IBM z/OS

Crypto Lockdown

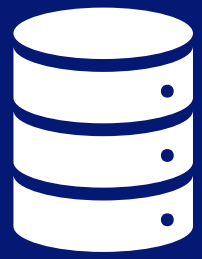
and other best practices

Eysha S. Powers
IBM Senior Technical Staff Member
Chief Architect, IBM Z Crypto Portfolio
eysha@us.ibm.com

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.

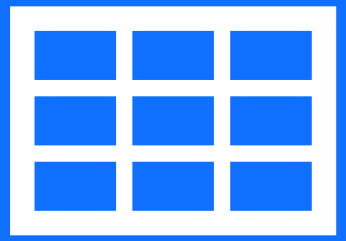
The z/OS crypto stack enables the encryption of data at rest and data in flight.

Disk and tape encryption

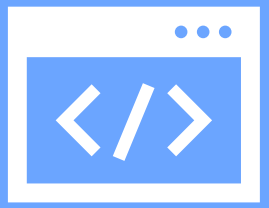


Data set level encryption

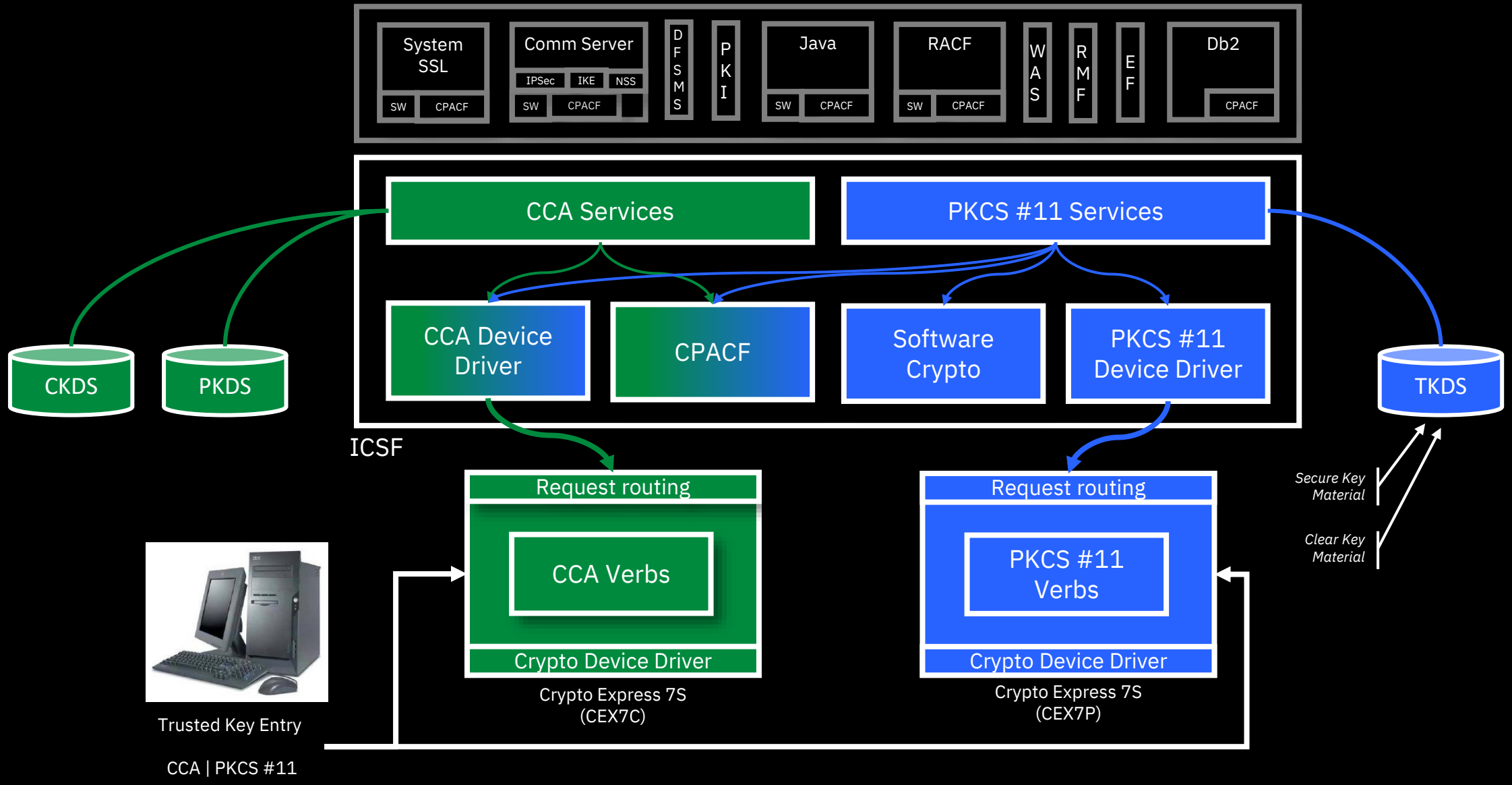
Database level encryption



Application-level encryption



z/OS Crypto Stack



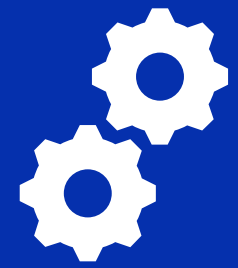
Why Lockdown

the z/OS crypto stack?

Reduce attack surface

Routine security hygiene

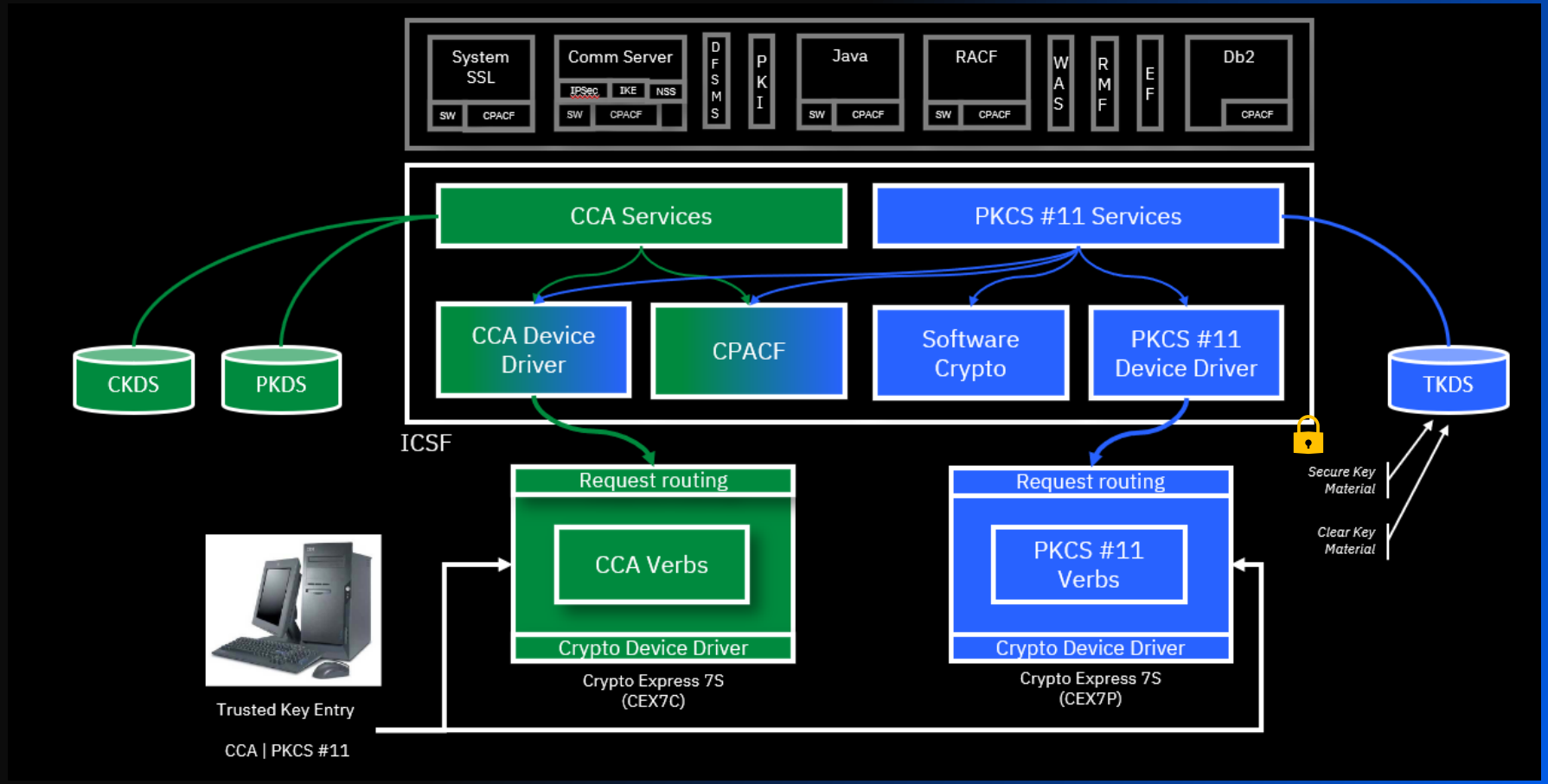
Exploit new capabilities and features



Regulatory compliance

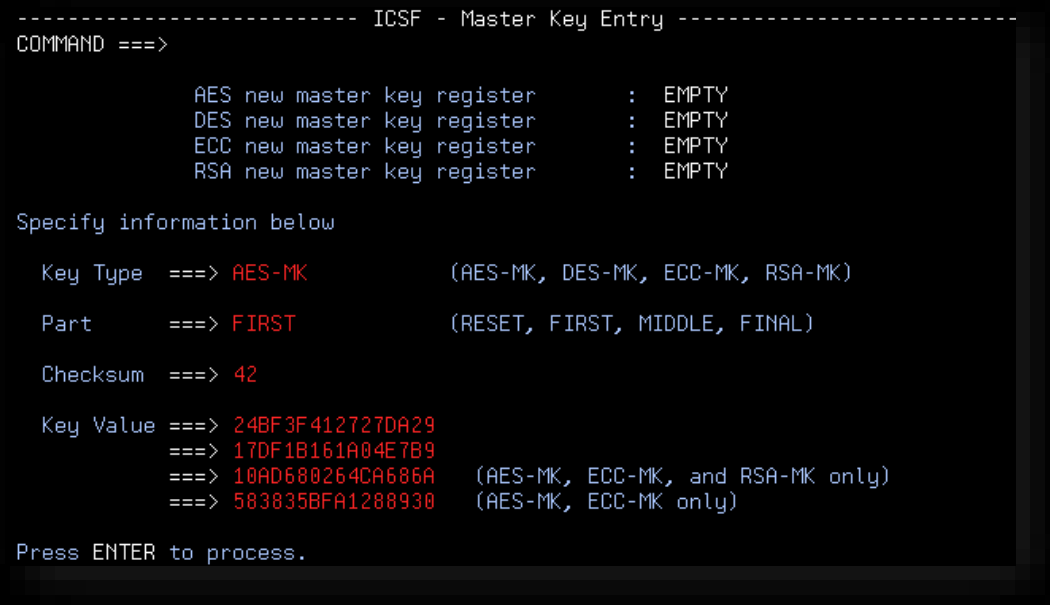
General Hardening

z/OS Integrated Cryptographic Services Facility



Limit authorization to CSFSERV resources

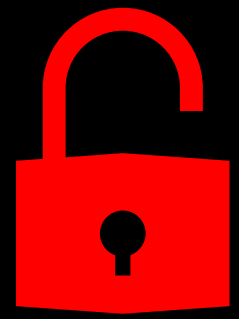
The CSFSERV class controls access to ICSF callable services and ICSF TSO panel utilities.



Risk Factors

By default, the CSFSERV class grants access to ICSF services and utilities if there is no covering profile except

- Key Data Set Update (CSFKDU)
- Key Data Set Record Retrieve (CSFRRT)



Lockdown Checklist

- ✓ CSFSERV class is ACTIVE
- ✓ CSFSERV class is RACLISTed
- ✓ CSFSERV class is GENERIC
- ✓ CSFSERV has a backstop profile (i.e. CSFSERV * or CSFSERV **) with UACC(NONE)
- ✓ Review and assess CSFSERV risk factors

CSFSERV Risk Assessment

Thanks to Roan Dawkins and the ICSF development team

High Risk?

- Services which can operate on all keys without checking CSFKEYS: CSFKDSL, CSFKDMR, CSFKDMW
- Services which allow an entire key record to be written to or read from any key data set without CSFKEYS checking: CSFKDU, CSFRRT
- Profiles protecting panels and utilities: CSFBRCK, CSFBRPK, CSFBRTK, CSFCMK, CSFCONV, CSFCRC, CSFDKCS, CSFGKF, CSFKGUP, CSFOPKL, CSFPCAD, CSFPMCI, CSFREFR, CSFRENC, CSFRSWS, CSFRWP, CSFSMK, CSFSSWS, CSFUDM, CSFMPS
- Services which update the CKDS / PKDS:
  - CKDS: CSFKRC, CSFKRC2, CSFKRW, CSFKRW2, CSFKRD, CSFKPI, CSFKPI2, CSFRKA
  - PKDS: CSFPKRC, CSFPKRW, CSFPKRD, CSFPKG, CSFPKI, CSFTBC, CSFRKD
- Services which import a clear key into your system: CSFCKI, CSFCKM, CSFKPI, CSFKPI2, CSFSKM, CSFSKI, CSFSKI2
- TKE special service: CSFPCI

Medium Risk?

- Services which use a key to perform a cryptographic operation (since keys may be protected using CSFKEY resources)

Low Risk?

- Services which provide information about the crypto environment: CSFIQF, CSFIQA
- Services which don’t use a key: CSFOWH, CSFOWH1, CSFRNG, CSFRNGL

Limit authorization to DATASET resources

The DATASET class protects data sets on DASD and tape, including the ICSF CKDS, PKDS, and TKDS.

By default, a user is granted ALTER access to all datasets that have a high level qualifier matching their user id.

Lockdown Checklist

- ✓ DATASET class active
- ✓ SETROPTS PROTECTALL(FAIL) is enabled (wherein GENERIC should also be enabled) or
  - ✓ DATASET resource is defined for the CKDS, PKDS and TKDS with UACC(NONE)

Enable crypto usage statistics

Each ICSF instance can be configured to collect cryptographic usage data when crypto operations are either performed by that ICSF instance or reported to that ICSF instance by the CSFSTAT callable service. Crypto usage data is written to SMF Type 82 Subtype 31 records.

- ENG: Tracks the usage of cryptographic engines. When enabled, ICSF tracks the usage of Crypto Express HSMs, regional cryptographic servers, CPACF, and software.
- SRV: Tracks the usage of cryptographic services. When enabled, ICSF tracks the usage of ICSF callable
- services and UDXes.
- ALG: Tracks the usage of cryptographic algorithms. When enabled, ICSF tracks the usage of cryptographic algorithms that are referenced in cryptographic operations. Key generation, derivation, and import have limited support.

By default, crypto usage tracking is disabled.

Available in z/OS 2.4 and ICSF HCR77C1 and later

Lockdown Checklist

- ✓ For “always on” tracking, enable STATS in the ICSF installation options data set (i.e. CSFPRMxx PARMLIB member)
- ✓ For “on demand” tracking, issue the SETICSF STATS operator command to enable and/or disable dynamically
- ✓ Review usage data for
  - Vulnerable algorithms and key lengths
  - Expected / unexpected crypto usage by jobs / users
  - Expected hardware crypto engine usage
  - Peak periods of crypto utilization

Enable key lifecycle auditing for all active key data sets

Each ICSF instance can be configured to audit the lifecycle of keys as they transition through the system. Keys can be audited from the time of their initial generation until their eventual disposal. A sample lifecycle of a key might be… key generated, key updated, key activated, key deactivated, key deleted. Audit data is written to SMF Type 82 Subtype 40, 41, and 42 records.

- AUDITKEYLIFECKDS enables the lifecycle auditing of keys in the CKDS
- AUDITKEYLIFEPKDS enables the lifecycle auditing of keys in the PKDS
- AUDITKEYLIFETKDS enables the lifecycle auditing of keys in the TKDS

By default, key lifecycle auditing is disabled.

Available in z/OS 2.3 and later

Lockdown Checklist

- ✓ For “always on” auditing, enable AUDITKEYLIFExKDS in the ICSF installation options data set (i.e. CSFPRMxx PARMLIB member)
- ✓ For “on demand” auditing, issue the SETICSF OPT operator command to enable and/or disable dynamically
- ✓ Consult usage data to determine
  - When keys were generated
  - When keys were activated or deactivated
  - When keys were updated
  - When keys were deleted

Enable key usage auditing for all active key data sets

Each ICSF instance can be configured to audit key usage. The SMF records are written continuously as indicated by the audit interval. SMF record contents reveal detailed information about the key in addition to the user identity associated with its use. Audit data is written to SMF Type 82 Subtype 44, 45, 46, and 47 records.

- AUDITKEYUSGCKDS enables usage auditing of keys in the CKDS
- AUDITKEYUSGPKDS enables usage auditing of keys in the PKDS
- AUDITPKCS11USG enables usage auditing of PKCS #11 keys

By default, key usage auditing is disabled.

Available in z/OS 2.3 and later

Lockdown Checklist

- ✓ For “always on” auditing, enable AUDITKEYUSExKDS or AUDITPKCS11USG in the ICSF installation options data set (i.e. CSFPRMxx PARMLIB member)
- ✓ For “on demand” auditing, issue the SETICSF OPT operator command to enable and/or disable dynamically
- ✓ Consult usage data to determine
  - Which key was used
  - Who used the key
  - When the key was used

Enable key reference date tracking

ICSF provides the ability to capture the last date a key was referenced *in a cryptographic operation* and store that date in the metadata section of the key record in the CKDS, PKDS, or TKDS. In conjunction with the CSFKDSL and CSFKDMR callable services, an administrator can query ICSF for keys referenced or not referenced within a period of time.

By default, key reference date tracking *is enabled*.

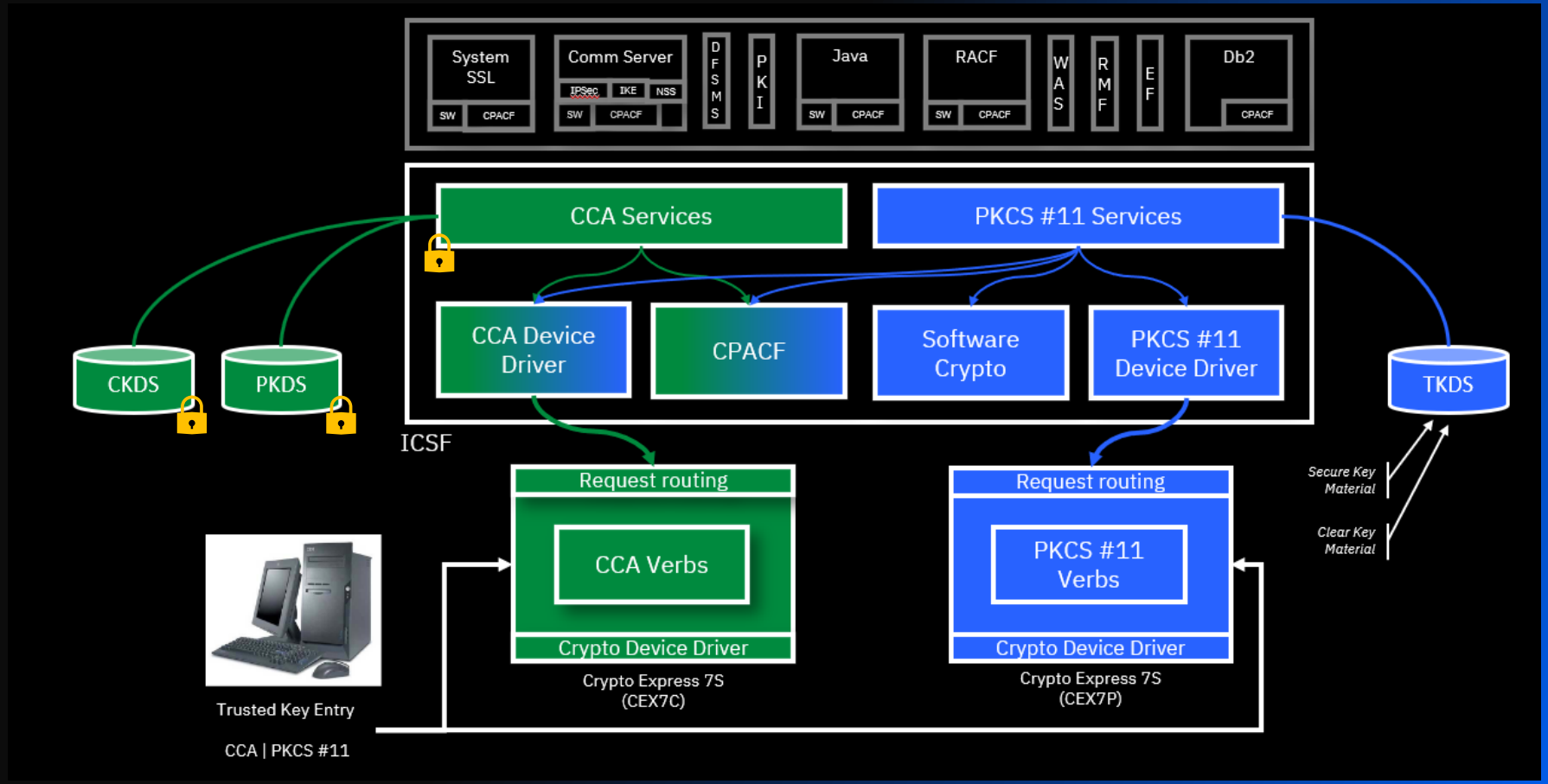
Available in z/OS 2.2 and ICSF HCR77B0 and later

Lockdown Checklist

- ✓ Requires KDSR format key data sets
- ✓ For “always on” enablement, set KDSREFDAYS greater than 0 in the ICSF installation options data set (i.e. CSFPRMxx PARMLIB member)
- ✓ For “on demand” enablement, update KDSREFDAYS in the ICSF installation options data set and issue the SETICSF REFRESH operator command to enable and/or disable dynamically
- ✓ Periodically examine which keys are no longer used for cryptographic operations. Determine if those keys should be archived or deactivated.

Hardening CCA

z/OS Integrated Cryptographic Services Facility



Limit authorization to CSFKEYS resources

The CSFKEYS class controls access to cryptographic keys in the CKDS and PKDS and enables/disables the use of protected keys.

With RACF-based SAF protection, CSFKEYS resources can be defined as discrete or generic (i.e. wildcard) profiles. As a result, *KDS key label naming conventions are important*.

Naming Conventions

The CSFKEYS class grants access to CCA cryptographic keys if there is no profile covering the label

Lockdown Checklist

- ✓ CSFKEYS class is ACTIVE
- ✓ CSFKEYS class is RACLISTed
- ✓ CSFKEYS class is GENERIC
- ✓ CSFKEYS has a backstop profile (i.e. CSFKEYS * or CSFKEYS **) with UACC(NONE)
- ✓ Choose your naming convention wisely!

Key Label Naming Conventions

A key label can consist of up to 64 characters. The first character must be alphabetic or a national character (#, $, @). The remaining characters can be alphanumeric, a national character (#, $, @), or a period (.).

Naming considerations:

- the LPAR associated with the key
- the type of data being encrypted
- the owner associated with the key
- the date the key was created
- the application intended to use the key
- The generic profile to protect the key
- A sequence number for the key

Policy-Based Dataset Encryption Example:

Key Label:

DATASET.<dataset_resource>.ENCRKEY.<seqno>

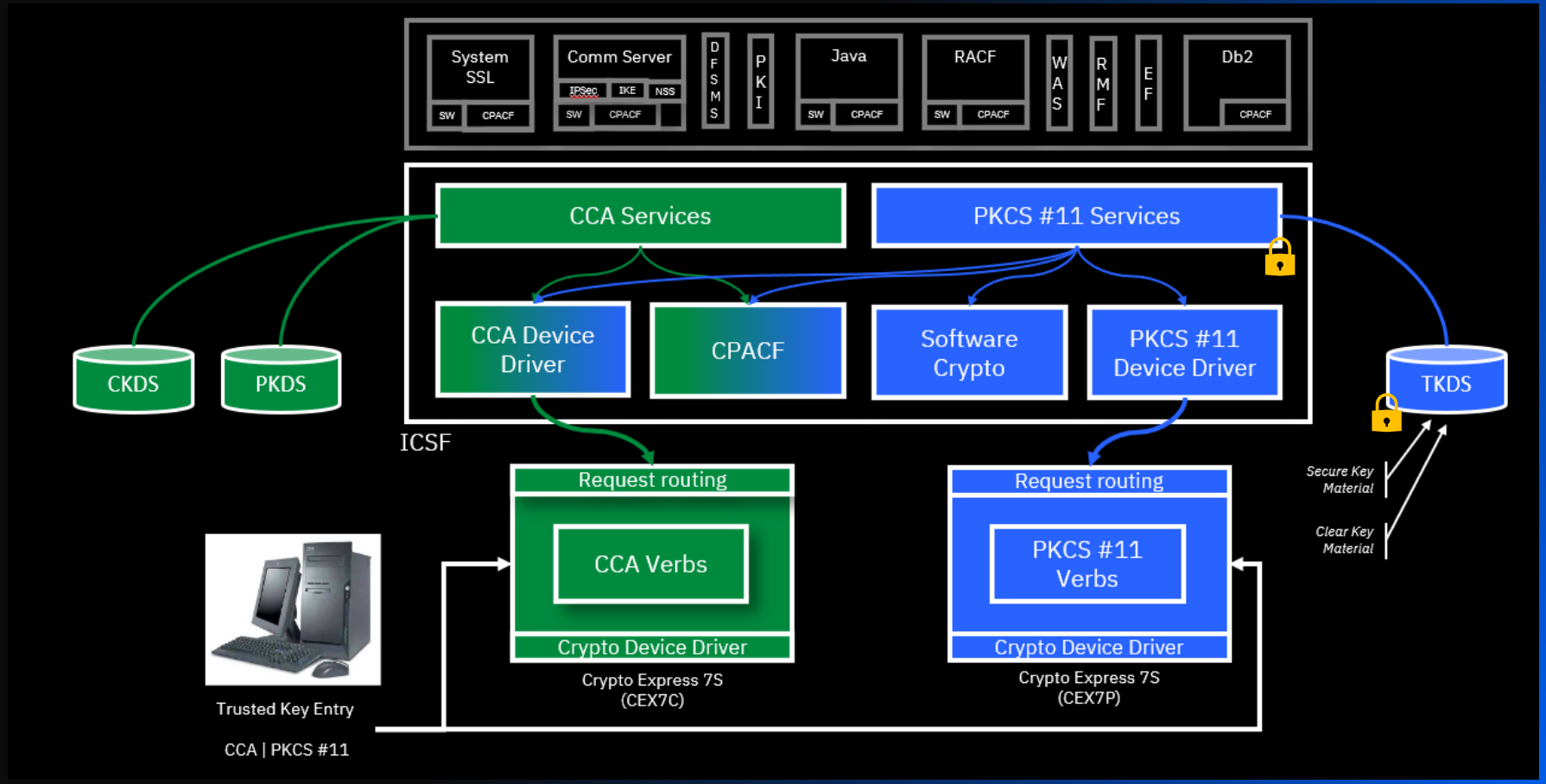
CSFKEYS Profile:

RDEFINE CSFKEYS DATASET. <dataset_resource>.ENCRKEY.* UACC(NONE)

Note: <dataset_resource> would be replaced with the DATASET resource and <seqno> would be replaced with a sequence number.

Hardening PKCS#11

z/OS Integrated Cryptographic Services Facility



Limit authorization to CRYPTOZ resources

The CRYPTOZ class controls access to and defines policy for cryptographic information within PKCS #11 tokens.

The CRYPTOZ class does not grant access to PKCS #11 cryptographic keys if there is no profile.

Lockdown Checklist

- ✓ CRYPTOZ class is ACTIVE
- ✓ CRYPTOZ class is RACLISTed
- ✓ CRYPTOZ class is GENERIC
- ✓ CRYPTOZ has a backstop USER.* or USER.** profile with UACC(NONE)
- ✓ CRYPTOZ has a backstop SO.* or SO.** profile with UACC(NONE)

Prioritize secure keys and/or disallow clear keys

With PKCS #11, clear key processing can be controlled by policy.

Key Security Objective	RACF Access	Action taken with PKCS #11 coprocessor not available or algorithm not supported	Action taken when PKCS #11 coprocessor available and algorithm supported
Generate no secure keys.	CONTROL	Sensitive – Clear Key Non-sensitive – Clear Key	Sensitive – Clear Key Non-sensitive – Clear Key
Use key sensitivity and environment to determine security	UPDATE or No Decision	Sensitive – Clear Key Non-sensitive – Clear Key	Sensitive – Secure Key Non-sensitive – Clear Key
Ensure keys explicitly marked sensitive are always secure keys	READ	Sensitive – Denied Non-sensitive – Clear Key	Sensitive – Secure Key Non-sensitive – Clear Key
Prevent generation or creation of any clear keys	NONE	Sensitive – Denied Non-sensitive – Denied	Sensitive – Secure Key Non-sensitive – Secure Key

By default, sensitive PKCS #11 keys will be generated as secure keys if an EP11 coprocessor is available, and the algorithm is supported.

Lockdown Checklist

- ✓ CRYPTOZ class is ACTIVE
- ✓ CRYPTOZ class is RACLISTed
- ✓ CRYPTOZ class is GENERIC
- ✓ To enforce that all keys in PKCS #11 tokens must be secure keys, define a CLEARKEY.* or CLEARKEY.** profile with UACC(NONE)

Enable FIPS mode to operate in compliance with FIPS

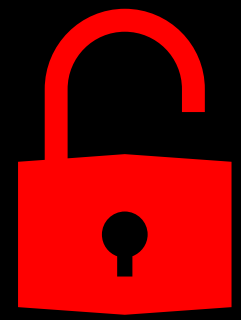
FIPS mode can be enabled to ensure that PKCS #11 requests are run in a FIPS 140-2 compliant fashion (i.e. algorithm and key size restrictions are enforced). When enabled known answer tests (KAT) are run.

- FIPSMODE(YES...): Indicates ICSF is to operate in FIPS mode.
- FIPSMODE(COMPAT...): Indicates ICSF is to operate in FIPS mode for applications that are not FIPS exempt.
- FIPSMODE(NO...): Indicates ICSF is to operate in FIPS no enforcement mode (i.e. FIPS on-demand). FIPS may be enabled by an application by setting CKA_IBM_FIPS140=TRUE on the request.

- FAIL(YES): ICSF will terminate abnormally if there is a failure in any FIPS tests.
- FAIL(NO): ICSF initializes even if tests fail. All PKCS #11 callable services will be unavailable if KAT failed.

If the FIPSMODE option is not specified, FIPSMODE(NO,FAIL(NO)) is the default.

- Algorithms and keys below 80 bits of strength
- RSA private keys without padding
- ...

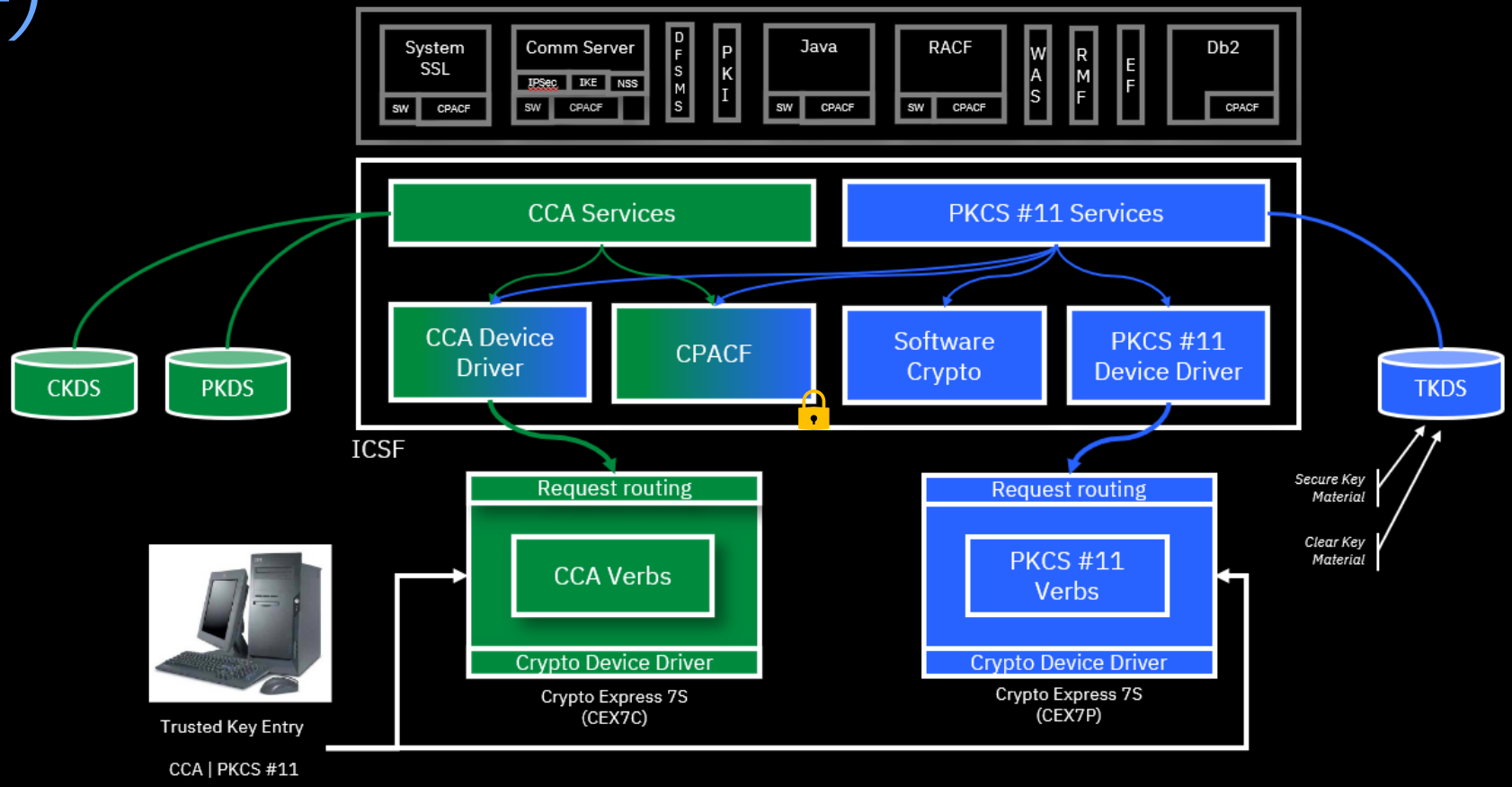


Lockdown Checklist

- ✓ CRYPTOZ class is ACTIVE
- ✓ CRYPTOZ class is RACLISTed
- ✓ CRYPTOZ class is GENERIC
- ✓ To enforce that all PKCS #11 tokens run in FIPS mode
  - ✓ Enable FIPSMODE(YES...) or
  - ✓ Enable FIPSMODE(COMPAT...) and define a backstop FIPSEXEMPT.* or FIPSEXEMPT.** profile with UACC(NONE)

General Hardening

IBM Z Central Processor Assist for Cryptographic Function (CPACF)



Ensure CPACF is enabled

CP Assist for Cryptographic Functions is implemented on every processor. SHA-1, SHA-2, and SHA-3 secure hashing and SHAKE extendable output functions are directly available to application programs.

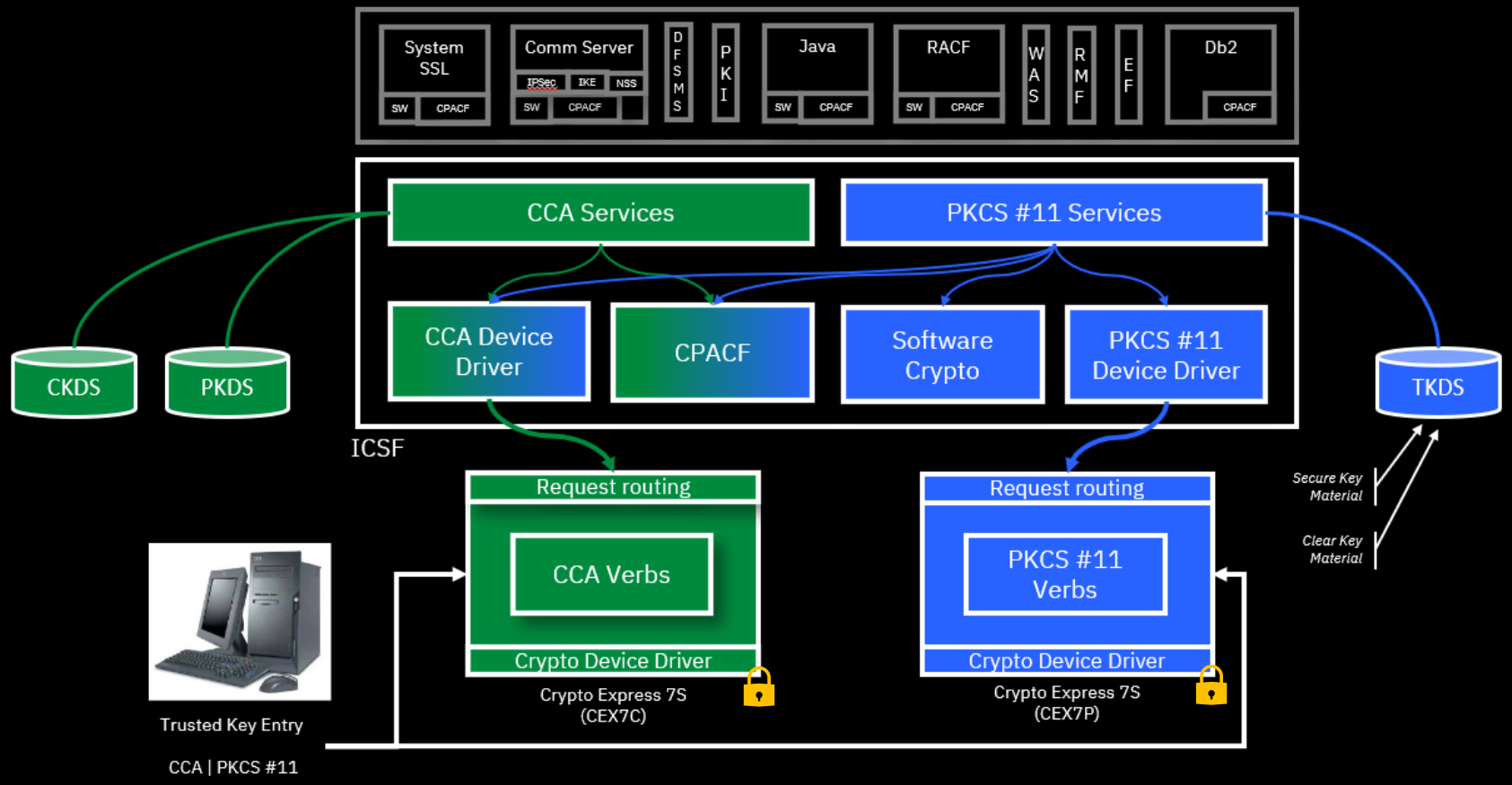
Feature code 3863, CP Assist for Cryptographic Functions (CPACF) DES/TDES Enablement - enables DES, TDES, and AES instructions on all CPs. *For z15, this feature also includes clear key ECC algorithm for NIST and Edwards curves.*

Feature code 3863, CP Assist for Cryptographic Functions (CPACF) is not enabled by default.

Lockdown Checklist (short & sweet!)

✓ Feature code 3863 is enabled

Hardening
IBM Crypto Express HSMs

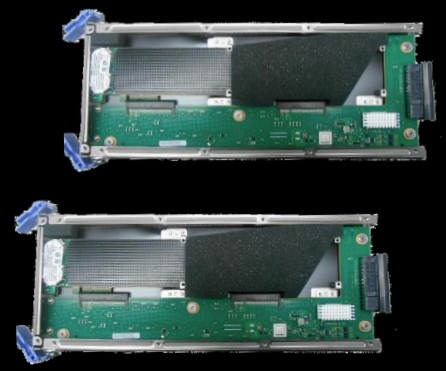


Redundant Crypto Express HSMs

Order at least 2 HSMs of each type needed in your environment (e.g. CCA Accelerator, CCA Coprocessor, EP11 Coprocessor) for redundancy. **Note:** A minimum of 2 features is required per z15.

If one HSM needs to be taken offline (for example, MCL upgrade), the second card loaded with the same secrets (e.g. Master Key) as the first would be able to handle the same requests.

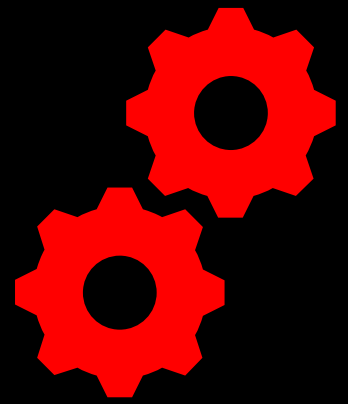
FC 0898 (2-port)



FC 0899 (1-port)

IBM z15 can support up to 60 Crypto Express 7S HSMs and z15 Model T02 can support up to 40 Crypto Express 7S HSMs.

Feature code 3863, CPACF, is a pre-requisite for Crypto Express enablement.



Lockdown Checklist

- ✓ Feature code 3863 is enabled
- ✓ Feature code 0899 and/or 0898 is enabled
- ✓ Two or more Crypto Express HSMs are available

Three or more HSMs for increased throughput?

Load (and backup) master keys

Master Keys should be separated into two or more key parts owned by different key officers to ensure separate of duties.

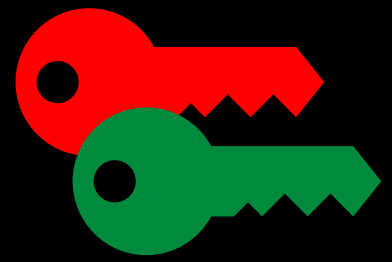
To reconstruct the complete Master Key, all key parts must be loaded (in any order). Therefore, no one person has access to the Master Key in its entirety.

Master Keys must be securely stored yet accessible in situations such as hardware upgrades and disaster recovery.

- TKE workstation users generate and store Master Key parts onto smart cards. Master Key parts are only accessible with possession of the smart card (something you have) and knowledge of the smart card pin (something you know).
- Master Key Entry panels and PPINIT users must carefully consider how to securely store Master Keys.

Crypto Express HSMs support one or more master key parts.

The TKE Workstation requires two or more master key parts.



Secure? It depends!

Lockdown Checklist

- ✓ Master keys are separated into two or more key parts
- ✓ Each master key part is owned by a different key officer
- ✓ All master key parts are securely backed up for disaster recovery
- ✓ Master keys are loaded / available on redundant Crypto Express HSMs

Review and/or set access control points

To execute services on the coprocessor, access control points must be enabled for each service in the domain role. The access control points available depend on the coprocessor you are using.

Note: *A TKE workstation is required to modify access control points.*

Trusted Key Entry

A new or a zeroized coprocessor (or domain) comes with an initial set of access control points (ACPs) that are enabled by default.

Secure? It depends!

Lockdown Checklist

- ✓ Review ACPs for
  - ICSF callable services (see the ICSF APG)
  - ICSF utilities (see the ICSF Admin Guide)
  - PKCS #11 (see the Writing PKCS #11 Guide)
- ✓ Use the TKE Workstation to modify the ACPs

Questions?

Stay tuned!
More to come!