise PKCS #11 Coprocessor (CEX5P) Secure Key PKA Operations

CEX5P Enterprise PKCS #11 Coprocessor Secure Key PKA Performance

CEX5P	1	1
Jobs	1	8
	Operations/sec	Operations/sec
Private Key Sign 1024 bit	2757	4941
Private Key Sign 2048 bit	1132	3808
Private Key Sign 4096 bit	213	764
Private Key Sign BrainPool 192 bit	2157	3408
Private Key Sign BrainPool 256 bit	1788	4939
Private Key Sign BrainPool 512 bit	737	3594
Private Key Sign Prime Curve 192 bit	2486	5170
Private Key Sign Prime Curve 256 bit	1997	5141
Private Key Sign Prime Curve 521 bit	979	4459
Private Key Verify 1024 bit	2804	3139
Private Key Verify 2048 bit	1810	1989
Private Key Verify 4096 bit	805	606
Private Key Verify BrainPool 192 bit	1373	3981

Private Key Verify BrainPool 256 bit	1073	3758
Private Key Verify BrainPool 512 bit	388	1808
Private Key Verify Prime Curve 192 bit	1677	4294
Private Key Verify Prime Curve 256 bit	1206	3933
Private Key Verify Prime Curve 521 bit	541	2614
Wrap Private Key 1024 bit	3124	3487
Wrap Private Key 2048 bit	2230	2517
Wrap Private Key 4096 bit	1139	1230
Unwrap Private Key 1024 bit	2075	2221
Unwrap Private Key 2048 bit	1006	2285
Unwrap Private Key 4096 bit	206	754

CEX5P IBM PKCS #11 Coprocessor PKA Key Generate Performance

CEX5P PKA Key Generate	
	Operations/sec
RSA CRT 1024 bit	7.751
RSA CRT 2048 bit	1.763
RSA CRT 4096 bit	0.24
EC Brainpool 192 bit	50.71
EC Brainpool 256 bit	41.65
EC Brainpool 512 bit	10.41

EC Prime Curve 192 bit	83.31
EC Prime Curve 256 bit	49.99
EC Prime Curve 521 bit	21.11

3.3.8 CEX5S Accelerator Performance

The CEX5S 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 V2.1 and ICSF FMID HCR77B0 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)

CEX5S Accelerator PKA Performance

CEX5A PKA Key Decrypt (PKD), Public Key Encrypt (PKE), and Digital Signature Verify (DSV)		
2964 CPs	4	4
CEX5A Adapters	1	1
Jobs	1	8
	Operations/sec	Operations/sec
PKD CRT 512 bit	10261	47232
PKD CRT 1024 bit	6589	25372
PKD CRT 2048 bit	1561	4951
PKD CRT 4096 bit	227	689
PKD ME 512 bit	6583	25248

PKD ME 1024 bit	1577	4994
PKE 512 bit	26641	87525
PKE 1024 bit	22871	106828
PKE 2048 bit	14007	75853
PKE 4096 bit	5692	21169
DSV CRT 512 bit	26704	107478
DSV CRT 1024 bit	22404	108202
DSV CRT 2048 bit	13916	76023
DSV CRT 4096 bit	5671	21168

The first result column of the above table is for measurements where one job was continuously executing the cryptographic operation using one CEX5S Accelerator card. As mentioned, the execution of the cryptographic operation in the CEX5S Accelerator is asynchronous to the z13 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 CEX5S Accelerator card. The increased throughput is due to the fact that tasks are always available for execution in the CEX5S Accelerator card due to the parallel threads that run in the z13 CPs. Thus the capability of the CEX5S Accelerator card for parallel execution of the cryptographic operation can be utilized.

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 / Internet 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 z13 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 z13 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 z13 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.

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 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 z13 the PKD operation will be routed for execution to the CEX5S CCA Coprocessor or CEX5S 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 z13 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 1024 bits or 2048 bits as noted in the table. The SSL data encryption was AES-128 bit and SHA-1 cipher except when stated otherwise. The symmetric key data encryption for AES-128 and SHA-1 was executed in CPACF hardware.
- One packet of 2048 Bytes was 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 CEX5S Accelerator or CEX5S 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 z13 Central Processors during the measurement interval.

Crypto Utilization %: Average utilization of the CEX5S Accelerator or CEX5S 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 results show the throughput of one CEX5S Accelerator adapter being on the order of 4 times the throughput of one CEX5S CCA Coprocessor adapter for the specific workload characteristics

and system environment used for these measurements. Thus for high SSL transaction rate environments, Accelerator is the preferred configuration mode for a CEX5S feature.

The resource consumption in system processing power for one SSL protocol handshake is on the order of 1/27,000 of the system (see table below) in the z/OS environment for a z13 Model N96 with 4 Central Processors and CEX5S features available (4 CEX5S Coprocessors or 1 CEX5S Accelerator).

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 V2.1 and Enhanced Cryptographic Support for z/OS V1.13-V2.1 (ICSF FMID HCR77B0)

z13 Model 2964-N96 (4 Central Processors)

Caching SID	RSA Key Length	Handshake	Client Auth.	ETR	CPU Util. %	Crypto Util. %
100%	1024	Avoided	no	28,766	62.35	NA
no	1024	Software	no	1,430	99.99	NA
no	1024	4 CEX5C	no	20,561	75.50	98.5
no	1024	1 CEX5A	no	21,275	78.85	94.5
no	1024	2 CEX5A	yes	8,232	42.94	62.8
no	2048	Software	no	243	100	NA
no	2048	1 CEX5C	no	3,693	14.32	99.8
no	2048	1 CEX5A	no	4,919	19.29	99.1

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 CEX5C 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 14.3 times and reduces the CP utilization by 25%. The 4 CEX5 Coprocessors were 98.5% utilized. Adding additional CEX5 Coprocessors to this environment would allow for a higher ETR.

The fourth row shows that a similar ETR can be achieved with just one CEX5S adapter configured in Accelerator mode. In this measurement the utilization of the CEX5 Accelerator was 94.5%. Adding additional CEX5 Accelerators to this environment would allow for a higher ETR.

If client authentication is required, the additional cryptographic operations necessary to authenticate the client reduce the throughput capacity of the server, as shown in row 5 of the table. A second CEX5 Accelerator was added to the system configuration for this measurement. The average utilization of the 2 Accelerators was 62.8%.

The final three rows demonstrate the effect that adding a CEX5S Coprocessor or Accelerator has on SSL handshake throughput when an RSA 2048 bit key is used for the Public Key operation required by the handshake. A system without CEX5 hardware supported 243 handshakes per second while using 100% of the available CPU. Adding just 1 CEX5S Coprocessor, the throughput increased to 3,693 handshakes per second and the CP Utilization decreased to only 14.32%. This demonstrates how off-loading the compute intensive processing associated with an SSL handshake increases system capacity and reduces CP Utilization. When the CEX5S was configured in Accelerator mode, the throughput increased to 4,919 handshakes per second with 19.29% CP Utilization. In both cases, the CEX5S was near full utilization. Adding CEX5S Coprocessors or Accelerators to the system would allow for a higher ETR as there is plenty of CPU available to handle additional workload.