IMS
Dataset Level Encryption
(Pervasive)

Dennis Eichelberger
IBM – Washington Systems Center
deichel@us.ibm.com

Pervasive Encryption and IMS

“It’s not a matter of if, but when.”

- Regulatory Compliance
- Industry standards
- Customer satisfaction

Many data breaches occur without immediate knowledge.

Average time from breach to discovery - on the order of 30 days

Some occurrences are later.

There are No
“do overs”

Pervasive Encryption and IMS

Topics

- Pervasive Encryption Umbrella
- IMS OSAM datasets
- IMS OSAM defined as a VSAM Linear Dataset – LDS
- IMS Implementation notes
- Things to look for

Pervasive Encryption

Compression

Algorithms are techniques that exploit redundancy in data to reduce the size of the data representation. Compression algorithms aren't meant to conceal data, but may do so, if the compression algorithm is secret - until somebody reverse-engineered the algorithm.

Algorithms may be kept secret with compression.

Compression works best on non random or consistent data.

Encoding

The process of changing data representation. The same data may be represented in different encodings (i.e. binary, hex, decimal, base64,...) which aren't (usually) meant to change the data's meaning. Size expansion / reduction may be a result of a different encoding. Encodings aren't meant to conceal data, but may do so, if the encoding / decoding algorithm is secret - until somebody reverse engineers the algorithm.

Algorithms are kept secret with encoding.

Encoding must be non random or consistent.

Encryption

The process of concealing information solely based on the secrecy of some smaller value, which is called "a key". Modern schemes allow for advanced functionality, such as operation on unknown data and guaranteeing the data integrity.

Algorithms are not secret with encryption. **Keys are secret.**

Encryption must be random.

Compression

```
Dennis             Eichelberger                     next
----+----1----+----2----+----3----+----4----+----5-
Dennis*Eichelberger*next…
```

where * is a code indicating length and character ' ' – blanks

Encoding

```
Dennis*Eichelberger*next…
----+----1----+----2----+----3----+----4----+----5-
De+is*Eichelb#g#*next…
```

where * is a code indicating length and character
where + is a code indicating two 'n's
where # is a code indicating the characters 'er'

Encryption

```
Dennis             Eichelberger                     next
----+----1----+----2----+----3----+----4----+----5-
^6/?;.SDIVM k@3@#4)msx.,po/onc][\istrtw{&\;sw=
```

a bunch of random stuff

Pervasive Encryption

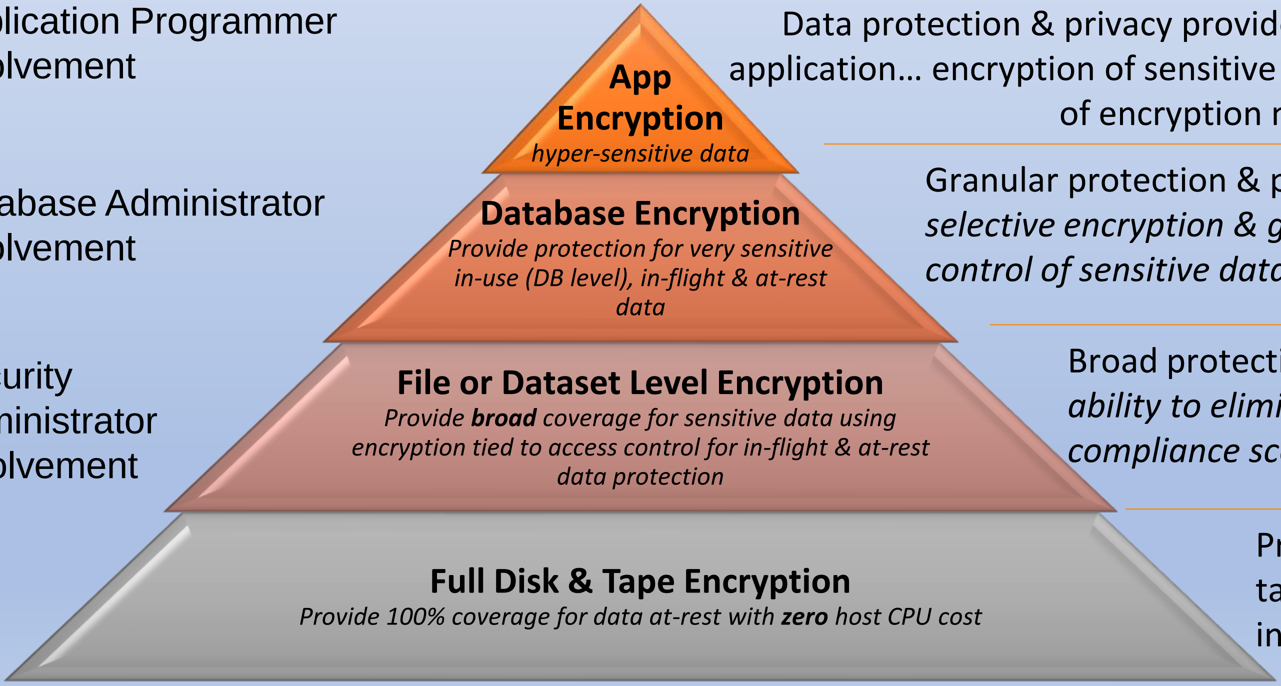
Levels of Encryption depend on where the data needs to be encrypted

Implementation depends on resources and expected results

Application Programmer involvement

Database Administrator involvement

Security Administrator Involvement



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

IBM

z/OS Data Set Encryption

- Enabled by policy
- Transparent to application
- Access control uses SAF
- Uses have differing access types
- Uses protected encryption keys managed by the host

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*

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

- Broadly encrypt data at rest
- Covers VSAM, Db2, IMS, Middleware, Logs, Batch, & ISV solutions[1]
- Encrypt in bulk for low-overhead
- Utilizes IBM z Systems integrated cryptographic hardware

Database Encryption

IBM GDEz Data Encryption for Db2 and IMS Databases

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

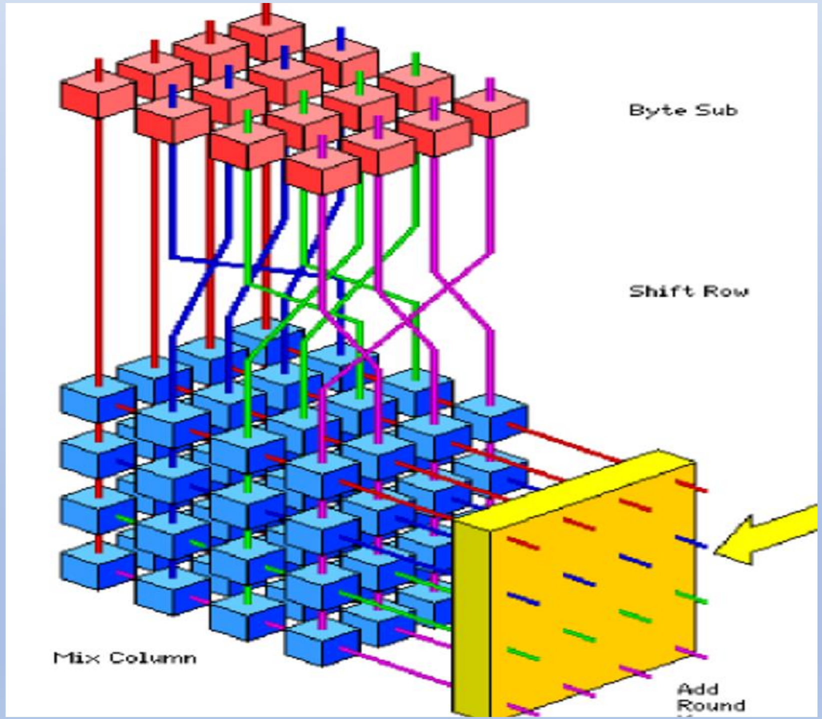
Granular protection & privacy managed by database processes… *selective encryption & granular key management control of sensitive data*

- Encrypts sensitive data at the Db2 row and column levels and IMS segment level
- Transparent to applications
- Separation of Duties (SOD) and granular access control
- Protects Data-In-Use within memory buffers
- Clear text data cannot be accessed outside DBMS access methods
- Persists the encrypted data in logs, image copy data sets, DASD volume backups
- Utilizes IBM z Systems integrated cryptographic hardware

Encryption Algorithms

- (Advanced Encryption Standard)
  - 128-, 192- or **256**- bit, commercially used algorithm

Rijndael Algorithm
- Block Cipher (16-byte blocks)
- 128, 192, 256-bit Key Length
- Multiple Rounds
- Four Steps per Round
  - Byte Substitution
  - Shift Row
  - Mix Column
  - Add Round Key



What are Encryption Keys?

- Master Keys
  - Used to encrypt and store user keys into the CKDS (Cryptographic Key Data Set)
  - Loaded into the CEXnn hardware, and stored NO WHERE else
- User Keys (Data Encrypting Keys)
  - Generated via ICSF services
  - Stored inside the CKDS
  - Public or Private
  - Clear or Secure
  - Used by the IBM GDEz Encryption Tool along with encryption algorithm to convert user data to Ciphertext for Database Encryption

Encryption Keys

- 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
  - CEXn hardware not required

- Secure Key
  - Key is only ever exposed within the 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, 3, 4, 5, 6
  - APIs available via Integrated Cryptographic Support Facility (ICSF)
  - Can be used from Java on z/OS platform

Encryption Keys

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

IMS OSAM Datasets

- Clear Key vs. Secure Key Performance
- Protected Key
  - A Secure Key wrapped in encryption copied from the hardware
  - Reduces I/O type calls
  - Obscures Secure Key in memory as an encrypted object
  - Performance improvement over Secure Key
  - Greatest benefit for bulk processing against database

IMS Datasets

VSAM Access Method – KSDS ESDS

OSAM access method is owned by IMS labs

- Optimized for IMS processing
  - Sequential buffering definitions
  - Dynamic Sequential buffer based on volume

- Very fast Channel Program

- Exclusive of DFSMS – Media Manager processing

IMS OSAM Datasets

IMS 15.2 Feature

OSAM May be allocated as a VSAM Extended Format Linear Dataset

- High Performance Ficon - z/HPF Capable
- HyperWrite Capable
- Dataset Level Encryption enablement
    - Encryption occurs at Dataset I/O time using Media Manager – DFSMS
- Sequential buffering available

IMS OSAM Datasets

APAR PI85987 PTFs UI66794/UI66795
base code for OSAM encryption

APAR PH17824 PTF UI67433, pre-req's PI85987
First fix

Customer cannot enable the OSAM function by installing these 2 APARs.
They must be installed before installing the following
They are pre-req's to the following

The 15.2 marker APAR PH16882 / UI67505 is required to enable.

IMS OSAM Datasets

IMS 15.2

Implications

Implementation

Change OSAM to VSAM Linear Dataset LDS

HiperWrite of HPF

Implement Dataset Level Encryption

Database Datasets must be reallocation and data copied.

This may require a database outage.

Note that existing VSAM KSDS & ESDS does not need to be changed to LDS to allow encryption at the dataset level.

IMS OSAM Datasets

VSAM Extended Format Linear Dataset Attributes

- Minimum CI size of 4096 bytes
- Any multiple of 4096 bytes
- Maximum CI size of 32768 bytes

IMS OSAM Datasets

Allocating VSAM Linear Datasets

- JCL

- SMS
  - STORCLAS
  - MGMTCLAS
- IDCams input

IMS OSAM Datasets

Allocating OSAM Linear Datasets

Allocating without Encryption

```
DEFINE CLUSTER -
(NAME(data set name) -
CONTROLINTERVALSIZE(4096) -
SHAREOPTIONS(3 3) -
CYL(200) -
DATACLAS(dataclas)-
STORCLAS(storclas)-
LINEAR)
```

Note that the VSAM **BWO** parameter is not applicable to a LINEAR Dataset definition

IMS OSAM Datasets

```
DEFINE CLUSTER -
    (NAME(DDS0027.IMSA.DI99PART) -
    VOLUME(SMSE01) -
    CONTROLINTERVALSIZE(2048) -
    SHAREOPTIONS(3 3) -
    CYLINDERS (20 0) -
    LINEAR)
```

Defining a CISIZE of 2048

Allocates a CISIZE of 4096

Allocations are rounded up to the next CISIZE 4096 increment.
e.g. 6144 becomes 8192

```
DATA ------- DDS0027.IMSA.DI99PART.DATA
  IN-CAT --- CATALOG.ESYSMVS.USER
  HISTORY
    DATASET-OWNER-----(NULL)     CREATION--------2020.198
    RELEASE----------------2     EXPIRATION------0000.000
    ACCOUNT-INFO-----------------------------------(NULL)
  PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
  ASSOCIATIONS
    CLUSTER--DDS0027.IMSA.DI99PART
  ATTRIBUTES
    KEYLEN-----------------0     AVGLRECL---------------0   UFSPACE-----------8192     CISIZE--------------4096
    RKP--------------------0     MAXLRECL---------------0   EXCPEXIT---------(NULL)    CI/CA----------------180
    SHROPTNS(3,3)   RECOVERY     UNIQUE           NOERASE   LINEAR        NOWRITECHK   UNORDERED          NOREUSE
```

IMS OSAM Datasets

Database Implications

- Database buffer definitions may need updating in the DFSVSMxx members
  - Datasets with increase CI size will no longer use smaller buffers
    - E.g. Dataset increased from 1024 byte block is now 4096 bytes and will require a larger buffer
- Root Anchor Points – RAPS may need adjustment to maintain the same number of roots per block
- HDAM RBN value may need adjusting
  - The RBN must not be higher than IMS allows
    - 8 Gigabytes
    - 4 Gigabytes if using HALDB OLR capability

```
    DBD
NAME=HDO8,ACCESS=(HDAM,OSAM),
    RMNAME=(DFSHDC40,5,120)
```

IMS OSAM Datasets

Database Implications

- Monitor Database Buffer Usage
  - IMS DC Monitor
  - IMS Performance Analyzer

- Monitor Database usage
  - Pointer Checker
  - Space Monitoring

- Adjusting DBD information may require a Database outage

Recommendation:
Perform and test any DBD adjustments in a test environment to reduce any potential outage to the customers

IMS OSAM Datasets

Implications

Performance

Implementation of Dataset Level Encryption

- May impact buffer usage
- May Impact locking times

IMS OSAM Datasets

Benefits

OSAM May be allocated as a VSAM Extended Format Linear Dataset

Using multiple of 4096 CISize

- High Performance Ficon - z/HPF Capable
- HyperWrite Capable to reduce mirroring latency
- Dataset Level Encryption enablement
  - Encryption occurs at Dataset I/O time using Media Manager – DFSMS
  - IMS buffers in memory are not encrypted
- IMS Tooling uses Media Manager for offload processes to zIIP

IMS OSAM Datasets

z/OS Data Set Encryption…

- **Data sets are defined as encrypted by specifying a key label at the *creation* of a new data set:**
  - SAF data set profile: Rules that associate a key label with a data set name pattern, via new **DATAKEY** parameter of the DFP RACF segment.
  - JCL, dynamic allocation, or TSO allocate (new **DSKEYLBL** parameter)
  - IDCAMS DEFINE (new **KEYLABEL** parameter)
  - SMS DATACLAS (new key label attribute)

- **Application transparency: Data is encrypted or decrypted when accessed via supported access methods:**
  - Data encryption/decryption occurs as data is written to or read from disk.
  - In-memory system or application data buffers remain in the clear.
  - Data remains encrypted during backup/recovery, migration/recall, and replication.
  - Access to key label is controlled through SAF permissions, in addition to traditional data set permissions.

- **Programs accessing data sets using other access methods (Media Manager, direct channel programs) cannot access data sets encrypted by DFSMS without modification.**

IMS OSAM Datasets

Encrypting OSAM Linear Datasets

Allocating for Using Encryption

```
DEFINE CLUSTER -
(NAME(data set name) -
CONTROLINTERVALSIZE(4096) -
SHAREOPTIONS(3 3) -
CYL(200) -
DATACLAS(dataclas)-
STORCLAS(storclas)-
KEYLABEL(keylabel)-
LINEAR)
```

Keylabel is the label for the encryption key

IMS OSAM Datasets

```
DEFINE CLUSTER -
    (NAME(DDS0027.IMSA.DI99PART) -
    VOLUME(SMSE01) -
    CONTROLINTERVALSIZE(4096) -
    SHAREOPTIONS(3 3) -
    KEYLABEL(DSELABEL) -
    CYLINDERS (20 0) -
    LINEAR)
```

Defining a Key Label to enable encryption

Data Set Encryption --- (YES)

Data will be encrypted when written and decrypted when read by DFSMS

```
RLSDATA
  LOG ----------------(NULL)    RECOVERY REQUIRED --(NO)     FRLOG ------------(NULL)
  VSAM QUIESCED -------(NO)     RLS IN USE ---------(NO)     LOGREPLICATE-------------(NO)
  LOGSTREAMID-----------------------------(NULL)
  RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
  RECOVERY TIMESTAMP GMT-------X'0000000000000000'
ENCRYPTIONDATA
  DATA SET ENCRYPTION-----(YES)
PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
```

IMS OSAM Datasets

Encryption Implementation

- Prepare an Encryption key & Key Label
  - Security or the like resource
- Unload the Database Dataset
- Delete / Define the new Database Dataset as a VSAM LINEAR dataset with the KEYLABEL definition name (supplied by security)
- Implement any update DBDs – if needed
- Implement any DBRC updates VSAM ➔ OSAM & LDS
- Implement any DBRC update *hlq*
- Load the Database Dataset
  - The Load job must have SAF authority to access the Key Label **AND** the dataset itself.

IMS OSAM Datasets

Encryption Considerations

- Any job or task needing access to read or update the data must authorized for use of BOTH the dataset AND the Key Label
    - IMS tasks
    - DLI & DBB jobs
    - File Manager – often under a TSO ID
    - Batch Backout
    - Log analysis routines
    - Forward Recovery
    - Pointer Analysis
    - IMS Tooling performing similar functions
    - User programs performing the same functions

    - There may be more – watch for
    - IEC161I messages with new Conditions, Reason and Explanation codes.

z/OS Data Set Encryption: Encryption By Policy

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

The preferred method of enabling data set encryption is to specify a key label in the DFP segment of the RACF Data Set profile.

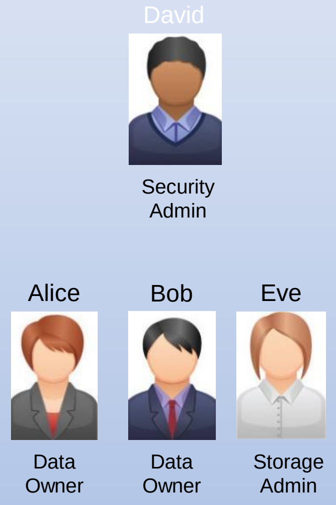
```
ALTDSD “PROJECTA.DATA.*’ UACC(NONE)
DFP(RESOWNER(ALICE) DATAKEY(key-label))
```

In the following example, Alice can read and update the data set. Bob can read the data set. Eve can read, update, delete, rename, move, or scratch the data set. But what content will they see?

```
PERMIT “PROJECTA.DATA.*” ID(ALICE)
ACCESS(UPDATE)

PERMIT “PROJECTA.DATA.*” ID(BOB) ACCESS(READ)

PERMIT “PROJECTA.DATA.*” ID(EVE) ACCESS(ALTER)
```



33

z/OS Data Set Encryption: Viewing the Content

Any user that needs access to the data set content in the clear must have access to the key label.

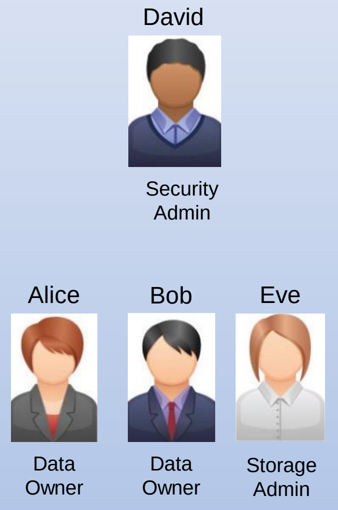
```
RDEFINE CSFKEYS * UACC(NONE)

RDEFINE CSFKEYS key-label UACC(NONE)

PERMIT key-label CLASS(CSFKEYS) ID(ALICE) ACCESS(READ)

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

PERMIT key-label CLASS(CSFKEYS) ID(EVE) ACCESS(NONE)
```



Eve has no access to the key label. So, even though she has UPDATE authority to manage the data set, she cannot view its contents.

In this example, Alice and Bob have access to the key label. So, they can view the data set contents in the clear.

34

IMS OSAM Datasets

What to Expect if a the USERID of an Address Space is Not Authorized to a Key Label

In general OPEN will fail for the data set in question, so the IMS behavior should be no different than if any other type of OPEN failure occurs. You would expect to see an IEC161I (for an open failure) followed by a RACF authorization failure notification like:

```
IEC161I data_set_name,,VCATSHR
ICH408I USER(OMVSADM ) GROUP(SYS1 ) NAME(OMVS ) 872 key_label CL(CSFKEYS )
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )

IEC161I 069(00000008,0000271C)-162,ISTEKLWP,RELOAD,DBX303,,,  617
IEC161I GEN500.PERVENC.DBX303,,CATALOG.PRJT01
```

followed by whatever error message the process deems appropriate for the open failure.

IMS OSAM Datasets

Encryption With Compression

Compress First

Encrypt Second

- Compression depends on patterns
  - Repeating characters
  - Repeating words or snippets

- Encryption creates random characters
  - Not usually repeated
  - Not usually patterned

Encrypting First will nullify the attempt to compress.

Pervasive Encryption and IMS OSAM Datasets

Pros & Cons

CON – OSAM access to a VSAM LDS may use more CPU than native OSAM

PRO – OSAM access to a VSAM LDS allows dataset level encryption

CON – OSAM data buffers in memory are in the clear

PRO – OSAM access to a VSAM LDS may still us Sequential Buffering

PRO – OSAM access to a VSAM LDS may us HPF and Hiperwrite

PRO – Use of VSAM LDS outperforms VSAM KSDS / ESDS

CON – SAF Key authority may require further administration

PRO – Application independent

Pervasive Encryption and IMS OSAM Datasets

Anomaly

Customer Task

Convert current Guardium Encryption (Database Level) to Dataset Level Encryption

Pervasive Encryption and IMS OSAM Datasets

Anomaly

Customer Database

PHIDAM HALDB with 5 Partitions

After converting to OSAM LDS and Dataset Level Encryption

Observed a 28% increase in CPU

Observed 25% - increase in BMP run time

Converting Guardium Encryption to Dataset Level Encryption A cautionary Case

Guardium encryption is at a Database level.

Invoked using the COMPRTN = exit in the DBD or SEGM statements of a database descriptor.

Some segments may be encrypted and others not.

Most cases only the data portion is encrypted
no Key fields or Indexes.

,COMRTN=(GENCRYPT,DATA)

Converting Guardium Encryption to Dataset Level Encryption A cautionary Case

Guardium encryption is at a Database level.

Runtime	=	27.10
EXCP	=	491,968

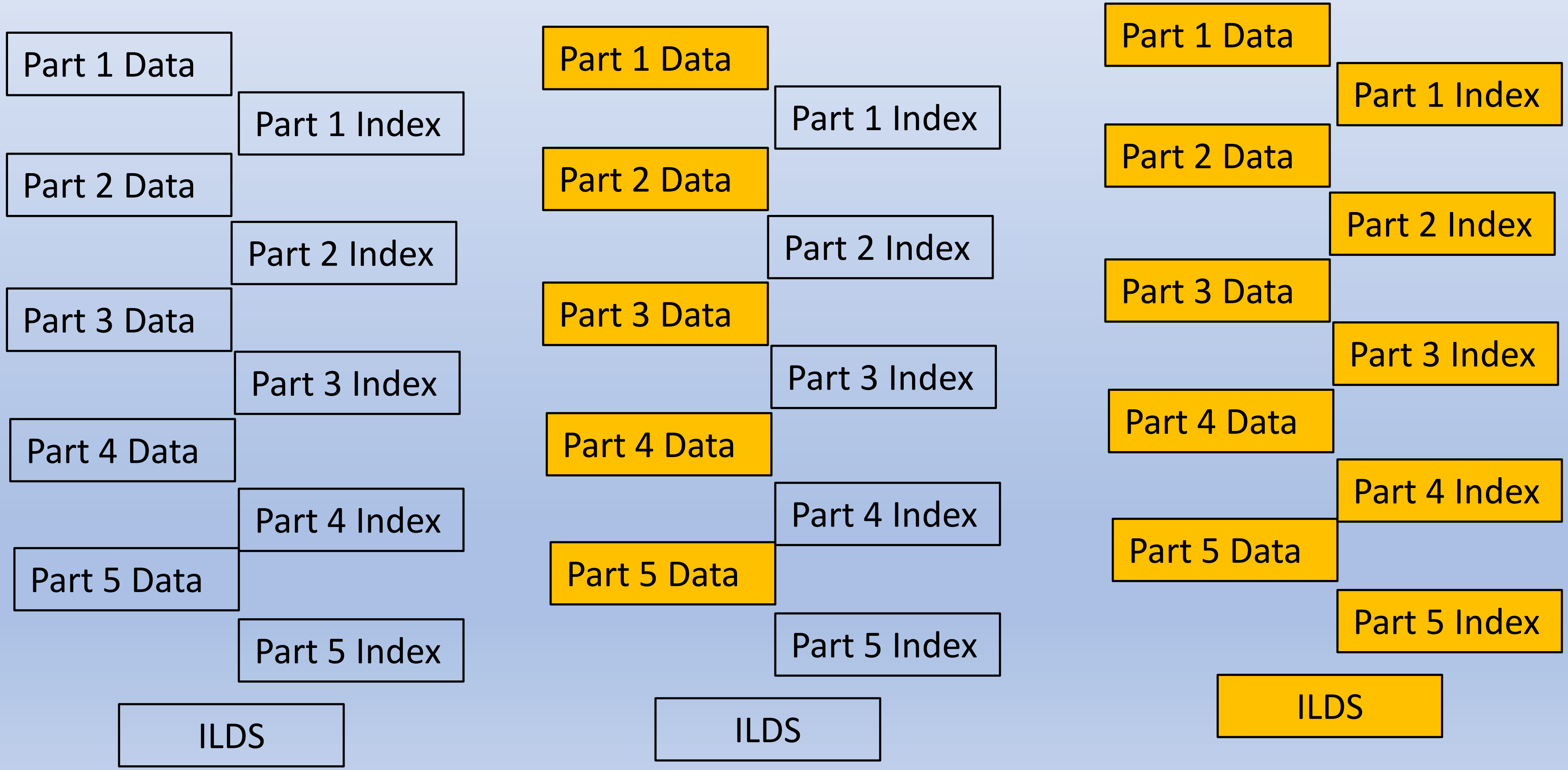
DFSMS Dataset level Encryption.

Runtime	=	37:00.5
EXCP	=	2,250,977

Runtime increase	=	27%
EXCP increase	=	79%

Converting Guardium Encryption to Dataset Level Encryption A cautionary Case

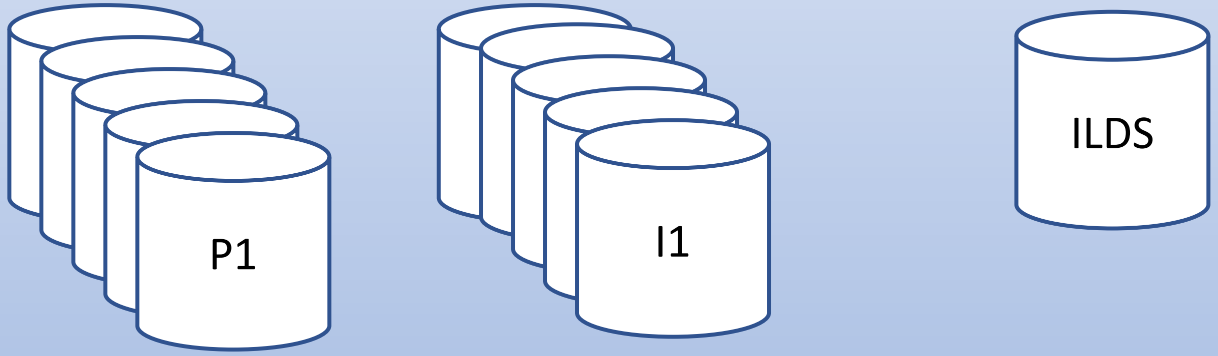
HALDB - PHIDAM 5 Partitions



Pervasive Encryption and IMS OSAM Datasets

Anomaly

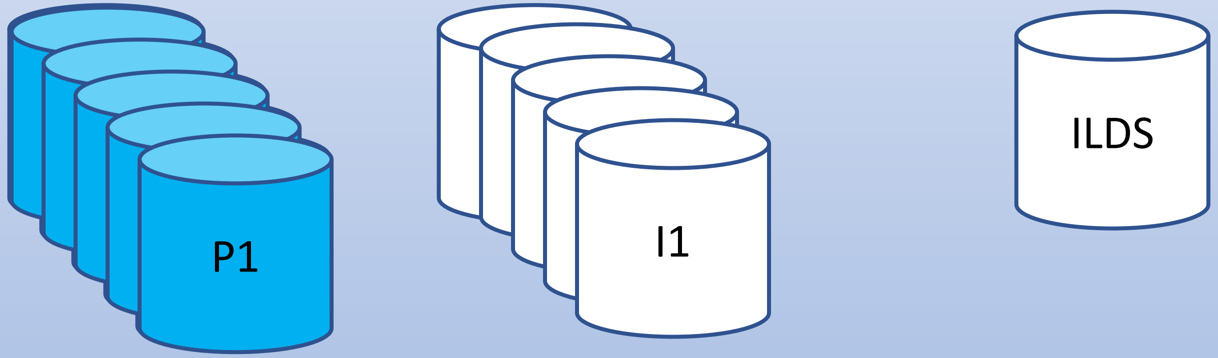
Dataset allocations for PHIDAM with 5 partitions



Pervasive Encryption and IMS OSAM Datasets

Anomaly

Dataset allocations for PHIDAM with 5 partitions



Five datasets will process encryption

Guardium Encryption

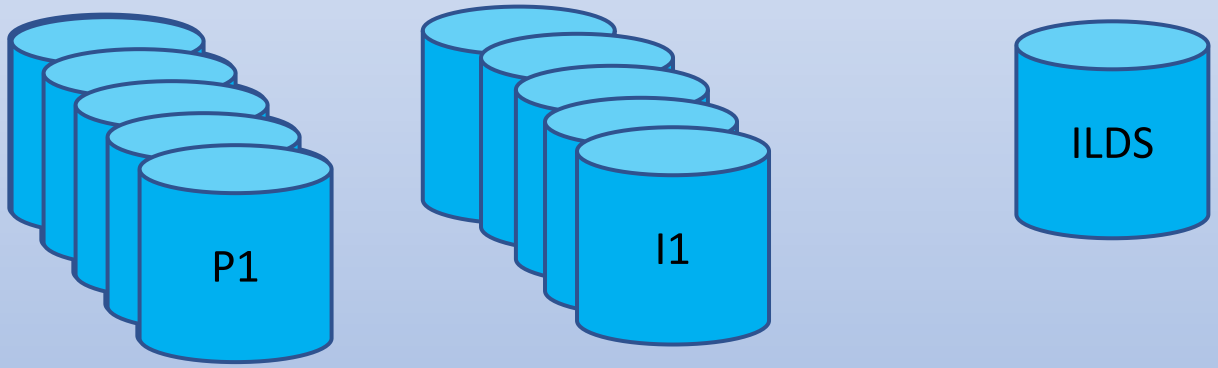
Implemented with COMPRTN exit in the IMS DBD definition.

Usually does not include the index information

Pervasive Encryption and IMS OSAM Datasets

Anomaly

Dataset allocations for PHIDAM with 5 partitions



Dataset Level Encryption

Implemented with SAF definitions for All datasets

Includes the index information

Number of Datasets increase by 55%
Time increase of 28%

~~Five datasets will process encryption~~

Eleven datasets will process encryption

IMS OSAM Datasets

IMS 15.2

OSAM May be allocated as a VSAM Extended Format Linear Dataset

- High Performance Ficon - z/HPF Capable
- HyperWrite Capable
- Dataset Level Encryption enablement
    - Encryption occurs at Dataset I/O time using Media Manager – DFSMS
- Sequential buffering available

IMS OSAM Datasets

APAR PI85987 PTFs UI66794/UI66795

base code for OSAM encryption

APAR PH17824 PTF UI67433, pre-req's PI85987

First fix

Customer cannot enable the OSAM function by installing these 2 APARs.

They must be installed before installing the following

They are pre-req's to the following

The 15.2 marker APAR PH16882 / UI67505 required to enable.

IMS OSAM Datasets

Notes

- Log record x’6204’
  - OSAM Control Blocks
- DFS0451I
  - OSAM I/O error information

The IMS Documentation contains

- Condition codes
- Reason codes
- Status codes

https://www.ibm.com/docs/en/ims/15.4.0?topic=dfs0500i-dfs0451i

IMS OSAM Datasets

IMS 15.2

Implications

Implementation

Change to VSAM Linear Dataset LDS

and / or

Implement Dataset Level Encryption

Database Datasets must be reallocation and data copied.

This may require a database outage.

Note that existing VSAM does not need to be changed to LDS to allow encryption.

Pervasive Encryption and IMS OSAM Datasets

Considerations

What needs to be encrypted?

- Data
    - Likely so. The data contains PCI and PII content
    - Likely so. To meet regulatory requirements
- Index
    - Maybe. It depends on the content. It may contain sensitive fields
    - May be needed to meet regulatory requirements
- ILDS
    - Likely not. It's an internal IMS structure

It probably is up to Security, Auditing or management anyway

Pervasive Encryption and IMS OSAM Datasets

Pros & Cons

CON – OSAM access to a VSAM LDS use slightly more CPU than native OSAM

PRO – OSAM access to a VSAM LDS allows dataset level encryption

CON – OSAM data buffers in memory are in the clear

PRO – OSAM access to a VSAM LDS may still us Sequential Buffering

PRO – OSAM access to a VSAM LDS may us HPF and Hiperwrite

PRO – Use of VSAM LDS outperforms VSAM KSDS / ESDS

CON – SAF Key authority may require further administration

PRO – Application independent

THANK YOU