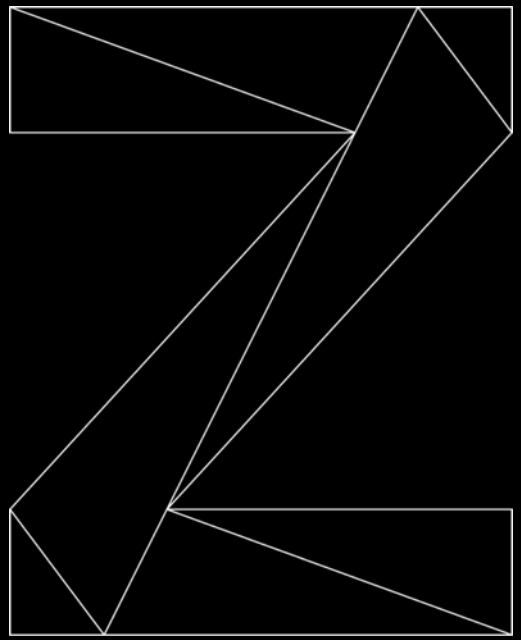
Introducing IBM z15 Cryptography

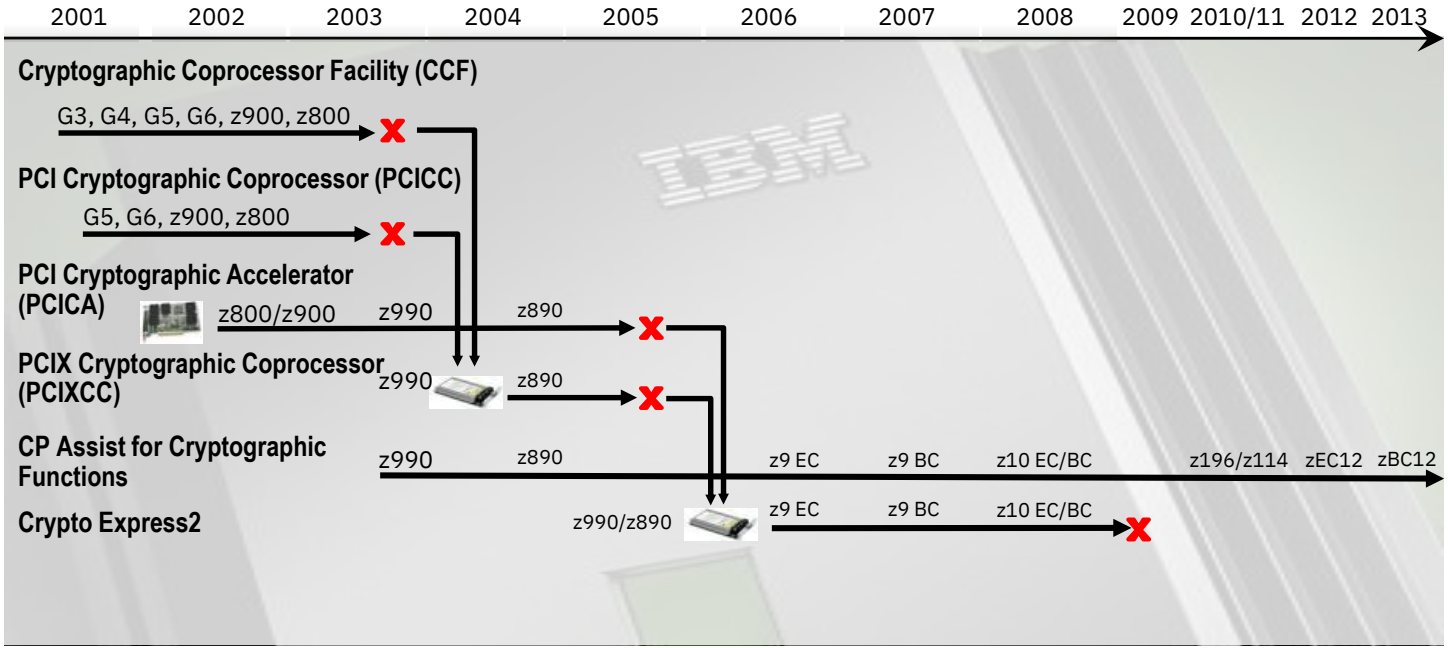
And the Ecosystem around it

Table of Contents

IBM Z Crypto History



IBM Z Crypto History

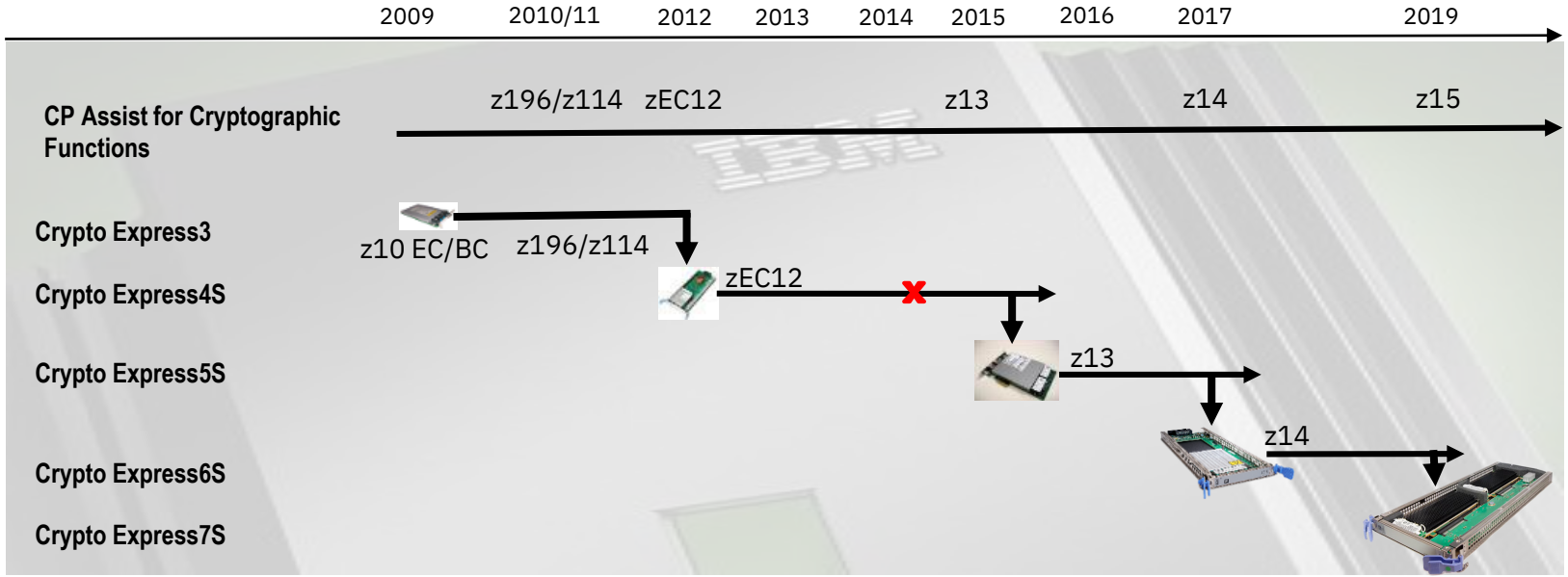


- Cryptographic Coprocessor Facility – Supports “Secure key” cryptographic processing
- PCICC Feature – Supports “Secure key” cryptographic processing
- PCICA Feature – Supports “Clear key” SSL/TLS acceleration
- PCIXCC Feature – Supports “Secure key” cryptographic processing
- CP Assist for Cryptographic Function allows limited “Clear key” crypto functions from any CP/IFL
  - NOT equivalent to CCF on older machines in function or Crypto Express2 capability
- Crypto Express2 – Combines function and performance of PCICA and PCICC

Hardware preceding CCF includes:

- IBM 3845 Channel Attached DES (1977)
- IBM 3848 Channel-Attached TDES (1979)
- IBM 4753 Channel-Attached CCA processor (1989)

IBM Z Crypto History



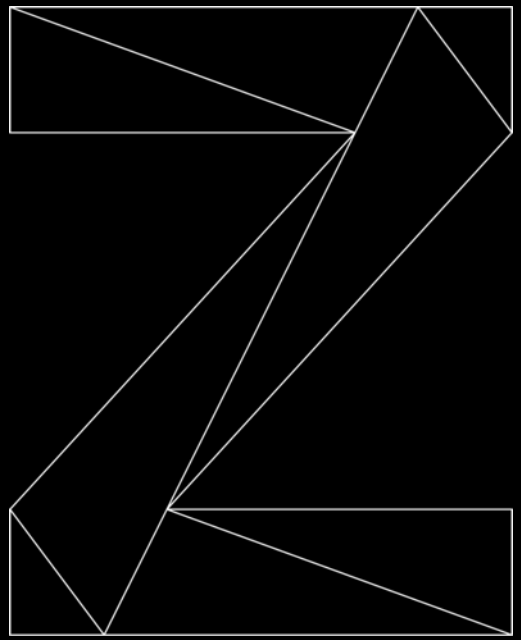
- CP Assist for Cryptographic Function allows limited “Clear key” crypto functions from any CP/IFL
  - NOT equivalent to CCF on older machines in function or Crypto Express2 capability
- Crypto Express3 – PCIe Interface, additional processing capacity with improved RAS
- Crypto Express4S - Enterprise PKCS #11
- Crypto Express5S - ECC HW acceleration and more RSA Engines
- Crypto Express6S – Enhanced Miniboot secure boot
- Crypto Express7S – Half Height Form Factor with enhanced Crypto performance

Over 40 Years of IBM Z Security & Encryption Solutions…

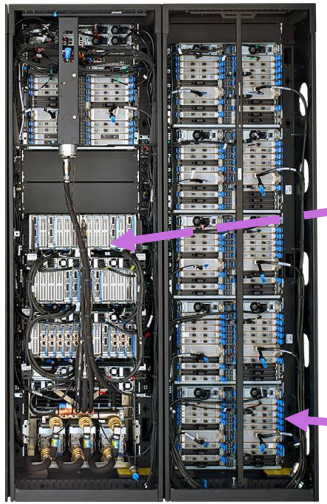
A History of Enterprise Security

- **IBM submits the Lucifer cipher to become the Data Encryption Standard (DES): 1974 - 1976**
- RACF: controls access to resources and applications: 1976
- Hardware Cryptography using IBM 3845 Channel Attached DES/TDES: 1977 - 1979
- IBM 4753 Channel Attached CCA Unit with smart cards and signature dynamics pen: 1989
- **Key management built into operating system (ICSF): 1991**
- **Distributed Key Management System (DKMS) (1990's)**
- **Trusted Key Entry (TKE) Workstation: ~1997**
- 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
- 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
- Crypto Express4S Crypto Coprocessor: 2012
- z Systems zEC12 with Enterprise PKCS#11: 2012
- Crypto Express5S Crypto Coprocessor: 2015
- z Systems z13 with Visa Format Preserving Encryption: 2015
- Multi-Factor Authentication for z/OS: 2016
- **Secure Service Container: 2016**
- **IBM z14 with Pervasive Encryption: 2017**
- **Crypto Express6S Hardware Security Module: 2017**
- **Crypto Express7S Hardware Security Module: 2019**
- **IBM Secure Execution for Linux: 2020**

IBM Z Crypto Hardware

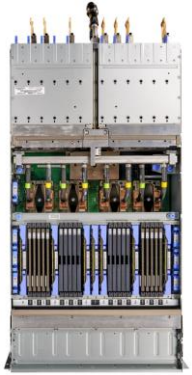


How is crypto supported in IBM Z hardware?

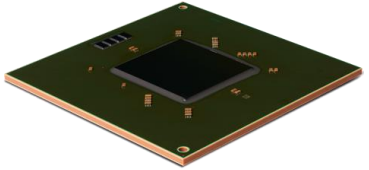


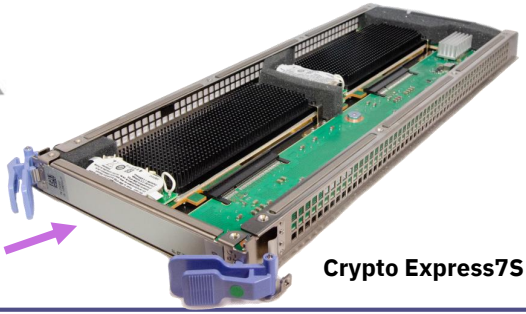
PCIe I/O Drawers

CPC Drawer



PU SCM
Each PU is capable of CPACF function





Crypto Express7S

With 64 bytes of input using 256-bit AES-CBC encryption, CPACF was measured to perform 6,140,117 operations per second.

Higher throughput may be obtained by enabling Integrated Accelerator for zEnterprise Data Compression (zEDC).

With 64 bytes of input using 256-bit AES-CBC encryption, a Crypto Express adapter was measured to perform 10,130 operations per second.

TKE is required for management of Crypto Express7S in EP11 or PCI-HSM mode (Tower or 1U)

Trusted Key Entry (TKE)

Smart Cards

Smart Card Readers

What is CPACF?

CP Assist for Cryptographic Function (CPACF)

IBM z hardware cryptographic function is 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

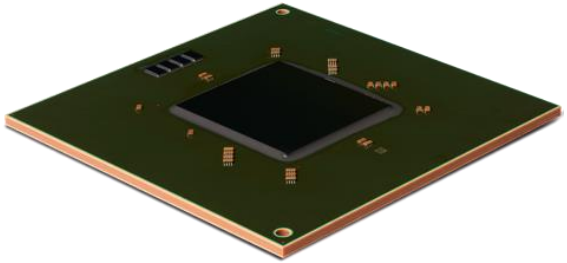
Provides a set of **symmetric cryptographic functions** and **hashing functions** for:

- Data privacy and confidentiality (DES, TDES, AES)
- Data integrity
  - SHA-1
  - SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)
  - SHA-3 (SHA3-224, SHA3-256, SHA3-384, SHA3-512)
  - SHAKE (SHAKE-128, SHAKE-256)
- Random Number generation
  - PRNG
  - TRNG

Message Authentication (DES, TDES, AES)

New **asymmetric** cryptographic functions with z15

- Elliptic Curve Digital Signatures
  - ECDH – P256, P384, P521
  - EdDSA – Ed25519, Ed448
- Elliptic Curve Key Exchange
  - ECDH - P256, P384, X25519, X449



Enhances the encryption/decryption performance of clear and protected key operations such as:

- SSL/TLS
- VPN
- Db2
- IMS
- Data sets and files
- Coupling Facility structures

IBM z15 CPACF Enhancements

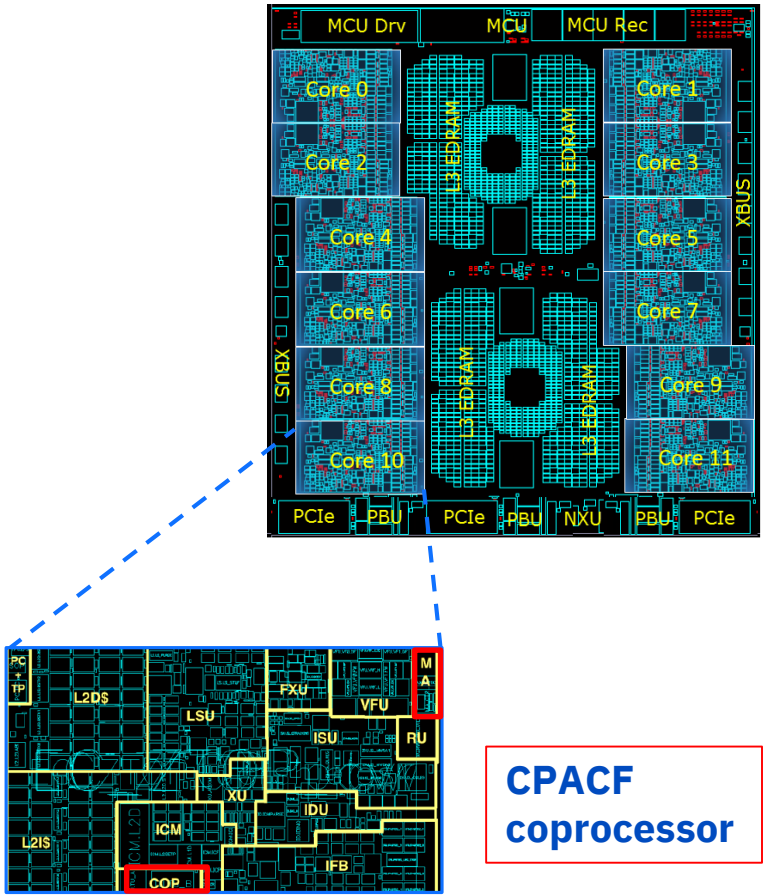
Elliptic Curve Operations

- **Elliptic Curve Digital Signatures**
  - ECDSA
    - Support for curves P256, P384, P521
  - EdDSA
    - Support for Ed25519 and Ed448
  - Protected key private key supported

Elliptic Curve Diffie Hellman Key Exchange
Support for curves P256, P384, P521, X25519, and X448

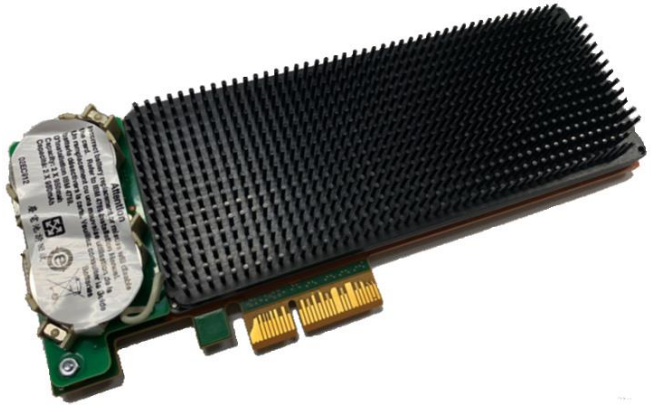
The performance is 10-300x faster than CEX7S

Higher throughput may be obtained by combining encryption with the Integrated Accelerator for zEnterprise Data Compression (zEDC)
https://www.ibm.com/downloads/cas/AM1PYZBB



CryptoExpress 7S based on the IBM 4769 Hardware Security Module

- Enhanced Cryptographic performance in half the size
- Designed to meet the following Security Standards:



HW - FIPS 140-2
NIST
Security standard for cryptographic modules
Assurance Level: Lvl 4

EP11- Common Criteria
International Certification Standard
Assurance Level: EAL4+

EP11- eIDAS
EU regulation: Electronic Identification, Authentication and Trust Services

CCA – GBIC
German Banking Industry Committee

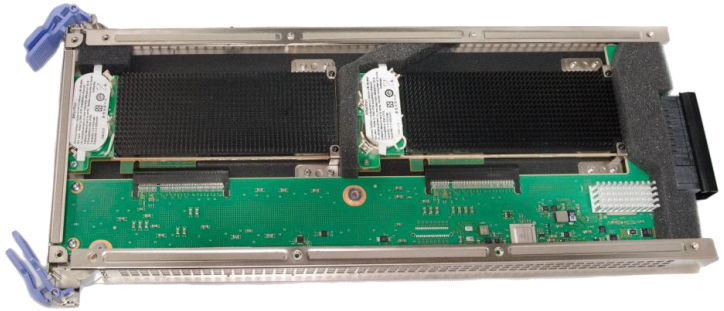
CCA - PCI HSM
Payment Card Industry Hardware Security Module Standard

CCA – MEPS
Carte Bancaire Methode d'Evaluation des Produits Securitaire

CCA – AusPayNet
Australian Payments Network

What is the benefit of Crypto Express adapters?

- Provides state-of–the art **tamper sensing and responding**, programmable hardware to **protect cryptographic keys**, sensitive cryptographic processing and sensitive custom applications
  - Unauthorized removal and/or tampering of the adapter zero-izes its content
- Suited to applications requiring high-speed **security-sensitive cryptographic operations** for data encryption and digital signing, and secure management and use of cryptographic keys
  - Functions targeted to Banking/Finance and Public sector
- Supports multiple **logically-separate cryptographic domains** for use by different LPARS.
- Provides both symmetric and asymmetric cryptographic functions.
- Supported by z/OS, z/VM, z/VSE, z/TPF and Linux on z Systems. **Note:** Crypto function exploitation may vary.



Crypto Express7S

DES/TDES w DES/TDES MAC/CMAC, AES, AESKW, AES GMAC, AES GCM, CMAC, MD5, SHA-1, SHA-2 (224,256,384,512), SHA3, SHA3 XOF, HMAC, VISA Format Preserving Encryption (VFPE), RSA (512, 1024, 2048, 4096), 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, Deterministic Random Number Generator (RNG), Prime Number Generator (PNG), Clear Key Fast Path (Symmetric and Asymmetric), **CCA only**: ISO-4 PIN Blocks, TR-34

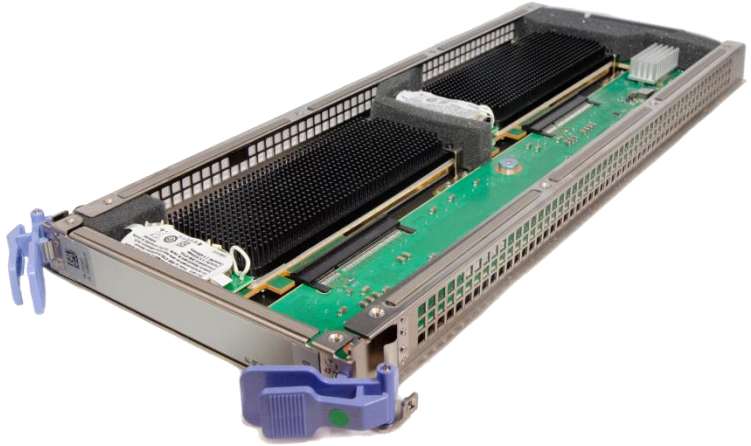
What are the different modes of operation for Crypto Express adapters?

Crypto Express adapters can be configured in three different modes

Accelerator Mode:
- Request is processed fully in hardware (versus PowerPC)
- Supports clear key RSA operations (e.g. SSL/TLS Acceleration)

CCA Coprocessor Mode:
- Supports the IBM Common Cryptographic Architecture (CCA) for financial services standards.
- Supports domain-segregated PCI-HSM Compliant mode. **Note:** Requires a TKE Workstation.
- Request is sent first to the internal IBM PowerPC for processing (default mode)
- Supports secure key crypto operations (i.e. keys encrypted by Master Keys)

EP11 Coprocessor Mode:
- Supports the PKCS #11 programming interface for public sector requirements. Designed for extended evaluations (FIPS and Common Criteria certifications)
- Request is sent first to the internal IBM PowerPC for processing (default mode)
- Requires the use of the TKE Workstation
- Supports secure key crypto operations (i.e. keys encrypted by Master Keys)



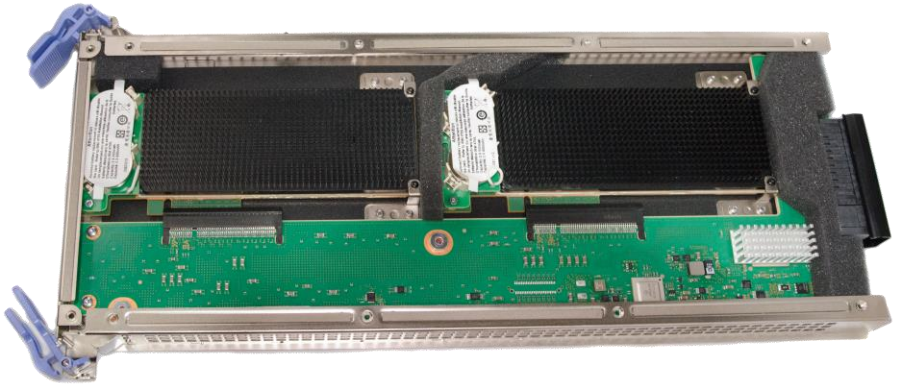
Designed to Meet Physical Security Standards
- FIPS 140-2 level 4
- ANSI X9.97
- Payment Card Industry (PCI) HSM
- Deutsche Kreditwirtschaft (DK)

Native PCIe card (FC 0890)
- Resides in the PCIe I/O drawer
- Requires CPACF Enablement (FC 3863)

CryptoExpress7S supporting 1 or 2 HSMs

Supporting Higher Cryptographic density with support of up to 60 HSMs
- IBM z15 and LinuxONE III water-cooled multi-frame (Models T01 and LT1) support up to 60 HSMs
- IBM z15 and LinuxONE III air-cooled single-frame (Models T02 and LT2) support up to 40 HSMs

CEX7S can be ordered in a 1 or 2 port configuration.



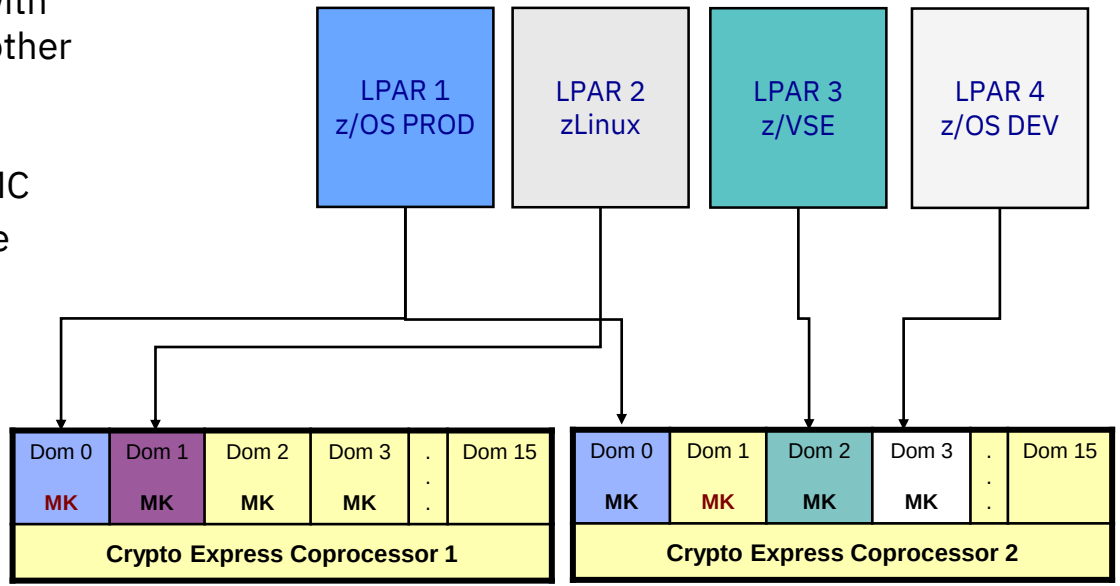
FC 0898 (2-port)

FC 0899 (1-port)

What are cryptographic domains?

IBM Z uses the concept of a cryptographic domains to virtualize the physical coprocessor and enable sharing across multiple System z logical partitions (LPARs)

- A coprocessor can be shared by multiple LPARs and different types of operating systems
- Secure Keys generated and wrapped with MK of one domain are not usable by another domain using different MK
- Cryptographic domain assignment is defined in LPAR image profile via SE/HMC
- IBM Z firmware enforces domain usage
- The Crypto Express coprocessor manages assignment of master keys to cryptographic domains.



What are Master Keys?

Master Keys are used to *protect sensitive cryptographic keys* that are active on your system.

Master Keys are *stored in secure hardware* on the crypto express card.

Master Keys are used only to *encipher and decipher keys*.

Master Keys should be *changed periodically*.

Master Key	Key Size	Protects
DES-MK	16-byte or 24-byte	DES keys
AES-MK	32-byte	AES and HMAC keys
RSA-MK	24-byte	RSA private keys
ECC-MK	32-byte	ECC, RSA-AESM, RSA-AESC keys
P11-MK	32-byte	PKCS #11 keys

How do you generate, maintain and manage Master Keys?

Recommended

Trusted Key Entry (TKE) Workstation

- **Most secure**; Dual controls; Separation of duties; Key material is not displayed
- Applicable for initialization of ICSF Key Data Sets (i.e. key stores) and Crypto Express adapters
- Applicable for master key rotation
- Required for EP11 Master Key management & PCI-HSM Master Key management
- Load and administer master keys across multiple IBM Z systems and geographies; Load master keys for inactive LPARs
- Separate, priced product

Trusted Key Entry (TKE) Workstation (Tower or 1U)

Smart Cards

Smart Card Readers

Default

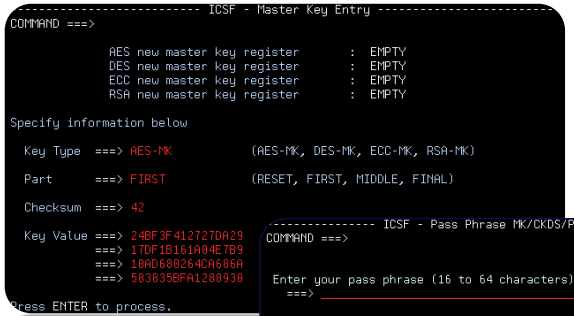
z/OS ICSF Master Key Entry Panels

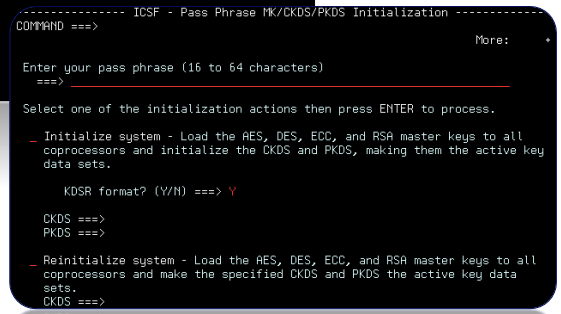
- **Less secure than TKE**; Separation of duties; Key material is displayed on panel
- Applicable for initialization of ICSF Key Data Sets (i.e. key stores) and Crypto Express adapters
- Applicable for master key rotation
- Included with z/OS and ICSF

Discouraged

z/OS ICSF Pass Phrase Initialization

- **Least secure**; No separation of duties
- Applicable for initialization of ICSF Key Data Sets (i.e. key stores) and Crypto Express adapters
- **NOT** applicable for master key rotation
- Included with z/OS and ICSF





Special Considerations for Master Keys

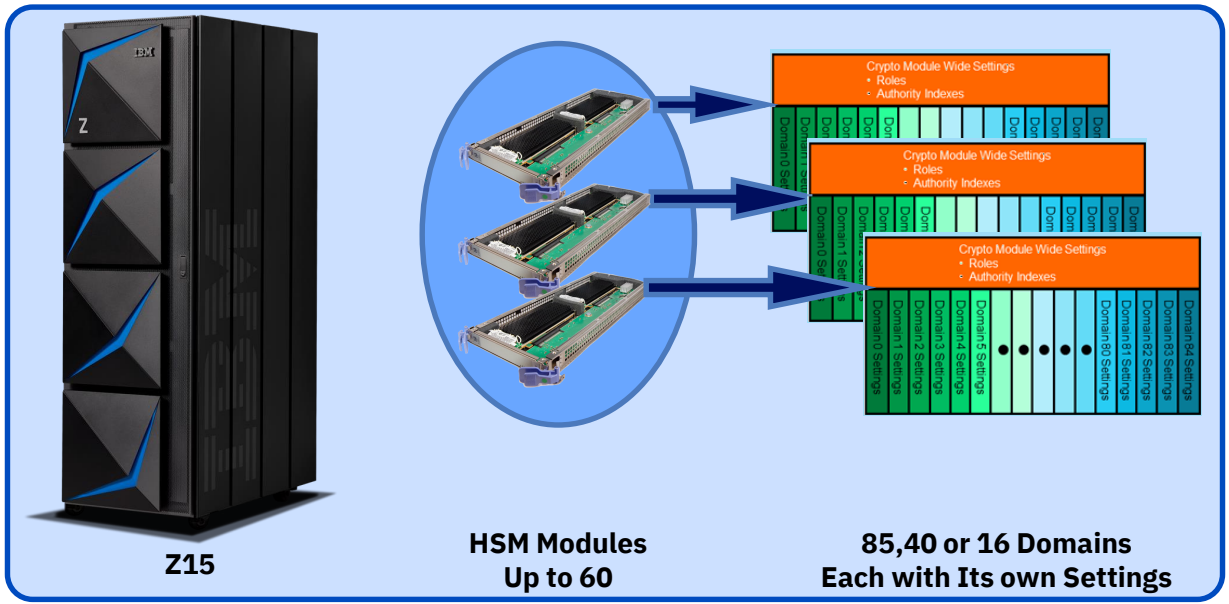
Master Keys are high value keys that must be protected.
– Loading Master Keys on a panel means that the key is viewable to passersby!
– The most secure way to load a Master Key is to use the TKE Workstation with smart cards.
  • The P11 Master Key may ONLY be loaded using a TKE Workstation.

If you plan to use the PPINIT or the Master Key Entry panels to manage Master Keys, consider how you would save the key material for future re-entry (e.g. new Crypto Express adapter, disaster recovery).
For disaster recovery, the same Master Keys must be loaded onto the backup system.

Option	Details	Pros	Cons
Print Screen	Use a Print Screen key or tool to capture the screen	Sensitive material can be immediately printed and stored in envelopes in a locked safe. No need to save on a local machine or USB stick.	Cannot use copy / paste to re-enter key material
Removable Storage Media	Copy and paste key material to a text file that is saved on a secure storage device (e.g. USB stick).	Easy to copy / paste the key material to the panels for re-entry.	The key material is only as secure as the storage media.

Trusted Key Entry (TKE) Workstation

TKE *is an appliance that simplifies the management of IBM Z Host Cryptographic Modules running in Common Cryptographic Architecture (CCA) or IBM Enterprise PKCS#11 (EP11) mode, using compliant level management techniques.*



- Features for Managing Module Scoped and Domain Scoped Administrative settings on Host Cryptographic Modules
- Secure, hardware-based Master Key and Operational key management
- Highly secure and efficient movement of administrative settings from one Host Cryptographic Module to another

TKE Workstation Features

Features for Managing Module-wide and Domain-specific Administrative settings on Host Cryptographic Modules

Featuring: Secure, simplified administrative management of multiple domain host cryptographic modules in complex configurations

Secure, hardware-based Master Key and Operational key management

Featuring: Compliant level hardware-based key management with proper encryption strengths, dual controls, and security relevant auditing

Highly secure and efficient movement of administrative settings from one Host Cryptographic Module to another

Providing: Secure, fast, and accurate deployment of new crypto modules on production, test, or disaster recovery systems

Popular Features

- Domain Grouping to broadcast a command to a set of domains
- Secure Loading of CCA Master Keys (MKs)
- Manage domains higher than 16
- Host Module Migration Wizards
- Enable/disable Access Control Points (ACPs)
- TKE Workstation Setup wizard
- Loading MKs for inactive LPARs
- Loading PIN decimalization tables
- Loading EP11 Master Key

TKE Workstation 9.2 – Minimum level to manage Crypto Express 7S

- **Base Hardware**
  - TKE 9.2 Workstation with a 4768 Cryptographic Adapter
    - TKE 9.2 Tower Workstation (FC 0086)
    - TKE 9.2 Rack-Mounted Workstation (FC 0085)
- **Additional Hardware:** Smart card readers and smart cards
  - Smart cards and readers are required for some TKE functions
    - Host module migration wizard
    - Management of EP11
    - NEW: Required for managing of PCI-HSM compliance mode
  - IBM Highly recommends using smart cards to hold key material
- **Migration Considerations**
  - The TKE Workstation feature must be assigned to a z14 server or later before you can place an order for TKE 9.1 or TKE 9.2 Licensed Internal Code (LIC).
  - TKE workstation features 0847 and 0849 are Lenovo based servers. They can't be upgraded past TKE 9.1 LIC.
  - You MUST install TKE 9.1 will all MCLs before you install TKE 9.2. Secure boot MCLs must be installed before the 9.2 upgrade.

- **Required for Managing Crypto Express 7**
- **Can configure TLS connection between the TKE and the Port for the ICSF Host Transaction Program.**
- **Access control tracking reports contain collection interval information:**
- **You have 3 choices for the type of hash pattern you want to see for you master keys**
- **With the latest EP11 host module code and Blue smart cards with the newest applet, you can set your EP11 Transport Wrapping Key policy to use a true 256-bit AES key**

TKE Application Updates

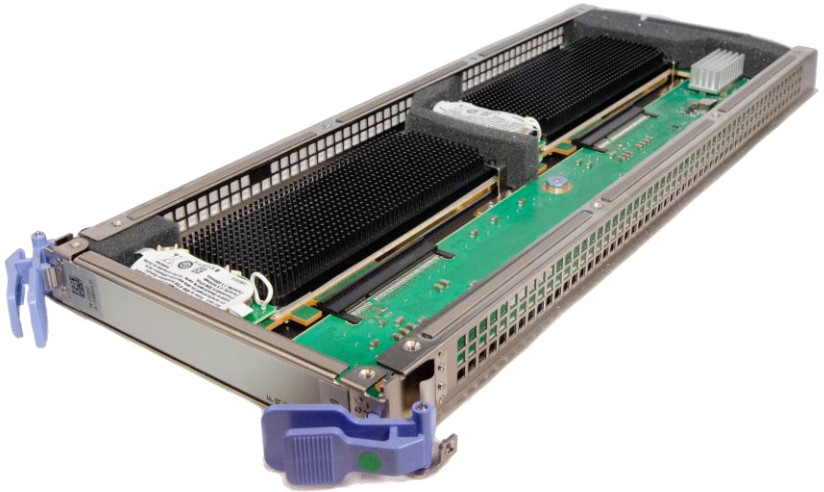
- Host definition can be configured to auto accept modules at open time
- The copy key part from binary file to smart card utility will allow you to select multiple files at one time
- Delete Host Crypto Module Role or Authority command issued from a domain group will be attempted on every module in the group
- Support for LINUX long passwords when the LINUX that support
- The TKE workstation profile wizard has a step to remove excess authority from the DEFAULT role

User-Defined Extensions (UDX)

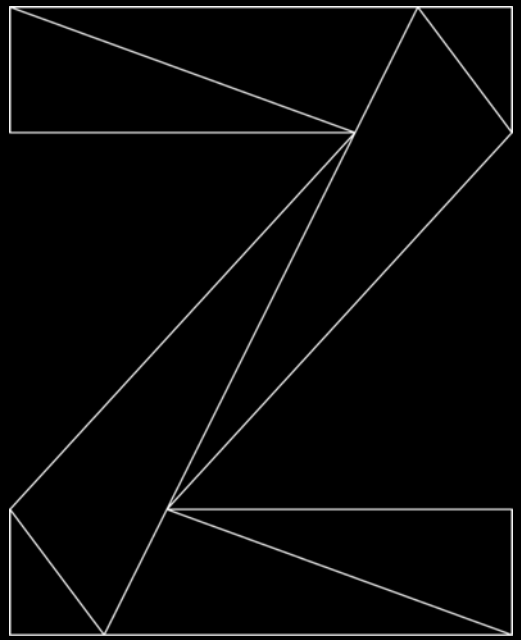
UDX support available for Crypto Express adapters defined as **CCA coprocessors**

- Supports **customized functions** in addition to the CCA API, which **execute inside the secure crypto adapter**
  - –Standard CCA functions plus UDX enhancements available
- Tied to specific versions of the CCA code and the related host code
  - –Must be rebuilt each time these IBM code modules change

Note: Installation of a UDX is a disruptive (non-concurrent) operation.

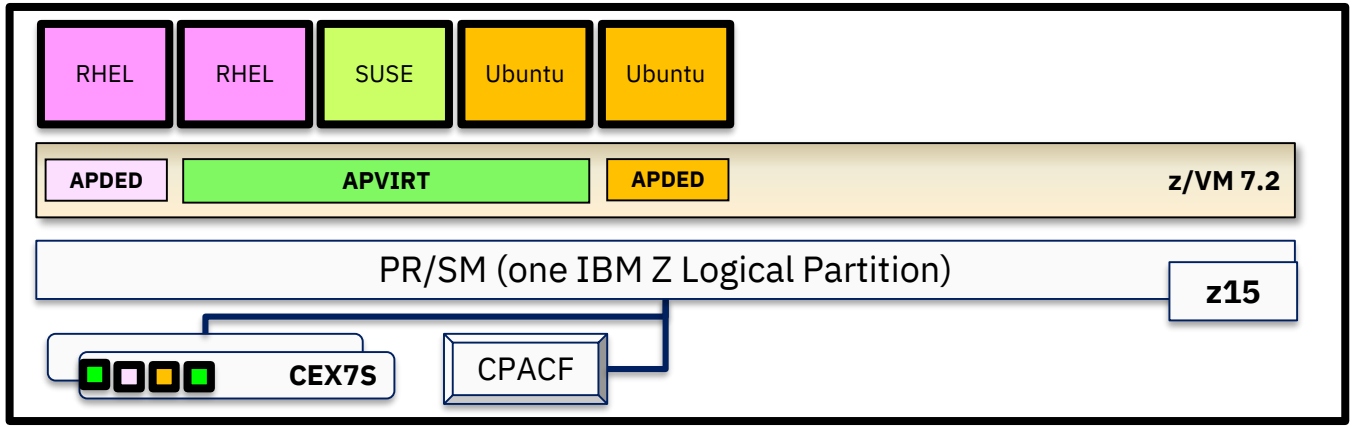


IBM Z Crypto Hardware Virtualization



z/VM Virtualization of Hardware Cryptography

Crypto Express features associated with your z/VM partition are **virtualized for the benefit of your guests**:



`APDED` (“Dedicated”)

Connects a particular AP domain (or set of domains) of one or more Crypto Express adapters directly to a virtual machine – no hypervisor interference

All card functions are available to the guest

`APVIRT` (“Shared”)

Virtual machine can access a collection of domains controlled by the hypervisor layer

Meant for clear-key operations only – sharing crypto material might otherwise break security policy.

Sample of Virtualization: LinuxONE Developer Cloud

Crypto operations: SSH (RSA, SHA-2, AES), and *whatever data is handled inside the guests*

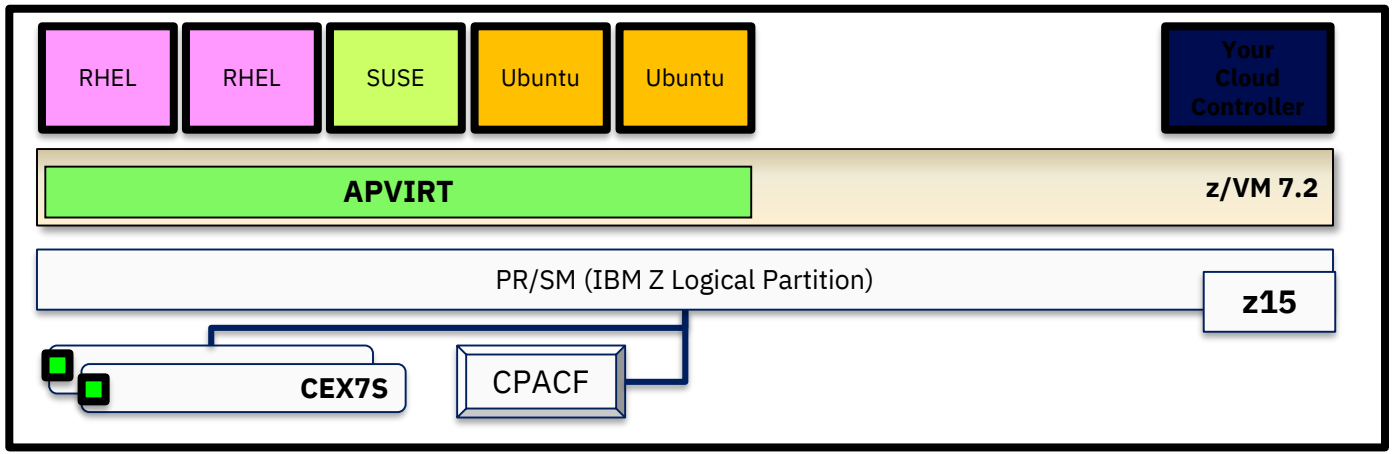
Environmental Requirements: Relocatable (it's a cloud)

Recommended Hardware:

– CPACF

– Crypto Express Accelerator in shared configuration (“APVIRT”)

- Assign 1 domain from 2-3 different features (hardware failover, performance)



Sample of Virtualization: Linux on IBM Z Blockchain (*not* HSBN)

Crypto operations: A lot. It's a Blockchain

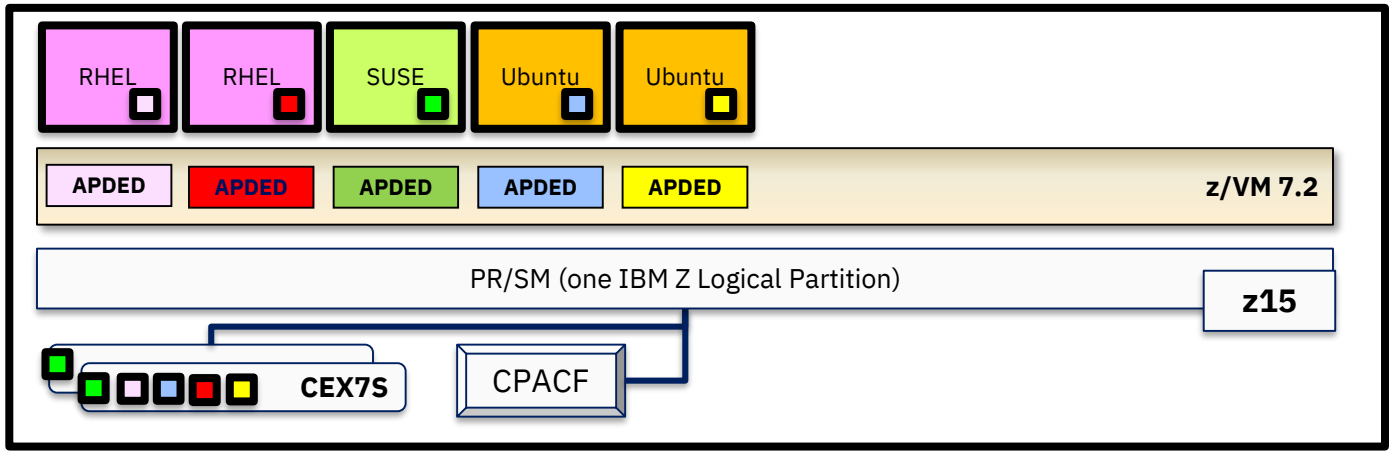
Environmental Requirements: Protection of key material. (It's a Blockchain.)

Recommended Hardware:

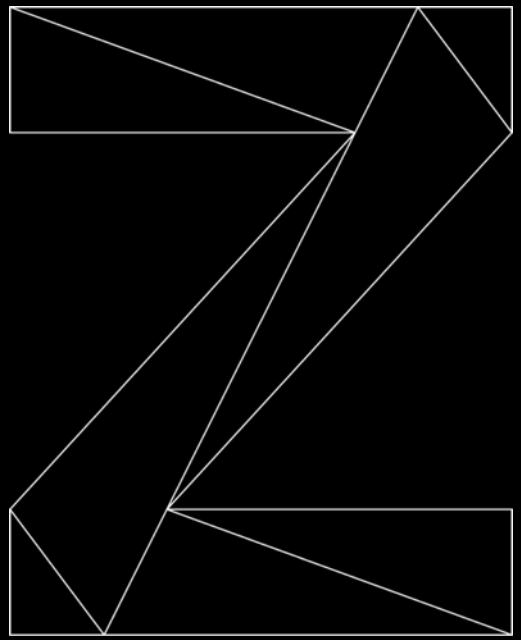
– CPACF (required for secure and protected key ops on the crypto adapters)

– Crypto Express Coprocessors

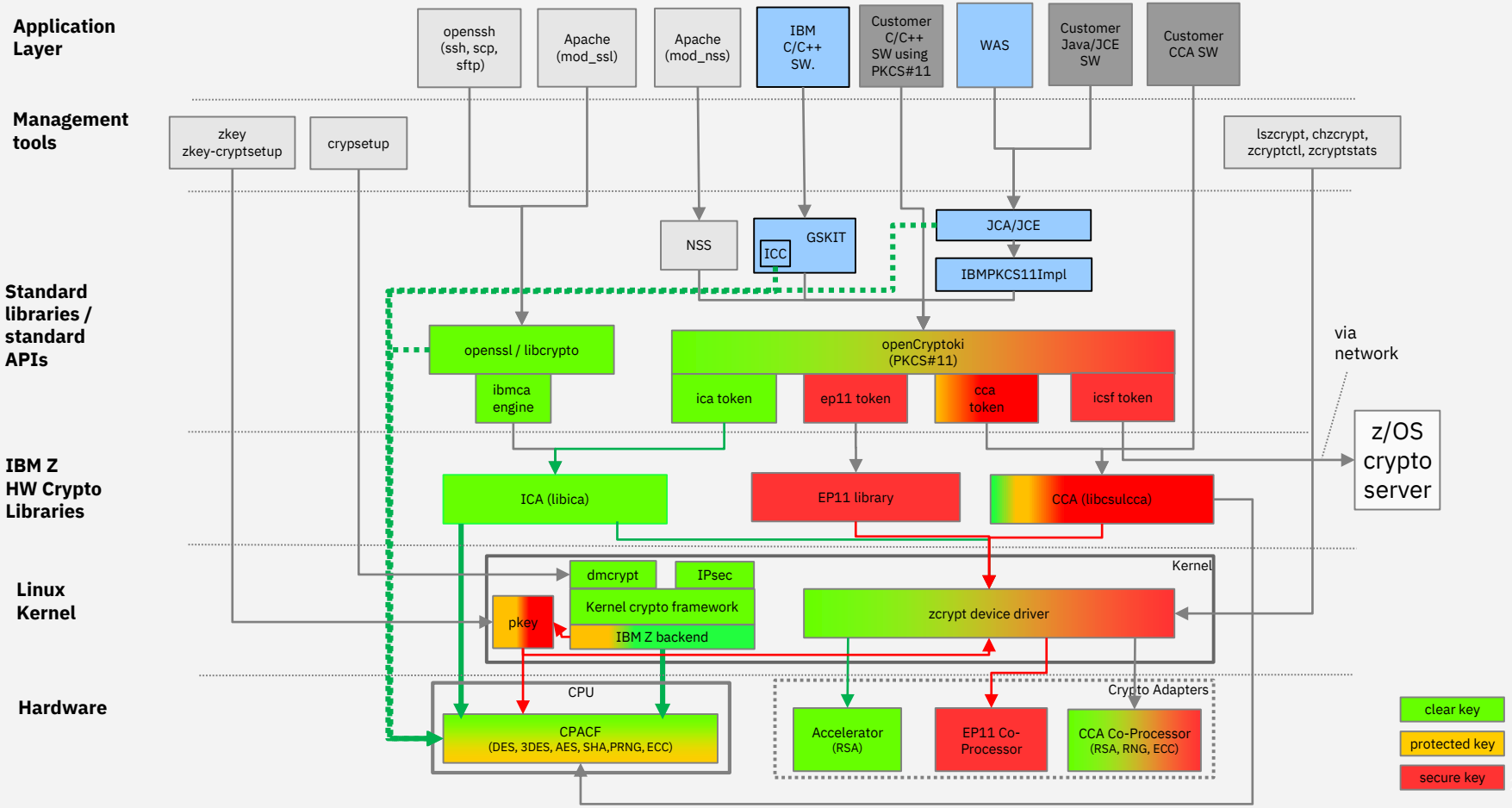
- One EP11 domain per guest participating in the Hyperledger fabric



IBM Z Crypto Software
Linux on z & LinuxOne



Linux on IBM Z crypto stack



The libica library

The libica library provides a C API for

Symmetric crypto & hash mechanisms (clear key CPACF support)

- Hash: SHA1, SHA224, SHA256, SHA384, SHA512
- DES / 3DES: ECB*, CBC*, CBC_CS, CFB, OFB, CTR, CBC_MAC, CMAC
- AES128/192/256: ECB*, CBC*, CBC_CS, CFB, OFB, CTR, CBC_MAC, CMAC
- AES 128/256: XTS
- AEAD: AES128/192/256: CCM, GCM

Clear key RSA up to 4k moduli

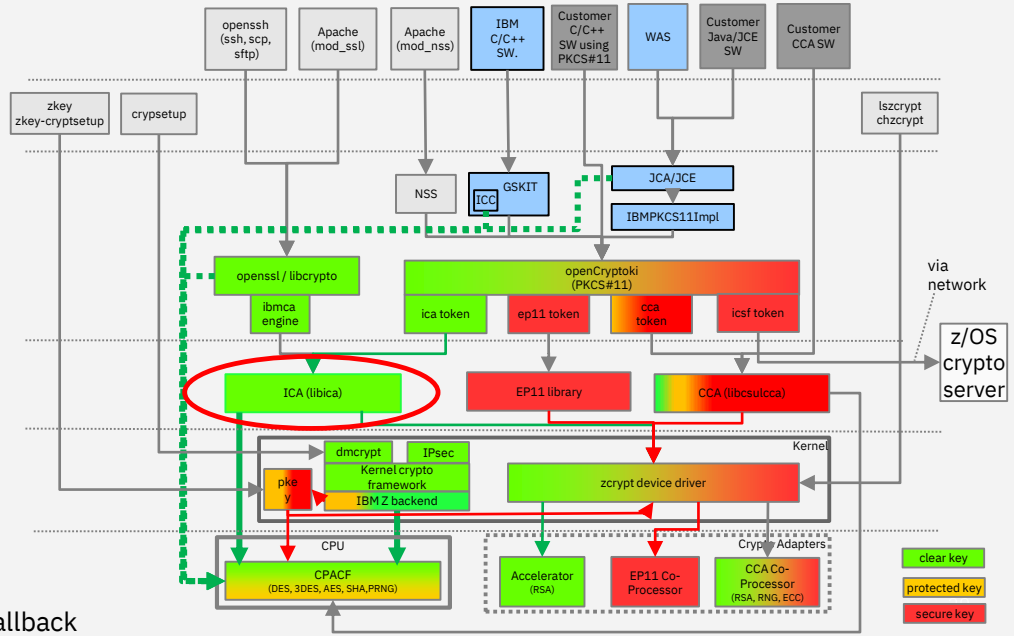
- CCA-coprocessor & accelerator support
- modular exponentiation:
  - ME*: encrypt/verify (decrypt/sign)
  - CRT*: decrypt/sign
- key generation (software only)

Elliptic-curve crypto

- via CPACF:
  - ECSDSA (P256, P384, P521, Ed25519, Ed448)
  - ECDH (P256, P384, P521, X25519, X448)
- via CCA adapter

Pseudo random numbers

- CCA-coprocessor/CPACF/kernel



*) with software fallback

openSSL / libcrypto

openSSL implements SSL and TLS protocols (TLS 1.3 since version 1.1.1)

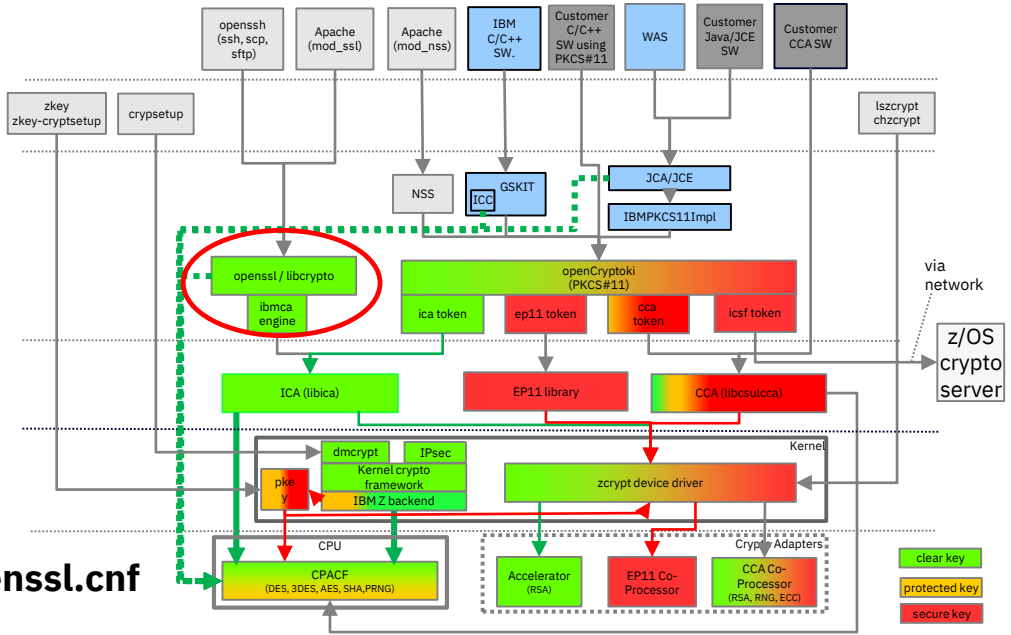
*) accepted upstream by the openSSL project.
**) backported to
- RHEL 7.6, RHEL 8.0,
- SLES 12 SP4, SLES 15,
- Ubuntu 18.04

libcrypto is the crypto library of openSSL
- used by many open source projects
- e.g. openssh, apache mod_ssl, nodes.js, PHP, postgres, MongoDB EE, Ruby

Version >= 1.0.x libcrypto has built-in CPACF support
- CPACF: SHA1, SHA2, SHA3*
- CPACF: AES: ECB, CBC, CTR, OFB*, CFB*, XTS, GCM**
- CPACF: GHASH
- CPACF: ECC*
- IBM Z SIMD: chacha20*, poly1305*
- IBM Z assembler: long number arithmetic

The ibmca dynamic engine supports
- CPACF: SHA1, SHA256
- CPACF: DES/3DES/AES: ECB, CBC, CFB, OFB
- CPACF: ECC
- CEX*A/CEX*C: RSA, DH, DSA
- CPACF, CEX*C: pseudo random number generation

Usage of ibmca engine must be configured in openssl.cnf



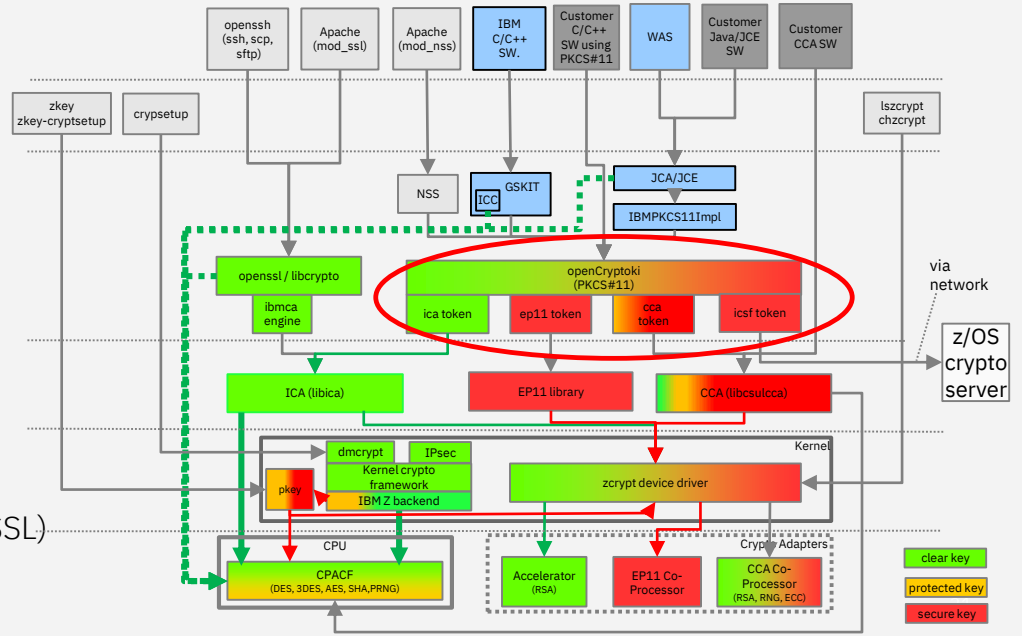
openCryptoki token types for Linux on IBM Z

ICA token
- Provides clear key cryptographic functions
- Uses libica
- Exploits CPACF, Crypto Express accelerators and CCA co-processors
- IBM Z specific

CCA token
- Provides secure key cryptographic functions
- Uses CCA library (libcsulcca)
- Exploits Crypto Express CCA co-processors
- IBM Z specific

EP11 token (since openCryptoki 3.1)
- Provides secure key cryptographic functions
- Exploits Crypto Express EP11 co-processors
- IBM Z specific

Soft token
- Provides clear key cryptographic functions
- Pure software implementation, relies on libcrypto (openSSL)
- Platform independent



Note: openCryptoki implements PKCS #11

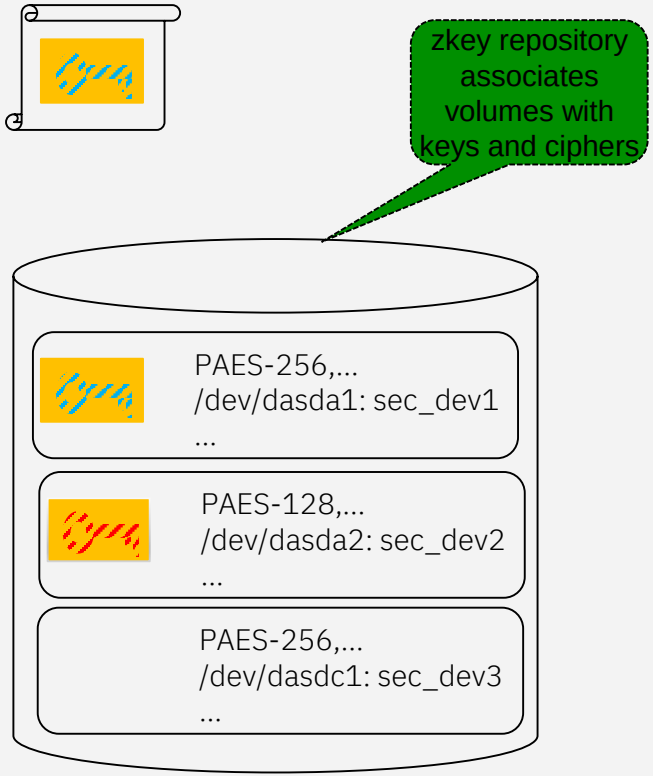
The zkey tool and repository

Basic function

– Generate files containing secure keys
– Re-encipher secure keys

Repository functions

– Generate secure keys in repositories associated with
  • Volume info
  • Cipher info
  • ...
– Show, modify repository contents
– Generate cryptsetup commands (for plain und LUKS2)
– Re-encipher keys in repository



Solves key volume association issue of plain format

Crypto adapter utilization

zcryptstats tool

works in LPAR only

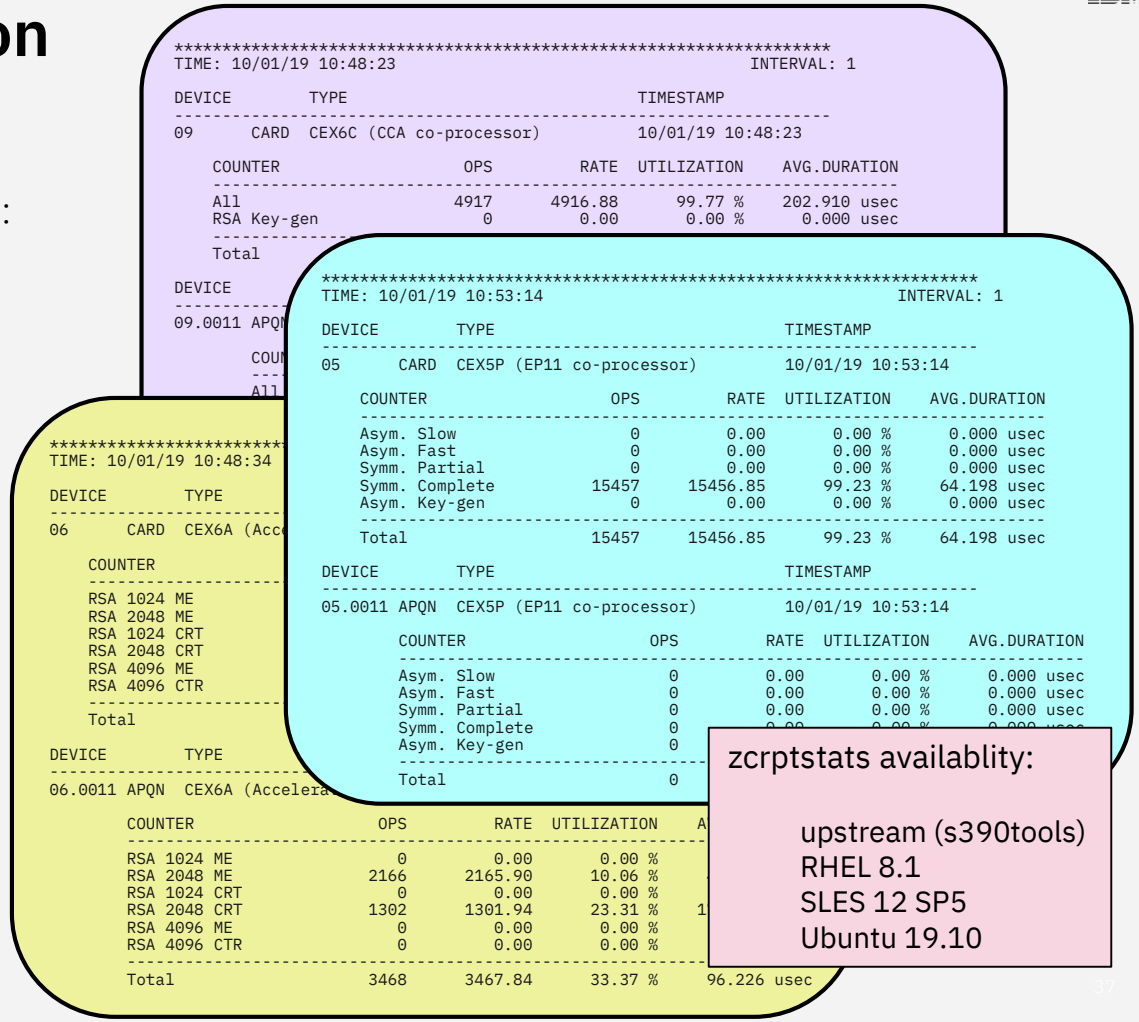
displays usage statistics of crypto adapters:

- Number of operations
- Rate (ops/sec)
- Utilization (%)
- Average duration of operation

per

  - adapter / AP queue
  - operation class / total

- Configurable
  - measuring interval
  - number of measurements
- Output formats
  - JSON
  - table (see right part of slide)
  - CSV

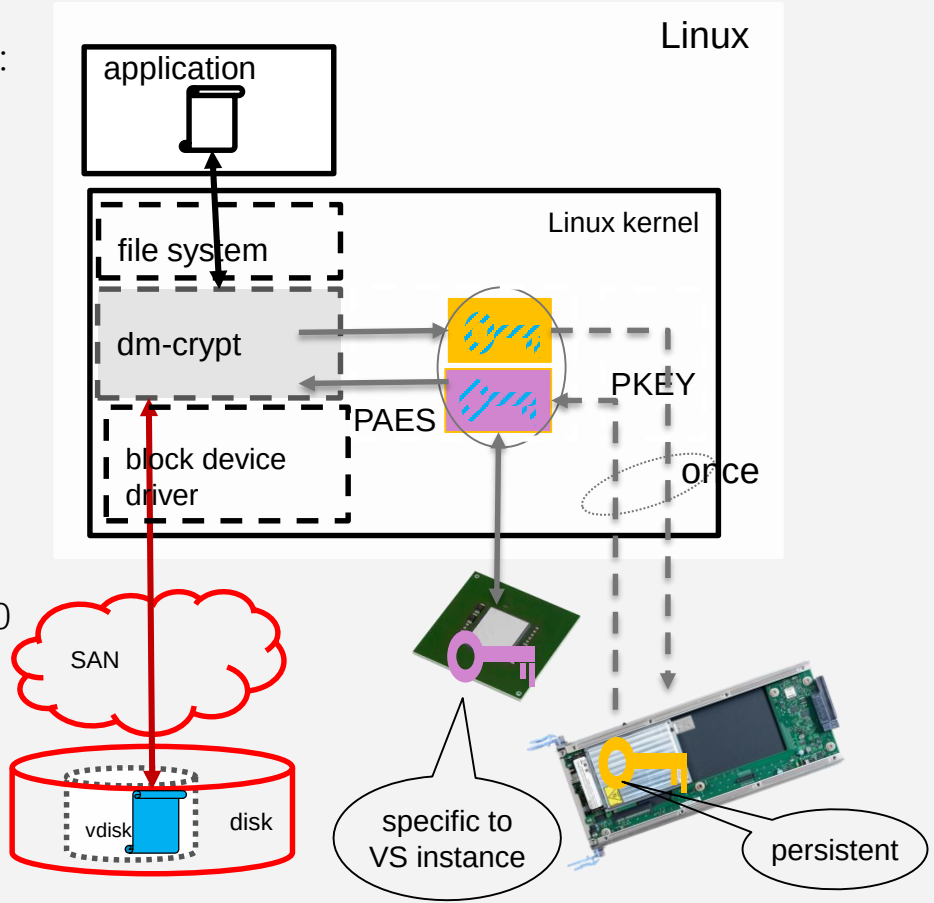


End-to-end data-at-rest encryption with protected key dm-crypt

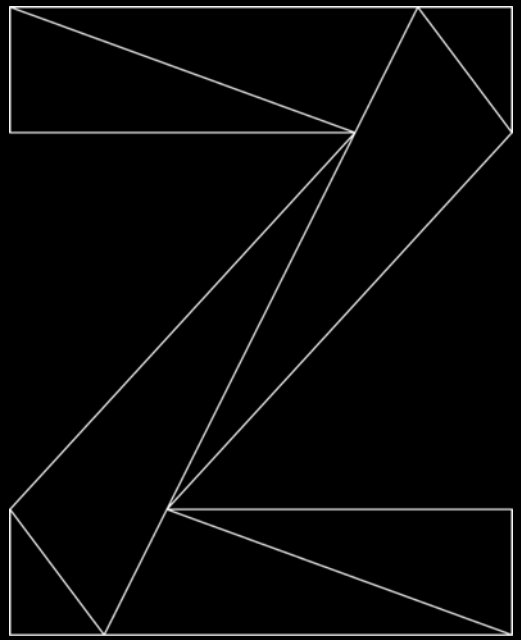
E2E data encryption
- The complete I/O path outside the kernel is encrypted:
  - HV, adapters, links, switches, disks

dm-crypt
- a mechanism for end-to-end data encryption
- data only appears in the clear in application
- Linux kernel component that
  - transparently for all applications
  - for a whole block device (partition or LV)
    - encrypts all data written to disk
    - decrypts all data read from disk

IBM Z support
- uses in-kernel crypto
  - can use IBM Z CPACF clear key crypto (with aes_390 installed)
  - can use IBM Z CPACF protected key crypto:
    - XTS-PAES as from the paes_s390 module
- Dm-crypt disk formats that support XTS-PAES
  - Plain format
  - LUKS2 format



IBM Z Crypto Software
z/OS



z/OS Integrated Cryptographic Services Facility (ICSF)

How are crypto operations performed on z/OS?

ICSF provides the application programming interfaces by which applications request cryptographic services such as:

- Encryption and Decryption
- Digital Signature Generation and Verification
- MAC Generation and Verification
- HMAC Generation and Verification
- Key and Key Pair Generation
- Key Derivation
- Key Agreement
- Data Hashing
- Random Number Generation
- Financial PIN Generate / Verify / Translate / Encrypt

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

ICSF uses cryptographic keys to:
Protect data
Protect and distribute additional keys
Verify message integrity
Generate, protect and verify PINs
Generate and verify signatures

ICSF provides panels to load CCA master key values onto secure cryptographic features, allowing the hardware features to be used by applications.

Common Cryptographic Architecture (CCA)

IBM Common Cryptographic Architecture

IBM proprietary cryptographic application programming interface (API) providing a broad range of cryptographic services including

standard cryptographic algorithms

financial services standards

z/OS ICSF Naming Conventions for CCA

CSNB* = CCA 31-bit Symmetric Key API
CSNE* = CCA 64-bit Symmetric Key API
CSND* = CCA 31-bit Asymmetric Key API
CSNF* = CCA 64-bit Asymmetric Key API

CCA Functions & Algorithms

- Encrypt / Decrypt (AES, DES, DES3, RSA)
- Sign / Verify (RSA, ECC)
- MAC Generate / Verify (AES, DES, DES3)
- HMAC Generate / Verify (HMAC)
- Key Generate (AES, DES, DES3, HMAC)
- Key Pair Generate (RSA, ECC)
- Key Agreement (ECC, DH)
- One Way Hash
- Random Number Generate
- Key Import / Export
- TR-31 Block Import / Export
- Financial Crypto
- PIN Generate / Verify / Translate
- PIN Encrypt
- Diversified Key Generate
- Derive Unique Key Per Transaction (DUKPT)
- … And Many More!

PKCS #11 Cryptographic Token Interface Standard

PKCS #11 Cryptographic Architecture

Originally published by RSA Laboratories, now maintained by OASIS

– Standard API for devices that hold cryptographic information and perform cryptographic functions

z/OS ICSF Naming Conventions for PKCS #11

CSFP* = PKCS #11 APIs

PKCS #11 Functions & Algorithms

- Encrypt / Decrypt (AES, DES, TDES, RSA)
- Sign / Verify (RSA, DSA, ECDSA)
- HMAC Generate / Verify
- Key Generate (DES, TDES, AES, Blowfish, RC4)
- Key Pair Generate (RSA, DSA, EC)
- Key Derivation
- Domain Parameter Generation (DH)
- One Way Hash
- Random Number Generate
- Wrap / Unwrap Key

Designed for portability and FIPS/Common Criteria certification

z/OS ICSF Key Data Sets

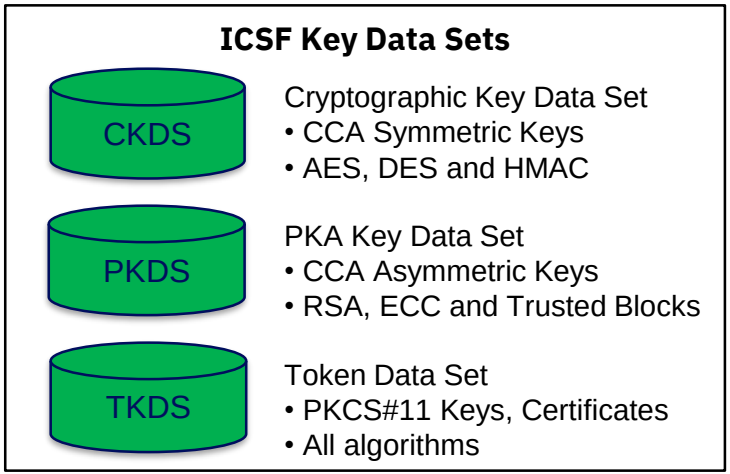
ICSF provides callable services and utilities to generate and store operational keys into ICSF Key Data Sets (KDS) and/or return the keys to the caller

Each KDS is a VSAM data set for persistent objects (e.g. keys, certificates) with programming interfaces for object management.

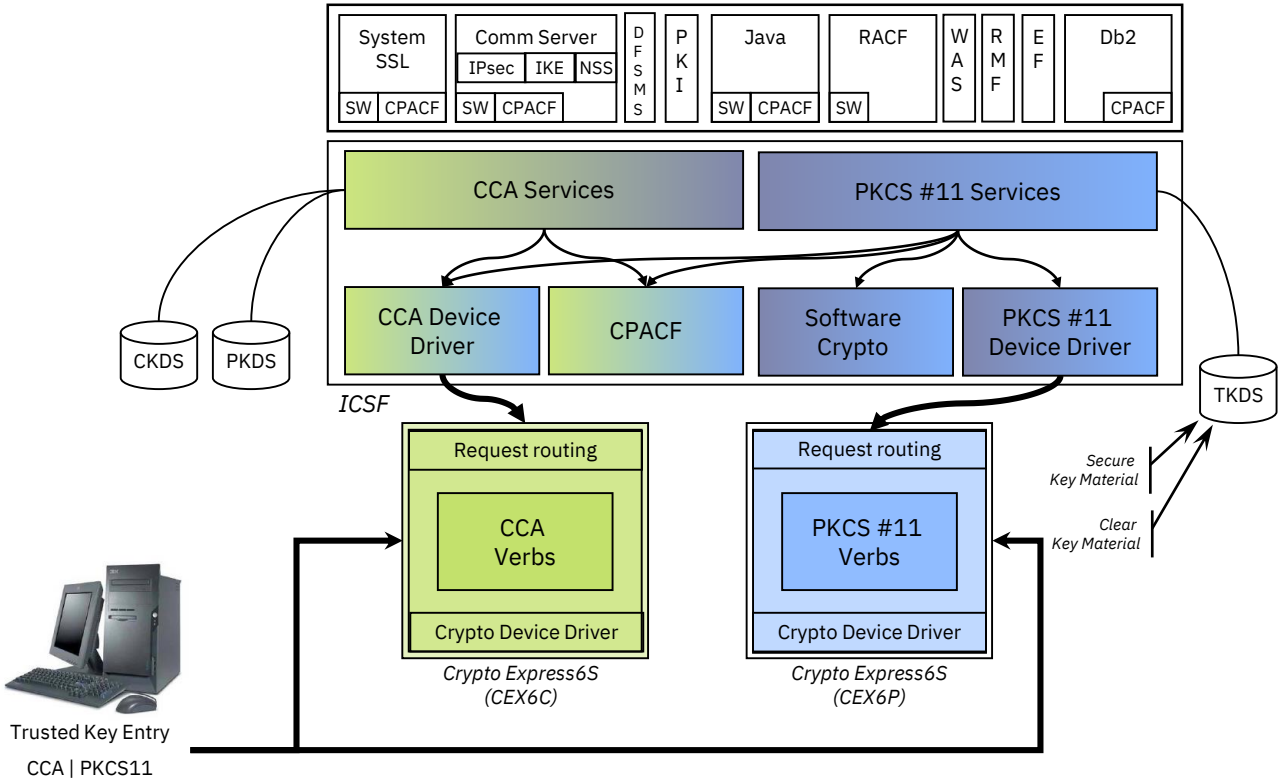
Each record in the KDS contains the object and other information about that object.

ICSF uses operational keys in cryptographic functions to

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



IBM Z Crypto Stack – z/OS



Which z/OS components exploit ICSF?

z/OS Software Components
System SSL
Java Cryptography Extension
RACF Security / RACDCERT
Db2 Database
PKI Services
IBM Tivoli Directory Server
Kerberos Network Authentication Service
Websphere MQ
Websphere Application Server
z/OS Communications Server
z/OS DFSMS
z/OS XCF and XES
OpenSSH
Kerberos (Network Authentication Services)
...

IBM & ISV Solutions
- Sterling Connect:Direct
- ...

How do you view the contents of a Key Data Set?

ICSF supports utilities that allow management of records in the Key Data Sets:

- CKDS – CKDS KEYS utility
- PKDS – PKDS KEYS utility
- TKDS – PKCS11 TOKEN utility

When the format of the KDS is the common record format (referred to as KDSR), the list of label will show:

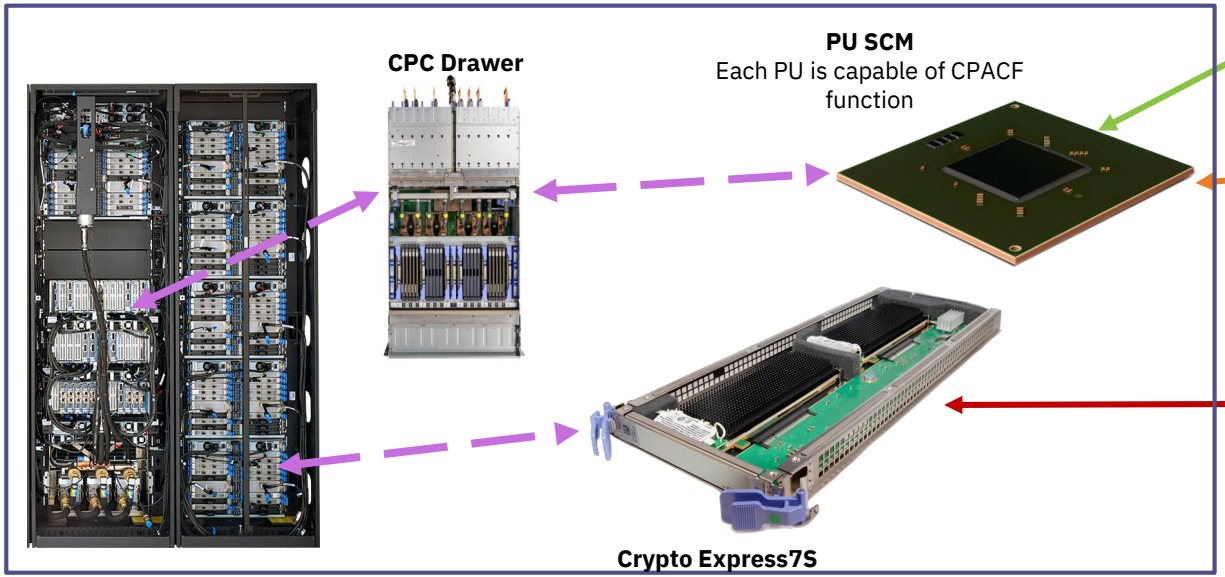
- state of the record (i.e. active, pre-active, deactivated, archived)
- options to display the key attributes and record metadata, delete the record, archive the record or recall the record.

When the format of the KDS is non-KDSR, the options will be to display key attributes and delete the record.

Note: Alternative methods to view the contents of a Key Data Set include IDCAMS REPRO, PKCS #11 Token (TKDS) Browser and the Key Dataset List (CSFKDSL) callable service.

What are clear, secure and protected keys?

Secure keys have key values that are encrypted by a Master Key on a tamper-responding CryptoExpress adapter.



High speed

Clear Key

Key values are not encrypted. Crypto operations may be performed in CPACF or on a Crypto Express adapter

Speed & Security Hybrid

Protected Key

Key values are encrypted under a CPACF wrapping key. Crypto operations are performed only using CPACF

High security

Secure Key

Key values are encrypted under a Master Key. Crypto operations are performed only on a Crypto Express adapter

Note: With z/OS data set encryption, protected keys are implicitly created from secure keys.

How do you audit key usage and the key life cycle?

Key usage auditing must be explicitly enabled in the ICSF Installation Options Data Set (IODS) or using the SETICSF OPT operator commands.

ICSF IODS Option	SMF Record Type
AUDITKEYUSGCKDS(TOKEN(YES),LABEL(YES),INTERVAL(*n*))	Type 82 Subtype 44
AUDITKEYUSGPKDS(TOKEN(YES),LABEL(YES),INTERVAL(*n*))	Type 82 Subtype 45
AUDITPKCS11USG(TOKENOBJ(YES),SESSIONOBJ(YES),NOKEY(YES),INTERVAL(*n*))	Type 82 Subtype 46 & Type 82 Subtype 47

Key life cycle auditing must be explicitly enabled in the ICSF Installation Options Data Set (IODS) or the SETICSF OPT operator commands.

ICSF IODS Option	SMF Record Type
AUDITKEYLIFECKDS(TOKEN(YES),LABEL(YES))	Type 82 Subtype 40
AUDITKEYLIFEPKDS(TOKEN(YES),LABEL(YES))	Type 82 Subtype 41
AUDITKEYLIFETKDS(TOKENOBJ(YES),SESSIONOBJ(YES))	Type 82 Subtype 42

How do you track crypto usage?

With HCR77C1, ICSF provides crypto usage tracking of applications and components that invoke ICSF services. Crypto usage tracking can be enabled/disabled at ICSF initialization using the **Installation Options Data Set (IODS)** or dynamically using **SETICSF OPT operator commands**.

ICSF IODS Option	SMF Record Type
STATS(ENG,SRV,ALG)	Type 82 Subtype 31

ENG: Tracks crypto engine usage. When enabled, ICSF tracks the usage of Crypto Express Adapters, Regional Cryptographic Servers, CPACF and Software.
SRV: Tracks crypto service usage. When enabled, ICSF tracks the usage of ICSF callable services and User-Defined Extensions (UDX).
ALG: Tracks crypto algorithm usage. When enabled, ICSF tracks the usage of crypto algorithms that are referenced in cryptographic operations.

Crypto usage data collection is synchronized to the SMF recording interval. Your SMFPRMxx member must contain:

- The collection interval (INTVAL)
- The synchronization value (SYNCVAL)
- The Crypto Usage Statistics Subtype 31 for ICSF Type 82 records (TYPE)

Starting ICSF at IPL

- IEASYSxx
  - ICSF=xx where xx is passed to the ICSF PROC
  - ICSFPROC=procname specifies the name of the PROC to start or NONE
- ICSF gets started as a system address space during IPL
- Calls to ICSF before it completes initialization will be paused until ICSF is available.
- Choice of options definition allows options to be shared with older releases of ICSF
  - Options can come from PARMLIB concatenation (CSFPRMxx)
  - Options can come from CSFPARM DD as before

Dynamic Service Activation

System administrators can now activate recently installed ICSF service without requiring an outage.

The **SERVICELIBS(YES)** keyword is supported in the ICSF Installation Options Data Set (IODS) to indicate that service libraries specified in the IODS should be used when starting ICSF.

- A new **SERVSCSFMOD0([dsn[,volser])** option indicates the SCSFMOD0 data set to use when starting ICSF.
- A new **SERVSIEALNKE([dsn[,volser])** option indicates the SIEALNKE data set to use when starting ICSF.

How It Works:

1. Update the service libraries in the IODS and issue **SETICSF OPT,REFRESH**.
2. Issue **D ICSF,SERVICELIBS** to display the SERVICELIBS value, the currently running libraries and the next service library values to be used after a SETICSF PAUSE.
3. Issue **SETICSF PAUSE** to pause new transactions and allow all in-progress transactions to complete.
    a. ICSF will go down after all in-progress transactions complete.
4. ICSF can be restarted using:
    a. Automatic Restart Manager (ARM) policy using new ARM element “SYSICSF_” + system name
    b. Custom automation on CSFM696I SETICSF PAUSE processing complete message
    c. Manual ICSF restart (eg. START CSF) after CSFM696I SETICSF PAUSE processing complete message
5. ICSF will restart with the new service modules.
6. ICSF will resume any paused requests and continue with normal processing.

How do you protect ICSF resources?

ICSF Keys, APIs and Utilities
- The CSFSERV class controls access to ICSF callable services and ***ICSF TSO panel utilities***.
- The CSFKEYS class controls access to cryptographic keys in the ICSF Key Data Sets (CKDS and PKDS) and ***enables/disables the use of protected keys***.
- The CRYPTOZ class controls access to, and defines a policy for PKCS#11 token in the Token Data Set (TKDS).
- The XCSFKEY class controls the ability to export a symmetric key with the Symmetric Key Export callable services.

ICSF Key Data Sets
- The DATASET class can be configured to protect the ICSF Key Data Sets.

ICSF MVS Console Commands
- The OPERCMDS class controls the ability to issue MVS console commands for “DISPLAY ICSF” and “SETICSF”.

Key Store Policy
- Define additional security policies pertaining to the use of key tokens.

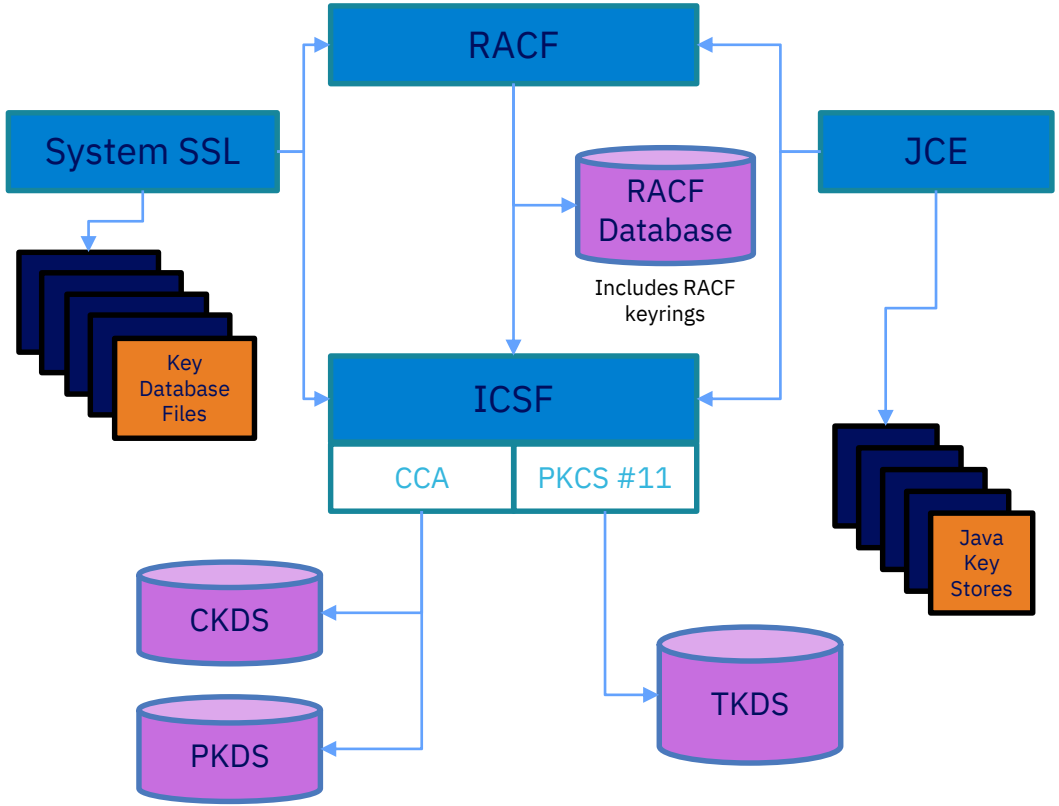
Note: CCA Coprocessor Access Controls on the cryptographic coprocessor can be used to further control cryptographic operations.

Additional z/OS Key Stores

- **RACF** provides the RACDCERT GENCERT command to generate and store keys into the RACF database and ICSF Key Data Sets (PKDS and TKDS). RACF also provides the RACDCERT CONNECT command to add certificates to RACF Keyrings.

- **SystemSSL** provides the gskkyman utility to generate and store certificates into key database files. SystemSSL can also read from RACF Keyrings and generate and store certificates into PKCS#11 Tokens (TKDS).

- **JCE** provides APIs and utilities to generate and store keys and certificates into ICSF Key Data Sets, RACF Keyrings, and Java Key Stores.



Cryptographic network protection on z/OS

z/OS provides 5 mechanisms to protect TCP/IP traffic:

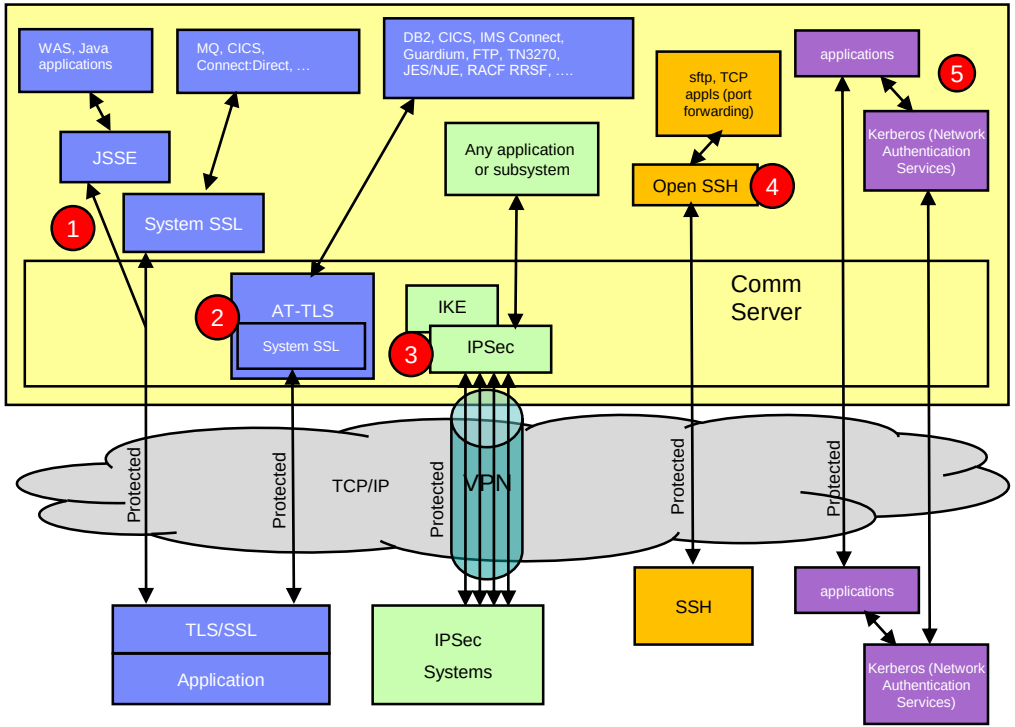
1. **TLS/SSL direct usage**
   - Application is explicitly coded to use this
   - Configuration and auditing is unique to each application
   - Per-session protection, TCP traffic only

2. **Application Transparent TLS (AT-TLS)**
   - TLS/SSL applied in TCP layer using System SSL
   - Configured in policy via Network Configuration Assistant
   - Typically transparent to application

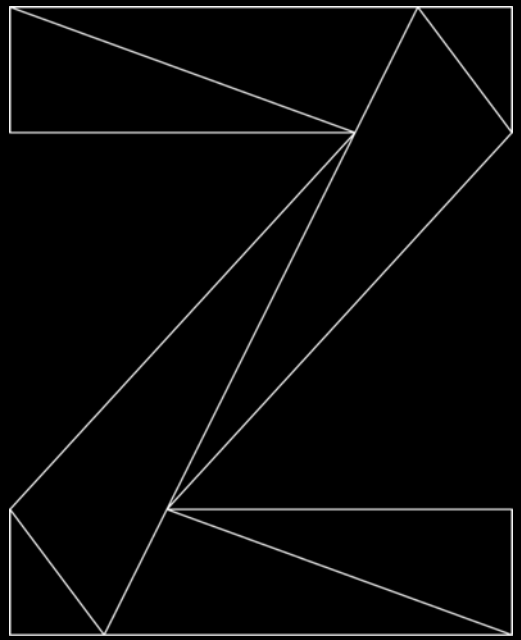
3. **Virtual Private Networks using IPSec and IKE**
   - “Platform to platform” encryption built into IP stack
   - Completely transparent to application
   - Configured in policy via Network Configuration Assistant
   - Wide variety (any to all) of traffic is protected
   - IKE negotiates IPSec tunnels dynamically

4. **Secure Shell using z/OS OpenSSH**
   - Mainly used for sftp on z/OS, but also offers secure terminal access and port forwarding, TCP traffic only
   - Configured in ssh configuration file and on command line

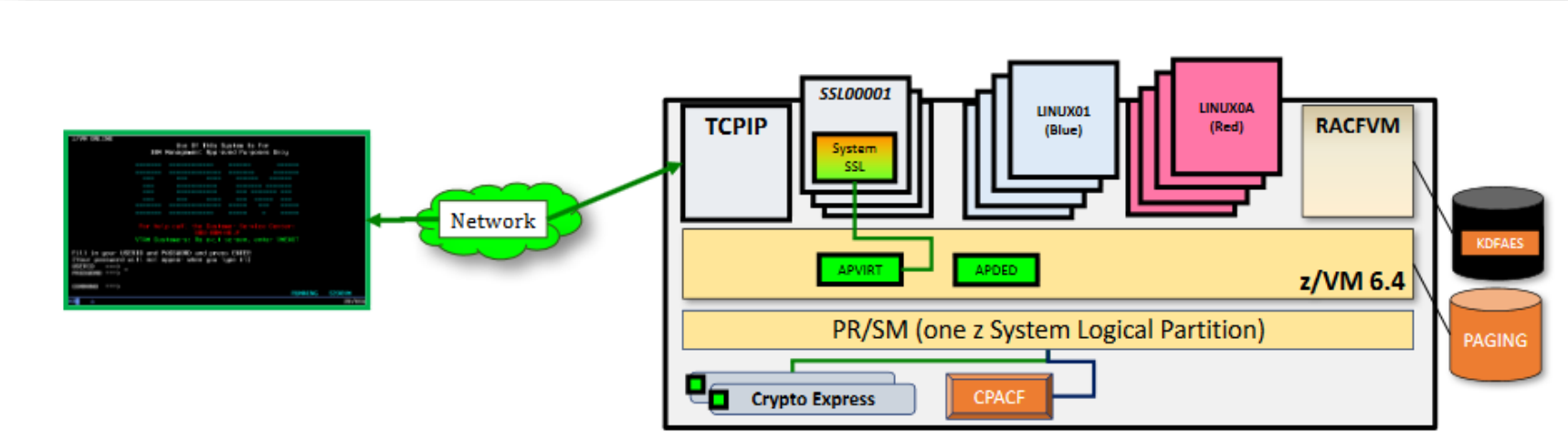
5. **Kerberos (Network Authentication Services)**
   - Primarily authentication services
   - Application is explicitly coded to use this
   - Multiple components configured in their own configuration files



IBM Z Crypto Software
z/VM



IBM Z Crypto Stack – z/VM



Virtualization of hardware crypto features for the benefit of all guests

Software crypto support for z/VM service virtual machines

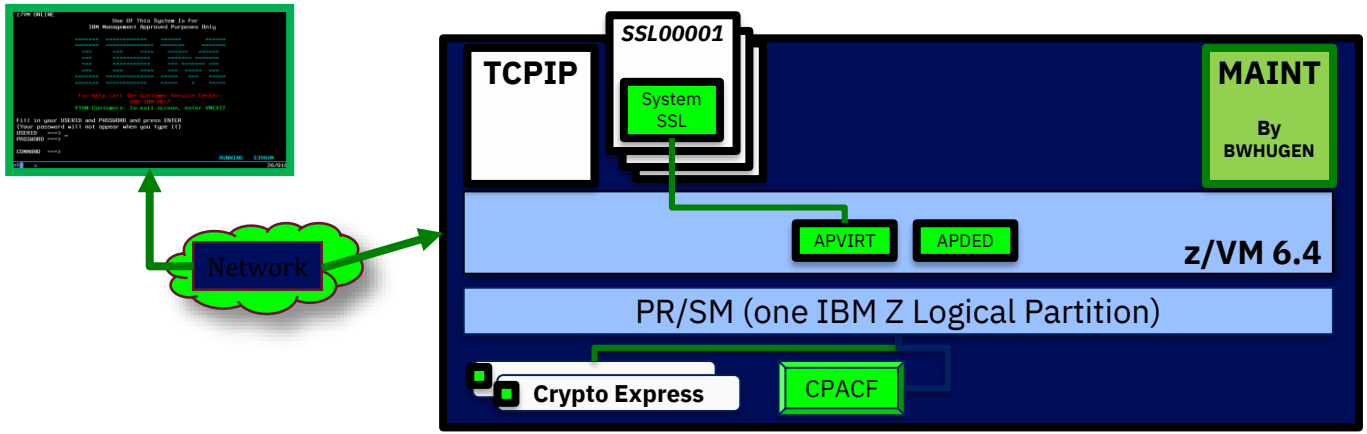
- z/VM System SSL (port of z/OS function) and ICSFLIB
- Pipes crypto stages for CMS application programming
- CPACF and Crypto Express offload whenever available

Some capabilities provided as hardware-only

- KDFAES password encryption for RACF/VM
- z/VM Encrypted Paging for z14 onward

Crypto Acceleration for the z/VM TLS/SSL Server

PTFs for APAR PI72106 for z/VM 6.4; available in later releases



If Crypto Express domains are defined for sharing (APVIRT), then the z/VM TLS/SSL Server will use them

- **Clear-key RSA operations** are the primary beneficiary
  - Handshaking, rather than data transfer – **benefit will come from a lot of connections**
  - Will still use CPACF when pertinent
- Meant as a performance enabler, not to replace key storage

Accelerate cryptographic operations for data in flight

- Connections to hypervisor
- Connections inside of the hypervisor

Dynamic Crypto Support for z/VM

http://www.vm.ibm.com/newfunction/#dynamic_crypto

z/VM 7.1

PTF for APAR VM66266

Dynamic Crypto support enables changes to the z/VM crypto environment without requiring an IPL of z/VM or its guests (e.g. Linux on Z).

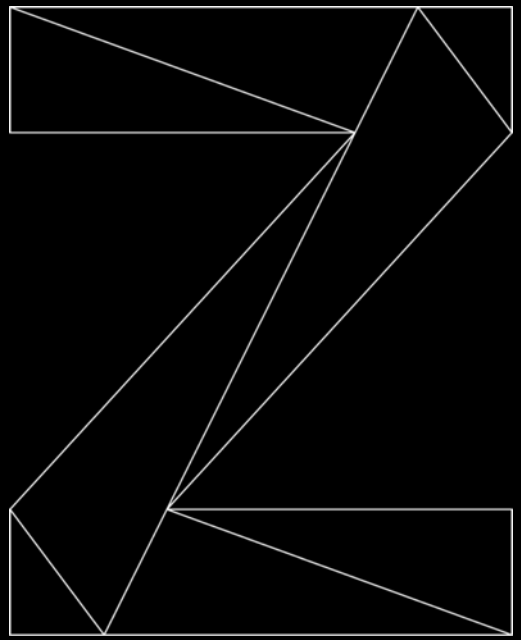
This allows:

- Less disruptive addition or removal of Crypto Express hardware to/from a z/VM system and its guests
- Less disruptive maintenance and repair of Crypto Express hardware attached and in-use by a z/VM system
- Reassignment and allocation of crypto resources without requiring a system IPL or user logoff/logon
- Greater flexibility to change crypto resources between shared and dedicated use.

Additionally, there are RAS benefits for shared-use crypto resources:

- Better detection of Crypto Express adapter errors with "silent" retrying of shared pool requests to alternative resources
- Ability to recover failed Crypto Express adapters
- Improved internal diagnostics for IBM service
- Improved logoff and live guest relocation latency for users of shared crypto.

IBM Z Crypto Software
z/VSE



z/VSE

z/VSE provides hardware-accelerated encryption support by exploiting cryptographic features on z Systems processors.

Crypto Express adapters

– RSA support

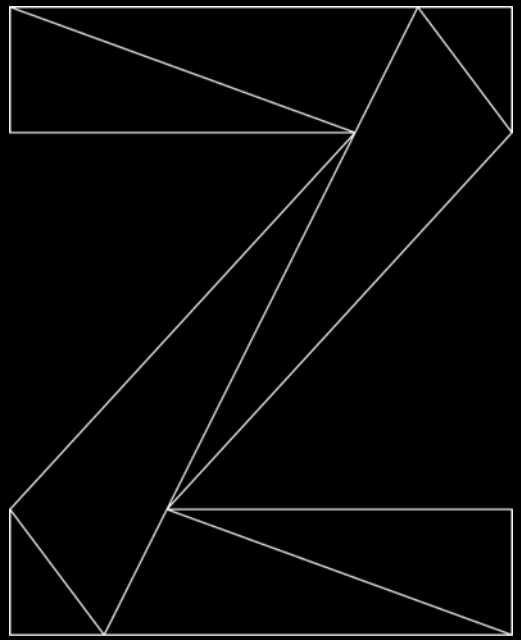
– ECC support

CP Assist for Cryptographic Function (CPACF)

– Symmetric algorithms such as Triple-DES, AES, or SHA.

Cryptographic hardware is transparently used by TCP/IP for z/VSE, IPv6/VSE and applications like Encryption Facility for z/VSE.

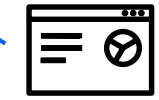
IBM Z Crypto Solutions



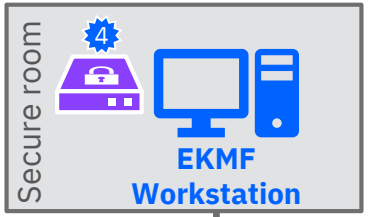
IBM Enterprise Key Management Foundation - EKMF

Browser-based key generation & management for

- ✓ Pervasive Encryption
- ✓ Cloud



Web Browser with EKMF Web

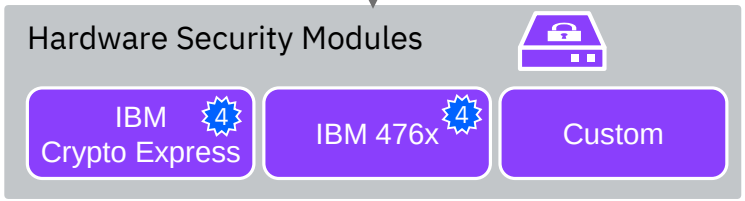


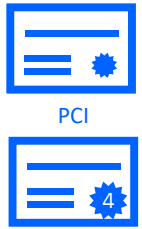
- ✓ Can be placed in secure room
- ✓ Utilizes IBM 4767 HSM
- ✓ Generate new keys by users authenticated with smart cards or automatically based on requests

Central EKMF repository

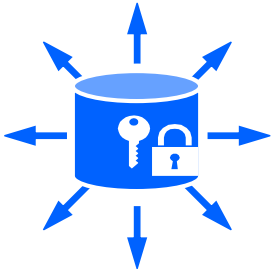
- ✓ Contains keys and metadata for all cryptographic keys produced by EKMF
- ✓ Easy backup and recovery of key material

Cloud key stores





EKMF Workstation features



Single central repository

- Including metadata like activation dates and usage
- Easy backup and recovery
- Easy introduction of new systems

Key Management

- Key generation, import, export, print
- Supporting NIST state model
- Using Hardware Security Modules (HSM)
- Automation & bulk processing

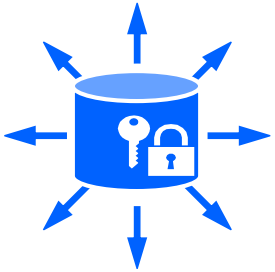
Key Management Policies

- Key Templates to unify key operations
- Import and management of existing keys based on templates
- Central support for multiple z/OS systems

Security & Compliance

- Secure Room is the recommended mode of operation
- Role-based access
- Smart Card authentication
- Dual control for sensitive operations
- Audit logging

EKMF Web for PE features

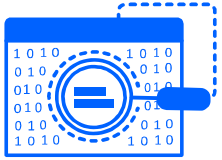


Single central key repository

- Stores metadata (activation dates, usage, etc.)
- Single-point backup and recovery

Key Management

- Generation based on policies
- According to NIST recommendations
- Using Hardware Security Modules (HSM)



Pervasive Encryption Support

- Dataset dashboard
- Import and management of existing PE keys
- Central support for multiple z/OS systems

Security & Compliance

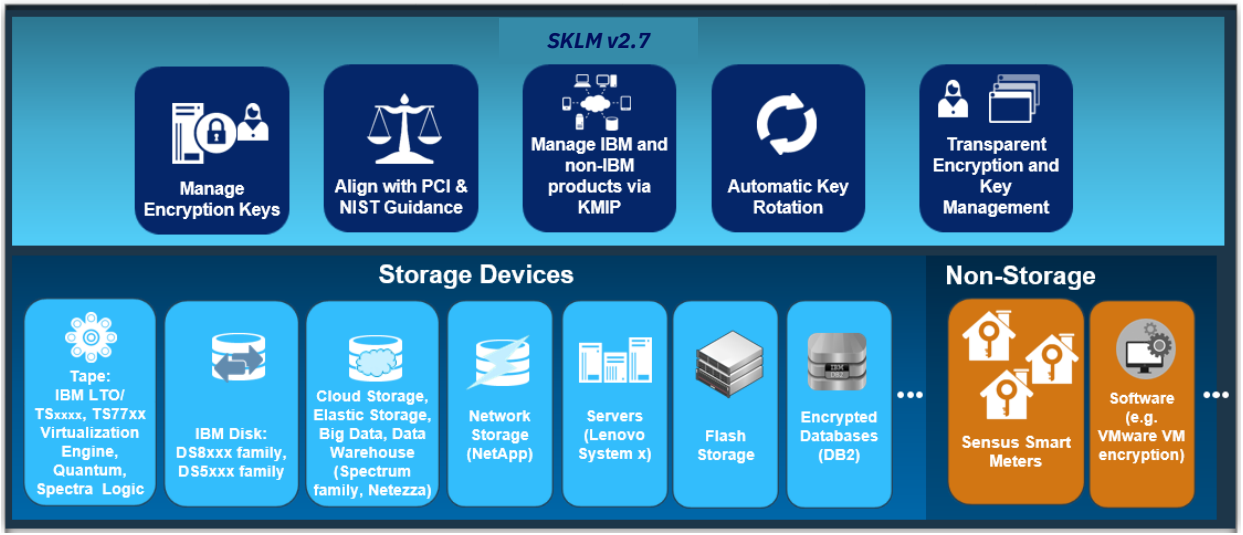
- Role-based access
- Dual control implemented using separation of privileges
- Audit logging

IBM Security Key Lifecycle Manager (SKLM)

IBM Security Key Lifecycle Manager provides centralized key management for self-encrypting devices.

Self-encrypting devices protect data if you lose control of the device.

- Data on the truck traveling between datacenters
- Data at rest within the datacenter
- Decommissioned storage devices



Key Management Features for SKLM

SKLM for Distributed Systems

SKLM v2.7 supports the IBM Proprietary Protocol (IPP) and industry-standard Key Management Interoperability Protocol (KMIP) for key distribution with storage devices.

Features include:

- Key generation, import and export
- Secure storage of key material
- Automatic assignment and rotation of keys
- Key serving at the time of use

SKLM for z/OS

SKLM for z/OS supports the IBM Proprietary Protocol (IPP) for key distribution with storage devices.

SKLM for z/OS can use ICSF through JCE hwkeytool or RACF GENCERT commands to push RSA key pairs to the ICSF PKDS and AES keys to the ICSF CKDS.

Features include:

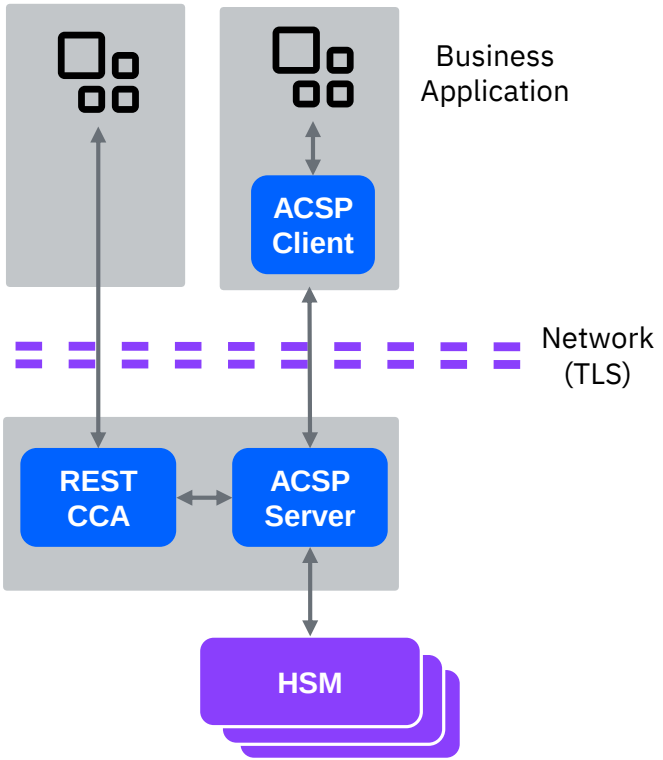
- Key generation, import and export
- Secure storage of key material
- Key serving at the time of use

Note: SKLM can not be used to manage z/OS data set encryption keys.

Crypto-as-a-Service Solution with the EKMF Advanced Crypto Service Provider (ACSP)

Advanced Crypto Service Provider (ACSP) enables applications in distributed environments with **access to cryptographic hardware over the network**

- Cost effective use HSMs
  - Fewer and better utilized HSMs
  - Reduced administration effort
- Easy deployment of cryptographic services
- FIPS 140-2 Level 4 certified HSMs (highest level of security)
- High Reliability, Availability, Scalability to meet crypto SLAs, also during peaks
- Enable Crypto Agility with custom APIs



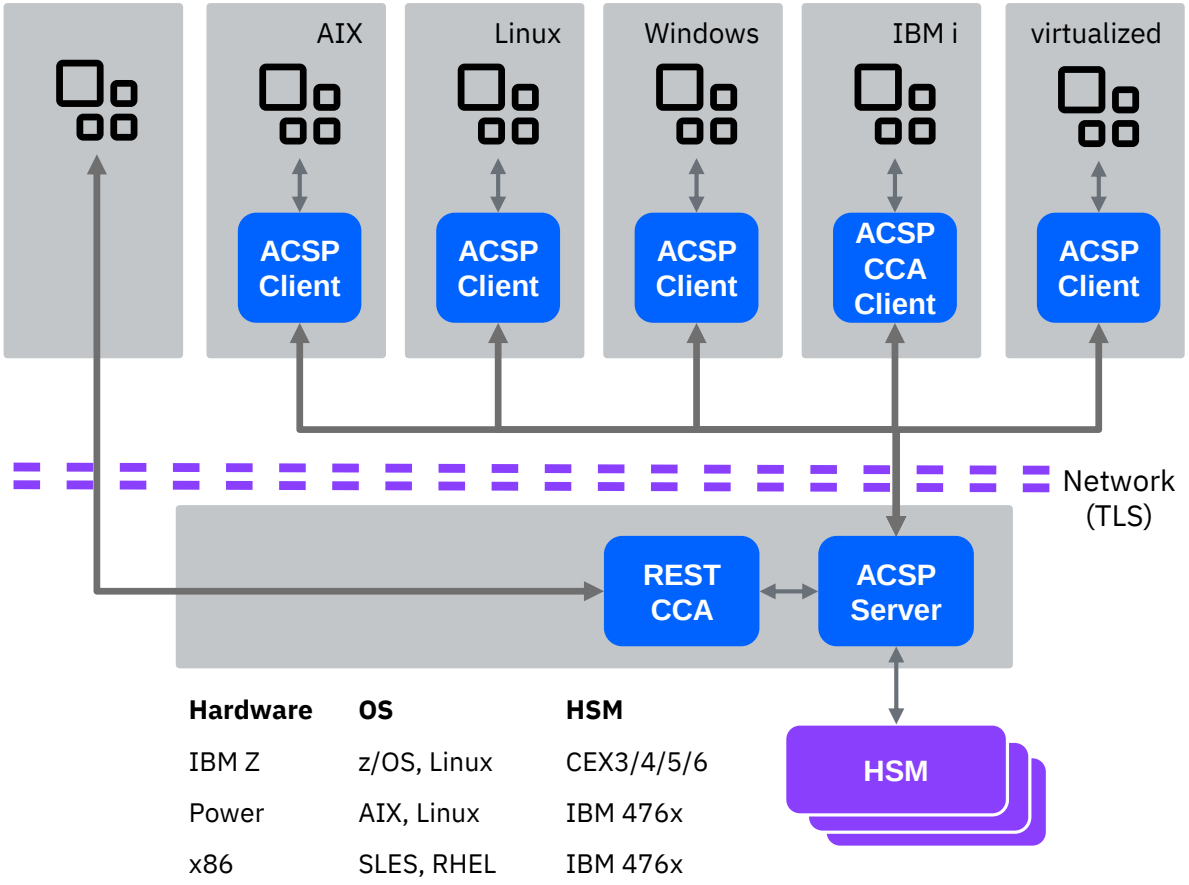
ACSP Supported APIs & Platforms

APIs:

- CCA in Java and C
- PKCS#11
- JCE
- REST services

TLS Transport

- Protected with client authentication
- Certificate to user mapping for granular access control (on z/OS using RACF)

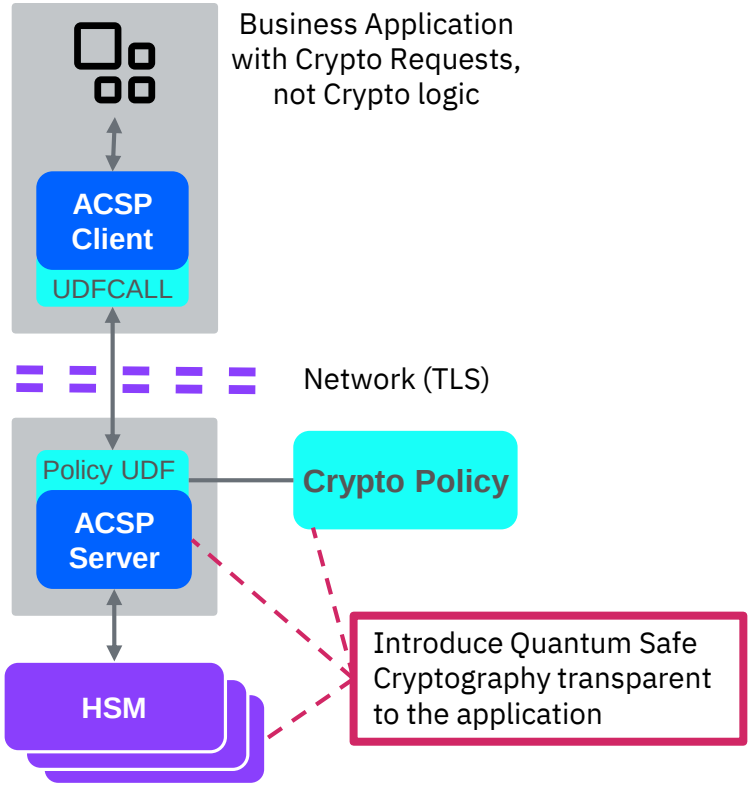


Hardware	OS	HSM
IBM Z	z/OS, Linux	CEX3/4/5/6
Power	AIX, Linux	IBM 476x
x86	SLES, RHEL	IBM 476x

Crypto Agility with ACSP User Defined Functions (UDF)

A User Defined Functions (UDF) is custom code which implements business specific functions.

- Runs in ACSP server, The ACSP Server exposes the UDF supporting the defined Cryptographic Policy
- Extends standard CCA/ICSF API
- Supports a varying number of parameters and parameter types
- Permission checking through RACF
- Ideal for implementing Enterprise Crypto as a Service functions



Encryption Facility (EF) for z/OS and z/VSE

The Encryption Facility is a host-based encryption and key-management solution specifically designed to protect sensitive data that's being exchanged with trusted business partners or archived for backup and recovery purposes.

Provides a business-to-business encryption capability to help companies that rely on exchange of tapes with their partners to complete these business transactions.

Leverages IBM Z software and hardware capabilities to encrypt and compress data as it's sent to tape.

Written in Java, so the client can be downloaded from the Internet and used on multiple platforms.

Encryption Facility provides services for:

- Public-key based encryption
- Passphrase-based encryption
- Modification detection of encrypted data
- Compression of packaged data before encryption
- Importing and exporting of OpenPGP certificates
  - Binary or ASCII armor format
- Digital signatures of data

Data encrypted with EF can be exchanged between different operating systems. EF for z/VSE can read encrypted files that were created by the Encryption Facility for z/OS or the z/OS Java client. The z/OS facilities can read encrypted files that were created by EF for z/VSE. Other platforms than z/VSE or z/OS may encrypt / decrypt such data as well.

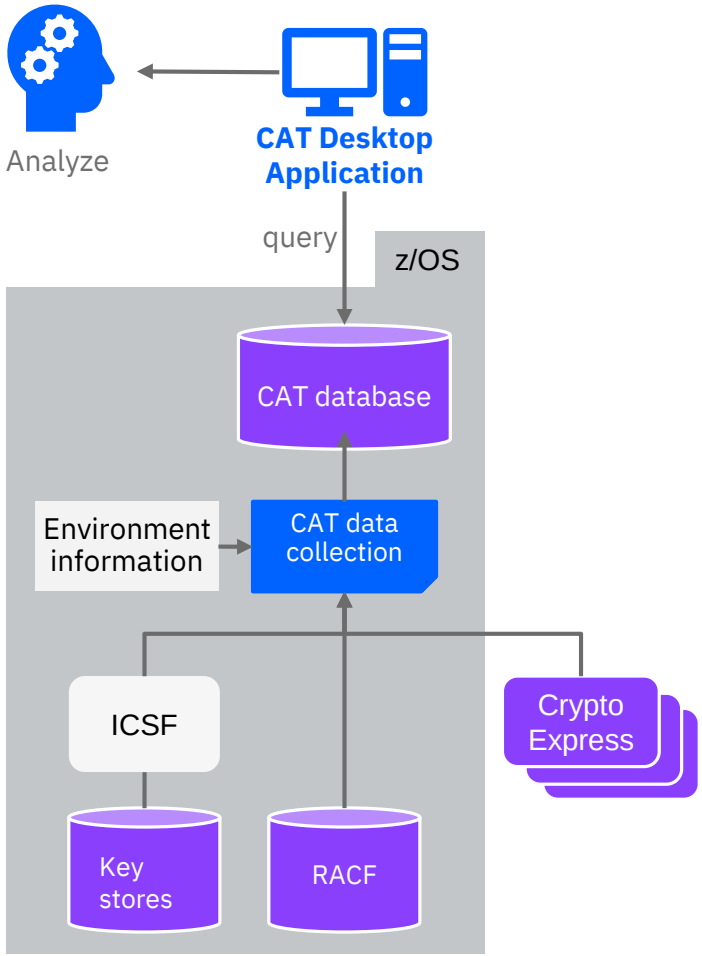
Crypto Analysis Tool (CAT) Overview

Crypto Analysis Tool (CAT)

- Collect security relevant information
- Analyze with easy-to-use graphical client (eclipse-based)

Highlights

- Offers a comprehensive data view of the cryptographic security on the system
- Allows monitoring to ensure that programs, keys and cryptographic functions are set up and protected, **complying with best practices**
- Eases **policy and compliance** enforcement
- Helps Administrators understand **weaknesses** and gaps to prioritize improvements



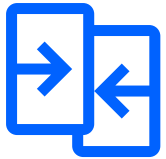
CAT functionality

Comprehensive overview of the cryptographic security of the system

Key data for better policy and compliance enforcement

Define and Compare with policies

Identify insecure keys and algorithms



Comparison of current crypto state with previous snapshot for error and problem determination or change control validation

zSecure Suite

Security audit and compliance

Enhanced data collection z
of SMF audit information from:
- RACF, Db2, CICS, IMS, MQ, SKLM, WAS, UNIX, Linux on System z, OMEGAMON XE on z/OS, FTP, Communication Server, TCP/IP, PDSE and more

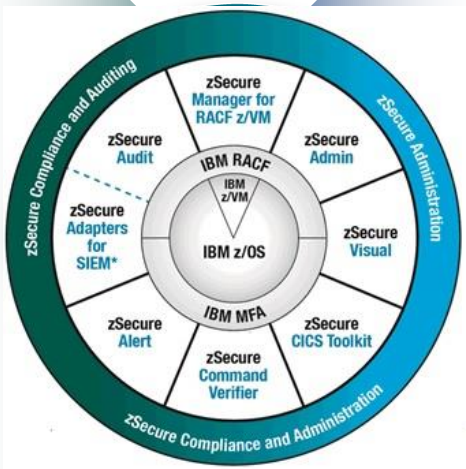
Automated remediation
to detect and prioritize potential threats with security event analysis

Real-time alerts of potential threats and vulnerabilities

Compliance monitoring and reporting
- PCI-DSS, STIGs, GSD331, and site-defined requirements

Comprehensive customized audit reporting

Detect harmful system security settings with automated configuration change checking



Administration management

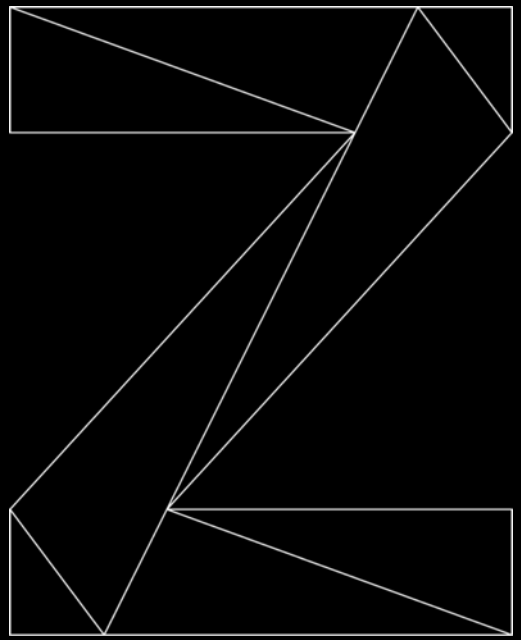
Reduce administrative overhead
with security management tasks

Prevent abuse of special roles and authorization
- Privileged user monitoring
- Entitlement checking for Identity governance

Enforce security policies
by blocking dangerous commands and potential errors

RACF data set cleanup
of unused security profiles and inactive / terminated users

IBM Z Pervasive Encryption



IBM Z Pervasive Encryption

Enabled through full-stack platform integration

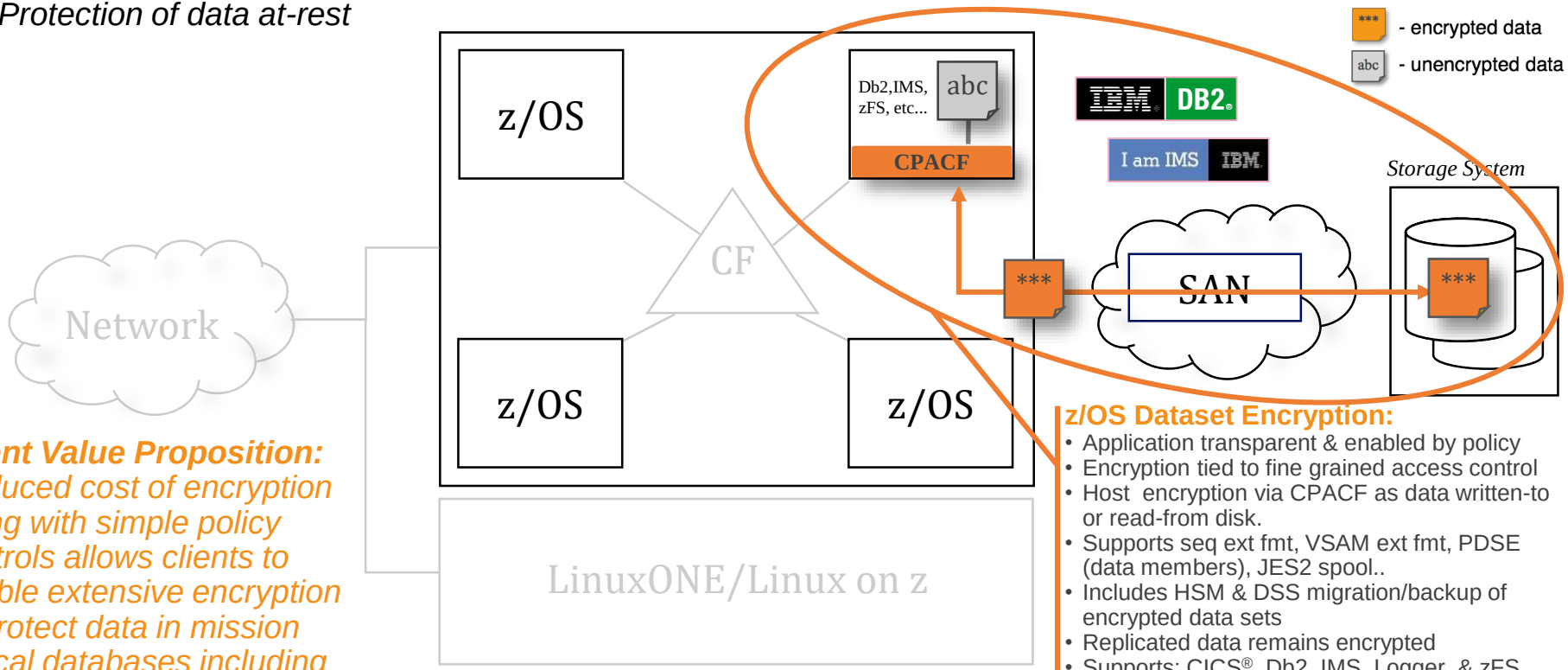
Integrated Crypto Hardware		Hardware accelerated encryption on every core – CPACF performance improvements of up to 7x Next Gen Crypto Express7S – up to 2x faster than prior generation
Data at Rest		Broadly protect Linux volumes and z/OS data sets using policy controlled encryption that is transparent to applications and databases
Clustering		Protect z/OS Coupling Facility data end-to-end, using encryption that's transparent to applications
Network		Protect network traffic using standards based encryption from end to end, including encryption readiness technology to ensure that z/OS systems meet approved encryption criteria
Hyper Protect Virtual Servers		Secure deployment of software appliances including tamper protection during installation and runtime, restricted administrator access, and encryption of data and code in-flight and at-rest
Key Management		The IBM Enterprise Key Management Foundation (EKMF) provides real-time, centralized secure management of keys and certificates with a variety of cryptographic devices and key stores.

And we're just getting started …

Data Protection // z/OS Dataset Encryption

z/OS 2.2 & above

Protection of data at-rest



Client Value Proposition:
Reduced cost of encryption along with simple policy controls allows clients to enable extensive encryption to protect data in mission critical databases including Db2®, IMS™ and VSAM

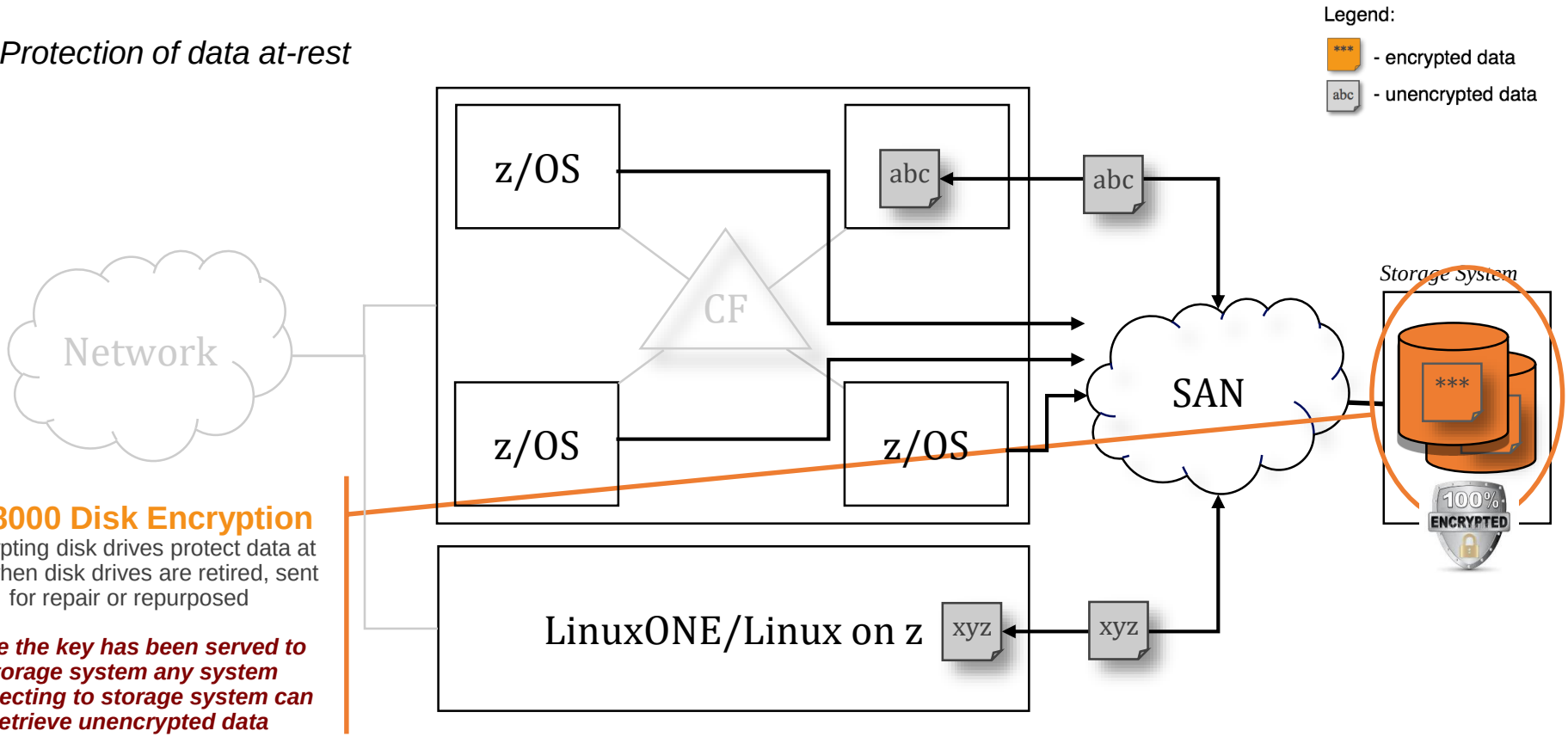
z/OS Dataset Encryption:

- Application transparent & enabled by policy
- Encryption tied to fine grained access control
- Host encryption via CPACF as data written-to or read-from disk.
- Supports seq ext fmt, VSAM ext fmt, PDSE (data members), JES2 spool..
- Includes HSM & DSS migration/backup of encrypted data sets
- Replicated data remains encrypted
- Supports: CICS®, Db2, IMS, Logger, & zFS

In-memory system or application data buffers will not be encrypted

Data Protection // Existing Disk Encryption

Protection of data at-rest



DS8000 Disk Encryption

Encrypting disk drives protect data at rest when disk drives are retired, sent for repair or repurposed

Once the key has been served to storage system any system connecting to storage system can retrieve unencrypted data

Data Protection // Coupling Facility Encryption

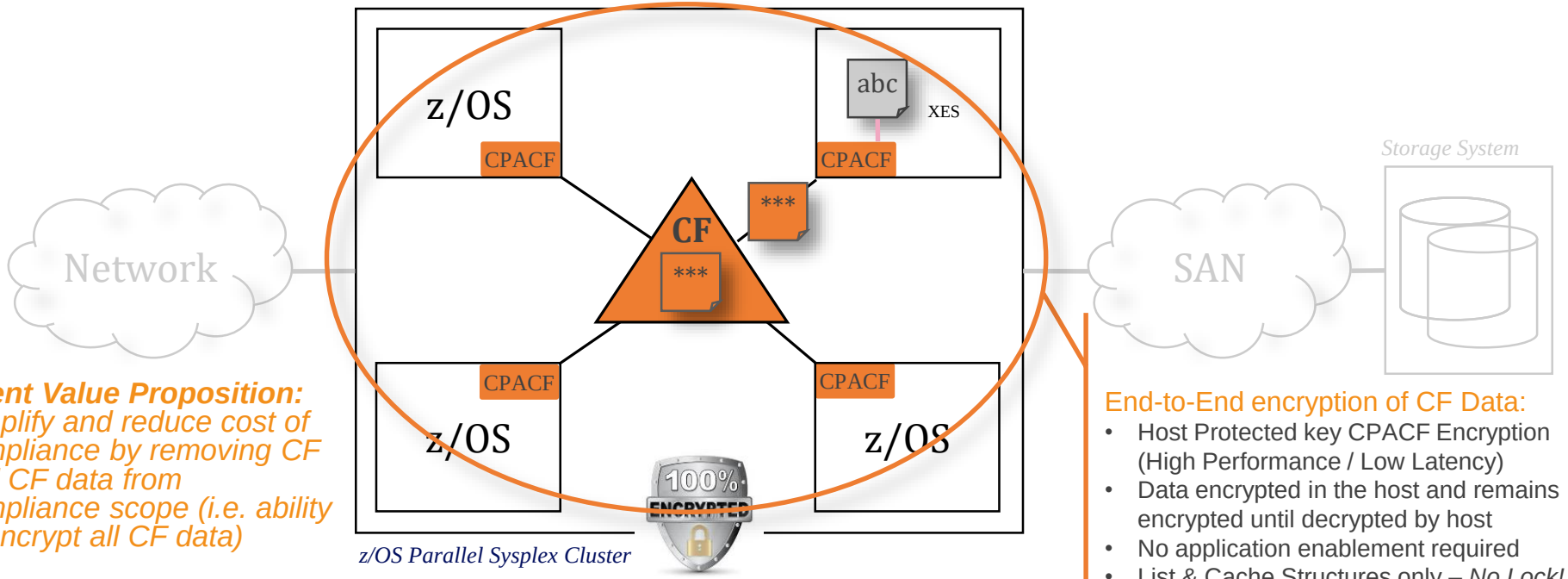
z/OS 2.3

Legend:

*** - encrypted data

abc - unencrypted data

Protection of data in-flight and in-use (CF)



Client Value Proposition:
Simplify and reduce cost of compliance by removing CF and CF data from compliance scope (i.e. ability to encrypt all CF data)

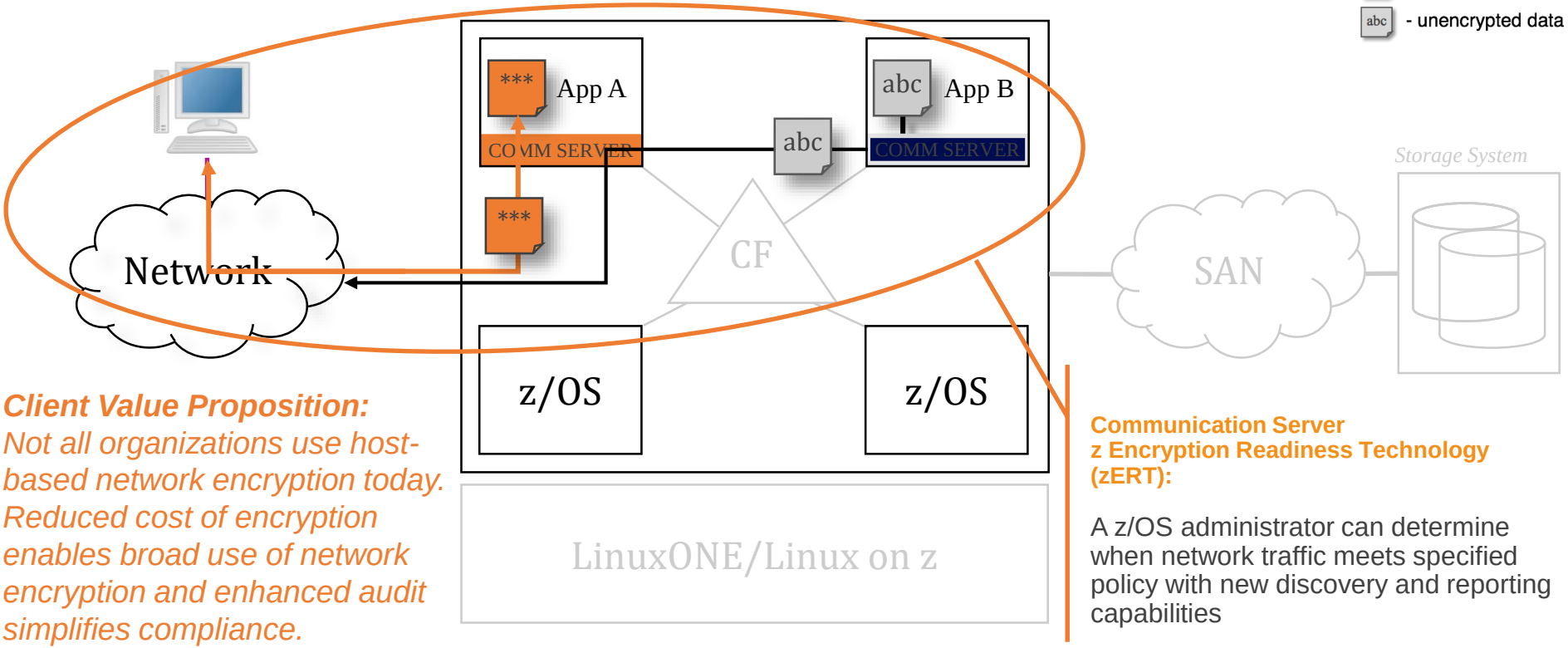
End-to-End encryption of CF Data:

- Host Protected key CPACF Encryption (High Performance / Low Latency)
- Data encrypted in the host and remains encrypted until decrypted by host
- No application enablement required
- List & Cache Structures only – *No Lock!*

Data Protection // z/OS Network Security

z/OS 2.3

Protection of data in-flight



Client Value Proposition:
Not all organizations use host-based network encryption today. Reduced cost of encryption enables broad use of network encryption and enhanced audit simplifies compliance.

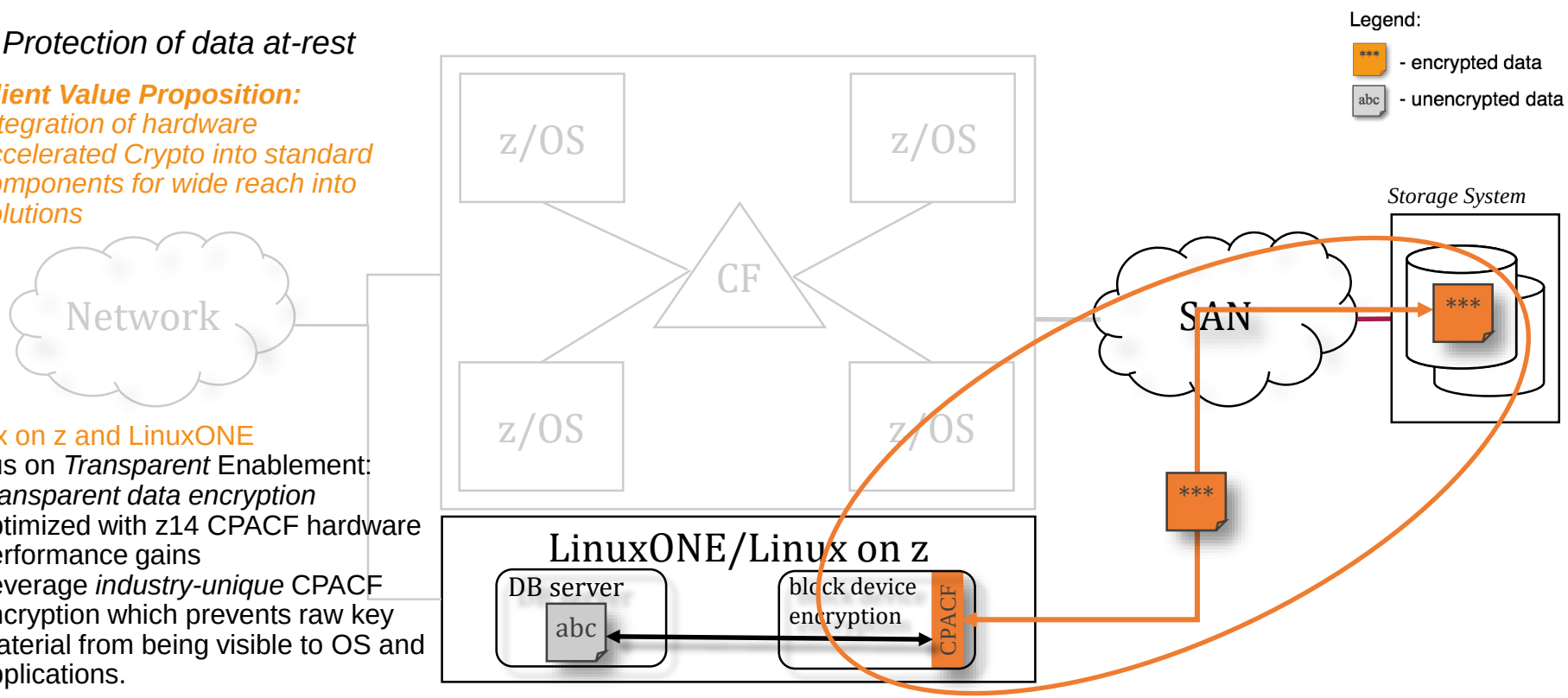
Communication Server
z Encryption Readiness Technology (zERT):

A z/OS administrator can determine when network traffic meets specified policy with new discovery and reporting capabilities

Data Protection // Linux on z Volume Encryption

Protection of data at-rest

Client Value Proposition:
Integration of hardware accelerated Crypto into standard components for wide reach into solutions

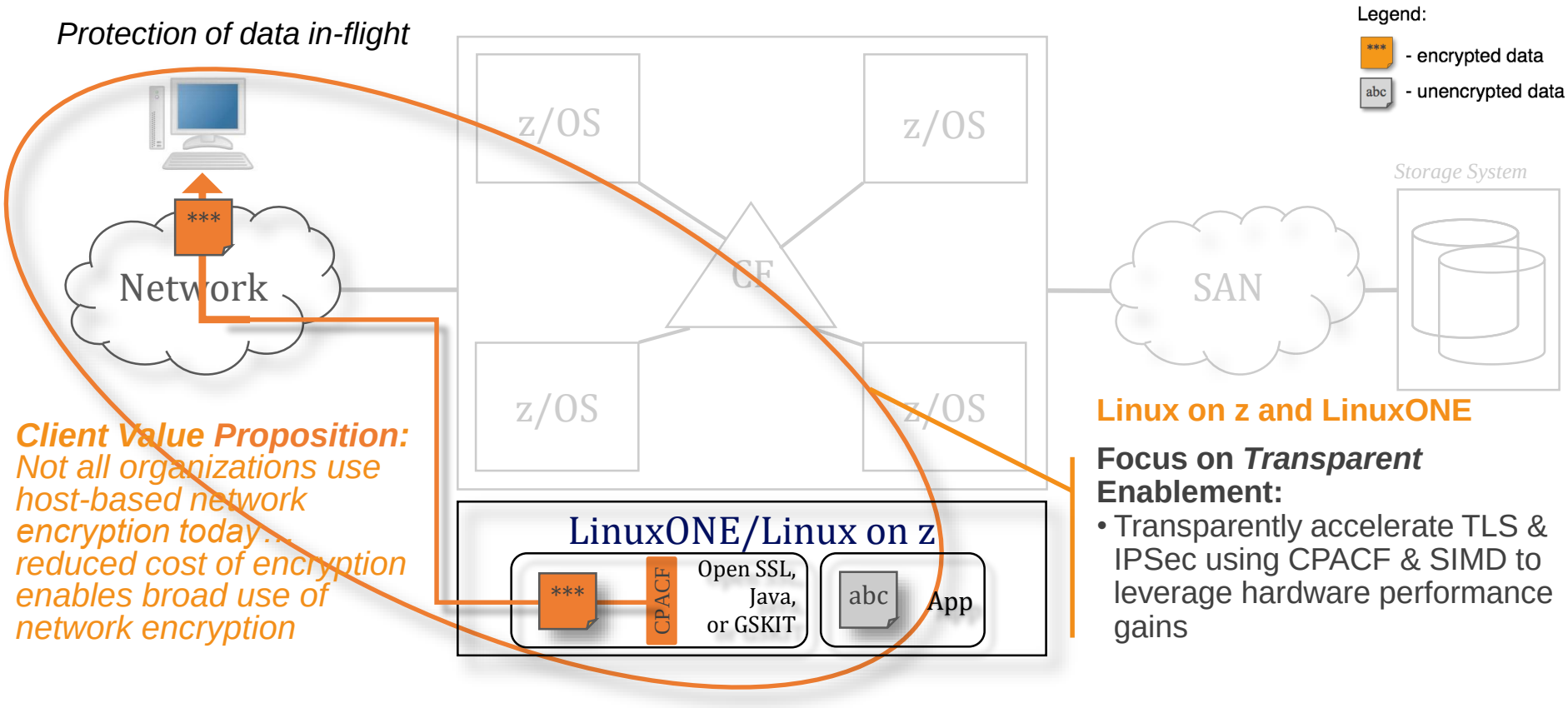


Linux on z and LinuxONE
Focus on *Transparent* Enablement:

- *Transparent data encryption* optimized with z14 CPACF hardware performance gains
- Leverage *industry-unique* CPACF encryption which prevents raw key material from being visible to OS and applications.

Data Protection // Linux on z Network Security

Protection of data in-flight



Client Value Proposition:
Not all organizations use host-based network encryption today… reduced cost of encryption enables broad use of network encryption

Linux on z and LinuxONE

Focus on *Transparent* Enablement:

- Transparently accelerate TLS & IPSec using CPACF & SIMD to leverage hardware performance gains

IBM Hyper Protect virtual servers

A secure virtualization platform that protects your critical Linux® applications throughout the DevSecOps lifecycle on IBM Z® and LinuxONE

Build applications with integrity

Leverage the secure image build process to sign images, validate code, and integrate into your CI/CD pipeline

Deploy workloads with trust

Validate the provenance of your applications before deployment

Manage applications with simplicity

Manage your infrastructure without visibility to sensitive code or data – RESTful API deployment

Encrypt & Sign critical solution components

Give your images access to the industry leading FIPS 140-2 level 4 Hardware Security Module for signing and encryption needs

IBM Hyper Protect Digital Assets Platform

Expanded Elliptic-Curve Cryptography support

Improved security, privacy, and performance for digital assets transactions

IBM Hyper Protect Digital Assets Platform

- Infrastructure foundation for securely hosting and transacting digital assets
- On-Premise
  - IBM Hyper Protect Virtual Servers
- Off-Premise
  - IBM Cloud Hyper Protect Crypto Services
  - IBM Cloud Hyper Protect Virtual Servers

Hierarchical Deterministic Wallets via Bitcoin Improvement Proposal (BIP0032)

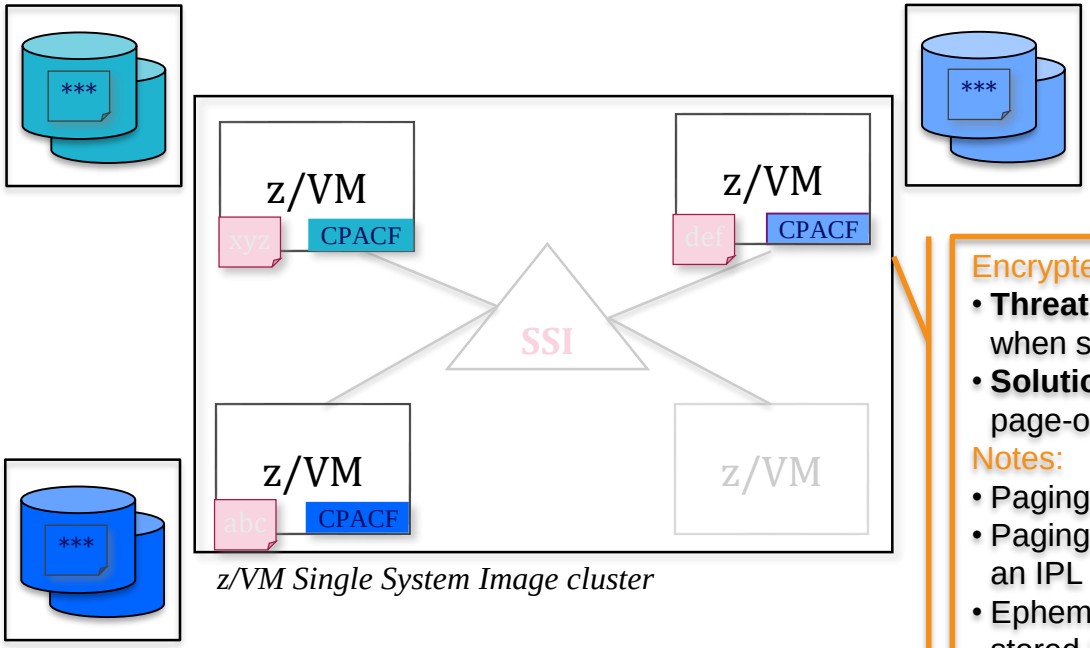
- Enable hierarchical deterministic wallets to execute signing entirely within a Hardware Security Module (HSM) boundary
- Deterministically generate unlimited number of public addresses to receive funds without knowledge of private key

Edwards-Curve Digital Signature Algorithm (EdDSA)

- Smaller keys (32 bytes) and signatures (64 bytes) deliver greater security and speed for signing and verification compared to Elliptical-Curve Digital Signature Algorithm (ECDSA)

Data Protection // z/VM Encrypted Paging

z/VM 6.4



z/VM Single System Image cluster

Client Value Proposition:
Protect guest paging data from administrators and/or users with access to volumes

Encrypted Paging

- **Threat**: access to sensitive data when stored on CP owned disk
- **Solution**: encrypt guest data on page-out.

Notes:

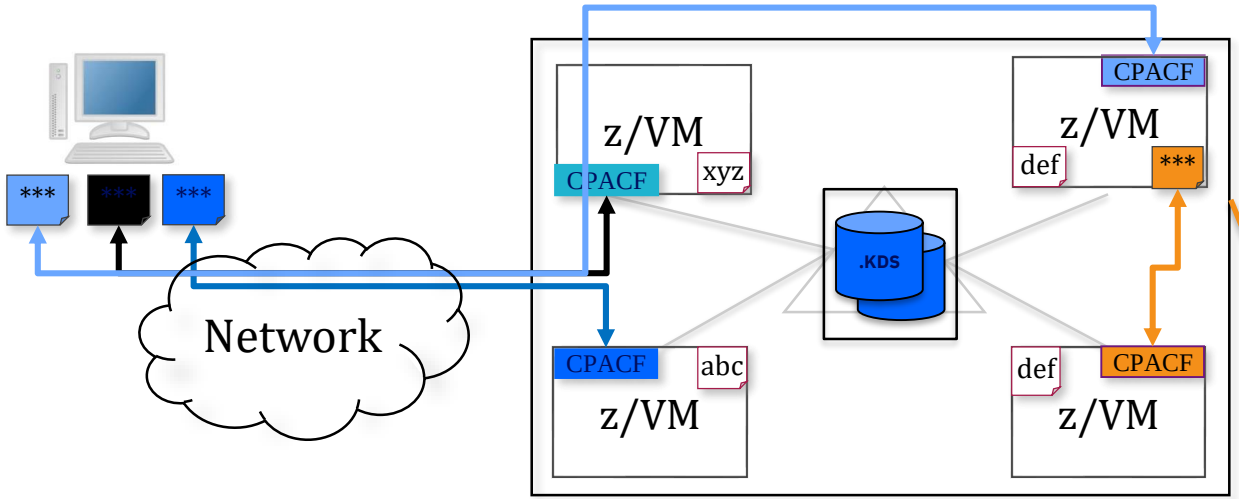
- Paging is not SSI-relevant
- Paging data does not need to survive an IPL
- Ephemeral CPACF protected-key stored in CP (not on disk somewhere)
- AES encryption
- Very low overhead via CPACF

Data Protection // z/VM Network Security

z/VM 6.4

Legend:
*** - encrypted data
abc - unencrypted data

Protection of data in-flight



z/VM Single System Image cluster

Client Value Proposition:
Not all organizations use host-based network encryption today… reduced cost of encryption enables broad use of network encryption

z/VM Secure Communications
- **Threat**: disclosure of sensitive data in flight to the hypervisor layer
- **Solution**: encrypt traffic in flight.

Notes:
- Automatic use of CPACF for symmetric algorithms
- One-line change to enable automatic use of Crypto Express features for acceleration of asymmetric algorithms
- Built on System SSL and ICSFLIB for z/VM

Data Protection // z/TPF Transparent Database Encryption

z/TPF 1.1.0.13

Technical Foundation

z/TPF at-rest Data Encryption

- Automatic encryption of at-rest data
- No application changes required
- Database level encryption using highly efficient CPACF HW crypto
- Includes data on disk and cached in memory
- Optionally can include data integrity checking to detect accidental or malicious data corruption

Client Value Proposition:
Transparent encryption of TPF database data plus reduced cost of encryption allows clients to enable extensive encryption of TPF data.

Additional Information

- Data encrypted using AES CBC (128 or 256)
- Optional integrity checking uses SHA-256
- Includes tools to migrate an existing DB from unencrypted to encrypted state or change the encryption key/algorithm for a given DB while transactions are flowing (no DB downtime)

Support shipped August 2016
(APAR PI56476)

Acknowledgements

Special thanks to the following individuals for lending their expertise and/or providing content:

- Jonathan Bradbury
- William Santiago-Fernandez
- Garry Sullivan
- Dave Evans
- Brian Hugenbruch
- Reinhard Buendgen
- Henrik Lyksborg
- Cecilia Carranza Lewis
- Mark Brooks
- Neil Johnson
- Chris Meyer
- Diana Henderson
- Michael Jordan
- Mark Gambino
- Isabel Arnold
- Harald Freudenberger

Contact Information

Roan Dawkins

System z Cryptography, IBM

dawkinsr@us.ibm.com

Eysha S. Powers

Enterprise Cryptography, IBM

eysha@us.ibm.com