IBM Education Assistance

Solution (Epic) Name: JES2 Encryption of SPOOL data sets

Agenda

- Trademarks
- Session Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Validation During ESP
- Session Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Session Objectives

- In this session we will introduce the new JES2 SPOOL Encryption and Compression function implemented in JES2 2.4.
- This new function provides a simple, transparent and consumable approach to enable extensive encryption and compression of SPOOL data sets.
- Select data sets may be identified for compression only.

Overview

- Who (Audience)
  - Security Administrators desiring to extend pervasive encryption to JES data
  - JES2 programmers trying to manage spool space usage
- What (Solution)
  - Extend the DFSMS model for encryption to JES managed data sets
  - Support compression to reduce amount of data to encrypt
- Wow (Benefit / Value, Need Addressed)
  - Pervasive encryption of sensitive information improves security of the data
  - Reducing size of data using compression can
    - Reduce storage requirements
    - In some cases improve performance of managing the data

Overview

- Security Administrators who are required to protect sensitive data on SPOOL can leverage the z Systems hardware encryption through existing policy management without application changes.
- Similar to DFSMS, this involves defining a record in the CKDS data set which can be identified and accessed via a 64 character key label.
- Use of this key label is secured via RACF profiles.
- JCL parameter DSKEYLBL or JESJOBS class profiles can be used to identify selective SYSOUT and instream data sets to be encrypted.
- Data to be encrypted will first be compressed providing storage efficiency.
- New COMPRESS= option on OUTCLASS(x) statement allows SPOOL data set to be compressed (even if not encrypting the data)

Usage & Invocation

- New function will encrypt JES data sets that reside on SPOOL
  - Instream data sets
  - SYSOUT data sets
- Optionally only compress JES data sets on SPOOL
  - Supports compression without encryption

Usage & Invocation – Key Labels

- Key labels are required to encrypt/decrypt data
- Key labels can be specified using
  - RACF – associated with new format JESJOBS class profile
    - ENCRYPT.*nodename.userid.jobname.dsname*
    - New JES segment and KEYLABEL field to store the label
  - JCL – DSKEYLBL keyword on the DD card (DD *, DATA, and SYSOUT)
    - Must pass FACILITY class profile check to use (like DFSMS data sets)
    - Acts as an override of any default key label provided in a JESJOBS class profile
  - A generic JESJOBS profile can provide system wide default key label
- ENCRYPT format JESJOBS profile lookup is only used to obtain key label
  - There is no authority check associated with these profiles
  - UACC and access lists are ignored

Usage & Invocation – Key Labels

- Access to key labels uses same rules as DFSMS pervasive encryption
  - Must have read access to the profile in the CSFKEYS class
  - Conditional access is supported using WHEN(CRITERIA(SMS(DSENCRYPTION)))
- Intent was to allow installations to use the same key label setup defined for DFSMS pervasive encryption
  - SPOOL encryption thought of as an extension of data set encryption
- CSFKEYS access to key labels in is addition to existing checks
  - JESSPOOL class access
  - SECLABEL dominance checking

Usage & Invocation – Key Labels

- Key label management patterned after DFSMS.
- Key labels are 1 to 64 characters. The first character must be alphabetic or a national character (#, $, @). The remaining characters can be alphanumeric, national or a period(.). Alpha characters must be capital letters.
- Key labels are defined in the CKDS by the ICSF administrator
  - See ICSF publication *z/OS Cryptographic Services Integrated Cryptographic Service Facility Application Programmer's Guide* for a description of key labels.
  - Also you will find the following to be very useful when enabling encrypted keys in callable services: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.csfb300/enuenc.htm
- For the JCL keyword DSKEYLBL, the key label value must be in quotes
  - Normal JCL rules for special characters

Usage & Invocation – Key Labels

- Examples
  - RACF
    - RDEFINE JESJOBS ENCRYPT.POK.IBMUSER.BWTJOB.TEST2 UACC(READ) JES(KEYLABEL(PAYROLL))
  - JCL
    - SYSOUT
      - //SYSPRINT DD SYSOUT=*,DSKEYLBL='PAYROLL'
    - Instream data set
      - //SYSUT1 DD DATA,DLM=RR,DSKEYLBL='ACCOUNTING'

Usage & Invocation – Data Compression

- Compression can be requested based on SYSOUT class
  - New option COMPRESS= on OUTCLASS(x) statement
    - Compression or blank truncation will be done, not both
  - New ability to alter most OUTCLASS parameters via $T command
- Encryption will imply compression
- Data sets using update mode GET/PUT not eligible for compression
  - Update record after it is written , GET record using RBA, update contents, PUT
  - Internally use by the JOURNAL data set
  - Is it used outside of internal IBM code?
    - Generic tracker call added to document update mode GET/PUT usage

Usage & Invocation – Activation

- New keyword ADVANCED_FORMAT indicates if new structures supporting encryption and compression are enabled.
  - Also implements other internal improvements
- Found on the $T SPOOLDEF command
  - Example: $T SPOOLDEF, ADVANCED_FORMAT=ENABLED
    - Activates the use of new format structures used by encryption and compression
  - Example 2: $T SPOOLDEF,ADVANCED_FORMAT=DISABLED
    - Encryption/compression of data sets is no longer performed
- **If ADVANCED_FORMAT is ENABLED at least once, down level members will no longer be allowed to join the MAS**
  - Down level members do not have code to process the new format structures
- $D SPOOLDEF,ADVANCED_FORMAT will display the current setting
- Short form alias keyword ADVF can also be used
  - Example: $T SPOOLDEF,ADVF=ENABLED

Usage & Invocation – SSI 80 updates

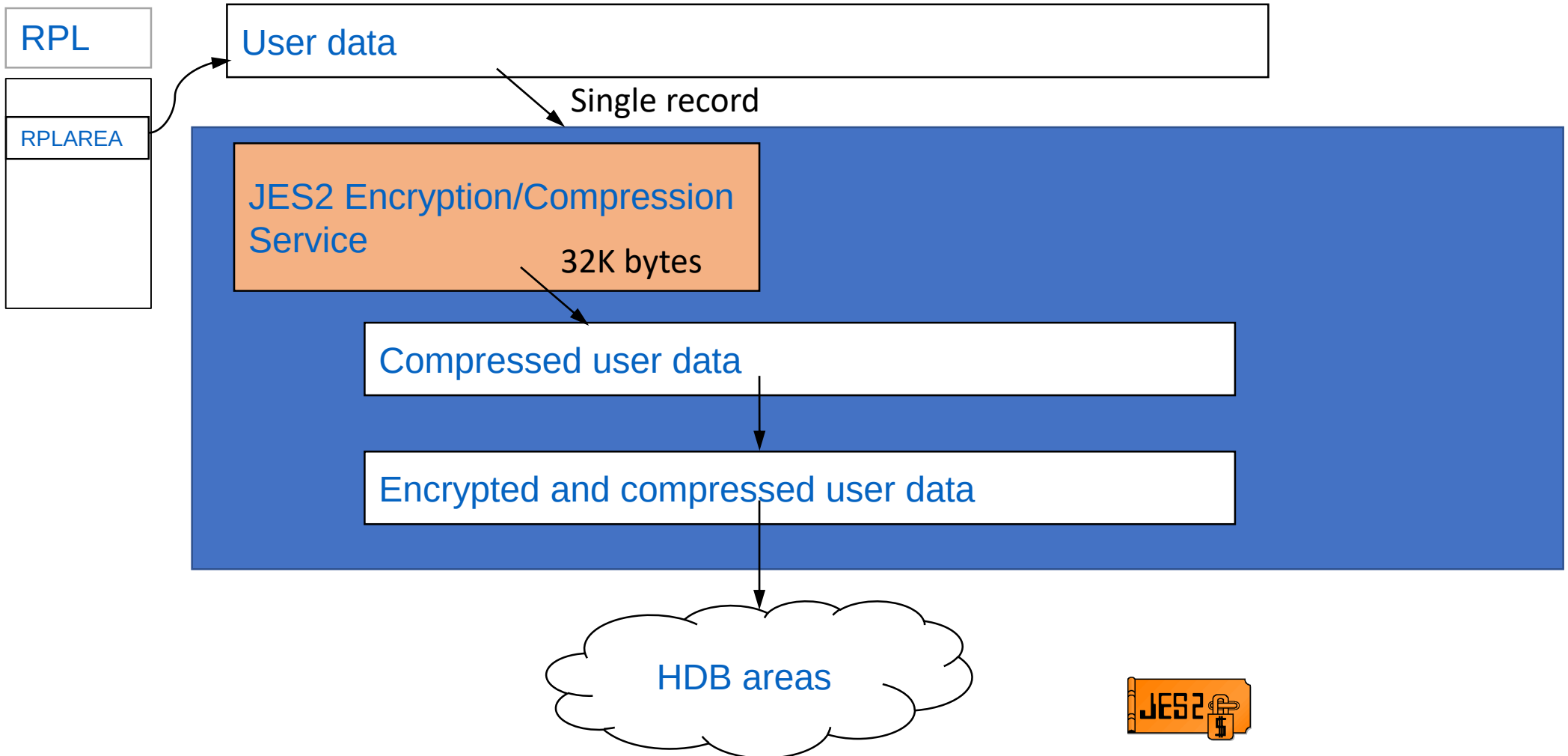
- Extended Status SSI (SSI 80) has been updated to return an additional section of encryption/compression information
  - Returned in a Verbose request
  - Section title is “SYSOUT Element Encryption Security Section” (STATSEES)
  - Reports indicators for whether a data set is encrypted and/or compressed
  - Reports the length and value of a key label associated with the data set

Usage & Invocation – Encryption/Compression Service

- New JES common IAZ service.
- Each encryption/compression service object scoped to data set open.
- Two flavors – PUT and GET.
- Provides one time encryption/compression setup per data set open.
- Also lessens encryption/compression overhead by allowing up to 32K bytes of data per call to relevant service.

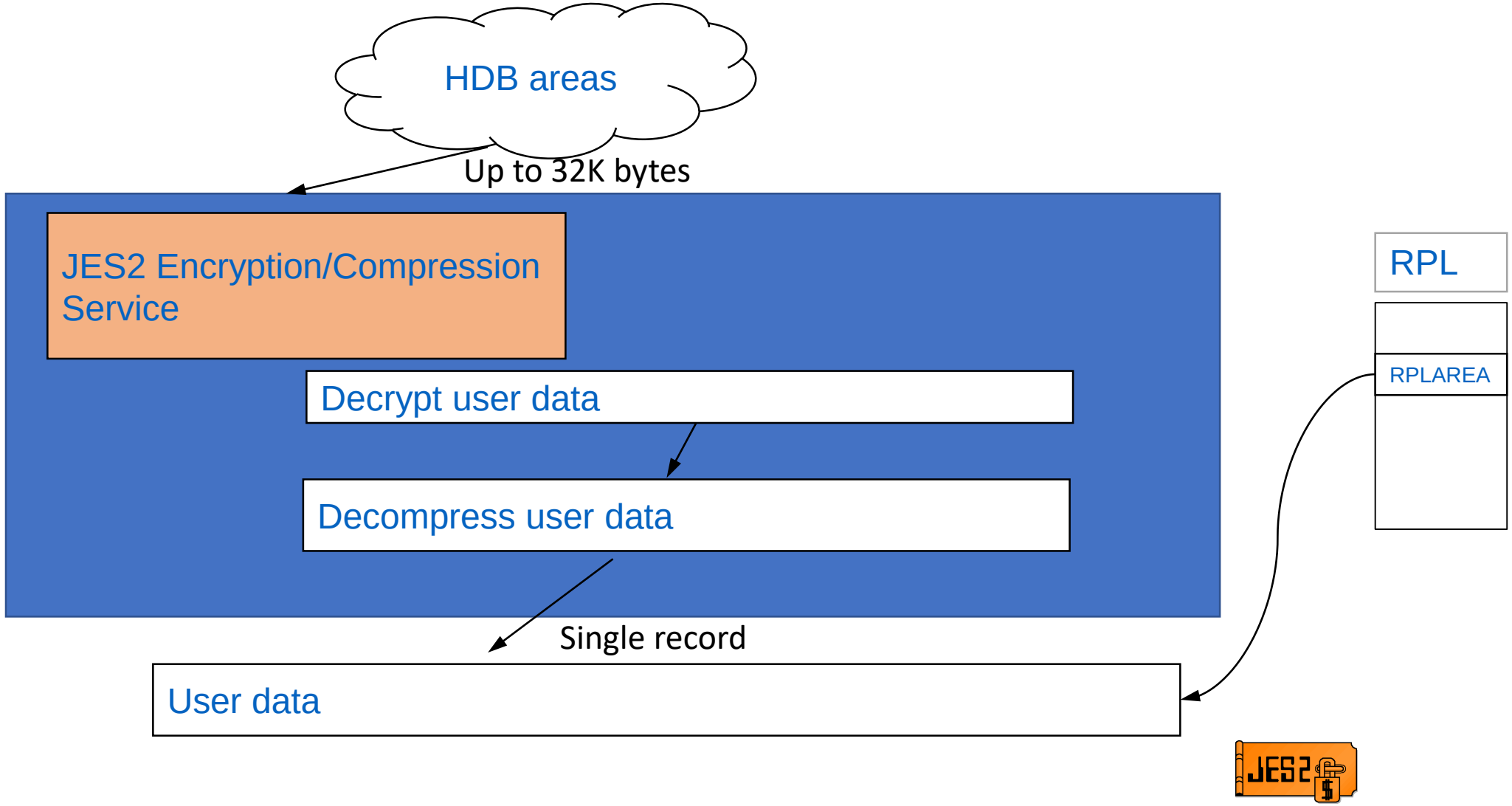
Usage & Invocation – Service –> Put

- Goal is to encrypt at the record level, and do minimal decryption.
- One time setup cost at data set open



Usage & Invocation – Service –> GET

- GET processing (including local printers and FSS):



Interactions & Dependencies

- New format HDB to support new options
  - MTTR -> MQTR changes
  - Additional fields added to HDB
    - Record numbers, Additional MQTRs, Control information
    - Intended to improve overall read processing
- Changes incompatible with older releases
  - New releases will tolerate old data formats but not other way around
- New option on SPOOLDEF to activate new format “One way” switch similar to past SPOOL changes (relative addressing, etc)
  - Once switched, cannot start older release into MAS
  - All members must be on new release
  - Prerequisite for activating compression and encryption
    - *Once active*, only controls for encryption are in RACF (FACILITY and JESJOBS class profiles)

Interactions & Dependencies

- To exploit this item, all systems in the Plex must be at the new z/OS level: Yes
- Relevant key labels are defined through ICSF.
- Software Dependencies
  - Data sets using update mode GET/PUT not eligible for compression.
    - Update record after it is written … GET record using RBA, update contents, PUT
    - Generic tracker will help identify these data sets.

- Hardware Dependencies
  - Compression hardware is used if available. If compression requested and hardware not available then data will not be compressed but no error will be issued.
    - SSI enhancements (and SDSF) will indicate if data is compressed/encrypted.
  - Compression along with encryption require the one way switch previously described.
- Exploiters
  - Any JES2 user.

Migration & Coexistence Considerations

- From JES2 z/OS 2.2 or z/OS 2.3
- APAR OA53860 needed on z/OS 2.2 or z/OS 2.3 member to coexist in a MAS with z/OS 2.4
- APAR OA53860 is also highly recommended for fall back
- Some new data structures created by z/OS 2.4 JES2 may result in problems if OA53860 is not installed.

Installation

- None

Validation During ESP

- ESP Validation Requested: Yes
- Use support to encrypt SYSOUT and instream data sets.

Session Summary

- New function adds protection for highly sensitive customer data.
  - Modeled after DFSMS support
- Storage utilization is enhanced with compression attribute.
- Encryption/compression can easily be identified on a data set or job basis.
- RACF used to lock down secure use of CKDS key labels.

Appendix

- Publications
- z/OS V2R4.0 JES Application Programming – SA32-0987-40
- z/OS V2R4.0 JES2 Commands – SA32-0990-40
- Z/OS V2R4.0 JES2 Diagnosis - GA32-0993-40
- z/OS V2R4.0 JES2 Initialization and Tuning Guide – SA32-0991-40
- z/OS V2R4.0 JES2 Initialization and Tuning Reference – SA32-0992-40
- z/OS V2R4.0 JES2 Installation Exits – SA32-0995-40
- z/OS V2R4.0 JES2 Macros – SA32-0996-40
- z/OS V2R4.0 JES2 Messages – SA32-0989-40
- z/OS V2R4.0 MVS JCL Reference - SA23-1385-40
- z/OS V2R4.0 MVS Using the Subsystem Interface – SA38-0679-40