Keys, Keys, Keys … Cryptographic Keys
Session 53820

Julie Bergh
Cyber Security Architect
Sirius a CDW Company
Julie.bergh@siriuscom.com
Julie.bergh@cdw.com

Abstract

- ICSF, CSFSERV, CSFKEY, CRYPTOZ
- Is this alphabet soup?
- What is this about?
- This session gives a basic understanding of key setup in RACF and the roles and responsibilities for setting this up.
  - References to ACF2 and Top Secret will also be made

Agenda

- Basics - Cryptography
- KEYS
- ICSF
- CSFSERV, CSFKEY, CRYPTOZ
- Auditing
- This session gives a basic understanding of key setup in RACF
  - References to ACF2 and Top Secret will also be made.

Basics - Cryptography

What is cryptography?

Cryptography is defined as the practice and study of techniques for secure communication in the presence of third parties (i.e. adversaries).

- Confidentiality – Preventing the disclosure of information to unauthorized individuals.
    - **Encrypt:** Convert clear text to cipher text
    - **Decrypt:** Convert cipher text to clear text

What is cryptography?

- Integrity – Maintaining and assuring the accuracy and consistency of data.
    - **Hash:** Translate clear text to a fixed length hash value

      Example (32-byte hash):

      1025 4AD0 04D2 C7D5 77EA ADA0 E4C8 B76F A290 2F7C D03B F03E B527 A045 E200 238F
    - **Sign:** Hash the clear text and encrypt the hash with a private key
    - **Verify:** Hash the clear text then decrypt the sender's hash using the sender's public key and compare the hash values
- Authentication – Verifying the identity of a party.
- Non-repudiation – Assuring that a party cannot deny that they created a message.

Keys – Just to name a few

Asymmetric

Public key

Symmetric

Secret

Public Key Cryptography

Standard #11 (PKCS #11)

DES Key

AES Key

CKDS

PKDS

TKDS

KEK

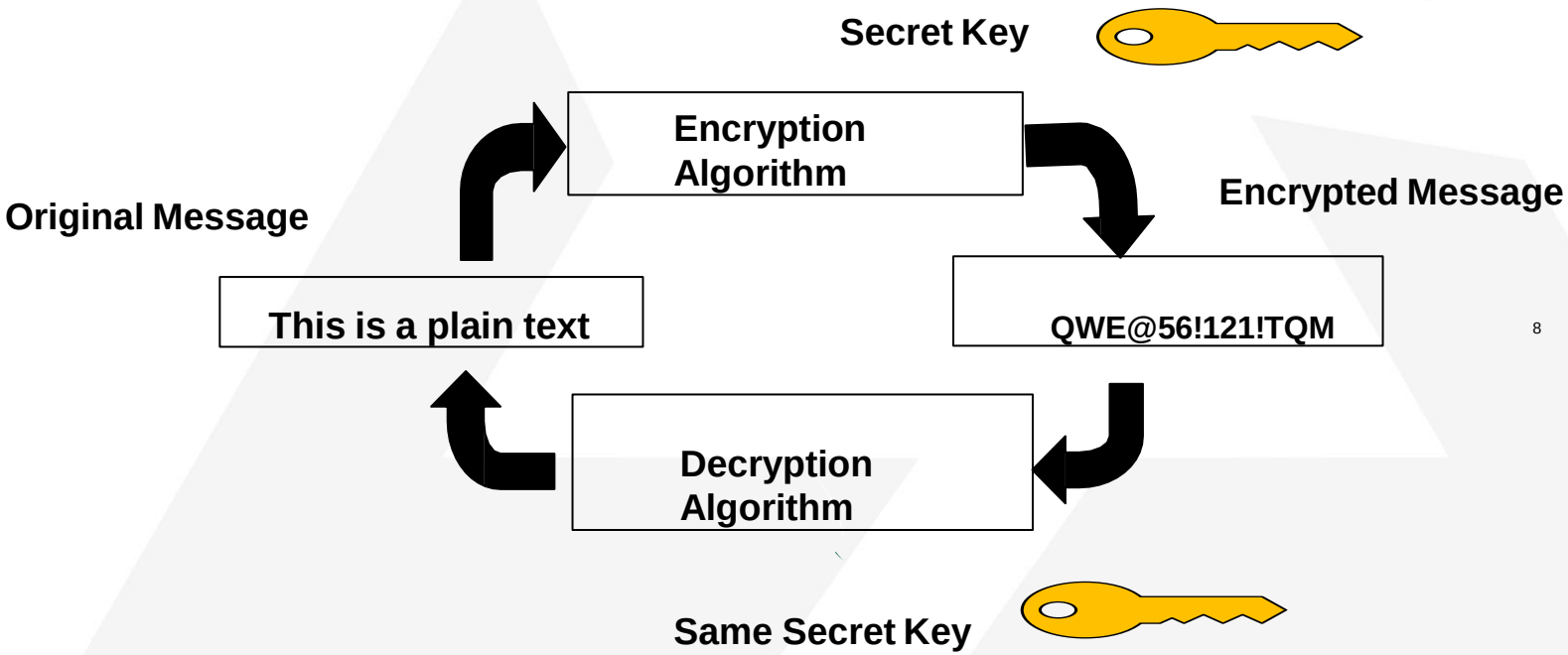
Clear key

Master Keys

Operational Keys

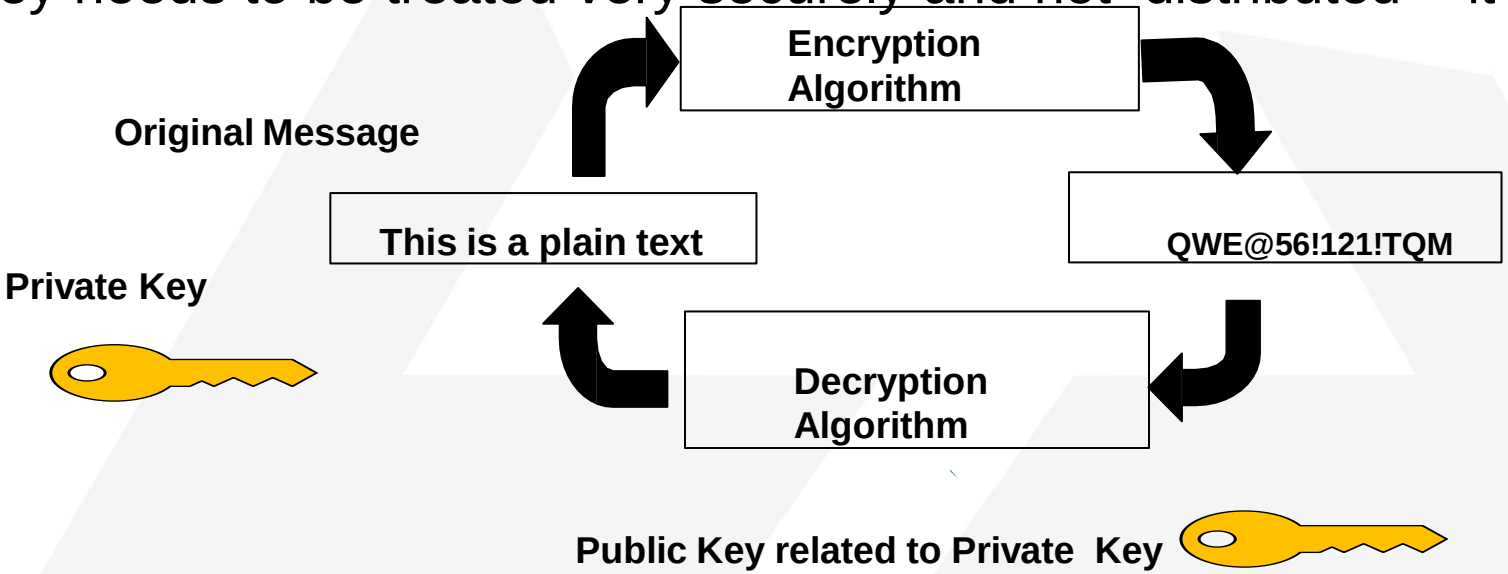
Symmetric Encryption

- Same key used for both encryption and decryption
- Provide data confidentiality
- Fast, used for bulk encryption/decryption
- Securely sharing and exchanging the key between both parties is a major issue
- Common algorithms: DES, Triple DES, AES



Asymmetric Encryption

- Public / private key pairs - 2 different keys
- A public key and a related private key are numerically associated with each other.
- Provide data confidentiality, integrity and non repudiation Data encrypted/signed using one of the keys (private) may only be decrypted/verified using the other key public)
- Public key is freely distributed to others, private key is securely kept by the owner. Only one public key can decrypt a private key
- Common algorithms: RSA, DSA, ECC
- Private key needs to be treated very securely and not distributed – it is keeping the secret



ICSF

ICSF

- Key Encrypting Keys (KEKs) are keys that protect (e.g. encrypt, wrap) other keys

Master Keys

Master keys are used only to encipher and decipher keys. They are used to protect sensitive cryptographic keys that are active on your system.

Master keys are stored in secure, tamper responding hardware.

Master key encrypted keys are considered secure keys.

Master keys should be changed periodically.

All master keys are optional. Secure keys are only supported when their associated master key is active.

Operational Keys

Operational keys are used in various cryptographic operations (e.g. encryption).

Operational keys may be stored in a key store (e.g. data set, file, database) or returned back to the caller.

Operational keys may be clear, secure or protected.

Symmetric KEKs

Encrypt symmetric keys with another symmetric key.

Asymmetric KEKs

Encrypt symmetric keys with RSA public keys

Use ECC key pairs to derive a symmetric key. Use the derived symmetric key to encrypt another symmetric key.

ICSF – Integrated Cryptographic Service Facility

- ICSF is a software component of z/OS that provides secure, high-speed cryptographic services in the z/OS environment. ICSF provides the following components:
  - An API for cryptographic operations
  - Key management and keystores (CKDS, PKDS, and TKDS) for cryptographic key material
  - Access to high-speed hardware Cryptographic Coprocessors and Accelerators, and CP Assist for Cryptographic Function (CPACF)

ICSF supports these two main types of cryptographic processes:

- Symmetric algorithms, in which the same key value is used in both the encryption and decryption calculations
- Asymmetric algorithms, in which a different key is used in the decryption calculation than was used in the encryption calculation

ICSF – Integrated Cryptographic Service Facility

ICSF groups the cryptographic keys into these categories:

- Master keys: Common Cryptographic Architecture (CCA) and PKCS #11
- Operational keys: CCA, PKCS #11, and regional cryptographic server.

ICSF uses master keys to protect other keys. Keys are active on a system only when they are encrypted under a master key variant, so the master key protects all keys that are used on the system. A key is in operational form when it has been encrypted under a master key variant. A key must be in operational form to be used with the cryptographic features.

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

z/OS ICSF Key Data Sets

- CKDS - Cryptographic Key Data Set (CKDS) stores operational DES, AES, and HMAC keys of all types. It contains an entry for each key.
  - An installation is not required to define a CKDS.
  - However, when a CKDS is not defined, secure CCA symmetric key functions are unavailable and ICSF cannot be used to manage CCA symmetric key tokens.
- PKDS – Public Key Data Set - Public Key Algorithm (ECC and RSA) public and private keys and trusted blocks can be stored in the PKDS.
  - Applications can use the dynamic PKDS callable services to create, write, read, and delete PKDS records. An installation is not required to define a PKDS.
  - However, when a PKDS is not defined, secure CCA asymmetric key functions are unavailable and ICSF cannot be used to manage CCA asymmetric key tokens.

ICSF – Integrated Cryptographic Service Facility

RACDCERT ADD(*data-set-name*)
[ID(*certificate-owner*) | SITE | CERTAUTH]
[TRUST | NOTRUST | HIGHTRUST]
[WITHLABEL(*'label-name'*)]
[PASSWORD(*'pkcs12-password'*)]
[PKDS[(*pkds-label* | *)]

z/OS ICSF Key Data Sets

- TKDS – Token Data Set - Clear and secure PKCS #11 objects can be stored in the TKDS.
  - Applications can use the PKCS #11 callable services to create, write, read, and delete PKCS #11 tokens and objects.

- Tokens are containers that hold digital certificates and keys. z/OS supports both clear and secure keys in the PKCS #11 tokens that are provided and managed by ICSF. RACF can be used to define define and manage certain certificate objects in a token (certificates, public keys, and private keys).

- RACDCERT ADDTOKEN(*token-name*)

ICSF Display

```
CSFM668I 09.49.31 ICSF OPTIONS 112
  SYSNAME = TST1          ICSF LEVEL = HCR77D1
    LATEST ICSF CODE CHANGE = 08/16/21
    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
    AUDITKEYLIFECKDS: Audit CCA symmetric key lifecycle events
      SYSNAME   LABEL     TOKEN
      TST1        Yes       Yes
    AUDITKEYLIFEPKDS: Audit CCA asymmetric key lifecycle events
      SYSNAME   LABEL     TOKEN
      TST1        Yes       Yes
    AUDITKEYLIFETKDS: Audit PKCS #11 key lifecycle events
      SYSNAME   TOKOBJ    SESSOBJ
      TST1        No        No
    AUDITKEYUSGCKDS: Audit CCA symmetric key usage events
      SYSNAME   LABEL     TOKEN     Interval Days/HH.MM.SS
      TST1        Yes       Yes                001/00.00.00
    AUDITKEYUSGPKDS: Audit CCA asymmetric key usage events
      SYSNAME   LABEL     TOKEN     Interval Days/HH.MM.SS
      TST1        Yes       Yes                001/00.00.00
    AUDITPKCS11USG: Audit PKCS #11 usage events
```

z/OS ICSF – Protecting Resources

- The CSFSERV class controls access to CCA and PKCS #11 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.

z/OS ICSF – Protecting Resources

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”.

FACILITY class - data set encryption can be restricted to only security Administrators - STGADMIN profiles

- XFACILIT class- Enables granular key label access

ICSF – Integrated Cryptographic Service Facility

- CRYPTOZ - Controls access to PKCS #11 tokens in the TKDS.
- CSFKEYS - Controls access to ICSF cryptographic keys in the CKDS and the PKDS.
  - GCSFKEYS - Resource group class for the CSFKEYS class.
- XCSFKEY Controls the exportation of ICSF cryptographic keys. Protects the symmetric key export of keys in the CKDS.
  - GXCSFKEY - Resource group class for the XCSFKEY class.
- CSFSERV - Controls access to ICSF cryptographic services protects the APIs and the panels.

ICSF – Integrated Cryptographic Service Facility

CRYPTOZ - The CRYPTOZ class controls access to and defines a policy for PKCS#11 tokens in the TKDS. PKCS#11 token access control is unique in that the tokens have different access levels and a differentiation between standard users and security officers. For each token, there are two resources in the CRYPTOZ class for controlling access to tokens:

- The CRYPTOZ class does not grant access to the resource if there is no profile

ICSF – Integrated Cryptographic Service Facility

- The resource USER.token-name controls the access of the User role to the token.
- The resource SO.token-name controls the access of the Security Officer (SO) role to the token.
- The CLEARKEY.token-name resource within the CRYPTOZ class controls the ICSF policy for creating a clear key versus a secure key.

- Tokens can be added using:
  - RACDCERT ADDTOKEN(*token-name*)
  - ICSF panels

```
racdcert addtoken(MFAtoken1)
 Insufficient authority to access token MFATOKEN1.


ICH408I USER(          ) GROUP(          ) NAME(JULES
  SO.MFATOKEN1 CL(CRYPTOZ )
  INSUFFICIENT ACCESS AUTHORITY
  FROM SO.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
READY
racdcert addtoken(mfatoken1)
 Insufficient authority to access token MFATOKEN1.
READY

ICH408I USER(          ) GROUP(          ) NAME(JULES
  CSF1TRC CL(CSFSERV )
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

ICSF – Integrated Cryptographic Service Facility

CSFKEYS

The CSFKEYS class controls access to the cryptographic keys in the CKDS and PKDS. A UACC of NONE is preferred for customer installations. Access to specific keys or tokens should be granted only to those processes and people who need access.

Note: The CSFKEYS class grants access to the key if there is no profile

Define appropriate profiles in the CSFKEYS class:
RDEFINE CSFKEYS label UACC(NONE)
other-optional-operands
where *label* is the label by which the key is defined in the CKDS or PKDS. Note that if an application uses a token instead of a key label, no authorization checking is done on the use of the key.

READ authority is the default authority for access to PKDS and CKDS labels for all usage.

ICSF – Integrated Cryptographic Service Facility

XCSFKEY

The XCSFKEY class controls the ability to export a symmetric key with the Symmetric Key Export (CSNDSYX, CSNFSYX, or CSNDSXD) callable service.

- GXCSFKEY - Resource group class for the XCSFKEY class

By enabling the Symmetric Key Label Export control for AES or DES keys, and creating profiles in the XCSFKEY resource class, you can increase the level of access authority that is needed to export AES or DES keys without increasing the level of authority that is needed to access the keys for other operations.

The XCSFKEY class does not grant access to the resource if there is no profile

The XCSFKEY class controls authorization checks when the symmetric key export services are called.

ICSF – Integrated Cryptographic Service Facility

CSFSERV

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

A UACC of NONE is preferred for customer installations. Access to cryptographic services and panels should be granted only to those processes and people who need access.

The CSFSERV class grants access to the service if there is no covering profile, but with the following exceptions:

– Key Data Set Update (CSFKDU and CSFKDU6)
– Key Data Set Record Retrieve (CSFRRT and CSFRRT6)

.

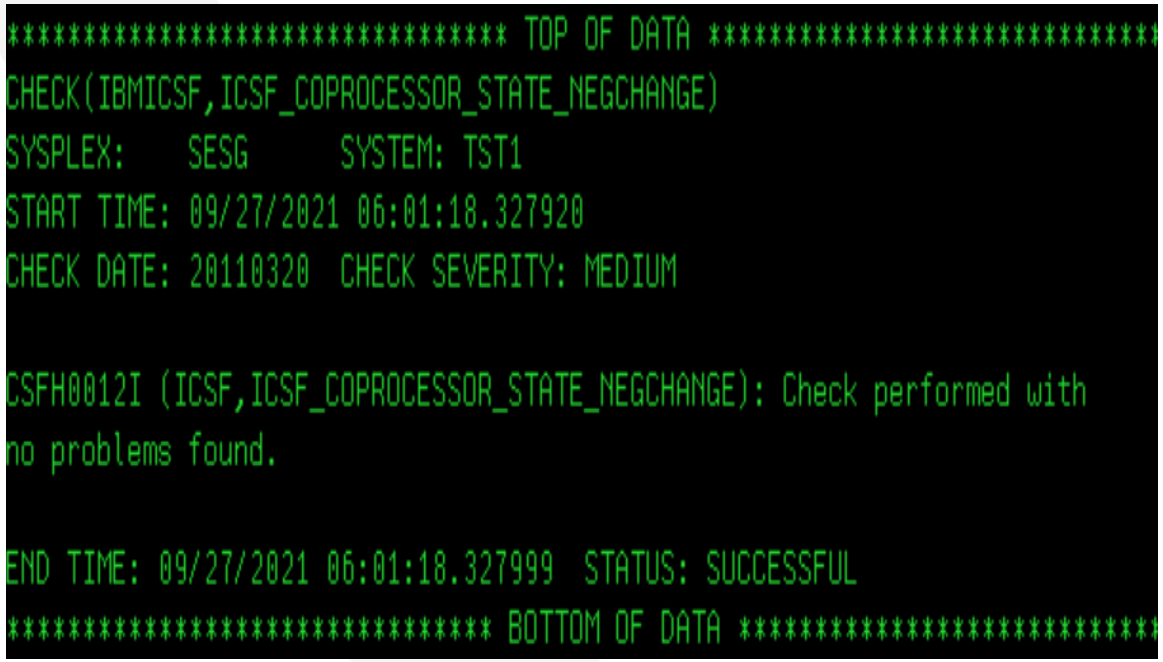
```
ICH408I USER([redacted] 1 ) 055
  CSFIQA CL(CSFSERV )
  WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
  FROM ** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
ICH408I USER([redacted] 1 ) 056
  CSF1GKP CL(CSFSERV )
  WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
  FROM ** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
ICH408I USER([redacted] 1 ) 057
  USER.MY.TEMP.TOKEN CL(CRYPTOZ )
  INSUFFICIENT ACCESS AUTHORITY
  FROM USER.** (G)
  ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
ICH408I USER([redacted] 1 ) 058
  SO.MY.TEMP.TOKEN CL(CRYPTOZ )
  INSUFFICIENT ACCESS AUTHORITY
  FROM SO.** (G)
  ACCESS INTENT(CONTROL)  ACCESS ALLOWED(UPDATE )
```

```
ICH408I USER([redacted]                                                    1 ) 334
  CSF1GKP CL(CSFSERV )
  WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
  FROM ** (G)
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
ICH408I USER([redacted]                                                    1 ) 335
  CSF1PKV CL(CSFSERV )
  WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
  FROM ** (G)
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
ICH408I USER([redacted]                                                    1 ) 336
  CSF1PKS CL(CSFSERV )
  WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
  FROM ** (G)
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
ICH408I USER([redacted]                                                    1 ) 337
  CSF1TRD CL(CSFSERV )
  WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
  FROM ** (G)
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
ICH408I USER([redacted]                                                    1 ) 338
  CSF1TRD CL(CSFSERV )
  WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
  FROM ** (G)
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
ICH408I USER([redacted]                                                    1 ) 339
  CSF1TRD CL(CSFSERV )
  WARNING: INSUFFICIENT AUTHORITY - TEMPORARY ACCESS ALLOWED
  FROM ** (G)
  ACCESS INTENT(READ    )  ACCESS ALLOWED(NONE    )
COMMANDS MAY BE ENTERED
```

ICSF & RACF Healthchecks

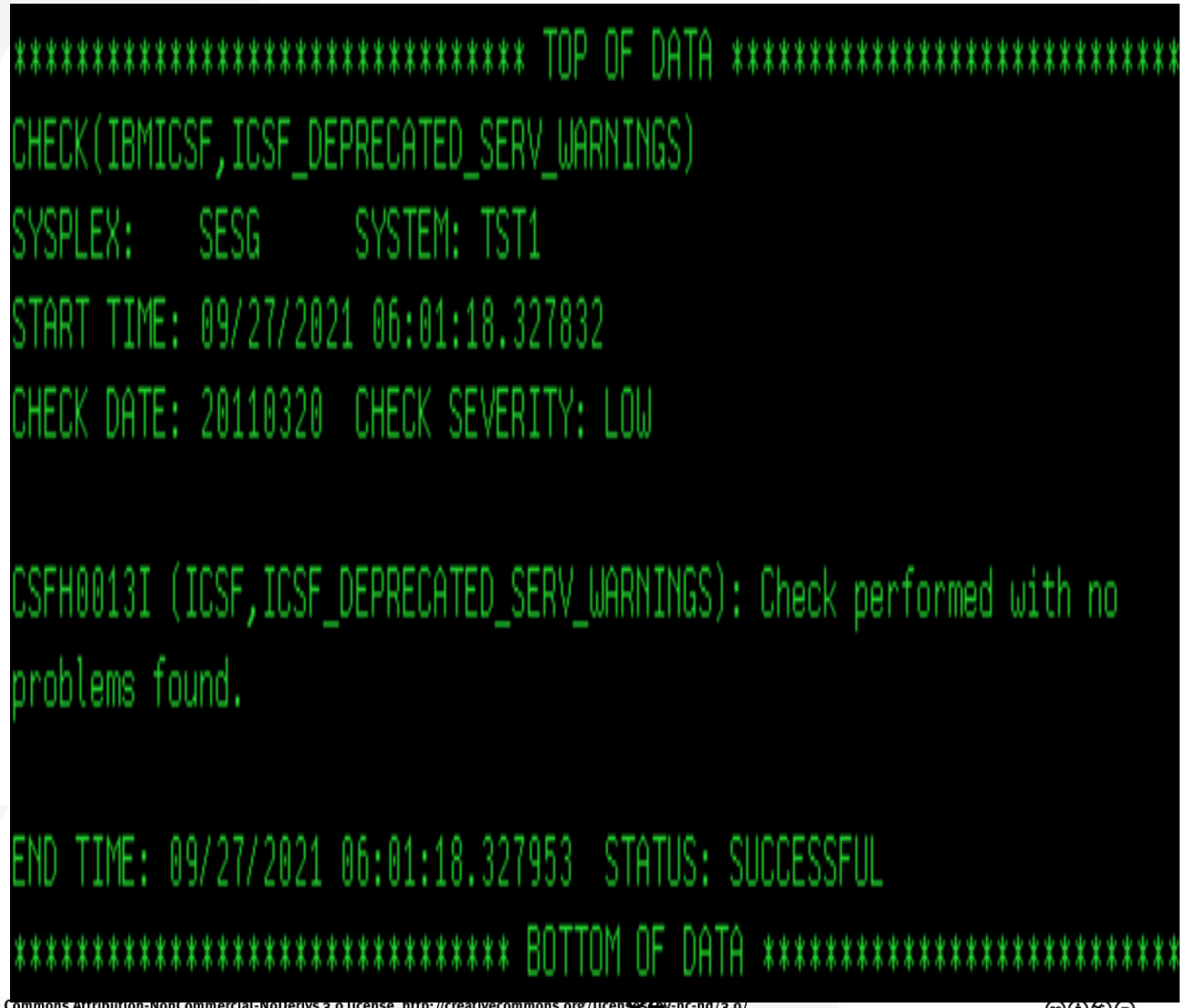
ICSF – Health Checks

ICSF_COPROCESSOR_STATE_NEGCHANGE - This is a status check. The check detects a degradation in the state of any cryptographic coprocessor or accelerator on the system. The check is activated during the initialization of ICSF. The check is performed on a daily basis.



ICSF – Health Checks

ICSF_DEPRECATED_SERV_WARNINGS - This is a deprecated services health check. The check detects the use of services that are no longer enhanced and are not recommended for continued use. The check is activated when the ICSF task is started and runs on a periodic (daily) basis.



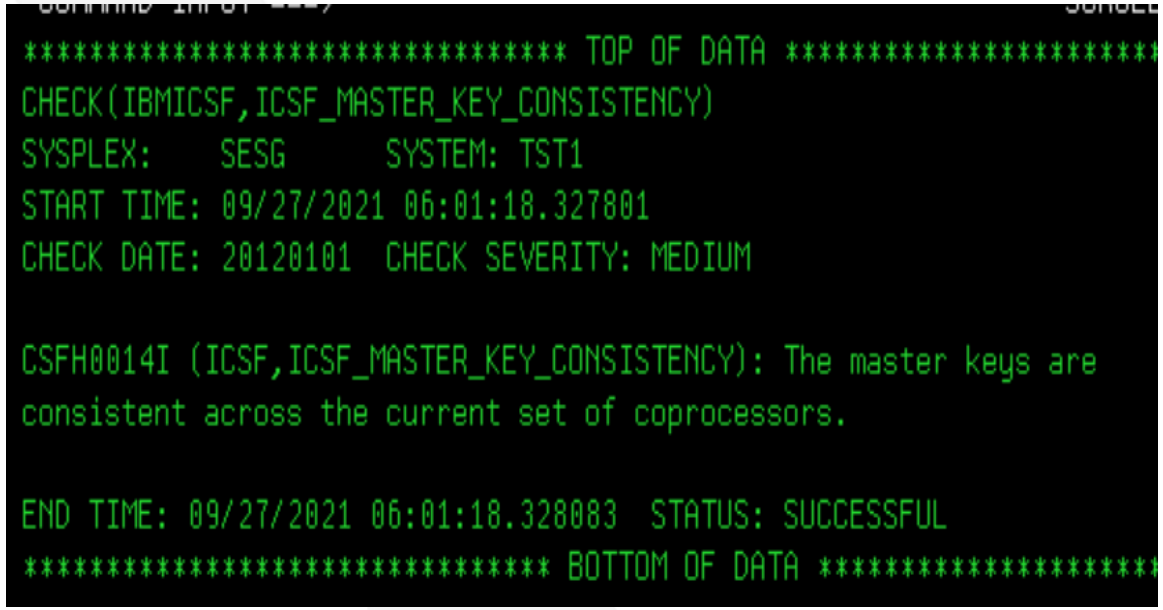
ICSF – Health Checks

ICSF_KEY_EXPIRATION - This is a status check. The check detects records in the active key data sets that have the key material validity end date metadata set and will expire within the specified interval. The active CKDS, PKDS, and TKDS are checked. The label of all records that will expire will be listed along with the expiration date.

```
********************************** TOP OF DATA ***********************
CHECK(IBMICSF,ICSF_KEY_EXPIRATION)
SYSPLEX:    SESG      SYSTEM: TST1
START TIME: 09/27/2021 06:01:18.327841
CHECK DATE: 20140101  CHECK SEVERITY: MEDIUM
CHECK PARM: DAYS(60)


CSFH0030I Cryptographic records expiring in 60 days.

Active CKDS: SYS1.SESG.CSF.CSFCKDSB
------------------------------------------------------------
None

Active PKDS: SYS1.SESG.CSF.CSFPKDSB
------------------------------------------------------------
None

CSFH0032I No KDS records will expire within the next 60 days.


END TIME: 09/27/2021 06:01:18.328318  STATUS: SUCCESSFUL
********************************** BOTTOM OF DATA ********************
```

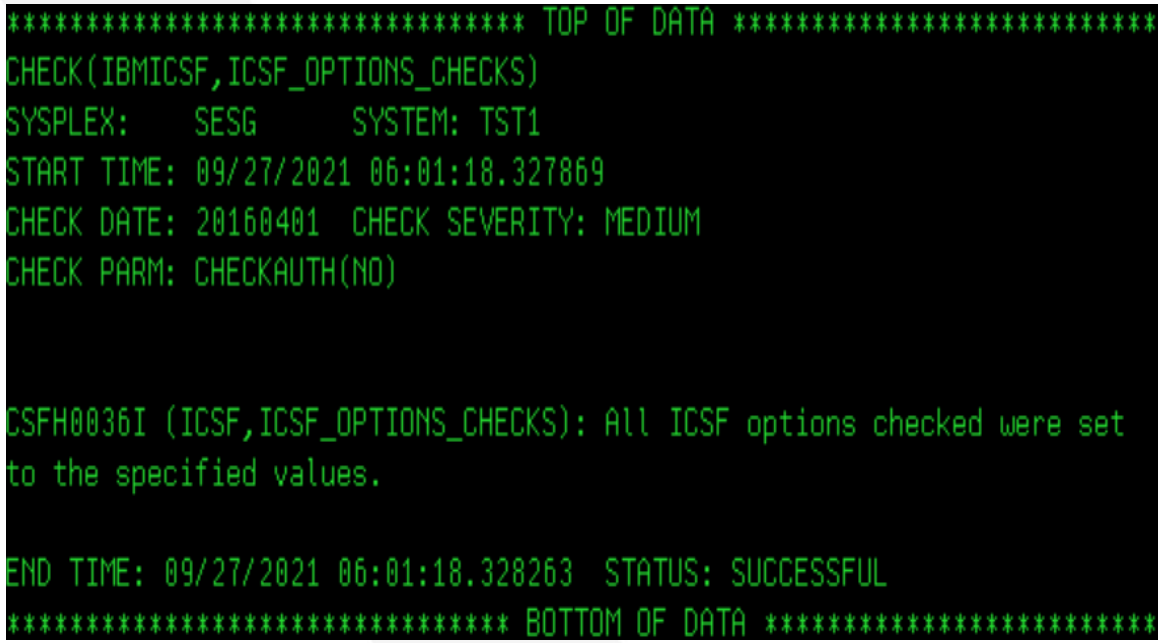
ICSF – Health Checks

ICSF_MASTER_KEY_CONSISTENCY - This is a master key health check. The check detects inconsistencies in the states of the coprocessor master keys. The check is activated when the ICSF task is started and runs on a periodic (daily) basis. The check determines when the state of a master key on at least one coprocessor is not in accord with the state on the other coprocessors.



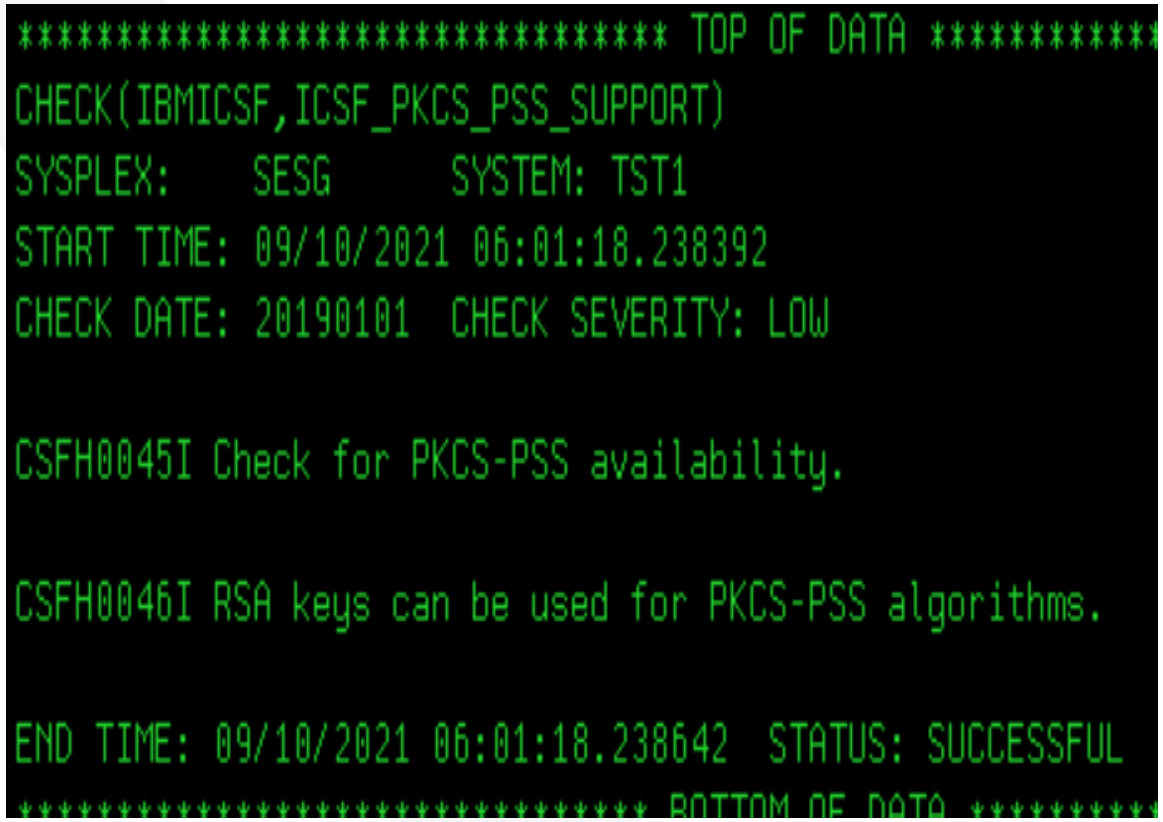
ICSF – Health Checks

ICSF_OPTIONS_CHECKS - This check examines the value of some of the ICSF installation options that can affect the performance of your ICSF applications. The check reports whether the current setting matches the option setting supplied in the parameter or the default setting. Any option setting not specified in the parameter will be checked for the default described for that option.



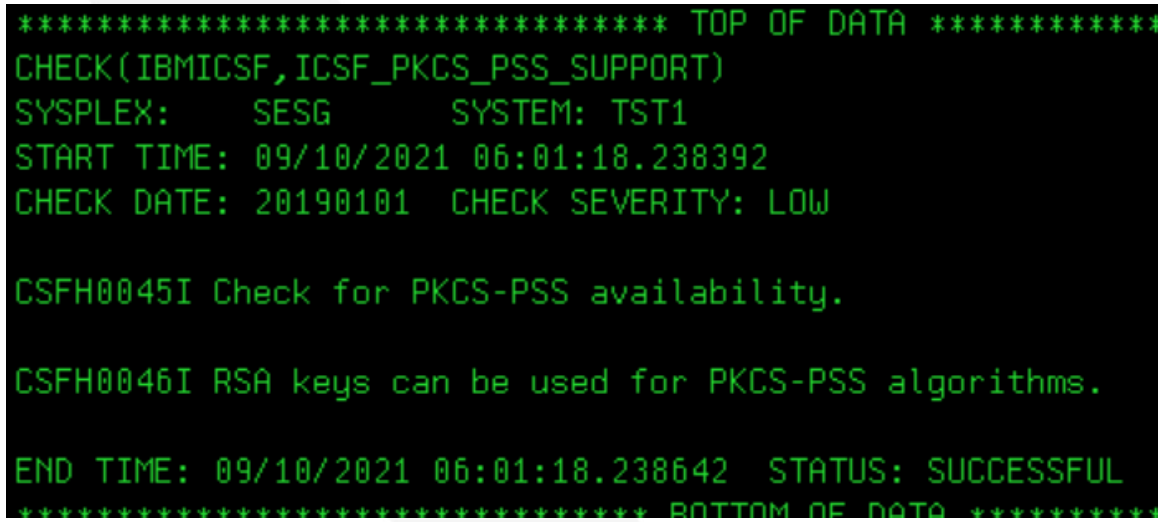
ICSF – Health Checks

ICSF_PKCS_PSS_SUPPORT - check detects whether the current hardware configuration supports PKCS-PSS algorithms. The ICSF administrator can use this check to determine if PKCS-PSS algorithms are available for operations, and if not, update their configuration with an active ECC master key or with a coprocessor of CCA 5.3 or above.



ICSF – Health Checks

ICSF_UNSUPPORTED_CCA_KEYS - The Cryptographic Coprocessor Feature (CCF) supported cryptographic algorithms that are now not supported on newer Crypto Express adapters. Keys for these unsupported algorithms and services may have been stored in the CKDS and PKDS. The ICSF_UNSUPPORTED_CCA_KEYS health check lists the label of records in the active CKDS and PKDS with keys that are not supported by the current cryptographic adapters. The ICSF administrator can use the list to determine if the records should be deleted. The administrator can archive the records if the format of the CKDS and PKDS is the common record format (KDSR).For the PKDS, all records with DSS public or private key tokens will be listed.For the CKDS, all records with DES ANSI X9.17 keys and DES data-translation keys (key type DATAXLAT) will be listed.



ICSF – Health Checks

ICSF_UNSUPPORTED_CCA_KEYS - The Cryptographic Coprocessor Feature (CCF) supported cryptographic algorithms that are now not supported on newer Crypto Express adapters. Keys for these unsupported algorithms and services may have been stored in the CKDS and PKDS. The

ICSF – Health Checks

The ICSF_WEAK_CCA_KEYS health check will list the labels of records in the active PKDS with keys that are considered cryptographically weak. The ICSF administrator can use the list to determine if the records should be replaced by cryptographically strong keys. The administrator can archive the records if the format of the PKDS is the common record format (KDSR).

ICSF – Health Checks

The ICSF migration checks are:

- ICSFMIG_DEPRECATED_SERV_WARNINGS - This is a migration check. If you are migrating to ICSF FMID HCR77A1 or a later release, you should run this check on your system before installing the new release of ICSF. The check detects the use of services which will not be supported in subsequent releases of ICSF. The check is not active when ICSF is started and must be activated to perform the check. Once activated the check will be performed on a daily basis.

- ICSFMIG_MASTER_KEY_CONSISTENCY - This is a migration check introduced in APAR OA39489. The check detects inconsistencies in the states of the cryptographic coprocessor master keys. The check is intended to warn the user of potential problems when migrating from pre-FMID HCR7780 releases of ICSF to the FMIDs HCR7780, HCR7790, or HCR77A0 releases of ICSF. The check is inactive when ICSF is started. When activated, it performs a one time check on the states of the coprocessor master keys. If a master key is not consistent across the available coprocessors, a problem condition is assumed and a health checker exception message is generated for the administrator's attention.

ICSF – Health Checks

The ICSF migration checks are:

- ICSFMIG7731_ICSF_RETAINED_RSAKEY - This is a migration check. The check detects the presence of retained keys on the cryptographic coprocessors. Retained keys will not be supported in subsequent releases of ICSF. Existing retained keys will become unusable.

- ICSFMIG77A1_COPROCESSOR_ACTIVE - This is a migration check. If you are migrating to ICSF FMID HCR77A1 or a later release, you should run this check on your system before installing the new release of ICSF.

- ICSFMIG77A1_TKDS_OBJECT - This is a migration check. If you are migrating to ICSF FMID HCR77A1 or a later release, you should run this check on your system before installing the new release of ICSF.

- ICSFMIG77A1_UNSUPPORTED_HW - This is a migration check. If you are migrating to ICSF FMID HCR77A1 or a later release, you should run this check on your system before installing the new release of ICSF.

ICSF – Health Checks

The following health checks require access to the listed CSFSERV service profiles:

- ICSF_KEY_EXPIRATION
  - CSFKDSL
  - CSFKDMR
- ICSFMIG7731_ICSF_RETAINED_RSAKEY
  - CSFRKL
- ICSF_UNSUPPORTED_CCA_KEYS
  - CSFKDSL
- ICSF_WEAK_CCA_KEYS
  - CSFKDSL

NOTE: If ICSF is running with CHECKAUTH(YES) specified in the options data set and the access to service through the CSFSERV SAF class is fully controlled, authority to specific services is required for some health checks

RACF – Health Checks

RACF_CSFKEYS_ACTIVE, for the CSFSERV resource class.

```
********************************** TOP OF DATA ************
CHECK(IBMRACF,RACF_CSFKEYS_ACTIVE)
SYSPLEX:    SESG      SYSTEM: TST1
START TIME: 09/27/2021 06:00:05.712263
CHECK DATE: 20140106  CHECK SEVERITY: MEDIUM
CHECK PARM: CSFKEYS


IRRH228I The class CSFKEYS is active.

END TIME: 09/27/2021 06:00:05.712503  STATUS: SUCCESSFUL
********************************** BOTTOM OF DATA ***********
```

RACF_CSFSERV_ACTIVE, for the CSFSERV resource class.

```
********************************** TOP OF DATA **********
CHECK(IBMRACF,RACF_CSFSERV_ACTIVE)
SYSPLEX:    SESG      SYSTEM: TST1
START TIME: 09/27/2021 06:00:05.712241
CHECK DATE: 20140106  CHECK SEVERITY: MEDIUM
CHECK PARM: CSFSERV


IRRH228I The class CSFSERV is active.

END TIME: 09/27/2021 06:00:05.712463  STATUS: SUCCESSFUL
********************************* BOTTOM OF DATA *********
```

RACF_SENSITIVE_RESOURCES, for the ICSF key data sets.

ICSF Dataset Report

S	Data Set Name	Vol	UACC	Warn	ID*	User
E	SYS1.SESG.CSF.CSFPKDSB	DCSYS1	Read	No	****	
E	SYS1.SESG.CSF.CSFCKDSB	DCSYS1	Read	No	****	

Auditing

CKDS - The audit records that are logged are dependent on the AUDITKEYLIFECKDS option setting.
PKDS - The audit records that are logged are dependent on the AUDITKEYLIFEPKDS option setting
TKDS - The audit records that are logged are dependent on the AUDITKEYLIFETKDS option setting.

RACF - CSFSERV, CSFKEYS, CRYPTOZ, XCSFKEYS

Auditing

Subtype 1 — is written whenever ICSF is started or the options refresh is performed.
Subtype 7 — is written when an operational key is imported from a coprocessor.
Subtype 8 — is written whenever the in-storage copy of the CKDS is refreshed.
Subtype 9 — is written whenever the CKDS is updated by a dynamic CKDS update service or the KDS Metadata write service or the CKDS KEYS utility.
Subtype 13 — is written whenever the PKDS is updated by a dynamic PKDS update service or the KDS Metadata write service or the PKDS KEYS utility.
Subtype 14 — is written when a clear master key part is entered on a cryptographic coprocessor.
Subtype 15 — is written whenever a retained key is created or deleted.
Subtype 16 — is written for each request and reply from calls to the CSFPCI service by TKE.
Subtype 18 — is written when the configuration of a coprocessor or accelerator changes.
Subtype 19 — no longer written.
Subtype 20 — is written periodically to record processing times for coprocessors or accelerators.

```
HCR77D1  -------------- Integrated Cryptographic Service Facility --------
OPTION ===>
System Name:  TST1                                Crypto Domain: 2
Enter the number of the desired option.

  1  COPROCESSOR MGMT -  Management of Cryptographic Coprocessors
  2  KDS MANAGEMENT   -  Master key set or change, KDS Processing
  3  OPSTAT           -  Installation options
  4  ADMINCNTL        -  Administrative Control Functions
  5  UTILITY          -  ICSF Utilities
  6  PPINIT           -  Pass Phrase Master Key/KDS Initialization
  7  TKE              -  TKE PKA Direct Key Load
  8  KGUP             -  Key Generator Utility processes
  9  UDX MGMT         -  Management of User Defined Extensions

     Licensed Materials - Property of IBM
     5650-ZOS Copyright IBM Corp. 1989, 2019.
     US Government Users Restricted Rights - Use, duplication or
     disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Press ENTER to go to the selected option.
Press END   to exit to the previous menu.
```

```
COMMAND ===>                                                  SCROLL ===> CSR
         Active CKDS: SYS1.SESG.CSF.CSFCKDSB
         Active PKDS: SYS1.SESG.CSF.CSFPKDSB
         Active TKDS:
  OPTION                                                      CURRENT VALUE
  ------                                                      -------------
  AUDITKEYLIFECKDS Audit CCA symmetric key lifecycle events   TOKEN(Y),LABEL(Y)
  AUDITKEYLIFEPKDS Audit CCA asymmetric key lifecycle events  TOKEN(Y),LABEL(Y)
  AUDITKEYLIFETKDS Audit PKCS #11 key lifecycle events        TOKO(N),SESSO(N)
  AUDITKEYUSGCKDS  Audit CCA symmetric key usage events       TOK(Y),LAB(Y),
                                                              INT(001/00.00.00)
  AUDITKEYUSGPKDS  Audit CCA asymmetric key usage events      TOK(Y),LAB(Y),
                                                              INT(001/00.00.00)
  AUDITPKCS11USG   Audit PKCS #11 usage events                TOKO(N),SESSO(N),
                                                              NOKEY(N),
                                                              INT(001/00.00.00)
  CHECKAUTH        RACF check authorized callers              NO
  CICSAUDIT        Audit CICS client identity                 NO
  COMPAT           Allow CUSP/PCF compatibility               NO
  COMPLIANCEWARN   Compliance Warn mode                       NOT SPECIFIED
  CTRACE           CTRACE parmlib used at ICSF startup        CTICSF00
  DEFAULTWRAP      Default symmetric key wrapping - internal  ORIGINAL
  DEFAULTWRAP      Default symmetric key wrapping - external  ORIGINAL
  DOMAIN           Current domain index or usage domain index 2
  FIPSMODE         Operate PKCS #11 in FIPS 140-2 mode        NO,FAIL(NO)
```

```
COMMAND ===>                                                  SCROLL ===> CSR
         Active CKDS: SYS1.SESG.CSF.CSFCKDSB
         Active PKDS: SYS1.SESG.CSF.CSFPKDSB
         Active TKDS:
  OPTION                                                      CURRENT VALUE
  ------                                                      -------------
  KDSREFDAYS         Number of days between reference updates  1
  KEYARCHMSG         JOBLOG message for archived key use       NO
  MASTERKCVLEN       Length of master key verification patterns ALL
  MAXSESSOBJECTS     Max non-auth pgm PKCS #11 session objects 65535
  REASONCODES        Source of callable services reason codes  ICSF
  RNGCACHE           Random Number Generate cache enabled      YES
  SERVICELIBS        Load ICSF using Service Data Sets         NO
  SERVSCSFMOD0       Data Set for Dynamic Service Update       IGNORED
  SERVSIEALNKE       Data Set for Dynamic Service Update       IGNORED
  SSM                Special Secure Mode enabled               YES
  STATS              Crypto Usage Statistics                   ENG,SRV,ALG
  STATSFILTERS       Crypto Usage Statistics Filters           NOTKUSERID(N)
  SYSPLEXCKDS        Sysplex consistency for CKDS updates      YES,FAIL(NO)
  SYSPLEXPKDS        Sysplex consistency for PKDS updates      YES,FAIL(NO)
  SYSPLEXTKDS        Sysplex consistency for TKDS updates      NO,FAIL(NO)
  USERPARM           User specified parameter data             USERPARM
  WAITLIST           Source of CICS Wait List if CICS installed default
********************************* Bottom of data ********************************
```

ICSF – Integrated Cryptographic Service Facility

CSFBRCK	CKDS Browser.
CSFCMK	Change master key utility, including the panel for a local change master key, the Coordinated KDS administration service, and CSFEUTIL.
CSFCONV	PCF CKDS to ICSF CKDS conversion utility.
CSFCRC	Coordinated KDS Administration.
CSFDKCS	Master key entry utility.
CSFEDC	Compatibility service for the PCF CIPHER macro.
CSFEMK	Compatibility service for the PCF EMK macro.
CSFGKC	Compatibility service for the PCF GENKEY macro.
CSFGKF	Generate key fingerprint. Required by KGUP if key lifecycle auditing is enabled.
CSFKGUP	Key generation utility program.
CSFOPKL	Operational key load.
CSFPCAD	Cryptographic processors management (activate/deactivate).
CSFPKDR	PKDS reencipher and PKDS refresh utilities.
CSFPMCI	Pass phrase master key/KDS initialization utility.

ICSF – Integrated Cryptographic Service Facility

CSFREFR	Refresh CKDS or PKDS utility, including the panels for a local refresh, the Coordinated KDS Administration service, and CSFEUTIL (CKDS) and CSFPUTIL (PKDS).
CSFRENC	Reencipher CKDS or PKDS utility, including the panels for a local refresh, the Coordinated KDS Administration service, and CSFEUTIL (CKDS) and CSFPUTIL (PKDS).
CSFRSWS	Administrative control functions utility (ENABLE).
CSFRWP	CKDS Conversion rewrap option
CSFRTC	Compatibility service for the CUSP or PCF RETKEY macro.
CSFSMK	Set master key utility.
CSFSSWS	Administrative control functions utility (DISABLE).
CSFUDM	User Defined Extensions (UDX) management functions.

ICSF – Integrated Cryptographic Service Facility

The CSFSERV class grants access to the service if there is no covering profile, but with the following exceptions:

– Key Data Set Update (CSFKDU and CSFKDU6)
– Key Data Set Record Retrieve (CSFRRT and CSFRRT6)

Does your RACF profile for CSFSERV look something like this

RDEF CSFSERV ** UACC(READ)

- ICSF, CSFSERV, CSFKEY, CRYPTOZ.
- Is this alphabet soup?
- What is this about?
- This session gives a basic understanding of key setup in RACF and the roles and responsibilities for setting this up.

Your feedback is important!

SHARE
EDUCATE • NETWORK • INFLUENCE

Submit a session evaluation for each session you attend:

www.share.org/evaluation

Digital Badges

Earn Your DevOps Wizard or Security Warrior Badge

Submit a Digital Badge form for each session attended:
https://forms.gle/mvqZPW7TNCwDCtmh9

Session ID: [53820]
Badge code: [gray902]