z14 Crypto Update

Eysha S. Powers
Enterprise Cryptography

Today’s Security Drivers

ADVANCED ATTACKS

HUMAN ERROR

INNOVATION

COMPLIANCE

SKILLS GAP

Why Encrypt? To Protect Sensitive Data

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 standard bodies and legislation.

Why Encrypt? To Comply with 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
  General Data Protection Regulation (GDPR),
- in the HiPAA,
- and in local legislation.

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

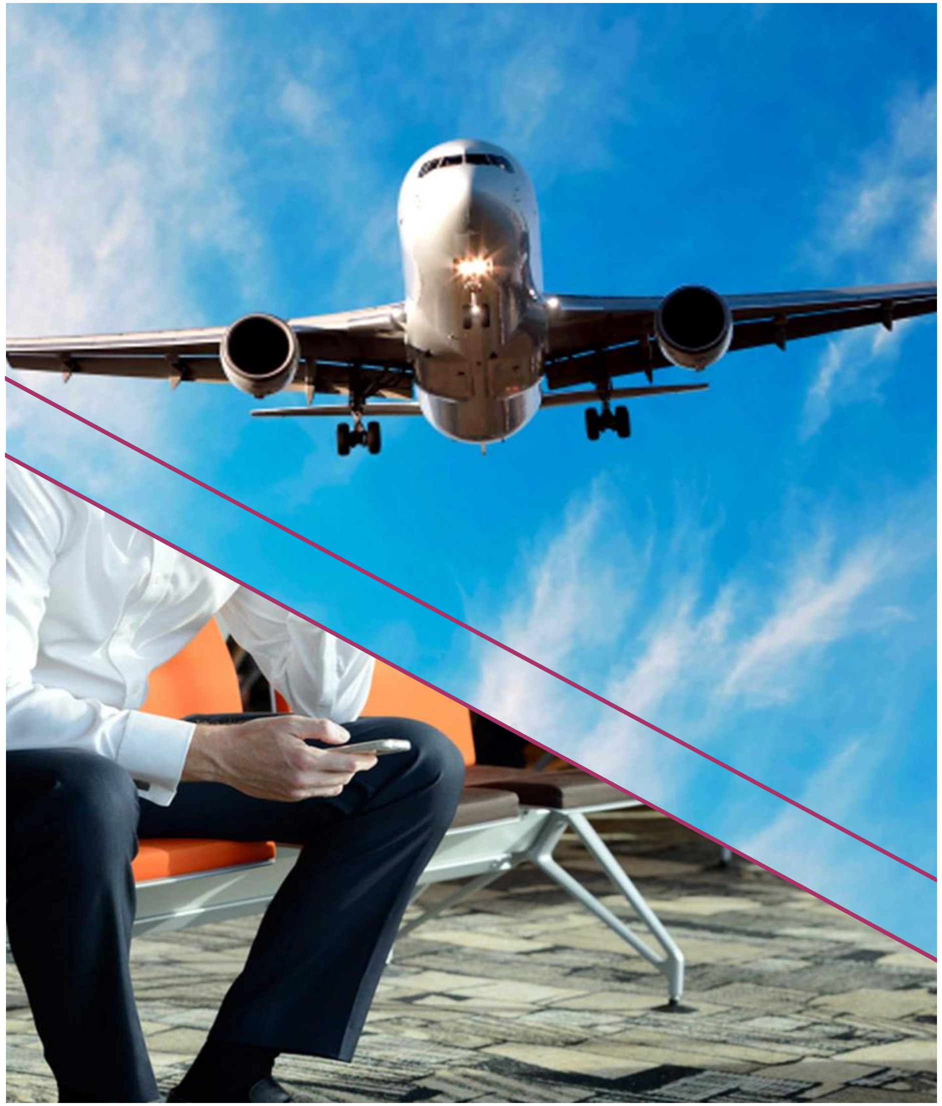
PCi DSS COMPLIANT

HIPAA

GDPR

When Should We Encrypt?

- **Data in flight**
  - Virtual Private Networks (VPNs)
  - SSL/TLS connections (using public/private keys and certificates and symmetric encryption)
- **Data at rest**
  - File and folder encryption – including the use of intermediate devices
  - Removable media (tape) encryption
- **Transactional environments**
  - Industry specific – finance (e.g. EMV smart cards)
  - 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…



Extensive use of encryption is one of the most impactful ways to help reduce the risks and financial losses of a data breach and help meet complex compliance mandates.

Table of Contents

- **IBM Z Crypto History**
- **IBM Z Crypto Hardware**
  - Crypto Express 6S (CEX6S)
  - CP Assist for Cryptographic Function (CPACF)
  - IBM Trusted Key Entry (TKE) Workstation

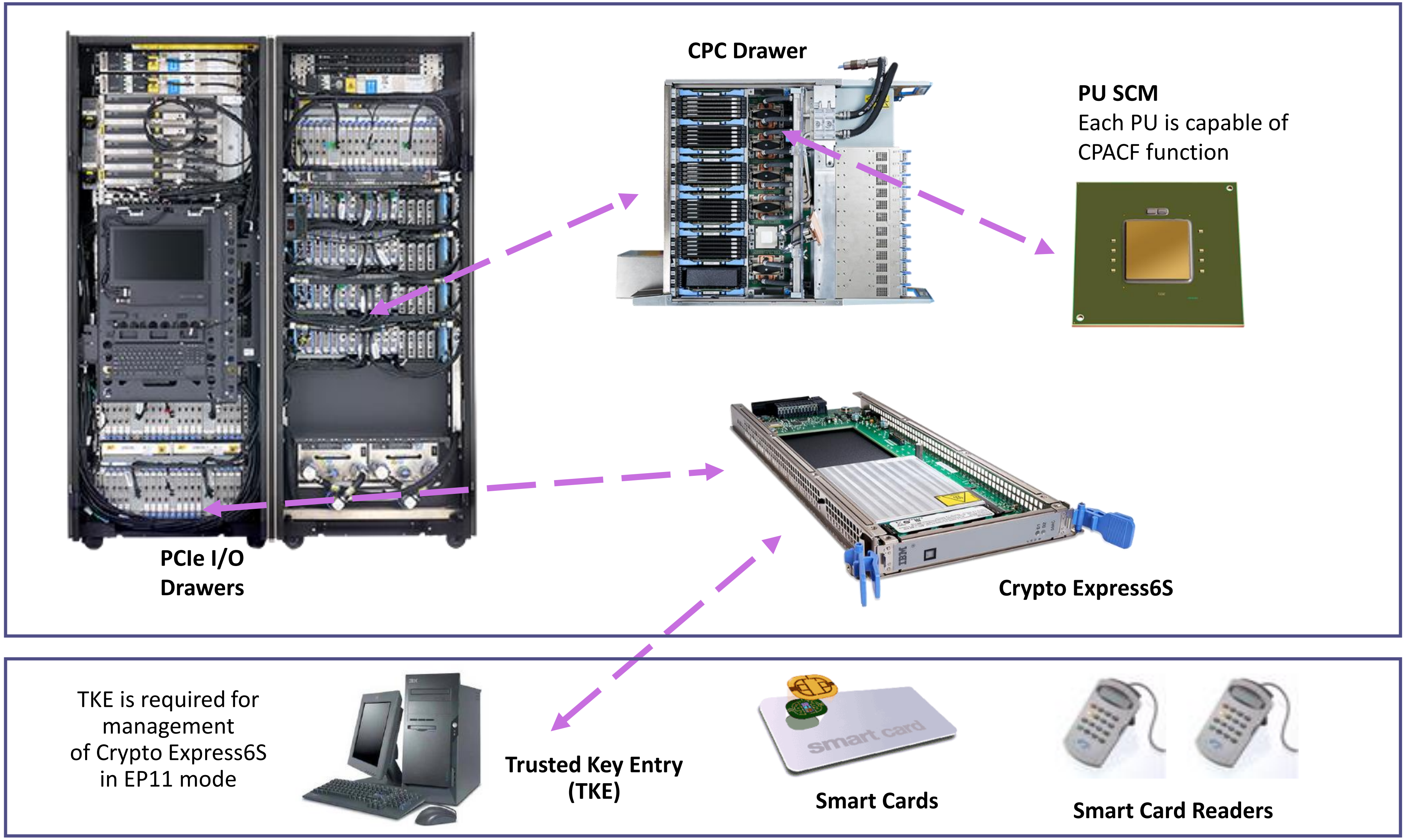
- **IBM Z Crypto Stack**
  - z/VM Crypto
  - z/VSE Crypto
  - Linux Crypto Stack
  - z/OS Crypto Stack
- **IBM Z Crypto Software**
  - z/OS Integrated Cryptographic Services Facility

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
- **IBM z14 with Pervasive Encryption: 2017**
- **Crypto Express6S Crypto Coprocessor: 2017**

IBM Z Crypto Hardware



CPC Drawer

PU SCM
Each PU is capable of
CPACF function

PCIe I/O Drawers

Crypto Express6S

TKE is required for management of Crypto Express6S in EP11 mode

Trusted Key Entry (TKE)

Smart Cards

Smart Card Readers

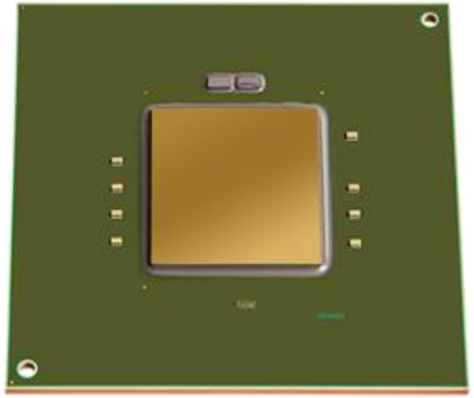
CP Assist for Cryptographic Function (CPACF)

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)
  - Support for AES
- Data integrity
  - MD5, SHA-1
  - SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)
  - New → SHA-3 (SHA3-224, SHA3-256, SHA3-384, SHA3-384, SHA3-512)
  - SHAKE (SHAKE-128, SHAKE-256)
- Random Number generation (PRNG, DRNG)
- Message Authentication



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

- SSL
- VPN
- Data storing applications (e.g. DB2, IMS)
- New → Data sets and files
- Coupling Facility

z14 CPACF Performance Enhancements

AES-GCM and AES-XTS encryption modes have encryption rates four to six times faster than z13

- **Galois Counter Mode (GCM)** is a stream cipher used primarily in SSL/TLS workloads.
  - Exploited by z/OS Comm Server and System SSL
    Note: It may be necessary to configure cipher suites to prefer GCM mode.
- **XEX-based tweaked-codebook mode with ciphertext stealing (XTS)** is a block cipher mode typically used for disk encryption.
  - Exploited by DFSMS z/OS data set encryption
- **Cipher Block Chaining (CBC)** is a block cipher used in several crypto applications. AES-CBC decryption on z14 offers a 4x increase in throughput for decryption operations.

Exploiters of the CPACF benefit from the throughput improvements of z14's CPACF such as:

- **DFSMS z/OS data set encryption** } New
- **Coupling Facility (CF) encryption** } New
- 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 Express Adapters

- Provide state-of–the art **tamper sensing and responding**, programmable hardware to **protect cryptographic keys**, sensitive cryptographic processing and sensitive custom applications
  - Unauthorized removal 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
- Support multiple **logically-separate cryptographic domains** for use by different LPARS.
- Provide 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 Express6S

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), 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, Clear Key Fast Path (Symmetric and Asymmetric)

Crypto Express6S - Modes

Crypto Express adapters can be configured in three different modes

Accelerator Mode:

- Request is processed fully in hardware (versus Power PC)
- Supports clear key RSA operations (e.g. SSL Acceleration)

CCA Coprocessor Mode:

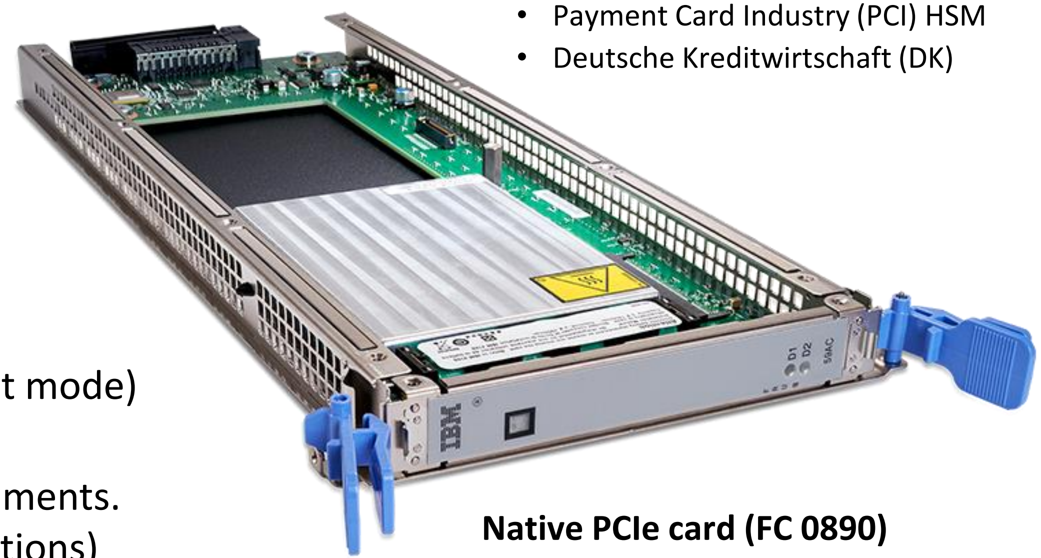
- Supports the IBM Common Cryptographic Architecture (CCA)
- New → **With CEX6S, supports domain-segregated PCI-HSM Compliant mode**
- Request is sent first to the internal IBM PowerPC for processing (default mode)

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

Designed to Meet Physical Security Standards

- FIPS 140-2 level 4
- ANSI 9.97
- Payment Card Industry (PCI) HSM
- Deutsche Kreditwirtschaft (DK)

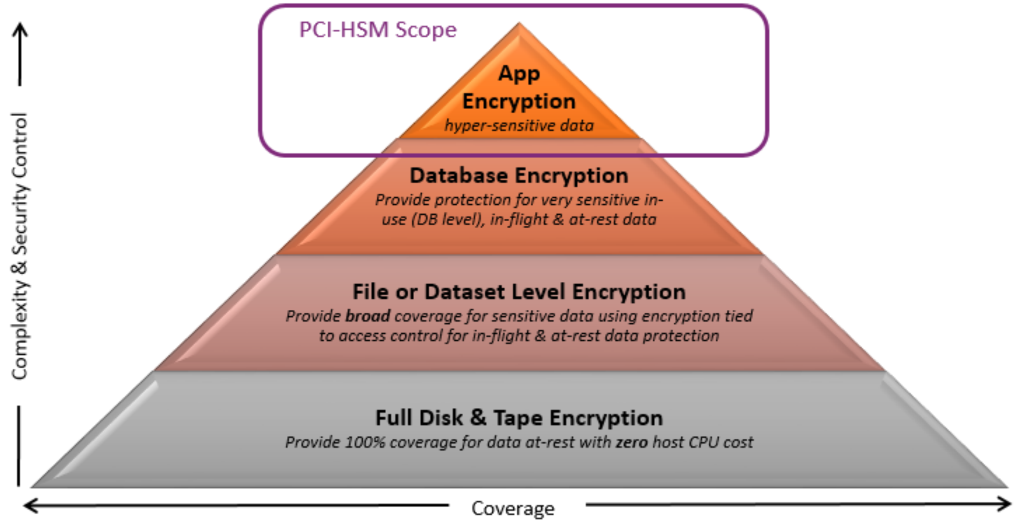


Native PCIe card (FC 0890)

- Resides in the PCIe I/O drawer
- Requires CPACF Enablement (FC 3863)
- Up to 16 features per server

What is PCI-HSM Compliance?

- The PCI-HSM (PCI Hardware Security Module) was developed to improve security in payment card systems.
- It imposes requirements in key management, HSM API functions, device physical security, controls during manufacturing and delivery, device administration, and a number of other areas. It prohibits many things that were in common use for many years, but are no longer considered secure.
- The result of these requirements is that applications and procedures often have to be updated, because they used some of the things that are *now prohibited*. While this is inconvenient and imposes some costs, it increases the resistance of the systems to attacks of various kinds.


PCI-HSM Scope
App Encryption
hyper-sensitive data
Database Encryption
Provide protection for very sensitive in-use (DB level), in-flight & at-rest data
File or Dataset Level Encryption
Provide broad coverage for sensitive data using encryption tied to access control for in-flight & at-rest data protection
Full Disk & Tape Encryption
Provide 100% coverage for data at-rest with zero host CPU cost
Complexity & Security Control
Coverage

Updating a system to use PCI-HSM compliant HSMs reduces the risk of loss for both the institution and its customers.

PCI-HSM and Crypto Express 6S Adapters

In response to the PCI-HSM standards, and adoption trends in the industry, CCA version 6.0 was designed to meet the "Payment Card Industry (PCI) PIN Transaction Security (PTS) Hardware Security Module (HSM) Version 3.0, June 2016" standard.

The features and enhancements in CCA 6.0 provide:

- The ability to simultaneously support PCI-HSM compliant applications and non-compliant applications
- Features to help you determine what parts of your current system need to be changed to be compliant
- Mandatory dual control for sensitive operations
- Separate logical key spaces to support both compliant and non-compliant workloads
- Secure auditing of sensitive operations
- Key usage restrictions for keys used in PCI-HSM compliant applications
- Cryptographically protected information about firmware versions in the HSM, which can be viewed from a remote administration workstation

IBM has provided these features and this environment to support your needs when dealing with applications subject to PCI standards. For more information on PCI Security Council and PCI standards, see https://www.pcisecuritystandards.org.

Hardware Requirements for PCI-HSM

The **Crypto Express6S**, running in Common Cryptographic Architecture (CCA) mode with **CCA version 6.0** is designed to comply with the Payment Card Industry (PCI) PIN Transaction Security (PTS) Hardware Security Module (HSM) Version 3.0, June 2016, standard.

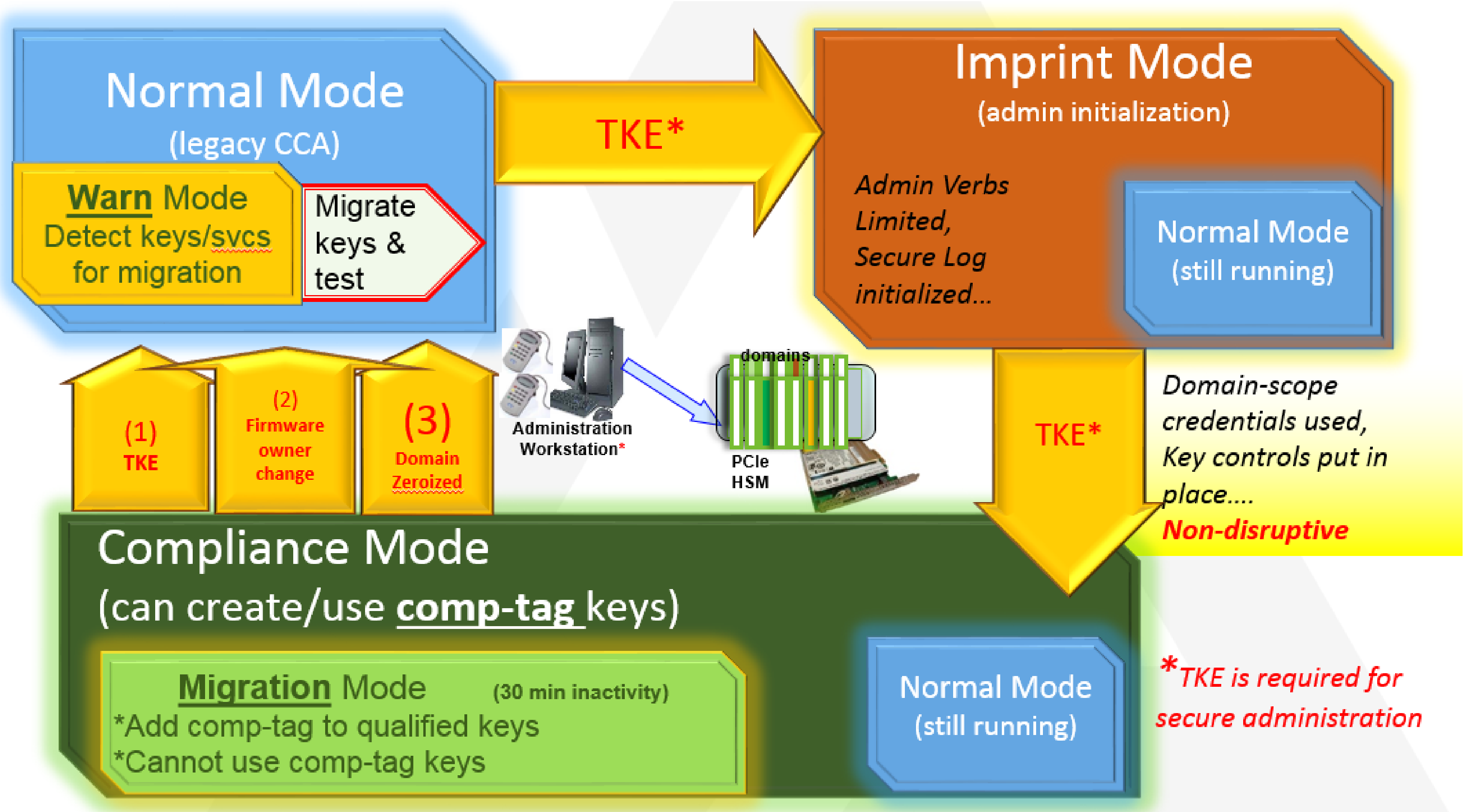
- Each domain on a module can be placed into PCI compliant mode. You can have some domains in PCI compliant mode while others are not.
- PCI compliant domain administration must be done:
  - From a TKE at the minimum level of ***TKE 9.0***
  - Using domain-specific administrators and mandatory dual controls

Crypto Express6S Adapter with CCA 6.0

IMPORTANT: If a domain is running in PCI compliant mode, a new type of PCI-compliant tagged key may be created. Compliant-tagged keys have strength and service restrictions. Only internal, fixed-length DES key tokens (double-length key size) can be compliant-tagged. Expect client application impacts when converting to compliant-tagged keys. Refer to ICSF and CCA documentation for more details.

Trusted Key Entry (TKE) Workstation 9.0

PCI-HSM Domain Mode Transitions



Normal Mode
(legacy CCA)

Warn Mode
Detect keys/svcs
for migration

Migrate
keys &
test

TKE*

Imprint Mode
(admin initialization)

*Admin Verbs
Limited,
Secure Log
initialized…*

Normal Mode
(still running)

(1)
TKE

(2)
Firmware
owner
change

(3)
Domain
Zeroized

Administration
Workstation*

domains

PCIe
HSM

TKE*

*Domain-scope
credentials used,
Key controls put in
place….*
Non-disruptive

Compliance Mode
(can create/use **comp-tag** keys)

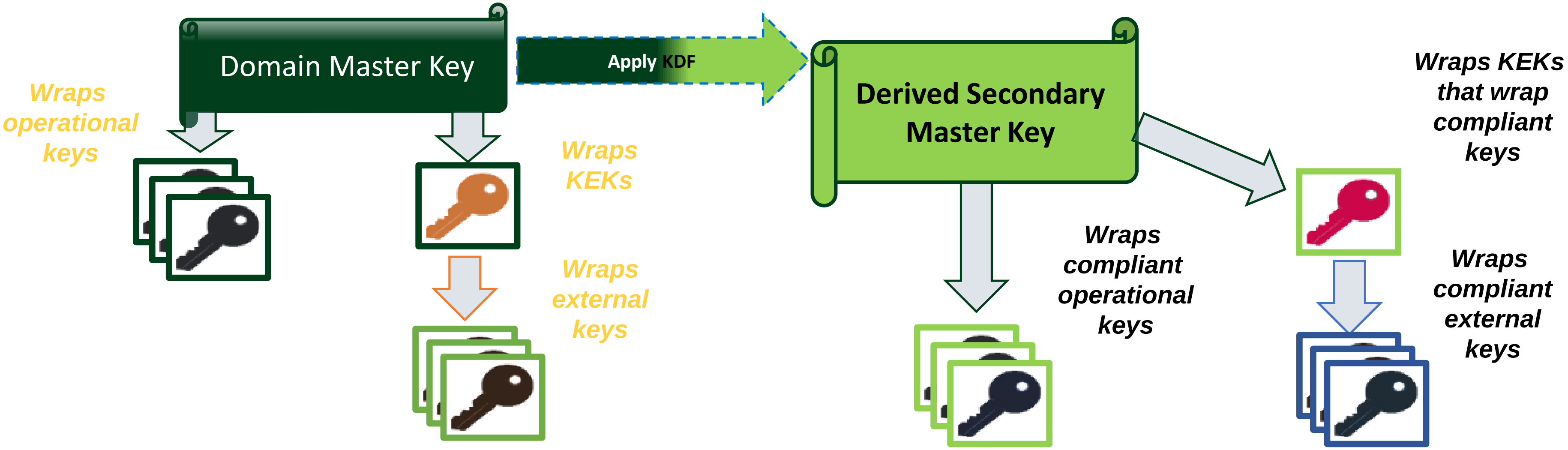
Migration Mode (30 min inactivity)
*Add comp-tag to qualified keys
*Cannot use comp-tag keys

Normal Mode
(still running)

**TKE is required for
secure administration*

PCI-HSM Master Key

The PCI-HSM (Secondary) Master Key is derived from the Domain Master Key. The Master Key Management process is unchanged.


Domain Master Key
Apply KDF
Derived Secondary Master Key
Wraps operational keys
Wraps KEKs
Wraps external keys
Wraps compliant operational keys
Wraps KEKs that wrap compliant keys
Wraps compliant external keys

CCA Architecture vs PKCS #11 Architecture

IBM Common Cryptographic Architecture (CCA)

- IBM proprietary cryptographic application programmers interface (API) providing a broad range of cryptographic services including
  - standard cryptographic algorithms
  - financial services standards

CCA Functions & Algorithms

- Encrypt / Decrypt (AES, DES, DES3, RSA)
- Sign / Verify (RSA, ECC)
- HMAC Generate / Verify (HMAC)
- Key Generate (AES, DES, DES3, HMAC)
- Key Pair Generate (RSA, ECC)
- Diversified Key Generate
- Derive Unique Key Per Transaction (DUKPT)
- … And More!

Designed for financial services standards and **PCI-HSM certification**

PKCS #11 Cryptographic Architecture

- Originally published by RSA Laboratories, now maintained by OASIS
- Defines a standard API for devices that hold cryptographic information and perform cryptographic functions
- Enterprise PKCS#11 – EP11

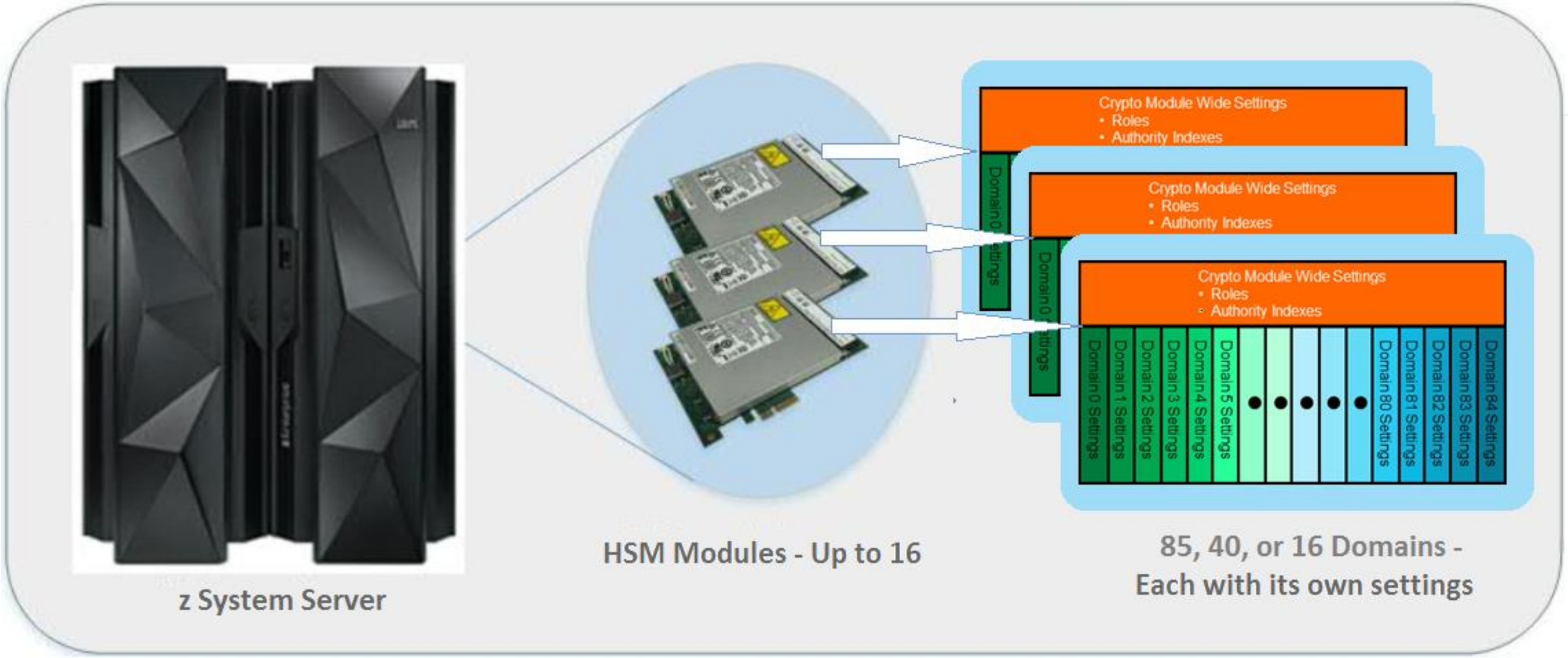
PKCS#11 Functions & Algorithms

- 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)
- Domain Parameter Generation (DH)
- … And More!

Designed for portability and FIPS/Common Criteria certification

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.



z System Server

HSM Modules - Up to 16

85, 40, or 16 Domains - Each with its own settings

- 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 Scoped and Domain Scoped 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
- Migration Wizards
- Enable/disable Access Control Points (ACPs)
- Loading MKs for inactive LPARs
- Loading PIN decimalization tables
- Loading EP11 Master Key

TKE Workstation 9.0

Base Hardware

- TKE 9.0 Workstation with a 4768 Cryptographic Adapter (required to manage Crypto Express 6S on z14)
  - TKE 9.0 Tower Workstation (FC 0086)
  - TKE 9.0 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

- TKE 8.0 & 7.3 (FC 0842 only) workstations can be upgraded to TKE 9.0 with purchase of 4768
- Omnikey Cardman 3821 smart card readers can be carried forward to any TKE 9.0 workstation
- Previously initialized and personalized smart cards can be carried forward and used on any TKE 9.0 workstation.

Feature Overview

Usability

- Copy smart card content to new zone
- Save/Restore data using TKE directory structure
- Ability to generate key parts without opening a host
- Reduced number of commands for domain group load, set, and clear (CCA 5.3 or later)
- New IBM-supplied role for TKE workstation profile group members

Workstation Integrity

- Can configure to force sign-on after boot
- Can use Multi Factor Authentication for workstation sign on

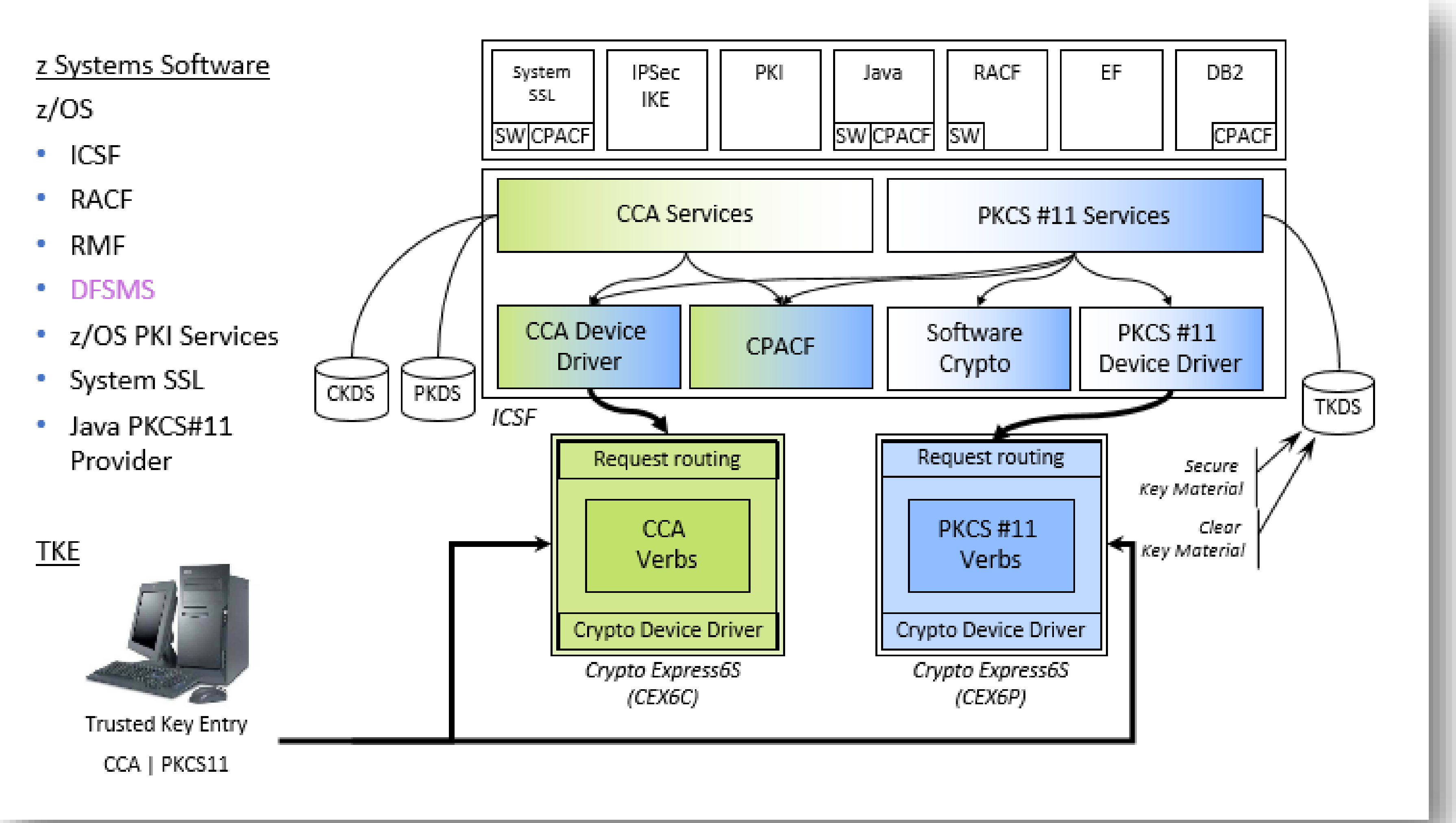
Audit

- TKE Audit Log browse application
- TKE heartbeat audit record for periodic status monitoring

PCI HSM

- TKE required to manage PCI-HSM compliant domains on Crypto Express6S

IBM Z Crypto Stack – z/OS



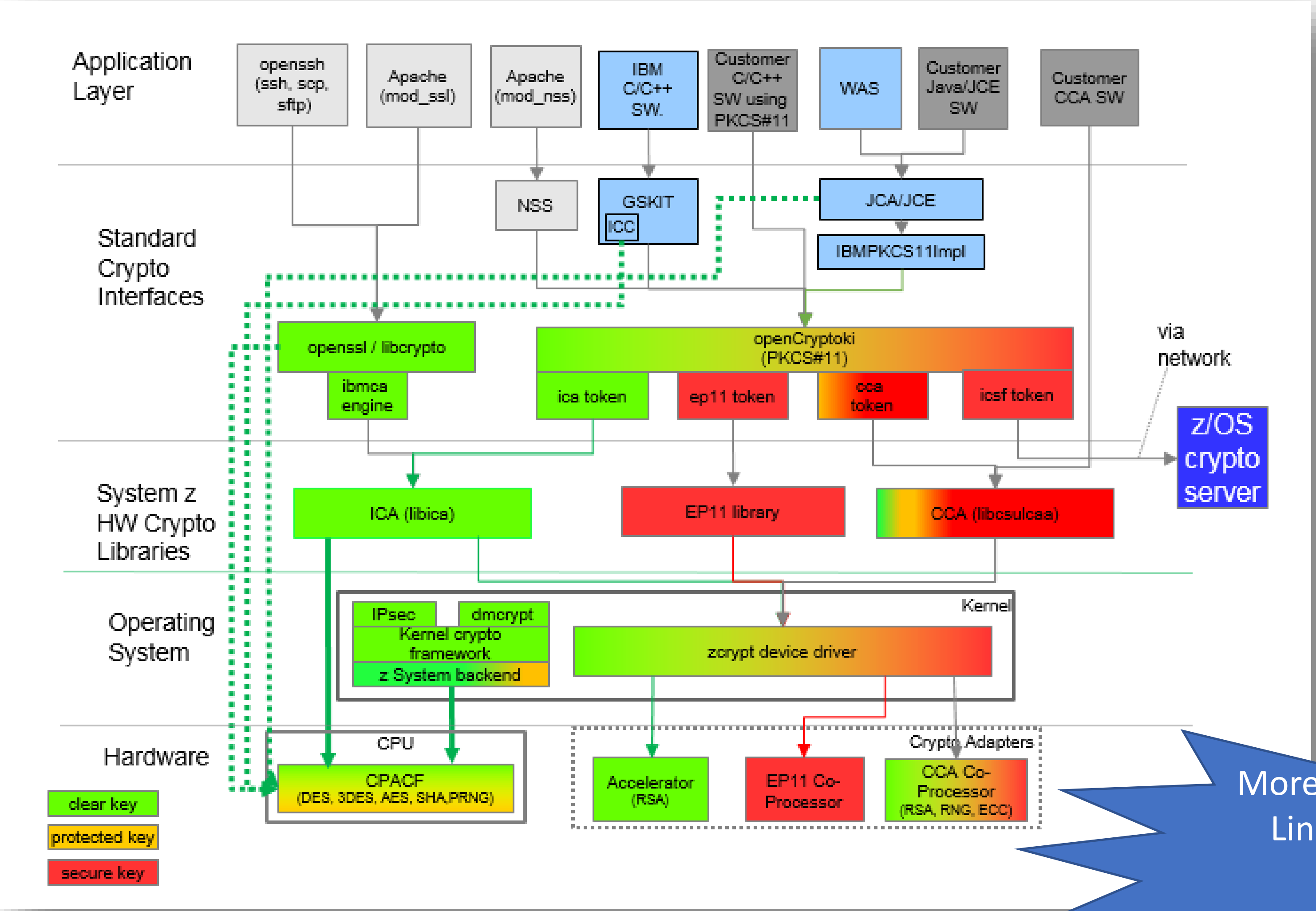
z Systems Software

z/OS

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

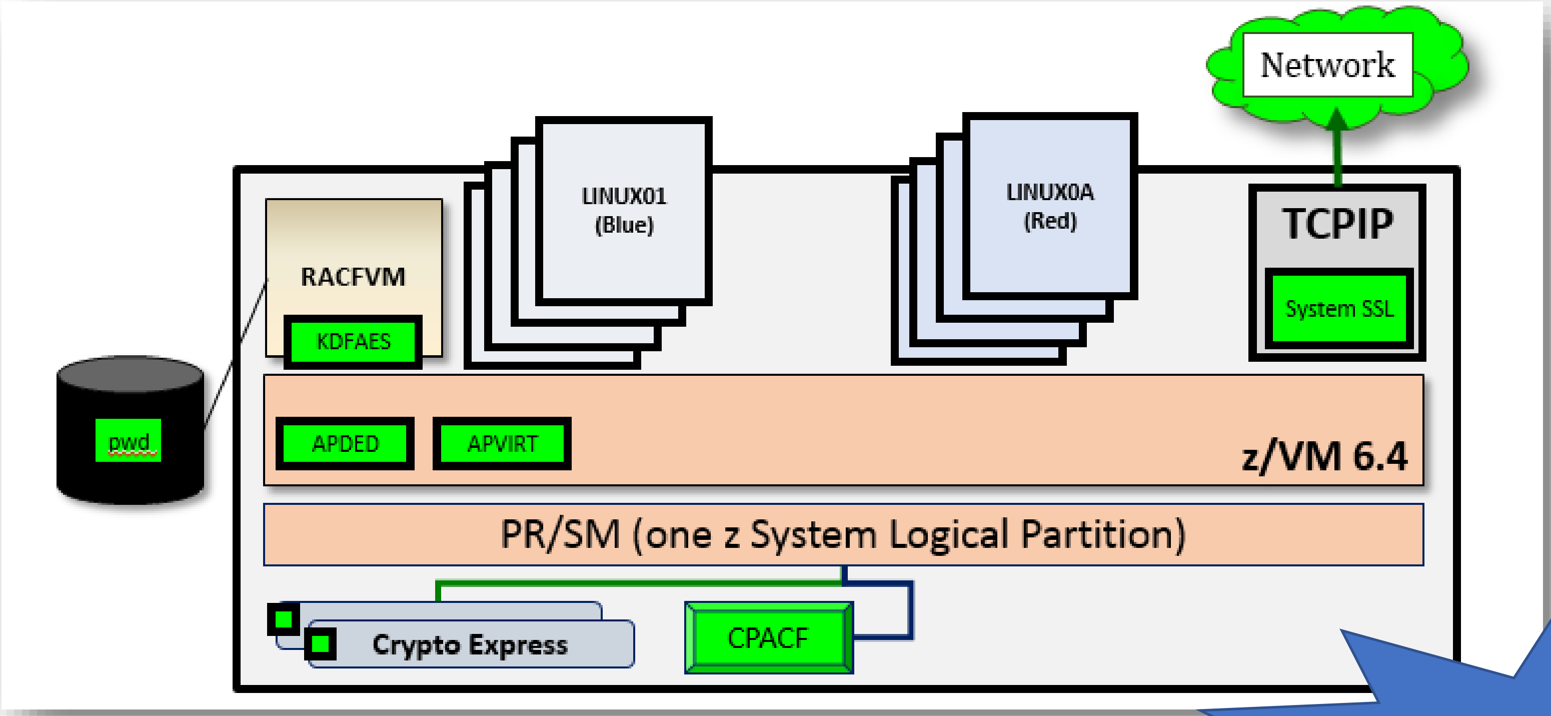
TKE

IBM Z Crypto Stack – Linux on z



More details in the Linux on IBM z Session

IBM Z Crypto Stack – z/VM



More details in the z/VM 6.4 Session

IBM Z Crypto Stack – 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.

z/OS Integrated Cryptographic Services Facility (ICSF)

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 functions.

ICSF uses cryptographic keys to:

- Protect data
- Protect and distribute additional keys
- Verify message integrity
- Generate, protect and verify PINs
- Generate and verify signatures

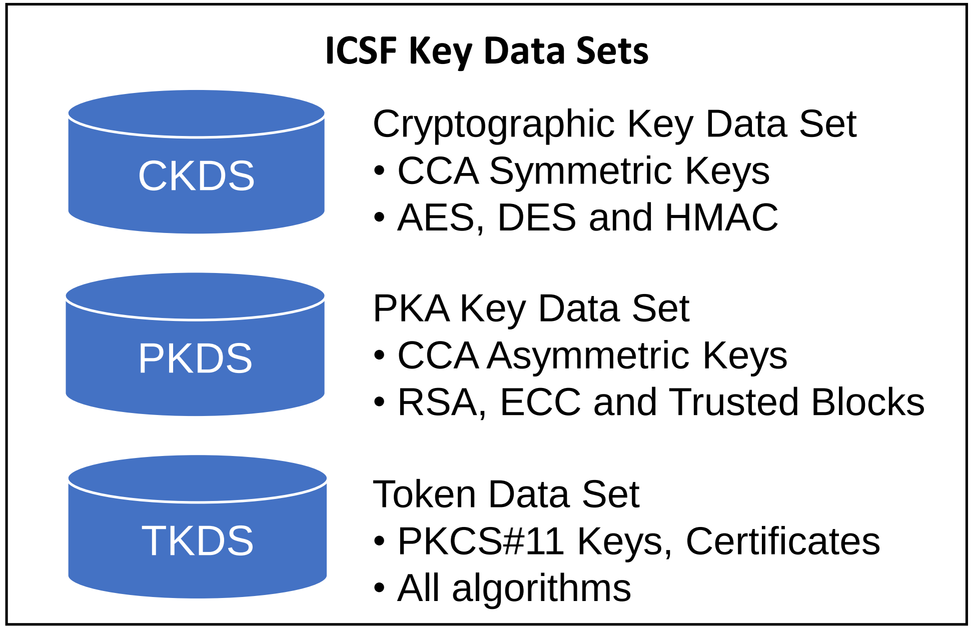
ICSF is the default means to load master key values onto secure cryptographic features, allowing the hardware features to be used by applications.

z/OS ICSF Key Data Sets

- ICSF provides callable services and utilities to generate and store operational keys into ICSF Key Data Sets (KDS)
- 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 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



ICSF Key Data Sets

CKDS

Cryptographic Key Data Set
- CCA Symmetric Keys
- AES, DES and HMAC

PKDS

PKA Key Data Set
- CCA Asymmetric Keys
- RSA, ECC and Trusted Blocks

TKDS

Token Data Set
- PKCS#11 Keys, Certificates
- All algorithms

z/OS ICSF – Protecting 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 ar 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 (TKDS).
- The XCSFKEY class controls the ability to export a symmetric key with the Symmetric Key I 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 IC: "SETICSF".

Key Store Policy

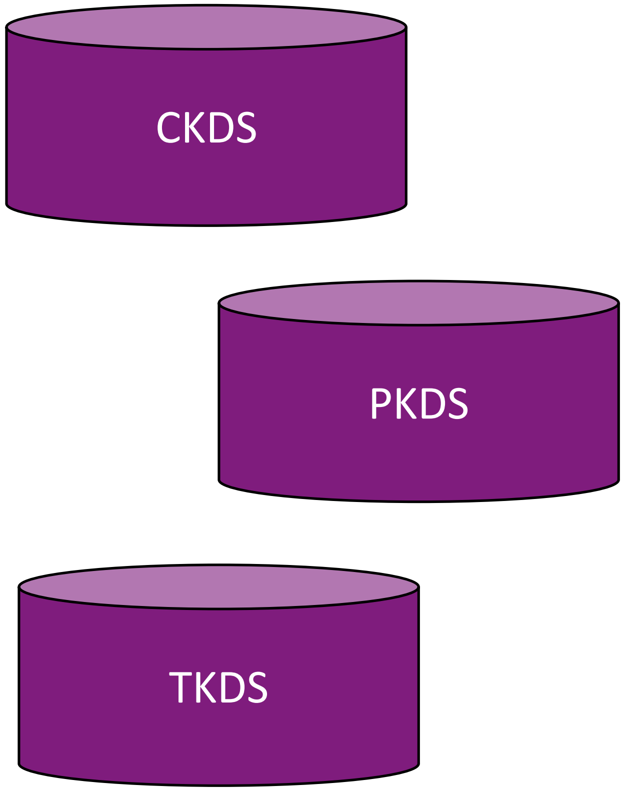
- Define additional security policies pertaining to the use of key tokens.

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

z/OS ICSF Releases

ICSF FMID	Web Deliverable	z/OS Base Release	z Hardware Release	GA Date
HCR77A1	WD #13	---	---	Sep 2013
HCR77B0	WD #14	z/OS V2R2	z13	Feb 2015
HCR77B1	WD #15	---	---	Nov 2015
HCR77B1 + OA49064 SPE	---	---	z13s	Mar 2016
HCR77C0	WD #16	z/OS V2R3	---	Oct 2016
HCR77C1	WD #17	---	z14	Sep 13, 2017

z/OS ICSF HCR77A1 – Common Record (KDSR) Format



CKDS

PKDS

TKDS

Offset	Number of Bytes	Field Name
0	72	Key label or handle
72	8	Reserved
80	1	Version
81	1	KDS type (CKDS, PKDS, TKDS)
82	2	Flags
84	4	Record length
88	8	Creation date
96	8	Creation time
104	8	Last update date
112	8	Last update time
120	4	Key material length
124	4	Key material offset
128	4	Metadata length
132	4	Metadata offset
136	4	Reserved

z/OS ICSF HCR77A1 – Common Record (KDSR) Format

Introduced in ICSF HCR77A1

- Supports the CKDS, PKDS and TKDS
- Enables variable-length metadata for KDS records
- Track key reference dates

Enhanced in ICSF HCR77B0

- Support key archival and recall
- New callable services to read and write metadata to KDS records
- New callable service to list KDS records

Using the Common Record Format

Allocate new Key Data Sets

- CKDS: SYS1.SAMPLIB(CSFCKDS3)
- PKDS: No change to allocation process
- TKDS: SYS1.SAMPLIB(CSFTKD2)

If there are no existing keys to convert then

- Initialize new Key Data Sets using the ICSF panels (all KDS types) or JCL job (TKDS)

If there are existing keys to convert to the new format

- Run the KDS Conversion utility from the ICSF KDS Management panels (for each KDS type to be converted)

z/OS ICSF HCR77B0 – Key Archiving

ICSF can archive records in key data sets (using KDSR common record format) . The record remains in the data set, but the key material in the record cannot be used. Any attempt to use the key material will fail unless the optional key archive use control is enabled which will allow the request to complete. An SMF record is logged in both cases.

1. Determine if archived keys should be allowed for cryptographic operations
   - Define the CSF.KDS.KEY.ARCHIVE.USE resource in the XFACILIT class to allow service requests using archived keys to succeed.
   - Specify KEYARCHMSG(YES|NO) in the ICSF installations options data set for ICSF to issue a joblog message on the first occurrence of an archived key successfully being used
2. Mark the archive flag using the Metadata Write service (CSFKDMW)
   - ICSF does not delete the key from the KDS
   - ICSF generates an SMF type 82 audit record
3. An application attempts to use an archived key
   - ICSF writes an SMF type 82 record indicating key use
   - ICSF allows the request to succeed or fail based on the CSF.KDS.KEY.ARCHIVE.USE resource
   - ICSF writes a joblog message based on the KEYARCHMSG option

z/OS ICSF HCR77B0 – Key Validity Dates

ICSF supports the ability to specify a period when the key material of a key data set record (in KDSR common record format) is active. The ICSF administrator can specify the start and end dates when the key material is active and ICSF will allow only the key material to be used by applications within those dates.

1. Set the start and end validity dates using the Metadata Write service (CSFKDMW)
   - The date cannot be set to a date in the past
2. An application attempts to use an inactive key
   - ICSF writes an SMF type 82 record indicating key use
   - ICSF fails the request

Note: Key material validity dates are checked before the record archived flag.

z/OS ICSF HCR77B1 - MVS Console Commands

ICSF HCR77B1 introduced new MVS Console Commands. **Note:** HCR77B1 must be running on z/OS 2.1 or later.

DISPLAY ICSF *command*

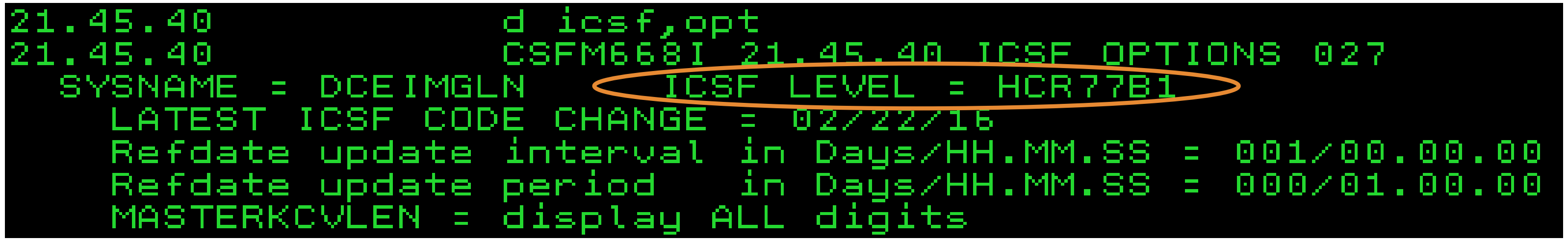
- Display the status of cryptographic devices
- Display certain ICSF options
- Display information about the active cryptographic key data sets (KDS)
- Display the status of master key registers
- List the systems available to participate in sysplex operations

SETICSF *command*

- Activate, deactivate or restart a cryptographic device
- Enable or disable updates to a KDS
- Change a subset of ICSF installation options

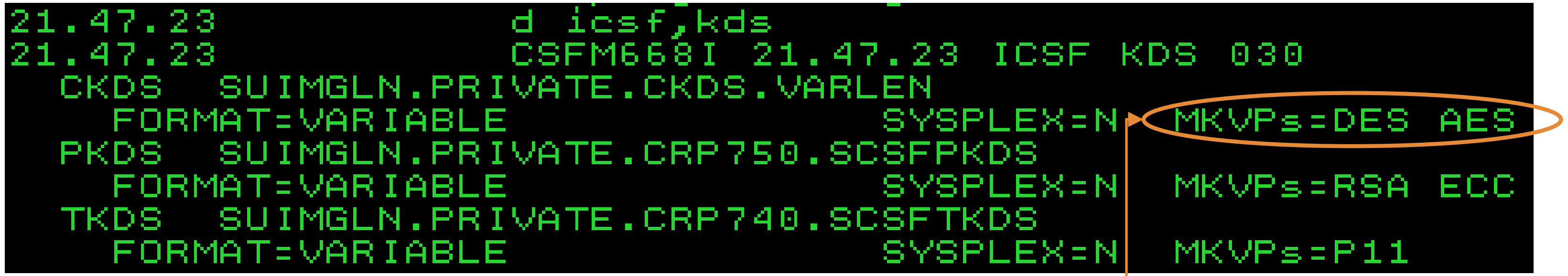
z/OS ICSF HCR77B1 - MVS Console Command Examples

D ICSF,OPT



```
21.45.40           d icsf,opt
21.45.40           CSFM668I 21.45.40 ICSF OPTIONS 027
  SYSNAME = DCEIMGLN     ICSF LEVEL = HCR77B1
    LATEST ICSF CODE CHANGE = 02/22/16
    Refdate update interval in Days/HH.MM.SS = 001/00.00.00
    Refdate update period   in Days/HH.MM.SS = 000/01.00.00
    MASTERKCVLEN = display ALL digits
```

D ICSF,KDS



```
21.47.23           d icsf,kds
21.47.23           CSFM668I 21.47.23 ICSF KDS 030
  CKDS  SUIMGLN.PRIVATE.CKDS.VARLEN
    FORMAT=VARIABLE               SYSPLEX=N  MKVPs=DES AES
  PKDS  SUIMGLN.PRIVATE.CRP750.SCSFPKDS
    FORMAT=VARIABLE               SYSPLEX=N  MKVPs=RSA ECC
  TKDS  SUIMGLN.PRIVATE.CRP740.SCSFTKDS
    FORMAT=VARIABLE               SYSPLEX=N  MKVPs=P11
```

MKVPs

z/OS ICSF HCR77C0 – Dynamic CSFPRMxx Refresh

ICSF HCR77C0 supports dynamic refresh of the Installation Options Data Set using the SETICSF command or CSFMPS callable service.

```
SETICSF OPTions,REFRESH,SYSPLEX(YES|NO)
```

The following installation option parameters are supported for dynamic refresh.

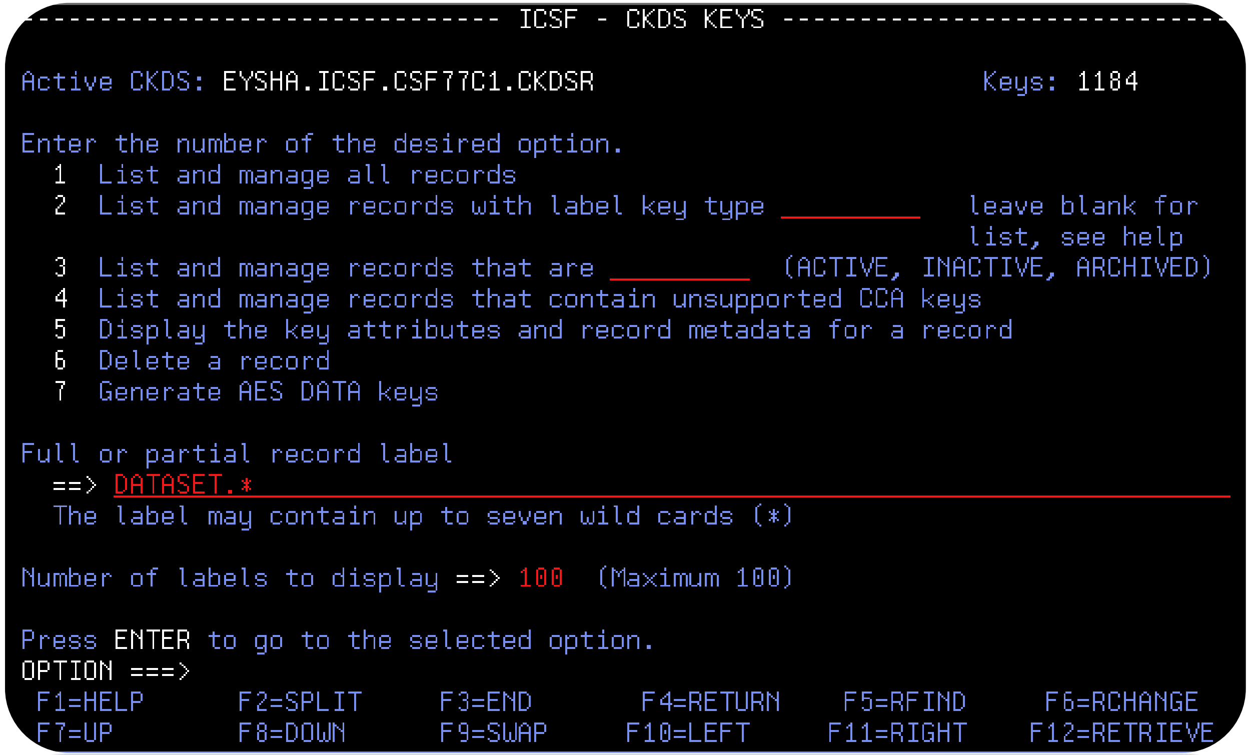
- BEGIN
- CHECKAUTH
- DEFAULTWRAP
- END
- FIPSMODE
- KEYARCHMSG
- KDSREFDAYS
- MASTERKCVLEN
- MAXSESSOBJECTS
- RNGCACHE
- SSM
- USERPARM
- WAITLIST
- AUDITKEYLIFECKDS
- AUDITKEYLIFEPKDS
- AUDITKEYLIFETKDS
- AUDITKEYUSGCKDS
- AUDITKEYUSGPKDS
- AUDITPKCS11USG

z/OS ICSF HCR77C1 – Regional Crypto Servers – Intl Algorithms

- Regional cryptographic servers are network-attached, standalone devices or dedicated Linux LPARs that perform geography-specific cryptography.
- These servers are secure key hardware security modules (HSMs) that operate similar to IBM's PKCS #11 secure coprocessors (CEXnP).
  - They are marketed and serviced by IBM vetted third party vendors.
- Secure keys are stored in the TKDS, protected by the Regional Cryptography Server Master Key (RCS-MK).

ICSF Release	Crypto Algorithms Supported	Crypto Operations Supported
HCR77B1	• Chinese SMx family of algorithms (SM4)	• Key Generation, Data Encryption, Data Decryption, Key Wrapping
HCR77B1 PTF OA49069	• Chinese SMx family of algorithms (SM4, SM2, SM3)	• HCR77B1 base plus Key Pair Generation, Hashing, Digital Signature Generation, Digital Signature Verification, Change Master Key
HCR77C1 (rollback to HCR77B1)	• Chinese SMx family of algorithms (SM4, SM2, SM3) • International algorithms (AES, DES, ECC and RSA)	• HCR77B1 PTF OA49069 base

z/OS ICSF HCR77C1 – CKDS Key Token Browser



```
------------------------------ ICSF - CKDS KEYS -------------------------------

Active CKDS: EYSHA.ICSF.CSF77C1.CKDSR                         Keys: 1184

Enter the number of the desired option.
  1  List and manage all records
  2  List and manage records with label key type _________   leave blank for
                                                             list, see help
  3  List and manage records that are _________  (ACTIVE, INACTIVE, ARCHIVED)
  4  List and manage records that contain unsupported CCA keys
  5  Display the key attributes and record metadata for a record
  6  Delete a record
  7  Generate AES DATA keys

Full or partial record label
  ==> DATASET.*
  The label may contain up to seven wild cards (*)

Number of labels to display ==> 100  (Maximum 100)

Press ENTER to go to the selected option.
OPTION ===>
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

With HCR77C1:

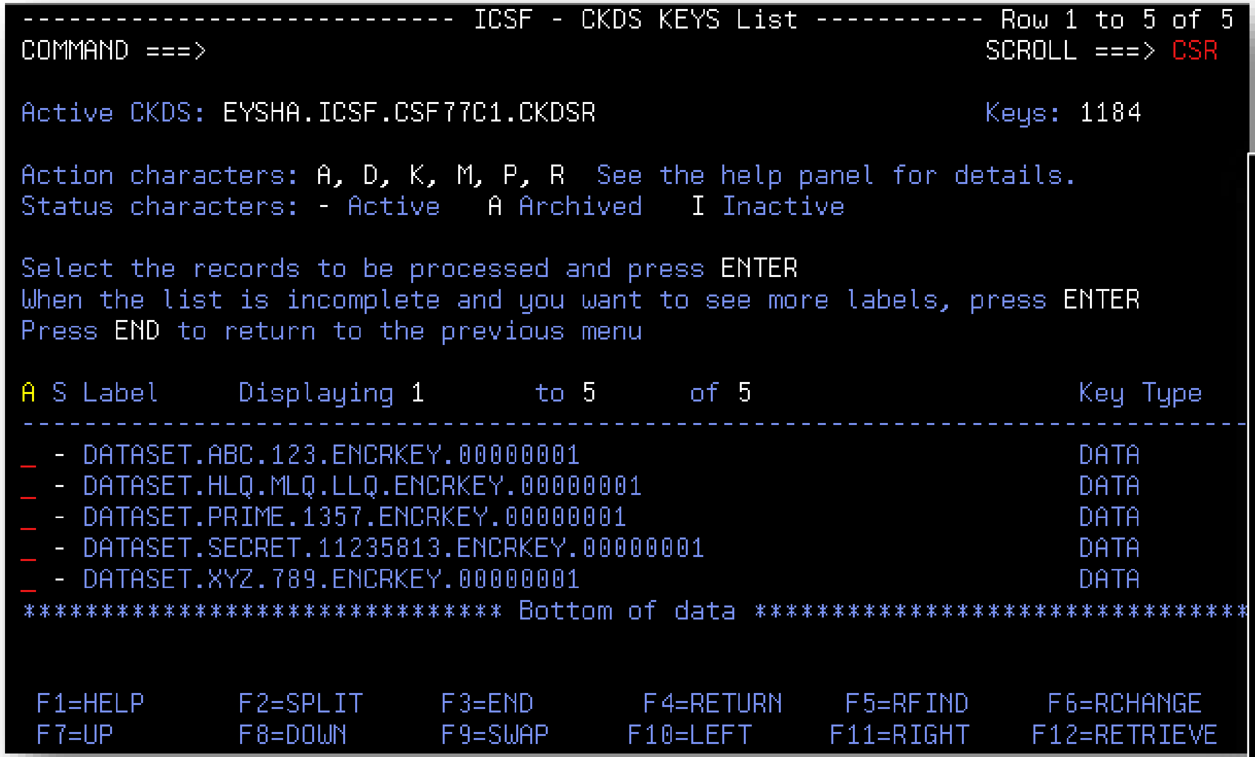
ICSF supports a CKDS Browser (ICSF Panel Option 5.5) that allows users to list records in the active CKDS.

When the format of the CKDS 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.

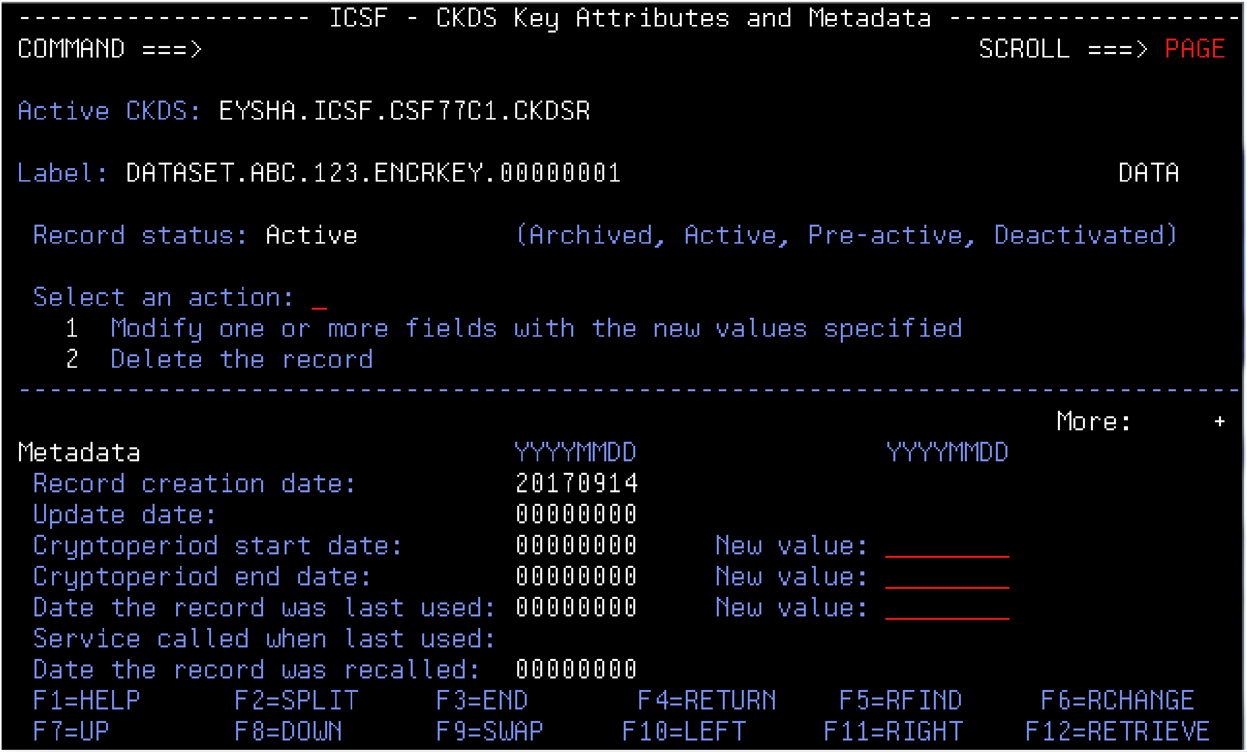
When the format of the CKDS is non-KDSR, the options will be to display key attributes and delete the record.

z/OS ICSF HCR77C1 – CKDS Key Token Browser



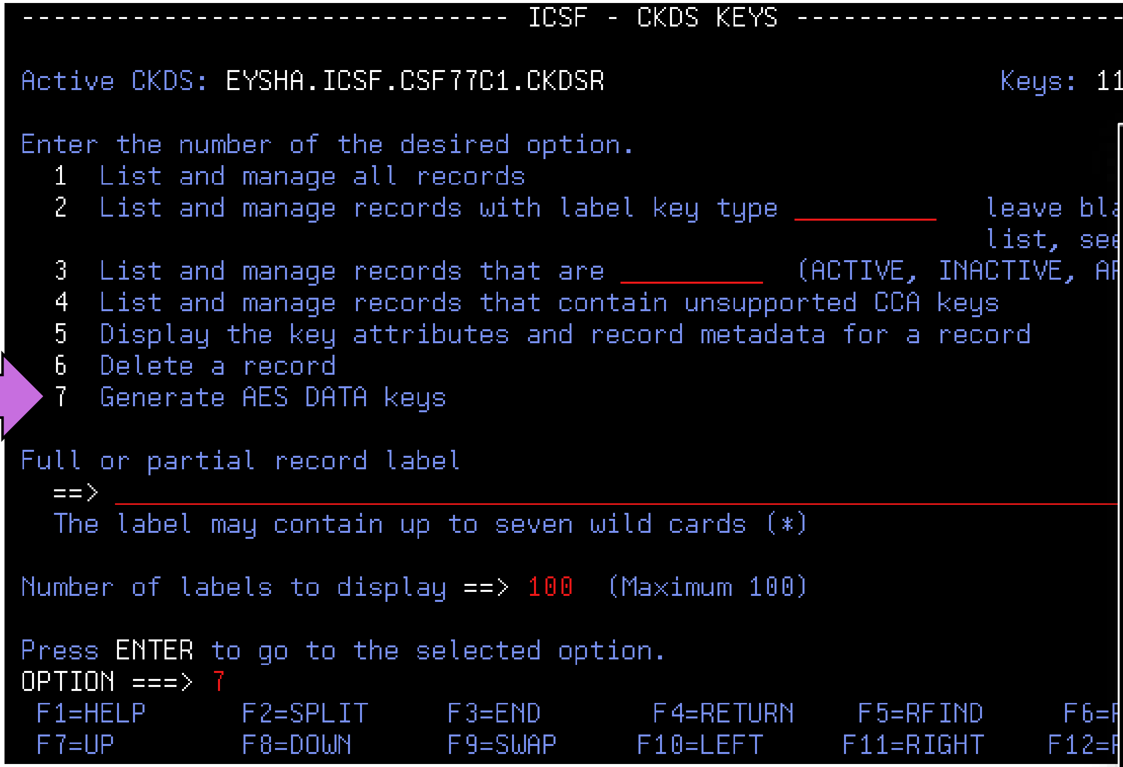
```
---------------------------- ICSF - CKDS KEYS List ----------- Row 1 to 5 of 5
COMMAND ===>                                                  SCROLL ===> CSR

Active CKDS: EYSHA.ICSF.CSF77C1.CKDSR                           Keys: 1184

Action characters: A, D, K, M, P, R  See the help panel for details.
Status characters: - Active    A Archived    I Inactive

Select the records to be processed and press ENTER
When the list is incomplete and you want to see more labels, press ENTER
Press END to return to the previous menu

A S Label     Displaying 1        to 5       of 5                     Key Type
--------------------------------------------------------------------------------
_ - DATASET.ABC.123.ENCRKEY.00000001                                  DATA
_ - DATASET.HLQ.MLQ.LLQ.ENCRKEY.00000001                              DATA
_ - DATASET.PRIME.1357.ENCRKEY.00000001                               DATA
_ - DATASET.SECRET.11235813.ENCRKEY.00000001                          DATA
_ - DATASET.XYZ.789.ENCRKEY.00000001                                  DATA
******************************* Bottom of data ********************************


 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

View key record fields and metadata



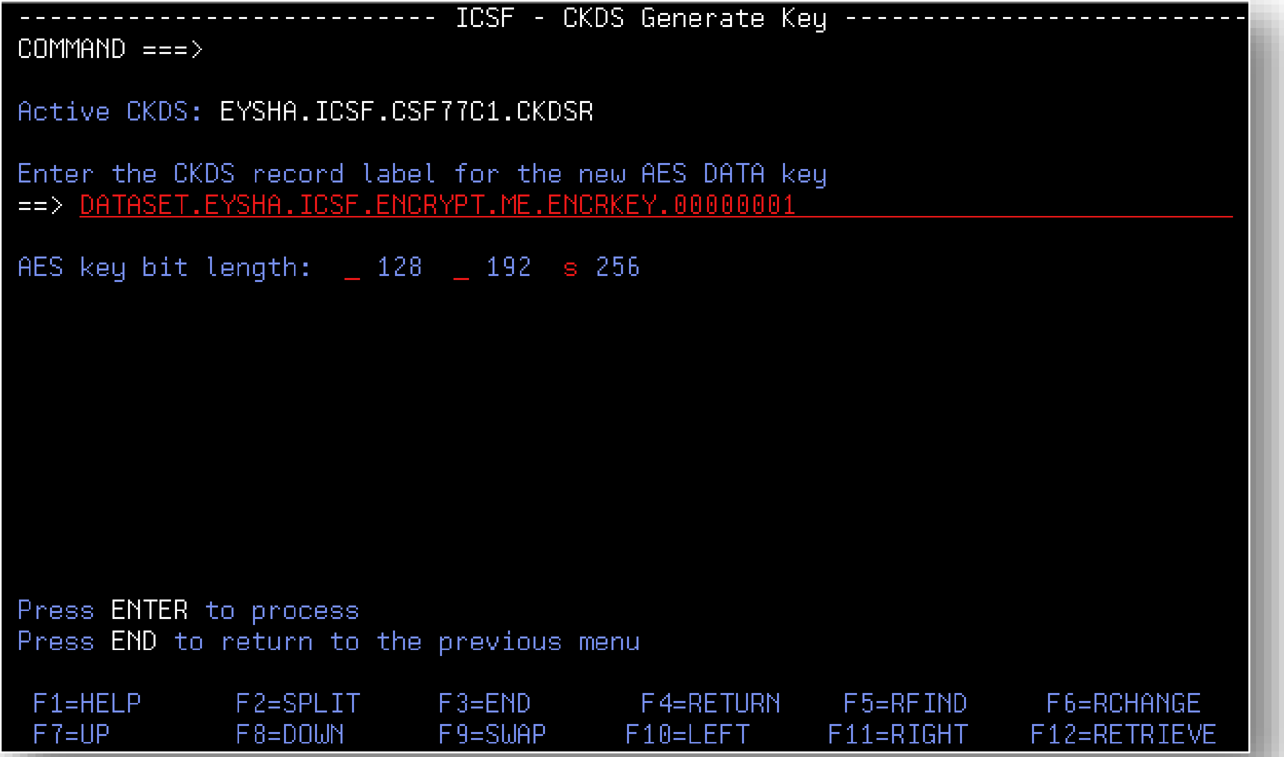
```
------------------- ICSF - CKDS Key Attributes and Metadata -------------------
COMMAND ===>                                                  SCROLL ===> PAGE

Active CKDS: EYSHA.ICSF.CSF77C1.CKDSR

Label: DATASET.ABC.123.ENCRKEY.00000001                                   DATA

 Record status: Active          (Archived, Active, Pre-active, Deactivated)

 Select an action: _
   1  Modify one or more fields with the new values specified
   2  Delete the record
--------------------------------------------------------------------------------
                                                                 More:      +
Metadata                        YYYYMMDD                YYYYMMDD
 Record creation date:          20170914
 Update date:                   00000000
 Cryptoperiod start date:       00000000     New value: ________
 Cryptoperiod end date:         00000000     New value: ________
 Date the record was last used: 00000000     New value: ________
 Service called when last used:
 Date the record was recalled:  00000000
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

z/OS ICSF HCR77C1 – CKDS Key Token Browser



```
------------------------------ ICSF - CKDS KEYS ------------------------------

Active CKDS: EYSHA.ICSF.CSF77C1.CKDSR                          Keys: 1149

Enter the number of the desired option.
  1  List and manage all records
  2  List and manage records with label key type ________   leave bla
                                                            list, see
  3  List and manage records that are ________  (ACTIVE, INACTIVE, AR
  4  List and manage records that contain unsupported CCA keys
  5  Display the key attributes and record metadata for a record
  6  Delete a record
  7  Generate AES DATA keys

Full or partial record label
  ==> ______________________________________________________________
  The label may contain up to seven wild cards (*)

Number of labels to display ==> 100  (Maximum 100)

Press ENTER to go to the selected option.
OPTION ===> 7
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=R
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=R
```

Generate AES DATA keys



```
-------------------------- ICSF - CKDS Generate Key --------------------------
COMMAND ===>

Active CKDS: EYSHA.ICSF.CSF77C1.CKDSR

Enter the CKDS record label for the new AES DATA key
==> DATASET.EYSHA.ICSF.ENCRYPT.ME.ENCRKEY.00000001

AES key bit length:  _ 128  _ 192  s 256










Press ENTER to process
Press END to return to the previous menu


 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

z/OS ICSF HCR77C1 – Crypto Usage Tracking

ICSF will provide crypto usage tracking of applications and components that invoke ICSF services in HCR77C1. 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

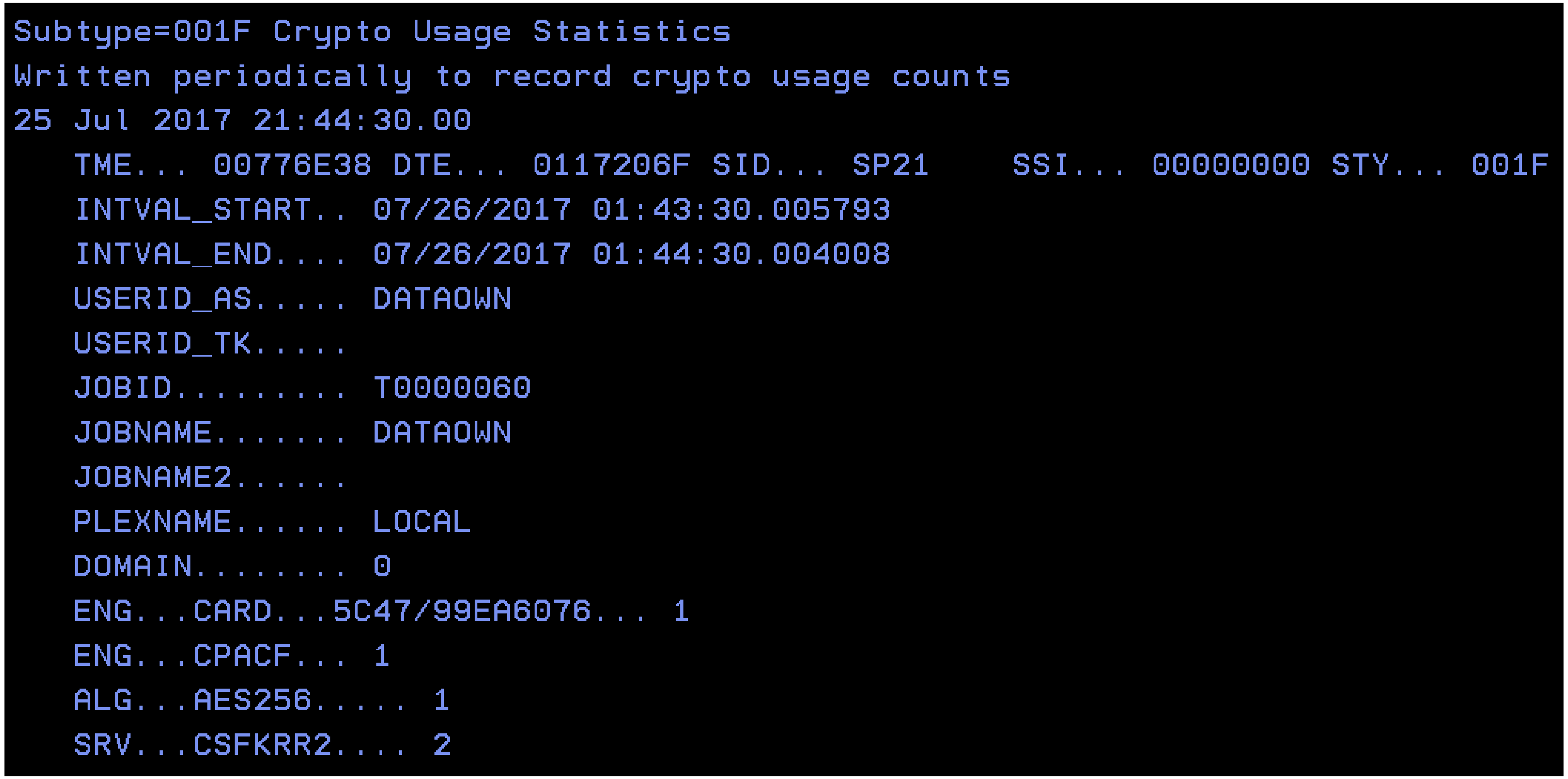
ENG: Tracks crypto engine usage. When enabled, ICSF tracks the usage of Crypto Express Adapters, Regional Cryptographic Servers, CPACF and Software.
SRV: Tracks crypto service usage. When enabled, ICSF tracks the usage of ICSF callable services and User Defined Extensions (UDX).
ALG: Tracks crypto algorithm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

Let’s take a look!

z/OS ICSF HCR77C1 – Crypto Usage Tracking Example



```
Subtype=001F Crypto Usage Statistics
Written periodically to record crypto usage counts
25 Jul 2017 21:44:30.00
   TME... 00776E38 DTE... 0117206F SID... SP21    SSI... 00000000 STY... 001F
   INTVAL_START.. 07/26/2017 01:43:30.005793
   INTVAL_END.... 07/26/2017 01:44:30.004008
   USERID_AS..... DATAOWN
   USERID_TK.....
   JOBID......... T0000060
   JOBNAME....... DATAOWN
   JOBNAME2......
   PLEXNAME...... LOCAL
   DOMAIN........ 0
   ENG...CARD...5C47/99EA6076... 1
   ENG...CPACF... 1
   ALG...AES256..... 1
   SRV...CSFKRR2.... 2
```

z/OS ICSF Exploiters

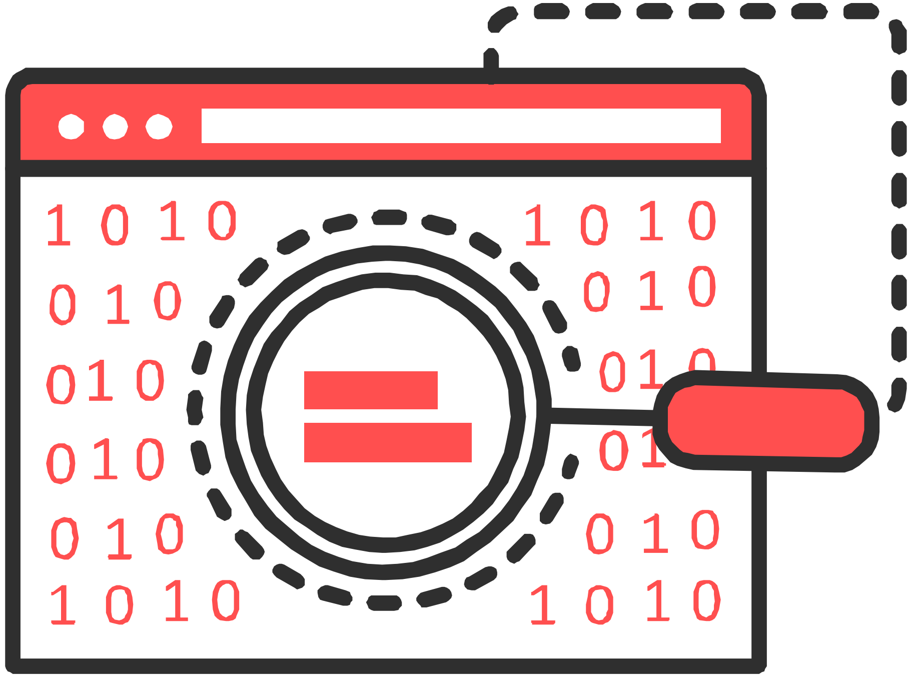
z/OS Software Components

- 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
- …

IBM & ISV Solutions

- Guardium Data Encryption
- Sterling Connect:Direct
- …

Questions?



Additional Resources

IBM Crypto Education Community
https://www.ibm.com/developerworks/community/groups/community/crypto

Appendix A: Crypto Ecosystem

- Enterprise Key Management Foundation (EKMF)
- Advanced Crypto Service Provider (ACSP)
- Security Key Lifecycle Manager (SKLM)
- Encryption Facility (EF) for z/OS
- Guardium Data Encryption (GDE)

Appendix A: Enterprise Key Management Foundation (EKMF)

Secure workstation

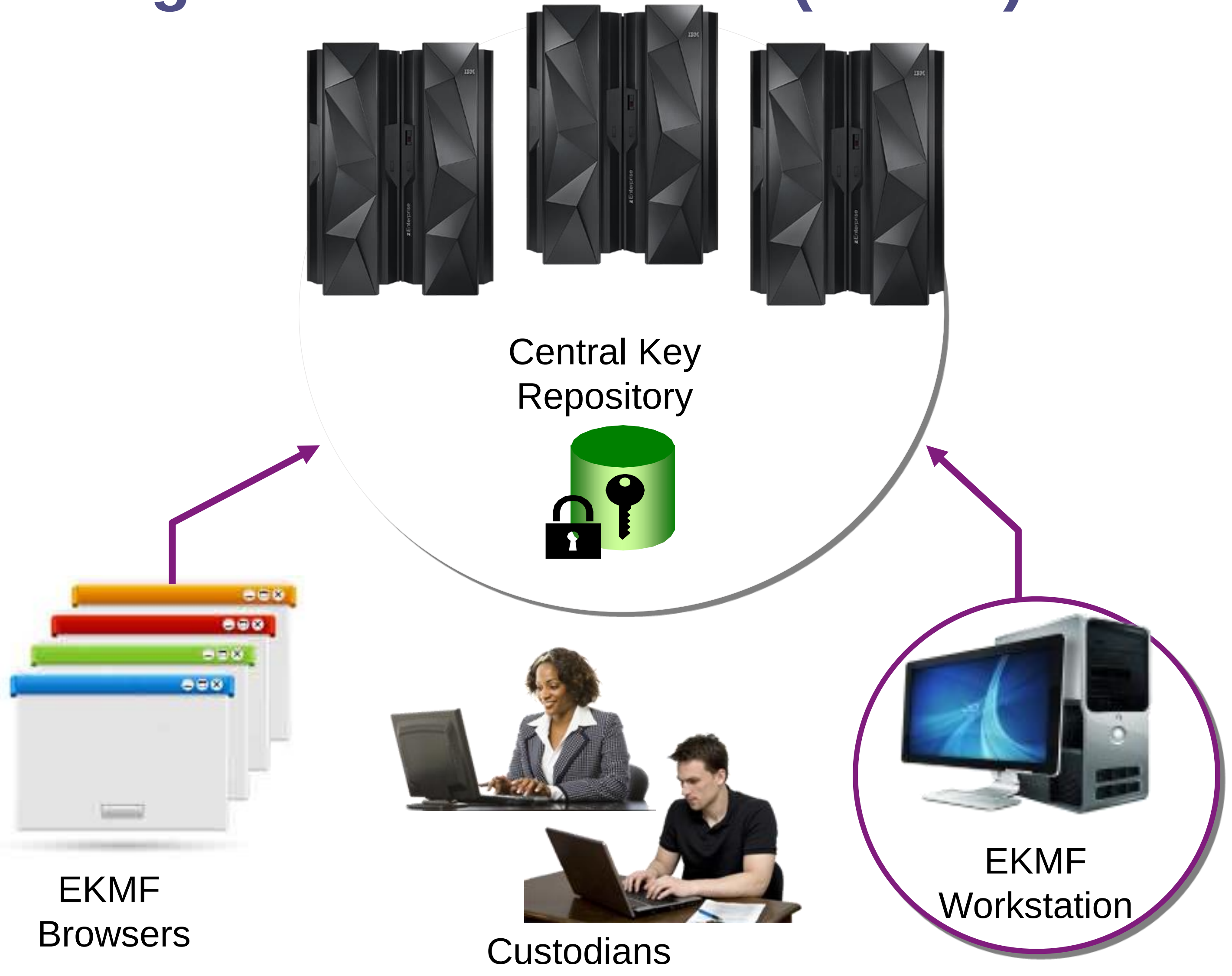
- is used for **generating** all new keys by users authenticated with **smart cards** or automatically based on requests. Workstation utilizes **IBM 4765/7**

Central repository

- contains **keys** and **metadata** for all cryptographic keys produced by the EKMF workstation.
  This enables easy **backup** and recovery of key material.

EKMF Browser

- features **monitoring** capabilities and enables **planning** of future key handling session to be executed on the workstation.



Central Key Repository

EKMF Browsers

Custodians

EKMF Workstation

Note that while this is a mainframe centric view, EKMF supports distributed platforms as well.

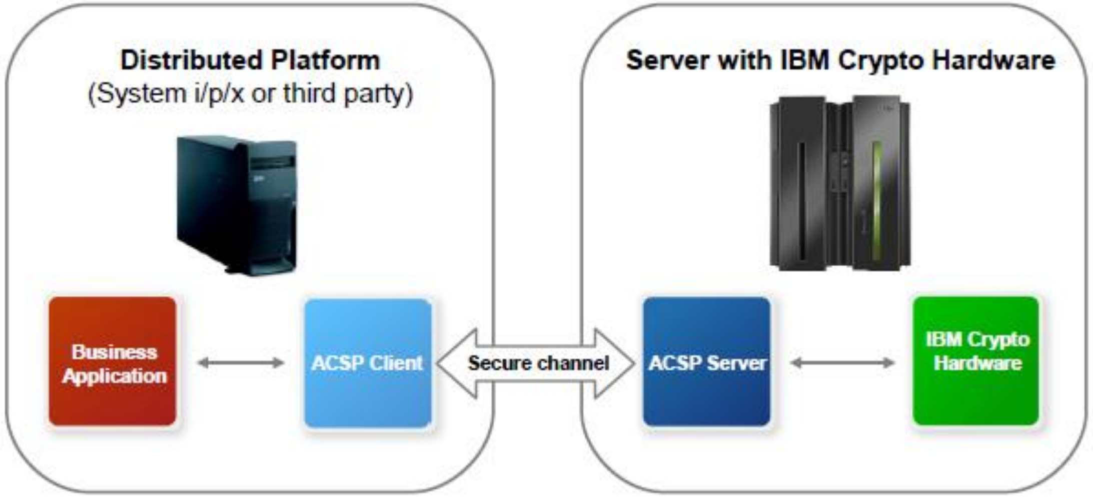
Appendix A: Advanced Crypto Service Provider (ACSP)

Expose crypto functions to client applications across heterogeneous systems and environments

- zEnterprise and zBX Application Serving Blades (ASB)
- IBM PureSystems
- Distributed platforms

Benefits

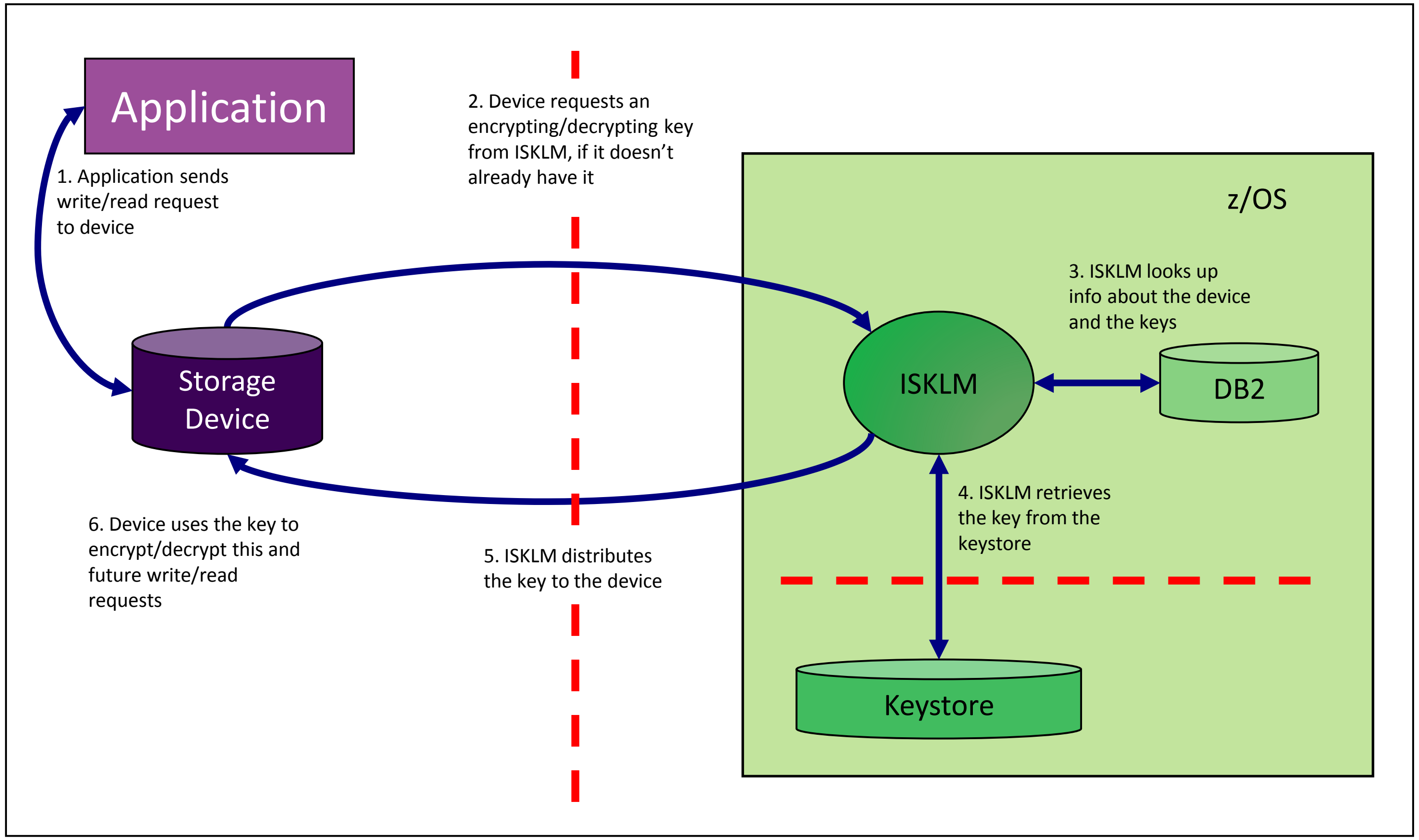
- Virtualization and cost effective use of available crypto capacity
- Reduced administration and simpler key management
- Crypto support for platforms with no native IBM crypto HW support
- Easier to develop/deploy applications using crypto
- High scalability, reliability, and availability
- Leverages existing business continuity plans and procedures



- **ACSP client platforms**
  - AIX, IBM i, Linux, Windows, z/OS, Linux on z
  - PureSystems
  - In reality, any Java platform
- **ACSP client APIs**
  - CCA in Java and C
  - PKCS#11
  - REST
- **Transport network**
  - IP
  - SSL/TLS protected (client/server auth)
- **ACSP server platforms**
  - z Systems: z/OS (CEX3/4/5)
  - System p: AIX (4765)
  - x86: SLES, RHEL (4765)
  - IBM PureSystems

Appendix A: Security Key Lifecycle Manager (SKLM)

IBM Security Key Lifecycle Manager for z/OS manages encryption keys for storage. It integrates with encrypting storage devices with hardware encryption for performance, Resource Access Control Facility (RACF), Integrated Cryptographic Service Facility (ICSF) and IBM Enterprise Key Management Foundation (EKMF).



1. Application sends write/read request to device

2. Device requests an encrypting/decrypting key from ISKLM, if it doesn’t already have it

3. ISKLM looks up info about the device and the keys

4. ISKLM retrieves the key from the keystore

5. ISKLM distributes the key to the device

6. Device uses the key to encrypt/decrypt this and future write/read requests

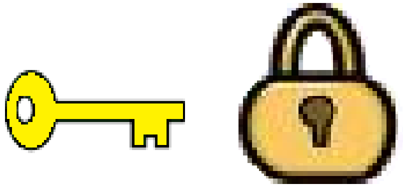
Appendix A: Encryption Facility (EF) for z/OS

The Encryption Facility for z/OS 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 z/OS and IBM 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.

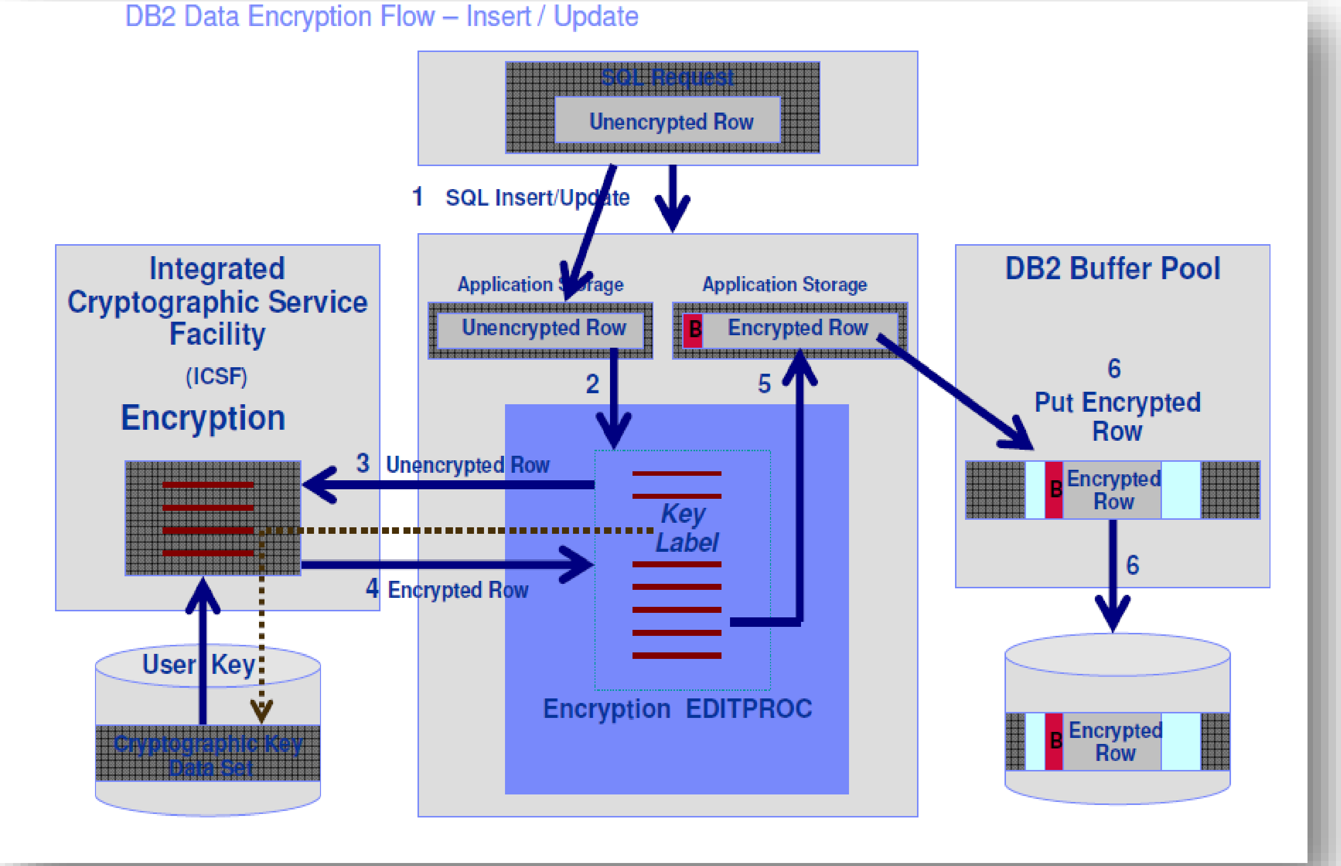
EF for z/OS provides services for:

- 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



Appendix A: Guardium Data Encryption (GDE)

Guardium Data Encryption provides row and field based encryption of DB2 and IMS data.



DB2 Data Encryption Flow – Insert / Update

Appendix B: UDX Support

UDX support available for Crypto Express6S features defined as CCA coprocessors

- Allows **additional functions** to the CCA API, which **execute inside the secure crypto feature**
  - 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 on z Systems

Appendix C: Additional z/OS ICSF Features

- HCR77A1 – Key Reference Date Tracking
- HCR77B1 – Format Preserving Encryption
- HCR77C0 – Key Lifecycle & Usage Auditing
- HCR77C1 – Optional CKDSN /PKDSN in CSFPRMxx
- HCR77C1 – CICS Audit Records

z/OS ICSF HCR77A1 – Key Reference Date Tracking

ICSF can track when cryptographic keys stored in the CKDS, PKDS or TKDS (in KDSR common record format) are referenced in cryptographic operations. ICSF will update the KDS record metadata with the last reference date associated with the key.

The last reference date is useful for determining :

- If a key has been recently used
- If a key can be archived or deleted

How to Use:

1. Convert KDS to KDSR Format
2. Update the ICSF installation options data set
   a. Set KDSREFDAYS to indicate how often to check for a new reference. For example, 1 day to check each day or 30 days to check for references in a 30-day time frame.
3. Use the keys in cryptographic operations
   a. When the key is used in a cryptographic operation, ICSF will update the reference date (depending on the KDSREFDAYS range).

z/OS ICSF HCR77B1 - Format Preserving Encryption

- Protect cardholder data
- Simplify application data encryption by allowing the encrypted data to flow through payment systems without requiring massive application redesign
- Satisfy audit requirements to have data encrypted during transmission across open, public networks

ICSF Callable Services:

- CCA Services:
  - CSNBFPEE – FPE Encipher
  - CSNBFPED – FPE Decipher
  - CSNBFPET – FPE Translate
  - CSNBPTRE – Encrypted Pin Translate Enhanced
- CPACF Services:
  - CSNBFLE – Field Level Encipher
  - CSNBFLD – Field Level Decipher

*Note: Please contact Visa for details on licensing the use of this technology.

z/OS ICSF HCR77C0 – Key Lifecycle & Usage Auditing

ICSF HCR77C0 supports auditing the complete life cycle of key material from creation to disposal including keys that may not be stored in an ICSF Key Data Set (in KDSR common record format).

	Pre-HCR77B0	HCR77B0	HCR77C0
Key Creation	YES	YES	YES
Key Generation	KDS Keys Only	KDS Keys Only	YES
Key Update	YES	YES	YES
Key Deletion	YES	YES	YES
Key Import	TKE OpKeyLoad Only	TKE OpKeyLoad Only	YES
Key Export	NO	NO	YES
Key Archival	N/A	YES	YES
Key Restore	N/A	YES	YES
Key State: Pre-Activation	N/A	NO	YES
Key State: Activated	N/A	NO	YES
Key State: Deactivated	N/A	NO	YES

z/OS ICSF HCR77C0 – Key Lifecycle & Usage Auditing

SMF Record type 82 records information about the events and operations of ICSF. Several new subtypes are available for Key Life Cycle and Key Usage Auditing:

- **Subtype 40** – written for lifecycle events related to symmetric CCA tokens.
- **Subtype 41** – written for lifecycle events related to asymmetric CCA tokens.
- **Subtype 42** – written for lifecycle events related to PKCS#11 objects.
- **Subtype 44** – written for usage events related to symmetric CCA tokens.
- **Subtype 45** – written for usage events related to asymmetric CCA tokens.
- **Subtype 46** – written for usage events related to PKCS#11 objects.
- **Subtype 47** – written for supported PKCS #11 usage events which don't involve an object.

z/OS ICSF HCR77C0 – Key Lifecycle & Usage Auditing

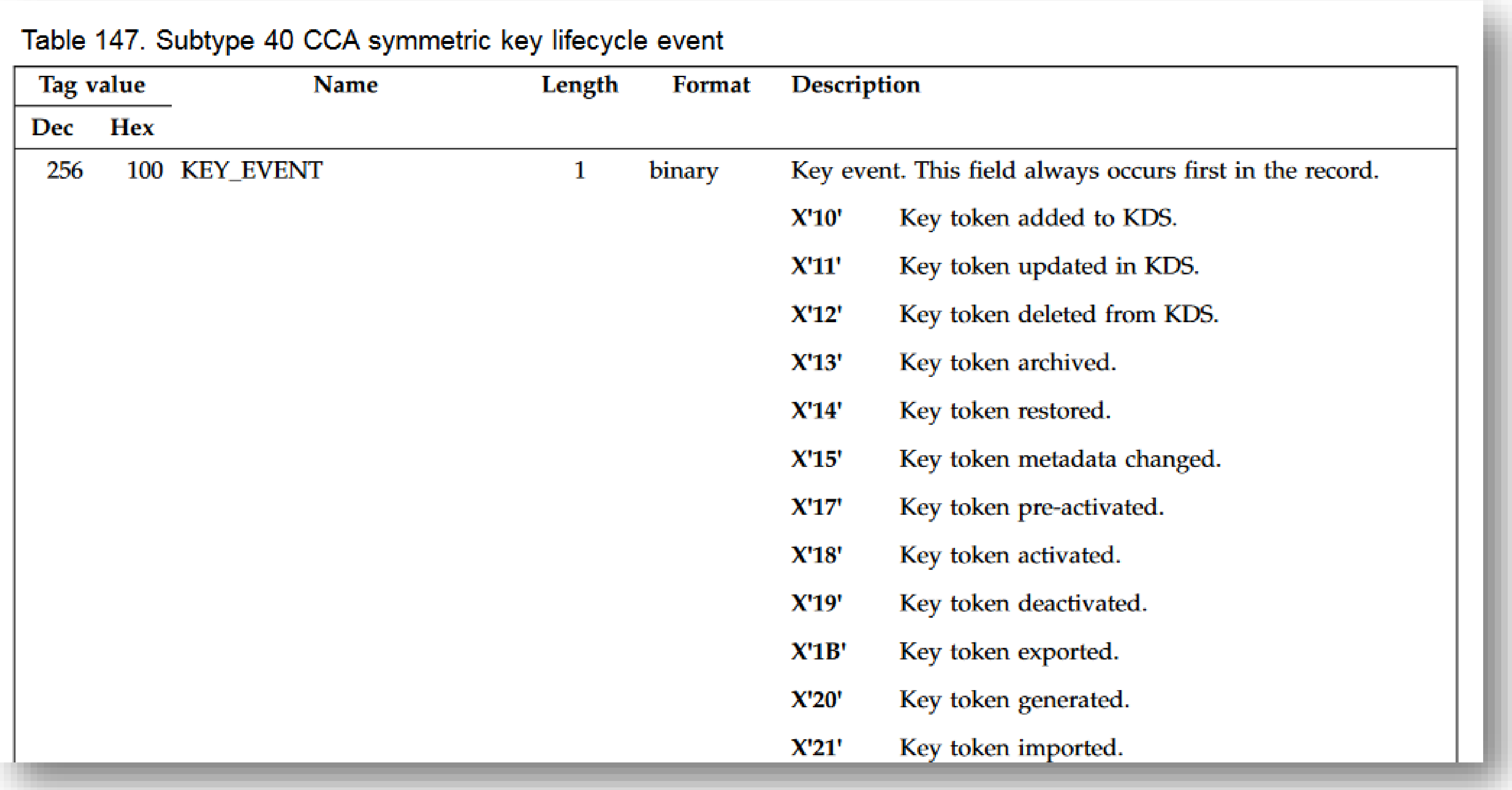


Table 147. Subtype 40 CCA symmetric key lifecycle event

Tag value		Name	Length	Format	Description	
Dec	Hex					
256	100	KEY_EVENT	1	binary	Key event. This field always occurs first in the record.	
					X'10'	Key token added to KDS.
					X'11'	Key token updated in KDS.
					X'12'	Key token deleted from KDS.
					X'13'	Key token archived.
					X'14'	Key token restored.
					X'15'	Key token metadata changed.
					X'17'	Key token pre-activated.
					X'18'	Key token activated.
					X'19'	Key token deactivated.
					X'1B'	Key token exported.
					X'20'	Key token generated.
					X'21'	Key token imported.

z/OS ICSF HCR77C0 – Optional CKDSN / PKDSN in CSFPRMxx

In HCR77C0, ICSF removed the requirement for a CKDSN and/or PKDSN in the ICSF Options Data Set.

This enables easier setup and configuration for those clients not wanting to exploit secure key operations or manage CCA symmetric or asymmetric key tokens.

z/OS ICSF HCR77C1 – CICS Audit Records

Prior to HCR77C1:

When a CICS transaction makes a call to ICSF, security checks were performed against the CICS started task user ID. As a result, SMF Type 80 Subtype 2 records contained the CICS started task user ID rather than the CICS client user ID.

With HCR77C1:

ICSF will support a new option CICSAUDIT(YES) in the ICSF Installation Options Data Set.

When a CICS transaction running on the Quasi Reentrant (QR) task calls an ICSF service:

- ICSF will extract the CICS client user id from the CICS client ACEE for inclusion in the log string for SMF Type 80 Subtype 2 records.
- ICSF will also extract the CICS application ID and CICS transaction ID for inclusion in ICSF SMF records.