Db2 for z/OS:
Introduction to encryption

Agenda

- Encryption environment
  - Remote access protection, data access protection
- Pervasive encryption
- Data set encryption and Db2 for z/OS
  - Transparent to application
- Column-based encryption in Db2 for z/OS
  - Application-aware
- Questions
- Summary

Encryption environment

Payment Card Industry
Data Security Standard

PCI DSS COMPLIANT

$ 4.88M

Average cost of a data breach in 2024 [1]

GDPR

European Union General
Data Protection Regulation

Digital Operational
Resilience Act
(DORA)

27.7%

Likelihood of an organization having a data breach over the next two years [2]

Health Insurance
Portability and
Accountability Act

HIPAA

1. Source: Cost of a Data Breach Report 2024 -- https://www.ibm.com/reports/data-breach
2. Source: 2017 Ponemon Cost of Data Breach Study: Global Analysis

Remote access protection

Network Layer

- IP Filtering
- IPSec
- Intrusion Detection Services
- AT-TLS

RACF

- Certificate Authentication (AT-TLS)
- Multi-factor Authentication
- Passphrase
- Passticket
- Kerberos
- DSNR Authorization
- SERVAUTH
- ID Token (IDT – JWT type tokens) *

Db2

- Trusted Connections
- System Profile Monitoring
  - MONITOR type CONNECTIONS FOR SECURITY *
- DSNLEUSR Stored Procedure

* Db2 13 for z/OS

Data access protection

Separation of Duties

- Granular Authorities
- * Install and Migrate without SYSADM
- Trusted Contexts and Roles
- * Transfer Ownership

Privacy controls

- Row Permission
- Column Mask
- Row level encryption (column encryption)
  - ~~ENCRYPT_TDES~~
  - ~~IBM Guardium Encryption Tool for IMS™ and Db2 for z/OS~~
  - * ENCRYPT_DATAKEY BIF (V12R1M505)
- ***Data set Encryption***

Audit

- Audit policies
- Audit changed data
  - Temporal
  - GENERATED ALWAYS AS clause
  - Temporal for security tables in Catalog *

*** Db2 12 for z/OS**

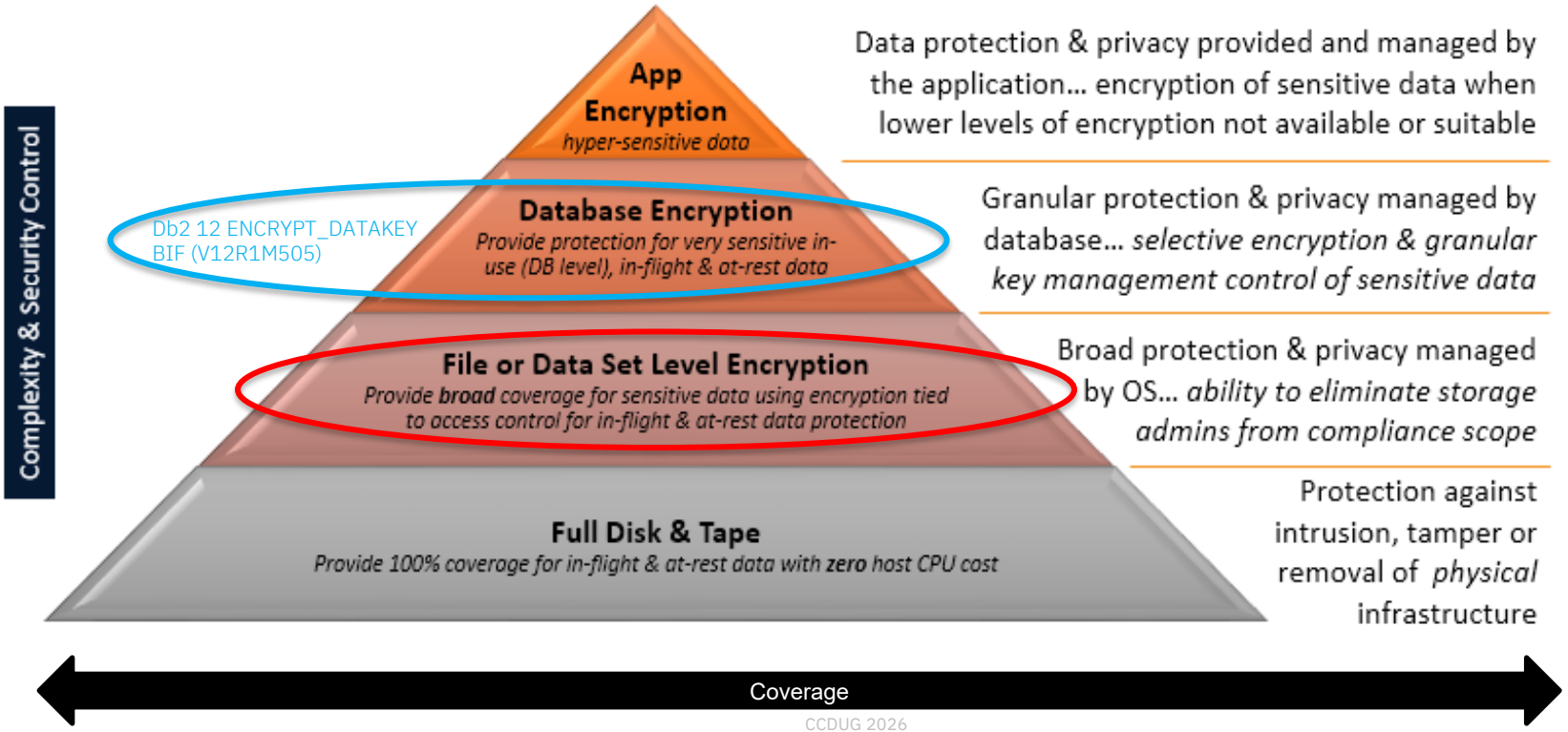
*** Db2 13 for z/OS**

Pervasive encryption

Pervasive encryption (September 2017)

- Intended to provide encryption capability and support broadly across the IT environment
  - Network
  - Z systems (initially z14)
  - Coupling facilities (CFs)
  - Storage area network (SAN)
  - [Storage media – already available]

Multiple layers of encryption of data



Data set encryption and Db2 for z/OS

Db2 support of z/OS data set encryption

- Db2 can transparently encrypt data at rest
  - Without database downtime
  - Without requiring the administrator to redefine objects (avoid disruption)
- No application changes required
- You can:
  - Encrypt active and archive log datasets
  - Encrypt catalog and directory table spaces
  - Encrypt user table spaces
- Utilizes z/OS DFSMS data set encryption support
  - Db2 12 V12R1M502 added additional controls to set up encryption policies using Db2 interfaces

Data set encryption overview

- DFSMS encrypts/decrypts records when written to or read from disk
- DFSMS managed data sets that support encryption of data at rest:
  - BSAM / QSAM
    - Sequential – Extended format only; originally did not include tape
  - VSAM and VSAM/RLS
    - KSDS, LDS, ESDS, RRDS, VRRDS – Extended format only
- Encryption type - AES 256 bit key (XTS, protected key)
- Key Label - A 64-byte label of the key in the ICSF CKDS that is used for the encryption/decryption of the data set
  - ICSF = integrated cryptographic support facility
  - CKDS = cryptographic key data set

DFSMS policy-based encryption

- Data sets are defined as encrypted by specifying a key label during creation of a new data set:
  1. RACF data set profile
  2. JCL, Dynamic Allocation, TSO Allocate, IDCAMS DEFINE
  3. SMS DATACLAS
- During data set open, DFSMS:
  - Checks the user access to the key label
  - Specifies the key label to ICSF to retrieve the secure / protected key from the CKDS
- ICSF:
  - Locates the secure key in the CKDS using the key label specified by DFSMS
  - Calls the adapter to unwrap the key value from the Master key
  - Rewraps the key value under a CPACF wrapping key to make it a protected key
  - Protected key stored in ICSF cache

DFSMS dataset encryption

- Application transparency
  - Data remains encrypted during backup/recovery, migration/recall
  - In memory system or application data buffers remain in the clear
  - Access to the key label is controlled through SAF permissions, in addition to traditional data set permissions
- Segregation of duties
  - Storage administrators need access to the data set but not access to the key label

Steps to enable encryption

1

ICSF Admin

Generates an encryption key and key label. Stores it in the ICSF CKDS

2

Security Admin

Sets up RACF for use of key label

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

– AND –

Permits Db2 use of the key label

In RACF, permit access to new resource in FACILITY class

3

Security / Database / Storage Admin

Associate the key label with the desired data set.

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

– OR –

IDCAMS DEFINE, TSO Allocate, etc.
In Db2, set key label for system objects and user objects

– OR –

In DFSMS, assign to data class

4

Database Admin

Migrate to encrypted data

Online REORG

Non-disruptive

Encrypting Db2 for z/OS system objects (1|3)

3

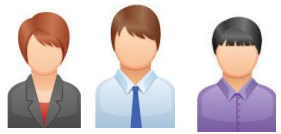
- Options to define a key label used by Db2 (precedence order):
  - 1. Security Admin can set a key label in the DFP segment of RACF data set profile using DATAKEY keyword

  - 2. Database System Admin can set a key label using ENCRYPTION_KEYLABEL system parameter
    - -SET SYSPARM command is required for the zparm value to take effect
    - Group scope: Takes effect on all the members of a data sharing group immediately
    - Security related parameter: Requires installation SYSADM or SECADM authority to set the zparm
    - Db2 DBM1 and MSTR address space IDs must be permitted access to the key label

  OR

  - 2. Storage Admin can set a key label using IDCAMS DEFINE
    - Only option to encrypt active logs; ENCRYPTION_KEYLABEL will not apply

  - 3. Storage Admin can set a key label in the DFSMS data class



Security / Database System Admin / Storage Admin

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

– OR –

In Db2, set key label using system parameter
OR
IDCAMS DEFINE, etc.

– OR –

In DFSMS, assign to data class

Encrypting Db2 for z/OS system objects (2|3)

4

- Active logs
  - Encrypt new active logs
    - Define active log data set as encrypted and issue the SET LOG command NEWLOG option to add the newly defined active log data set to the active log inventory without stopping Db2
  - Encrypt all active logs
    - Stop Db2. Copy the contents of the active log data set to an encrypted data set. Restart Db2.
    - Db2 13: online removal of active log data set; SET LOG REMOVELOG with V13R1M500
- Archive logs (originally only on disk; tape support added z/OS 3.2)
  - New archive logs automatically encrypted based on the key label setting
- Catalog and directory table spaces
  - Execute REORG TABLESPACE utility to encrypt table spaces and index spaces in DSNDB06 and DSNDB01
  - Encrypt DSNDB01.SYSUTILX – Execute RECOVER utility followed by REBUILD INDEX ALL

Database Admin

Online REORG

Can use ENCRYPTION_KEYLABEL system parameter

Encrypting Db2 for z/OS system objects (3|3)

- Display encryption key label using DFSMS interfaces, SMF records
- Run REPORT TABLESPACESET utility to display key label associated for each catalog and directory table spaces using SHOWKEYLABEL option
- Issue –DISPLAY LOG command to obtain current key label information for current active log data sets
- Issue – DISPLAY ARCHIVE command to obtain current key label information for archive log data sets that are in use

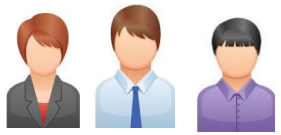
Encrypting Db2 for z/OS user objects (1|3)

3

- Options to define a key label for user objects encryption (precedence order):
  - 1. Security Admin can set a key label in the RACF data set profile DFP segment using DATAKEY keyword
  - 2. Storage Admin (or Database Admin) can set a key label via IDCAMS, TSO, etc...

OR

  - 2. Application Database Admin can set a key label using SQL interfaces: CREATE / ALTER with STOGROUP / TABLE
    - Enabled with APPLCOMPAT V12R1M502
  - 3. Storage Admin can set a key label in the DFSMS data class



Security / Database / Storage Admin

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

– OR –

IDCAMS DEFINE, etc.
OR
In Db2, set key label using SQL interfaces

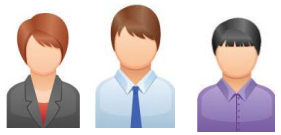
– OR –

In DFSMS, assign to data class

Encrypting Db2 for z/OS user objects (2|3)

3

- SQL CREATE / ALTER STOGROUP – with KEY LABEL option
  - Adds a key label at the storage group level to encrypt all the table spaces using the storage group
    - Only option for multi-table table spaces (deprecated!)
- SQL CREATE / ALTER TABLE – new KEY LABEL option
  - Adds a key label at the table level to encrypt all the table spaces associated with the table
  - Includes explicitly or implicitly created base table space, auxiliary table spaces, XML table spaces, index spaces
  - Supported only for tables that reside in a universal table space (UTS) or a partitioned table space



Security / Database / Storage Admin

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

– OR –

IDCAMS DEFINE, etc.
OR
In Db2, set key label using SQL interfaces

– OR –

In DFSMS, assign to data class

Encrypting Db2 for z/OS user objects (3|3)

- Execute the REORG utility to encrypt existing table spaces
- New table spaces or partitions defined are encrypted using the key label based on the hierarchy (precedence order)
- Run REPORT TABLESPACESET utility to display key label for the table spaces used by each table using the SHOWKEYLABEL option

4

Database Admin

Online REORG to encrypt existing user objects

Utilities considerations (1|2)

- All online utilities support table spaces and indexes whose underlying VSAM data sets are encrypted
- Input / Output data sets
  - Key label can also be specified using
    - JCL DSKEYLBL option
  - Authorization ID of the job requires access to the key label for any encrypted input or output data sets
- Stand alone utilities
  - Authorization ID of the job requires access to the key label for any encrypted data sets

Utilities considerations (2|2)

- Db2 managed table spaces and index spaces
  - Utilities used to convert to encrypted data sets (except when REUSE option is specified)
    - REORG TABLESPACE or REORG INDEX
    - LOAD REPLACE
    - REBUILD INDEX
    - RECOVER from image copies – PIT or full recovery
  - PART or DSNUM option to encrypt / decrypt at the partition level

- User managed table spaces and index spaces
  - IDCAMS DELETE / DEFINE with the KEYLABEL option
  - Execute RECOVER and/or REBUILD INDEX utilities to restore the data
- FlashCopy image copies (FCIC), DFSMSdss concurrent image copies, shadow data sets
  - Allocated with the same key label as the table space or index

Db2 dataset encryption considerations

- Compression
  - Db2 compression works seamlessly with data set encryption
  - Compression is performed first
- Performance
  - There will be small CPU cost for encryption, mitigated by CPACF
    - CP assist for cryptographic function – on-chip
  - zBNA tool (z Systems Batch Network Analyzer) can help estimate
    - **https://www.ibm.com/support/pages/ibm-z-batch-network-analyzer**

Db2 dataset encryption summary

- For Db2 for z/OS, a key label can be defined by the security administrator, DBA, or storage administrator
- DBA can use Db2 REORG utility to seamlessly migrate Db2 data sets to encrypted data sets with no application outages
- Make sure all disaster recovery user ids and sites have access to any key labels used to protect Db2 data sets and the key management system is fully deployed across the enterprise
- Recommendation: plan and implement enterprise security and encryption strategy

Column encryption in Db2 for z/OS

APPLCOMPAT(V12R1M505) built-in functions (BIFs)

- ENCRYPT_DATAKEY (data string, keylabel, algorithm) allows
- Column-based encryption of security-sensitive data
- String and numeric datatypes are supported
  - Resulting datatype is VARBINARY for non-LOB and BLOB for LOB input value
  - Schema change required for cipher text column
- Use of ICSF protected keys and RACF key label protection
  - Primary authid requires permit for key label defined in CSFKEYS class
- AES256D|R – 256-bit AES CBC mode encryption algorithm
  - ‘D’ – fixed initialization vector to generate a cipher text
  - ‘R’ – random initialization vector to generate a cipher text

- DECRYPT_DATAKEY_BIGINT … _CHAR … _INTEGER …
- Datatype dependent BIFs: DECRYPT_DATAKEY_xxxx (encrypted data)

Using ENCRYPT_... and DECRYPT_... BIFs

- ENCRYPT_DATAKEY and DECRYPT_DATAKEY_xxxx
- Application aware
- Application specifies the key label
  - ENCRYPT_DATAKEY input parameters include the key label
- Application specifies the algorithm
  - AES256D – duplicate values in the column will be encrypted the same way
    - Can use for predicate matching
  - AES256R – duplicate values in the same column will be encrypted differently
    - More secure, but no predicate matching
- DECRYPT_DATAKEY...
- DECRYPT_DATAKEY_xxxx - 'xxxx' specifies the data type of output
  - Data type must match in the input data type

Column-based encryption

- Schema change required:
  - New column for the encrypted value
  - You may want to remove, delete, or mask the old column (or new table?)
- Encrypted column(s) remain encrypted in the Db2 buffer pools

Name	Surname	“PII value”
Mark	Rader	123-45-6789

Name	Surname	“PII value”	“encrypted PII”
Mark	Rader	123-45-6789	1011011100101

Questions?

Summary

- Encryption techniques can be applied to Db2 data and interactions including
  - Network – AT-TLS preferred
    - AT-TLS = application transparent – transport layer security
    - Secure ports
  - Data set encryption
  - Column-based encryption

- Db2 for z/OS encryption practice should be part of enterprise-wide security policy

Thank you!

Tori DiDonato
victoria.felt@ibm.com

Mark Rader
mrader@us.ibm.com