88. GSE z/OS ExpertenForum
Pervasive Encryption How To

18. April 2018

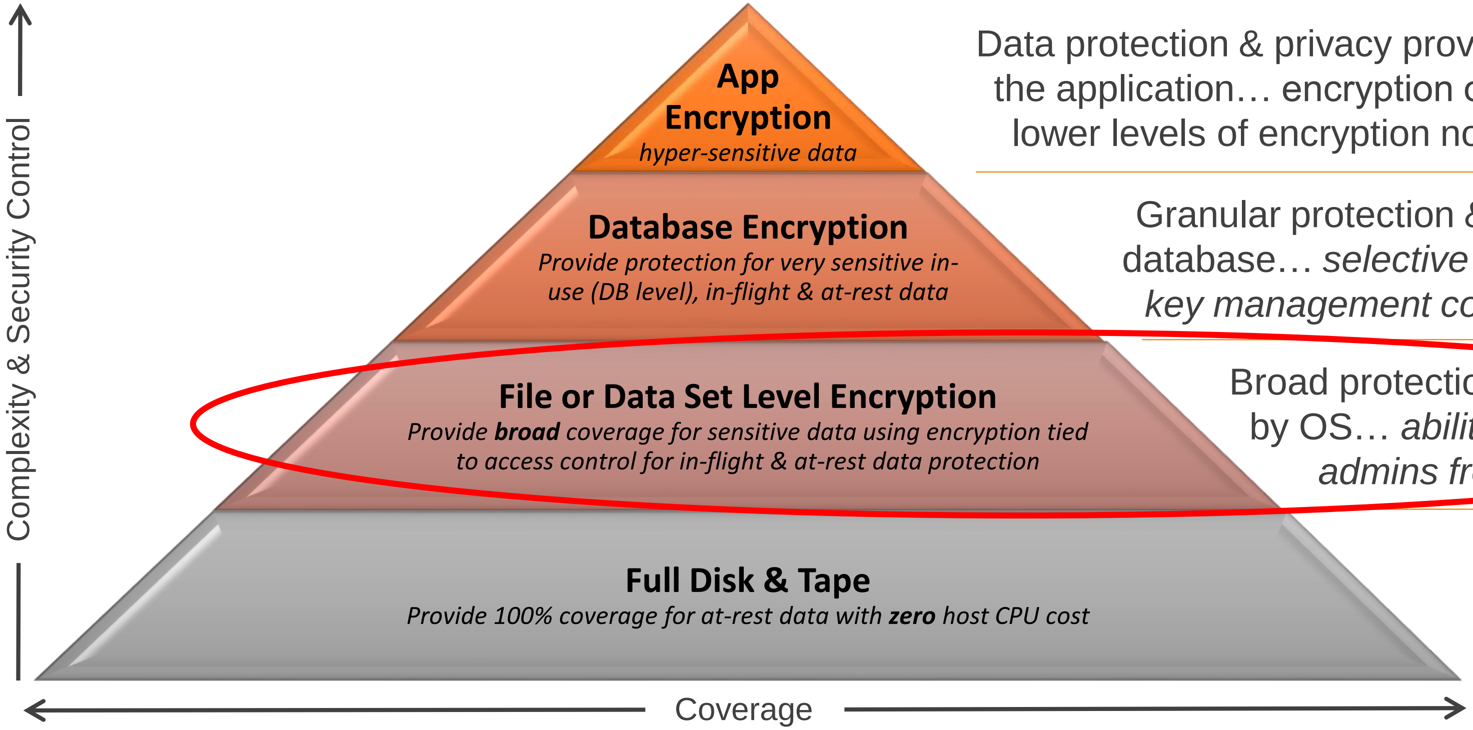
Heinz Tschumi
STG Switzerland
htsi@ch.ibm.com

Agenda

- **Pervasive Encryption**
  - Role of z/OS data set encryption
  - Getting Started
- **Level of Protection (HW and SW Support)**
- **Key Management**

Multiple layers of encryption for data at rest

Robust data protection



App Encryption
hyper-sensitive data

Database Encryption
Provide protection for very sensitive in-use (DB level), in-flight & at-rest data

File or Data Set Level Encryption
*Provide **broad** coverage for sensitive data using encryption tied to access control for in-flight & at-rest data protection*

Full Disk & Tape
*Provide 100% coverage for at-rest data with **zero** host CPU cost*

Data protection & privacy provided and managed by the application… encryption of sensitive data when lower levels of encryption not available or suitable

Granular protection & privacy managed by database… *selective encryption & granular key management control of sensitive data*

Broad protection & privacy managed by OS… *ability to eliminate storage admins from compliance scope*

Protection against intrusion, tamper or removal of *physical* infrastructure

Pervasive Encryption with IBM Z

Enabled through tight platform integration

Integrated Crypto Hardware	Hardware accelerated encryption on every core, CPACF performance improvements of 7x Crypto Express6S – PCIe Hardware Security Module (HSM) & Cryptographic Coprocessor
Data (at Rest)	Broadly protect z/OS data sets and Linux file systems using policy controlled encryption that is transparent to applications and databases
Clustering	Protect z/OS Coupling Facility data end-to-end, using encryption that's transparent to applications
Network	Protect network traffic using standards based encryption from end to end, including encryption readiness technology to ensure that z/OS systems meet approved encryption criteria
Secure Service Container	Secure deployment of software appliances including tamper protection during installation and runtime, restricted administrator access, and encryption of datThe a and code in-flight and at-rest
Key Management	The IBM Enterprise Key Management Foundation (EKMF) provides real-time, centralized secure management of keys and certificates with a variety of cryptographic devices and key stores

And we're just getting started …

Comparison of data at rest encryption

Full Disk Encryption

- Protects at the DASD subsystem level
- **All or nothing encryption**
- Only data at rest is encrypted
- Single encryption key for everything
- **No application overhead**
- **Prevents exposures on**
  - **Disk removal**
  - **Box removal**
  - **File removal**

z/OS Data Set Encryption

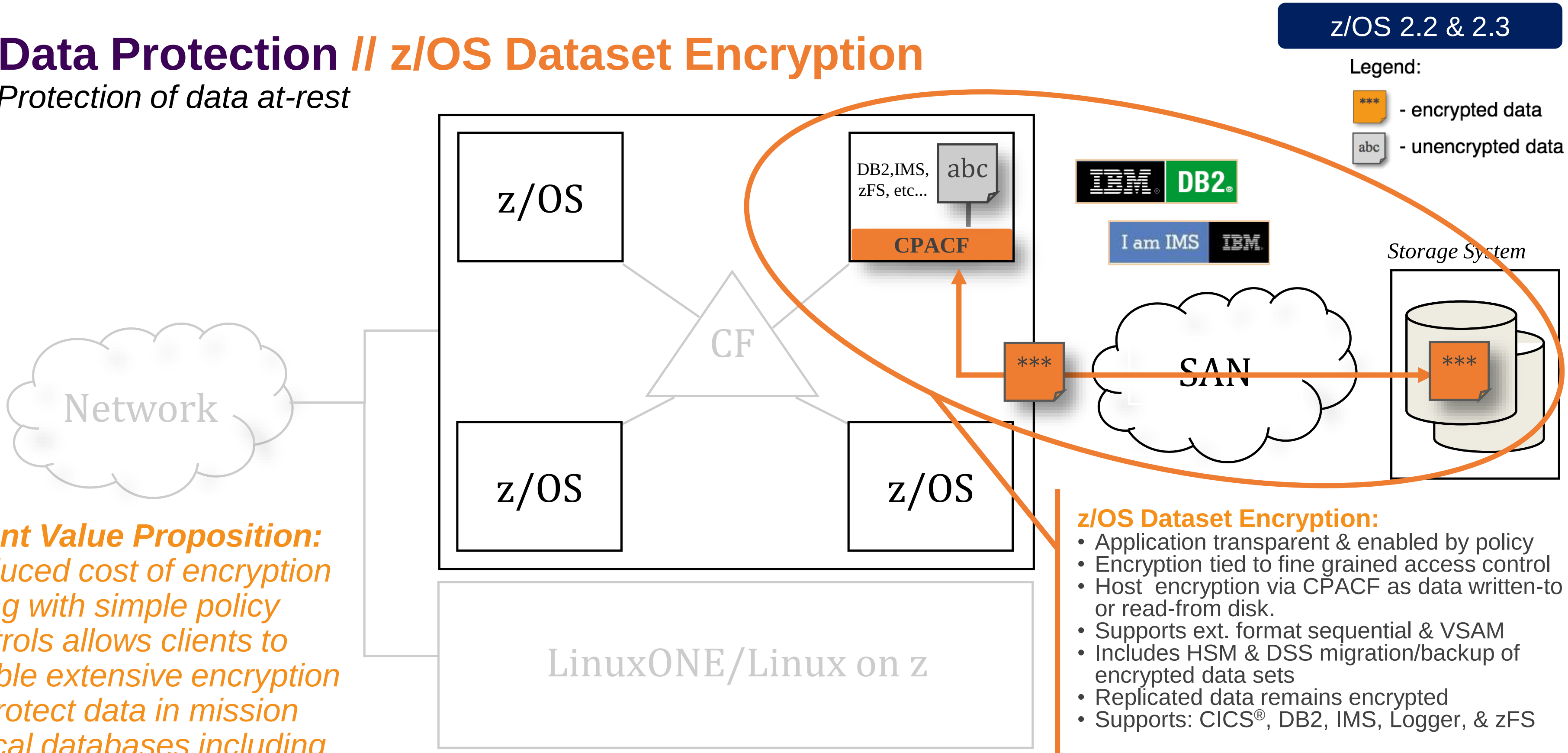
- Broadly encrypt data at rest
- Covers VSAM, DB2, IMS, Middleware, Logs, Batch, & ISV Solutions
- **Transparent to applications**
- Encryption …
  - **By policy**
  - **Tied to access control**
  - **Keys controlled by host**
- Encrypt in bulk for low-overhead
- Prevents exposures on
  - Mis-identification or mis-classification of sensitive data
  - Compliance findings related to unencrypted data

z/OS Database Encryption

- Data remains encrypted inside the database
- **Data in memory buffers is also protected**
- **Very flexible key granularity**
  - Down to the row and column level for DB2
  - Segment level for IMS
- **Excellent separation of duties**
- **Transparent to applications**
- Prevents exposures on
  - Unauthorized viewing of encrypted sensitive data
  - Non-DBMS data access
  - Unauthorized access to DBMS generated datasets

Data Protection // z/OS Dataset Encryption

Protection of data at-rest



Client Value Proposition:
Reduced cost of encryption along with simple policy controls allows clients to enable extensive encryption to protect data in mission critical databases including DB2®, IMS™ and VSAM

z/OS Dataset Encryption:
- Application transparent & enabled by policy
- Encryption tied to fine grained access control
- Host encryption via CPACF as data written-to or read-from disk.
- Supports ext. format sequential & VSAM
- Includes HSM & DSS migration/backup of encrypted data sets
- Replicated data remains encrypted
- Supports: CICS®, DB2, IMS, Logger, & zFS

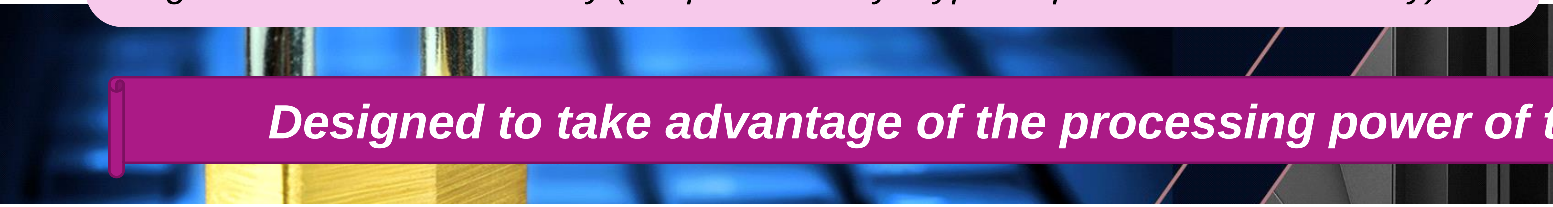
In-memory system or application data buffers will not be encrypted

z/OS Data Set Encryption – Client Value

*Clients who are required to protect customer data can leverage the z Systems hardware encryption for **data at rest** through existing **policy management**… **without application changes.***

- 1 – No application changes required
- 2 – Data set level granularity
- 3 – Supports separation of access control for data set and encryption key label
- 4 – Enabled through RACF and / or SMS policy
- 5 – Audit readiness

***Key label:** 64-byte label of an existing key in the ICSF CKDS used for access method encryption/decryption.*
***Encryption type:** AES-256 bit key (XTS, protected key). Note: AES-256 key must be generated as a secure key (i.e. protected by crypto express AES Master Key)*



Designed to take advantage of the processing power of the z14

Application transparency via access methods

Supported access methods/data set types

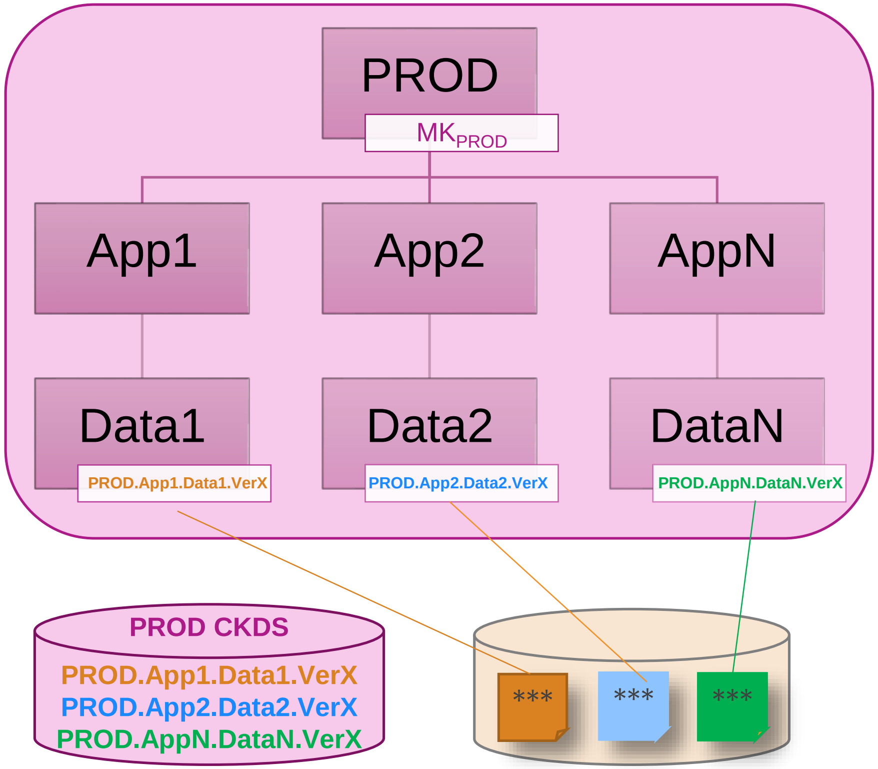
- **BSAM/QSAM**
  - **Sequential data sets**
    - **Extended format only**
- **VSAM and VSAM/RLS**
  - **KSDS, ESDS, RRDS, VRRDS, LDS**
    - **Extended format only**

Transparent! No application changes or awareness that sequential or VSAM data is encrypted when accessed using the standard access method APIs.

Covers DB2, IMS, zFS, CICS/VSAM, Middleware, Logs, Batch, & ISV Solutions. Refer to product documentation for information regarding support.

2 Naming Conventions & Granular Access Control

Leveraging naming conventions & z Security to enforce separation across application instances



- Naming conventions can be used to segment applications, data, and keys, e.g.
  - Environment: PROD, QA, TEST, DEV
  - Application: App1, App2,…, AppN
  - Data-Type: Account, Payroll, Log
  - Version: Ver1, Ver2,…,Verx
- Application resources (data sets, encryption keys) can be assigned names based on naming conventions, e.g.
  - PROD.APP2.LOG.VER10
  - PROD.APP1.PAYROLL.KEY.VER7
- Security rules can be used to enforce separation with granular access control for application resources and encryption keys

Flexible! Data set encryption is designed to be flexible in allowing as much granularity as desired when identifying key labels for data sets. There is no limit as to how many key labels and encryption keys are used across the data sets…however, planning for key management is critical.

Access Control - Segregation of Duties

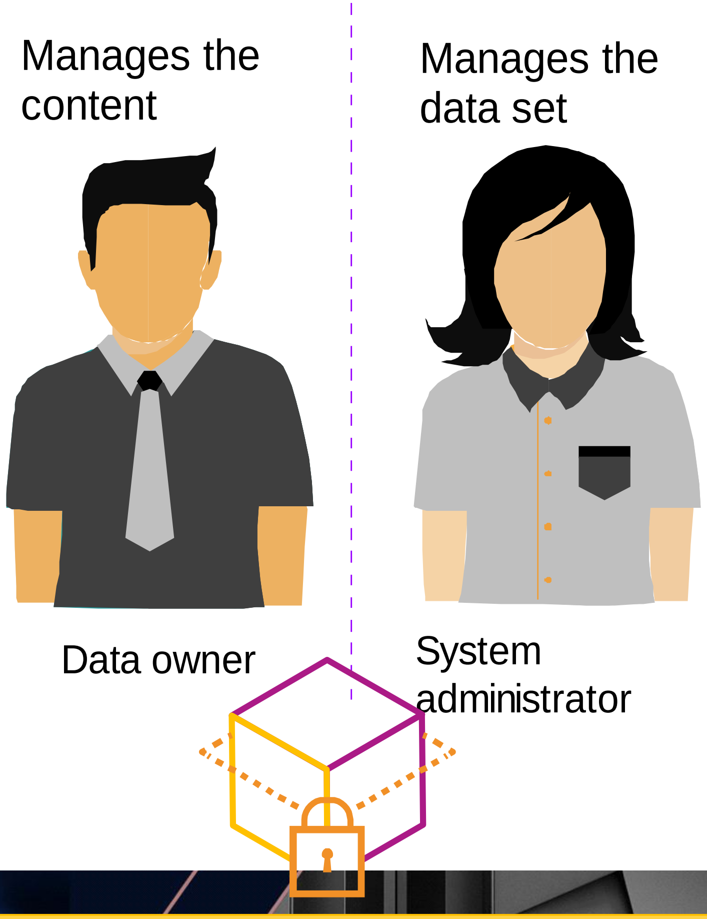
- Data owners that ***must* access content** will need authority access to the data set ***as well as*** access to the encryption key label

- Storage administrators who only ***manage* the data sets** need access to the data set ***but not*** access to the key label (thus protecting access to the content)

- **Different keys can be used to protect different data sets – ideal for multiple tenants or data set specific policies.**

- Prevent administrators from accessing the content

- **Many utilities can process data preserving encrypted form**
  - COPY, DUMP and RESTORE
  - Migrate/Recall, Backup/Recover, Dump/Data Set Restore
  - PPRC, XRC, FlashCopy®, Concurrent Copy, etc.



Limit access to data in clear! Remove certain roles from compliance scope….by controlling access to the data through policies.

Creating encrypted data sets via policy

A data set is defined as ‘encrypted’ when a **key label** is supplied on allocation of a ***new*** sequential or VSAM extended format data set

A **key label** supplied in any of the following (using ***order of precedence*** as follows):

- **RACF Data set profile DFP segment**
- **JCL, Dynamic Allocation, TSO Allocate, IDCAMS DEFINE**
- **SMS Construct: Data Class**
  - **Note: Can specify data class on ISPF 3.2 to allocate an encrypted data set**

Ease of use! Easy to create an encrypted data set just by specifying a key label. Even easier when enabled via RACF or SMS policy.

4

New data set allocation via policy based storage management

- **DFSMS Storage Management Subsystem (SMS) derives key label (from one or more sources) to be used for the *encrypted data set***
  - **Derived key label stored in Catalog**
    - New encryption cell (non-VSAM NVR, VSAM VVR)

      64-byte Key label; Encryption type (AES256); Encryption mode (XTS); ICV; Key verification value

> *Once key label stored in catalog for a data set, no ability to alter it.* ***Any subsequent change to RACF Data set profile or Data Class will not affect existing data sets***

  - **Encryption indicator set in volume table of contents (VTOC)**
    - Format 1/Format 8 DSCB flag (DS1ENCRP)
  - **New allocation message indicating data set is an *encrypted data set* with derived key label**

```
IGD17150I DATA SET dsname IS ELIGIBLE FOR ACCESS METHOD ENCRYPTION.
KEY LABEL IS (key_label)
```

5

Audit readiness

Auditor can rely on system interfaces, not individuals, for compliance.

- Encryption attributes displayed in various system interfaces
  - SMF records
  - DCOLLECT records
  - LISTCAT
  - IEHLIST LISTVTOC

Simplifies compliance! Allows enhanced tooling to help simplify the audit process.

zSecure 2.3 Pervasive Encryption Support

Command Verifier: Command Verifier policy for DATAKEY

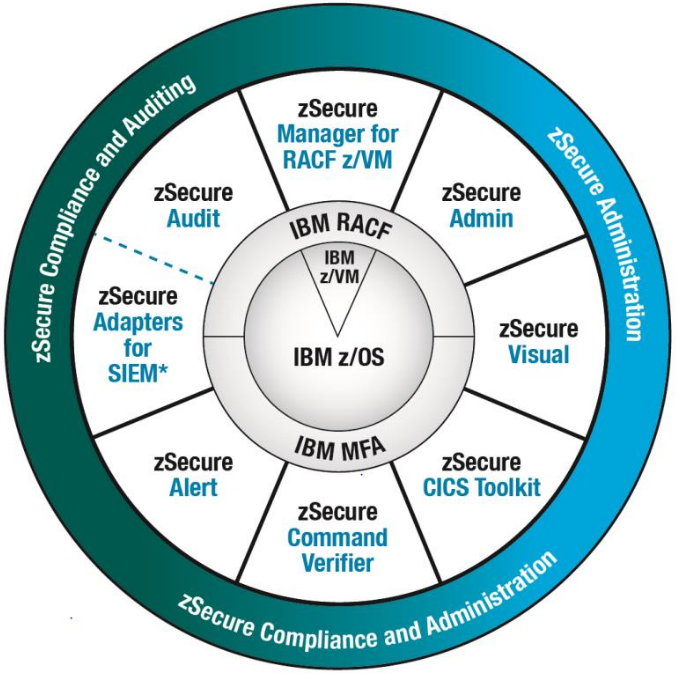
Admin: Easy administration DATAKEY on DFP segment

Audit: Report on non-VSAM and VSAM data sets key labels
- Extend existing report types DSN / SENSDSN

Audit: Report key protection CSFKEYS
- New report types ICSF_SYMKEY, ICSF_PUBKEY

Audit: Report which systems sharing DASD can decrypt ds

Audit: Extend report type SMF
- Type 14/15 non-VSAM and Type 62 VSAM keylabel use
- ICSF
- zERT records to show encryption strengths



zSecure also collects, formats and enriches data set encryption information that is sent to SIEMs including IBM QRadar® for enhanced enterprise-wide security intelligence.

Enhanced tooling simplifies the audit process.

Considerations for data set encryption usage

Extended format data sets

- Allocated with DSNTYPE keyword
  - *JCL **DSNTYPE=EXTREQ or EXTPREF***
  - *SMS Data class **DSNTYPE=EXTR or EXTP***
- SMS-managed DASD data sets
- Can be compressed format
  - *SMS Data class **COMPACTION***
    - ***Sequential: Generic, Tailored, zEDC***
    - ***VSAM KSDS: Generic***
- Restrictions
  - System data sets (such as Catalogs, SHCDS, HSM data sets) should not be created as extended format, unless otherwise specified.
  - Cannot be opened for EXCP processing
  - Sequential compressed format data sets cannot be opened for UPDATE processing

❖Data set types that are ***not*** extended format
- ❖Basic and Large format sequential
- ❖PDS/PDSE
- ❖BDAM
- ❖Tape data sets

❖Note: The following sequential data sets **cannot be** extended format
- Temporary data sets
- SORTWK data sets

❖Data sets that **can be** allocated as extended format
- ❖Db2 (table spaces and logs)
- ❖IMS (certain dbs, logs, trace data sets)
- ❖CICS/VSAM
- ❖zFS
- ❖Etc

Note: Review product documentation for support.

Data set encryption requires extended format

Data set encryption restrictions

- System data sets (such as Catalogs, SHCDS, HSM data sets) must not be encrypted, unless otherwise specified

- Data sets used before ICSF is started must not be encrypted

- Sequential (non-compressed) extended format data sets with a BLKSIZE of less than 16 bytes cannot be encrypted

- Encrypted data sets only supported on 3390 device types

Consider enabling data set level compression

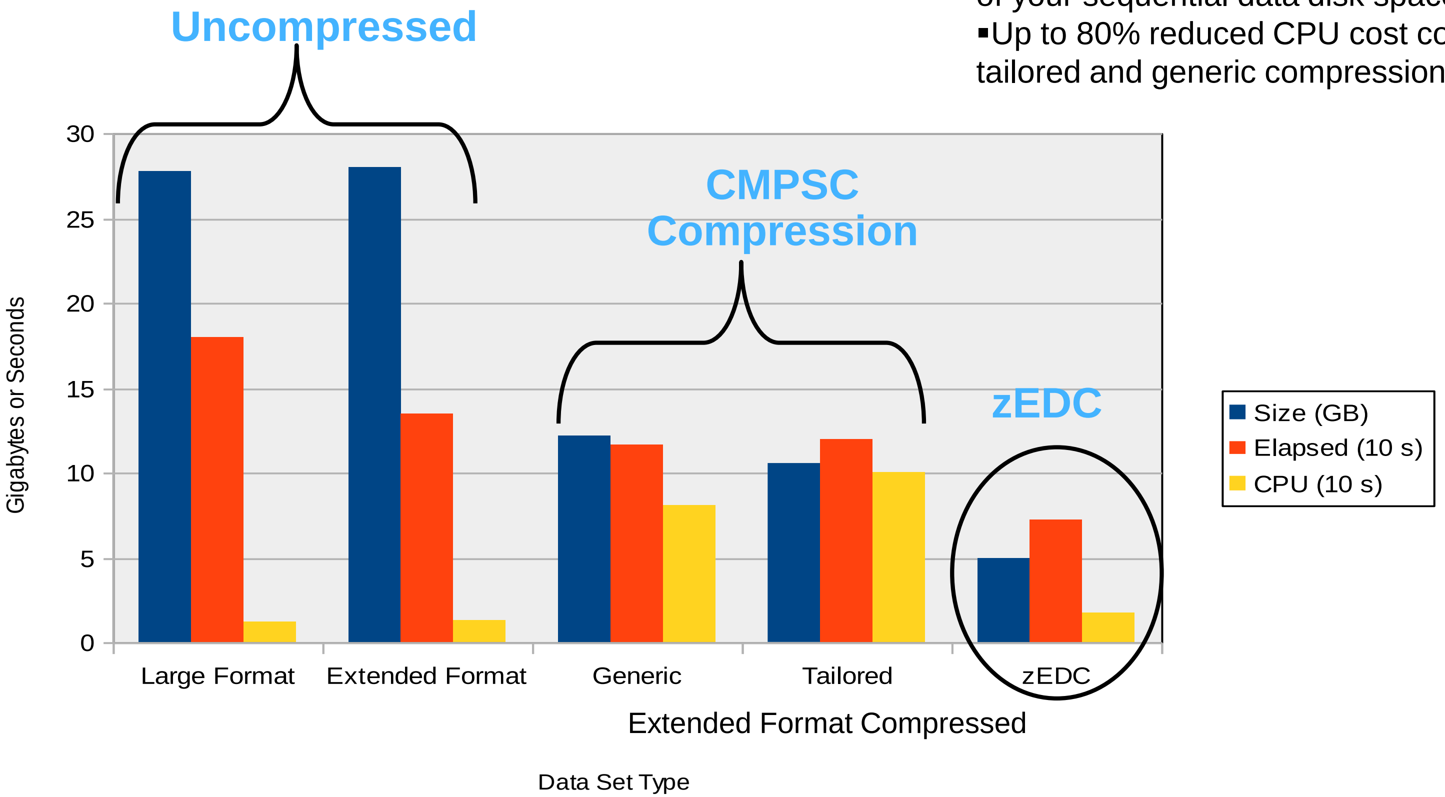
- **Encrypted data does not compress**
  - Creating encrypted data sets may impact expected savings with disk or tape device compression.
  - Backup and migration of encrypted data sets may impact expected savings with disk or tape device compression.
  - Replicated data that is being compressed in the SAN infrastructure by DWDM technology will no longer be effective trying to compress encrypted data

Where possible, convert to compressed format data sets

- When data set level compression requested, access methods handle compression before encryption for compressed format encrypted data sets.
  - **Data class COMPACTION option**

Sequential data compression with zEDC

- Compresses data up to 4X, saving up to 75% of your sequential data disk space
- Up to 80% reduced CPU cost compared to tailored and generic compression options



Backup, Migration and Replication

- System services that manage the data set (as opposed to the data) ensure the data remains in encrypted form
  - During DFSMSdss functions, COPY, DUMP and RESTORE
  - During DFSMShsm functions, Migrate/Recall,Backup/Recover, Abackup/Arecover, Dump/Data Set Restore, FRBACKUP/FRRECOV DSNAME.
  - During track based copy (PPRC, XRC, FlashCopy, Concurrent Copy, etc) operations since read track will get the track image which has the already encrypted data.
    - The recovery system must have the same key information.

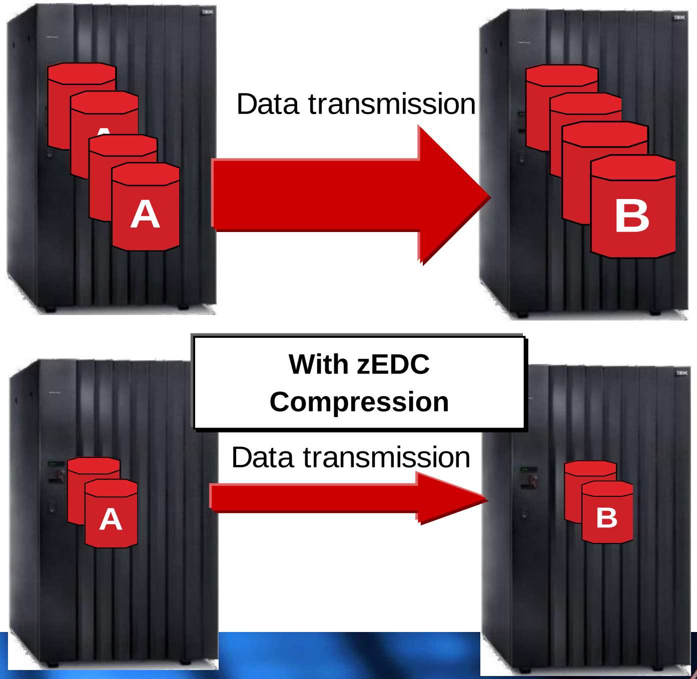
Data remains encrypted as it migrates to the cloud with Transparent Cloud Tiering

Storage admins (or others) that perform these system services would not require access to the key label.

Data Replication

Replication technologies which move data in physical format maintain data in encrypted (and compressed) format

- Take advantage of the reduced storage requirements with data compression
  - For sequential data sets, zEDC compression recommended to significantly reduce the amount of data transferred as well as the elapsed time to complete the transfer.



Key material must be available on target systems to access encrypted data sets

Transmitting data

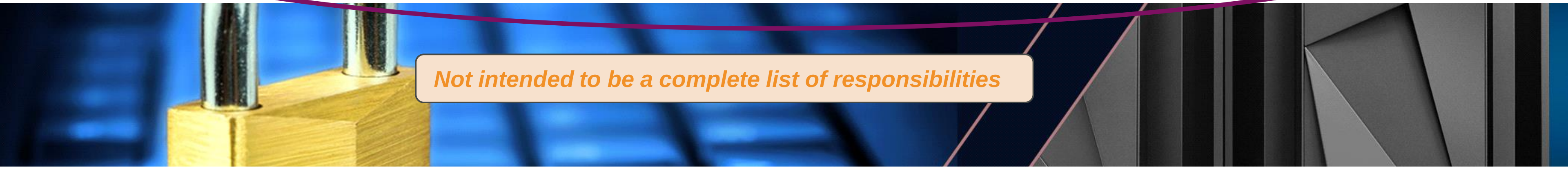
- System services that transmit data will typically retrieve the data using the access methods, thus the data in encrypted data sets is decrypted within these services prior to transmit.
- When transmitting sensitive data, as today, use the secure versions of these services.
  - Connect: Direct
  - FTPS
  - XMIT

Users/System admins performing these functions will require access to the key label.

Implementing data set encryption

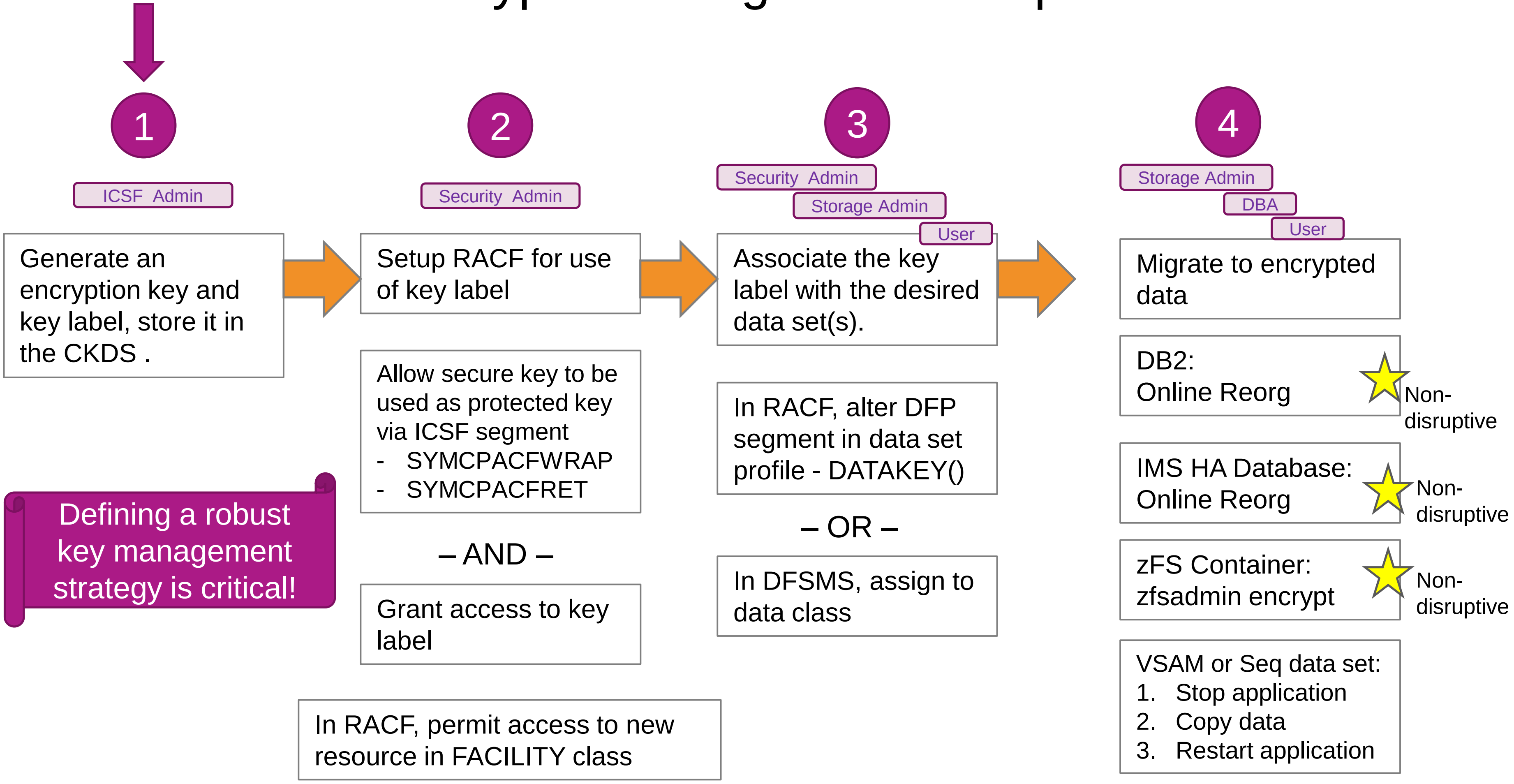
Implementing encryption at the data set level

Role	ICSF Admin	Security Admin	Security Auditor	Systems Prog	Storage Admin (Data Mgr)	User (Data Owner)
Objective	Responsible for key mgmt. (defining keys, key labels), working with key mgmt. system; Manages ICSF and key changes	Identify data sets that need to be encrypted; Tie encryption to user access; Responsible for creating RACF profiles, assigning access to key labels	Update audit reports; Ensure audit and reporting compliance	Ensures system (hw/sw) supports encryption; work with Security Admin to determine if migration action needed to allow encryption	Assigns encryption to specific data classes; manage backup, migration and replication	Automatically create encrypted data sets; Runs applications, submits jobs
How	Defines key labels in CKDS associated with secure AES256 keys	Update key label in RACF data set profile; Modify user profiles with key labels and access permissions to files	List the catalog, etc to display encryption status	Ensure all systems that may need to access the data have the CKDS	Set key labels for data class using storage mgmt. panels (ISMF); Updates ACS rtns	Add key label to JCL or IDCAMS DEFINE CLUSTER;
Benefit	Manages key repository	Encrypt sensitive data; Prevent unauthorized access to data based on profiles	Determine encryption status to meet compliance	Manages HW/SW level on systems to support encryption	Manages SMS constructs that enable encryption	Automate creation of encrypted files without code changes



Not intended to be a complete list of responsibilities

z/OS data set encryption – High Level Steps



1

ICSF Admin

Generate an encryption key and key label, store it in the CKDS .

Defining a robust key management strategy is critical!

2

Security Admin

Setup RACF for use of key label

Allow secure key to be used as protected key via ICSF segment
- SYMCPACFWRAP
- SYMCPACFRET

– AND –

Grant access to key label

In RACF, permit access to new resource in FACILITY class

3

Security Admin

Storage Admin

User

Associate the key label with the desired data set(s).

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In DFSMS, assign to data class

4

Storage Admin

DBA

User

Migrate to encrypted data

DB2:
Online Reorg

Non-disruptive

IMS HA Database:
Online Reorg

Non-disruptive

zFS Container:
zfsadmin encrypt

Non-disruptive

VSAM or Seq data set:
1. Stop application
2. Copy data
3. Restart application

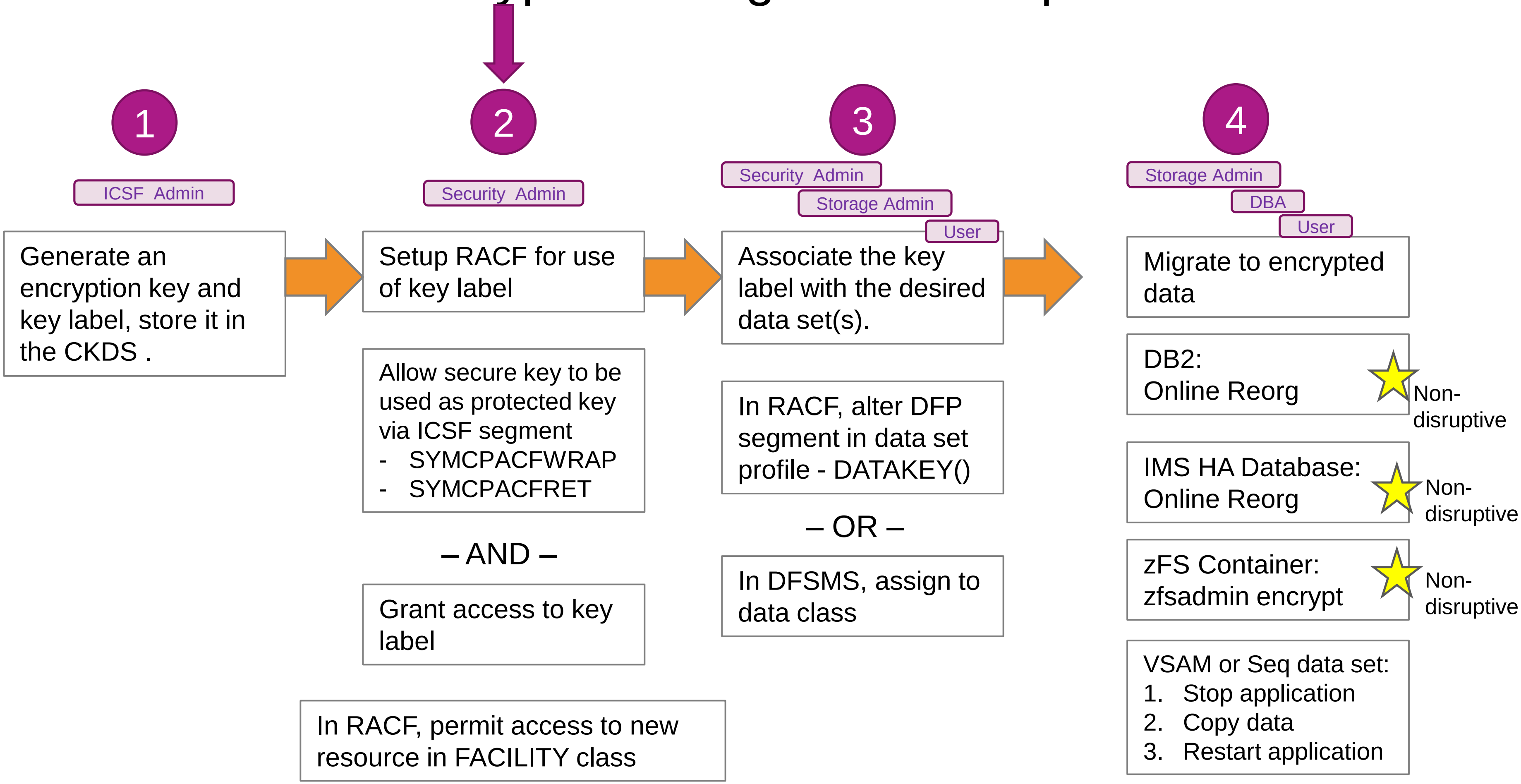
1 Prepare ICSF CKDS for use

Setup key repository

- **ICSF Admin** must ensure keys exist
  - Key labels defined in CKDS associated with secure AES256 keys
    - CKDS (key material) must be accessible across systems in the sysplex and replicated to sites that will access the encrypted data sets
  - Various methods available to create key label and data keys, ***for example***
    - CKDS Browser
    - ICSF services
      - CSNBKGN: Generate an AES 256-bit data key (token)
      - CSNBKRC2: Creates a key label in the CKDS with associated data key (token)
    - KGUP

Rexx example to create keys
https://www.ibm.com/developerworks/community/blogs/79c1eec4-00c4-48ef-ae2b-01bd8448dd6c/entry/Rexx_Sample_Secure_Key_Generate_256_bit_AES_DATA_key?lang=en

z/OS data set encryption – High Level Steps



1

ICSF Admin

Generate an encryption key and key label, store it in the CKDS .

2

Security Admin

Setup RACF for use of key label

Allow secure key to be used as protected key via ICSF segment

- SYMCPACFWRAP
- SYMCPACFRET

– AND –

Grant access to key label

In RACF, permit access to new resource in FACILITY class

3

Security Admin

Storage Admin

User

Associate the key label with the desired data set(s).

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In DFSMS, assign to data class

4

Storage Admin

DBA

User

Migrate to encrypted data

DB2:
Online Reorg ☆ Non-disruptive

IMS HA Database:
Online Reorg ☆ Non-disruptive

zFS Container:
zfsadmin encrypt ☆ Non-disruptive

VSAM or Seq data set:

1. Stop application
2. Copy data
3. Restart application

2a

Prepare for access method access to ICSF CKDS Key provisioning service*

Setup SAF resources for ICSF service

- **Security Admin** sets up access to the **ICSF CKDS Key Record Read2** (CSNBKRR2) service
  - Define the RACF profile such that no one has access to the ICSF services. ***For example,***

    RDEFINE CSFSERV * UACC(NONE)

  - Allow everyone to have access to the callable service CSNBKRR2. ***For example,***

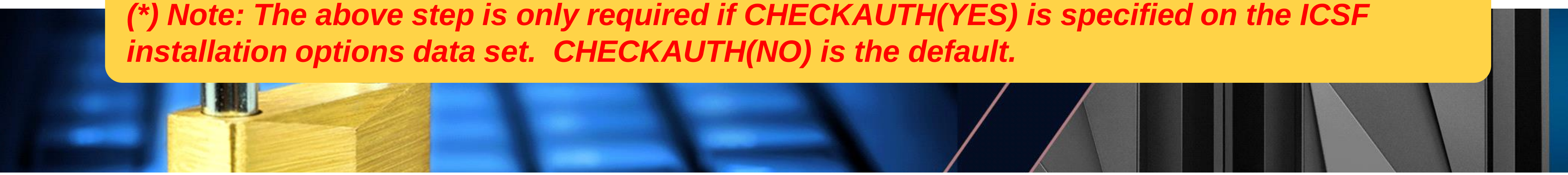
    RDEFINE CSFSERV CSFKRR2 UACC(READ)

    or

    PERMIT CSFKRR2 CLASS(CSFSERV) ID(*) ACCESS(READ)

The above are examples intended to show how an installation might set up CSFSERV profiles.

(*) Note: The above step is only required if CHECKAUTH(YES) is specified on the ICSF installation options data set. CHECKAUTH(NO) is the default.



2b

Prepare system to allow data set encryption

Set up SAF resource to enable data set encryption based on key label specification

- **Security Admin** must consider whether migration action should prevent creation of encrypted data sets via new resource in FACILITY class: **STGADMIN.SMS.ALLOW.DATASET.ENCRYPT**
  - ***Ensure all systems that may need to access the data have the CKDS with key material required to decrypt the data sets AND are at the correct HW/SW levels.***
    - All systems in the sysplex, remote sites, fall-back systems, …

```
RDEFINE FACILITY STGADMIN.SMS.ALLOW.DATASET.ENCRYPT UACC(NONE)
```

- To allow the system to create encrypted data sets when the key label is specified via a method *outside of the DFP segment in the RACF data set profile*, the user must have at least **READ authority** to the following new **resource in the FACILITY class.**

```
RALTER FACILITY STGADMIN.SMS.ALLOW.DATASET.ENCRYPT UACC(READ)
```

Allows security admin to control who can create encrypted data sets.

2c

Setup access to key labels

Setup SAF resources for key-label

- **Security Admin** sets up profiles in the CSFKEYS general resource class based on installation requirements. ***Any user that must access data in the clear must have access to the key label***
- **Security Admin** must also update the ICSF segment of the covering profile to allow ICSF to return a protected key: SYMCPACFWRAP(YES) SYMCPACFRET (YES)
- The following are ***examples:***
  - Define the RACF profile such that no one has access to key-label

    RDEFINE CSFKEYS key-label UACC(NONE)
  - Add the ICSF segment keywords to use the key label for a protected key

    RALTER CSFKEYS key-label ICSF(SYMCPACFWRAP(YES) SYMCPACFRET (YES))
  - To allow key label to be used by JOHN when accessed by any application

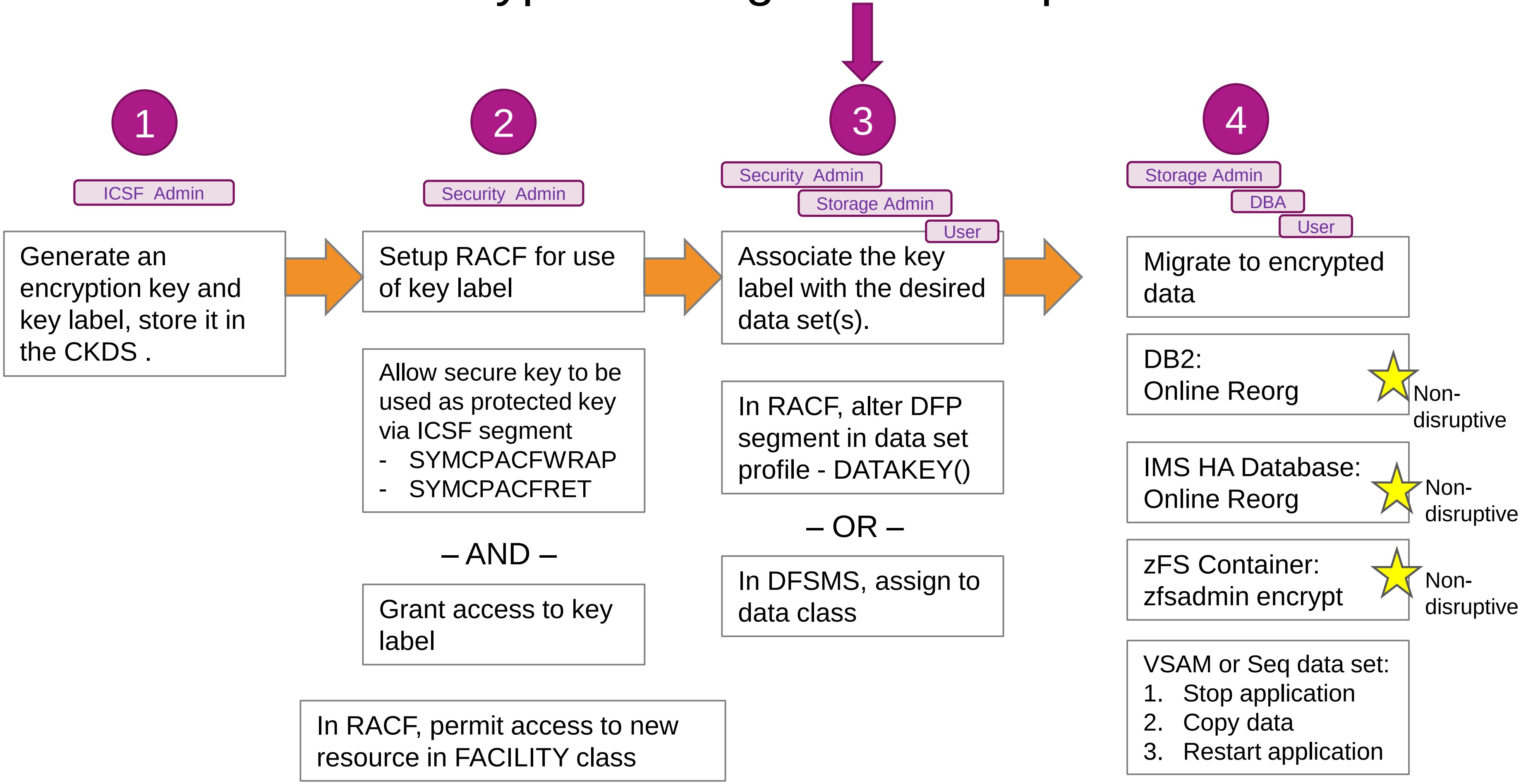
    PERMIT key-label CLASS(CSFKEYS) ID(JOHN) ACCESS(READ)
  - To allow key label to be used by MIKE only when accessed by DFSMS

    PERMIT key-label CLASS(CSFKEYS) ID(MIKE) ACCESS(READ) WHEN(CRITERIA(SMS(DSENCRYPTION)))
  - To allow key label to be used by any user only when accessed by DFSMS

    PERMIT key-label CLASS(CSFKEYS) ID(*) ACCESS(READ) WHEN(CRITERIA(SMS(DSENCRYPTION)))

The above are examples intended to show how an installation might set up CSFKEYS profiles based on access requirements. Designed to support separation of access: data owner vs data manager.

z/OS data set encryption – High Level Steps



1

ICSF Admin

Generate an encryption key and key label, store it in the CKDS .

2

Security Admin

Setup RACF for use of key label

Allow secure key to be used as protected key via ICSF segment

- SYMCPACFWRAP
- SYMCPACFRET

– AND –

Grant access to key label

In RACF, permit access to new resource in FACILITY class

3

Security Admin

Storage Admin

User

Associate the key label with the desired data set(s).

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In DFSMS, assign to data class

4

Storage Admin

DBA

User

Migrate to encrypted data

DB2:
Online Reorg

Non-disruptive

IMS HA Database:
Online Reorg

Non-disruptive

zFS Container:
zfsadmin encrypt

Non-disruptive

VSAM or Seq data set:

1. Stop application
2. Copy data
3. Restart application

3 Creating encrypted data sets – supplying key labels

A data set is defined as ‘encrypted’ when a **key label** is supplied on allocation of a ***new*** sequential or VSAM ***extended format*** data set

A **key label** supplied via new keywords in any of the following sources (using ***order of precedence*** as follows):

- **RACF Data set profile DFP segment**
- **JCL, Dynamic Allocation, TSO Allocate, IDCAMS DEFINE**
- **SMS Construct: Data Class**

3 Prepare for encryption on new data set allocation – OPTIONS for assigning key label

Setup RACF policy to supply key label in DFP segment

- **Security Admin** can update RACF DS profile to request encryption by adding key label

Setup SMS policy to supply key label on data class

- **Storage Admin** can update specific data class(es) via ISMF to request encryption by adding key label
- **Storage Admin** can update ACS routines via ISMF to select data classes enabled for encryption

Setup job(s) to supply key label on JCL

- **User** can modify JCL to allocate specific data sets as encrypted by adding key label

Modify application to supply key label on DEFINE

- **User** can modify an application to allocate specific data sets as encrypted by adding key label to dynamic allocation request or IDCAMS DEFINE CLUSTER.

3 Prepare for encryption on new data set allocation – OPTIONS for assigning key label

Setup RACF policy to supply key label in DFP segment

- **Security Admin** can update RACF DS profile to request encryption by adding key label

Setup SMS policy to supply key label on data class

- ~~**Storage Admin** can update specific data class(es) via ISMF to request encryption by adding key label~~
- ~~**Storage Admin** can update ACS routines via ISMF to select data classes enabled for encryption~~

Setup job(s) to supply key label on JCL

- ~~**User** can modify JCL to allocate specific data sets as encrypted by adding key label~~

Modify application to supply key label on DEFINE

- ~~**User** can modify an application to allocate specific data sets as encrypted by adding key label to dynamic allocation request or IDCAMS DEFINE CLUSTER.~~

Note: To only allow new encrypted data sets through RACF policy (and thus controlled by security admin), do not provide users read access to resource STGADMIN.SMS.ALLOW.DATASET.ENCRYPT

```
RDEFINE FACILITY STGADMIN.SMS.ALLOW.DATASET.ENCRYPT UACC(NONE)
```

OPTION: DFP segment in RACF data set profile

- Label of an existing key in the ICSF CKDS used by access methods for encrypting/decrypting sequential and VSAM data
- Provides granularity for different key labels to be used based on RACF profiles

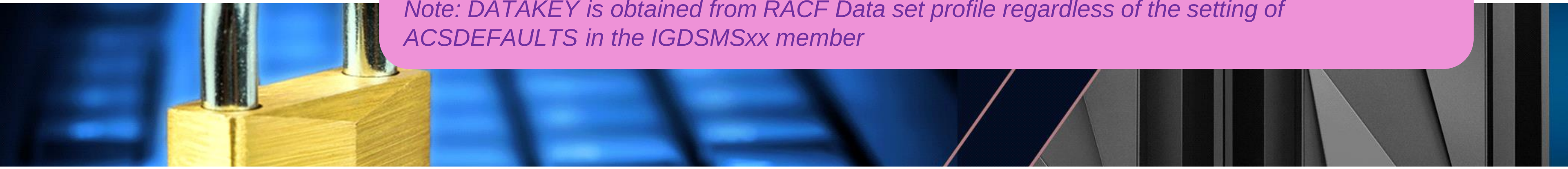
```
ALTDSD 'PROJECTA.DATA.*' UACC(NONE) DFP(RESOWNER(iduser1)DATAKEY(Key-Label))
```

Command Keyword	Meaning
DATAKEY(Key-Label)	Identifies the KEY LABEL in ICSF CKDS used to encrypt/decrypt the data
NODATAKEY	Removes a key label if defined to the RACF DPF segment

Key label only used for new data set create

Any subsequent change to RACF Data set profile will not affect existing data sets

Note: DATAKEY is obtained from RACF Data set profile regardless of the setting of ACSDEFAULTS in the IGDSMSxx member



OPTION: JCL, Dynamic Allocation and TSO Allocate

- New keyword to be used for DASD data sets
  - **DSKEYLBL=key-label**
    - Key label of an existing key in ICSF CKDS used by access methods for encrypting/decrypting sequential and VSAM data

```
//DD1    DD    DSN=DSN1,DISP=(NEW,CATLG),DATACLAS=DSN1DATA,MGMTCLAS=DSN1MGMT,
//                 STORCLAS=DSN1STOR,DSKEYLBL='LABEL.FOR.DSN1'
```

- For dynamic allocation text unit: DALDKYL
- For TSO allocate: DSKEYLBL(label-name)

DSKEYLBL is effective only if the new data set is on DASD. It is ignored for device types other than DASD, including DUMMY.

Key label only used for new data set create

OPTION: Creating a new VSAM data set via IDCAMS

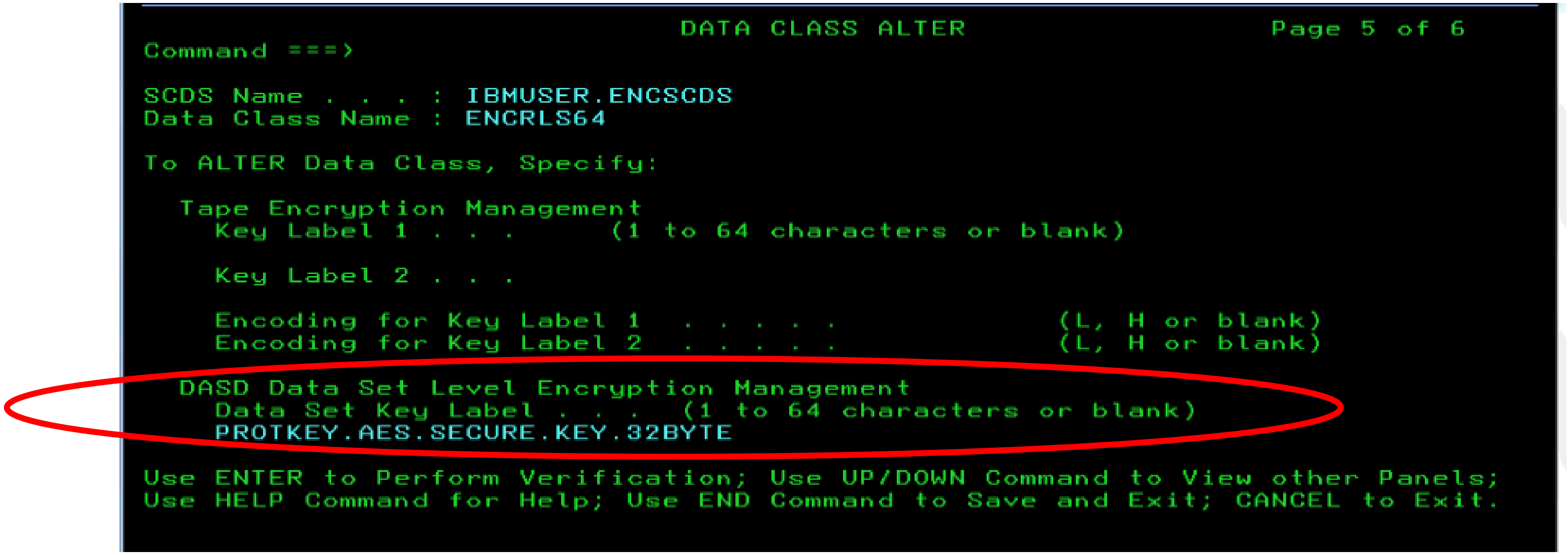
- New parameter on DEFINE for CLUSTER
  - **KEYLABEL=key-label**
    - Key label of an existing key in ICSF CKDS used by access methods for encrypting/decrypting sequential and VSAM data
    - Used for both cluster and any alternate index

```
DEFINE CLUSTER -
(NAME(DSN1.EXAMPLE.ESDS1) -
RECORDS(100 500) -
RECORDSIZE(250 250) -
KEYLABEL (LABEL.FOR.DSN1) -
NONINDEXED )
```

OPTION: SMS Construct: Data Class

Data Class identifies key label to be used when creating a new data set.

- Key label of an existing key in ICSF CKDS used by access methods for encrypting/decrypting sequential and VSAM data



```
                              DATA CLASS ALTER                    Page 5 of 6
Command ===>

SCDS Name . . . : IBMUSER.ENCSCDS
Data Class Name : ENCRLS64

To ALTER Data Class, Specify:

  Tape Encryption Management
    Key Label 1 . . .      (1 to 64 characters or blank)

    Key Label 2 . . .

    Encoding for Key Label 1 . . . . .              (L, H or blank)
    Encoding for Key Label 2 . . . . .              (L, H or blank)
  DASD Data Set Level Encryption Management
    Data Set Key Label . . . . (1 to 64 characters or blank)
    PROTKEY.AES.SECURE.KEY.32BYTE

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Key label only used for new data set create

Prepare for extended format on new data set allocation - OPTIONS for DSNTYPE

Setup SMS policy to request extended format via data class

- **Storage admin** can update specific data class(es) via ISMF to request extended format via DSNTYPE option
  - *SMS Data class **DSNTYPE=EXTR or EXTP***
- **Storage admin** can update ACS routines via ISMF to select data classes enabled for extended format

Setup job(s) to request extended format on JCL

- **User** can modify JCL to allocate specific data sets as encrypted by adding key label
  - *JCL **DSNTYPE=EXTREQ or EXTPREF***

Restriction note: Sequential extended format data sets cannot be opened for EXCP.

Data set encryption requires extended format

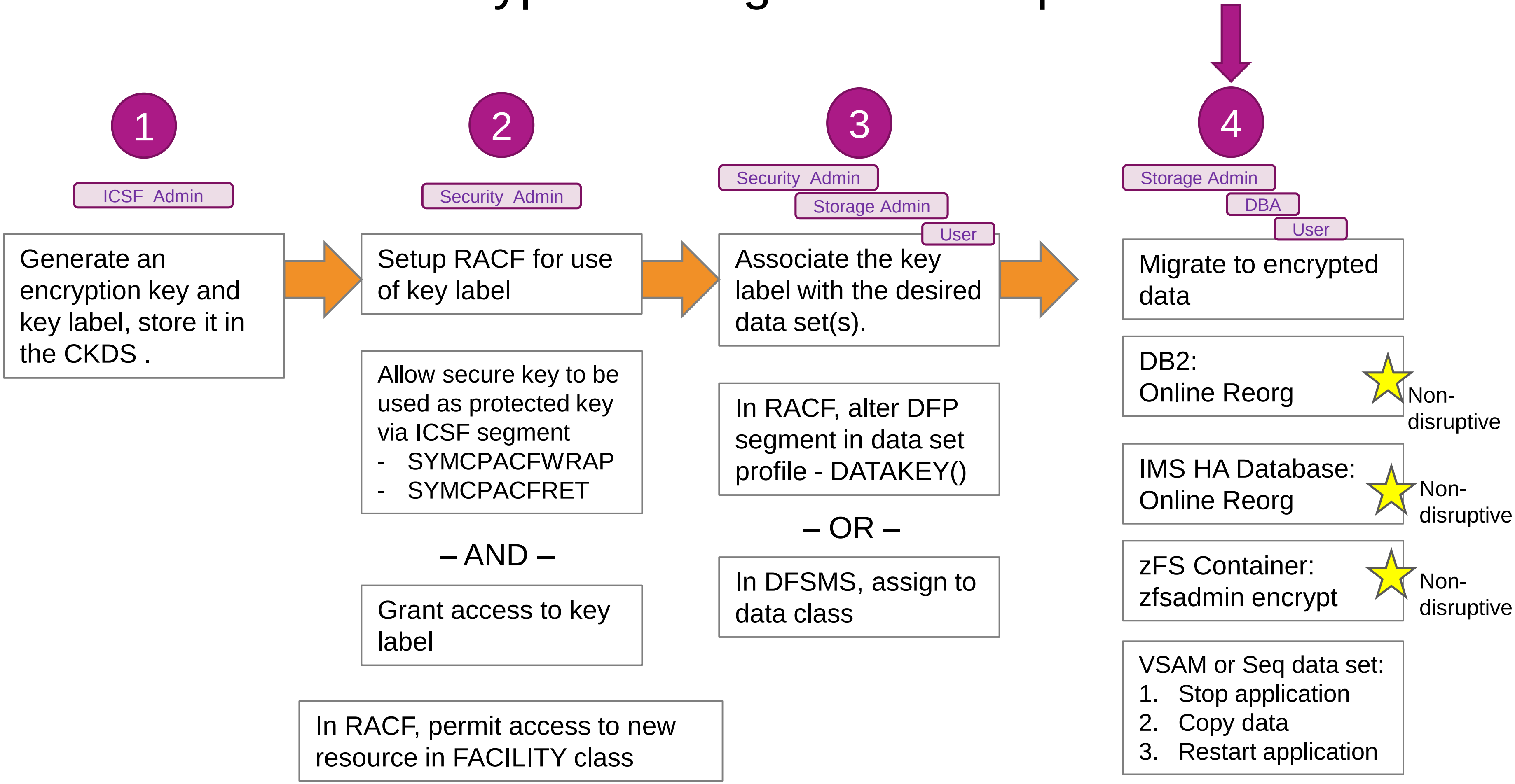
Optionally, prepare for **data set compression** on new data set allocation

Setup SMS policy to request compression

- **Storage admin** can update specific data class(es) via ISMF to request compression via COMPACTION option
  - Sequential extended format data sets support generic, tailored, or zEDC compression
  - VSAM extended format KSDS supports generic compression (Only KSDS can be compressed format)
- **Storage admin** can update ACS routines via ISMF to select data classes enabled for compression

Restriction note: Sequential compressed format data sets cannot be opened for UPDATE.

z/OS data set encryption – High Level Steps



1

ICSF Admin

Generate an encryption key and key label, store it in the CKDS .

2

Security Admin

Setup RACF for use of key label

Allow secure key to be used as protected key via ICSF segment
- SYMCPACFWRAP
- SYMCPACFRET

– AND –

Grant access to key label

In RACF, permit access to new resource in FACILITY class

3

Security Admin

Storage Admin

User

Associate the key label with the desired data set(s).

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In DFSMS, assign to data class

4

Storage Admin

DBA

User

Migrate to encrypted data

DB2:
Online Reorg

Non-disruptive

IMS HA Database:
Online Reorg

Non-disruptive

zFS Container:
zfsadmin encrypt

Non-disruptive

VSAM or Seq data set:
1. Stop application
2. Copy data
3. Restart application

4a Converting existing data sets to encryption

*super-admin

Storage admin(*) or **user** can copy an existing data set to a new target data set allocated as encrypted.

- No utility available to perform a conversion without decrypting data from source and re-encrypting data onto target
- Standard utilities can be used to perform the copy, for example
  - ISPF 3.3 Copy data set
  - IDCAMS REPRO
  - IEBGENER

DB admin: For high availability, DB2 and IMS provide non-disruptive migration to encryption with DB online reorg function

The above could also be used to re-key an existing encrypted data set or DB to a new key.

4b

Accessing data in encrypted data sets

- **User** can access data in encrypted data sets
  - **When accessed via BSAM, QSAM, VSAM or VSAM/RLS**
    - ***Transparent access*****…no application changes:**

      Data encrypted on writes and decrypted on reads

Transparent to any applications or middleware making use of VSAM, QSAM, BSAM access methods. Refer to individual product documentation to confirm support of z/OS data set encryption.

For those applications that use the licensed Media Manager services, changes to Media Manager interfaces required to access encrypted data sets.

4b

How can I be sure the data is encrypted?

- Encryption attributes displayed in various system interfaces
  - SMF records
  - DCOLLECT records
  - LISTCAT
  - IEHLIST LISTVTOC
  - Catalog Search Interface (CSI)
  - ISITMGD
- **To view encrypted data, can use DFSMSdss PRINT Tracks**

Fallback to no encryption

- Prevent creation of new encrypted data sets
  - Prevent new data sets to be allocated as encrypted when key label is specified via a source other than the RACF DFP segment

```
    RDEFINE FACILITY STGADMIN.SMS.ALLOW.DATASET.ENCRYPT UACC(NONE)
                    OR
    RALTER FACILITY STGADMIN.SMS.ALLOW.DATASET.ENCRYPT UACC(NONE)
```

  - Prevent new data sets to be allocated as encrypted when key label is specified via the RACF DFP segment
    - Remove key label from all RACF DFP segments, for example

```
ALTDSD 'PROJECTA.DATA.*' UACC(NONE) DFP(RESOWNER(iduser1) NODATAKEY)
```

- Copy any encrypted data sets to unencrypted data sets
  - Use a copy utility such as ISPF 3.3 Copy, IEBGENER, IDCAMS REPRO
    - Note: DFSMSdss COPY does not support converting attributes between source and target

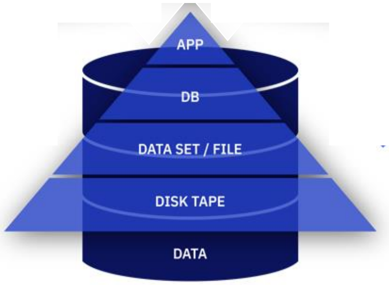
Pervasing Encryption - Getting Started

z/OS Data Set Encryption - Getting Started

- Choose an application
- Prepare test environment
- Enable encryption (4 steps)
- Test & verify
- Plan for production rollout

Pervasive encryption client advocacy program

z/OS Data Set Encryption – Choose an application



Questions:

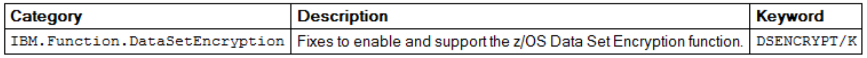
- Is your enterprise driving a top down encryption initiative?
  - e.g. GDPR, PCI DSS, etc..
- What do you expect to be the first use case for data set encryption?

- CICS/VSAM application
- DB2 database
- IMS database
- Batch workload
- Log data sets (system logger)

Note: Data set encryption supports extended format sequential and VSAM

z/OS Data Set Encryption – Prepare test environment

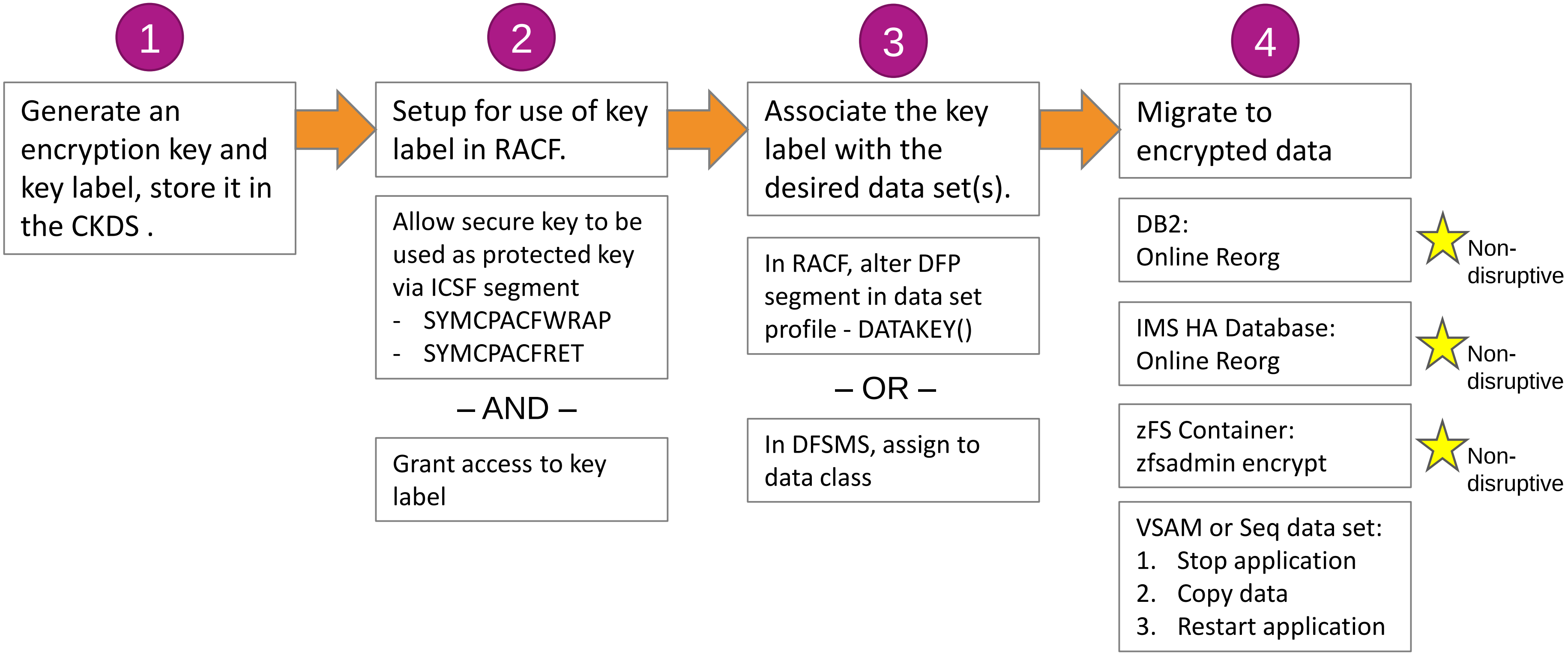
- Hardware
  - CPACF protected key (z196 or later for AES-XTS mode)
  - Crypto Express3 or later required for secure key
  - Recommend use of Crypto Express in test to validate crypto operational procedures (e.g. master key loading, master key change, etc…)
- Setup & Configure ICSF
  - Load AES master key
  - Recommend installing latest ICSF web deliverable (HCR77C1) *(Can generate AES DATA keys using CKDS Browser)*
- Install/Update Base Software
  - DFSMS - z/OS 2.2 + service or z/OS 2.3



Category	Description	Keyword
IBM.Function.DataSetEncryption	Fixes to enable and support the z/OS Data Set Encryption function.	DSENCRYPT/K

  - RACF – z/OS 2.2 + service or z/OS 2.3
  - ICSF – HCR77A0-B1 + service or HCR77C0-C1
- Install/Update Exploitation Software
  - DB2, IMS, logger… vendor products?

z/OS Data Set Encryption – Enable Encryption (4 steps)



1

Generate an encryption key and key label, store it in the CKDS .

2

Setup for use of key label in RACF.

Allow secure key to be used as protected key via ICSF segment
- SYMCPACFWRAP
- SYMCPACFRET

– AND –

Grant access to key label

3

Associate the key label with the desired data set(s).

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In DFSMS, assign to data class

4

Migrate to encrypted data

DB2:
Online Reorg

★ Non-disruptive

IMS HA Database:
Online Reorg

★ Non-disruptive

zFS Container:
zfsadmin encrypt

★ Non-disruptive

VSAM or Seq data set:
1. Stop application
2. Copy data
3. Restart application

z/OS Data Set Encryption – Plan for production rollout

Questions:

- Is ICSF environment configured for Parallel Sysplex?
- Is ICSF environment configured for DR?
- Is an Enterprise Key Management system deployed?

- Configure ICSF & key store for high availability
- Configure ICSF & key store for DR
- Configure periodic logical back up of key store
- Deploy Enterprise Key Management system for backup & recovery
- Consider use of host based compression
- Plan key label naming convention and access control
- Evaluate encryption overhead

Enterprise Key Management – Operational Keys

Encryption of data at enterprise scale requires robust key management

- The current key management landscape can be characterized by clients who have …
  - … already deployed an enterprise key management solution
  - … developed a self-built key management solution
  - … not deployed an enterprise key management solution

Key management for pervasive encryption must provide …

- Policy based key generation
- Policy based key rotation
- Key usage tracking
- Key backup & recovery

EKMF

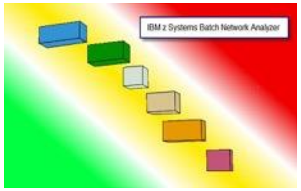
The IBM Enterprise Key Management Foundation (EKMF) provides real-time, centralized secure management of keys and certificates in an enterprise with a variety of cryptographic devices and key stores.

z/OS Data Set Encryption – Evaluate impact

Estimating CPU Cost of Data Protection

z Batch Network Analyzer (zBNA)

zBNA 1.8.1

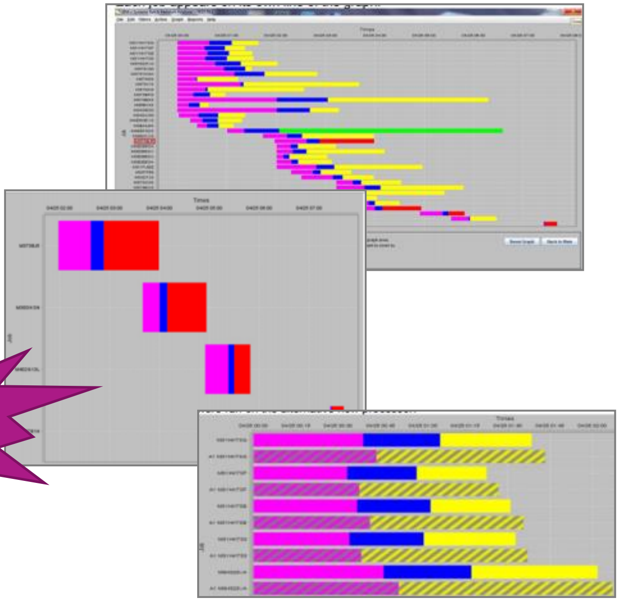


zBNA Background:

- A no charge, “as is” tool originally designed to analyze batch windows
- PC based, and provides graphical and text reports
- Available on techdocs for customers, business partners, and IBMers
  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS5132
- Previously enhanced for zEDC to identify & evaluate compression candidates

zBNA Encryption Enhancements:

- Enhanced to help clients estimate encryption CPU overhead based on actual client workload SMF data
- Ability to select z13 or z14 as target machine
- Support provided for
  - z/OS data set encryption
  - Coupling Facility encryption



Version 1.8.1 Available on 8/31/2017

Note: z/OS Capacity Planning tool zCP3000 also updated to provide encryption estimates
http://w3-03.ibm.com/support/americas/wsc/cpsproducts.html

Level of Protection (HW and SW Support)

What are key encrypting keys (KEKs)?

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

Master Keys

Master keys are used only to encipher and decipher keys.

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.

Symmetric KEKs

Encrypt symmetric keys with another symmetric key.

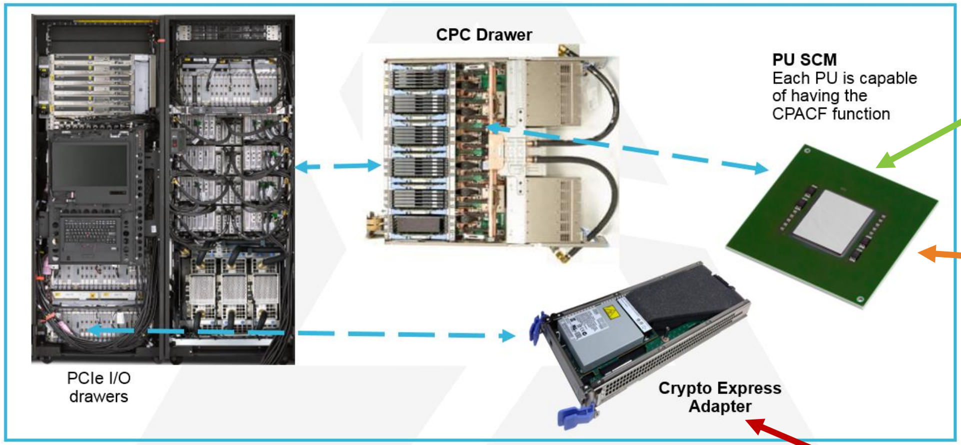
Asymmetric KEKs

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.

Understanding Clear, Secure and Protected Keys

Secure keys have key values that are encrypted by a Master Key on a tamper-responding CryptoExpress adapter.



High speed

Clear Key

Key values are not encrypted. Crypto operations may be performed in CPACF or on a Crypto Express adapter

Speed & Security Hybrid

Protected Key

Key values are encrypted under a CPACF wrapping key. Crypto operations are performed only using CPACF

High security

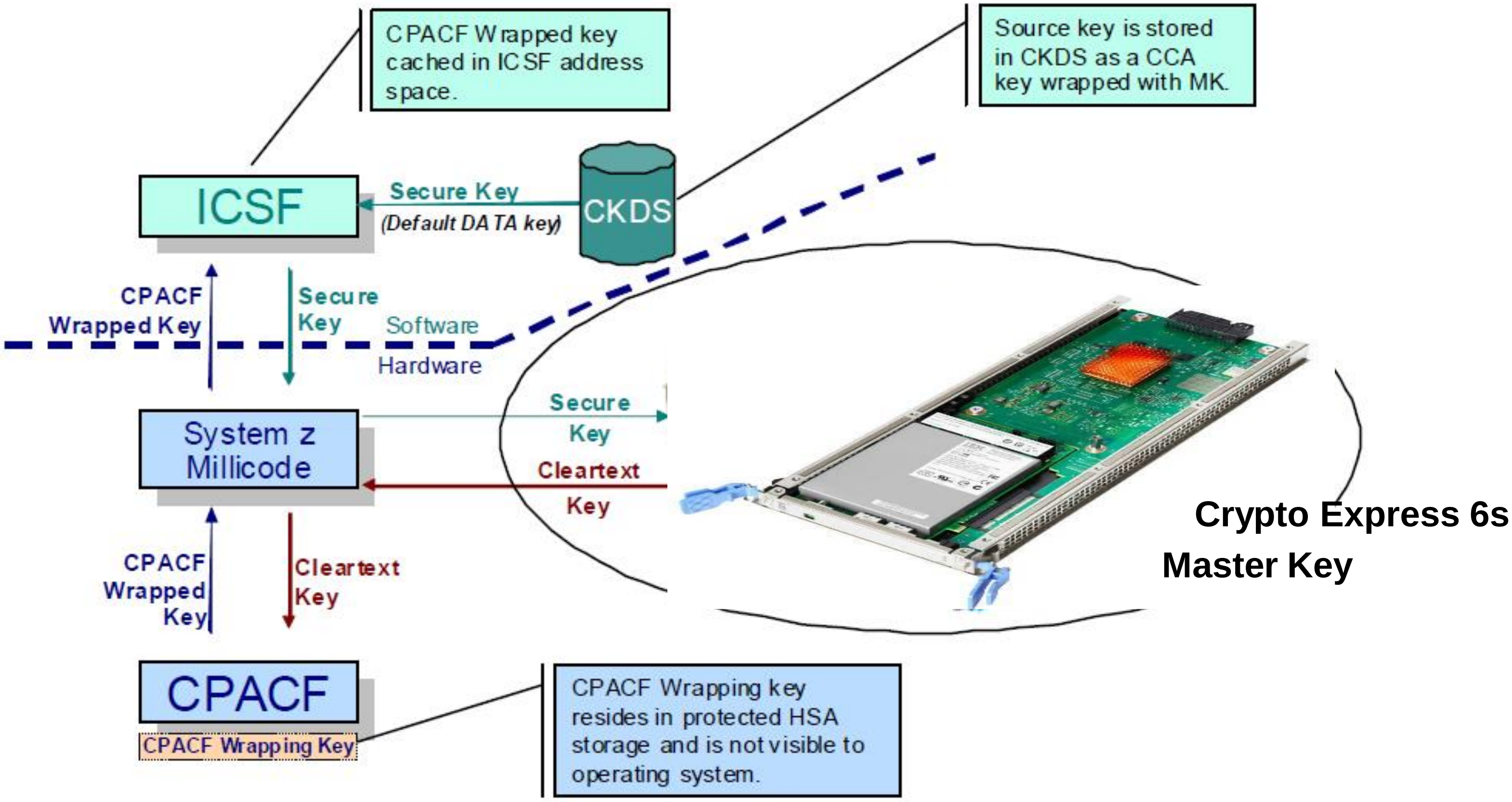
Secure Key

Key values are encrypted under a Master Key. Crypto operations are performed only on a Crypto Express adapter

Note: With z/OS data set encryption, protected keys are implicitly created from secure keys.

Only protected keys created from secure keys should be used for Pervasive Encryption.

Protected Key using Wrapping Key



Resources

IBM Systems Lab Services — IBM Z and LinuxONE

Pervasive Encryption Readiness Assessment

Overview

The vision of Pervasive Encryption is to provide a simple, transparent, and consumable approach to enable extensive encryption of data in-flight and at-rest to substantially reduce the costs associated with protecting data and achieving compliance mandates. This offering has been designed to assess where you may be on your journey to full enablement of pervasive encryption. IBM Systems Lab Services consultants will help assess your current state, and give you a roadmap to where you ultimately want to be.

Benefits

Based on a short interview about your current IBM Z or LinuxONE enabled features and pervasive encryption objectives, IBM Systems Lab Services consultants will work with you to determine what your steps should be for full pervasive encryption enablement.

In addition, IBM Lab Services consultants will provide you deeper insights into the components you will need to configure and what the best practices of key management on IBM Z and LinuxONE should be.

IBM Systems Lab Services can follow this initial assessment with a full portfolio offering of services to assist in deployment.

Key Features

- Review of the current state toward pervasive encryption.
- Identification of the steps to take to start pervasive encryption.
- Overview of the best practices in key management on IBM Z and LinuxONE

Duration

24-60 hours depending on the complexity of your environment.

How to contact us:

IBM Sellers can find a Lab Services Opportunity Manager in your area ->
http://ibm.biz/LabServicesOM

IBM Business Partners and Clients can contact us at
https://www-03.ibm.com/systems/services/labservices/contact.html

Or send an email to ibmsls@us.ibm.com

Resources: IBM Knowledge Center

***IBM Z Pervasive Encryption* – Link to technical resources about pervasive encryption, including data set encryption**

Publications

- ***z/OS DFSMS Using the New Functions* – Data Set encryption implementation information**
- ***z/OS DFSMS Using Data Sets* – Data Set encryption implementation information**
- *z/OS DFSMS Introduction*
- *z/OS DFSMSdfp Storage Administration*
- *z/OS DFSMS Managing Catalogs*
- *z/OS DFSMS Access Method Services Command Reference*
- *z/OS DFSMS Macro Instructions for Data Sets*
- *z/OS DFSMSdfp Advanced Services*
- *z/OS DFSMSdfp Diagnosis*
- *z/OS DFSMSdss Storage Administration Reference*
- *z/OS DFSMShsm Data Areas*
- *z/OS DFSMS Installation Exits*
- *z/OS MVS Initialization and Tuning Reference*
- *z/OS MVS System Commands*
- *z/OS MVS JCL Reference*
- *z/OS MVS System Management Facility (SMF)*
- *z/OS MVS System Messages Volume 1, 2, 6, 7 and 8*
- *z/OS MVS Programming: Authorized Assembler Services Guide*
- *z/OS Summary of Message and Interface Changes*
- *z/OS Migration*

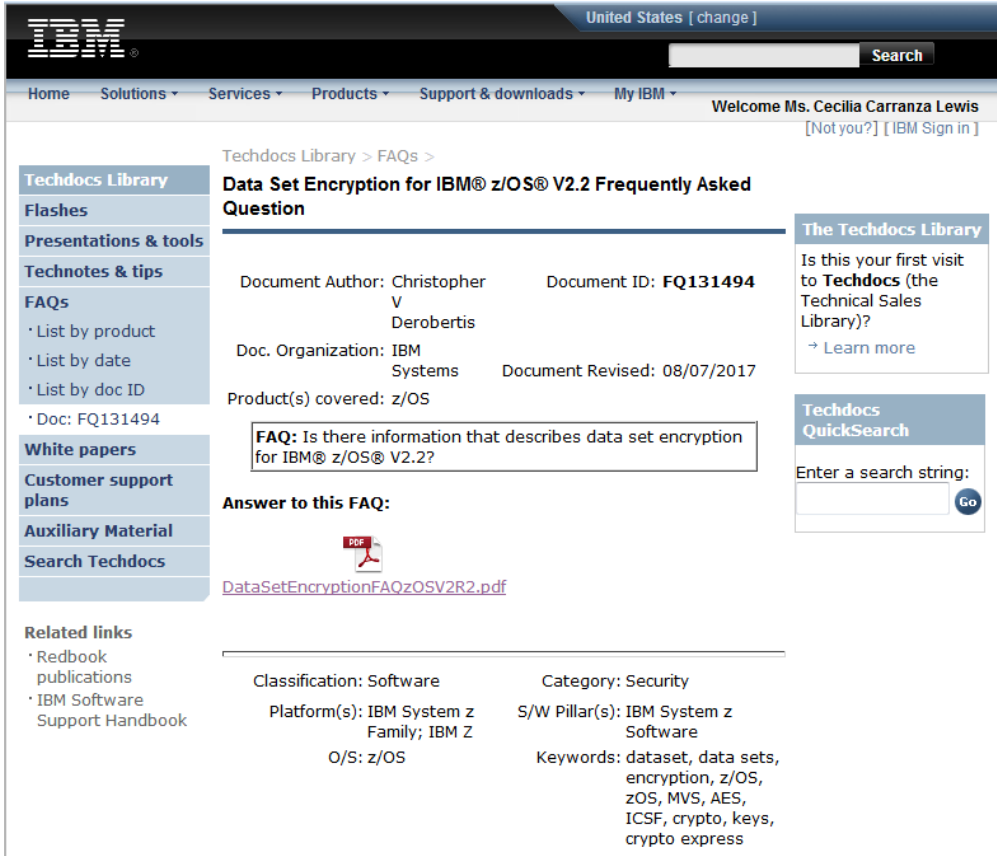
z/OS DFSMS V2.2 pub pdf package for data set encryption.
http://publibz.boulder.ibm.com/zoslib/pdf/OA50569.pdf

Draft IBM Redbooks publication available!!
Getting Started with z/OS Data Set Encryption

Resources:
Technote for z/OS V2.2

Techdoc contains

- Support provided in V2.2
- Complete list of maintenance
- HW/SW requirements
- Restrictions
- Exploiter support
  - DB2, IMS, CICS, MQ, zFS, zSecure



Welcome Ms. Cecilia Carranza Lewis
[Not you?] [IBM Sign in]

Techdocs Library > FAQs >

Data Set Encryption for IBM® z/OS® V2.2 Frequently Asked Question

Document Author: Christopher V Derobertis

Document ID: **FQ131494**

Doc. Organization: IBM Systems

Document Revised: 08/07/2017

Product(s) covered: z/OS

FAQ: Is there information that describes data set encryption for IBM® z/OS® V2.2?

Answer to this FAQ:

DataSetEncryptionFAQzOSV2R2.pdf

Classification: Software

Category: Security

Platform(s): IBM System z Family; IBM Z

S/W Pillar(s): IBM System z Software

O/S: z/OS

Keywords: dataset, data sets, encryption, z/OS, zOS, MVS, AES, ICSF, crypto, keys, crypto express

Related links
- Redbook publications
- IBM Software Support Handbook

The Techdocs Library

Is this your first visit to **Techdocs** (the Technical Sales Library)?

→ Learn more

Techdocs QuickSearch

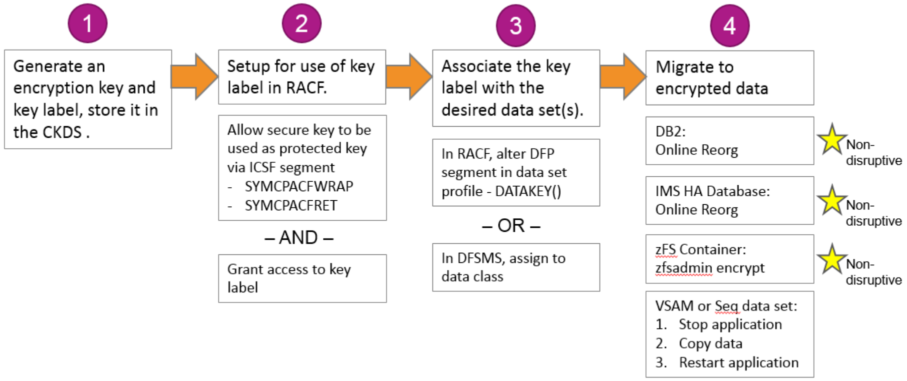
Enter a search string:

www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FQ131494

Resources: Proof of Technology

New!!
Learn how to implement these steps..via hands on training

z/OS data set encryption – Enable Encryption (4 steps)



1 Generate an encryption key and key label, store it in the CKDS .

2 Setup for use of key label in RACF.

Allow secure key to be used as protected key via ICSF segment
- SYMCPACFWRAP
- SYMCPACFRET

– AND –

Grant access to key label

3 Associate the key label with the desired data set(s).

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In DFSMS, assign to data class

4 Migrate to encrypted data

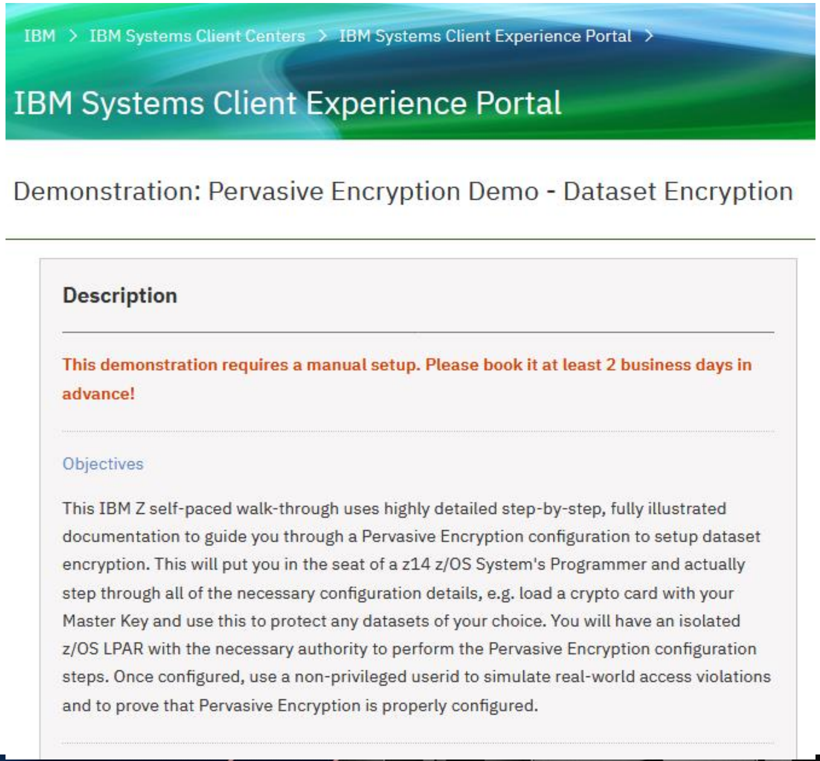
DB2: Online Reorg ☆ Non-disruptive

IMS HA Database: Online Reorg ☆ Non-disruptive

zFS Container: zfsadmin encrypt ☆ Non-disruptive

VSAM or Seq data set:
1. Stop application
2. Copy data
3. Restart application

Hands On PoT https://ibm.biz/client-experience-portal



IBM > IBM Systems Client Centers > IBM Systems Client Experience Portal >

IBM Systems Client Experience Portal

Demonstration: Pervasive Encryption Demo - Dataset Encryption

Description

This demonstration requires a manual setup. Please book it at least 2 business days in advance!

Objectives

This IBM Z self-paced walk-through uses highly detailed step-by-step, fully illustrated documentation to guide you through a Pervasive Encryption configuration to setup dataset encryption. This will put you in the seat of a z14 z/OS System's Programmer and actually step through all of the necessary configuration details, e.g. load a crypto card with your Master Key and use this to protect any datasets of your choice. You will have an isolated z/OS LPAR with the necessary authority to perform the Pervasive Encryption configuration steps. Once configured, use a non-privileged userid to simulate real-world access violations and to prove that Pervasive Encryption is properly configured.

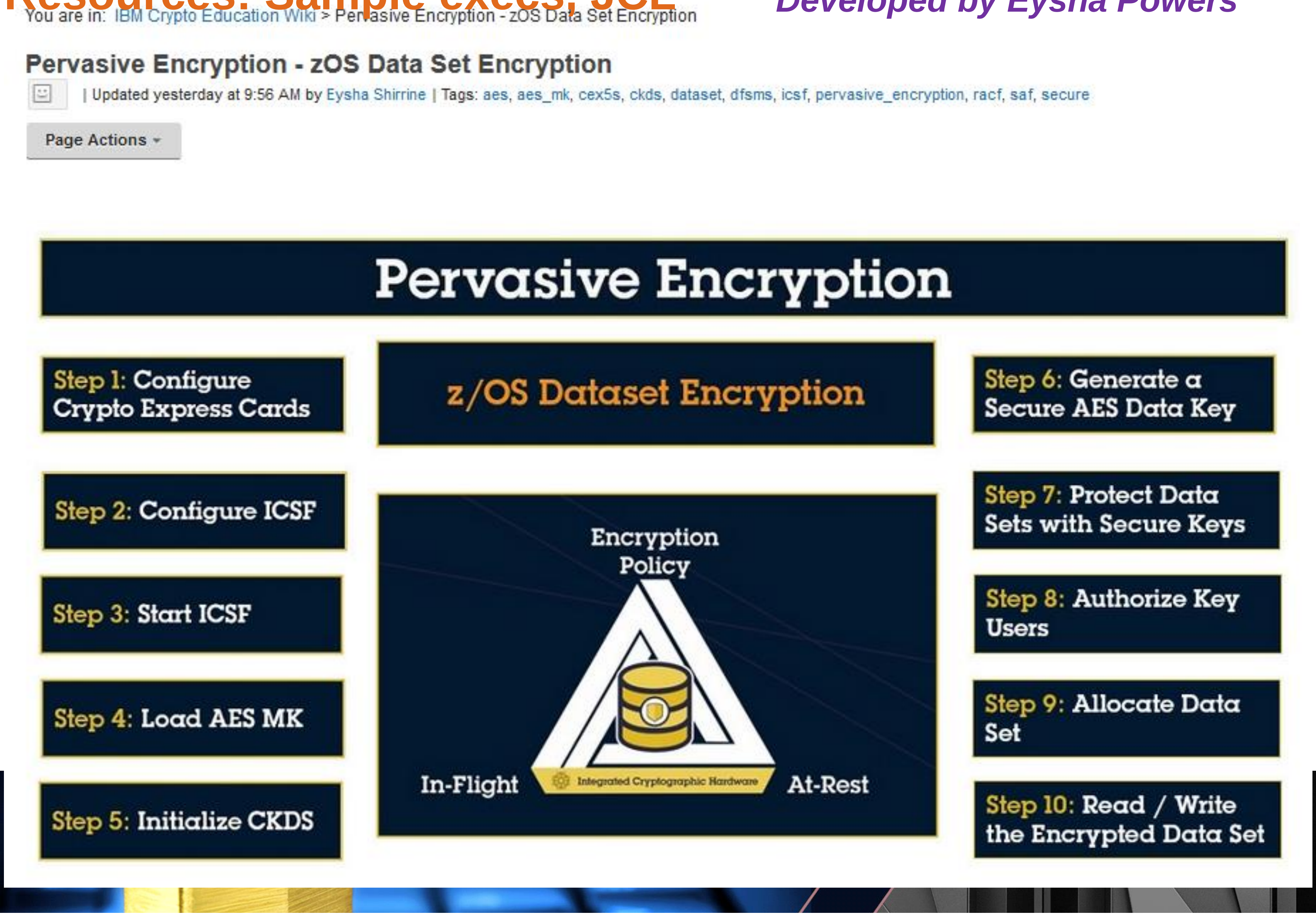
…or watch someone implement the steps

https://www.youtube.com/watch?v=zdSXRUSmkb4

How to Implement Pervasive Dataset Encryption on IBM z/OS

Resources: Sample execs, JCL

Developed by Eysha Powers



You are in: IBM Crypto Education Wiki > Pervasive Encryption - zOS Data Set Encryption

Pervasive Encryption - zOS Data Set Encryption

| Updated yesterday at 9:56 AM by Eysha Shirrine | Tags: aes, aes_mk, cex5s, ckds, dataset, dfsms, icsf, pervasive_encryption, racf, saf, secure

Page Actions

Pervasive Encryption

z/OS Dataset Encryption

Step 1: Configure Crypto Express Cards

Step 2: Configure ICSF

Step 3: Start ICSF

Step 4: Load AES MK

Step 5: Initialize CKDS

Step 6: Generate a Secure AES Data Key

Step 7: Protect Data Sets with Secure Keys

Step 8: Authorize Key Users

Step 9: Allocate Data Set

Step 10: Read / Write the Encrypted Data Set

Key Management

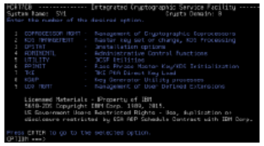
- Master Key
- Operational Key
  - Data Key
  - Application Key

What IBM tools are available to manage keys?

Integrated Cryptographic Services Facility (ICSF)

ICSF provides callable services and utilities that generate, store, and manage keys, and also perform cryptographic operations.

Supports Master Keys and Operational Keys



```
HCR77C0 -------------- Integrated Cryptographic Service Facility ------
System Name:  SY1                                Crypto Domain: 8
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, 2015.
     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.
OPTION ===>
```

Trusted Key Entry (TKE) Workstation

TKE securely manages multiple Cryptographic Coprocessors and keys on various generations of IBM Z from a single point of control.

Supports Master Keys and Operational Keys

Let's take a closer look

Enterprise Key Management Foundation (EKMF)

EKMF securely manages keys and certificates for cryptographic coprocessors, hardware security modules (HSM), cryptographic software, ATMs, and point of sale terminals.

Supports Operational Keys

Security Key Lifecycle Manager (SKLM)

SKLM v2.7 provides key storage, key serving and key lifecycle management for IBM and non-IBM storage solutions using the OASIS Key Management Interoperability Protocol (KMIP) and IBM Proprietary Protocol (IPP).

Supports Operational Keys <u>for Self Encrypting Devices (SEDs)</u>

35

ICSF

Integrated Cryptographic Service Facility (ICSF)

Base element of z/OS that provides cryptographic services

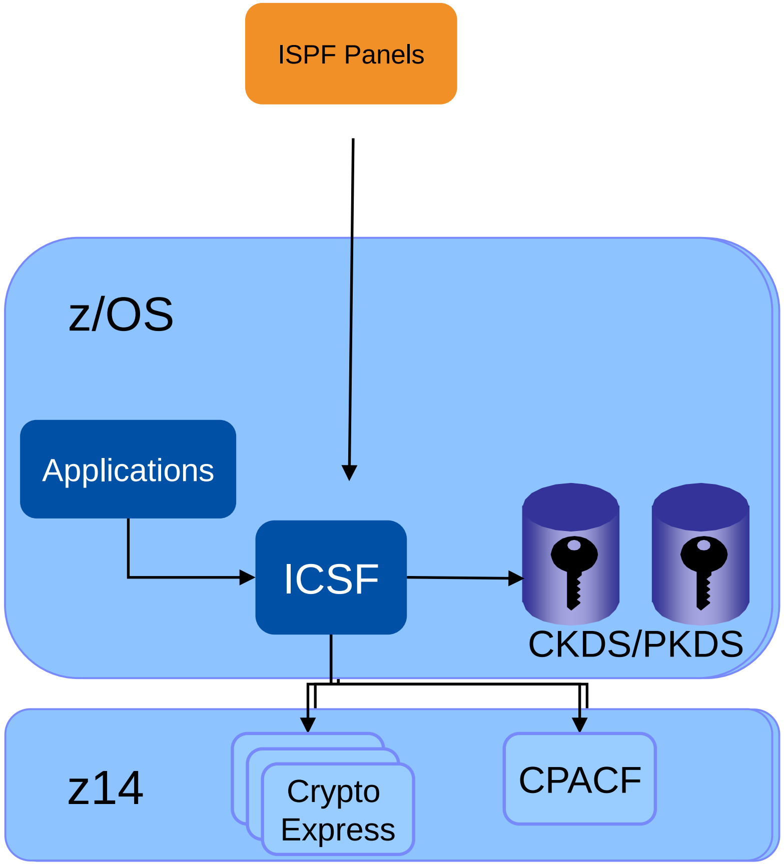
Provides an application programmers interface (API) for applications that need to perform crypto

Provides basic key management

Keystores (CKDS, PKDS, TKDS) for cryptographic key material

Provides access to:
Hardware Cryptographic Coprocessors, Cryptographic Accelerators
CP Assist for Cryptographic Function (CPACF)

ICSF

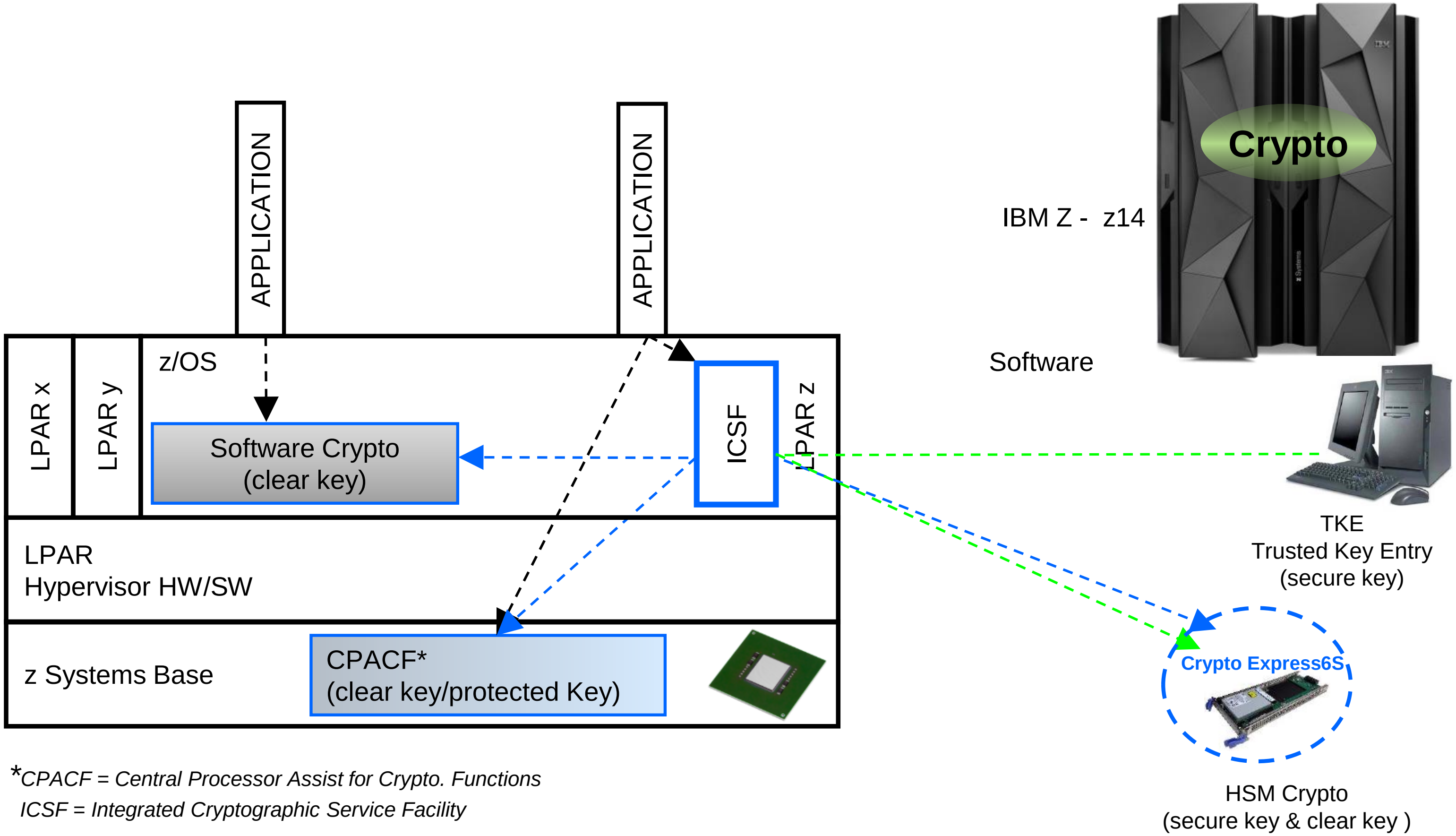

ISPF Panels
z/OS
Applications
ICSF
CKDS/PKDS
z14
Crypto Express
CPACF

ICSF Services

- **Standard Cryptographic Functions:**
  - Encryption and Decryption of Data
  - Hashing algorithms
  - Digital signatures
  - Message Authentication Codes (MACs)
  - Key generation and distribution

- **Protocols and Standards:**
  - Secure Sockets Layer
  - PKCS #11

Crypto support in IBM Z - z14 (z/OS)



*CPACF = Central Processor Assist for Crypto. Functions
ICSF = Integrated Cryptographic Service Facility

Introducing IBM Enterprise Key Management Foundation

Provide a centralized key management solution that leverages clients' investments in IBM Z hardware cryptography for the ultimate protection of sensitive keys and meeting compliance standards

Solution Summary

- Provides a simple centralized key management system which adheres to industry standards
- Provides a foundation that can be tailored to address the needs of multiple industry segments to assist key officers in enforcing requirements set forth by an enterprise key management policy
- Features crypto analytic capabilities that help identify compliance issues and to assist key officers in understanding who has access to key material

Solution Benefits

- Provide higher quality of service by efficient key management and automation
- Leverages clients investments in Z
- Simplifies business continuity considerations for mission critical key material

Requirements – Bill of Materials

- z196/z114/zEC12/zBC12/z13/z14 & CryptoExpress3/4S/5/6
- z/OS ICSF Version 1.13 and or DB2 V11 or higher
- IBM EKMF license
  - Installation and Configuration Services
  - Maintenance and Support

IBM EKMF Architecture & Components

Secure workstation

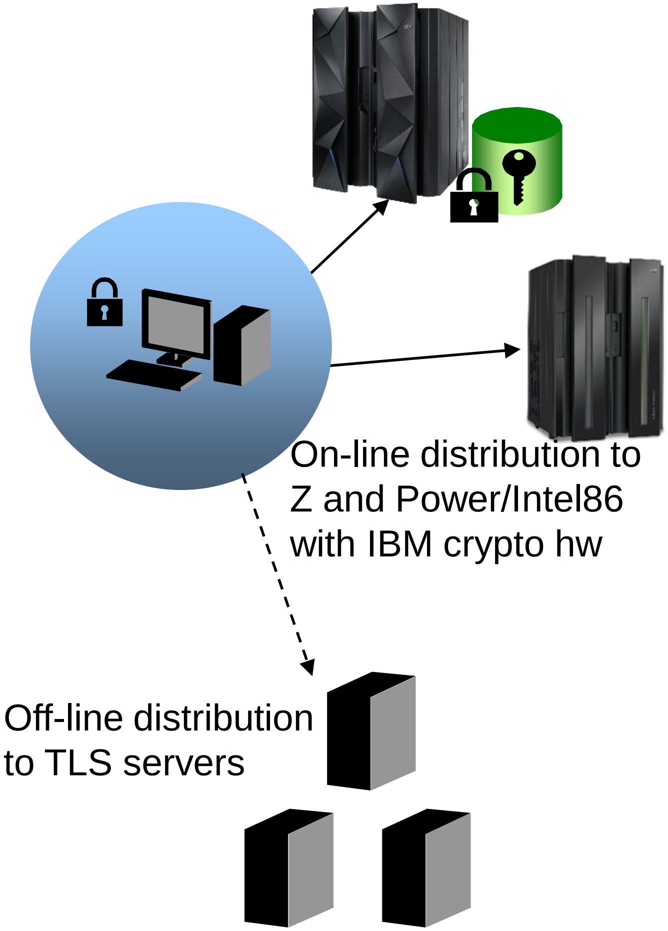
- Centralized key management operations
- Secure hardware – IBM 4765/4767
- Two factor authentication, dual control, group logon, split knowledge, and audit logging
- Support for several workstations
- Printer

Database

- Configuration data
- Keys and metadata
- Audit log (also in SMF)
- Hosted on one LPAR or server
- Available on z/OS, Windows, Linux, AIX

Key stores

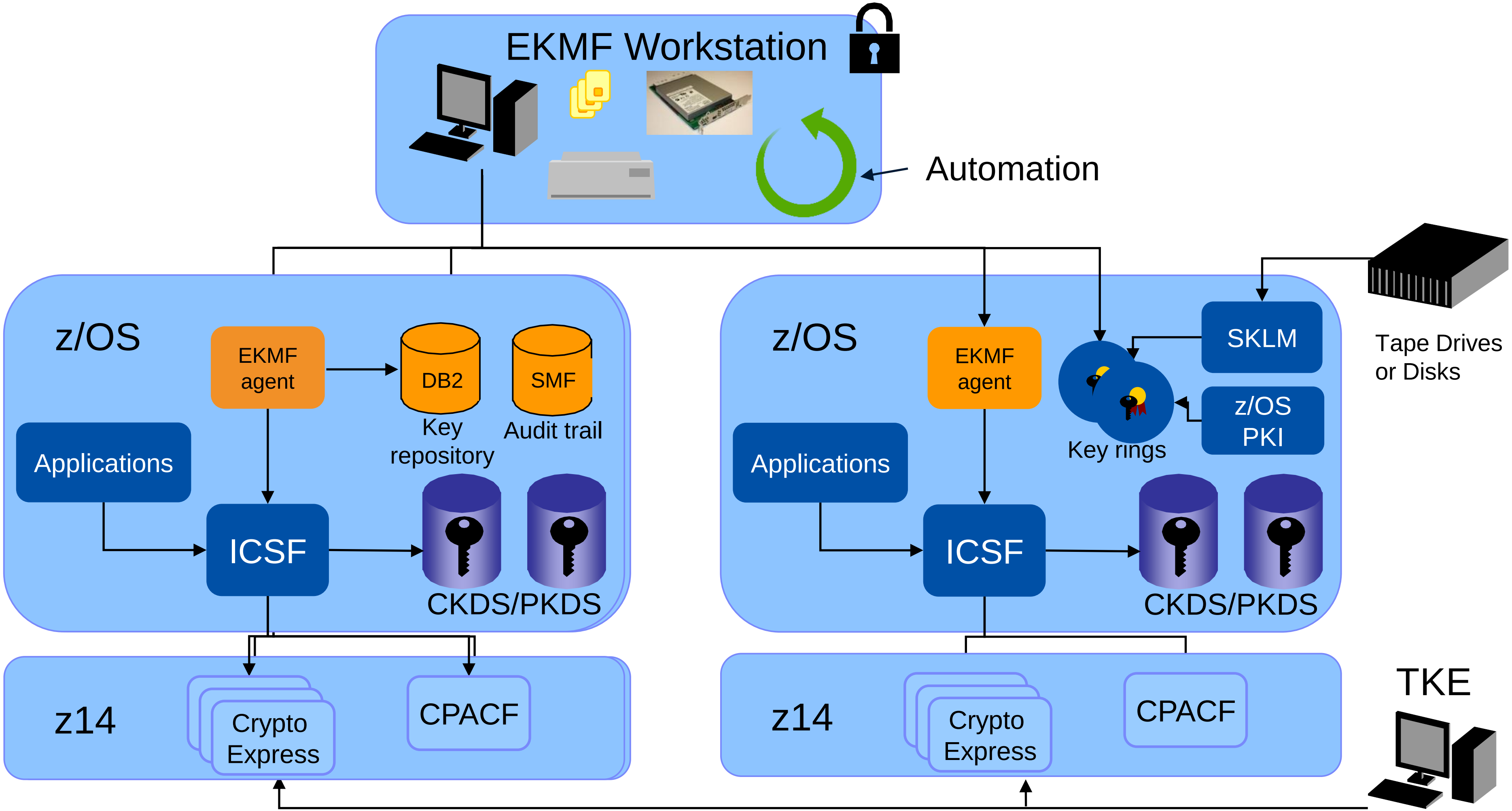
- Distribution – Push mechanism
- Online with all Z machines and all servers with 4765/4767 in the enterprise, managing the keys in ICSF / 476x key stores
- Supports: ICSF, CCA, RACF, Websphere DataPower, Thales, TLS, PKCS#11



On-line distribution to Z and Power/Intel86 with IBM crypto hw

Off-line distribution to TLS servers

Architectural Overview – EKMF and IBM Z crypto ecosystem



Thank You

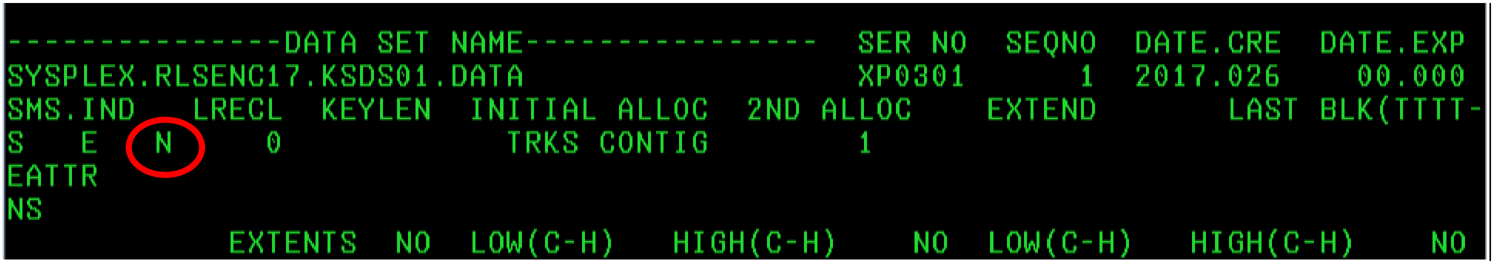
BACKUP

Verifying data set encryption status

Identifying an encrypted data set by data set attributes

1) Volume

- **LISTVTOC** – displays volume level information
  - Data set info includes new encryption attribute under field ‘SMS.IND’

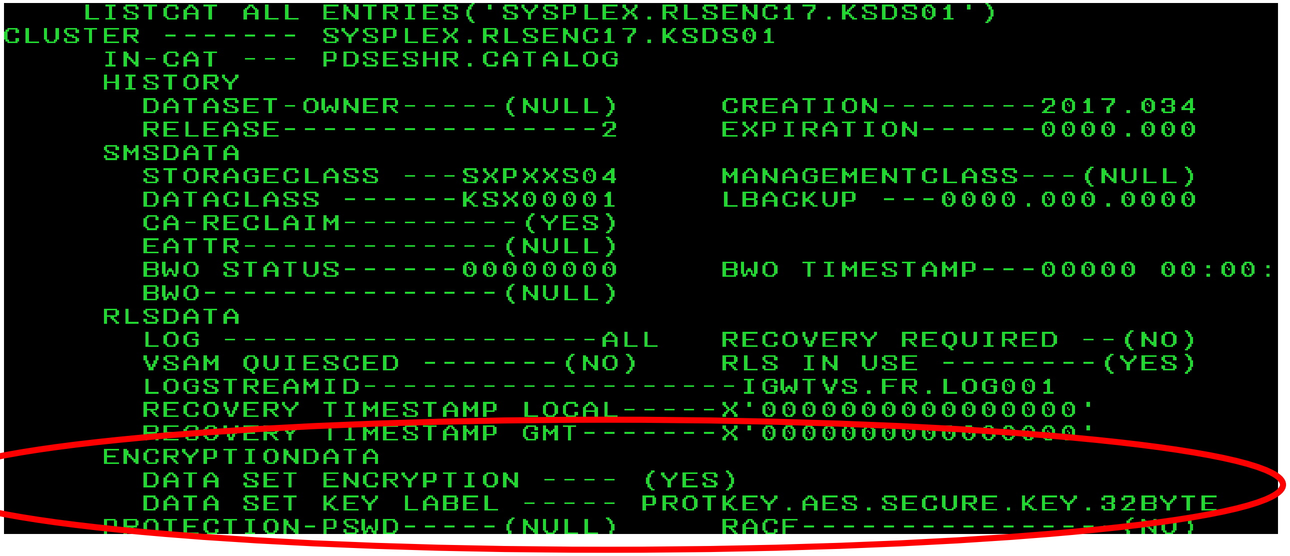


```
---------------DATA SET NAME----------------  SER NO  SEQNO  DATE.CRE  DATE.EXP
SYSPLEX.RLSENC17.KSDS01.DATA                  XP0301      1  2017.026    00.000
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST BLK(TTTT-
S   E   N     0            TRKS CONTIG        1
EATTR
NS
            EXTENTS  NO  LOW(C-H)   HIGH(C-H)    NO  LOW(C-H)   HIGH(C-H)    NO
```

Identifying an encrypted data set by data set attributes

2) Catalog

- LISTCAT – displays catalog level information
  - Data set info displays key label and Encryption flag

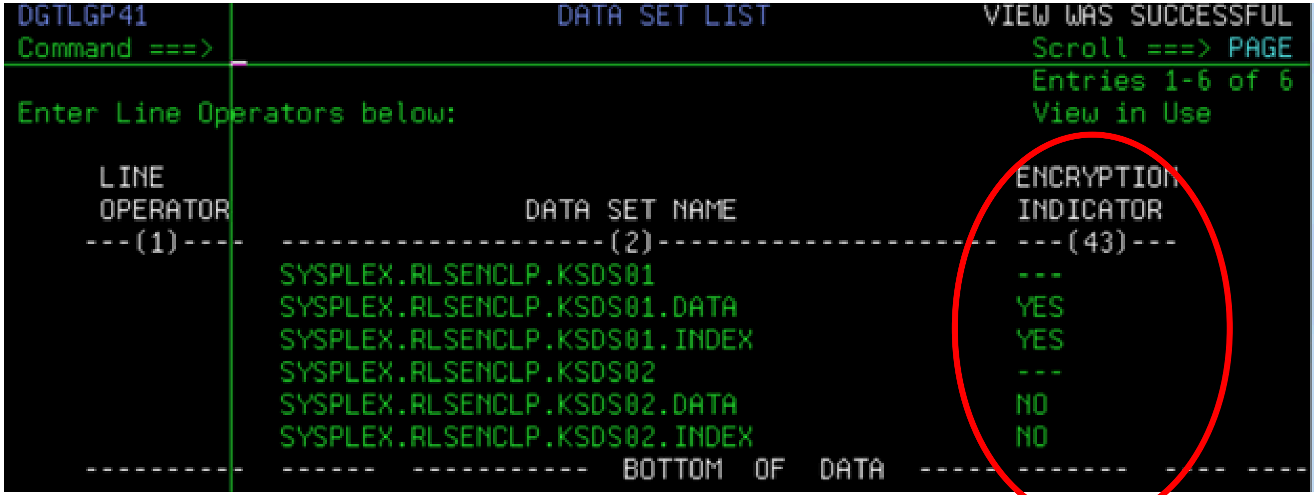


```
    LISTCAT ALL ENTRIES('SYSPLEX.RLSENC17.KSDS01')
CLUSTER ------- SYSPLEX.RLSENC17.KSDS01
     IN-CAT --- PDSESHR.CATALOG
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2017.034
       RELEASE----------------2     EXPIRATION------0000.000
     SMSDATA
       STORAGECLASS ---SXPXXS04     MANAGEMENTCLASS---(NULL)
       DATACLASS ------KSX00001     LBACKUP ---0000.000.0000
       CA-RECLAIM---------(YES)
       EATTR-------------(NULL)
       BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:
       BWO---------------(NULL)
     RLSDATA
       LOG -------------------ALL   RECOVERY REQUIRED --(NO)
       VSAM QUIESCED -------(NO)    RLS IN USE ---------(YES)
       LOGSTREAMID-----------------IGWTVS.FR.LOG001
       RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
       RECOVERY TIMESTAMP GMT-------X'0000000000000000'
     ENCRYPTIONDATA
       DATA SET ENCRYPTION ----  (YES)
       DATA SET KEY LABEL -----  PROTKEY.AES.SECURE.KEY.32BYTE
     PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
```

Identifying an encrypted data set by data set attributes

3) SMS policy

- **ISMF** Data set list panel
  - Encryption flag/type



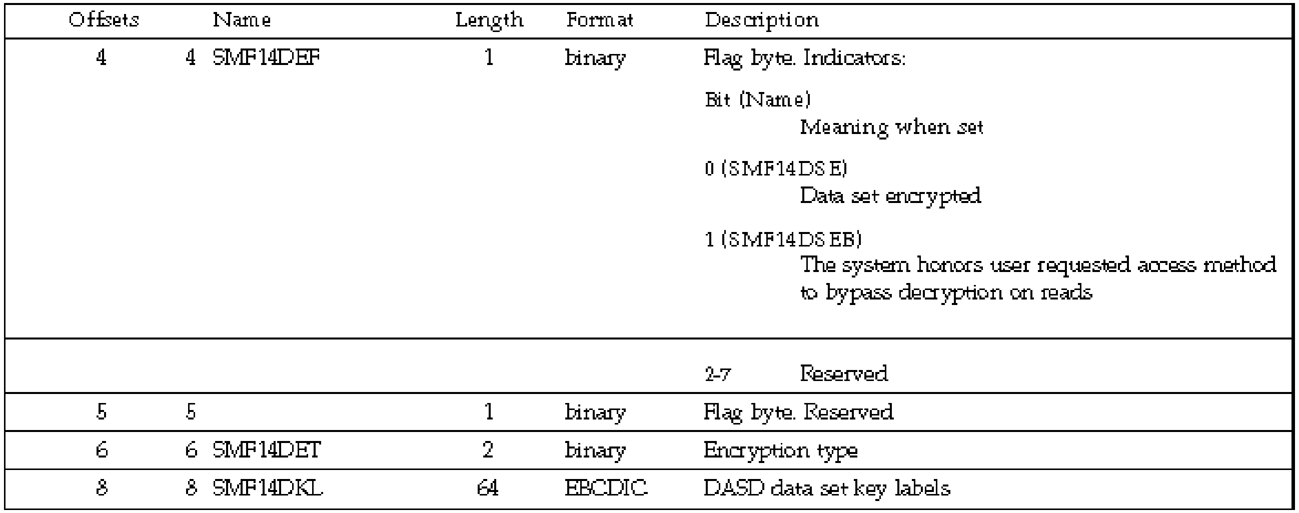
```
DGTLGP41                      DATA SET LIST              VIEW WAS SUCCESSFUL
Command ===>                                                Scroll ===> PAGE
                                                            Entries 1-6 of 6
Enter Line Operators below:                                 View in Use

     LINE                                                 ENCRYPTION
    OPERATOR               DATA SET NAME                  INDICATOR
   ---(1)----  --------------------(2)--------------------  ---(43)---
               SYSPLEX.RLSENCLP.KSDS01                        ---
               SYSPLEX.RLSENCLP.KSDS01.DATA                   YES
               SYSPLEX.RLSENCLP.KSDS01.INDEX                  YES
               SYSPLEX.RLSENCLP.KSDS02                        ---
               SYSPLEX.RLSENCLP.KSDS02.DATA                   NO
               SYSPLEX.RLSENCLP.KSDS02.INDEX                  NO
   ----------  ------  -----------  BOTTOM  OF  DATA  -----  -------  ----  ----
```

Note: In order to display the Encryption Indicator, make sure “Acquire Data from Volume – Yes” is selected in DATA SET SELCTION ENTRY PANEL

Identifying an encrypted data set by SMF

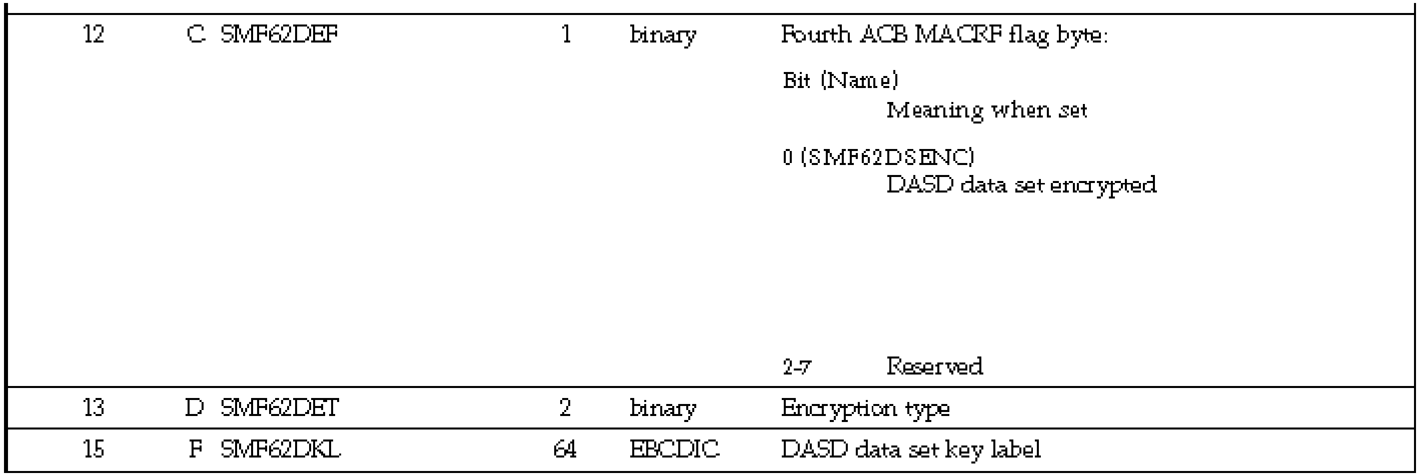
- **SMF records**
  - **SMF Type 14/15** (Sequential data sets)
    - New DASD encryption section with key label and encryption type fields



Offsets		Name	Length	Format	Description
4	4	SMF14DEF	1	binary	Flag byte. Indicators: Bit (Name) Meaning when set 0 (SMF14DSE) Data set encrypted 1 (SMF14DSEB) The system honors user requested access method to bypass decryption on reads
					2-7 Reserved
5	5		1	binary	Flag byte. Reserved
6	6	SMF14DET	2	binary	Encryption type
8	8	SMF14DKL	64	EBCDIC	DASD data set key labels

Identifying an encrypted data set by SMF

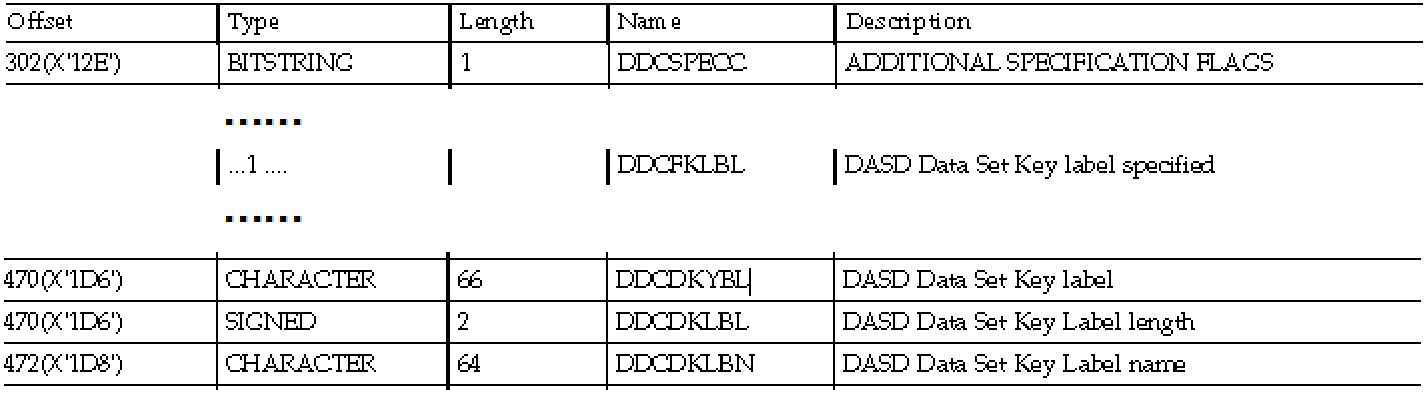
- **SMF records**
  - **SMF Type 62** (VSAM data sets)
    - New DASD encryption information with key label and encryption type fields



12	C	SMF62DEF	1	binary	Fourth ACB MACRF flag byte: Bit (Name) Meaning when set 0 (SMF62DSENC) DASD data set encrypted 2-7 Reserved
13	D	SMF62DET	2	binary	Encryption type
15	F	SMF62DKL	64	EBCDIC	DASD data set key label

Identifying an encrypted data set by DCOLLECT

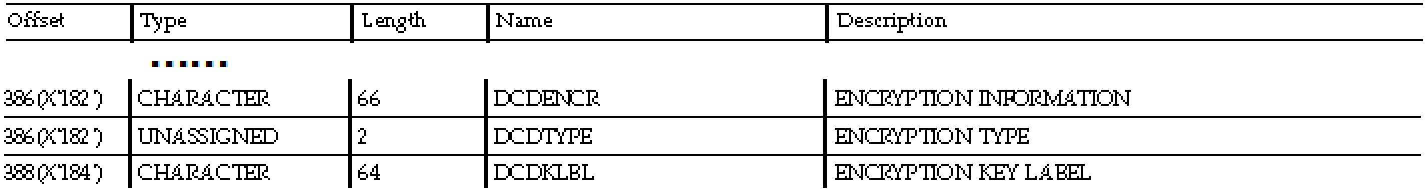
- **DFSMS Data Collection Facility**
  - DCOLLECT – system/data level information
    - Data class definition record Type 'DC': New key label field



Offset	Type	Length	Name	Description
302(X'12E')	BITSTRING	1	DDCSPECC	ADDITIONAL SPECIFICATION FLAGS
	....			
	...1		DDCFKLBL	DASD Data Set Key label specified
	....			
470(X'1D6')	CHARACTER	66	DDCDKYBL	DASD Data Set Key label
470(X'1D6')	SIGNED	2	DDCDKLBL	DASD Data Set Key Label length
472(X'1D8')	CHARACTER	64	DDCDKLBN	DASD Data Set Key Label name

Identifying an encrypted data set by DCOLLECT

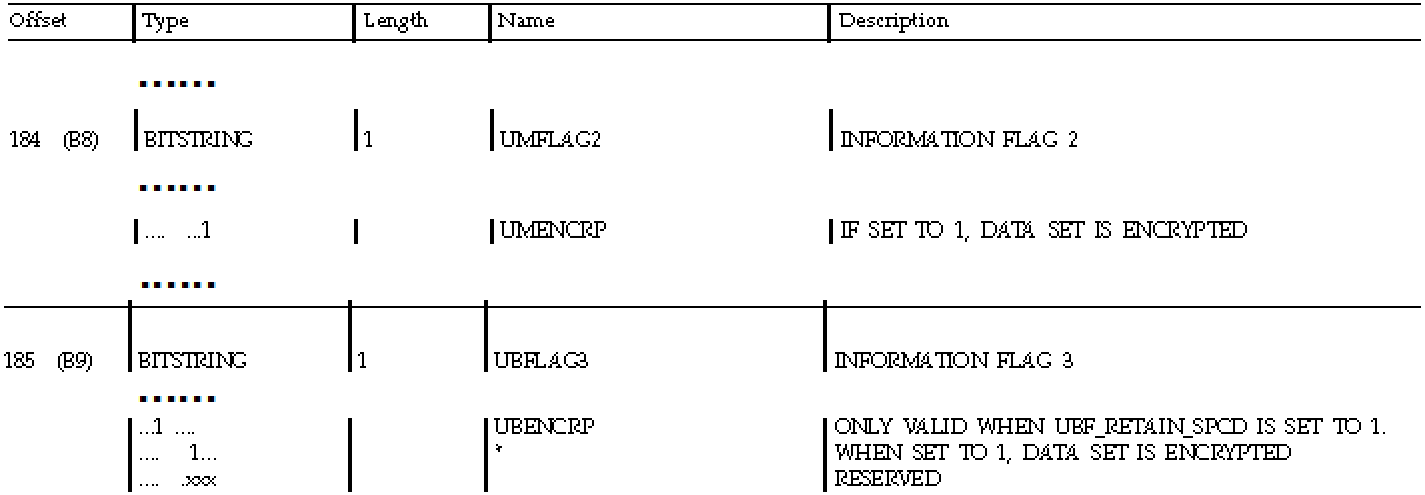
- **DFSMS Data Collection Facility**
  - **DCOLLECT** – system/data level information
    - Data set info record Type 'D' : New key label field



Offset	Type	Length	Name	Description
	......			
386 (X'182')	CHARACTER	66	DCDENCR	ENCRYPTION INFORMATION
386 (X'182')	UNASSIGNED	2	DCDTYPE	ENCRYPTION TYPE
388 (X'184')	CHARACTER	64	DCDKLBL	ENCRYPTION KEY LABEL

Identifying an encrypted data set by DCOLLECT

- **DFSMS Data Collection Facility**
  - **DCOLLECT** – system/data level information
    - HSM migration/backup record: Encryption flag



Offset	Type	Length	Name	Description
	......			
184 (B8)	BITSTRING	1	UMFLAG2	INFORMATION FLAG 2
	......			
	1		UMENCRP	IF SET TO 1, DATA SET IS ENCRYPTED
	......			
185 (B9)	BITSTRING	1	UBFLAG3	INFORMATION FLAG 3
	......			
	...1 .... 1... xxx		UBENCRP *	ONLY VALID WHEN UBF_RETAIN_SPCD IS SET TO 1. WHEN SET TO 1, DATA SET IS ENCRYPTED RESERVED

Identifying encryption SW support by Programming Interfaces

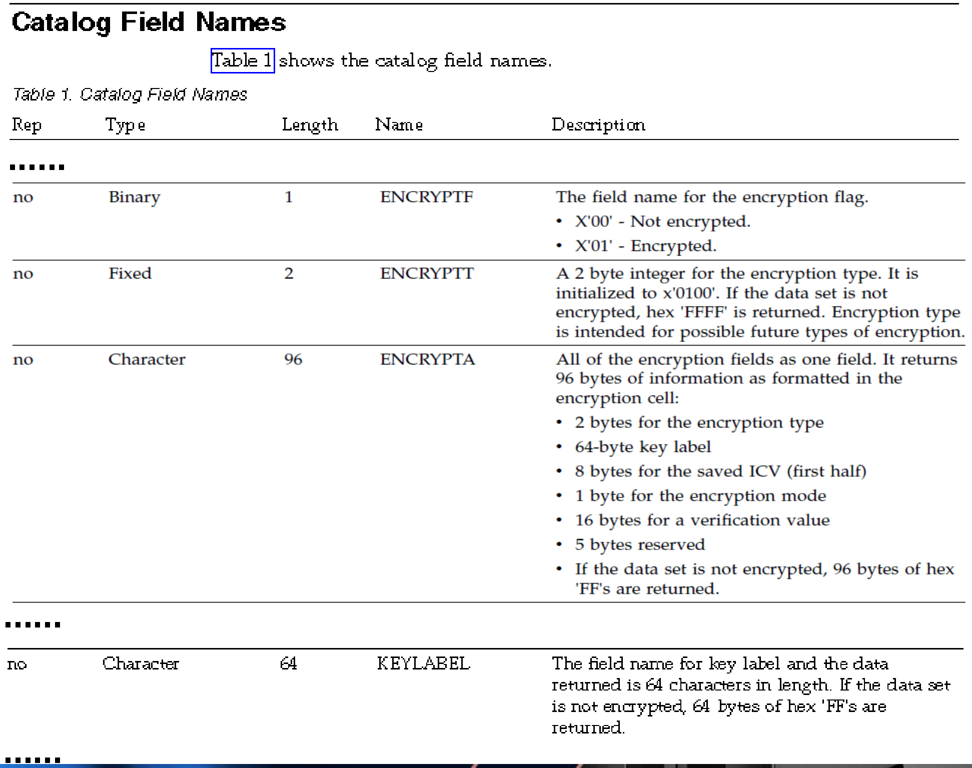
- **DFSMS Features Area (DFA)**
  - **DFAENCRYPT** New flag to indicate DFSMS data set encryption SW installed

60 (3C)	Bit string	4	DFAFEAT9	Features byte 9
	1...		DFAJ3AA	JES3_ALLOC_ASSIST ENABLED
	.1..		DFAMEMUX	Reserved
	…1. ….		DFAPDSEG	PDSE Generation support is installed
	…1 ….		DFAZEDCCMP	zEDC Compression support is installed
	…. xxx.			
	.... …1		DFAENCRYPT	Data set encryption support is installed
…				

Identifying an encrypted data set by Programming Interfaces

1) Catalog

- CSI (catalog search interface)
  - Key label, Encryption flag/type, Encryption cell



Catalog Field Names

Table 1 shows the catalog field names.

Table 1. Catalog Field Names

Rep	Type	Length	Name	Description
......				
no	Binary	1	ENCRYPTF	The field name for the encryption flag. • X'00' - Not encrypted. • X'01' - Encrypted.
no	Fixed	2	ENCRYPTT	A 2 byte integer for the encryption type. It is initialized to x'0100'. If the data set is not encrypted, hex 'FFFF' is returned. Encryption type is intended for possible future types of encryption.
no	Character	96	ENCRYPTA	All of the encryption fields as one field. It returns 96 bytes of information as formatted in the encryption cell: • 2 bytes for the encryption type • 64-byte key label • 8 bytes for the saved ICV (first half) • 1 byte for the encryption mode • 16 bytes for a verification value • 5 bytes reserved • If the data set is not encrypted, 96 bytes of hex 'FF's are returned.
......				
no	Character	64	KEYLABEL	The field name for key label and the data returned is 64 characters in length. If the data set is not encrypted, 64 bytes of hex 'FF's are returned.
......				

Identifying an encrypted data set by Programming Interfaces

2) BSAM/QSAM macro

- **ISITMGD** – returns attributes related to sequential data sets
  - Encryption flag **ISMENCRP** ON if the DASD data set is encrypted by the access methods.