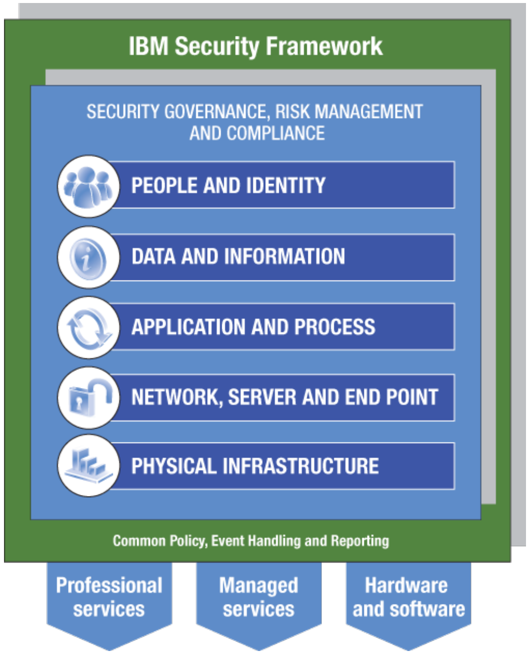
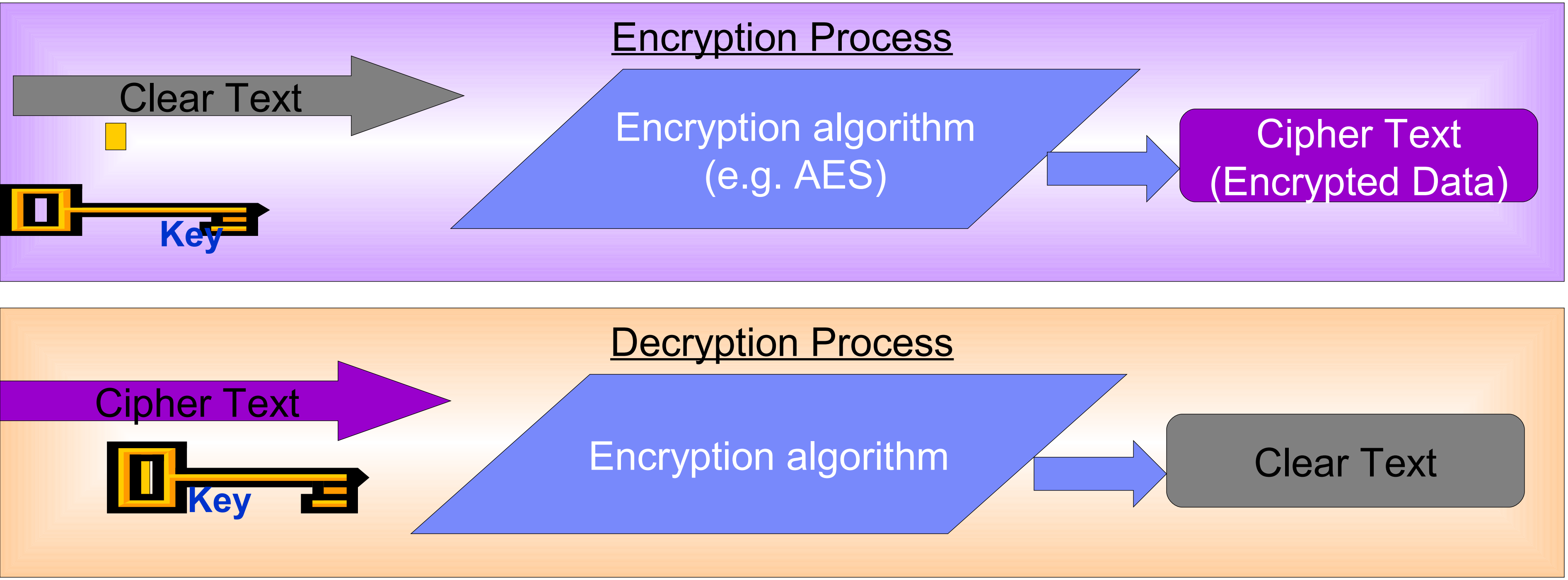
Overview of ICSF services on zSeries
and
IBM Encryption Tool for IMS and DB2 Databases

Part 1 – Encryption for z/OS



Encryption is a technique used to help protect data from unauthorized access



- Data that is not encrypted is referred to as “clear text”
- Clear text is encrypted by processing with a “key” and an encryption algorithm
  - Several standard algorithms exist, include DES, TDES and AES
- Keys are bit streams that vary in length
  - For example AES supports 128, 192 and 256 bit key lengths

Encryption Algorithms – which ones?

- DES
  - Data Encryption Standard – 56 Bit, **viewed as weak and generally unacceptable by some institutes (e.g. NIST/FIPS)**
- TDES
  - Triple Data Encryption Standard – 128 bit, **universally accepted algorithm**.
- AES
  - Advanced Encryption Standard 128 or 256 bit. **Newest commercially used algorithm**
- What is acceptable?
  - DES is viewed as unacceptable
  - TDES is viewed as acceptable and NIST compliant
  - AES 128 or 256 is also viewed as acceptable and strategic
- For more information
  - TDES NIST* Special Publication 800-67 V1 entitled "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher" and can be found at http://csrc.nist.gov/publications/nistpubs/800-67/SP800-67.pdf
  - TDES NIST FIPS Publication 197 entitled "Announcing the Advanced Encryption Standard (AES)" and can be found at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
  - * NIST – Nat. Institute of Standards and Technology

Hardware Requirements – for TDES and AES

- TDES is well supported in both current z9 and z10 hardware combinations
- **The Encryption tool (5666-P03)** will generate exits that can support AES 128, AES 192, or AES 256 keys. However, the type of IBM server determines whether the support for that key length is supported
  - AES 128 support is supported in the hardware (KMC* instruction) on z9 and z10.
  - AES 192 and 256 support is supported in the hardware (KMC* instruction) on z10 only.
  - AES 256 support is supported in the software (ICSF API) on z9.
- Our suggestion is to implement 128 bit AES on z9
- or
- 256 bit AES on z10 for the best performance experience
- KMC- DB2 only (Cipher Message with Chaining)

ICSF - Integrated Cryptographic Service Facility

z/OS Integrated Software Support for Data Encryption

- Enhanced Key Management
  (Cryptographic Key Data Set (CKDS) Key Repository)

  Key Creation and Distribution

  Public and Private Keys
  Secure and Clear Keys
  Master Keys

- Access Control for CKDS via Security Access Facility (SAF)

  ❖ Control access to ICSF Callable Services
  ❖ Control access to *Key Labels* (Key Alias) stored in the CKDS

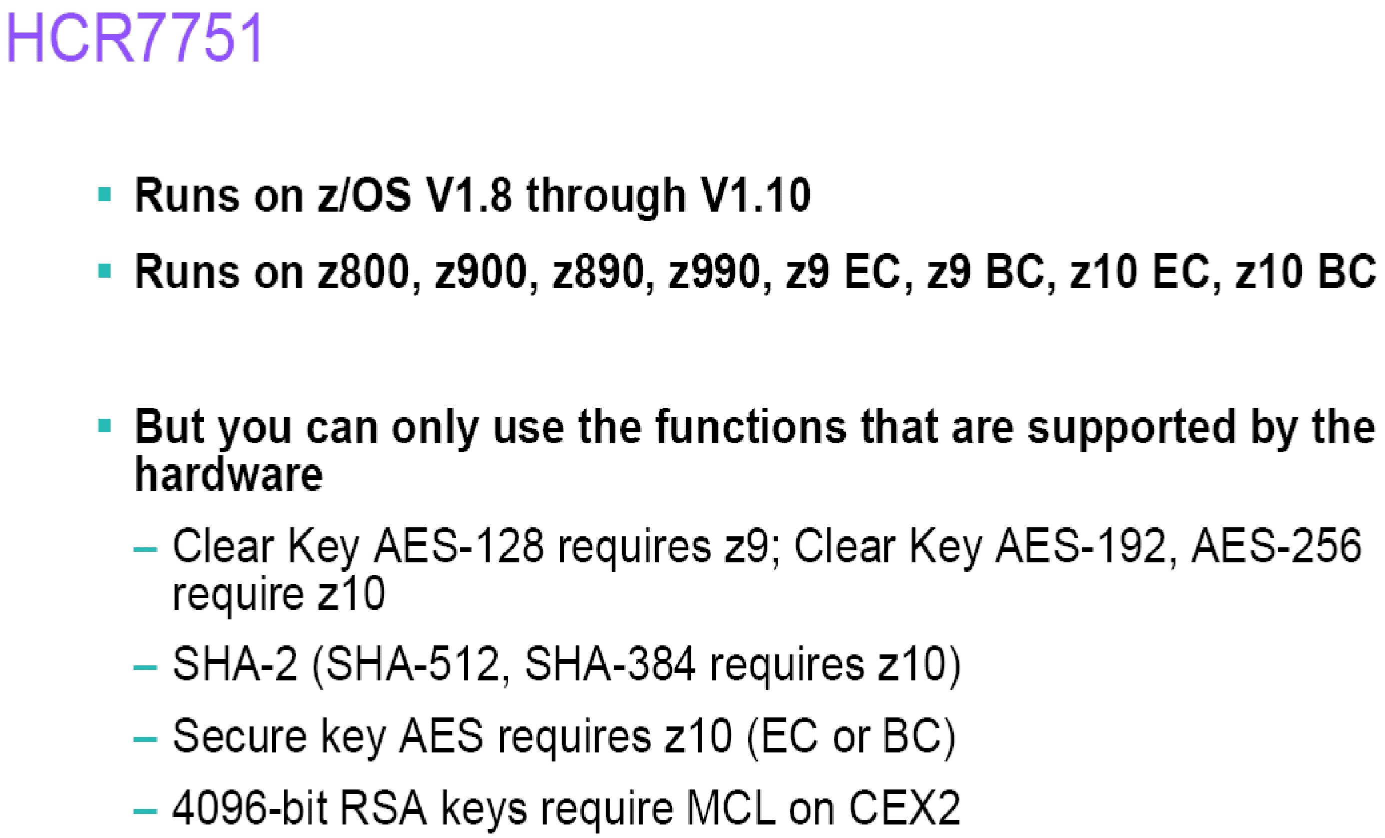
- Hardware and Software Implementation of AES (z9/z10 CPACF feature*)

- Operating System S/W API Interface to Cryptographic Hardware

- Procedures for creating Installation-Defined Callable Services (UDX*)

- * CPACF=CP Assist for Cryptographic Function *UDX=User Defined Extensions

ICSF Product



HCR7751

- Runs on z/OS V1.8 through V1.10
- Runs on z800, z900, z890, z990, z9 EC, z9 BC, z10 EC, z10 BC

- But you can only use the functions that are supported by the hardware
  - Clear Key AES-128 requires z9; Clear Key AES-192, AES-256 require z10
  - SHA-2 (SHA-512, SHA-384 requires z10)
  - Secure key AES requires z10 (EC or BC)
  - 4096-bit RSA keys require MCL on CEX2

ATS Technote has been published that discusses HCR7751 highlights – PRS3472

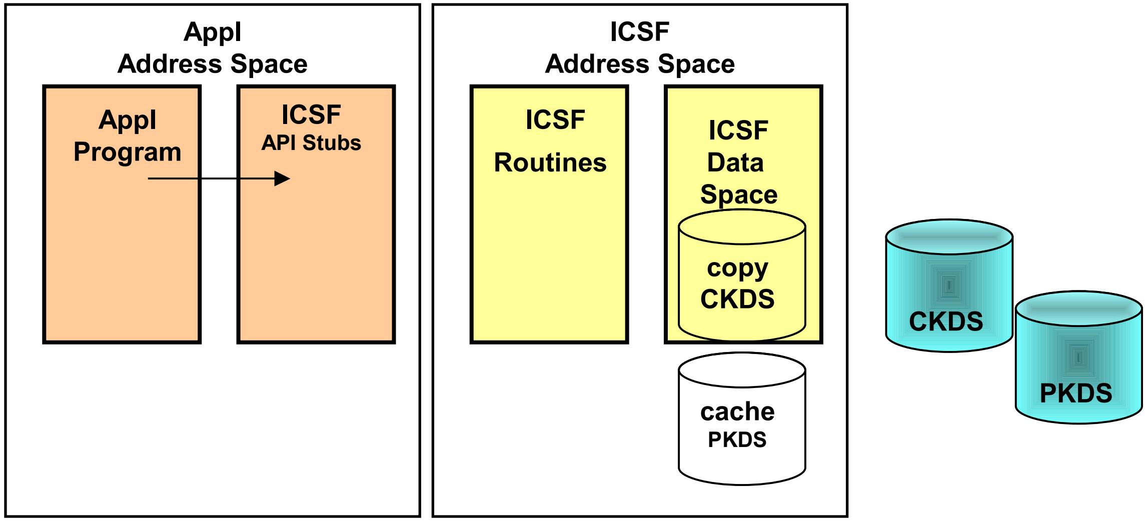
ICSF – Integrated Cryptographic Service Facility

No charge component of zOS (FMID HCR77xx)

A {set of APIs} callable from COBOL, C++, ASM for:

- Symmetric encryption/decryption routines
- Asymmetric encryption/decryption routines
- Random number generator
- Hashing/message digests
- Pin generation
- Key generation

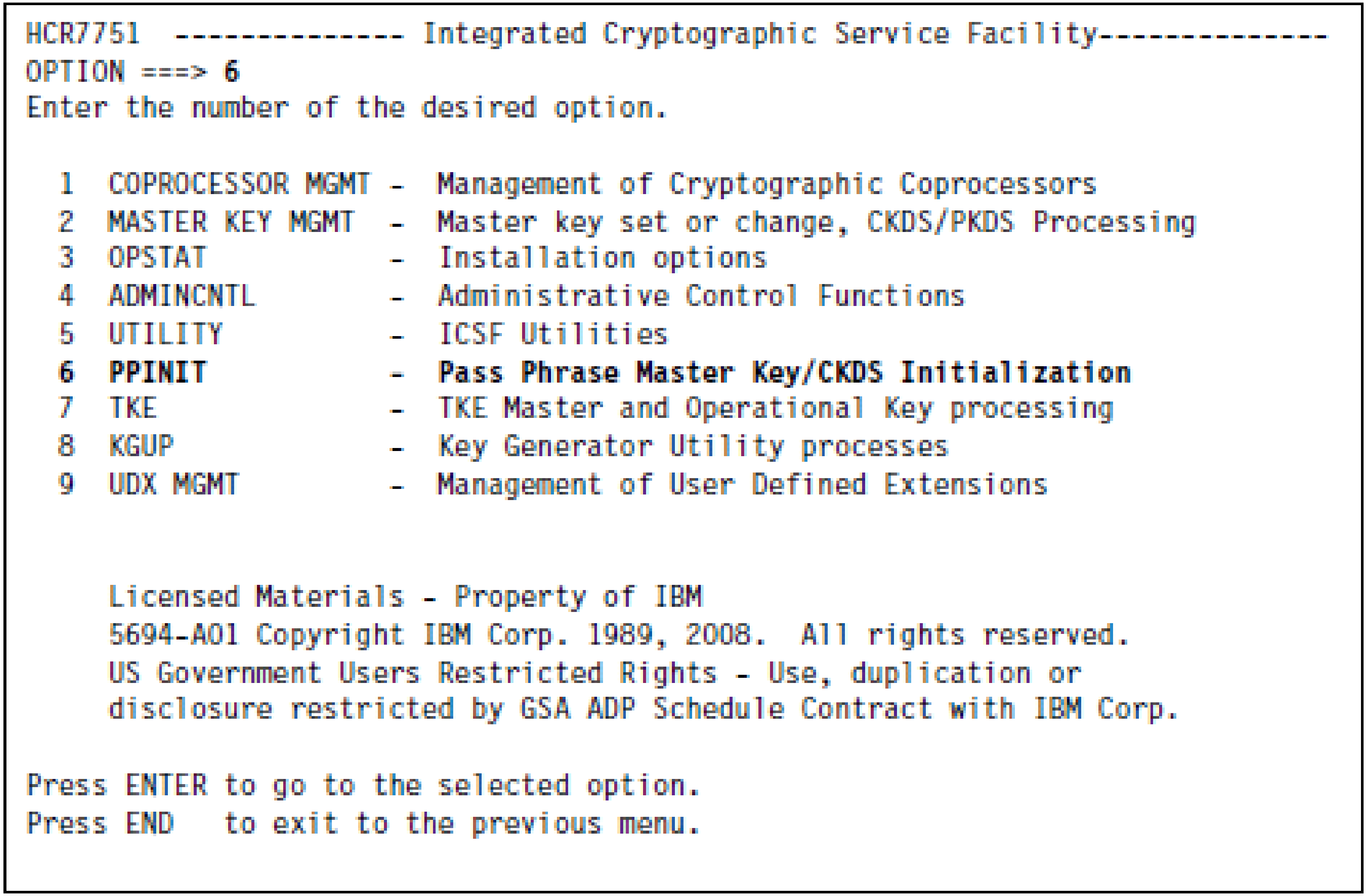
- Algorithms are full standard open implementations per FIPS 140-2 and NIST
- APIs will schedule hardware to offload cryptographic work from standard CPs
- Operational keys maintained in VSAM Linear Data Sets
  - Keys retrieved by key label (vs memorizing bizarre hex strings)



CKDS – Cryptographic Key Dataset

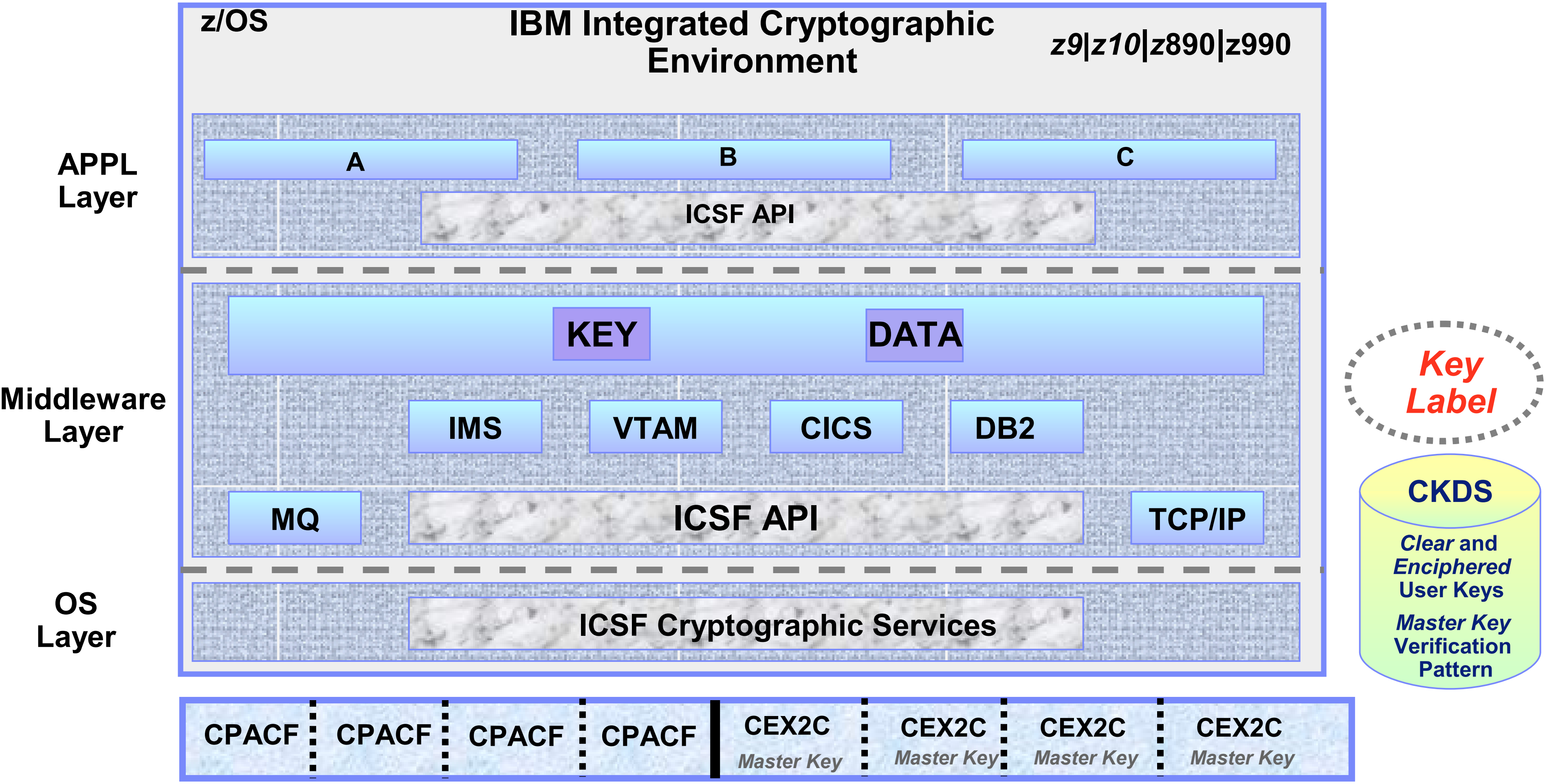
- Key element of the IBM encryption solution on z/OS
- VSAM Key Sequenced Dataset
- Contents are ICSF generated data encrypted keys
- Accessed by ICSF API and Services
  - Key Label (known by application requestor) used to find key record in the CKDS
- Copy of CKDS cached in operating system storage at first ICSF invocation for performance
- CKDS administration performed using ICSF services and ISPF interfaces.
- Use of specific individual keys can be controlled via RACF profiles and permissions
- CEX2C hardware feature required for use……
  - Except with a combination of HCR7751 and clear key only, then CEX2C is optional

ICSF – Integrated Cryptographic Service Facility



```
HCR7751  -------------- Integrated Cryptographic Service Facility--------------
OPTION ===> 6
Enter the number of the desired option.

  1  COPROCESSOR MGMT -  Management of Cryptographic Coprocessors
  2  MASTER KEY MGMT  -  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
     5694-A01 Copyright IBM Corp. 1989, 2008.  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.
```

IBM Encryption Flow



CP Assist for Cryptographic Functions

- Problem State Instructions
- Clear Keys Only
- DES/TDES Encryption
- AES (128 Bit)
- SHA-1 (256 on z9)

Crypto Express 2 Coprocessor

- ICSF Access Only (Key 0)
- Master Key Stored Within Crypto Express 2 Feature
- Secure Key DES/TDES Encryption
- SSL Accelerator
- Tamper Resistant

System z9 Cryptographic Support Summary

CP Assist for Cryptographic Function (CPACF) “free”

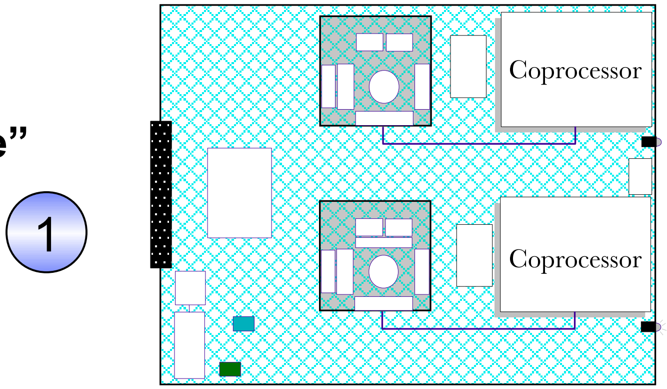
- Supports DES, TDES and SHA-1
- Standard on System z9/z10 (feature code 3863)
- Standard on every CP and IFL
- Advanced Encryption Standard (AES)
- Secure Hash Algorithm – 256 (SHA-256)
- Pseudo Random Number Generation (PRNG)

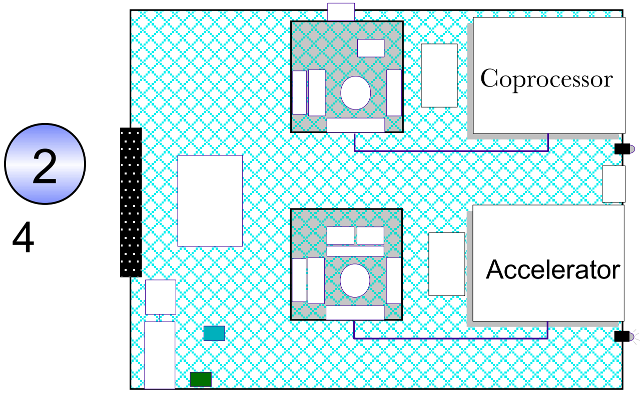
Crypto Express2 (feature code 0853) “fee”

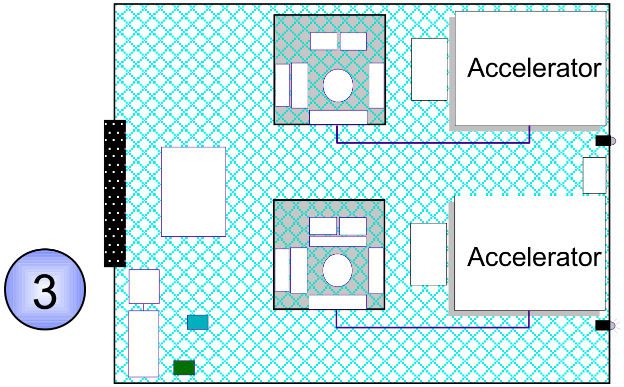
- Two configuration modes
- Coprocessor (default)
- Federal Information Processing Standard (FIPS) 140-2 Level 4 certified
- “Tamper Resistant”
- (Secure Key) – “Exclusive”
- SSL Accelerator (Handshake offload)

Three configuration options

- Default set to Coprocessor (1)
- SSL Acceleration (3)
- Mixture of configuration (2)







What are Keys? (An ICSF Perspective)

- DES Master Key
  - Loaded into the CEX2C hardware, and stored NOWHERE else
  - Used to generate, encrypt, and store user keys into the CKDS (Cryptographic Key Data Set)
- User Keys (Data Encrypting Keys)
  - Generated via ICSF services
  - Used by the **IBM Encryption Tool** along with encryption algorithm to convert user data to cybertext
  - Stored inside the CKDS
  - Clear or Secure

Cryptography on z/OS

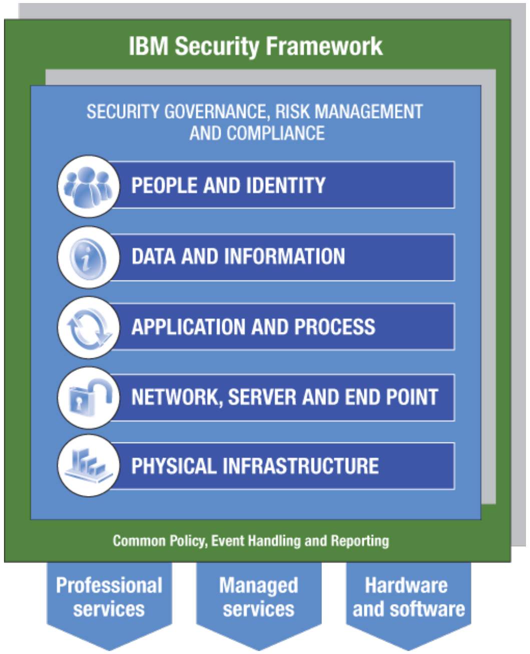
- Clear Key
  - Key is exposed in the storage of processor
  - Can be viewed in dump of storage
  - If correctly interpreted can expose data
  - Sometimes acceptable for short-lived keys with other constraints
  - Used in software based cryptography
  - Used by CPACF
  - Used by Crypto Express 2 (Configured as CEX2A)

- Secure Key
  - Key is only ever exposed in bounds of a secure processor
  - Can never be seen in storage
  - Dump will not reveal key
  - Key is held encrypted under Master key
  - Crypto Express 2 (Configured as CEX2C) provides this function for System z
  - APIs available via Integrated Cryptographic Support Facility (ICSF)
  - Can be used from Java on z/OS platform

Prior to the introduction of z10/CEX3C protected key option, we recommended Clear key encryption due to performance characteristics of Secure Key, we are now changing that recommendation (if clear key is viewed as “weak”).

Some general comments on secure/clear key

- **Clear Key vs. Secure Key Performance**
  - Clear key elapsed time performance is **MUCH** superior than secure key
  - Secure key (performed inside the CEX2C) is generally viewed as more secure from a cryptographic perspective
  - Clear key uses special instructions that run on the z9 – z10 general purpose processors, so performance is measured in milliseconds
  - Secure key encryption is dispatched to run on the cryptographic coprocessors on the CEX2C crypto feature. This tends to be measured in microseconds as this is essentially an I/O operation.
  - Secure key elapsed time measurements (depending on workload and SQL type) can be from 10x to 40x worse than clear key
  - Secure key is probably **NOT** appropriate for most (to date all) OLTP workloads, but each customer needs to make this encryption decision based on their security requirements and performance expectations



Part 2 – IBM Encryption Tool for IMS and DB2 Databases

IBM Data Encryption for IMS and DB2 Databases 1.01 (5655-P03)

Standard DB2 EDITPROC for Accessing Cryptographic Functions

- All Supported DB2 Versions
- Member of IBM IMS | DB2 Tools Family of Products
- Pre-coded EDITPROC for encryption of DB2® Data
- Encryption/Decryption occurs at the DB2 Row Level
- Unique EDITPROC can be defined for each DB2 Table
- Exploits z/OS Integrated Cryptographic Service Facility (ICSF)
- Exploits zSeries CPACF Cryptographic Hardware Directly
- Requires no changes to your applications
- Fast implementation

Edit Procedures (EDITPROC) are Programs That:

- Transform Data on INSERT | UPDATE | LOAD
- Restore Data to Original Format on SELECT
- Transformations on Entire ROW
- Supported by Utilities
- Implemented via Create Table specification
- Requires unload/load of data

IBM Data Encryption for IMS and DB2 Databases
Implementation Summary

Configure the Integrated Cryptographic Service Facility (ICSF)

Enable CP Assist for Cryptographic Functions (CPACF) (z890/z990/z9/z10)
(FC 3863 - This Feature subject to US Export Restrictions)

Install and enable CEX2C (Crypto Express 2) feature
(FC 0863 – Chargeable feature)

Generate and store in the Cryptographic Key Data Set (CKDS) Key Labels

Build the IMS User Exit or DB2 EDITPROC

Generate Data Encryption Key with ICSF ISPF

Obtain Key Label from ICSF Administrator

Use the Sample JCL Provided or the ISPF Panels to generate EDITPROC

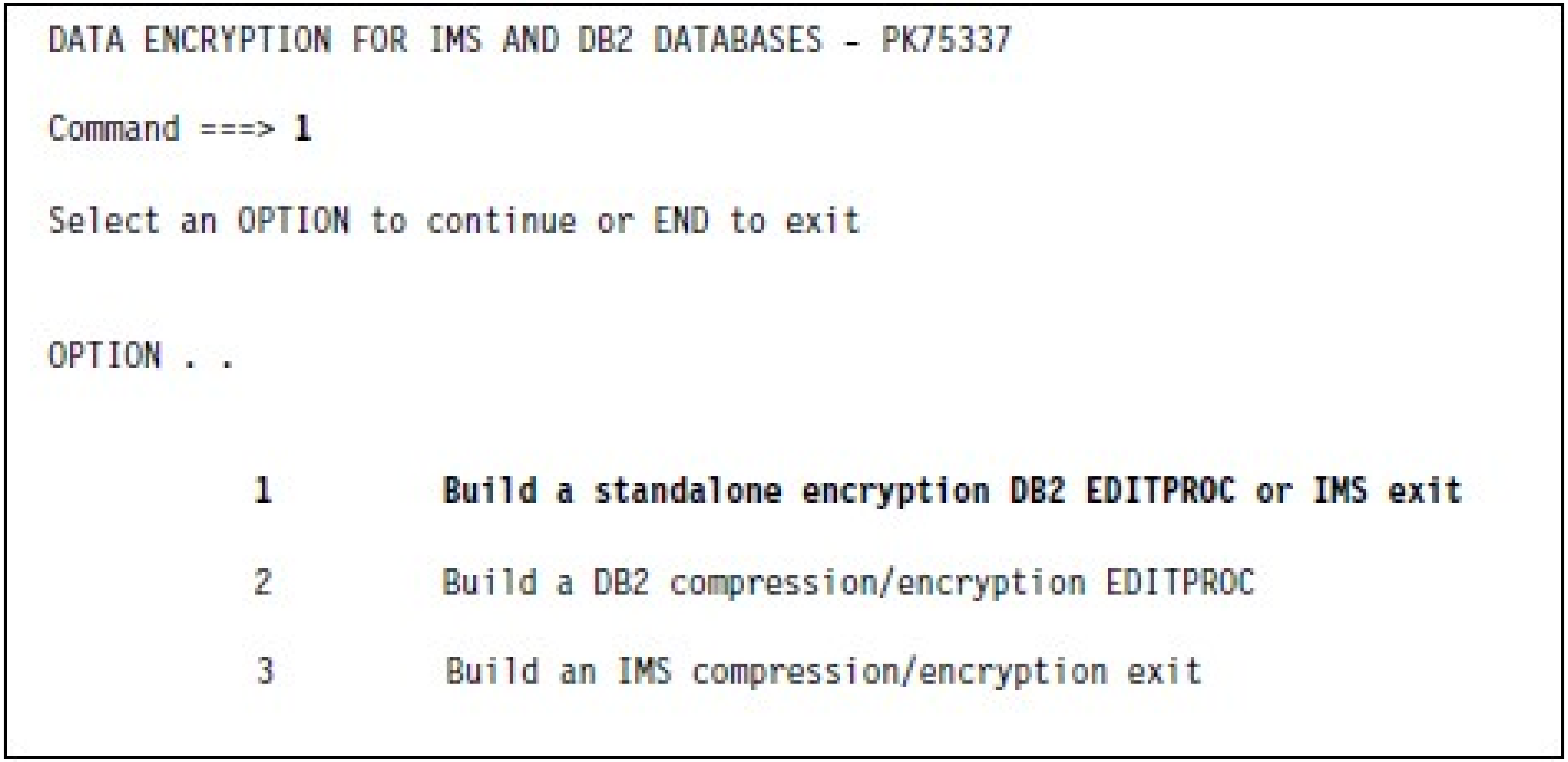
Back - Up and Unload Databases

Create Exits for IMS or EDITPROCS for DB2

Reload the Databases: Data Bases will be Encrypted

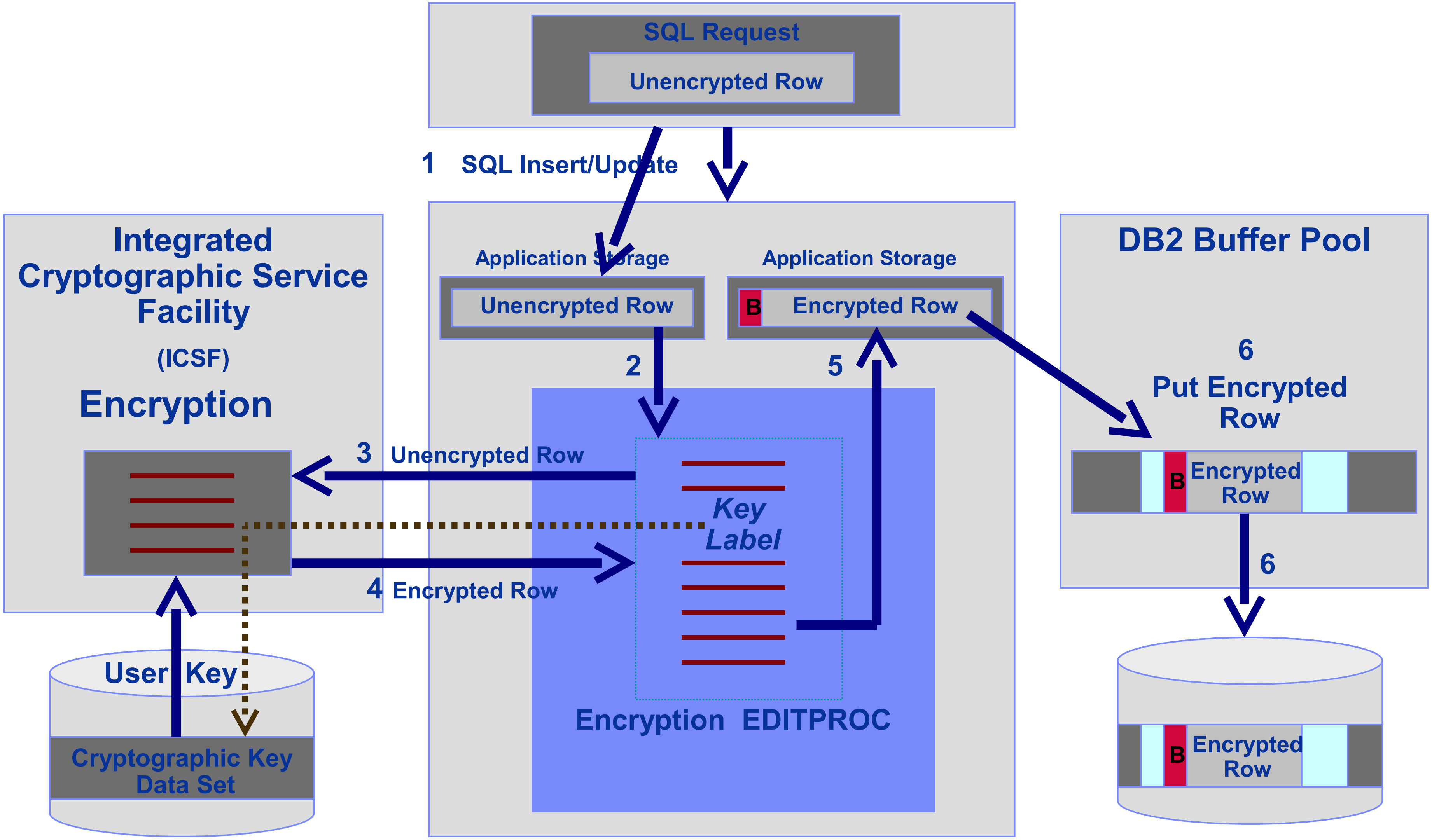
Validate your Output

Main menu for Data Encryption for IMS and DB2 Databases Tool



```
DATA ENCRYPTION FOR IMS AND DB2 DATABASES - PK75337

Command ===> 1

Select an OPTION to continue or END to exit


OPTION . .


          1         Build a standalone encryption DB2 EDITPROC or IMS exit

          2         Build a DB2 compression/encryption EDITPROC

          3         Build an IMS compression/encryption exit
```

DB2 Data Encryption Flow – Insert / Update



Compression support for Encryption Tool

- PTFs UK41354 - V8 and UK41355 – V9
- A new option EXTNDICT has been added for DSN1COMP.   EXTNDICT specifies the 8 character name of the link editable object deck built from the DSN1COMP-created compression dictionary.
- Specifying option EXTNDICT requires to also provide a   DSN1DICT DD statement in the DSN1COMP Job. DSN1DICT defines the output data set to which the generated object module is written and stored for follow-on processing
- Remember, if this is an encrypted table, you need to unload and then reload before running DSN1COMP.
- Also, ensure that the COMPRESS attribute for the associated tablespace is turned off

```
//BUILD    EXEC PGM=DSN1COMP,
// PARM='DSSIZE(4G),EXTNDICT(dictname),ROWLIMIT(99999)'
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD DSN=DSNCAT.DSNDBD.DBIA2401.TPIA2401.I0001.A254,
//            DISP=SHR
//DSN1DICT DD DSN=&&OBJ,
//         DISP=(,PASS),
//         UNIT=SYSALLDA,SPACE=(TRK,(8,4)),
//         DCB=(LRECL=80,BLKSIZE=4000,RECFM=FB)
```

Compression support for Encryption Tool – sample output

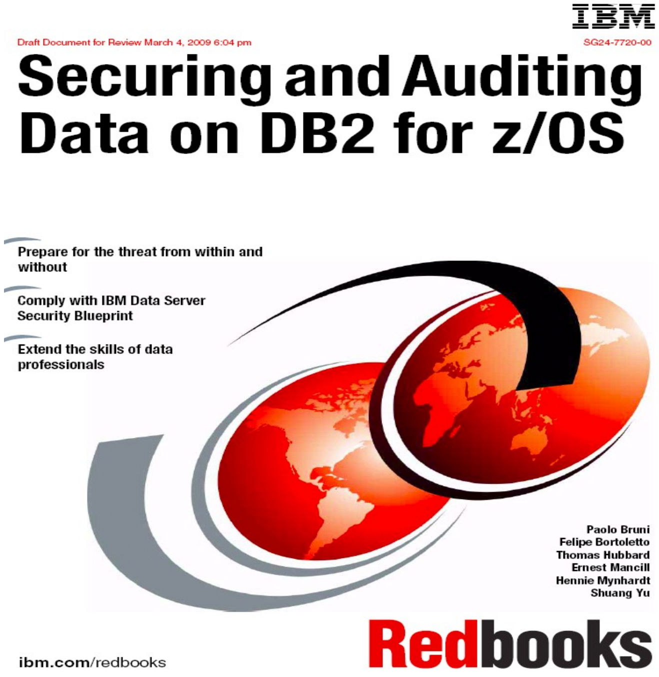
```
DSN1940I DSN1COMP COMPRESSION REPORT
           2,441  KB WITHOUT COMPRESSION
           2,034  KB WITH COMPRESSION
              16  PERCENT OF THE BYTES WOULD BE SAVED

           7,743  ROWS SCANNED TO BUILD DICTIONARY
          99,999  ROWS SCANNED TO PROVIDE COMPRESSION ESTIMATE
           4,096  DICTIONARY ENTRIES

              27  BYTES FOR AVERAGE UNCOMPRESSED ROW LENGTH
              23  BYTES FOR AVERAGE COMPRESSED ROW LENGTH

              16  DICTIONARY PAGES REQUIRED
             700  PAGES REQUIRED WITHOUT COMPRESSION
             612  PAGES REQUIRED WITH COMPRESSION
              12  PERCENT OF THE DB2 DATA PAGES WOULD BE SAVED

DSN1937I DSN1COMP TXT-DECK DSN1DICT BUILT          1,173  RECORDS WRITTEN
DSN1994I DSN1COMP COMPLETED SUCCESSFULLY,             670  PAGES PROCESSED
```

Encryption Tool for DB2 and Editproc Restrictions

- EDITPROC based restrictions planned to be removed
  - DB2 V9 to lift restrictions
    - Long column names
    - Data Types - XML
    - IDENTITY
    - ROWID
    - Alter Add Columns
  - Still Restricted
    - LOB
    - SECLABEL
  - Still need unload/drop/create/load to initially encrypt data
  -

IBM Encryption Tool for DB2 - Utilities

- So, what about Utilities?
- In general if your utility access is against the LVDS directly, you might have issues:
  - IBM DB2 Offline Utilities – such as DSN1PRNT, DSN1COPY would show the encrypted data
  - Third party tools and utilities might show unpredictable results
- All IBM Online utilities are supported by Encryption Tool for DB2
  - IBM Utilities invoke the DB2 Data Manager and don't access the page data from the underlying VSAM datasets directly
- e.g. HIGH Performance Unload is supported
  - DB2 HPU will drive edit procedure


IBM

Draft Document for Review March 4, 2009 6:04 pm

SG24-7720-00

Securing and Auditing Data on DB2 for z/OS

- Prepare for the threat from within and without
- Comply with IBM Data Server Security Blueprint
- Extend the skills of data professionals

Paolo Bruni
Felipe Bortoletto
Thomas Hubbard
Ernest Mancill
Hennie Mynhardt
Shuang Yu

Redbooks

ibm.com/redbooks