ICSF – Key Usage

NY RACF Users Group
15 March 2017

Roan Dawkins

dawkinsr@us.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

BigInsights	DFSMSdss	FICON*	IMS	RACF*	System z10*	zEnterprise*
BlueMix	DFSMShsm	GDPS*	Language Environment*	Rational*	Tivoli*	z/OS*
CICS*	DFSORT	HyperSwap	MQSeries*	Redbooks*	UrbanCode	zSecure
COGNOS*	DS6000*	IBM*	Parallel Sysplex*	REXX	WebSphere*	z Systems
DB2*	DS8000*	IBM (logo)*	PartnerWorld*	SmartCloud*	z13	z/VM*
DFSMSdfp						

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
TEALEAF is a registered trademark of Tealeaf, an IBM Company.
Windows Server and the Windows logo are trademarks of the Microsoft group of countries.
Worklight is a trademark or registered trademark of Worklight, an IBM Company.
UNIX is a registered trademark of The Open Group in the United States and other countries.
* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Agenda

- **Background**
- **A History of Key Usage**
- **Last Used Tracking**
- **Current Key Usage**
  - FIPS-compliant usage
  - Key fingerprints

Background

z/OS Integrated Cryptographic Services Facility (ICSF)

ICSF works with the hardware cryptographic features and the Security Server (RACF element) to provide secure, high-speed cryptographic services in the z/OS environment.

- ICSF provides the application programming interfaces by which applications request cryptographic services.
- ICSF callable services and programs can be used to generate, maintain, and manage keys that are used in the cryptographic functions.

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

IBM Common Cryptographic Architecture (CCA) for z/OS ICSF

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

z/OS ICSF Naming Conventions for CCA

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

CCA Functions & Algorithms

- Encrypt / Decrypt (AES, DES, DES3, RSA)
- Sign / Verify (RSA, ECC)
- MAC Generate / Verify (AES, DES, DES3)
- HMAC Generate / Verify (HMAC)
- Key Generate (AES, DES, DES3, HMAC)
- Key Pair Generate (RSA, ECC)
- Key Agreement (ECC, DH)
- One Way Hash
- Random Number Generate
- Key Import / Export
- TR-31 Block Import / Export

Financial Crypto

- PIN Generate / Verify / Translate
- PIN Encrypt
- Diversified Key Generate
- Derive Unique Key Per Transaction (DUKPT)
- CVV Generate / Verify
- Secure Messaging for Keys / Pins
- … And Many More!

PKCS#11 Cryptographic Token Interface Standard for z/OS ICSF

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

z/OS ICSF Naming Convention for PKCS#11

- CSFP* = PKCS#11 APIs

PKCS#11 Functions & Algorithms

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

Designed for portability and
FIPS/Common Criteria certification

Protecting Usage of Keys and Services

- ICSF maintains 3 keystores (aka key datasets or KDSs) for application use
  - Cryptographic Key Dataset (CKDS). Stores CCA symmetric keys.
  - Public Key Dataset (PKDS). Stores CCA asymmetric keys.
  - Token Key Dataset (TKDS). Stores PKCS#11 keys.

- The CSFSERV class controls access to ICSF callable services
- The CSFKEYS class controls access to cryptographic keys in the ICSF Key Data Sets (CKDS and PKDS).
- The CRYPTOZ class controls access to, and defines a policy for PKCS#11 token in the Token Key Data Set (TKDS).
- The XCSFKEY class controls the ability to export a symmetric key with the Symmetric Key Export callable services.

How Keys are Used

- ICSF services allow keys to be used in 2 forms
  - Via a key label [1]
  - Via a key token [2]
- A key label maps to a key token that is stored in a KDS. The key token is retrieved from the KDS and used.
- A key token is used directly

[1] – For the TKDS, we use the term “handle” in our publications. It is analogous to a label for CKDS and PKDS. We will use the term label generically in this presentation.

[2] – PKCS#11 keys may only be used by their label

A History of Key Usage

Auditing the SAF profile

- Whenever a key label is used, a SAF check is performed against the appropriate SAF class for the label being used
- If SAF auditing is enabled for the covering profile or the entire class, you may obtain SMF Type 80 records for the access attempt. The record would indicate who made the attempt against the key label.
- Drawbacks:
  - You would get one record for each access attempt. May lead to a lot of records.
  - Only applies to keys stored in a KDS
  - There is no information available about the key
  - There is no indication of how the key is being used.

Key Token Access Controls

- Several years ago, as a consequence of access controls of key token use, we gained some ability to know how keys outside the KDS were being used
- When enabled, whenever a key token is used, we check to see if it is contained in the KDS
- If it is, a SAF check is performed against the matching key label
- Expands the view of who is using which labels but maintains all the drawbacks as when using key labels

Last Used Tracking

Overview

- Customers were having difficulty understanding which keys in their KDS were still being used
- This was causing a key management headache and leading to extremely bloated KDSs because there was a healthy fear around removing keys
- In 2013 (HCR77A1), ICSF introduced support for tracking the last time a key was used
- The last used date is hardened to the KDS. It can be retrieved via the Key Dataset Metadata Read (CSFKDMR) callable service. This allows management of key labels based on when they were last used.
- This feature is enabled by default and controlled by the KDSREFDAYS option

Note: Requires KDS in the KDSR format

Current Key Usage

Overview

- The current key usage support records which user is using which key and how
- May audit events related to keys within the KDS as well as those outside the KDS
- Auditing is configurable by KDS key type
- Audit records contain additional information about the user and the key
- Audit logs are in the form of Type 82 SMF records
- Duplicate key usages are aggregated over an interval and logged as a single SMF record, reducing the volume of records
- Audit records contain a key fingerprint which acts as a pseudo-unique identifier for a key
- Only successful usages are logged

Configuration Options

- AUDITKEYUSGCKDS(LABEL(YES or NO), TOKEN(YES or NO), INTERVAL(interval))
- AUDITKEYUSGPKDS(LABEL(YES or NO), TOKEN(YES or NO), INTERVAL(interval))
- AUDITPKCS11USG(TOKENOBJ(YES or NO),SESSIONOBJ(YES or NO),INTERVAL(interval))

Notes:

- Default is NO for all options
- Default interval is 24 hours. Maximum interval is also 24 hours.

SMF Records

- **Subtype 44** – will be written for usage events related to symmetric CCA tokens.
- **Subtype 45** – will be written for usage events related to asymmetric CCA tokens.
- **Subtype 46** – will be written for usage events related to PKCS#11 objects.

Sample SMF Formatted Output

```
Subtype=002C CCA Symmetric Key Usage Event
Written for usage events related to symmetric CCA tokens
25 Jun 2015 13:08:28.35
   TME... 00482FD3 DTE... 0115176F SID... SP21    SSI... 00000000 STY... 002C
   STOD.. 06/25/2015 17:08:23.895178
   ETOD.. 06/25/2015 17:08:23.895179
   SRV... CSFSYX
   USGC.. 1
   LBL... @20150625.SUIMGKH.ICSF.SYSPLEX.AES.EXPORTER.1                    EXPORTER
   TOKFMT Variable
   KSEC.. Wrapped by MK
   KALG.. AES
   KTYP.. EXPORTER
   KUSGC.
          Key-usage field count: 4
            'FC00'x
            '0000'x
            'E000'x
            'F800'x
End User Identity...
   USRI.. IBMUSER
```

Sample Options Display Output

```
AUDITKEYUSGCKDS: Audit CCA symmetric key usage events
  SYSNAME   LABEL    TOKEN     Interval Days/HH.MM.SS
  SY1        Yes      Yes                000/01.00.00
AUDITKEYUSGPKDS: Audit CCA asymmetric key usage events
  SYSNAME   LABEL    TOKEN     Interval Days/HH.MM.SS
  SY1        Yes      Yes                000/01.00.00
AUDITPKCS11USG: Audit PKCS #11 usage events
  SYSNAME   TOKOBJ   SESSOBJ   NOKEY  Interval Days/HH.MM.SS
  SY1        Yes      Yes       Yes             000/01.00.00
```

Key Usage: FIPS-compliant usage

Background

- From a key usage standpoint, FIPS compliance only applies to usage of PKCS#11 keys
- Prior to HCR77A0, FIPS compliance was set at the system or application level
- FIPS On Demand processing (introduced in HCR77A0) allowed a system to, by default, not require FIPS compliance but instead allow applications to request FIPS-compliance on a request by request basis.
- This made it more difficult to demonstrate FIPS compliance

Overview

- Helps installations demonstrate compliance when running in FIPS On Demand mode [FIPSMODE(NO)]
- Information logged includes the current FIPSMODE configuration setting as well as whether the user or system requested FIPS compliance
- This is additional information that is included in the PKCS#11 key usage event if the request was successfully processed according to the FIPS 140-2 Level 1 standard. Otherwise, no FIPS-related information is logged.

Key Usage: Key Fingerprint

Overview

- Audit records use a key fingerprint to help distinguish one key value from another
- The key fingerprint is an identifier which is likely to be unique for differing key values but is not guaranteed. Different keys with the same key type and clear key value will always have the same key fingerprint.
- Key fingerprints are stored in the metadata section of KDSR records.
- For events that only involve a token, the key fingerprint is generated as needed.

CCA Key Tokens

Token	Method	Length
Fixed-length secure DES/AES	ENC-ZERO	3
Fixed-length clear DES/AES	ENC-ZERO	3
Variable-length external DES	ENC-ZERO	3
Variable-length AES	SHA-256	3
Variable-length HMAC	SHA2VP1	3
[1] RSA/ECC	SHA-1	3
DSS/Trusted blocks	Not supported	

[1] - The key fingerprint for asymmetric keys and certificates are calculated according to method 1 for generating the subject key identifier (SKI) of RFC 3279 (http://tools.ietf.org/html/rfc3279).

PKCS#11 Key Objects

Object	Method	Length
Secret - DES/AES/Blowfish	ENC-ZERO	3
Secret – RC4/Generic Secret	SHA-1	3
[1] Public/Private – RSA/DSA/DH/ECC	SHA-1	3
[1] Certificate	SHA-1	3
Data/Domain Parms/State	Not supported	

[1] - The key fingerprint for asymmetric keys and certificates are calculated according to method 1 for generating the subject key identifier (SKI) of RFC 3279 (http://tools.ietf.org/html/rfc3279).

Bibliography

- z/OS Cryptographic Services ICSF System Programmer's Guide
- z/OS Cryptographic Services ICSF Administrator's Guide
- z/OS Cryptographic Services ICSF Application Programmer's Guide
- z/OS Cryptographic Services ICSF Overview

Questions?

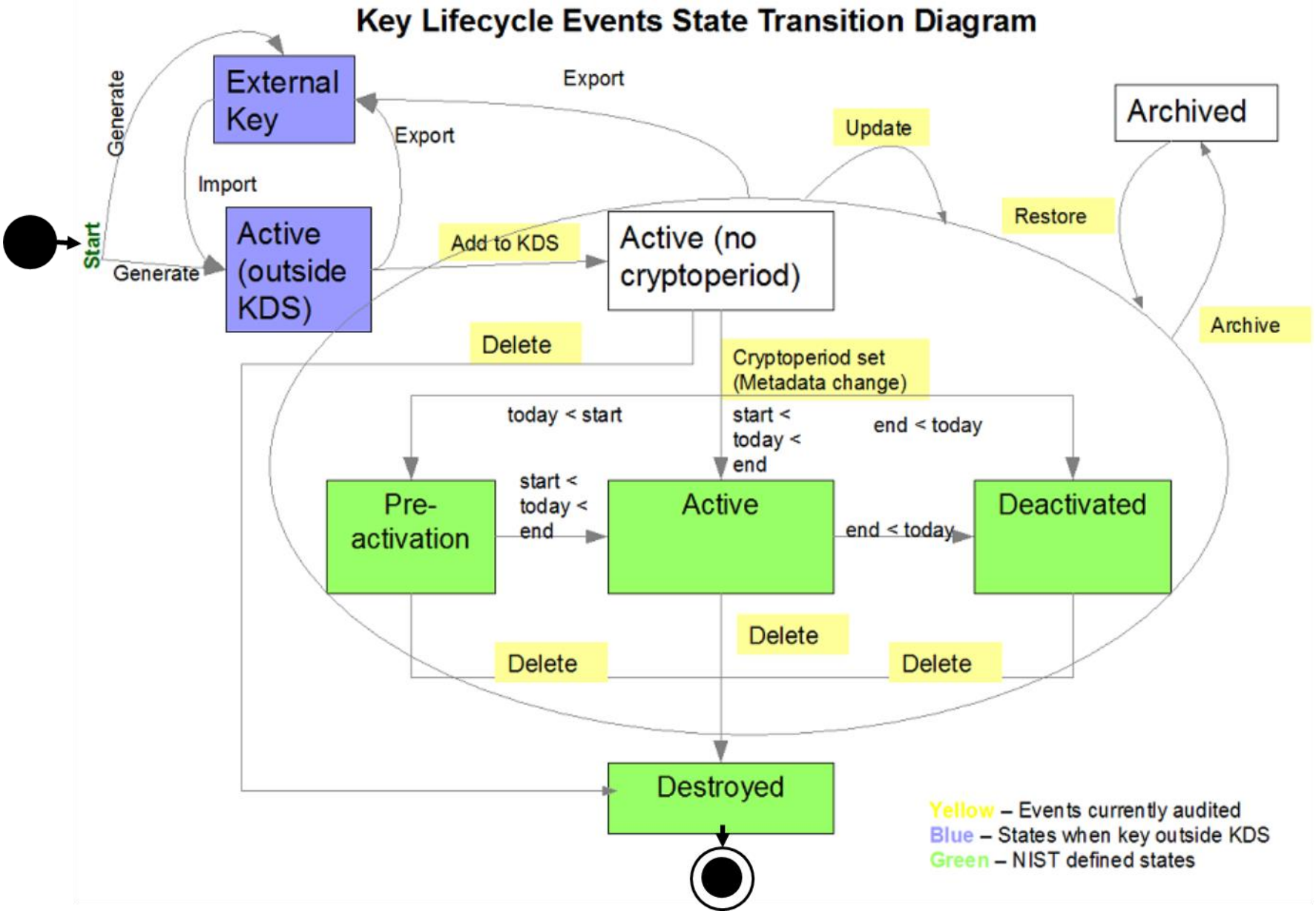
Backup

Key Lifecycle

Overview

Audits the complete lifecycle of key material from creation to disposal

Event	Pre-HCR77B0	HCR77B0	HCR77C0
Key Generated	Only if added to KDS	Only if added to KDS	Y
Key Added to KDS	Y	Y	Y
Key Updated in KDS	Y	Y	Y
Key Deleted	Y	Y	Y
Key Imported	TKE Op Key Load only	TKE Op Key Load only	Y
Key Exported	N	N	Y
Key Archived	N/A	Y	Y
Key Restored	N/A	Y	Y
Key Pre-Activated	N/A	N	Y
Key Activated	N/A	N	Y
Key Deactivated	N/A	N	Y
Key Metadata Changed	N/A	N	Y



Configuration Options

- AUDITKEYLIFECKDS(TOKEN(YES or NO),LABEL(YES or NO))
- AUDITKEYLIFEPKDS(TOKEN(YES or NO),LABEL(YES or NO))
- AUDITKEYLIFETKDS(TOKENOBJ(YES or NO),SESSIONOBJ(YES or NO))

Note: Default is NO for all options

SMF Records

- **Subtype 40** – will be written for lifecycle events related to symmetric CCA tokens. When auditing of labels is enabled, replaces subtype 9 (CKDS Update).
- **Subtype 41** – will be written for lifecycle events related to asymmetric CCA tokens. When auditing of labels is enabled, replaces subtype 13 (PKDS Update).
- **Subtype 42** – will be written for lifecycle events related to PKCS#11 objects. When auditing of token objects is enabled, replaces subtype 23 (TKDS Update).

Sample SMF Formatted Output

```
Subtype=40 CCA Symmetric Key Lifecycle Event
Written for lifecycle events related to symmetric CCA tokens
15 Sep 2015 17:18:37.57
   TME... 005F16CD DTE... 0115258F SID... SP21    SSI... 00000000 STY... 0028
   KEV... Key Exported
   SRV... CSFSXD
   KNM... AES#CIPHER$128BIT#2C.16
   TOKFMT Variable
   KSEC.. Wrapped by MK
   KALG.. AES
   KTYP.. CIPHER
   KUSGC. 02C0000000
          Key-usage field count: 2
            'C000'x
            '0000'x
```

Sample SMF Formatted Output (contd)

```
ICSF Server Identity...
   USRI.. SYSTASK
   GRPN.. SYS1
   JBN... CSFEC2
   RST... 10:28:53.69
   RSD... 14 Sep 2015
   SUID.. 4040404040404040
End User Identity...
   USRI.. ECHAN
   GRPN.. SYS1
   TRM... LOCALC11
   JBN... ECHAN
   RST... 10:00:18.27
   RSD... 15 Sep 2015
   SUID.. 4040404040404040
```

Sample Options Display Output

```
AUDITKEYLIFECKDS: Audit CCA symmetric key lifecycle events
  SYSNAME   LABEL    TOKEN
  SY1        Yes      Yes
AUDITKEYLIFEPKDS: Audit CCA asymmetric key lifecycle events
  SYSNAME   LABEL    TOKEN
  SY1        Yes      Yes
AUDITKEYLIFETKDS: Audit PKCS #11 key lifecycle events
  SYSNAME   TOKOBJ   SESSOBJ
  SY1        Yes      Yes
```