ORACLE

Oracle Transparent Data Encryption (TDE) and General-Purpose Deduplication Technology

Technical Brief

August 2023, Version 1.0

Public

Purpose Statement

This technical brief explores the effect of the Oracle Transparent Data Encryption (TDE) on generic deduplication technology and solutions offered by the Zero Data Loss Recovery Appliance (ZDLRA) and Oracle Database Autonomous Recovery Service (ZRCV).

Disclaimer

This document, in any form, software, or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement, nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of Contents

List of images

Introduction

Modern-day cyber resiliency and ransomware protection solutions have dominated IT industry in the last several years. As a mitigating factor, companies are increasingly moving to end-to-end unified encryption practices spanning from production to backup and focusing on securing data at the *application level*.

There are three different ways to encrypt data for a database:

- File system encryption – protects against theft of the storage media, no protection against theft of the database files during a ransomware attack
- Data encryption (usually at the application level) – good security, but expensive to implement and has a significant impact on performance
- Database encryption (transparent data encryption) – data is encrypted at the database level so that if ransomware scrapes the database files, the information they get is not usable. But data is automatically decrypted for the application so there are no expensive application changes required. Performance impact is low, and so is the maintenance overhead.

For Oracle databases, Transparent Data Encryption (TDE) features:

- Data is encrypted at the database level, rendering stolen data unusable
- Data is automatically decrypted for the application, so no application changes are required
- Database workload optimized for low performance & maintenance overhead
- Encryption keys are only accessible by privileged databases, enforcing separation of duty between DBA and backup/storage teams.

While TDE is Oracle's best practice for data encryption, it's important to understand the effect of TDE databases on backup and secondary storage solutions which use deduplication technology.

Deduplication Fundamentals

Let's revisit the fundamentals of deduplication technology for a better understanding of why TDE has such a crucial impact:

There are two types of deduplication technology implementation available today: file/object-level and block-level. Both are based on "fingerprinting" concept to identify unique data in the data path to ensure that non-unique data in the system are not processed. Fingerprinting is fundamentally based on the "hash algorithms" such as MD5, SHA1, SHA256, etc. Any of these algorithms (or a combination of them) produces a unique identifier in the storage system that allows comparing client data to already stored data in the backend storage system to identify the net-new data vs. previously stored old data that were backed up during the past cycles. Fingerprints are generated on two levels for higher efficiency:

1. File or object level fingerprint to ensure that any files or objects in the source system that stayed unchanged will not be backed up. Typical scenario: a file name was changed, but not the content.
2. If a new file or an object is identified, then proceed to block-level fingerprinting to ensure that only changed blocks within a file or an object will be backed up. Typical scenario: Microsoft Exchange backup file upends all changed data to the tail of the backup image, leaving over 95% of backup data already backed up by the previous backup cycle.

With block-level fingerprinting algorithms, there are two unique approaches known as "fixed block" and "variable block" deduplication, which differ in the way a file or an object is parsed into block-level chunks. "Fixed block" means that a file or an object gets split based on the algorithm-specific (e.g. 4K, 8K, 64K, etc.) blocks from the first byte to the last. In certain cases, like Microsoft Exchange files or file systems, this approach is fairly efficient and generally imposes minimal overhead on the production system. Nevertheless, if any new data is inserted in the middle of a file or an object – it does not provide any way to identify the net-new data in the backup stream since the boundaries of the data blocks will shift.

In contrast, the "Variable block" algorithm analyzes the bitstream to identify changes and re-aligns the new data during processing as required, improving deduplication effectiveness against existing data in the storage system, but it comes with the additional cost on production CPU resources.

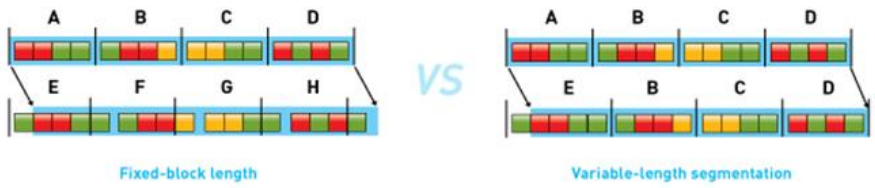

Figure 1: Comparison of Fixed & Variable-length Deduplication Algorithms

To complicate matters, ***ANY GLOBAL CHANGES*** in the source bitstream can affect the overall efficiency of the deduplication process, as both file-level and block-level fingerprinting essentially become useless. Such global operations that would reorganize or randomize the entire file or an object include ***COMPRESSION*** and ***ENCRYPTION***.

Storage and backup solution vendors generally advise customers ***NOT*** to enable Oracle Recovery Manager (RMAN) compression feature when backing up Oracle databases, as shown in the below example from a vendor's documentation:

Compression:

> *All supported versions of RMAN can apply a binary compression algorithm (BZIP2) to the backup set. This results in less disk space being used at the cost of significantly greater CPU consumption during the backup operation.* ***When using a <storage system> as the target, RMAN lossless compression should not be used, as pre-compression of the backup streams will randomize the data patterns and defeat deduplication.***

This effect of the global source bitstream change applies to ***ANY*** general-purpose deduplication platform as it is not aware of the source data changes and results in the deduplication process becoming unpredictable in analysis and fingerprinting.

Oracle TDE and Deduplication

Let's now review the specific effect of TDE on deduplication. Since encryption randomizes the initial bitstream (RMAN Level 0 or full backup), it results in a high probability that backup files become unique in the system, thereby defeating storage deduplication.

There are two ways to implement TDE encryption:

- RMAN encryption
- Oracle source database encryption

In the case of the RMAN encryption, the effect on generic deduplication will be identical to compression, as we already discussed in the "Deduplication Fundamentals" section. In the case of the source database-enabled TDE, RMAN will pass through the encrypted data, so it is essential to understand the full impact of enabling TDE-encrypted data at the source:

1. The initial RMAN L0 (full) backup deduplication ratio will be close to zero.
2. With TDE enabled, encryption also introduces ***SALT(*)*** data with subsequent backups, forcing any generic deduplication to re-read the entire file. While there might be some efficiency in deduplication cycles, the overall backup performance and storage efficiency will be significantly affected. SALT, which is a random string added to data before encryption, is a way to strengthen the security of encrypted data (see details below).
    a. Due to Oracle TDE backups resulting in higher generic dedupe backup storage consumption, customers often need to re-evaluate backup and storage investments.
    b. Due to the data file changes, most of the files will be identified as net-new and will be read by any deduplication client or the backend storage deduplication, imposing significant production overhead, especially on the client deduplication residing on the production server.
    c. Since general-purpose deduplication backup and storage products rely on RMAN backup data, 'incremental forever/virtual synthetic' strategies will also result in higher backup consumption, reducing their efficiency.
3. ***In addition to SALT(*), if a customer performs TDE key rotation, deduplication effectiveness will drop since the data file headers will change and data may be re-encrypted. This will force deduplication processes to re-read all source data to establish a new baseline for analysis and fingerprinting. All these data changes can result significant increase in production overhead and backup capacity consumption.***

(*) What is SALT:

1. Per ***Oracle Database 21c "Advanced Security Guide"*** Glossary: SALT is a way to strengthen the security of encrypted data. Salt is a random string that is added to the data before it is encrypted, making it more difficult for attackers to steal the data by matching patterns of ciphertext to known ciphertext samples. Salt is often also added to passwords, before the passwords are hashed, to avoid dictionary attacks, a method that attackers use to determine sensitive passwords. The addition of salt to a password before hashing makes it more difficult for intruders to match the hash values (sometimes called verifiers) with their dictionary list of common password hash values, because they do not know the salt beforehand.
2. Per ***Oracle Database 21c "Advanced Security Guide"*** section 5.4: "If you encrypt a table column without specifying an algorithm, then the column is encrypted using the AES192 algorithm. TDE adds salt to plaintext before encrypting it. Adding salt makes it harder for attackers to steal data through a brute force attack. TDE also adds a Message Authentication Code (MAC) to the data for integrity checking. The SHA-1 integrity algorithm is used by default. (Starting with Oracle Database release 21c, SHA-1 is deprecated. If you use TDE column encryption, then Oracle recommends that you implement TDE tablespace encryption instead.)"

Oracle Zero Data Loss Recovery Appliance (ZDLRA) and Zero Data Loss Autonomous Recovery Service (ZRCV) technology vs. General-Purpose Deduplication

With Oracle database-centric data protection technology, ***storage efficiency for database workloads is optimized by design*** compared to the general-purpose deduplication solutions. There are three major components in the solution:

- *Oracle Database Incremental-forever backup strategy with **Block Change Tracking (BCT)** speeds up backup performance by only reading from and writing out changed blocks to RMAN backups, even if the source database is TDE encrypted.*
- *Integration with the **Data Guard Real-Time Redo Transport** to ensure Zero Data Loss transaction-level database protection.*
- *Oracle proprietary backend **Virtual Full backup generation** from incremental backups, with an automated **Restore Validation** process to ensure recoverability.*

It allows ZDLRA/ZRCV to bring the following add-on value to storage optimization:

- RMAN uses the BCT feature to identify which blocks need to be backed up: RMAN reads only these blocks, including Transparent Data Encrypted (TDE) blocks. With Exadata Engineered Systems, this filtering

is performed on the storage cells to minimize the impact on compute servers and reduce data that needs to be transferred between compute and storage nodes.
    - In contrast, other storage deduplication platforms require the entire file to be read in order to achieve high storage savings.
- If additional L0 backups are performed, ZDLRA/ZRCV will ingest the backup and deduplicate any identical blocks, even if the source database is encrypted or encryption keys are rotated. This is accomplished by leveraging the unencrypted metadata in each block.
- Let's review an example of backing up a non-TDE encrypted 10TB size Oracle DB with 2% daily change rate:
    - Deduplication solutions rely on repetitive, full backups to effectively recognize duplicate data.
    - RMAN incremental backups with BCT enabled will read ***only 200GB*** of changed blocks from the disk and send to ZDLRA/ZRCV to produce virtual full backups, versus reading ***10TB*** for full backups to the deduplication storage platform.
- General-purpose deduplication solutions typically read Oracle DB files into the 128K-size buffer. Oracle Databases typically operate with 8K block size, and therefore, deduplication must read and write the entire 128K storage segment when only a few Oracle blocks changed (24K) within the segment, as shown in ***Figure 2*** below. In contrast, RMAN incremental backups identify changes at the 8K-size block level within the 128K-size buffer – a much more granular approach to identifying actual data change.

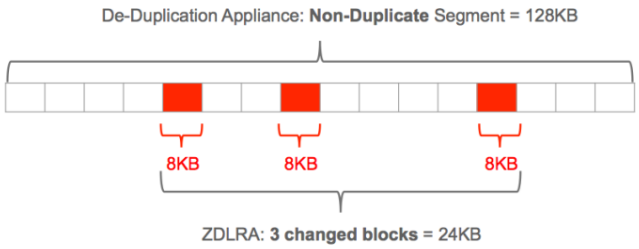

Figure 2: Deduplication vs. BCT

- ZDLRA/ZRCV works on the ***Incremental Forever*** model: it does not require traditional weekly full backups to maintain proper recovery SLAs by providing full Point-in-Time (PIT) verified backups without dependency on the Full (L0)+L1+archive logs approach.

Summary

1. ***Oracle Transparent Data Encryption (TDE) is the recommended encryption method for Oracle databases operating at the Oracle block level, minimizing performance impact, and does not require application changes.***
2. ***Turning on Oracle TDE encryption can reduce deduplication storage savings, potentially affecting all storage sizing and creating a***

problematic situation of the backup system running out of space while protecting business-critical applications.

3. *There is potentially more overhead introduced to the production system since the same source data – pre and post-TDE – may need to be re-read by the client-side deduplication process, thus increasing CPU and network consumption.*
4. ***Oracle's awareness of TDE data and block-level incremental forever backups provides end-to-end TDE encryption-based database security and highly efficient storage-saving Encrypted Backup technology.***

General-Purpose Deduplication Platform	Oracle Recovery Appliance and Autonomous Recovery Service
Generic storage deduplication must address any data source: file systems, apps, DBs, storage snaps, etc.	Oracle DB Block Change Tracking (BCT) provides superior performance and space efficiency for database backups.
Requires integration with the data source owners, e.g., Oracle, MSFT, NetApp, etc.	Purpose-built for protecting business-critical workloads, tightly integrated with Oracle DBs
Has to use traditional RMAN L0/L1 backups and archive log sweeps	Fast, Oracle block-level incremental backups with "virtual full" backend-created PIT recovery points for fast, verified recovery with Zero Data Loss
Low dedupe ratio: cannot deduplicate L0s against L1s or between archive logs especially if TDE is implemented	Oracle DB-centric storage optimization for TDE and non-TDE data, with predictable capacity calculations for sizing purposes.

Additional Resources

- Recovery Appliance Product Central
  - https://www.oracle.com/zdlra
- Technical Resources and Customer Stories
  - https://www.oracle.com/engineered-systems/zero-data-loss-recovery-appliance/technologies/
- AskTom Backup & Recovery Office Hours
  - https://asktom.oracle.com/pls/apex/asktom.search?office=1341
- Oracle MAA Blog
  - https://sites.oracle.com/site/maa/
- Defending and Recovering from Ransomware Attacks Tech Talk
  - https://go.oracle.com/LP=107483?elqCampaignId=288439#On-Demand-Webinars

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle