Vanguard Security Conference

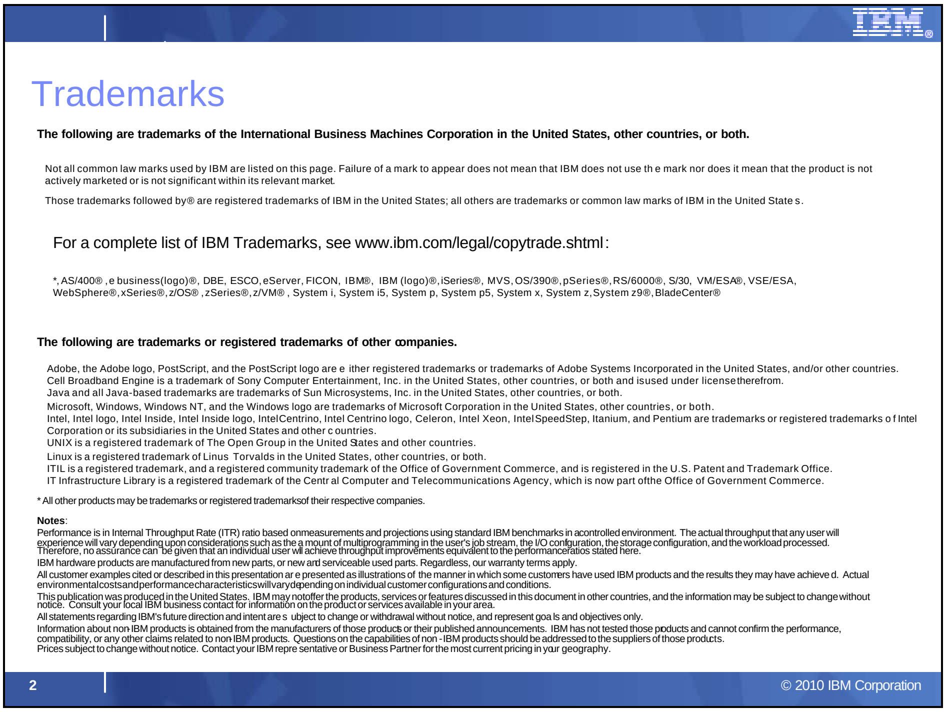
April 19 - 22 2010 – Las Vegas Nevada

Session Title: Cryptography on the zSeries Mainframe

Session ID: ACR12

Speaker Name: Ernest Nachtigall **CISSP;CISA**

THE Open GROUP Master Certified IT Specialist



Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use th e mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United State s.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400® , e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS® , zSeries®, z/VM® , System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are e ither 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 isused under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.
Intel, Intel logo, Intel Inside, Intel Inside logo, IntelCentrino, Intel Centrino logo, Celeron, Intel Xeon, IntelSpeedStep, Itanium, and Pentium are trademarks or registered trademarks o f Intel Corporation or its subsidiaries in the United States and other c ountries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
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.
IT Infrastructure Library is a registered trademark of the Centr al Computer and Telecommunications Agency, which is now part ofthe Office of Government Commerce.

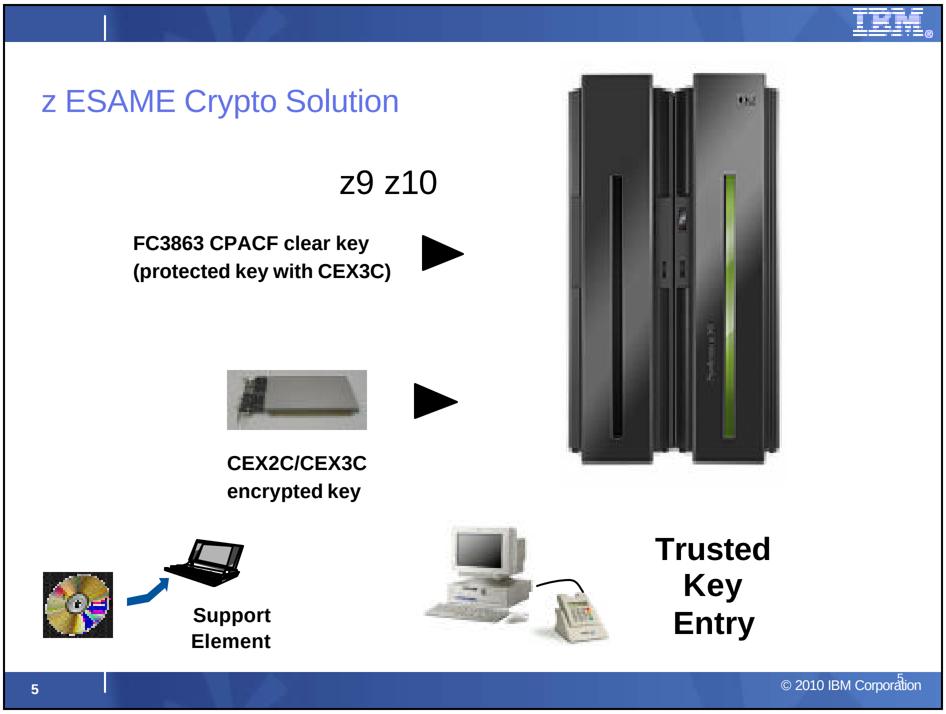
* All other products may be trademarks or registered trademarksof their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based onmeasurements and projections using standard IBM benchmarks in acontrolled environment. The actual throughput that any user will experience will vary depending upon considerations such as the a mount 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 performanceratios 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 ar e presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieve d. Actual environmentalcostsandperformancecharacteristicswillvarydepending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may notoffer 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 s ubject to change or withdrawal without notice, and represent goa ls 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 repre sentative or Business Partner for the most current pricing in your geography.

Agenda

- z10 System Overview
- z10 Cryptographic Hardware
- z10 Cryptographic Functionality
- Data Encryption
- SAF Implications

z9 z10 Overview



z ESAME Crypto Solution

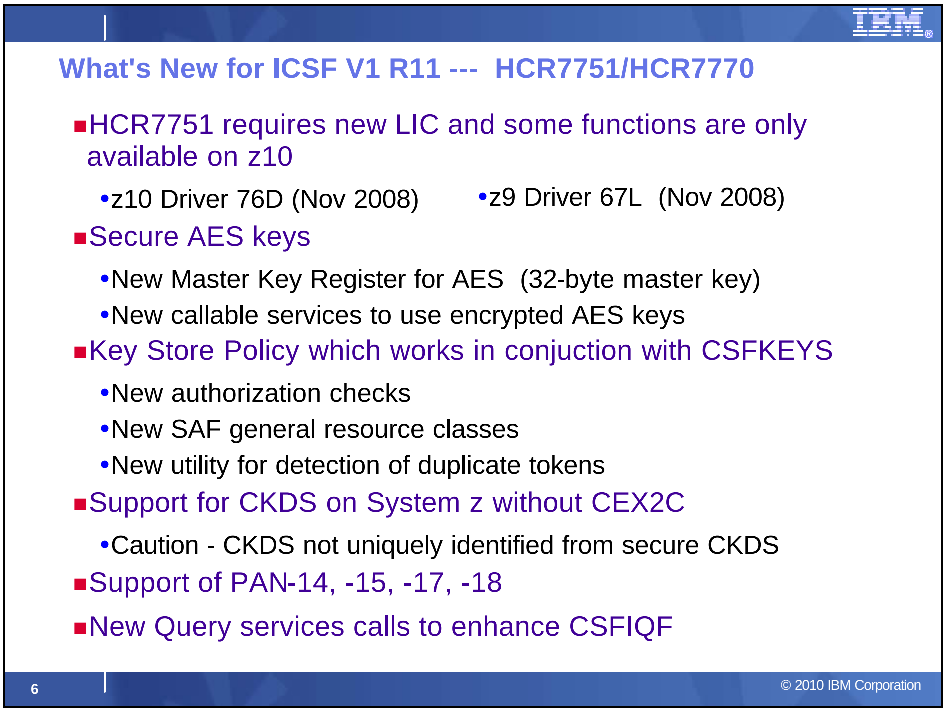
z9 z10

FC3863 CPACF clear key
(protected key with CEX3C)

CEX2C/CEX3C
encrypted key

Support
Element

Trusted
Key
Entry



What's New for ICSF V1 R11 --- HCR7751/HCR7770

- HCR7751 requires new LIC and some functions are only available on z10
  - z10 Driver 76D (Nov 2008)
  - z9 Driver 67L (Nov 2008)
- Secure AES keys
  - New Master Key Register for AES (32-byte master key)
  - New callable services to use encrypted AES keys
- Key Store Policy which works in conjuction with CSFKEYS
  - New authorization checks
  - New SAF general resource classes
  - New utility for detection of duplicate tokens
- Support for CKDS on System z without CEX2C
  - Caution - CKDS not uniquely identified from secure CKDS
- Support of PAN-14, -15, -17, -18
- New Query services calls to enhance CSFIQF

What's New for ICSF V1 R11 --- HCR7770 (Nov 20)

- PKCS #11 enhancements
  - DSA
  - Diffie-Hellman
  - Elliptic Curve cryptography
  - HMAC
  - Blowfish
  - RC4
  - AES GCM (Galois/Counter Mode)
- Path length Improvements
- ICSF Non-cancelable, non-swapable
  - CSFMMAIN becomes CSFINIT
- New Query services calls to enhance CSFIQF
- **Protected key**

CPACF *z9 z10*

- DES (56, 112, 168 bit)
- SHA-1, *SHA-256*
- *AES-128*
- *PRNG*

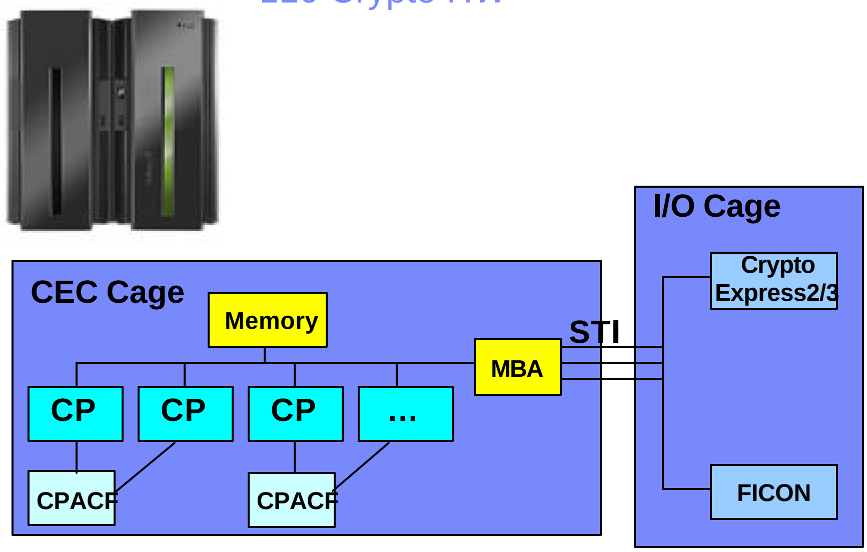
z10 only

- AES-192, AES-256
- SHA-224, SHA-384, SHA-512

z10 SHA-2

- SHA-224 Initialization value
  - C1059ED8 367CD507 3070DD17 F70E5939 FFC00B31 68581511 64F98FA7 BEFA4FA4
- SHA-256 Initialization value
  - 6A09E667 BB67AE85 3C6EF372 A54FF53A 510E527F 9B05688C 1F83D9AB 5BE0CD19
- SHA-384 Initialization value
  - CBBB9D5DC1059ED8 629A292A367CD507 9159015A3070DD17 152FECD8F70E5939 67332667FFC00B31 8EB44A8768581511 DB0C2E0D64F98FA7 47B5481DBEFA4FA4
- SHA-512 Initialization value
  - 6A09E667F3BCC908 BB67AE8584CAA73B 3C6EF372FE94F82B A54FF53A5F1D36F1 510E527FADE682D1 9B05688C2B3E6C1F 1F83D9ABFB41BD6B 5BE0CD19137E2179

z10 Crypto HW



Crypto OP CODES

5 Machine Instructions

Documented in z/OS™ Principles of Operation (POP)

Problem state instructions, as such, can be used directly in applications without going through ICSF.

Instruction Names and Mnemonics

Cipher Message (KM)
Cipher Message with Chaining (KMC)
Compute Intermediate Message Digest (KIMD)
Compute Last Message Digest (KLMD)
Compute Message Authentication Code (KMAC)
Generate Pseudo Random Number

Clear Key Crypto (CPACF)

High Speed Symmetric Algorithms imbedded in each CP

available via ICSF as API's (CSNBSYD/CSNBSYE) or as new operation codes (OP CODES)

"SOFTWARE ENCRYPTION" with algorithm code in hardware

DES TDES SHA-1 AES-128 on z9 (MD5 and AES-192/256 via ICSF)

AES-192 AES-256 SHA-192 SHA-224 SHA-384 on z10

Encryption/Decryption keys are clear (not encrypted) in user address space

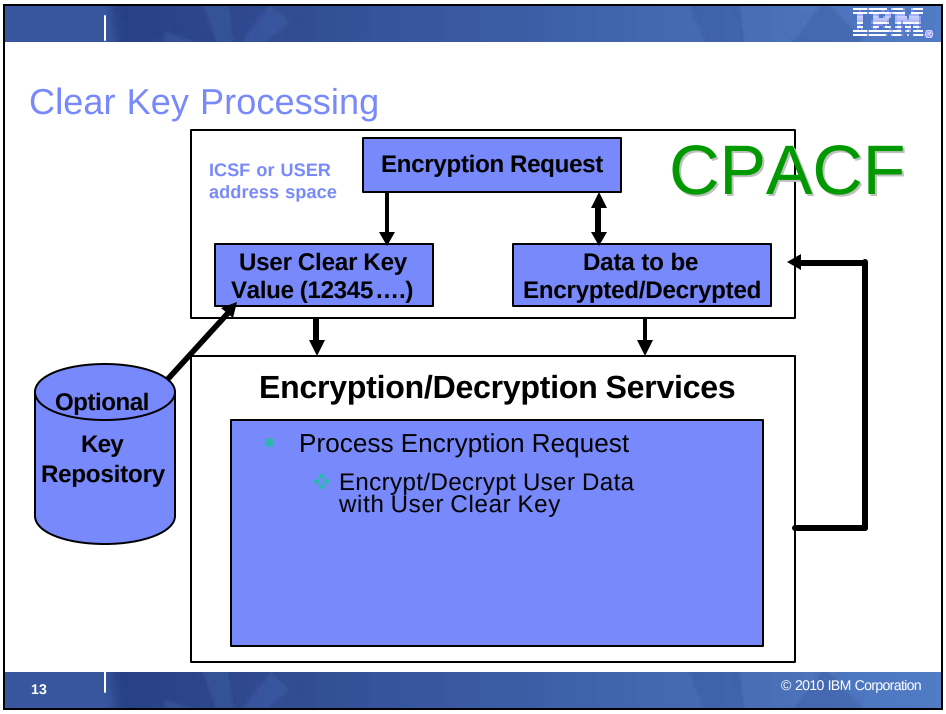
typically not appropriate or allowed for sensitive processing such as VISA, MasterCard, INTERAC, LINK

can be mitigated to offer certain in-house functions

file archive to tape (IBM Encryption Facility)

ICSF user defined functions, keys in clear in the ICSF address space only

Specifically designed for WEB (SSL/TLS/TN3270/FIREWALL) type applications, short duration applications, throw-away key values or semi-protected key values (IMS DB2 Data Encryption tool)



Clear Key Processing

CPACF

ICSF or USER address space

Encryption Request

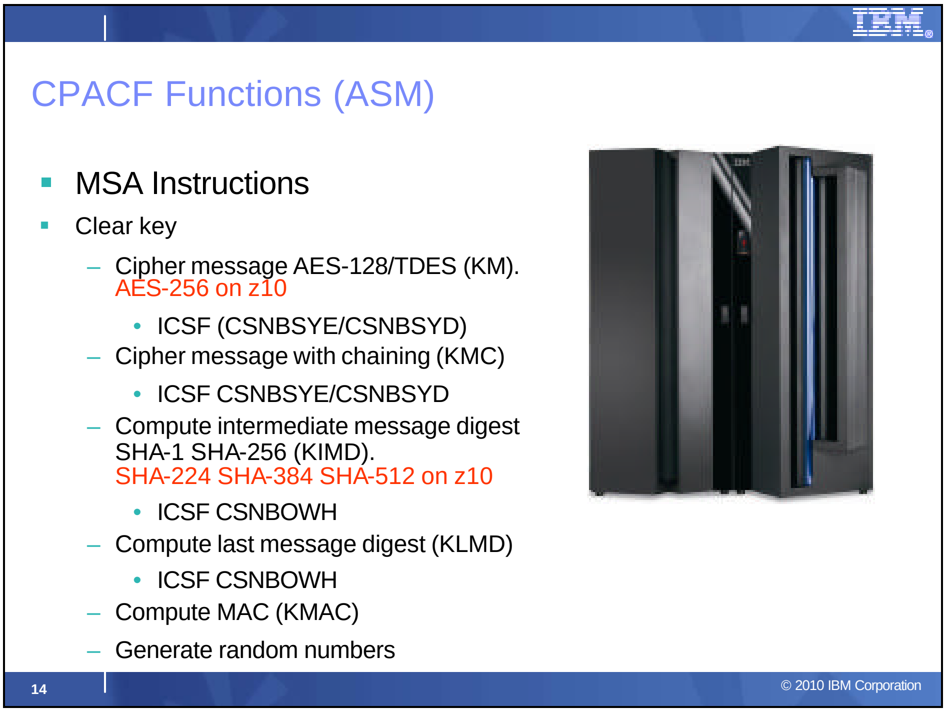
User Clear Key Value (12345….)

Data to be Encrypted/Decrypted

Optional Key Repository

Encryption/Decryption Services

- Process Encryption Request
  - Encrypt/Decrypt User Data with User Clear Key



CPACF Functions (ASM)

- MSA Instructions
- Clear key
  - Cipher message AES-128/TDES (KM). AES-256 on z10
    - ICSF (CSNBSYE/CSNBSYD)
  - Cipher message with chaining (KMC)
    - ICSF CSNBSYE/CSNBSYD
  - Compute intermediate message digest SHA-1 SHA-256 (KIMD). SHA-224 SHA-384 SHA-512 on z10
    - ICSF CSNBOWH
  - Compute last message digest (KLMD)
    - ICSF CSNBOWH
  - Compute MAC (KMAC)
  - Generate random numbers

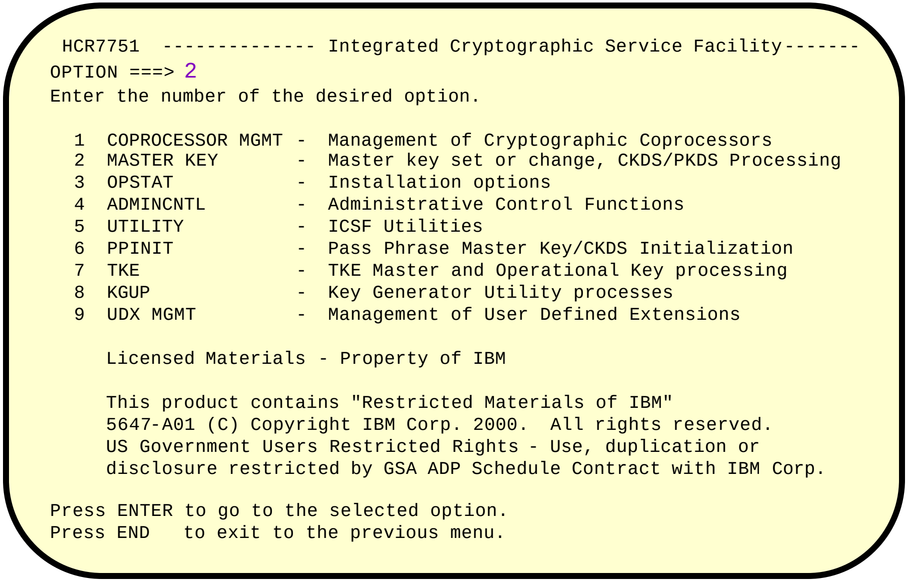
Database Encryption

- Data Encryption for IMS and DB2 Databases
- Row level encryption
- No application changes
- Uses EDITPROC
- Provides user-customizable, pre-coded exits for encrypting IMS and DB2 data
- Exploits zSeries and z9/z10 Crypto Hardware features, which results in low overhead encryption/decryption
- Uses the ANSI Data Encryption Algorithm (DEA), also known as the U.S. National Institute of Science and Technology (NIST) Data Encryption Standard (DES) algorithm
- Works at and is customizable at the IMS segment level or DB2 table level
- Conforms to the existing OS/390 and z/OS security model
- Optimized CPACF or CCF processing

ICSF non-CEX2C System CKDS Support

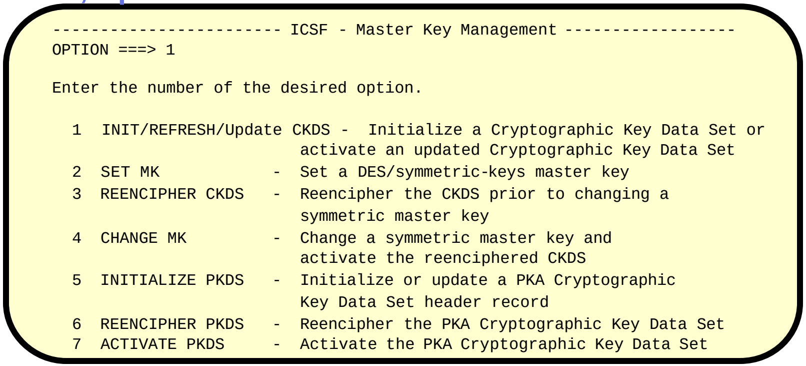
- This support enables users without any crypto coprocessors (secure devices) to store clear keys within a CKDS
  - only for z990/890, z9, & z10
  - not the same as a secure CKDS
- CAUTION:
  - A CKDS initialized on a system without CEX2C cannot be used with a system that has coprocessors.
  - This CKDS type cannot be updated to support systems with coprocessors
- A PKDS is required but not used and cannot be used
- To Use
  - Create CKDS and PKDS
  - Initialize the non-coprocessor CKDS
    - New panel under INIT/REFRESH/UPDATE CKDS

ICSF non-CEX2C System CKDS Support . . .

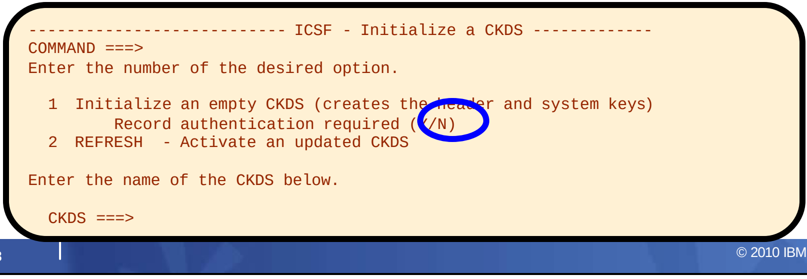


```
 HCR7751  -------------- Integrated Cryptographic Service Facility-------
OPTION ===> 2
Enter the number of the desired option.

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

     Licensed Materials - Property of IBM

     This product contains "Restricted Materials of IBM"
     5647-A01 (C) Copyright IBM Corp. 2000.  All rights reserved.
     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.
```

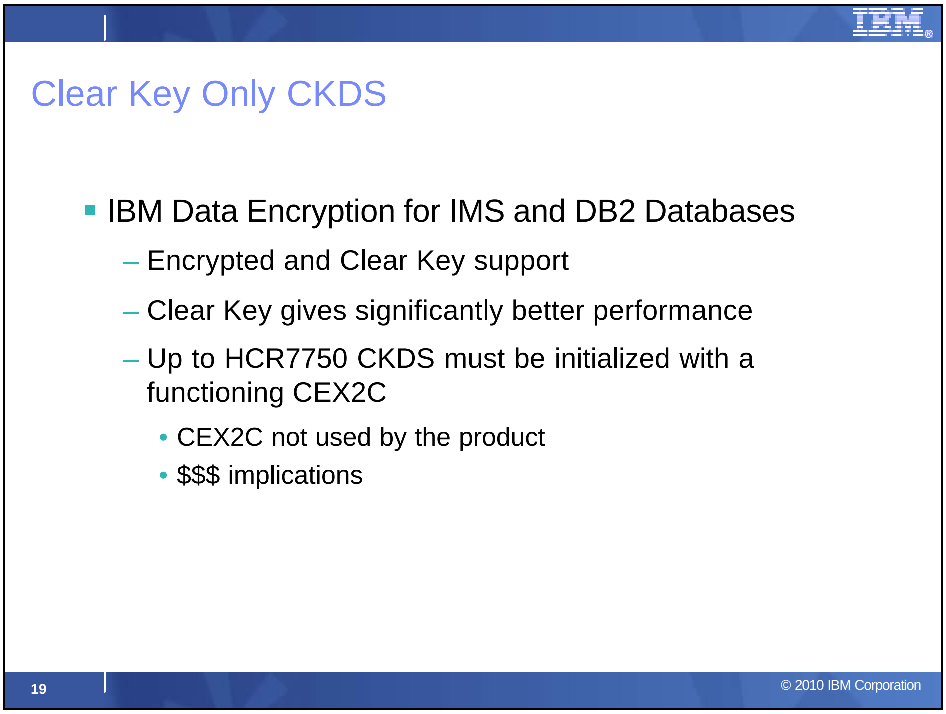
Now, Update the CKDS as Needed . . .



```
------------------------ ICSF - Master Key Management -----------------
OPTION ===> 1

Enter the number of the desired option.

  1  INIT/REFRESH/Update CKDS -  Initialize a Cryptographic Key Data Set or
                                 activate an updated Cryptographic Key Data Set
  2  SET MK               -  Set a DES/symmetric-keys master key
  3  REENCIPHER CKDS      -  Reencipher the CKDS prior to changing a
                             symmetric master key
  4  CHANGE MK            -  Change a symmetric master key and
                             activate the reenciphered CKDS
  5  INITIALIZE PKDS      -  Initialize or update a PKA Cryptographic
                             Key Data Set header record
  6  REENCIPHER PKDS      -  Reencipher the PKA Cryptographic Key Data Set
  7  ACTIVATE PKDS        -  Activate the PKA Cryptographic Key Data Set
```

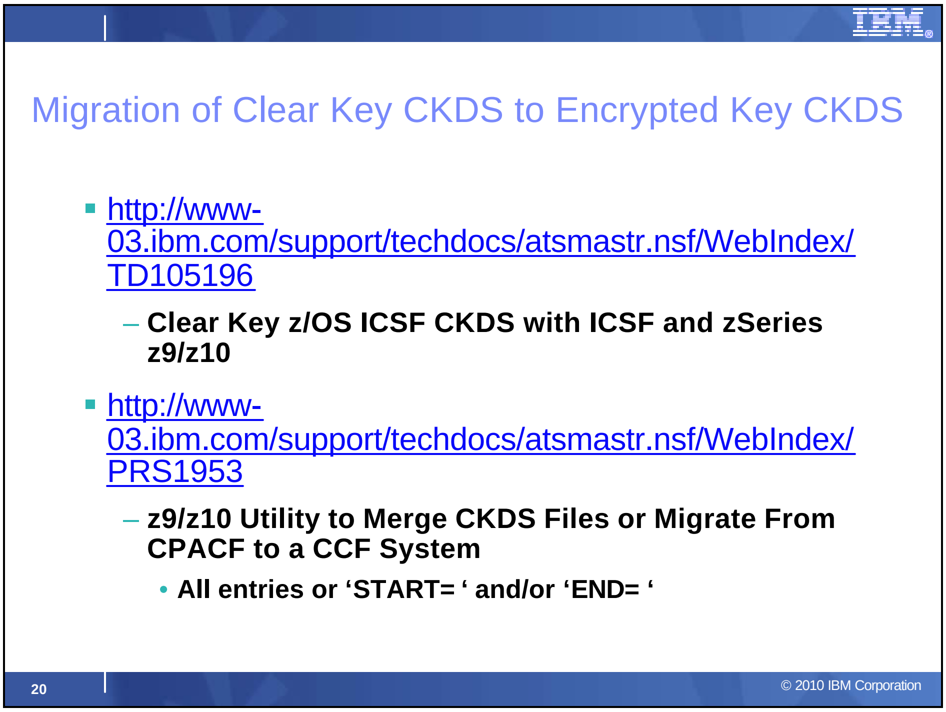


```
--------------------------- ICSF - Initialize a CKDS -------------
COMMAND ===>
Enter the number of the desired option.

  1  Initialize an empty CKDS (creates the header and system keys)
        Record authentication required (Y/N)
  2  REFRESH  - Activate an updated CKDS

Enter the name of the CKDS below.

  CKDS ===>
```



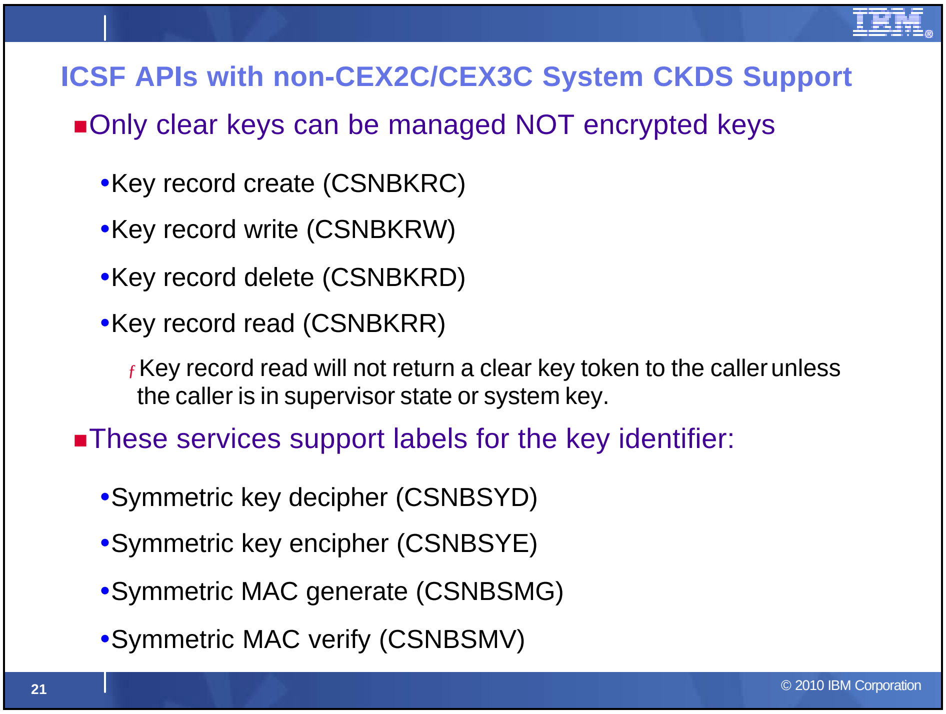
Clear Key Only CKDS

- IBM Data Encryption for IMS and DB2 Databases
  - Encrypted and Clear Key support
  - Clear Key gives significantly better performance
  - Up to HCR7750 CKDS must be initialized with a functioning CEX2C
    - CEX2C not used by the product
    - $$$ implications



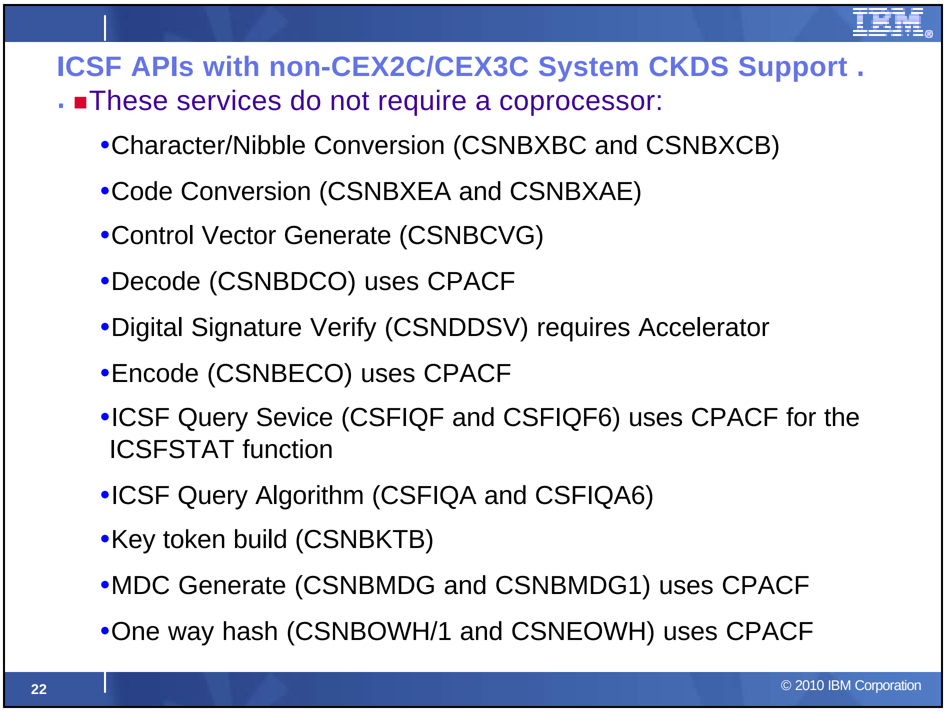
Migration of Clear Key CKDS to Encrypted Key CKDS

- http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD105196
  - **Clear Key z/OS ICSF CKDS with ICSF and zSeries z9/z10**
- http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1953
  - **z9/z10 Utility to Merge CKDS Files or Migrate From CPACF to a CCF System**
    - **All entries or ‘START= ‘ and/or ‘END= ‘**



ICSF APIs with non-CEX2C/CEX3C System CKDS Support

- Only clear keys can be managed NOT encrypted keys
  - Key record create (CSNBKRC)
  - Key record write (CSNBKRW)
  - Key record delete (CSNBKRD)
  - Key record read (CSNBKRR)
    - ƒ Key record read will not return a clear key token to the caller unless the caller is in supervisor state or system key.
- These services support labels for the key identifier:
  - Symmetric key decipher (CSNBSYD)
  - Symmetric key encipher (CSNBSYE)
  - Symmetric MAC generate (CSNBSMG)
  - Symmetric MAC verify (CSNBSMV)



ICSF APIs with non-CEX2C/CEX3C System CKDS Support .

- These services do not require a coprocessor:
  - Character/Nibble Conversion (CSNBXBC and CSNBXCB)
  - Code Conversion (CSNBXEA and CSNBXAE)
  - Control Vector Generate (CSNBCVG)
  - Decode (CSNBDCO) uses CPACF
  - Digital Signature Verify (CSNDDSV) requires Accelerator
  - Encode (CSNBECO) uses CPACF
  - ICSF Query Sevice (CSFIQF and CSFIQF6) uses CPACF for the ICSFSTAT function
  - ICSF Query Algorithm (CSFIQA and CSFIQA6)
  - Key token build (CSNBKTB)
  - MDC Generate (CSNBMDG and CSNBMDG1) uses CPACF
  - One way hash (CSNBOWH/1 and CSNEOWH) uses CPACF

ICSF APIs with non-CEX2C/CEX3C System CKDS Support . . .

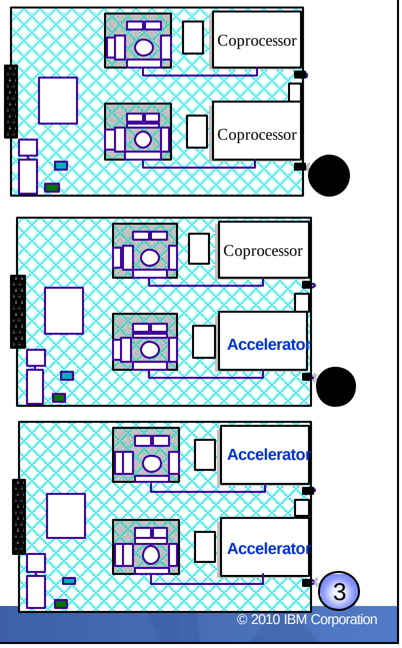
- These services do not require a coprocessor . . .:
  - PKA Decrypt (CSNDPKD) - requires Accelerator
  - PKA Encrypt (CSNDPKE) ZERO-PAD formatting only - requires Accelerator
  - PKA Key Token Build (CSNDPKB)
  - PKA Public Key Extract (CSNDPKX)
  - PKCS11 Token Record Create (CSFPTRC)
  - PKCS11 Token Record Delete (CSFPTRD)
  - PKCS11 Token Record List (CSFPTRL)
  - PKCS11 Get Attribute Value (CSFPGAV)
  - PKCS11 Set Attribute Value (CSFPSAV)

ICSF APIs with non-CEX2C/CEX3C System CKDS Support . . .

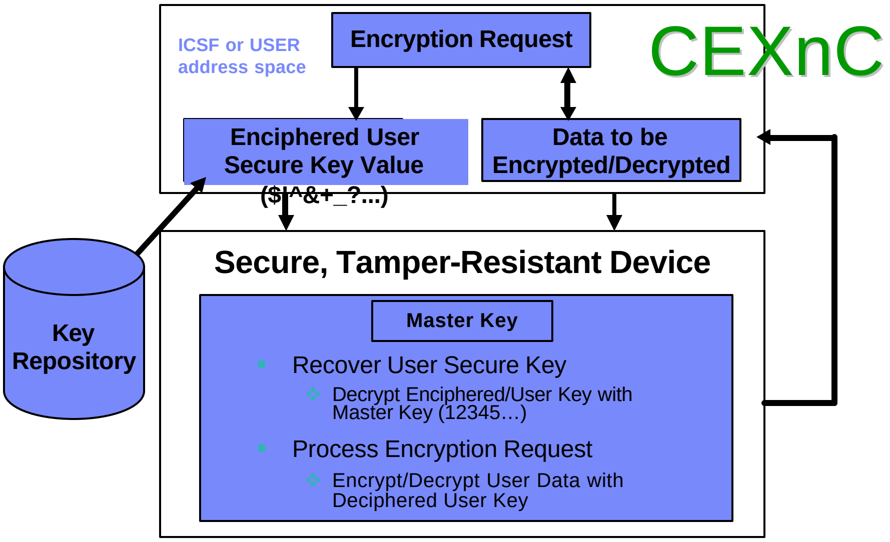
- These services do not require a coprocessor . . .:
  - Symmetric Key Decipher (CSNBSYD and CSNBSYD1) uses CPACF
  - Symmetric Key Encipher (CSNBSYE and CSNBSYE1) uses CPACF
  - Symmetric MAC Generate (CSNBSMG, CSNBSMG1, CSNESMG, and CSNESMG1)
  - Symmetric MAC Verify (CSNBSMV, CSNBSMV1, CSNESMV, and CSNESMV1)
  - X9.9 Data Editing (CSNB9ED)

z10 Crypto Express2/3 Configuration

- Secure Coprocessor **(default)**
  - Provides both Secure key" and "Public key" functionality and performance equivalent to PCIXCC/Crypto Express2 features on z990
  - "Secure key" improved performance compared to PCIXCC on z990 (requires multitasking)
  - "Public key" equivalent performance to PCICA on z990
  - No action required (default configuration)
  - SSL at 1000-2000/second
- Accelerator
  - Provides only 3 "Public key" functions with enhanced performance
  - Must be configured using the HMC
  - SSL at 3000-6000/second



Secure Key Processing



CEXnC

ICSF or USER address space

Encryption Request

Enciphered User Secure Key Value ($!^&+_?...)

Data to be Encrypted/Decrypted

Key Repository

Secure, Tamper-Resistant Device

Master Key

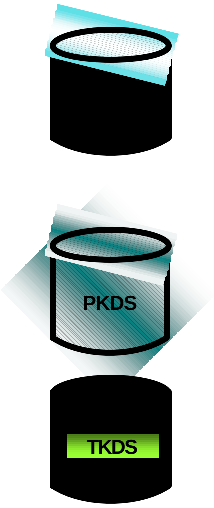
- Recover User Secure Key
  - Decrypt Enciphered/User Key with Master Key (12345…)
- Process Encryption Request
  - Encrypt/Decrypt User Data with Deciphered User Key

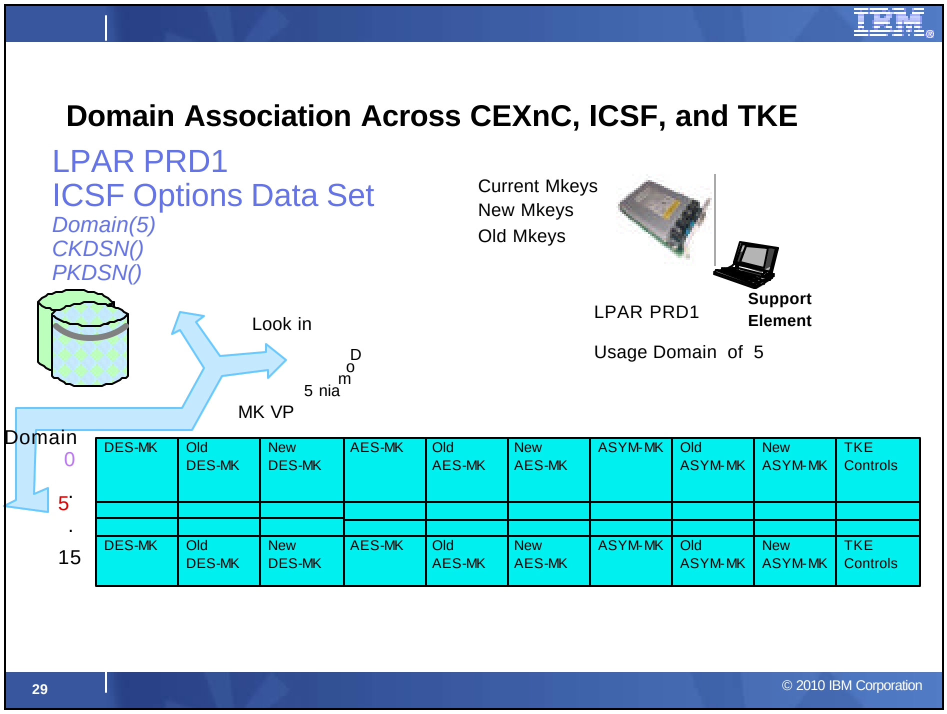
ICSF CEXnC Functions

- Encipher/Decipher
  - ICSF CSNBENC/CSNBDEC
- PIN
- MAC
  - X9.9, X9-19
  - ISO16609 CBC TDES MAC
    - Strengthen data integrity
- Random Number Generate
- Key Generate
- Key Management
- Remote key loading for ATM's and POS
  - More flexible key management and privacy

Master Keys . . .

- DES Master Key
  - ƒ DES-MK protects secure DES Keys stored in Cryptographic Key Data Set
  - ƒ Can change dynamically in native mode
  - ƒ Stored in CEXnC, not CKDS
- AES Master Key
  - ➢AES-MK protects AES secure keys stored in the CKDS
  - ➢Can change dynamically
  - ➢Stored in CEXnC, not CKDS
- PKA
  - ƒ Called ASYM-MK
  - ƒ Protect Application Keys stored in Public Key Data Set (PKDS)
  - ƒ Stored in CEXnC, not PKDS
    - ƒ PKDS contains ASYM-MK HASH for CEXnC/ICSF verification
- PKCS#11
  - ƒ Clear keys





Domain Association Across CEXnC, ICSF, and TKE

LPAR PRD1
ICSF Options Data Set
Domain(5)
CKDSN()
PKDSN()

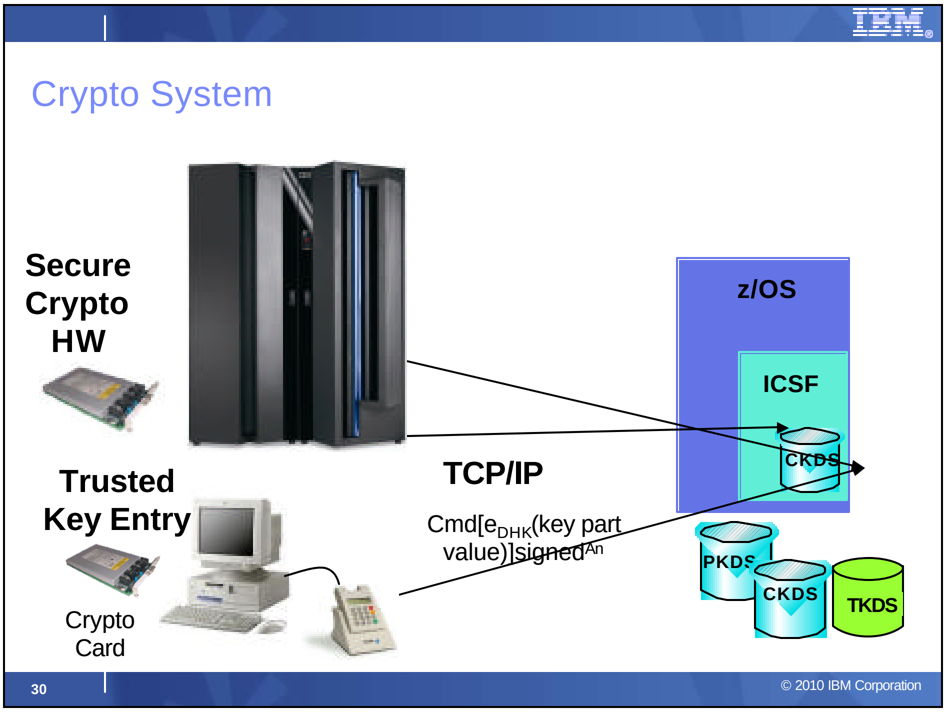
Current Mkeys
New Mkeys
Old Mkeys

Support Element

LPAR PRD1

Usage Domain of 5

Domain										
0	DES-MK	Old DES-MK	New DES-MK	AES-MK	Old AES-MK	New AES-MK	ASYM-MK	Old ASYM-MK	New ASYM-MK	TKE Controls
5										
.										
15	DES-MK	Old DES-MK	New DES-MK	AES-MK	Old AES-MK	New AES-MK	ASYM-MK	Old ASYM-MK	New ASYM-MK	TKE Controls



Crypto System

Secure Crypto HW

Trusted Key Entry

Crypto Card

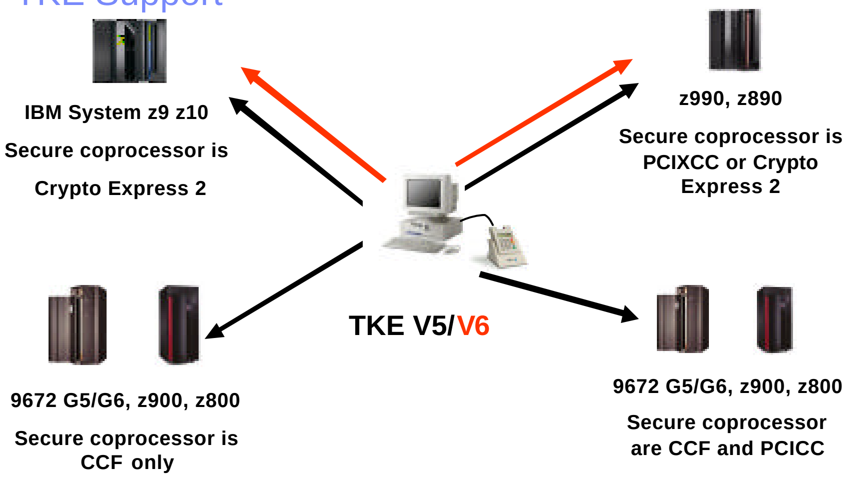
TCP/IP

Cmd[e_{DHK}(key part value)]signed^{An}

z/OS

ICSF

TKE Support



IBM System z9 z10

Secure coprocessor is

Crypto Express 2

z990, z890

Secure coprocessor is PCIXCC or Crypto Express 2

TKE V5/V6

9672 G5/G6, z900, z800

Secure coprocessor is CCF only

9672 G5/G6, z900, z800

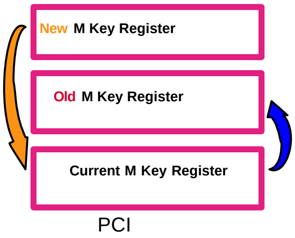
Secure coprocessor are CCF and PCICC

Master Keys . . .

	DES Master Key Storage	AES Master Key Storage	PKA Master Key Storage Areas
CEXnC Not CPACF	DES-MK	AES-MK	ASYM-MK
	New DES-MK	New AES-MK	New ASYM-MK
	Old DES-MK	Old AES-MK	Old ASYM-MK

First Time DESAES Master Key Entry Process

- Must be in Special Secure Mode
- Enter (PPINIT) or process key part values
- Set the Master Key registers
- CKDS
  - For first-time, empty CKDS (IDCAMS DEFINE)
  - Initialize CKDS/PKDS
  - Perform SET of Master Key, system keys added automatically,



New M Key Register

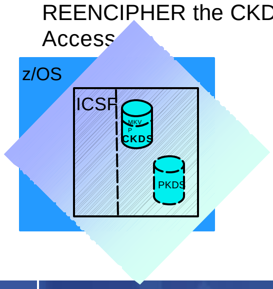
Old M Key Register

Current M Key Register

PCI

Later DES Master Key Entry Process (New LPAR/DR)

- Must be in Special Secure Mode
- Must run COMPAT(NO)
- Enter key part values into New Master Key (NMK) Register
- Based on Status of CKDS activate the New Master Key, if CKDS header record contains MKVP
  - Matching MKVP of contents in NMK, do SET
  - Different than MKVP of contents in NMK, do CHANGE and REENCIPHER the CKDS first, perhaps Disable Dynamic CKDS Access



z/OS

ICSF

MKVP

CKDS

PKDS

New M Key Register

Old M Key Register

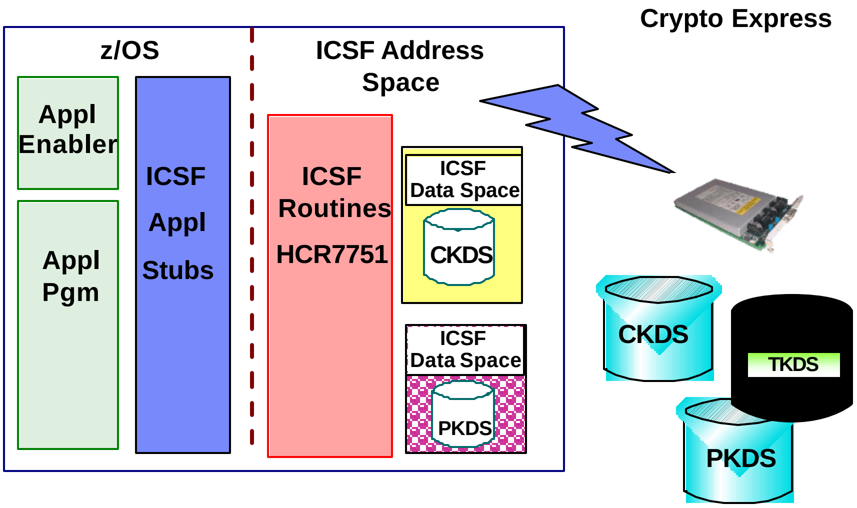
Current MK Register

PCI

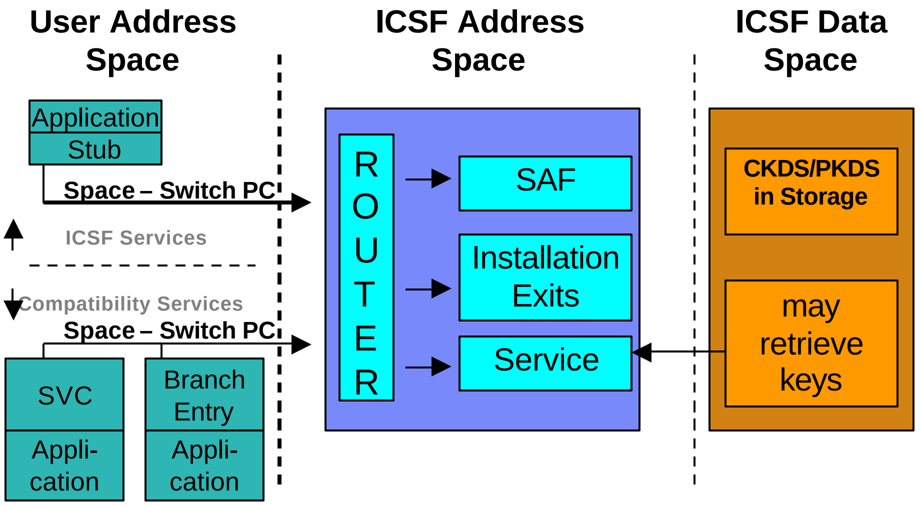
ICSF

- ICSF is a no charge system task that provides a tool kit for application access to cryptographic functions
- ICSF provides load balancing across cryptographic hardware (CEXnC)
- ICSF provides a secure storage for cryptographic keys (CKDS, PKDS)
- ICSF checks SAF access to functions and keys that it stores for you
- ICSF is not in itself a full key management system

ICSF – Interface to the Hardware



ICSF Internals



User Address Space

ICSF Address Space

ICSF Data Space

Application Stub

Space – Switch PC

ICSF Services

Compatibility Services

Space – Switch PC

SVC

Appli-cation

Branch Entry

Appli-cation

ROUTER

SAF

Installation Exits

Service

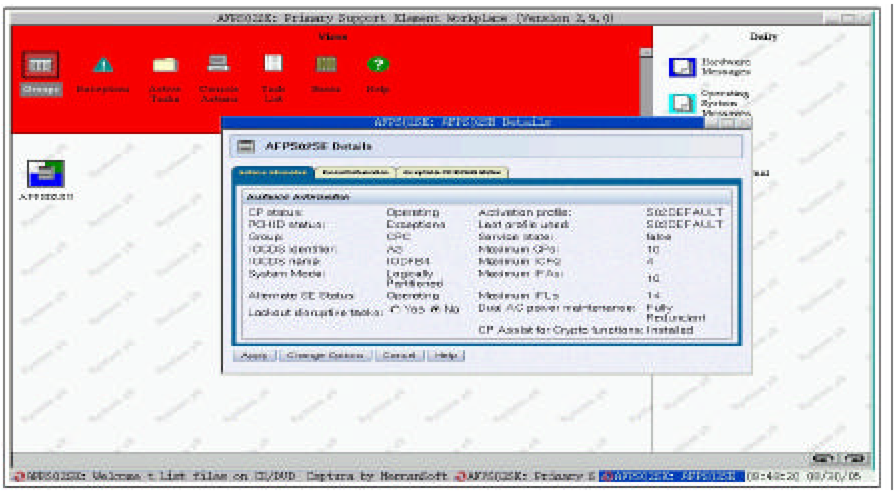
CKDS/PKDS in Storage

may retrieve keys

Hardware (SE) Functions

- Add feature 3863
- Configure LPAR crypto domains
- Configure CEXnC/ CEXnA

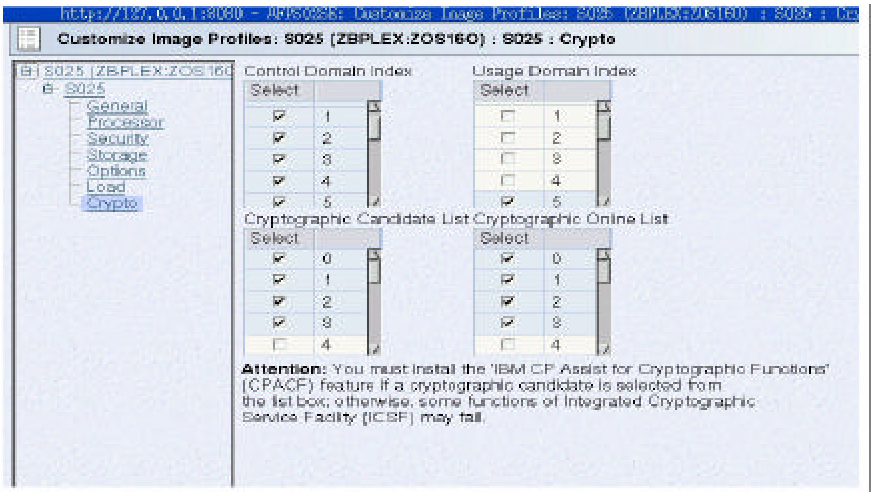
FC 3863 Installed


AFPS02SE: Primary Support Element Workplace (Version 2.9.0)
Views
Groups
Exceptions
Active Tasks
Console Actions
Task List
Books
Help
Daily
Hardware Messages
Operating System Messages
AFPS02SE
AFPS02SE: AFPS02SE Details

AFPS02SE Details

CP status:	Operating	Activation profile:	S02DEFAULT
PCHID status:	Exceptions	Last profile used:	S02DEFAULT
Group:	CPC	Service state:	false
IOCDS identifier:	A3	Maximum CPs:	10
IOCDS name:	IODF84	Maximum ICFs:	4
System Mode:	Logically Partitioned	Maximum IFAs:	10
Alternate SE Status:	Operating	Maximum IFLs:	14
Lockout disruptive tasks:	○ Yes ● No	Dual AC power maintenance:	Fully Redundant
		CP Assist for Crypto functions:	Installed

Crypto Definitions (z10 Dynamic)


http://127.0.0.1:8080 - AFPS02SE: Customize Image Profiles: S025 (ZBPLEX:ZOS160) : S025 : Cry
S025 (ZBPLEX:ZOS160
S025
General
Processor
Security
Storage
Options
Load
Crypto

Customize Image Profiles: S025 (ZBPLEX:ZOS160) : S025 : Crypto

Control Domain Index

Select	
☑	1
☑	2
☑	3
☑	4
☑	5

Usage Domain Index

Select	
□	1
□	2
□	3
□	4
☑	5

Cryptographic Candidate List

Select	
☑	0
☑	1
☑	2
☑	3
□	4

Cryptographic Online List

Select	
☑	0
☑	1
☑	2
☑	3
□	4

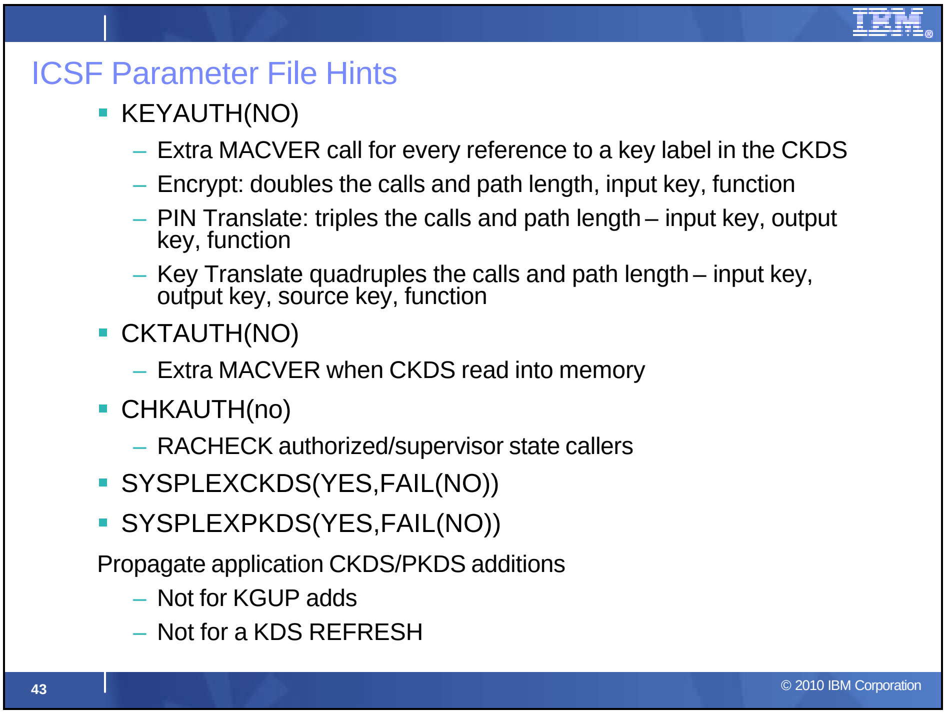
Attention: You must install the 'IBM CP Assist for Cryptographic Functions' (CPACF) feature if a cryptographic candidate is selected from the list box; otherwise, some functions of Integrated Cryptographic Service Facility (ICSF) may fail.

SAF (RACF/ACF2/TopSecret)

- ICSF Issues SAF calls to two resources
  - – CSFSERV
    - What service is requested
      - – Not done for non-crypto based calls such as ASCII-EBCDIC translation or Clear Key Encrypt/Decrypt (CSNBSYE/CSNBSYD)
    - I can encrypt, but not decrypt (secure key)
  - – CSFKEYS
    - What key label is requested form the xKDS
    - I can encrypt, but not with production keys (based on label)
  - – ICSF Administrator's Guide Chapter 3
  - – ICSF is also a user subject to SAF rules for internal functions
- XFACILITgeneral resource class in SAF (RACF) controls use of tokens stored in the CKDS and PKDS
- XCSFKEY general resource class in SAF controls who can export a token using the Symmetric Key Export API (CSNDSYX)

z/OS ICSF FMIDs

z/OS & z/OS.e	ICSF FMID	Web Deliverable Name
V1.8	HCR7731	Enhancement for Crypto support in V1R6/R7 (included in base)
	HCR7750	Crypto support for V1R7-R9 & z/OS.e V1R7-R8
	HCR7751	Crypto support for V1R8-R10 & z/OS.e V1R8
V1.9	HCR7740	Enhancement for Crypto support in V1R9 (included in base)
	HCR7750	Crypto support for V1R7-R9 & z/OS.e V1R7-R8
	HCR7751	Crypto support for V1R8-R10 & z/OS.e V1R8
V1.10 V1.11	HCR7750	Crypto support for V1R7-R9 & z/OS.e V1R7-R8 (included in base)
	HCR7751 HCR7770	Crypto support for V1R8-R10 & z/OS.e V1R8 Protected Key

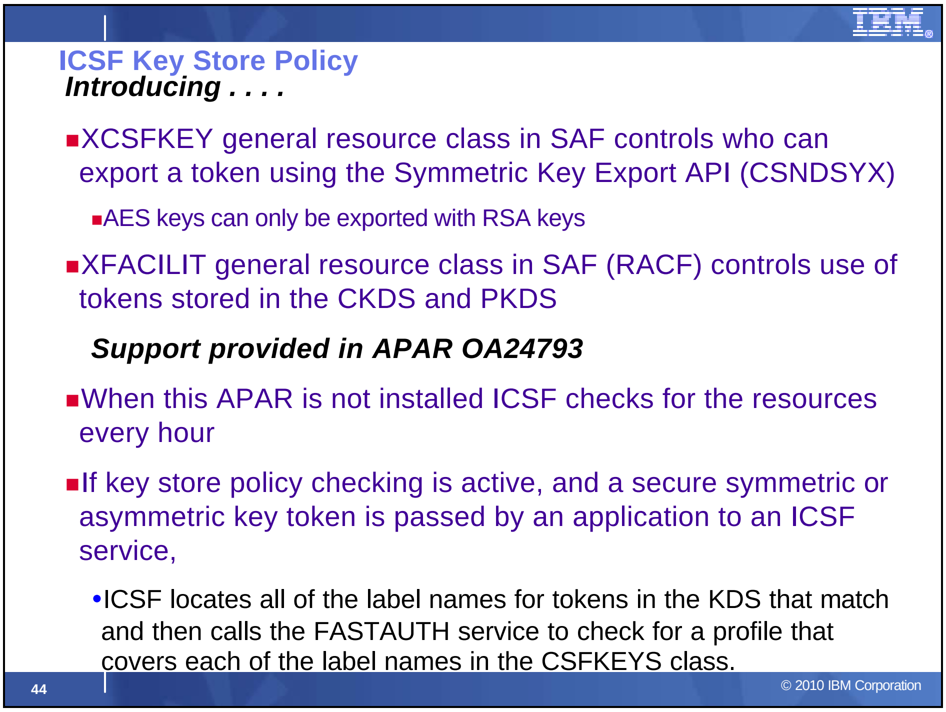


ICSF Parameter File Hints

- KEYAUTH(NO)
  - Extra MACVER call for every reference to a key label in the CKDS
  - Encrypt: doubles the calls and path length, input key, function
  - PIN Translate: triples the calls and path length – input key, output key, function
  - Key Translate quadruples the calls and path length – input key, output key, source key, function
- CKTAUTH(NO)
  - Extra MACVER when CKDS read into memory
- CHKAUTH(no)
  - RACHECK authorized/supervisor state callers
- SYSPLEXCKDS(YES,FAIL(NO))
- SYSPLEXPKDS(YES,FAIL(NO))

Propagate application CKDS/PKDS additions

- Not for KGUP adds
- Not for a KDS REFRESH



ICSF Key Store Policy

Introducing . . . .

- XCSFKEY general resource class in SAF controls who can export a token using the Symmetric Key Export API (CSNDSYX)
  - AES keys can only be exported with RSA keys
- XFACILIT general resource class in SAF (RACF) controls use of tokens stored in the CKDS and PKDS

Support provided in APAR OA24793

- When this APAR is not installed ICSF checks for the resources every hour
- If key store policy checking is active, and a secure symmetric or asymmetric key token is passed by an application to an ICSF service,
  - ICSF locates all of the label names for tokens in the KDS that match and then calls the FASTAUTH service to check for a profile that covers each of the label names in the CSFKEYS class.

ICSF Key Store Policy - XCSFKEY

- The XCSKEY class profiles expands the protection against keys being sent outside of system

  CSF.XCSFKEY.ENABLE.AES

  CSF.XCSFKEY.ENABLE.DES
- Currently only the CSNBKEX API allowing export of DES/TDES keys has the capability of SAF protection of use
- XCSFKEY class only protects keys associated with CSNDSYX API meaning this profile allows protection of AES or DES keys exported by a RSA public key
- Those users or applications sending AES or DES keys outside of the system should have the appropriate authority under the XCSFKEY class appropriate profile
  - Applications that use SSL/TLS are examples of those that would need access to XCSFKEY if defined

ICSF Key Store Policy - XCSFKEY How ???

- The XCSKEY class profiles are

  RDEFINE XFACILIT CSF.XCSFKEY.ENABLE.AES

  RDEFINE XFACILIT CSF.XCSFKEY.ENABLE.DES
- XCSFKEY class controls who can export a token using the Symmetric Key Export callable service (CSNDSYX)
- Key policy control profiles in the XFACILIT class do not have to be active or RACLISTed

ICSF Key Store Policy - What????

- RACF can also be used to protect the use of key tokens passed in when calling a service using the Key Store Policy
- Key store policies give users permission to:
  - use a secure symmetric or an asymmetric key token with an ICSF service
  - supports a default token access value
- In addition, there is a key store policy control to prevent duplicate tokens with different key labels from being stored in the CKDS or PKDS
- Use the XFACILIT class to define a key store policy that controls use of key tokens that are stored in the CKDS & PKDS
  - Activate key store checking for CKDS or PKDS
  - Define policy control when Sym or Asym key token existing outside CKDS or PKDS is used
  - Activate policy for duplicate keys within CKDS or PKDS

ICSF Key Store Policy - What???? . . .

- Key Store Policy for KDS Label Checking
  - CSF.CKDS.TOKEN.CHECK.LABEL.WARN or CSF.CKDS.TOKEN.CHECK.LABEL.FAIL
  - CSF.PKDS.TOKEN.CHECK.LABEL.WARN or CSF.PKDS.TOKEN.CHECK.LABEL.FAIL
- Key Store Policy options for KDS
  - Key Store Policy supports both WARN and FAIL mode via profile name definition rather than by the SETROPTS setting
    - When the profile activing keystore policy checking ends with WARN, ICSF writes an 82 type SMF record containing an indictor that the key store policy checking is in WARN mode. The application result would have been success or failure and a list of all the labels that matched the token the application used is provided. The application is granted access to use the key token.
    - When the key store policy checking ends with FAIL, 80 type SMF records are written by RACF and the application is denied access. The resource name in the RACF SMF record is the first label that failed the check.

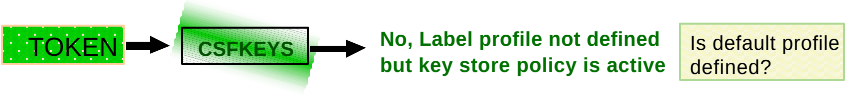
ICSF Key Store Policy - How????

■Key Store Policy for key tokens not in KDS

- Key tokens that do not have a label due to being stored outside the KDS' can now have SAF protection
  - ƒ No profile would exist in CSFKEYS so only CSFSERV would provide protection for function but not for key value use
  - ƒ ICSF services will look for a DEFAULT.LABEL profile in the CSFKEYS class to determine application access for key use

■Define key store policy when a secure symmetric or an asymmetric key token not in a KDS is used

- CSF.CKDS.TOKEN.CHECK.DEFAULT.LABEL
  CSF.PKDS TOKEN.CHECK.DEFAULT.LABEL



ICSF Key Store Policy - How???? . . .

■Key Store Policy for key duplicates

- Key tokens whose key value when in the clear matches any other key token's key value when in the clear will be determined
  - ƒ If duplicate key checking active, applications attempting to write keys to the KDS with a value that exist will be prevented from writing the record
  - ƒ Key Duplicate checks both the 64-byte label and the control vectors associated with the key that represent the 8-byte key type for DES/TDES symmetric keys only (AES and RSA keys have no keytype)

■Define key store policy profile for key duplicates

- CSF.CKDS.TOKEN.NODUPLICATES
  CSF.PKDS.TOKEN.NODUPLICATES
- Applications will be prevented from using ICSF services to write a key token containing a duplicate key value to the the KDS
- This policy profile denies applications from doing what can be done via KGUP, ICSF's Key Generation Utility Program, with the "group label" option

ICSF Key Store Policy - How???? . . .

- New Batch Utility - CSFDUTIL to find duplicates

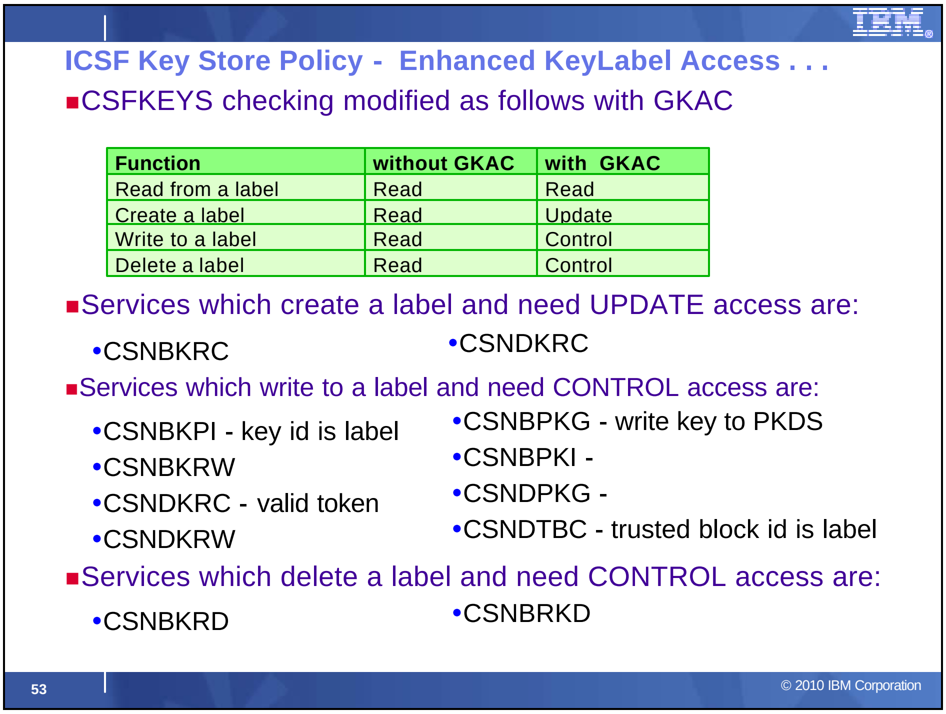
```
//DUTIL    EXEC PGM=CSFDUTIL
//SYSOUT  DD SYSOUT=A
//SYSIN      DD *
     CKDSN(ckds.name)
/*
```

  - May wish to disable dynamic KDS services
- Output from CSFDUTIL about any duplicates found

CKDS		PKDS	
Column	**Value**	**Column**	**Value**
1-64	Key label	1-64	Key label
67-74	Key type from KDS record	67-74	Create date
77-84	Create date	77-84	Create time
87-94	Create time	87-94	Last update date
97-104	Last update date	97-104	Last update time
107-114	Last update time		

ICSF Key Store Policy - Enhanced KeyLabel Access

- This support enables Granular Keylabel Access Control (GKAC) based on service
- The profiles that exist in the XFACILIT class for this allow FAIL or WARN
  - CSF.CSFKEYS.AUTHORITY.LEVELS.FAIL
    - FAIL form will perform the CSFKEYS SAF check within a service and disallow the action if the check fails with 8/16004
  - CSF.CSFKEYS.AUTHORITY.LEVELS.WARN
    - The .WARN form will perform the SAF check, but continue if the caller has at least READ access to the profile
- If both profiles defined, .FAIL has precedence
- Message issued at ICSF startup and anytime XFACILIT profiles are defined, deleted, or the class deactivated
  - CSFM610I GRANULAR KEYLABEL ACCESS CONTROL IS *state.* where *state* is either *ENABLED or DISABLED.*

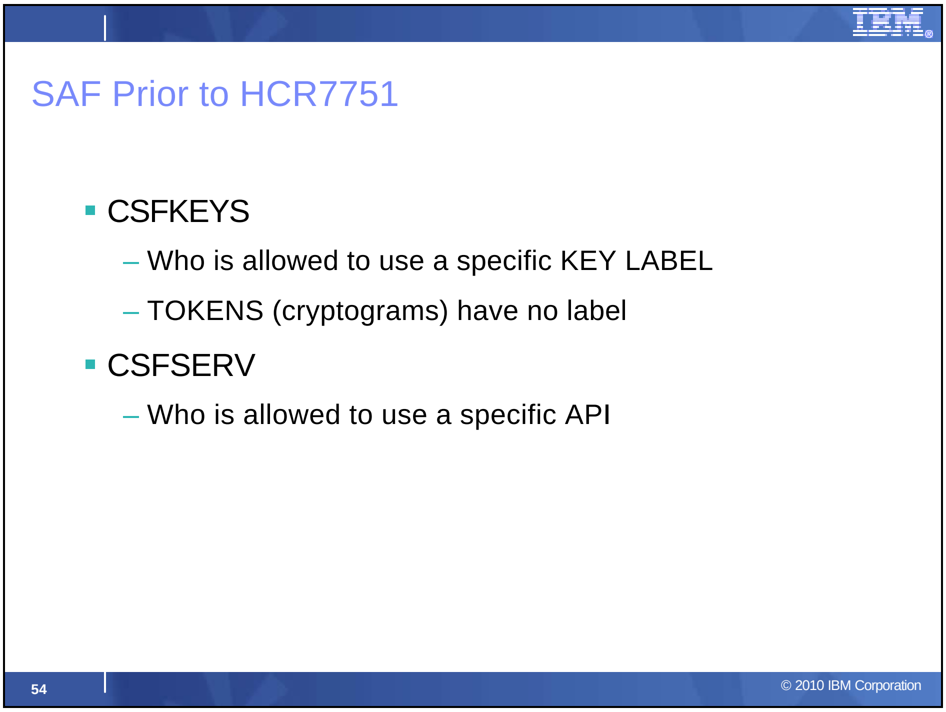


ICSF Key Store Policy - Enhanced KeyLabel Access . . .

- CSFKEYS checking modified as follows with GKAC

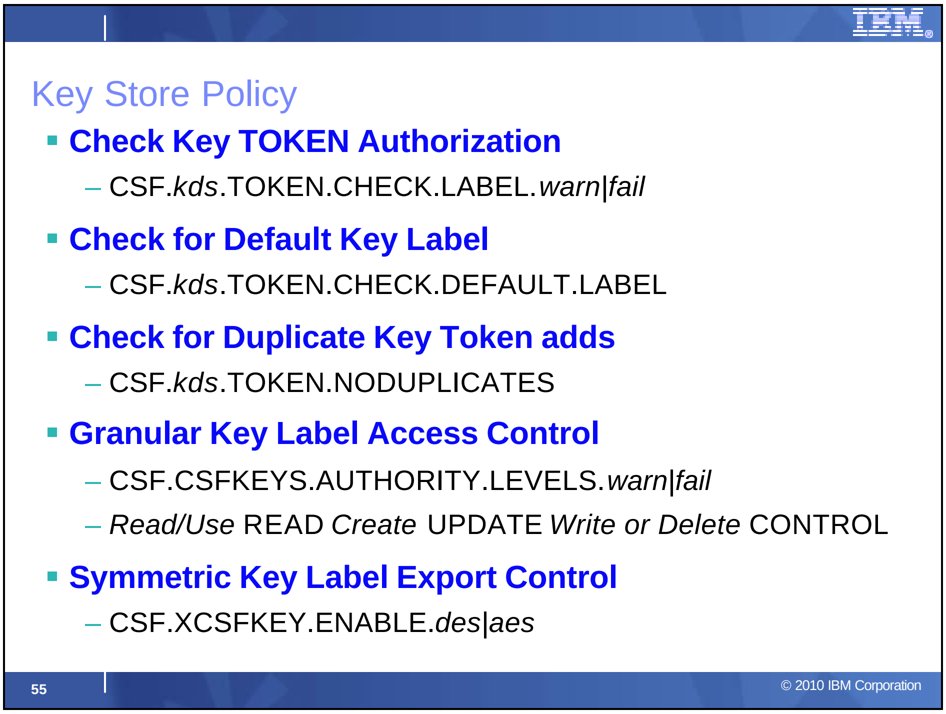
Function	without GKAC	with GKAC
Read from a label	Read	Read
Create a label	Read	Update
Write to a label	Read	Control
Delete a label	Read	Control

- Services which create a label and need UPDATE access are:
  - CSNBKRC
  - CSNDKRC
- Services which write to a label and need CONTROL access are:
  - CSNBKPI - key id is label
  - CSNBKRW
  - CSNDKRC - valid token
  - CSNDKRW
  - CSNBPKG - write key to PKDS
  - CSNBPKI -
  - CSNDPKG -
  - CSNDTBC - trusted block id is label
- Services which delete a label and need CONTROL access are:
  - CSNBKRD
  - CSNBRKD



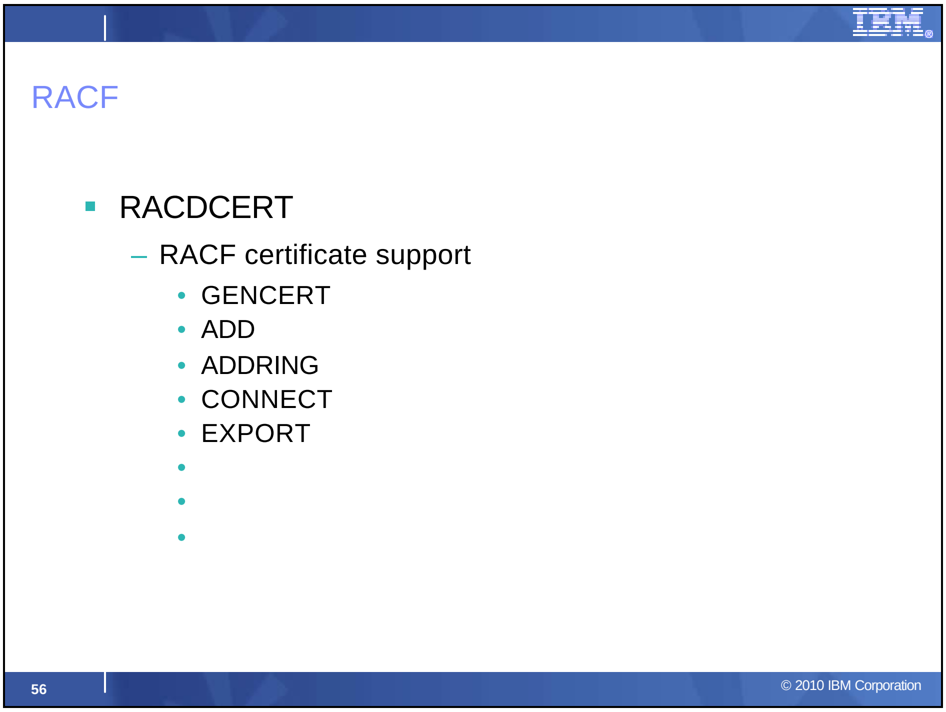
SAF Prior to HCR7751

- CSFKEYS
  - Who is allowed to use a specific KEY LABEL
  - TOKENS (cryptograms) have no label
- CSFSERV
  - Who is allowed to use a specific API



Key Store Policy

- **Check Key TOKEN Authorization**
  - CSF.*kds*.TOKEN.CHECK.LABEL.*warn*|*fail*
- **Check for Default Key Label**
  - CSF.*kds*.TOKEN.CHECK.DEFAULT.LABEL
- **Check for Duplicate Key Token adds**
  - CSF.*kds*.TOKEN.NODUPLICATES
- **Granular Key Label Access Control**
  - CSF.CSFKEYS.AUTHORITY.LEVELS.*warn*|*fail*
  - *Read/Use* READ *Create* UPDATE *Write or Delete* CONTROL
- **Symmetric Key Label Export Control**
  - CSF.XCSFKEY.ENABLE.*des*|*aes*



RACF

- RACDCERT
  - RACF certificate support
    - GENCERT
    - ADD
    - ADDRING
    - CONNECT
    - EXPORT
    -
    -
    -

References

- ATS TechDocs Web Site
  - http://www-1.ibm.com/support/techdocs/atsmastr.nsf
    - search on CRYPTO
- IBM Web Libraries
  - http://www-1.ibm.com/servers/eserver/zseries/zos/bkserv/
  - http://www-1.ibm.com/servers/eserver/zseries/library/online_pubs.html
  - http://www-1.ibm.com/servers/eserver/zseries/library/whitepapers/
  - http://app-06.www.ibm.com/servers/resourcelink
  - http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp3747.html?Open
- Standards
  - http://www.ietf.org/
  - http://csrc.nist.gov/cryptval/140-1/1401val.htm
  - http://www.rsasecurity.com/rsalabs/standards/
- Free Stuff
  - http://www.infosecuritymag.com/
  - http://www.scmagazine.com/index2.html
  - http://www.schneier.com/crypto-gram.html

Questions

Programming can be fun, so can cryptography; however they should not be combined.

--Kreitzberg and Shneiderman

The Pause That Refreshes