System z Crypto in a Distributed World

Eysha S. Powers

IBM Corporation

Wednesday, August 12, 2015: 10:00am – 11:00am

Session Number 17622

#SHAREorg

Table of Contents

IBM has been providing Security & Encryption Solutions for over 30 years…

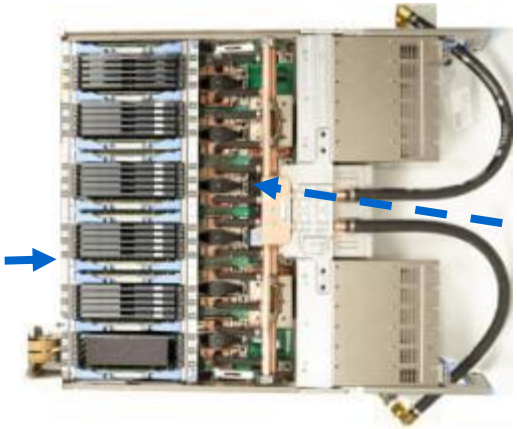
A History of Enterprise Security

- RACF: controls access to resources and applications: 1976
- Hardware Cryptography: 1977
- Key management built into operating system (ICSF): 1991
- Distributed Key Management System (DKMS) (1990's)
- Intrusion Detection Services (IDS): 2001
- z/OS PKI Services: create digital certificates & act as Certificate Authority (CA) – 2002
- Multilevel Security (MLS): 2004
- Encryption Facility for z/OS: 2005
- TS1120 Encrypting Tape Drive: 2006
- LTO4 Encrypting Tape Drive: 2007
- License ECC Technology from Certicom: 2008
- Tivoli Encryption Key Lifecycle Manager: 2009
- Self-Encrypting Disk Drives, DS8000: 2009
- System z10 CPACF Protected Key Support: 2009
- Crypto Express3 Crypto Coprocessor: 2009
- z Systems z196 with additional CPACF encryption modes: 2010
- z Systems zEC12 with Public Key Cryptography Standards – Enterprise PKCS#11

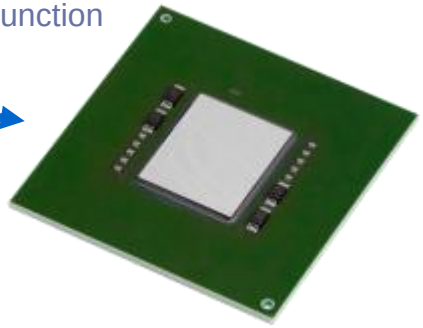
Hardware Crypto Support in z13

PCIe I/O drawers

CPC Drawer



PU SCM
Each PU is capable of having the CPACF function



Crypto Express5S

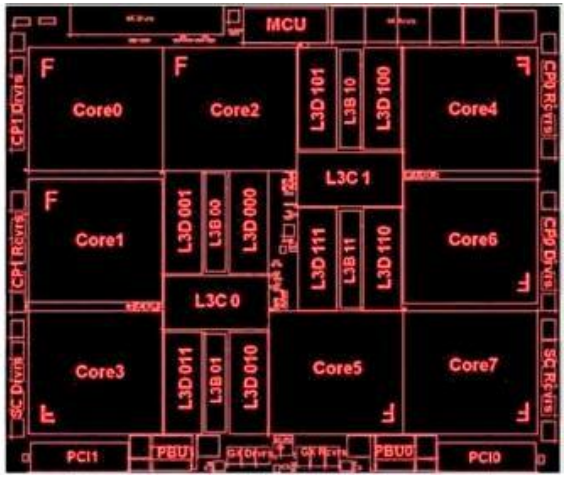
TKE required for management of Crypto Express5S and EP11

Trusted Key Entry (TKE)

Smart Cards

Smart Card Readers

CPACF - CP Assist For Cryptographic Functions



Supported Algorithms	Clear Key	Protect Key
DES, T-DES	Y	Y
AES128	Y	Y
AES192	Y	Y
AES256	Y	Y
SHA-1	Y	N/A
SHA-256	Y	N/A
SHA-384	Y	N/A
SHA-512	Y	N/A
PRNG	Y	N/A
DRNG	Y	N/A

Provides a set of symmetric cryptographic functions and hashing functions for:

- Data privacy and confidentiality
- Data integrity
- Random Number generation
- Message Authentication

Enhances the encryption/decryption performance of clear-key operations for

- SSL
- VPN
- Data storing applications

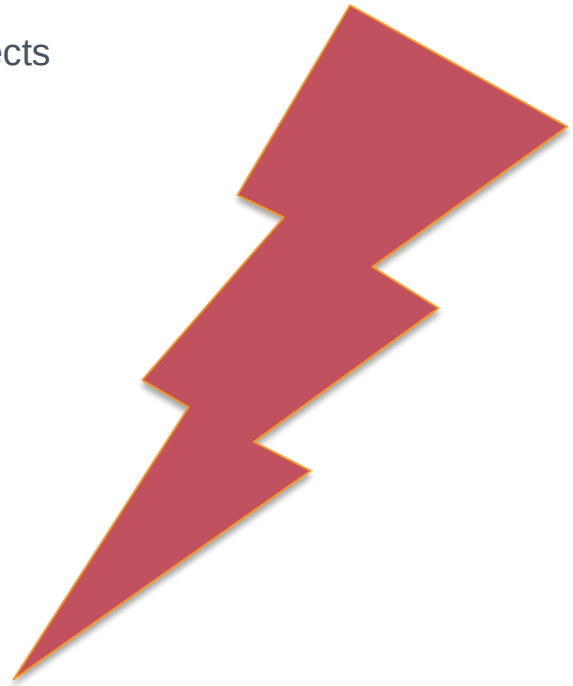
- Available on every Processor Unit defined as a CP, IFL, zAAP and zIIP
- Supported by z/OS, z/VM, z/VSE, z/TPF and Linux on z Systems
- Must be explicitly enabled, using a no-charge enablement feature (#3863),
  - SHA algorithms enabled with each server
- Protected key support for additional security of cryptographic keys
  - Crypto Express5S required in CCA mode

z13 CPACF Performance Enhancements

CP Assist for Cryptographic Function Co-processor redesigned from "ground up“

Enhanced performance over zEC12

- Does not include overhead for COP start/end and cache effects
- Enhanced performance for large blocks of data
  - AES: **2x throughput** vs. zEC12
  - TDES: **2x throughput** vs. zEC12
  - SHA: **3.5x throughput** vs. zEC12



Exploiters of the CPACF benefit from the throughput improvements of z13's CPACF such as:

- DB2/IMS encryption tool
- DB2® built in encryption
- z/OS Communication Server: IPsec/IKE/AT-TLS
- z/OS System SSL
- z/OS Network Authentication Service (Kerberos)
- DFDSS Volume encryption
- z/OS Java SDK
- z/OS Encryption Facility
- Linux on z Systems; kernel, openssl, openCryptoki, GSKIT

Crypto Express5S

- One PCIe adapter per feature
  - Initial order – two features
- Designed to be FIPS 140-2 Level 4
- Installed in the PCIe I/O drawer
- Up to 16 features per server
- Prerequisite: CPACF (#3863)
- Designed for 2X performance increase over Crypto Express4S

Three configuration options for the PCIe adapter

- Only one configuration option can be chosen at any given time
- All card secrets are erased when switching to or from EP11 Coprocessor mode

Accelerator	
TKE	N/A
CPACF	NO
UDX	N/A
CDU	YES(SEG3)

Clear Key RSA operations and SSL acceleration

CCA Coprocessor	
TKE	OPTIONAL
CPACF	REQUIRED
UDX	YES
CDU	YES(SEG3)

Secure Key crypto operations

EP11 Coprocessor	
TKE	REQUIRED
CPACF	REQUIRED
UDX	NO
CDU	NO

Secure Key crypto operations

Business Value

- High speed advanced cryptography; intelligent encryption of sensitive data that executes off processor saving costs
- PIN transactions, EMV transactions for integrated circuit based credit cards(chip and pin), and general-purpose cryptographic applications using symmetric key, hashing, and public key algorithms, VISA format preserving encryption(VFPE), and simplification of cryptographic key management.
- Designed to be FIPS 140-2 Level certification to meet regulations and compliance for PCI standards

CEX5S Cryptographic Units

- DES/TDES w DES/TDES MAC/CMAC
- AES, AESKW, AES GMAC, AES GCM, AES XTS mode, CMAC
- MD5, SHA-1, SHA-2 (224,256,384,512), HMAC
- VISA Format Preserving Encryption (VFPE)
- RSA (512, 1024, 2048, 4096) -> Performance improvement
- ECDSA (192, 224, 256, 384, 521 Prime/NIST)
- ECDSA (160, 192, 224, 256, 320, 384, 512 BrainPool)
- ECDH (192, 224, 256, 384, 521 Prime/NIST)
- ECDH (160, 192, 224, 256, 320, 384, 512 BrainPool)
- Montgomery Modular Math Engine
- RNG (Random Number Generator)
- PNG (Prime Number Generator) -> NEW
- Clear Key Fast Path (Symmetric and Asymmetric)

z/OS Integrated Cryptographic Services Facility (ICSF)

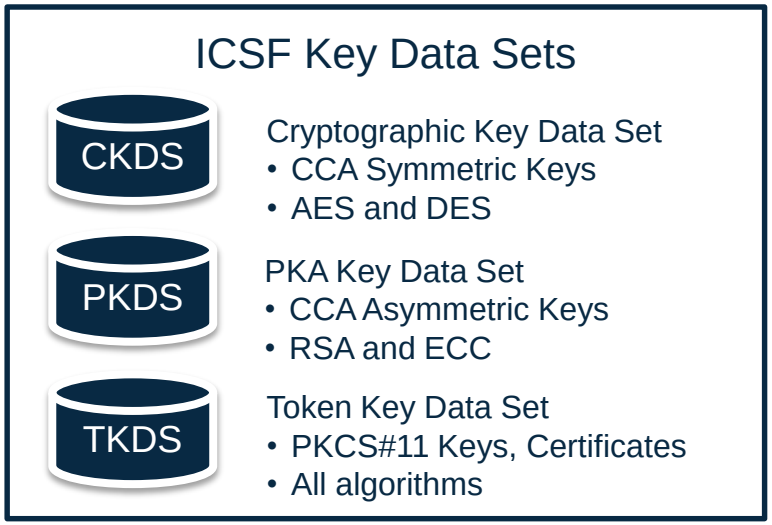
ICSF provides the application programming interfaces by which applications request cryptographic services.

ICSF is the default means to load master key values onto secure cryptographic features, allowing the hardware features to be used by applications.

ICSF callable services and programs can be used to generate, maintain, and manage keys that are used in the cryptographic functions.

ICSF uses cryptographic keys to:

- Protect data
- Protect other keys
- Verify that messages were not altered
- Generate, protect and verify PINs
- Distribute keys
- Generate and verify signatures

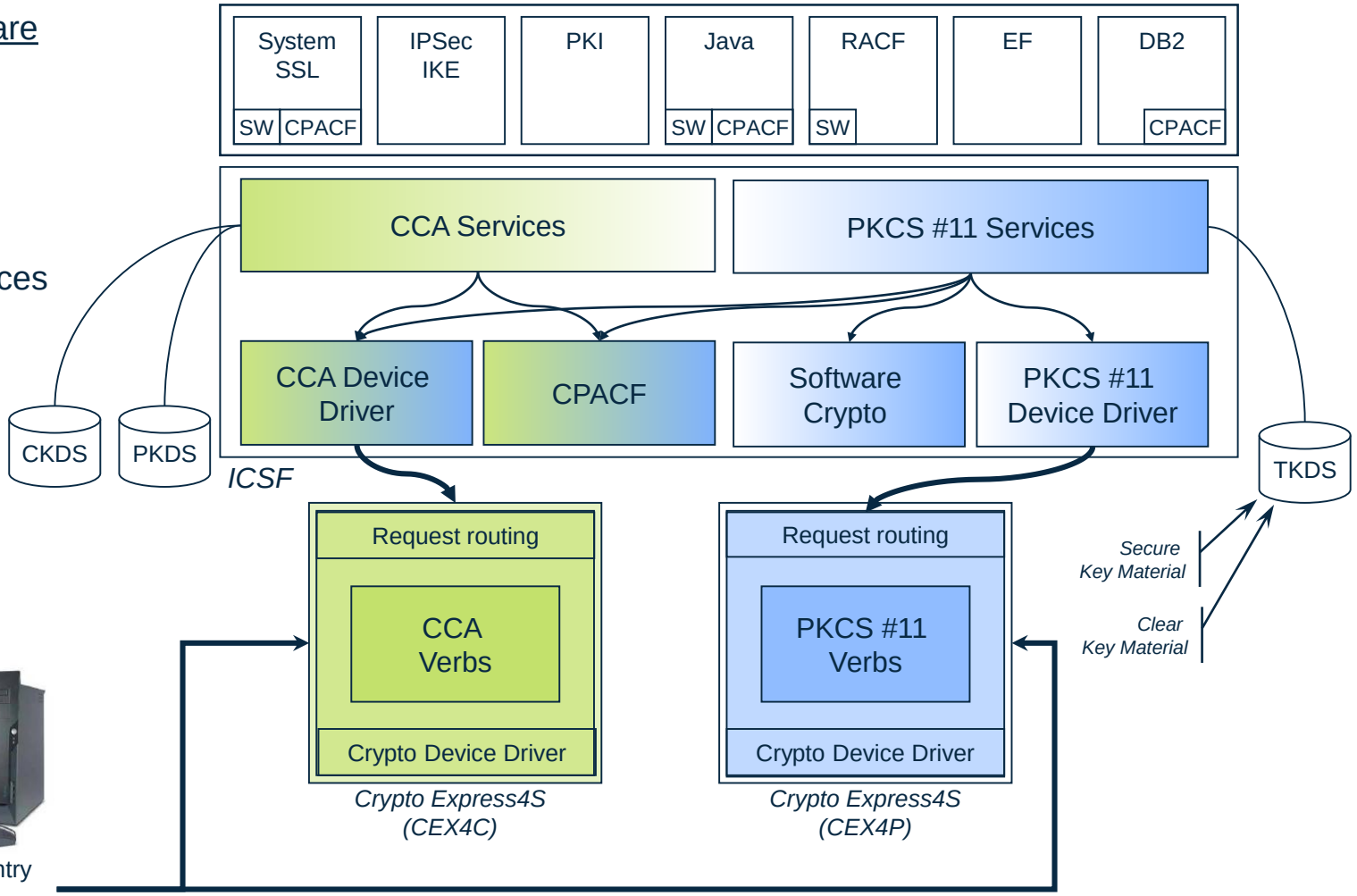


z/OS Crypto Stack

<u>z Systems Software</u>

z/OS

- ICSF
- RACF
- RMF
- z/OS PKI Services
- System SSL
- Java PKCS#11 Provider



<u>TKE</u>

Trusted Key Entry

CCA | PKCS11

Cryptography is used in a variety of places…

- **Data in flight**
  - Virtual Private Networks (VPNs)
  - SSL/TLS connections (using public/private keys and certificates and symmetric encryption)
  - Messaging infrastructures (using SSL/TLS or shared secrets)
  - WS-Security and SOA
- **Data at rest**
  - File and folder encryption – including the use of intermediate devices
  - Removable media (tape) encryption
- **Transactional environments**
  - Industry specific – finance
  - Mandates highly trust-worthy cryptography
  - Smart ID cards, ePassports…
- **For sharing user credentials between organizations – the establishment of trust**
  - Via certificate exchanges
  - Federated Identity Management
  - Credential formats such as SAML, OpenID Connect…

Encryption Protects IT Assets

The need for encryption is dictated by the need to protect IT assets such as sensitive data or to secure transactions. For certain applications such as payments and health care, the requirements stem directly from the standard bodies and the legislation.

Encryption is Mandated by Industry Standards

The security of sensitive data, such as personal information or payments data, is mandated

- by the payment cards industry via the PCI-DSS and PCI-PIN standards,
- via the European Union in the Data Protection Directive 95/46/EC,
- in the HiPAA, and in local legislation.

Encryption is a necessary mechanism to help you stay compliant with these regulations.

Business Problem

- ✓ The increasing need for secure encryption techniques is a must
- ✓ This need is driven across your enterprise
- ✓ This means that every mid-level server would need to have crypto hardware of some type installed...

…Or does it?

Consider…

Your z/OS LPAR has the crypto hardware and software to do:

- Encryption and decryption of data
- Key generation and distribution
- Personal Identification Numbers (PINs)
- Message Authentication Codes (MACs)
- Hashing of data
- Digital signatures
- EMV integrated circuit card specifications
- Card-verification values
- Translation of data and PINs in networks
- Secure Electronic Transaction
- And more…

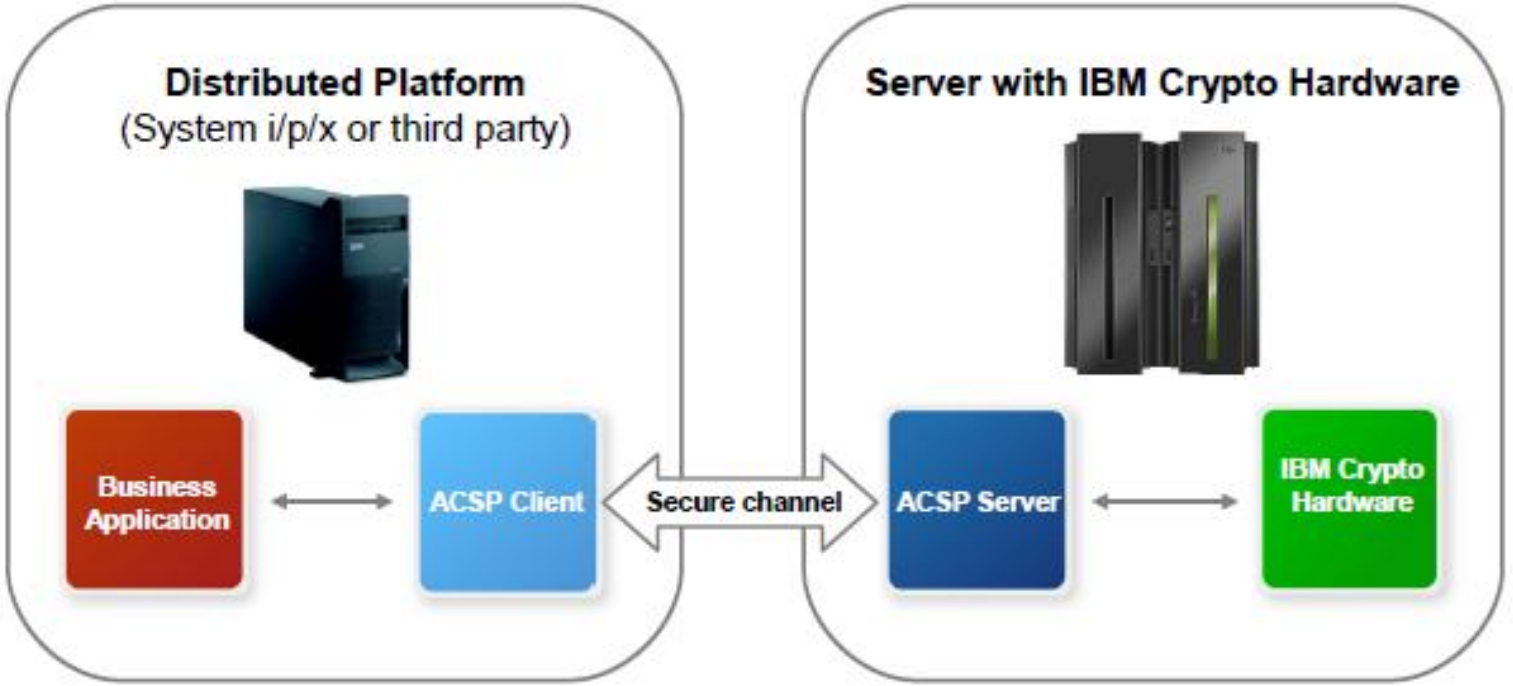
What if APIs were available to invoke z/OS crypto functions from remote applications and platforms.

IBM Advanced Crypto Services Provider

The Advanced Crypto Service Provider is a remote crypto services solution that enables distributed clients to access cryptographic hardware on IBM Systems over a network.

- Utilizes existing z Systems infrastructure (cost efficient)
- Uses multiple crypto processors with ICSF doing the load balancing (scalable)
- Provides efficient ACSP implementation (high performance)
- Enables centralized management (efficient operation, policy compliance)

IBM z Systems Crypto in a Distributed World



- ACSP client platforms
  - AIX, IBM i, Linux, Windows
  - PureSystems
  - In reality, any Java platform
- ACSP client APIs
  - CCA in Java and C
  - PKCS#11

- Transport network
  - IP
  - SSL/TLS protected (client/server auth)

- ACSP server platform
  - **System z: z/OS (CEX3/4/5)**
  - System p: AIX (4765)
  - System x: SLES, RHEL (4765)
  - IBM PureSystems

The ACSP Concept

- Utilize z Systems crypto capacity as the Net HSM
  - AIX and Linux servers supported as well
- Expose crypto functions to client applications across heterogeneous systems and environments
  - zEnterprise and zBX Application Serving Blades (ASB)
  - IBM PureSystems
  - Distributed platforms
- Benefits
  - Virtualization and cost effective use of available crypto capacity
  - Reduced administration and simpler key management
  - Crypto support for platforms with no native IBM crypto HW support
  - Easier to develop/deploy applications using crypto
  - High scalability, reliability, and availability
  - Leverages existing business continuity plans and procedures

The ACSP Client

Java Client (CCA)

- jCCA Java Framework
- **CCA** superset covering all IBM 4765 verbs and z/OS ICSF Services including **CPACF** and **EP11** services
- Supports User Defined eXtension (UDX) calls from Client
  - The IBM 4765 CC offers custom programming
- Supports User Defined Functions (UDF) on Server
  - Cut down on the number of functions needed to complete a transaction to cut down on overhead

C Client (CCA)

- Simulating CCA C interface environments for current CCA implementations to make existing applications code compatible.
  - Integrates with other applications C++, .Net, C#
- AIX: CSUFCCA, Linux: CSULCCA, Windows: CSUNCCA, IBM i: CSUCCCA

The ACSP Server

- Load Balancing
  - ACSP server on system z can load balance using the ICSF SW
- Scalability
  - Can use Sysplex Distributer to spread crypto across LPARs
  - CEX3/4/5 can be turned up if necessary
- SAF protection
  - Leverage over 50 years of security
  - Uses Client identity for fine grain access control
- Centralized Key Management
  - Can integrate DKMS (EKMF) to manage keys across a number of z/OS systems

Let's take a closer look…

The ACSP Java Client

- Uses jCCA framework
- Simple configuration of jCCA/ACSP using properties files
- Integrates well with the Eclipse Java Development environment.
- If application is targeted z/OS ICSF environment:
  - Application can be developed using ACSP Server in z/OS
  - Can be deployed using the native jCCA without changes
  - Both 31 or 64 bit z/OS ICSF CCA services can be used – jCCA/ACSP maps

Example: One Way Hash

Translate clear text to a fixed length hash value

Parameters:

- Clear text (full or partial text)
- Hashing Algorithm (e.g. MD-5, SHA-1, SHA-2)
- Chaining Information

Output: Hash

Sample output (20-byte hash):

DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17

Example: One Way Hash (cont'd)

```
CSNBOWH bow = new CSNBOWH();
  try {
    bow.addRule(CSNBOWH.SHA256);
    bow.addRule(CSNBOWH.ONLY);
    bow.setText(CCAUtil.S2B("SHARE Orlando 2015"));
    bow.execute();
    System.out.println("hash: " +
          CCAUtil.bytes2HexString(bow.getHash()));
  }
  catch (CCAException ce) {
    System.out.println("failure: " + ce);
  }
```

Demonstration:
Remote Crypto Application using ACSP

Additional Resources

IBM Crypto Education Community

https://www-304.ibm.com/connections/communities/community/crypto

IBM System z Development Blog

https://ibm.biz/zsystems_development

Thank You!

Feel free to connect on…

- Twitter: http://www.twitter.com/EyshaShirrine
- LinkedIn: http://www.linkedin.com/in/eysha