Traditional z/TPF Database (Find and File) Encryption

Database Subcommittee

Michael Shershin

2024 TPF Users Group Conference
May 5-8, New Orleans, LA

IBM Z

IBM

Disclaimer

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Problem Statement

Requirements for sensitive information like PCI (payment card industry), PII (personal identifiable information) and GDPR (general data protection regulation in the European Union) are becoming more stringent. Sensitive information in DASD records (data at rest) needs to be encrypted to protect the information.

As-Is

Sensitive information in z/TPFDF files can be encrypted by z/TPFDF support without any application changes.

- APARs PI56476 and PJ43935 provided this support in 2016.
- One key aspect of this support is the ability to save information (decryption key name) in the physical record that is used to decrypt the data. z/TPFDF can save this information because it controls the data in the physical record.

Sensitive information can also be kept in traditional z/TPF databases.

- Physical records in traditional z/TPF databases are controlled by users.
- z/TPF secure key support allows applications to encrypt some or all of the data in a traditional z/TPF database record, but requires application changes to do so.

Pain Points

Traditional z/TPF databases

- In a physical record users control all data starting at location x’008’ to the end of the record.
- The z/TPF operating system has no place to save the decryption key name in the record to allow the record to be decrypted.
  - One possibility is to have the decryption key name in another associated record. However, this would require double the amount of I/O when processing encrypted records. Also, a significant amount of additional file space would be needed.

To-Be

Provide an option for the z/TPF operating system to encrypt traditional z/TPF databases.

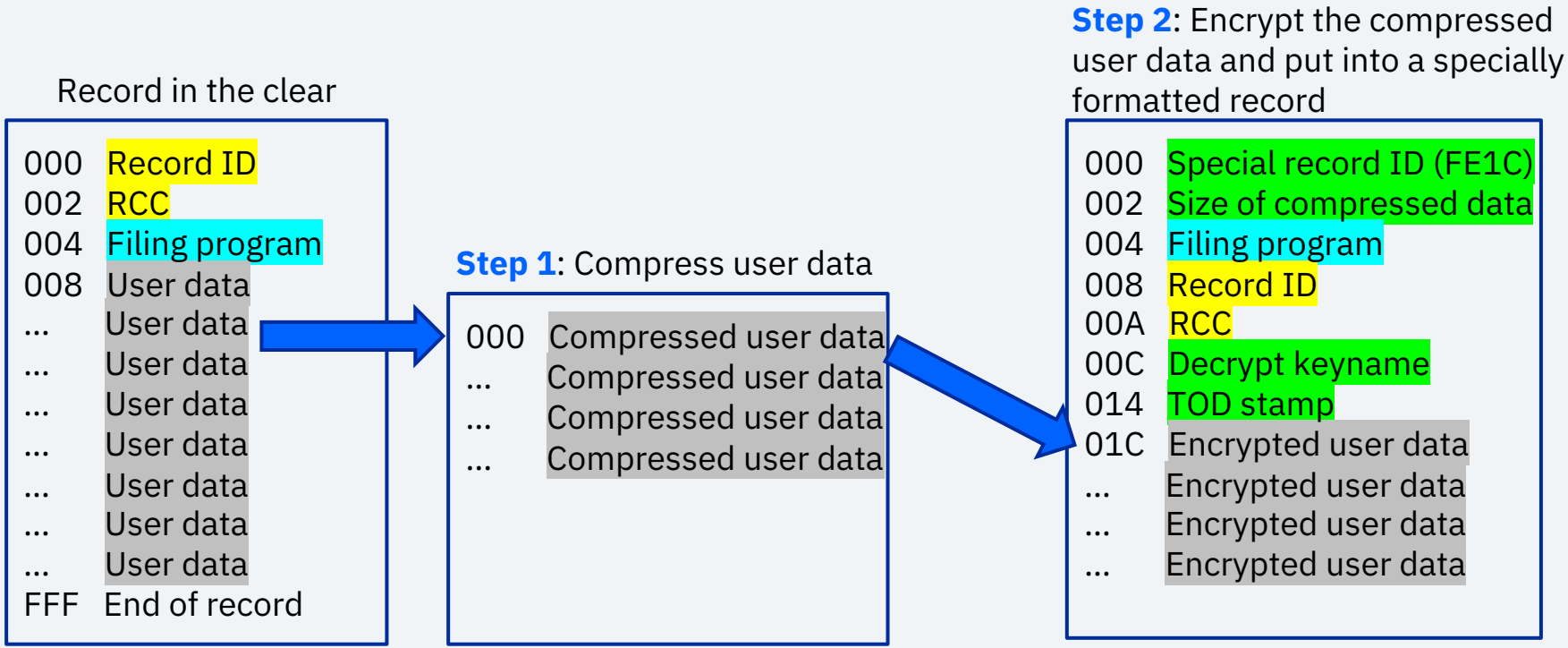
- No application changes will be required.
- No additional I/O will be required.
- Intended purpose is for user data. It is not intended for systems data.

The key is to free space in the physical record so that information to decrypt the record can be saved. Two methods to free space will be supported.

- Hardware compression will be used to compress user data in records. The compressed user data will be encrypted.
- If 20+ bytes at the end of the record are zeros, encrypt the user data up to the zeros.

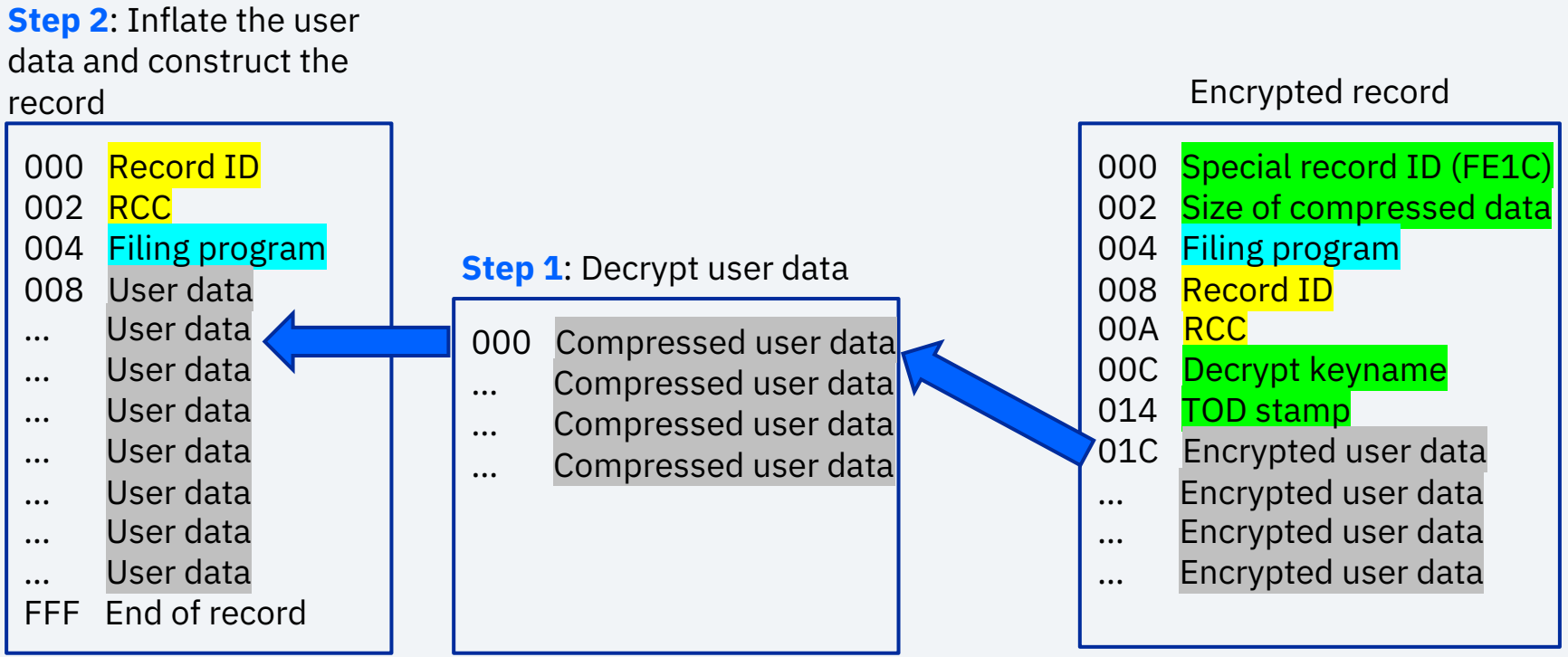
Technical Details

Method 1 – On a **file** compress the user data and then encrypt the compressed data



Technical Details

Method 1 – On a **find** decrypt the compressed data and then inflate the user data



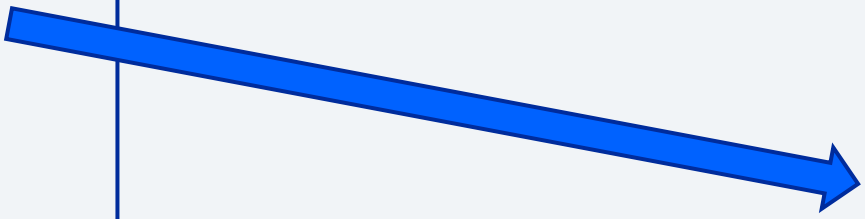
Technical Details

Method 2 – On a **file** if the bottom of the record contains zeros, encrypt the user data

Step 1: Encrypt the user data, put into a specially formatted record, and set size of compressed data to zero.

Record in the clear

000	Record ID
002	RCC
004	Filing program
008	User data
...	User data
...	User data
...	User data
...	User data
FE8	Zeros
FF0	Zeros
FF8	Zeros
FFF	End of record



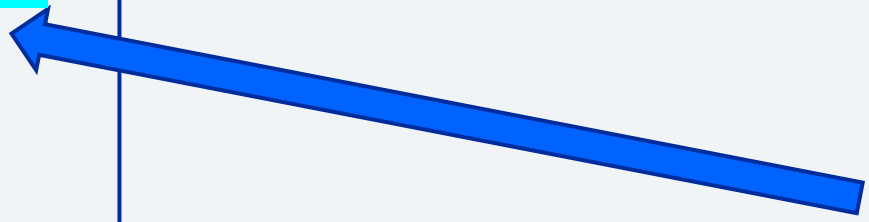
000	Special record ID (FE1C)
002	0000
004	Filing program
008	Record ID
00A	RCC
00C	Decrypt keyname
014	TOD stamp
01C	Encrypted user data
...	Encrypted user data
...	Encrypted user data
...	Encrypted user data
...	Encrypted user data

Technical Details

Method 2 – On a **find** if the size of the compressed data is zero, decrypt the user data

Step 1: Decrypt the user data, construct the record, and put zeros at end of the record

000	Record ID
002	RCC
004	Filing program
008	User data
...	User data
...	User data
...	User data
...	User data
FE8	Zeros
FF0	Zeros
FF8	Zeros
FFF	End of record



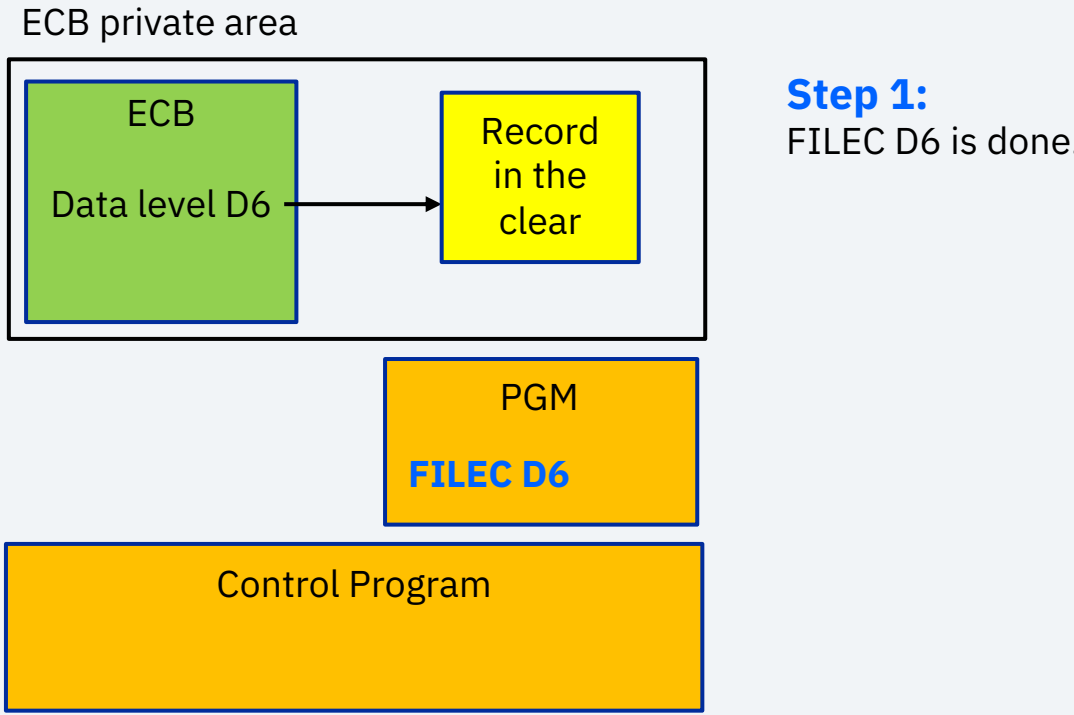
Encrypted record

000	Special record ID (FE1C)
002	0000
004	Filing program
008	Record ID
00A	RCC
00C	Decrypt keyname
014	TOD stamp
01C	Encrypted user data
...	Encrypted user data
...	Encrypted user data
...	Encrypted user data
...	Encrypted user data

Technical Details

FILE

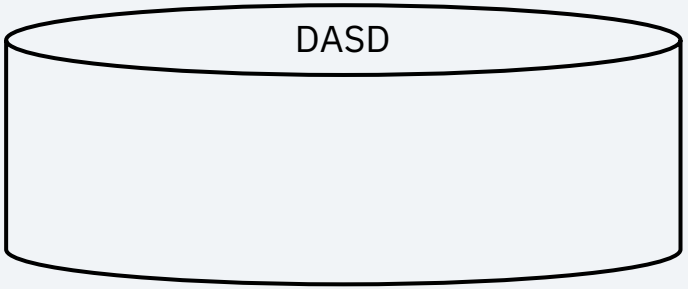
An encrypted record that is a VFA candidate



Step 1:
FILEC D6 is done.

Logging tape

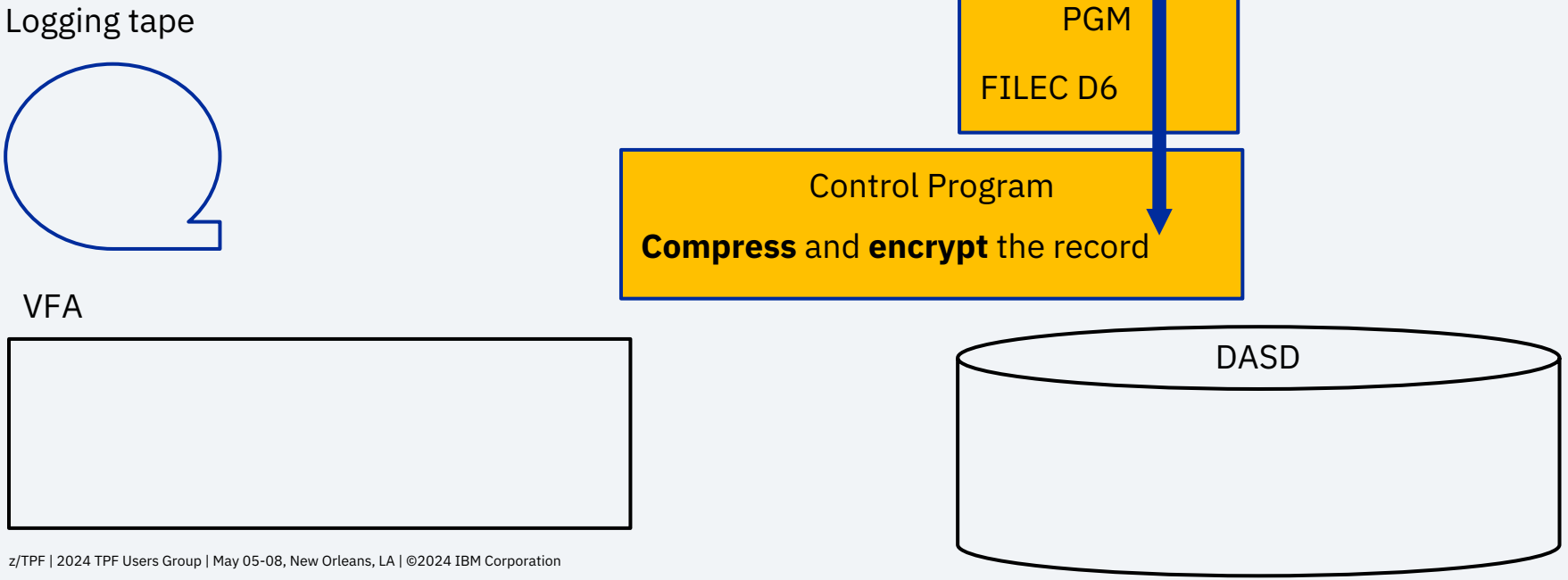
VFA



Technical Details

FILE

An encrypted record that is a VFA candidate

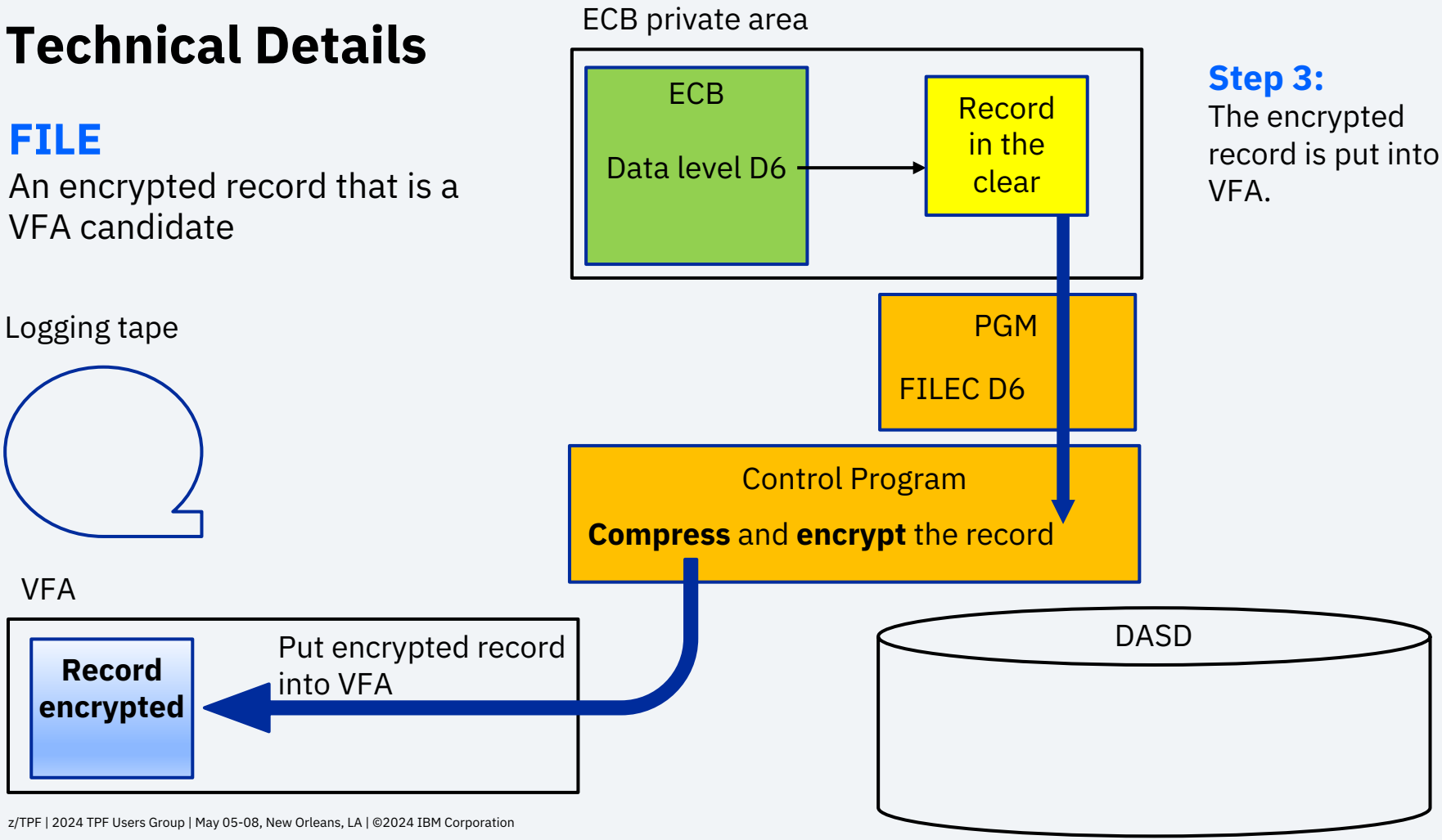


Step 2:
The record is compressed and encrypted in the control program.

Technical Details

FILE

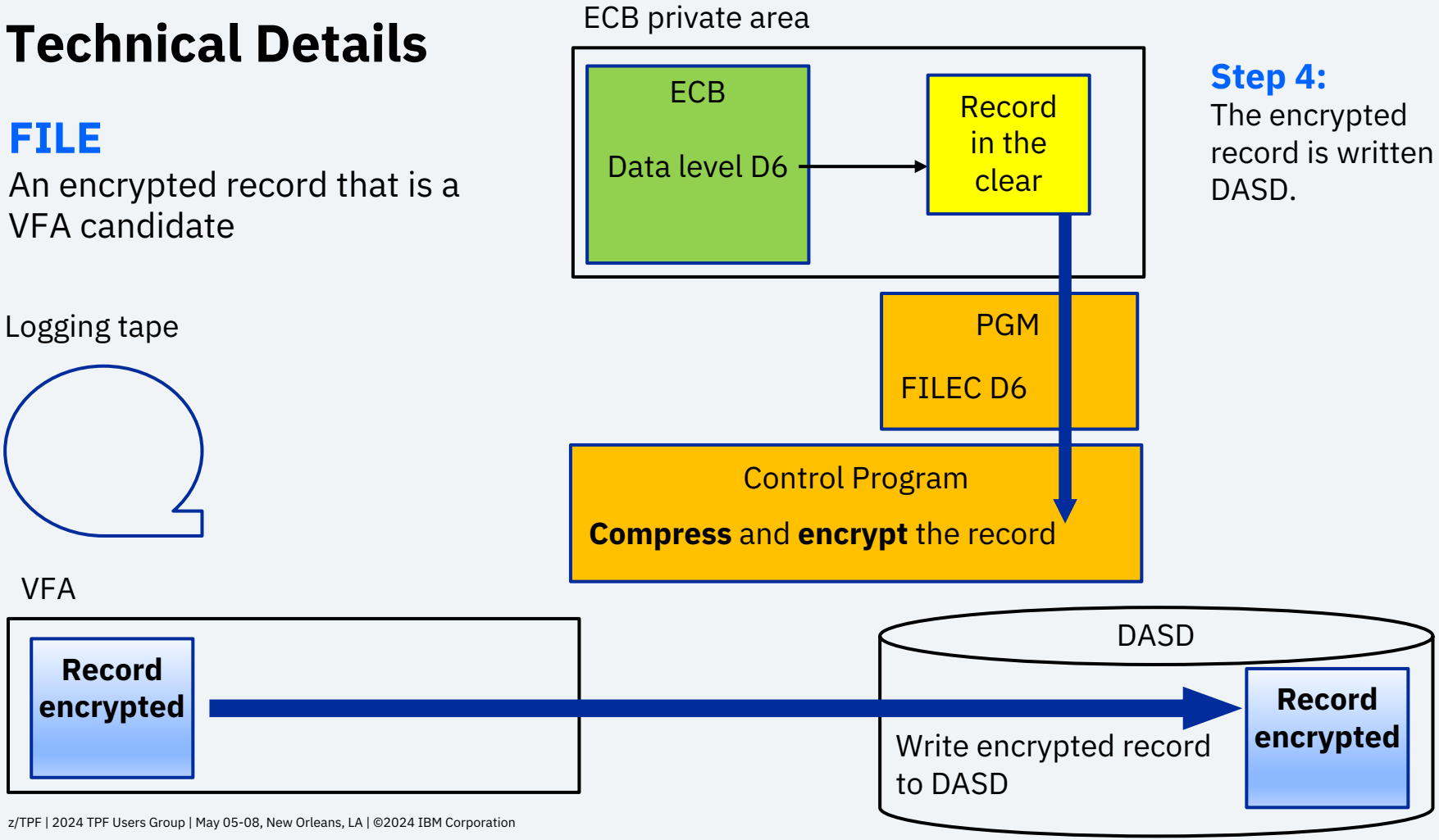
An encrypted record that is a VFA candidate



Technical Details

FILE

An encrypted record that is a VFA candidate

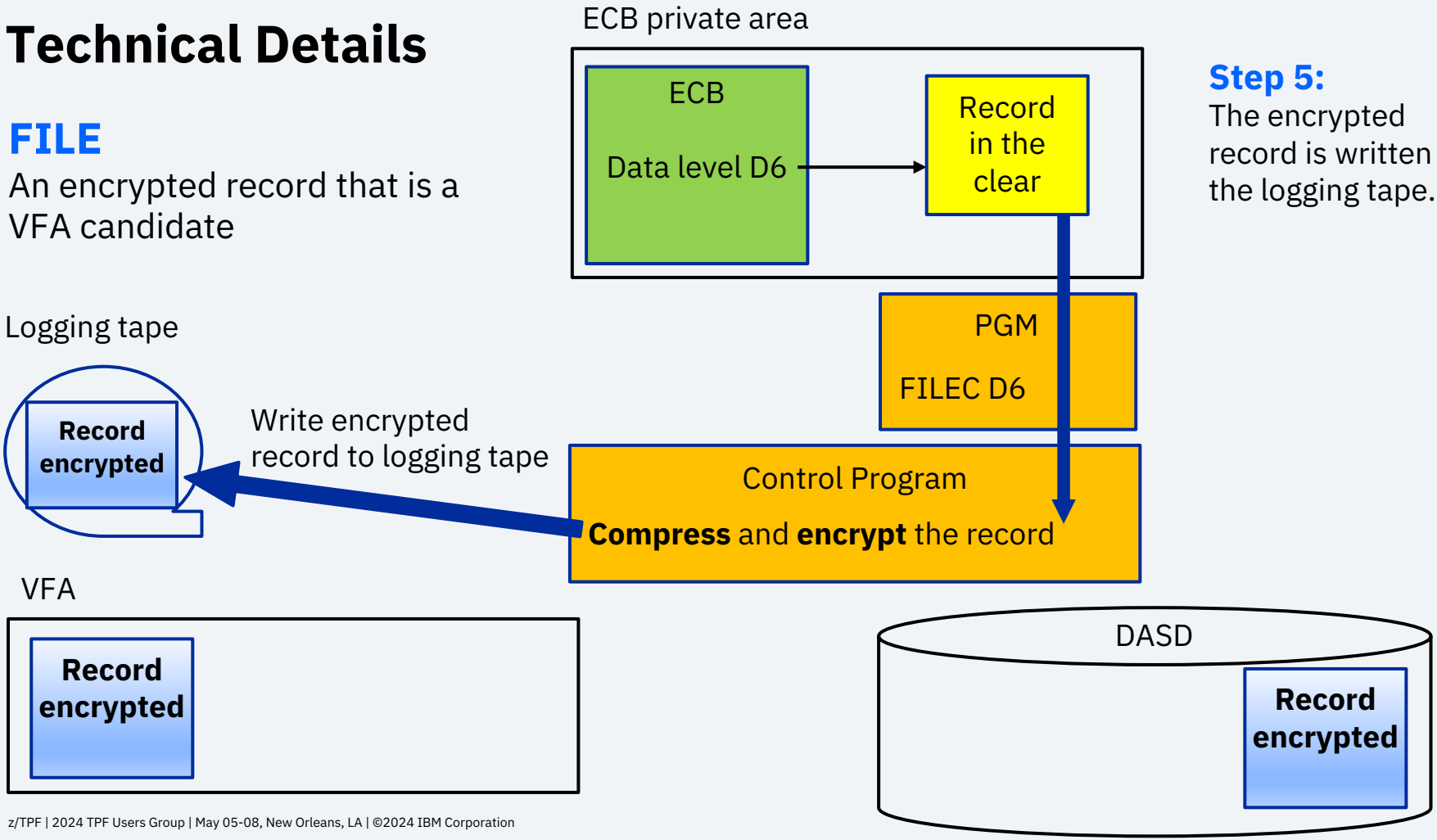


Step 4:
The encrypted record is written to DASD.

Technical Details

FILE

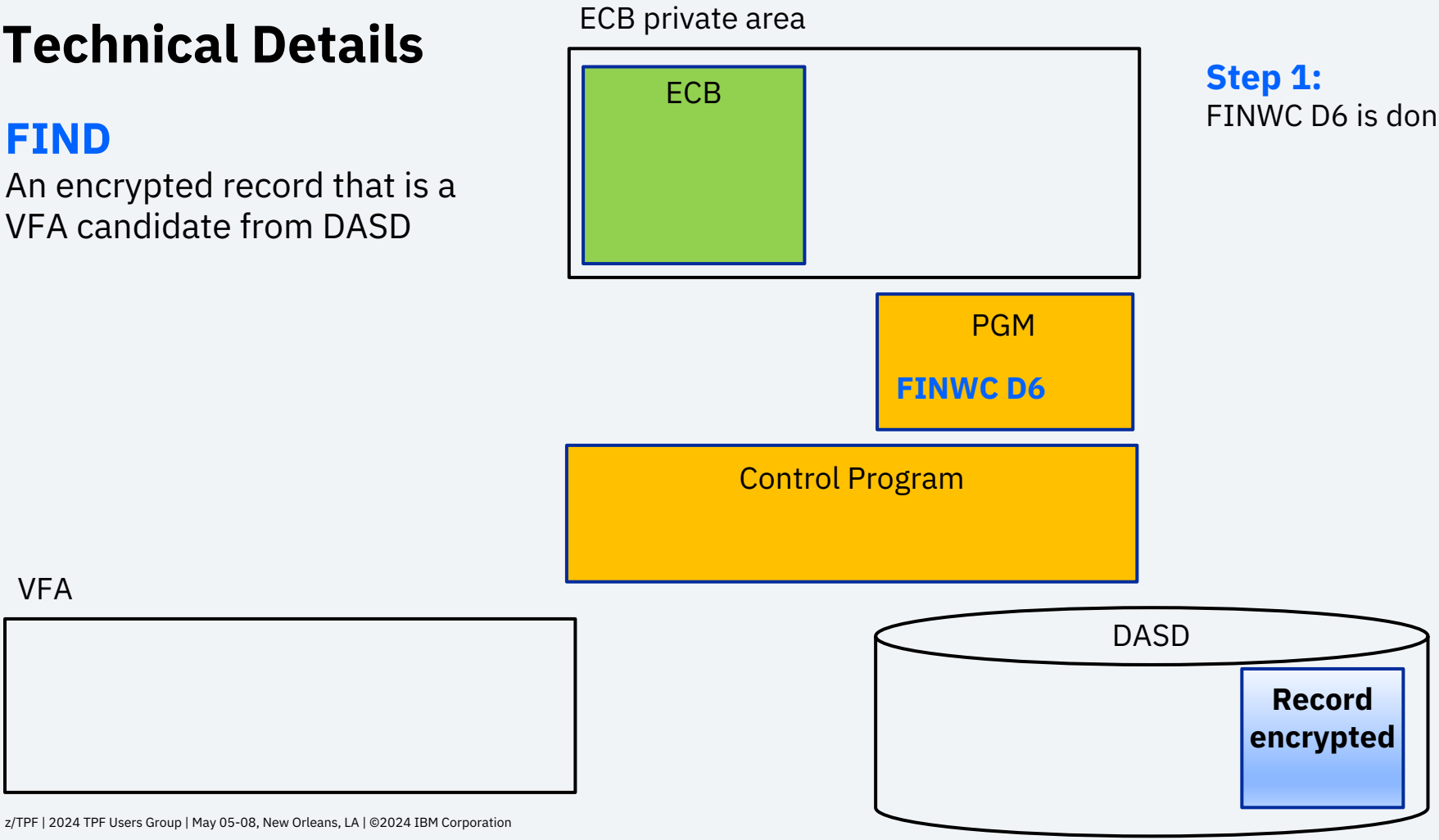
An encrypted record that is a VFA candidate



Technical Details

FIND

An encrypted record that is a VFA candidate from DASD

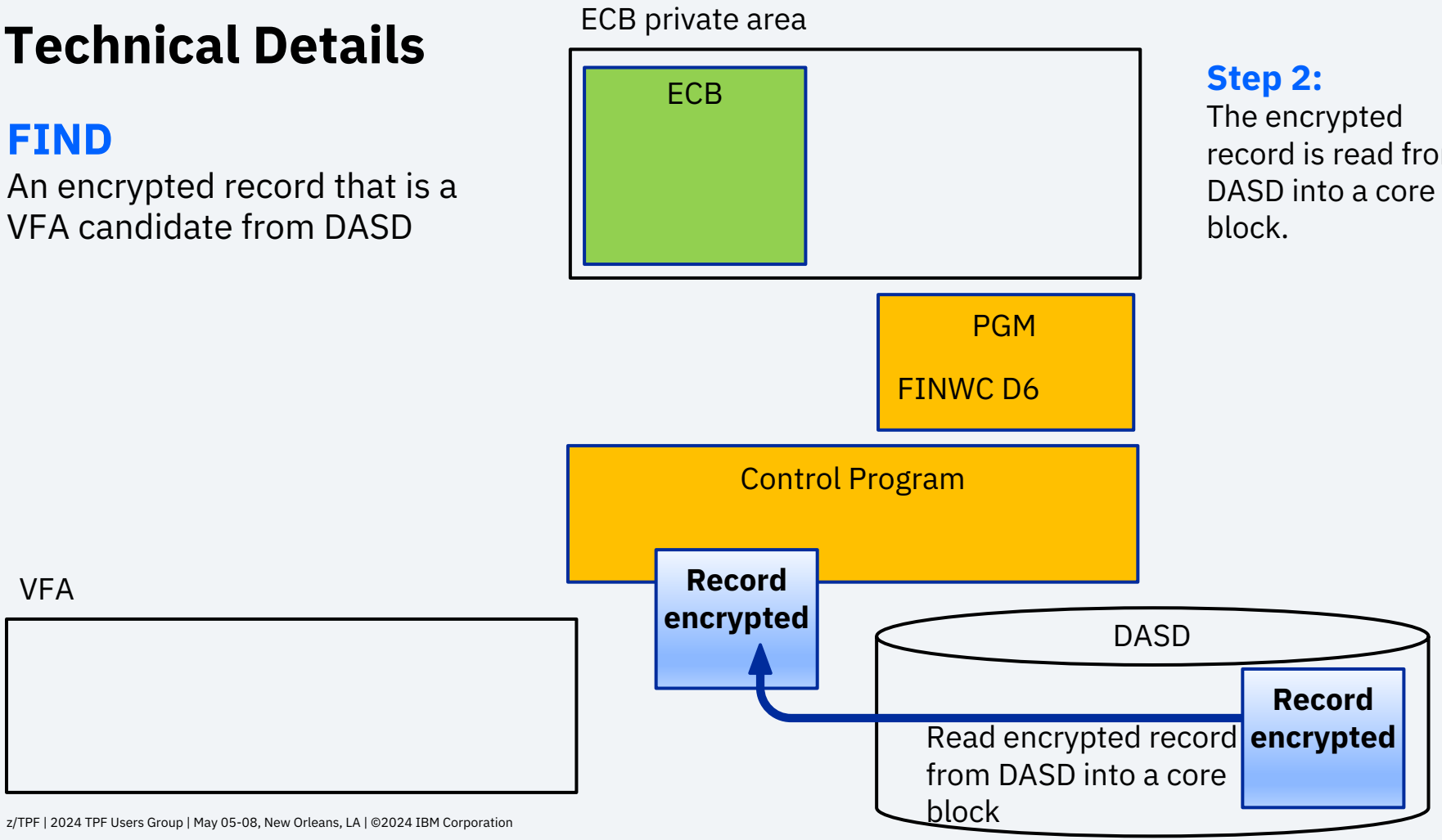


Step 1:
FINWC D6 is done.

Technical Details

FIND

An encrypted record that is a VFA candidate from DASD

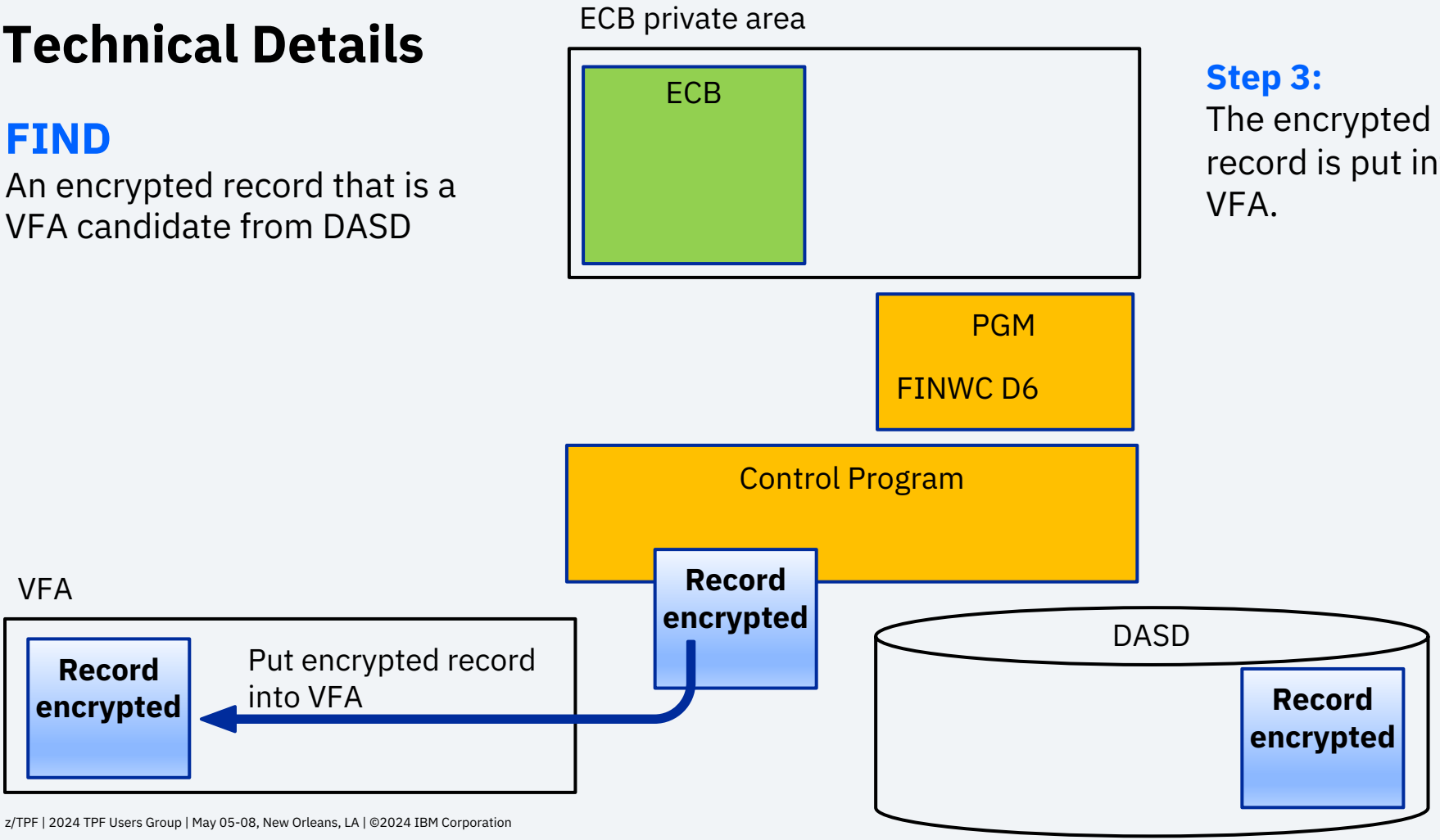


Step 2:
The encrypted record is read from DASD into a core block.

Technical Details

FIND

An encrypted record that is a VFA candidate from DASD

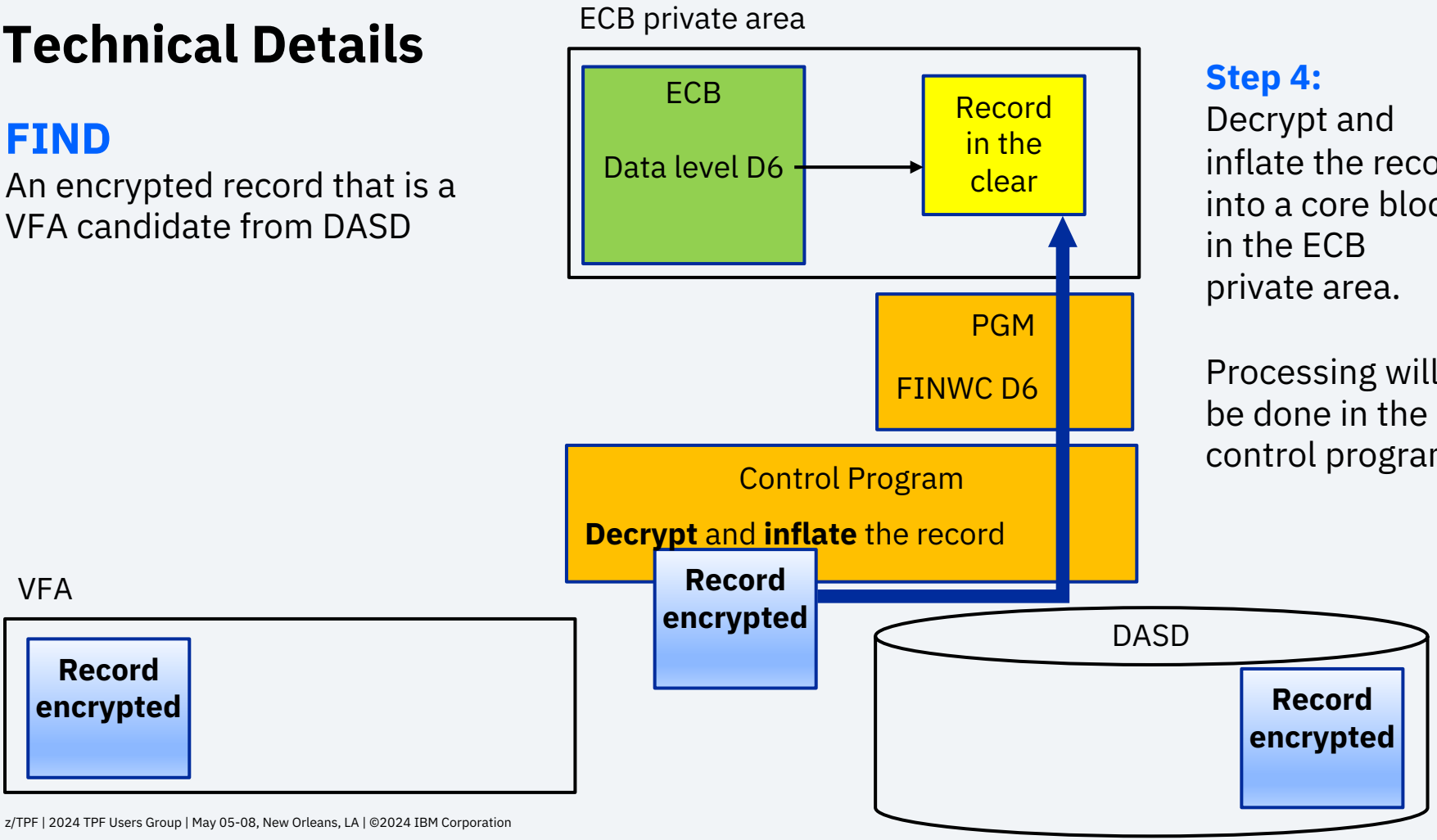


Step 3:
The encrypted record is put into VFA.

Technical Details

FIND

An encrypted record that is a VFA candidate from DASD



Step 4:
Decrypt and inflate the record into a core block in the ECB private area.

Processing will be done in the control program.

Technical Details

Characteristics

- IBM z15 or later processors will be required for traditional z/TPF database encryption.
  - Hardware compression is required.
  - Data can be encrypted using quantum safe cipher AES-256
- Records will be encrypted in VFA.
- Records will be written to logging and exception recording tapes as encrypted.

Technical Details

Characteristics (continued)

- Encryption will be supported for all record sizes: 381, 1055, and 4 KB.
- Encryption will be done when the following file macros are called:
  - FILEC, FILNC, FILUC, FILSC, and OFLNC
- Decryption will be done when the following find macros are called:
  - FINDC, FINHC, FINWC, FIWHC, FINSC, and FINRC
- Encryption and decryption will not be done when the FDCTC macro is called.
  - Not an issue for application access to traditional z/TPF databases.
- Encryption and decryption will not be done for GDS or VSAM records.

Technical Details

ZDFIL and ZDREC characteristics

- For debugging application problems, default displays of encrypted records will be decrypted before displaying the data.
  - The following commands will display the data in the clear:
    - ➔ ZDFIL 00000000F8428000 0.FFF
    - ➔ ZDREC LPSTV1.0 0.FFF

- For debugging system issues, a new option will be added to ZDFIL and ZDREC to display the raw data.
  - The following commands will display the encrypted data:
    - ➔ ZDFIL 00000000F8428000 0.FFF **DECRYPT-NO**
    - ➔ ZDREC LPSTV1.0 0.FFF **DECRYPT-NO**

Technical Details

Added overhead

- When a record is filed and the record is encrypted or compressed and encrypted, additional path length is needed.
  - Likewise, when a record is found and the record is decrypted or decrypted and inflated, additional path length is needed.
- The following support will be used to help mitigate the cost.
  - Hardware encryption and compression will be used.
  - Similar to z/TPFDF database encryption support, the code to encrypt and decrypt (and to compress and inflate) traditional z/TPF databases will be TE eligible.

- Impact of the overhead depends on the databases that are encrypted.
  - Frequency that records are filed and found is key to factor in determining impact.

Technical Details

Control

- The record ID will be used to determine whether a record will be encrypted.
- The RIAT (record ID attribute table) entry for a specific record ID will contain an encryption key name that indicates whether the record should be encrypted.
  - The ZKEYS GENERATE command is used to create a new key.
  - The ZKEYS ACTIVATE command is used to activate a new key.
  - The ZRTDM MODIFY command will be used to assign an encryption key name to a record ID.
    - Or a RIAT can be loaded with the image loader (TLDR).
- Supported ciphers will be AES-128-CBC and AES-256-CBC.
- If a record ID entry in the RIAT has an encryption key name defined, all records that are filed using the record ID will be encrypted.

Example of ZRTDM MODIFY to set the encryption key name

```
==> ZRTDM MODIFY RECID-FC39,KEYNAME-MYENAME1
CSMP0097I 11.02.59 CPU-B SS-BSS  SSU-HPN  IS-01
RTDM0014I 11.02.59 RIAT ENTRY DISPLAY FOR RECORD ID FC39
                   VFAF-NO     , LOCKF-DASD, RCSF-RET , SKIPMFX-NO
                   VFAP-NO     , LOCKP-DASD, RCSP-RET , SKIPMST-NO
                   LOG-NO , XCP-YES, RESTORE-YES, UEXIT-NO , PDELAY-  0
                   RTP0-NIU , RTP1-NIU , RTP2-NIU , RTP3-NIU ,  RTP4-NIU
                   RTP5-NIU , RTP6-NIU , RTP7-NIU , RTP8-NIU ,  RTP9-NIU
                   KEYNAME-*NOTENC*

RTDM0014I 11.02.59 RIAT ENTRY DISPLAY FOR RECORD ID FC39
                   VFAF-NO     , LOCKF-DASD, RCSF-RET , SKIPMFX-NO
                   VFAP-NO     , LOCKP-DASD, RCSP-RET , SKIPMST-NO
                   LOG-NO , XCP-YES, RESTORE-YES, UEXIT-NO , PDELAY-  0
                   RTP0-NIU , RTP1-NIU , RTP2-NIU , RTP3-NIU ,  RTP4-NIU
                   RTP5-NIU , RTP6-NIU , RTP7-NIU , RTP8-NIU ,  RTP9-NIU
                   KEYNAME-MYENAME1

RTDM0004I 11.02.59 MODIFY  PROCESSING COMPLETED FOR IMAGE TPF02   +
```

Technical Details

Migration to use encryption

- Define the keys.
  - ZKEYS GENERATE
  - ZKEYS ACTIVATE
- For the record ID that will be encrypted, update the RIAT with the encryption. key name.
  - ZRTDM MODIFY
- All records that are filed with the record ID will be encrypted.
  - Filing of records in traditional z/TPF databases will happen naturally by transactional work or by application utilities.
  - There is no system utility that files records in traditional z/TPF databases.
    - The system does not know the locking protocol that is used for the records.

Technical Details

Recoup updates

- Provide support for you to understand the percentage of records that are encrypted by record ID.
  - ✓ Recoup chain chase will collect counts of encrypted physical records by record ID.
  - ✓ Displays will be provided to see the counts of encrypted records by record ID that chain chase processing found.

Technical Details

Recoup updates continued

- There is an expectation that keys are rotated on a periodic basis.
  - The key can be changed using one of the following methods:
    - Create a new key by using the ZKEYS command and update the RIAT entries by using the ZRTDM MODIFY command.
    - Define a new key by using the ZKEYS command with the same encryption key name but with a different decryption key name. No RIAT change is needed.
  - Records will be encrypted by using the new key when they are filed.
  - Some currently encrypted records might not be re-encrypted because they are not updated. They will have an old decryption key name in the record.
  - Need to identify encrypted records that are using an old decryption key name.
    - Recoup will provide an option to log file addresses that are using an old decryption key name. Similar to the REFFROM option.
  - Need to know when the old key is no longer used so that it can be safely deleted.

Technical Details

Recoup updates continued

- Recoup will NOT update any user records to use encryption.
  - Physical records will be encrypted when the RIAT entry for the record ID has an encryption key name and the physical record is filed.
  - When a physical record is filed, there is a locking protocol that is used to serialize updates.
  - Recoup does not know the locking protocol that is used when updates are made to a physical record.
    - Recoup will NOT update user records.
  - Updating physical records to use encryption or a new key is the responsibility of the user.

Technical Details

Data collection

- Collect and report additional data including:
  - Rate of files of physical records that are encrypted.
  - Rate of files of physical records that are encrypted and compressed.
  - Rate of finds of physical records that are decrypted.
  - Rate of finds of physical records that are decrypted and inflated.

Value Statement

User data in traditional z/TPF database records can be encrypted without any application changes. Data will be encrypted in the following locations:

- On DASD
  - Production DASD and any copy of production (Disaster recovery system, point-in-time copies, test systems)
- In VFA
- In flight over channels
- On logging tapes

Conclusion

You will be able to protect sensitive data using encryption in traditional z/TPF databases without any application changes.

- You will be able to selectively choose which databases will be encrypted by record ID.
- You will be able to gather diagnostic information about encrypted records when recoup is run.
  - Get counts of encrypted records by record ID
  - Log information about encrypted file addresses for diagnostic purposes
- You will be able to observe the frequency that encryption is used from data collection.
- Requires IBM z15 or later processors to use this support

Target delivery date

The target delivery date is 4Q2024.

Be a sponsor user

Sponsor users assist in design and implementation, and your feedback drives our development cycle.

Target personas

- Application architects
- Systems developers
- Performance analysts

Begins

On going

Interested? Contact

Michael Shershin e-mail: shershin@us.ibm.com

Thank you