Traditional z/TPF Database Encryption

System Control Program

Tim Backus

2025 TPF Users Group Conference
May 04-07, Austin, TX

IBM Z

Executive Summary

Yumi
Chief Technology Officer

With traditional z/TPF database encryption, you can protect sensitive information at rest in z/TPF databases that are accessed by find and file APIs with no application changes and no downtime.

Problem Statement

Requirements for sensitive information are becoming more stringent.

- Sensitive information in DASD records (data at rest) needs to be encrypted to protect the information.
- New security standards consider data in a memory cache to be at rest; therefore, data in memory needs to be encrypted. Disk-level encryption alone is not sufficient.
- Although z/TPFDF encryption support was delivered years ago, that support does not include traditional z/TPF databases.

User Story

Andres
Database Admin

As part of a company-wide effort, Andres is tasked with encrypting records that hold customer-specific data that resides in a traditional z/TPF database.

The records in question hold sensitive client information in the form of PNRs.

User Story

Andres
Database Admin

Because PJ47147 is available, Andres can use this support to encrypt the records that hold customer PNR data.

No application changes were necessary.

ALTERNATE REALITY

Andres
Database Admin

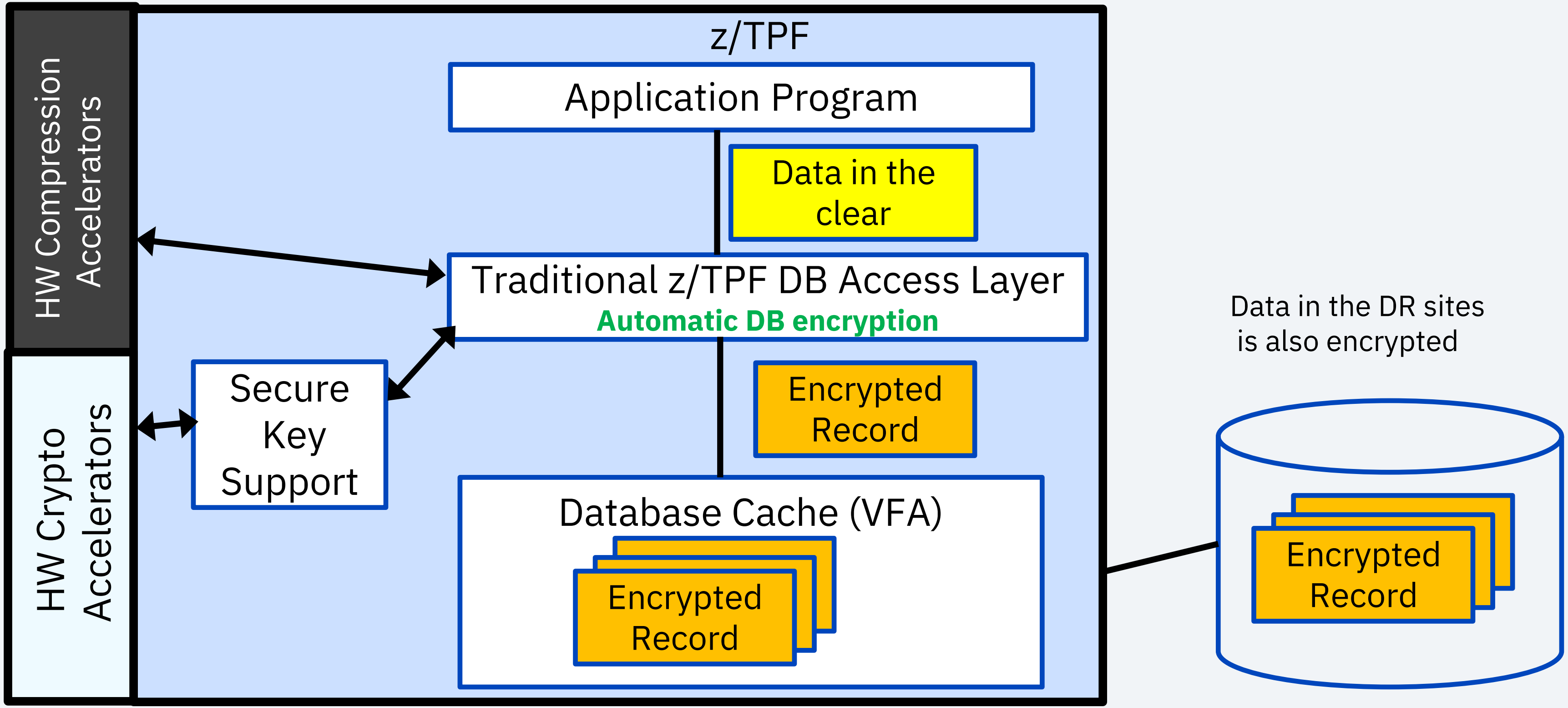
Andres must implement and test an in-house encryption system for his company's traditional z/TPF databases.

He is showing signs of severe mental anguish.

Overview

- Traditional z/TPF database encryption requires hardware compression, so a z15 or later is required.
- Keys that are used for encrypting and decrypting data are created and managed using the z/TPF Secure Keystore.
- **No changes are required to applications** because they still interact with with data in the clear.

Overview



Data in the DR sites is also encrypted

Overview

- Use the ZRTDM MODIFY command to enable encryption for a specific record ID.
- Supported ciphers are AES-128-CBC and AES-256-CBC.
- AES-256-CBC is quantum-safe.

Technical Details – Encryption Method #1

When a record is filed, if the last 20 bytes are zero, the record is encrypted.

- The last 20 bytes are not encrypted because they contain the decryption key.
- The entire record is written to DASD. This includes the encrypted portion and the 20 control bytes at the end.

When a record is read, it is decrypted, and the last 20 bytes are set to zero.

Technical Details – Encryption Method #2

When a record is filed, if the last 20 bytes are **not** zero, the record is compressed before it is encrypted.

- If compression reduces the size of the record by at least 20 bytes, the record is encrypted.
- In the very unlikely event compressing the data does not reduce the data size by at least 20 bytes, the record is not encrypted.

When a record is read, it is decompressed and decrypted if it was previously encrypted.

Technical Details – Operations

- Encryption occurs when a FILEC, FILNC, FILUC, FILSC, or OFLNC macro is called.
- Decryption occurs when a FINDC, FINHC, FINWC, FIWHC, FINSC, or FINRC macro is called.
- Commands like ZDREC and ZDFIL show unencrypted data.

Technical Details – Scope

- Data is encrypted:
  - When it is at rest in VFA
  - On DASD devices
  - On logging and exception recording tapes
  - In flight over channels to DASD control units and tape control units
  - When copied locally or to other DASD control units for disaster recovery (DR) purposes

Technical Details – Control

- If a record ID entry in the RIAT has an encryption key name defined, all records that are filed by using the record ID will be encrypted.
- To set the encryption key in the RIAT, you can:
  - Use the ZRTDM MODIFY command.
  - Load a RIAT by using the image loader.

Technical Details – Control

- You can change the encryption key and even the encryption algorithm while the database is being used.
  - Updating the key or algorithm does not require database downtime.
- Use the ZDFEC command to show encryption and compression statistics of traditional z/TPF databases.
- Recoup can be used to understand the number of encrypted records when the support is used.

Value Statement

Traditional z/TPF database encryption provides support for industry-standard encryption of your traditional z/TPF databases without changes to the application.

Data in VFA, DASD, and logging and exception tapes is encrypted. Data in flight to control units is also encrypted.

Conclusion

APAR PJ47147 (September 2024) delivered support for traditional z/TPF database encryption.

More detailed information can be found in the 2024 TPFUG presentation for this support, found here.

Thank you

IBM