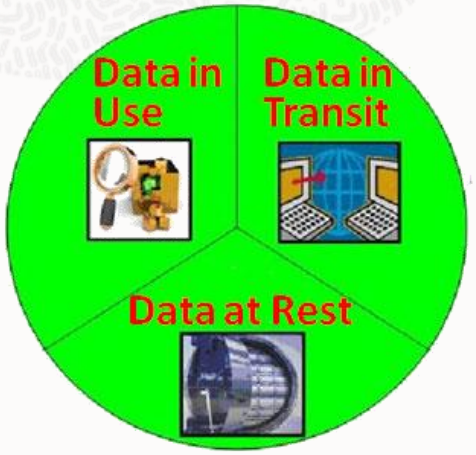
ORACLE

データベースの暗号化について

日本オラクル株式会社
大澤 清吾, DBSC 運営委員
2022年6月16日

「データセキュリティ」の考え方
～米国での取り組みと考え方

- 米国国防のセキュリティ基準を制定しているNSA(National Security Agency)では、2003年にNet-Centric Data Strategyという文書の中でデータの状態を3つに分類
  - **Data at Rest(保存状態)**
  - **Data in Transit(転送状態)**
  - **Data in Use(利用状態)**



NSAによるデータの3モード

- Data at RestとData in Transitの状態での「**暗号化**」を指示。
  => 認可で権限を持っていることを確認し、暗号鍵を使って暗号化されたデータを参照
- Data in Use(利用状態)については、「**最小権限（Least Privilege）**」と「Need to know」の原則との考えに乗っ取った対策を指示
  => 利用者が業務に必要となるデータのみを表示し、最小限の情報にとどめる
  例)「データベース管理者に対してもデータを見せない」、
  「一般利用者には役割に応じた必要な情報のみを提供する」

大統領令を発令、国家機関にサイバーセキュリティを向上する措置を指示

リアルタイムでデータ保護に集中するために、「ゼロトラストアーキテクチャ」の導入指示

データセントリックなセキュリティ・モデルの実装が必要とされています。

Sec.3. Modernizing Federal Government Cybersecurity.

60日以内に、各政府機関の長はゼロトラストアーキテクチャを実装するための計画を策定し、報告書を提出する

180日以内に、政府機関システムは多要素認証を採用し、保存中および転送中のデータに対しても暗号化を適用する

60日以内に、国家安全保証システムは大統領令のセキュリティ要件と同等またはそれ以上の要件を適用する

ゼロトラストアーキテクチャ

不正アクセスの拡大（ラテラル・ムーブメント）を制限し、異常又は不正な活動を探知し、「脅威が変化を続ける状況の中で、**リアルタイムでデータ保護に集中する**ために、インフラのすべての側面を通して、包括的なセキュリティ監視、きめ細かなリスクベースのアクセス制御、そして、システムセキュリティの自動化を組み合わせて導入する」ものです。

出典：Executive Order on Improving the Nation’s Cybersecurity　MAY 12, 2021　PRESIDENTIAL ACTIONS

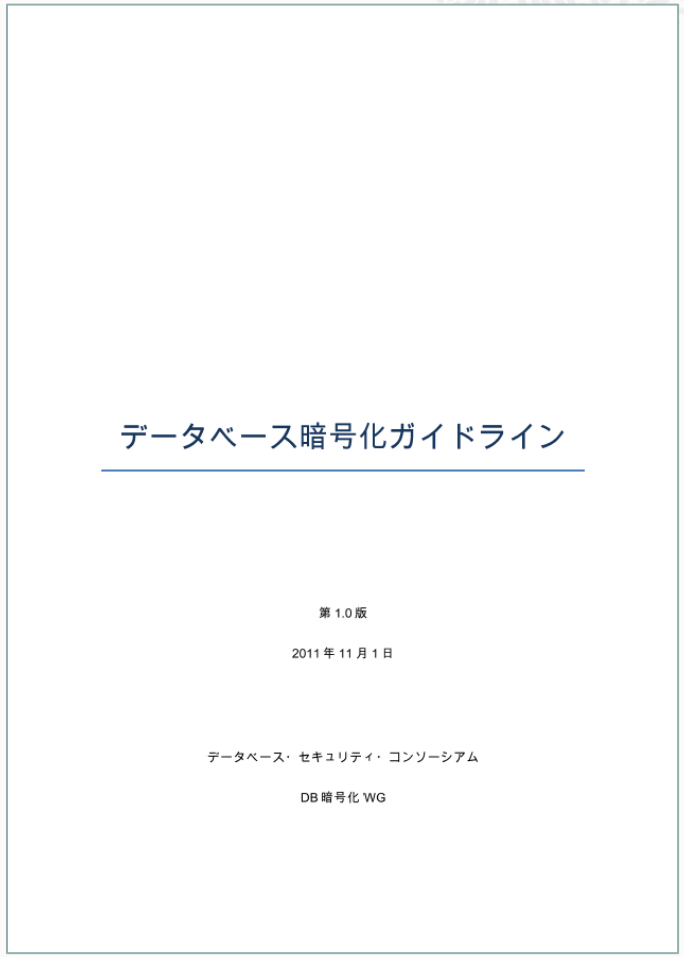
データベースの暗号化について

『データベース暗号化ガイドライン 第1.0版』

昨今のDBセキュリティにおいて、DB暗号化は情報漏えいに対する抜本的な対策であり、正しい運用を実装することで、リスクの低減もしくは漏えい防止を実現できるものである。

しかしDBの暗号化については、その手間や煩雑な運用イメージやパフォーマンスの低下、心理的不安要素などから敬遠されてきた。

本ガイドラインは、DB暗号化に対する正しい知識と運用方法を示し、各企業/団体の「DBセキュリティ対策」の導入の為の指標としてセキュリティ向上に貢献することを目的とするものである。



引用: http://www.db-security.org/report/cg_seika.html

データベースの暗号化対策概要

暗号化については、以下の3つの観点の対策が必要となる。

1. データベースの暗号化
2. 暗号鍵管理
3. 通信経路の暗号化

本セッションでは、Data at Rest(保存状態) を中心に説明を行う。

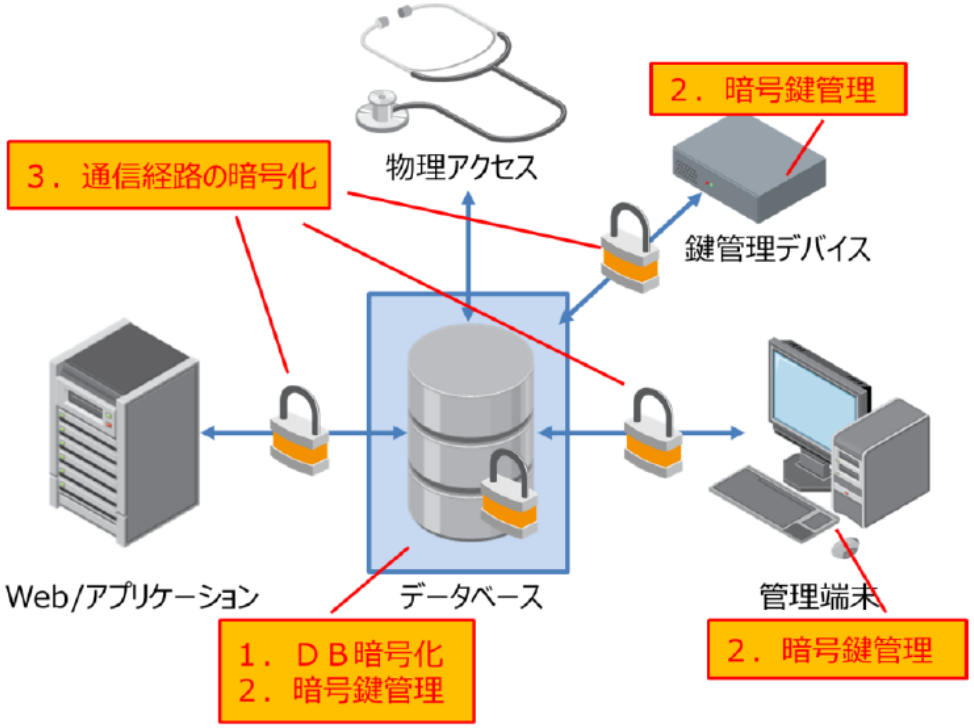


図2．暗号化対策概要

引用: http://www.db-security.org/report/cg_seika.html

DB暗号化の手法

HDDによる暗号化

- OS、ファイルシステム、ストレージ機器などが備えている暗号化機能を利用する方式
- OSからファイルアクセスがあり、ストレージシステムへ書き込まれるときに自動で暗号化され、ファイルシステムから読み込むときに透過的に暗号化されたデータが復号

DBテーブルによる暗号化

- DBMSが内部的にデータを暗号化して格納し、暗号化されたデータを透過的に復号してアプリケーションに戻す方式
- データベースには暗号化された状態でデータが格納され、該当テーブルに格納されたデータを参照する際にすべてのデータが復号

DBカラムによる暗号化

- DBMSが内部的にデータを暗号化して格納し、暗号化されたデータを透過的に復号してアプリケーションに戻す方式
- データベースには暗号化された状態でデータが格納され、暗号化されたカラムのデータを参照する際にデータが復号

アプリケーションによる暗号化

- 暗号化APIなどを利用して、アプリケーション側でデータを暗号化してから、データベースに格納する方式
- データベースには暗号化された状態でデータが格納され、データベースへの問い合わせで暗号化されたデータが戻り、復号はアプリケーション側で実施

暗号化方式による比較（攻撃に対する対策と有効性）

Oracle Databaseの暗号化技術での比較

攻撃内容	ストレージによる暗号化	DBテーブルの暗号化	DBカラムの暗号化	アプリケーションによる暗号化
ディスクの物理盗難	○	○	○	○
データファイル取得		○	○	○
バックアップファイル取得		○	○	○
エクスポートファイル取得		○	○	○
メモリダンプ取得		×	○	○
性能への影響	ディスクI/O頻度に応じて劣化	ディスクI/O頻度に応じて劣化	暗号化した列へのアクセス頻度に応じて劣化	暗号化した列へのアクセス頻度に応じて劣化
懸念事項	OSからは元のデータが見える		パフォーマンスの大幅な劣化 索引の利用に制限 データ量が暗号化対象の量に比例して増加	パフォーマンスの大幅な劣化 索引の利用に制限 データ量が暗号化対象の量に比例して増加
その他	OS経由でバックアップすると暗号化されない			アプリケーションの書き換えが必要

暗号鍵管理の方法

ファイルとしての暗号鍵管理

- データベース内ないし外部サーバでの鍵を管理
- 追加コストなく利用が可能
- 特権ユーザによる鍵のアクセスが容易で、また鍵のバックアップを行った際、そのバックアップメディアの管理についても検討が必要
- 鍵の更新などにおける世代管理についても十分検討されているか確認が必要

専用H/Wでの暗号鍵管理

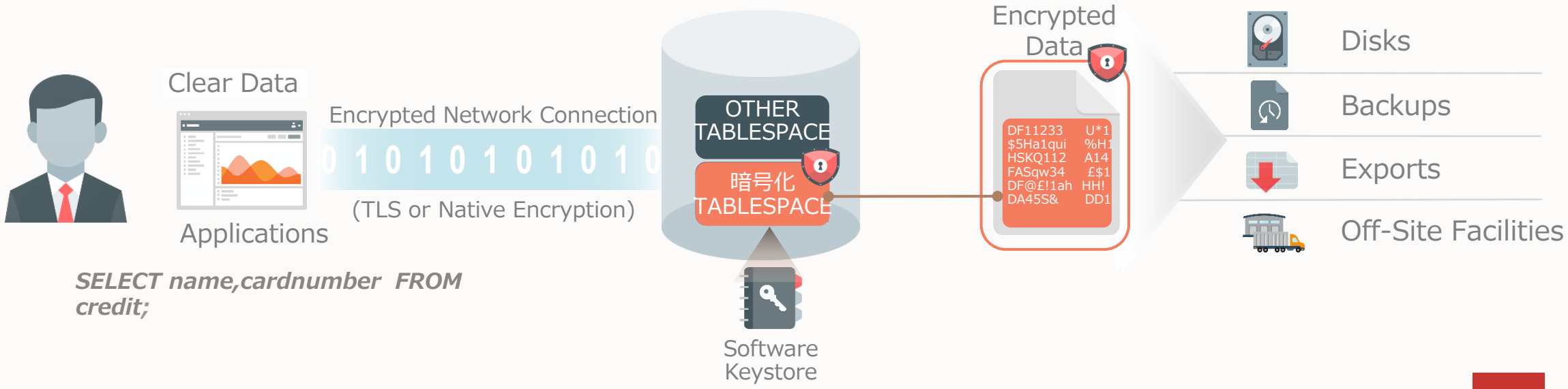
- 暗号鍵を管理する専用H/Wハードウェアセキュリティモジュール（HSM）を利用
- 導入コストが発生
- 鍵に対する耐タンパー性を持ち、暗号化のオフロードや鍵の世代管理、管理者の職務分掌機能などが利用できるため、管理が容易となる
- 導入時には効果を最大限発揮できるよう、しっかりとした内部統制に沿った設定および運用が必要である。

引用: http://www.db-security.org/report/cg_seika.html

オラクルでの実装方法について

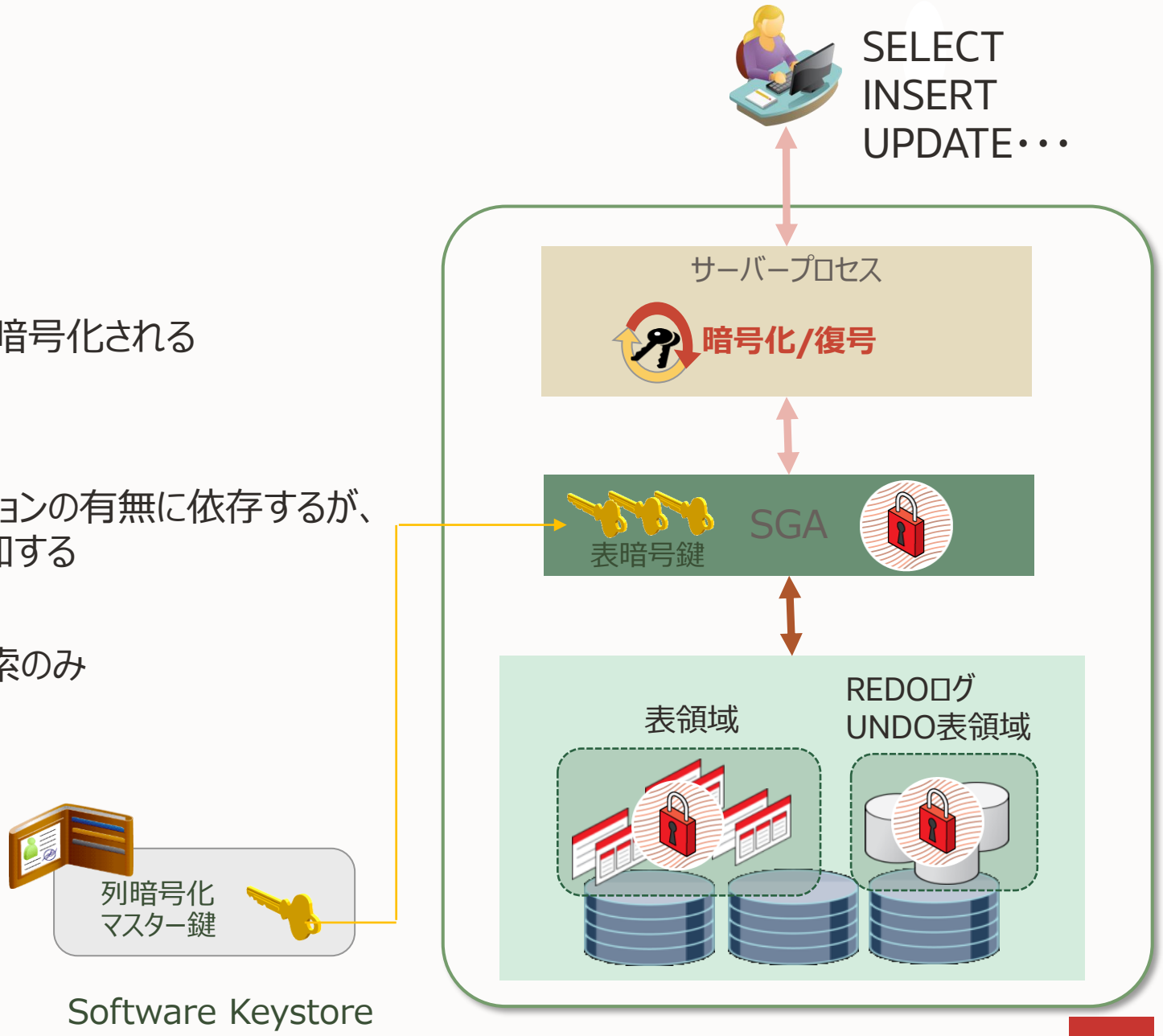
Oracle Database の暗号化技術：Transparent Data Encryption (TDE)

- アプリケーションからは透過的にデータの暗号化/復号
  - 既存のアプリケーション（SQL）を改修する必要はなし
- **列暗号**、**表領域暗号**の2種類の暗号化方式
- 強力な暗号アルゴリズムを利用した暗号化を実施
  - NISTの標準共通鍵暗号方式 AES(128/192/256bit) に対応
- Oracle Wallet や Hardware Security Module を利用した暗号鍵管理メカニズム



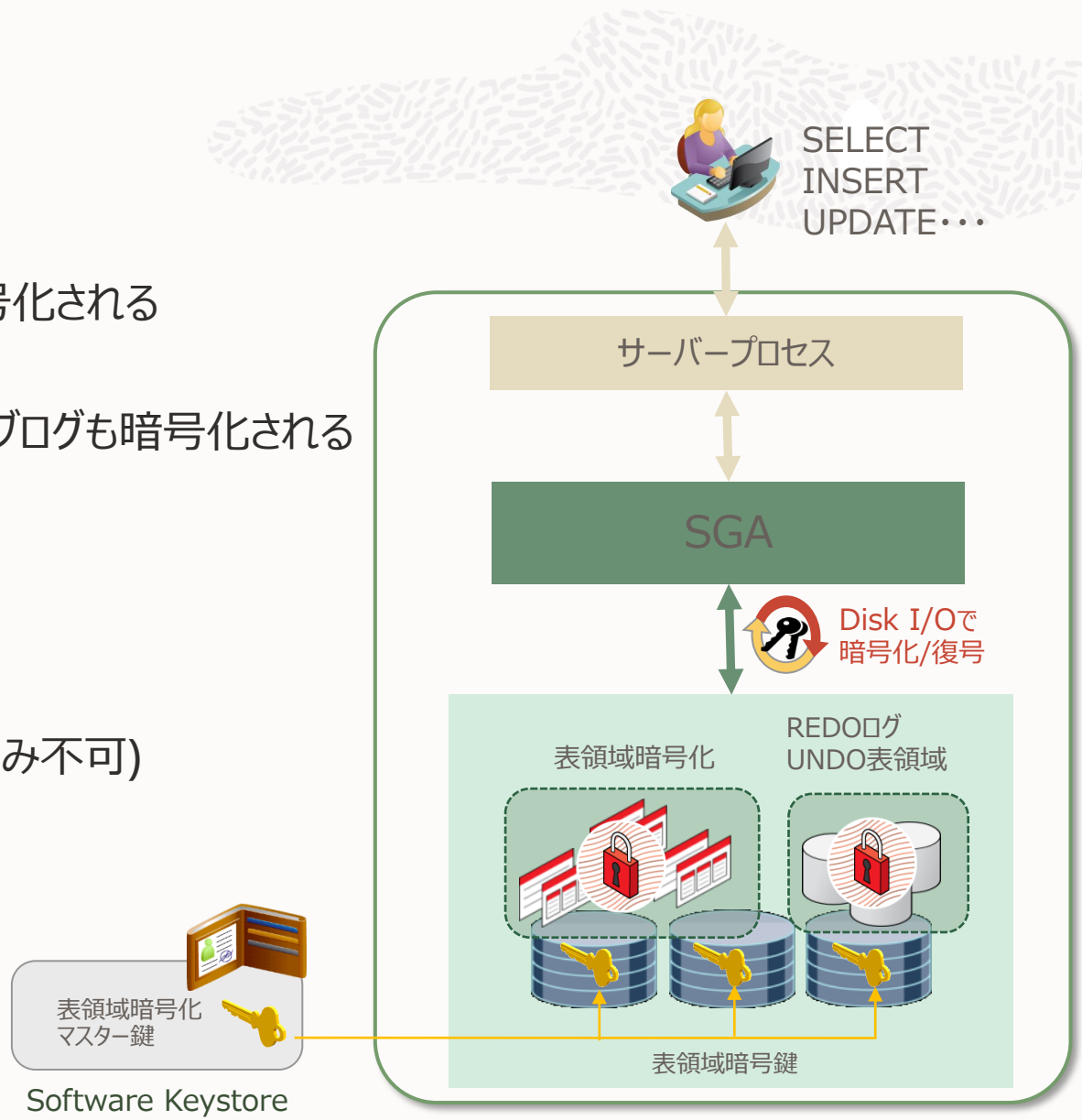
TDE 列暗号化 (10gR2～)

- 列単位での暗号化
- サーバプロセスで暗号化・復号
- REDOログ、UNDO表領域、アーカイブログも暗号化される
- SGAのバッファキャッシュは、暗号化されている
- 暗号化するとデータサイズは増加する
- ※暗号化アルゴリズムやSALT, MACオプションの有無に依存するが、暗号化される列の値ごとに1～52Byte増加する
- 表暗号鍵はデータディクショナリに格納
- 暗号列への索引は、B-Tree索引の一意検索のみ
- データ型とデータ長の制限が若干あり



TDE 表領域暗号化 (11gR1～)

- 表領域単位での暗号化
- 表領域内の表や索引などのオブジェクトはすべて暗号化される
- データブロックに対するI/Oで暗号化・復号
- REDOログ、UNDO表領域、一時表領域、アーカイブログも暗号化される
- SGAのバッファキャッシュ上は暗号化されていない
- 暗号化してもデータサイズは増加しない
- 表領域暗号鍵はデータファイルのヘッダーに格納
- 暗号列への索引に制限なし
- ほとんどすべてのオブジェクトが暗号化可能 (BFILEのみ不可)
- 既存表領域を暗号化する一括変換のサポート
- キャッシュヒットの高いSQLは性能への影響を受けず
  ディスクへのRead/Writeの多いSQLは影響を受ける



TDEマスター暗号鍵のアーキテクチャ

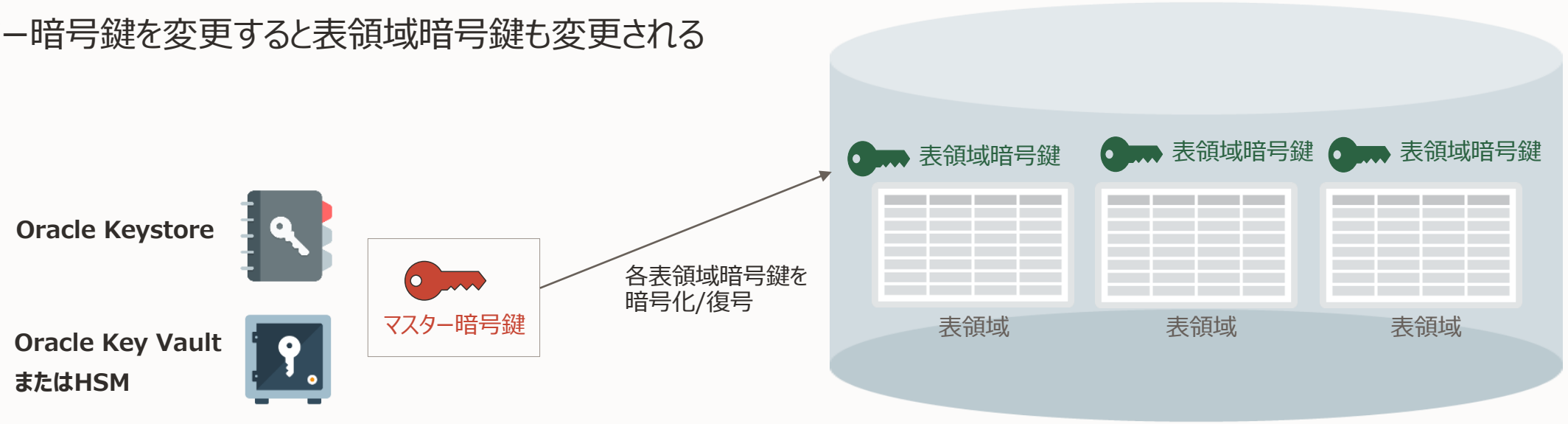
マスター暗号鍵は、データベースに基本的に1つ
- マルチテナント環境の場合は、各PDBごとに自身のマスター暗号鍵を作成することも可能

マスター暗号鍵のデフォルト保管先は、PKCS#11互換のSoftware Keystore(Oracle Keystore)のファイルに格納される
- Oracle Key VaultやHSM(Hardware Security Device)に格納し、ネットワーク通信で連携することも可能

マスター暗号鍵は、実際にデータを暗号化している表領域暗号鍵を暗号化・復号する

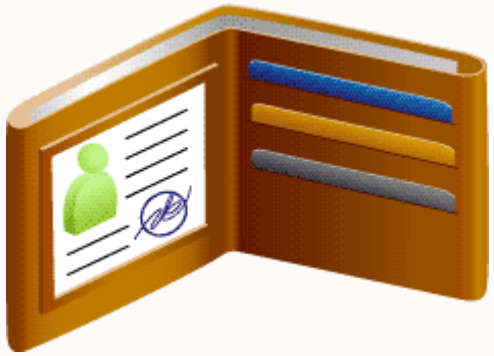
マスター暗号鍵をオープンすると暗号化されたデータにアクセスできる

マスター暗号鍵を変更すると表領域暗号鍵も変更される



マスター暗号鍵の状態・履歴を参照 (12c～)

V$ENCRYPTION_KEY

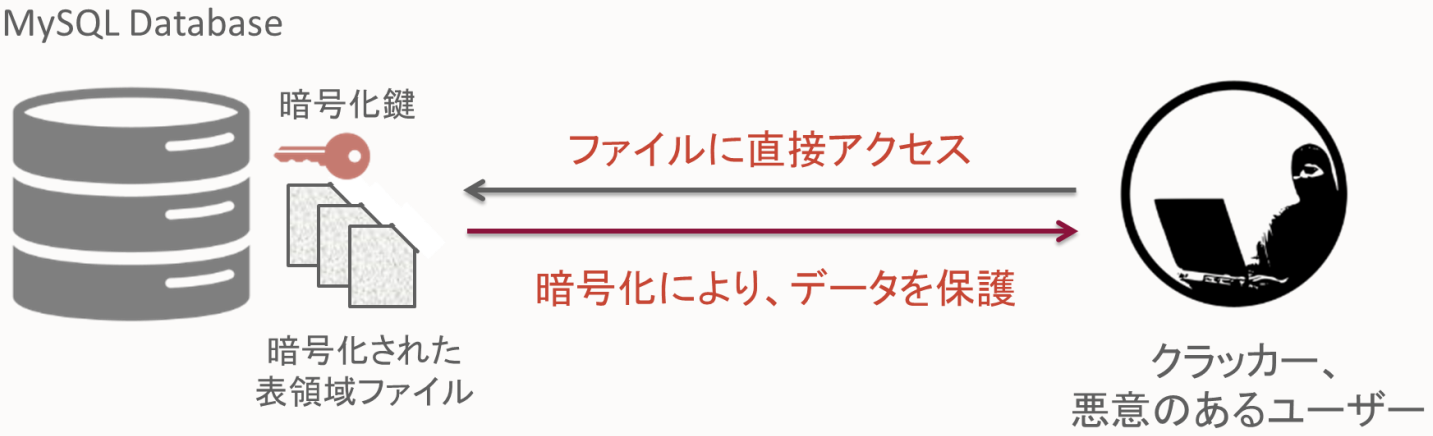


KEY_ID ・・・マスター暗号鍵
TAG ・・・任意のタグ（コメント）
CREATION_TIME ・・・マスター暗号鍵の作成日
ACTIVATION_TIME ・・・マスター暗号鍵のアクティブ日
BACKED_UP ・・・バックアップの有無
・・・・

KEY_ID	Tag	ACTIVATION_TIME	
QyB22Az…	システム移行のため	2021-12-30 08:02:00	In Use
L88bUKK…	2021年定期変更	2021-07-01 07:58:32	Rotated
EaZdf290…	バックアップ作業	2021-03-15 06:52:17	Rotated
ddJwwWJ…	2020年定期変更	2020-07-01 10:11:02	Rotated
GGgSB36…	システムC/O	2020-01-01 08:25:33	Rotated

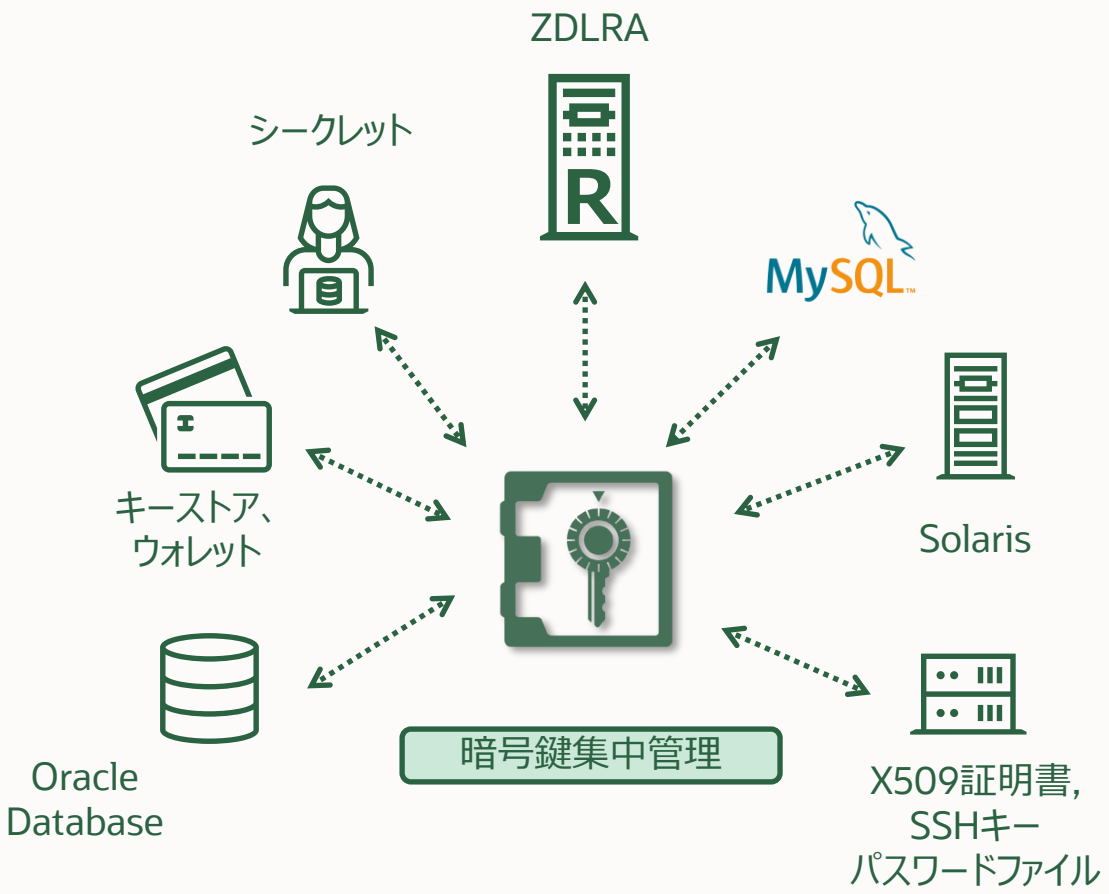
MySQL の暗号化技術：
MySQL Enterprise TDE（Transparent Data Encryption）

- シンプルなデータ暗号化（AES256）
- OSのファイルシステム上、バックアップメディア上のデータを保護
- 鍵管理機能を含む（鍵の保護、ローテーション、など）
  - Oracle Key Vaultと連携可能（KMIP v1.2に対応）
  - 鍵管理は、セキュリティ対策上重要なポイント
  - 2層暗号化鍵アーキテクチャを採用（マスター暗号化鍵、表領域鍵から構成）
- アプリケーションの変更不要
- 簡単な設定で導入可能
- 高パフォーマンス（オーバーヘッドが少ない）



暗号鍵の紛失、盗難に対する対策

暗号鍵や資格情報やパスワードなどのオブジェクトをセキュアに保管し、一元管理します。



Key Vault の特徴

認証に必要な情報を一元管理するアプライアンスを提供

保守、監視、バックアップ等の一連のサイクルをサポート

2種類の方式を提供

- 暗号鍵、資格情報のバックアップ & ダウンロード
- 暗号鍵のオンライン相互通信 (Direct Network Connection)

暗号化消去について

暗号化消去について

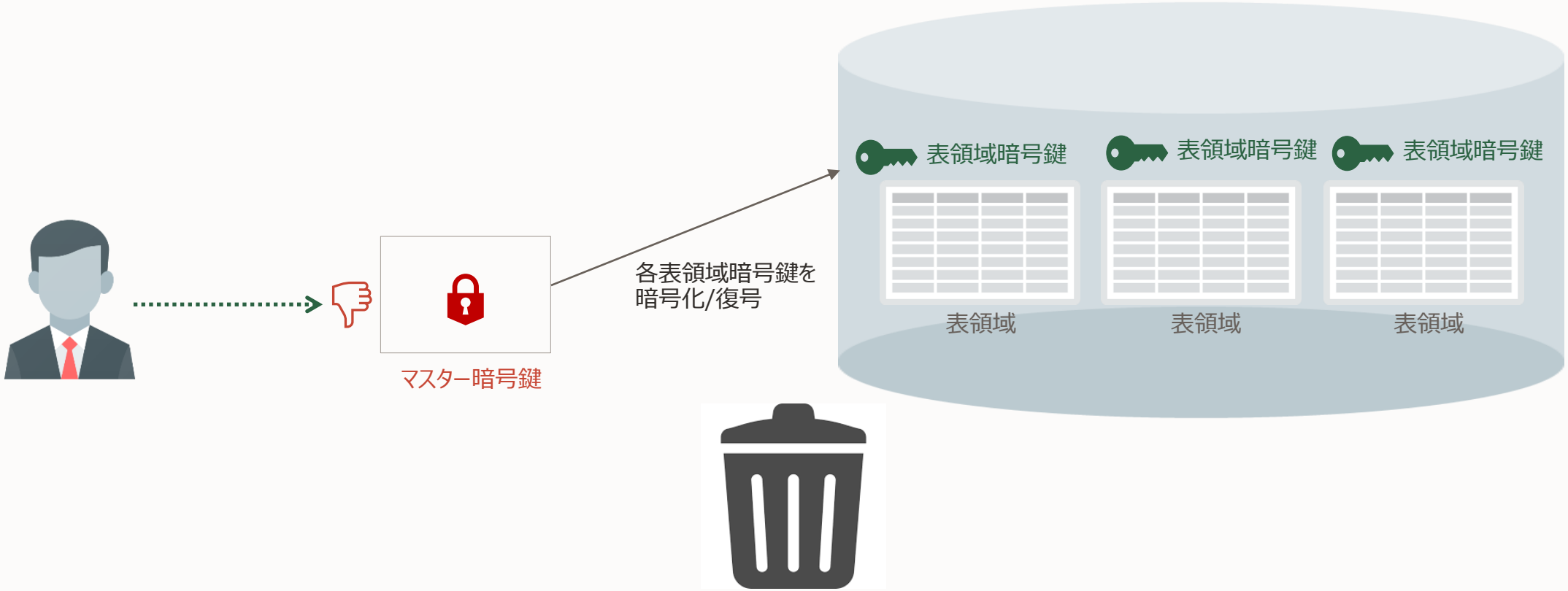
「暗号化消去」とは、情報を電磁的記録メディアに暗号化して記録しておき、情報の抹消 が必要になった際に情報の復号に用いる暗号鍵を抹消することで情報の復号を不可能にし、 情報を利用不能にする論理的削除方法をいう。

暗号化消去の適用時のポイント

1. 安全な暗号化アルゴリズムが利用されている事
2. データは保存する前に暗号化する。
3. 暗号鍵が安全に管理されていること

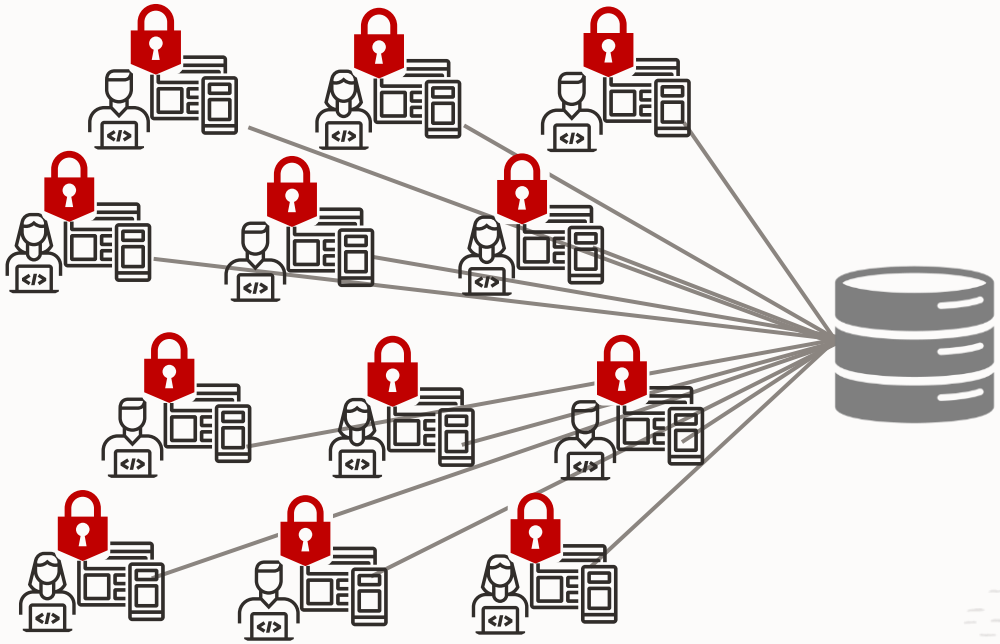
データベースでの暗号化方法について

マスター暗号鍵を削除することで、利用者からはアクセスが行えない



アプリケーションの暗号化について

アプリケーション毎に鍵が分散している場合、それぞれを適切に消去する必要がある



その他

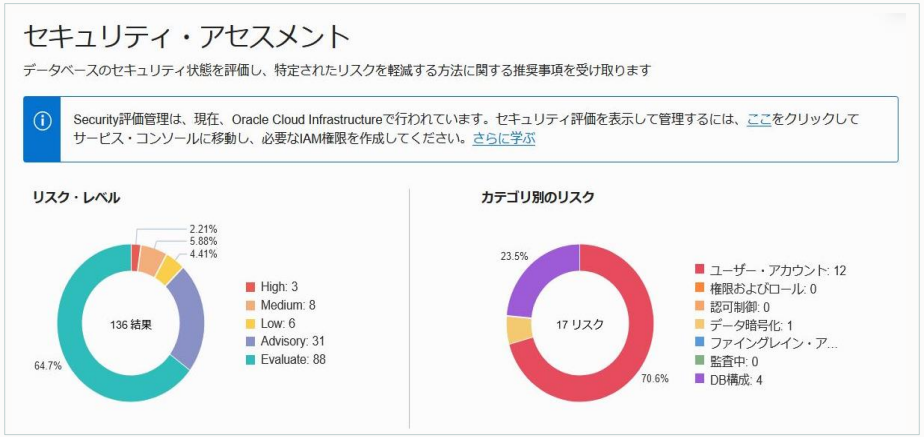
立ち止まって考えていること

1. 会社で守るべき情報の定義の確認
2. 機密情報が格納しているデータベースはどこにあるのか
3. 暗号化が正しく実装されているのか

暗号化が正しく実装されているのか

オンプレミスの場合

- Database の設定を確認する
- DBセキュリティ対策状況の可視化ツールを活用
- 鍵管理ソリューションを活用する

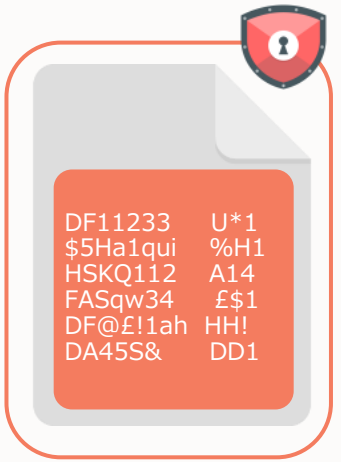


Oracle Database の場合：
アセスメントツール(Data Safe)の活用

クラウドの場合

- **強制的な暗号化機能**を活用する
- Database の設定を確認する
- DBセキュリティ対策状況の可視化ツールを活用
- 鍵管理ソリューションを活用する

強制的な暗号化



ご参考：Oracle Database 表領域の暗号化変換

表領域を新たに作成する必要がなく、

SQLで表領域を暗号化・復号を一括変換

既存の表領域を暗号化する手間や時間を大幅に短縮

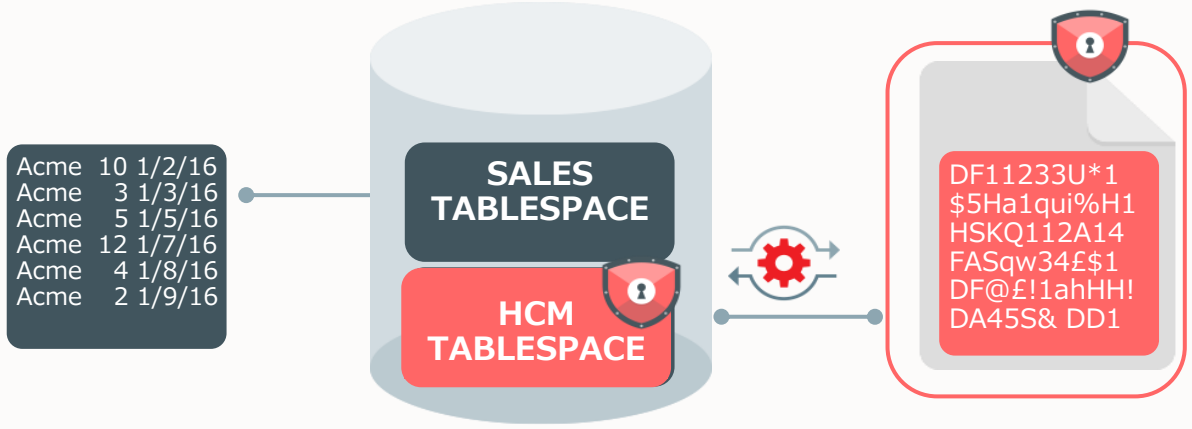
従来では暗号化できなかったSYSTEM, SYSAUX, UNDO, TEMPなどのシステム領域も暗号化することで
データベースのフル暗号化が可能

データにアクセス可能なまま表領域を暗号化

- **Online Encryption Conversion**

表領域がオフライン時に暗号化

- **Offline Encryption Conversion**



これまでは、暗号化した表領域を作成し、暗号化した表領域にデータを移行を行い、
暗号化されていない既存の表領域を削除が必要だったため、より運用が簡単に

さいごに：データ・セキュリティ対策で取り組むべきこと

データ・セキュリティ対策で考慮すべきこと (*)

1. **データ保護のセキュリティポリシーの策定**
   - 重要情報の定義 / リスク分析
   - アカウント管理 / ログ取得ポリシーの策定
2. **データセキュリティ対策の実装**
   - 防御系のセキュリティ対策（予防的対策）
     - 初期設定/認証/アクセスコントロール/暗号化など
   - 検知、追跡系のセキュリティ対策（発見的対策）
     - ログの管理/不正アクセス検知/ログ分析

今後もとめられる対策

セキュリティ構成の評価

重要情報の発見

アカウントのリスク評価

監査証跡の記録
(発見的対策)

暗号化/認証/アクセス制御 等
(予防的対策)

暗号化もとても大切ですが、データ・セキュリティ対策は、ポリシーの策定のために必要となるセキュリティ対策の可視化や守るべき情報を整理した上でなどの対策を行うことが重要です。

* 参照：データベースセキュリティコンソーシアム『データベースセキュリティガイドライン 第2.0版』


ORACLE