DB暗号化実装事例のご紹介

DB暗号化WGリーダー　髙岡　隆佳

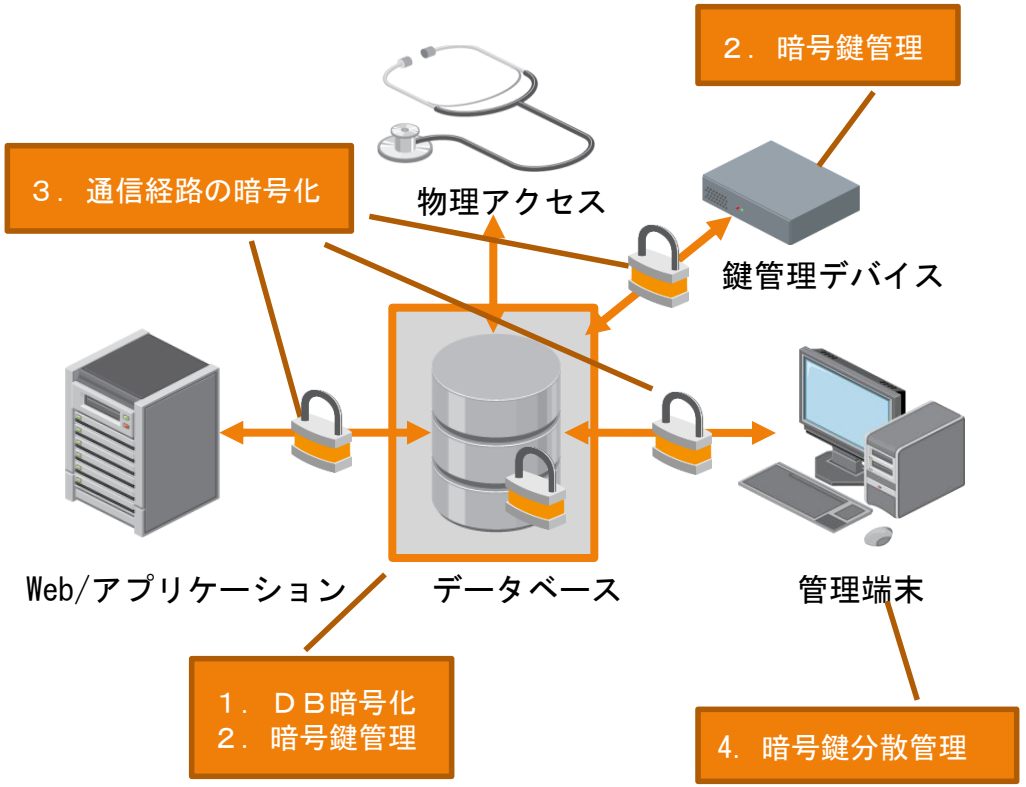
アジェンダ

- DB暗号化実装方法とレベル
- 実装事例紹介

DB暗号化実装方法とレベル

暗号化対策概要

- DB暗号化ガイドラインにおける暗号化対策
  - DB暗号化
  - 暗号鍵管理
  - 通信経路の暗号化
  - 暗号鍵分散管理



DB暗号化

- 暗号化分類
  - ハードディスク（HDD）暗号化
  - DBテーブル暗号化（オブジェクト含む）
  - DBカラム暗号化（アプリケーションでの暗号化、DBでの暗号化）
  - バックアップデータの暗号化

- 各ソリューション毎にその概要・メリット・デメリット・導入ガイドライン・実際の導入事例についてはDB暗号化ガイドライン1.0版にて紹介。

脅威と対策のマッピング

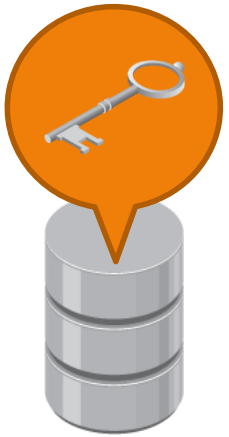
対応箇所	HDD抜き取り	バックアップ盗難	メモリダンプ解析	暗号鍵の盗難	パケットの盗聴	内部DBAによる暗号鍵の悪用
DB暗号化（HDD暗号化）	●	x	x	△	x	x
DB暗号化（DBテーブルの暗号化）	●	●	●	△	x	x
DB暗号化（DBカラムの暗号化）	●	●	●	△	x	x
アプリでのDB暗号化（DBカラムの暗号化）	●	●	x	△	x	x
暗号鍵管理（ソフトウェア）	−	−	●	△	−	△
暗号鍵管理（HSM）	−	−	●	●	●	●
通信経路の暗号化	−	−	−	●	●	−
暗号鍵アクセス制御	−	−	−	−	−	●
暗号鍵の分散管理	−	−	−	−	−	●

● −対応可能、 △ −製品によっては対応可能、 x −対応不可能、 ー −範疇外

- どこまでのリスクに対する対応が必要か判断

暗号化の効果

- 暗号化＝暗号化されていないデータとの分離→漏洩対策
- 効果は鍵の強度、管理手法に依存する
  - 鍵が誰にでも（管理者含む）アクセスされては暗号化の意味がない
  - 必要なときに必要な人が必要な分だけ鍵にアクセス
    - 鍵に対するユーザ（管理者）アクセスポリシーはどうする？
  - 暗号鍵が安全で完全性を保たれなければならない
    - 暗号鍵の適切な保存・管理をどうする？
- 鍵管理とアクセス制御が正しく設定されると･･･
  - 物理的な漏洩に効く！（HDD持ち出し、ベーステーブル持ち出し）
  - 内部不正に効く！（特権ユーザによる職権乱用）



PCI DSSに見るDBセキュリティ要件

- 暗号化における要件は共通している

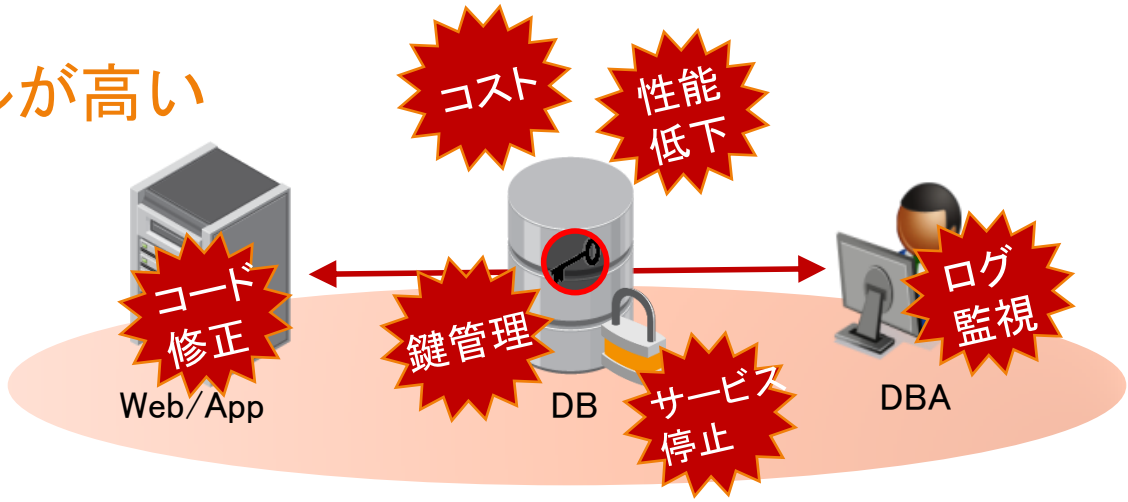
Build and Maintain a Secure Network	1. Install and maintain a firewall configuration to protect data
	2. Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	3. Protect stored data
	4. Encrypt transmission of cardholder data and sensitive information across public networks
Maintain a Vulnerability Management Program	5. Use and regularly update anti-virus software
	6. Develop and maintain secure systems and applications
Implement Strong Access Control Measures	7. Restrict access to data by business need-to-know
	8. Assign a unique ID to each person with computer access
	9. Restrict physical access to cardholder data
Regularly Monitor and Test Networks	10. Track and monitor all access to network resources and cardholder data
	11. Regularly test security systems and processes
Maintain an Information Security Policy	12. Maintain a policy that addresses information security

暗号化と鍵管理

鍵へのアクセス制御と監査

DB暗号化の課題

- 様々な懸念事項
  - パフォーマンスの低下（フルテーブルスキャン等）
  - 導入、および鍵更新に伴うシステム停止
  - 暗号化に伴うデータの肥大、データ長変更等
  - 暗号鍵の徹底した管理
  - 鍵に対するアクセスログの監査、管理
  - 導入コスト、工数
- →導入のハードルが高い



近年のDB暗号化ソリューション

- 課題の克服と新機能の登場
  - 暗号化のH/Wオフロード
    - AES-NI（インテル）との連携
    - 専用H/Wでの暗号化（TPM, HSM）
  - オンライン暗号化機能
    - システム停止なく暗号化の導入をサポートする機能
  - 専用H/Wでの鍵管理
    - 改ざん、不正利用の防止、鍵更新の自動化など
    - 開発を伴わないインテグレーション
  - トークナイゼーション（Tokenization）
    - データの匿名化による監査対象の削減
    - セキュリティ対策箇所の縮小
    - TSP（Tokenization Service Provider）による中小企業でのPCI DSS対策費用負荷の軽減

DB暗号化実装方法と事例

DB暗号化製品一覧

- DBMS付属の暗号化オプション
  - DB2 EE
  - MS SQL TDE
  - Oracle TDE (Advanced Security/Label Security Database Vault等)
  - Protegrity Database Protector for Teradata
- 3rdパーティ社製暗号ソフトウェア
  - CypherGate eCypherGate
  - D’ Amo
- 3rdパーティ社製暗号ハードウェア
  - SafeNet DataSecure
  - Thales nShield
- 暗号化製品全般
  - HDD暗号化製品
  - ストレージ暗号化製品
  - その他

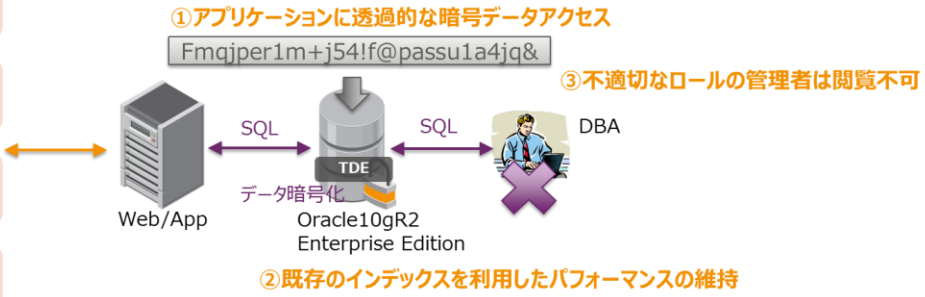
＊会社名順

事例1: 総務省制定ガイドラインへの対応

DB	・Oracle 10gR2
暗号化対象	・総務省のポリシーに沿った機密データの暗号化
テーブルサイズ	
データ移行時間	
暗号化製品	・Oracle Advanced Security Option
暗号化方式	・カラム単位
暗号鍵管理	・Oracle Wallet
アクセス制御	・ソフトウェア（ロールベースのアクセス管理）
通信経路暗号化	
導入までの期間	・2ヶ月
導入費用概算	

導入のポイント

・既存アプリケーションを修正することなく暗号化の適用が可能、既存のインデックスを暗号カラムに適用することでパフォーマンスを維持、OSのアップグレードのみで暗号化対応可能なため、コストと時間を最小限に抑えることが可能。

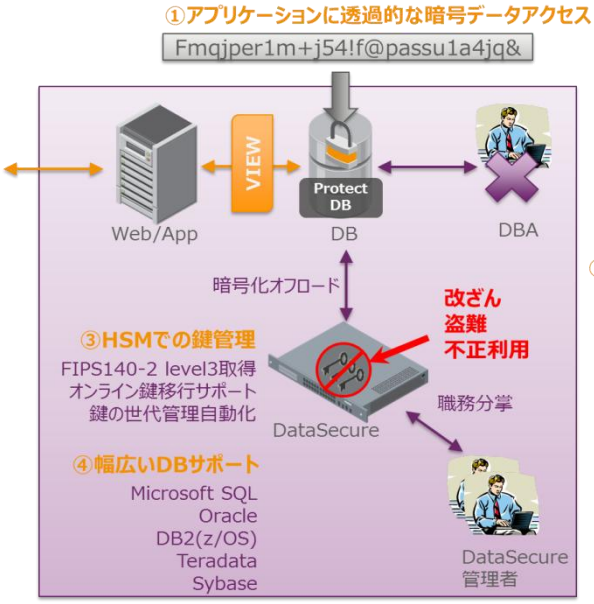


事例2: 自社決済サービスDBのPCI DSS対応

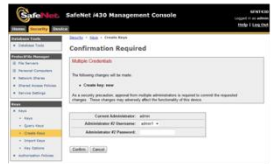
DB	·Oracle 10gR2/MS SQL2008
暗号化対象	·クレジットカード番号、および一部個人情報を含む5カラム
テーブルサイズ	·100～200万件
データ移行時間	·約15分
暗号化製品	·SafeNet DataSecure i450
暗号化方式	·カラム単位
暗号鍵管理	·HSM（DataSecure）、鍵の世代管理自動化、オンラインでの鍵交換対応
アクセス制御	·ソフトウェア（鍵への時間単位、参照回数制限）
通信経路暗号化	·SSL（DB-DataSecure間、DataSecure-管理端末間）
導入までの期間	·4ヶ月
導入費用概算	·1,200万円

導入のポイント

·異なるDBの暗号化を単一の管理システムで提供、HSMによる鍵管理の自動化、既存アプリケーションに修正を必要としない導入形態、DBに対するオンラインデータ移行・鍵更新の対応、PCI-DSSコンプライアンスに対する幅広い網羅性（要件３、４、７、９、10）、内部DBAに対するマスキングによるデータ保護、Active/Activeクラスタ構成によるサービス継続性、サーバ単位での暗号化ライセンス体系。



②データのマスキング
内部DBAの目視によるデータ漏洩を防御



⑤管理者の職務分掌機能
設定変更に複数の管理者認証

事例2: 自社決済サービスDBのPCI DSS対応

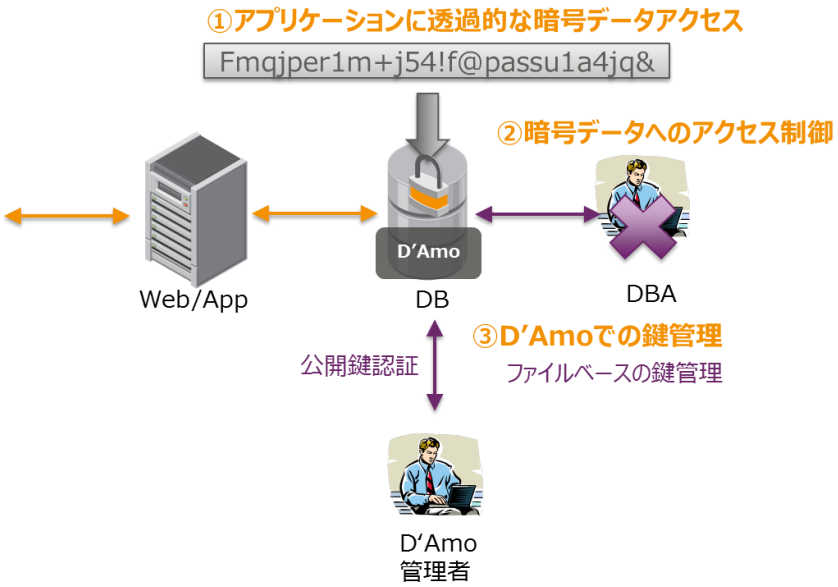
- 既存Oracle/MSSQL Server混在環境において、当初はそれぞれOracle /SQL2008の暗号化オプションの採用を検討していた。
- それぞれ別の管理を行うよりは、双方のDBMSを一元管理できるソリューションを、ということでDataSecureの検討を開始した。
- DataSecureがHSMであることからも、煩わしい暗号鍵管理がセキュアかつ自動化され、Oracle/MSSQL Serverに対する暗号鍵・セキュリティポリシーが一元管理できることで運用コストを下げることができた。
- クレジットカード番号を検索キーとしているが、DataSecureは暗号カラムにDomain Indexを適用できることから、一致クエリにおけるフルテーブルスキャンが回避でき、パフォーマンスを維持できた。
- 決済会社へ送信する決済ファイルを保存しているファイルサーバもDataSecureで暗号化対応することで、包括的にPCI DSSへ準拠ができた。
- サービスを止めずに鍵のオンライン更新が可能な点や、管理者一人の権限ではセキュリティポリシーを変更させない強固な内部統制機能により、PCI DSS各種管理要件の実装が容易だった。
- 鍵の使用を許可する時間帯、参照回数、参照権限を指定できる点は、PCI DSS要件7を実現するのに最適だった。

事例3：社内コンプライアンス対応

DB	・Microsoft SQL server
暗号化対象	・社員個人情報を含む2テーブル、各2カラム
テーブルサイズ	・40万件
データ移行時間	・約10分
暗号化製品	・D’ Amo for SQL
暗号化方式	・カラム単位
暗号鍵管理	・ファイルでの暗号鍵管理（D’ Amo内）
アクセス制御	・D’ Amoによるアクセスコントロール設定 （暗号化・復号権限）
通信経路暗号化	
導入までの期間	・1ヶ月
導入費用概算	・150万円

導入のポイント

・アプリケーションの再開発が不要で、暗号化・アクセスコントロールの設定が容易。



事例3: 社内コンプライアンス対応

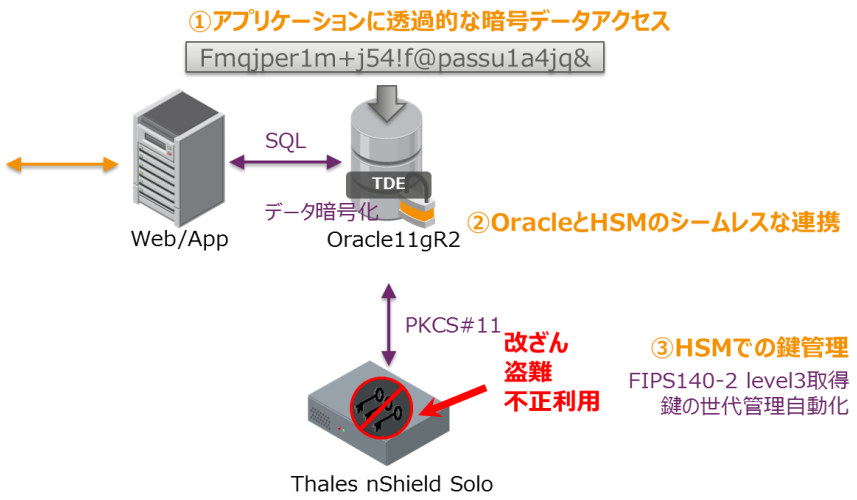
- 社内で稼働するデータベースに対して暗号化を適用した運用を行う。という社内コンプライアンスに対応するため、データベースの暗号化が検討された。
- 当初データベースアプリケーションを修正する方向で考えられていたが、可能な限り早急に暗号化を実装する必要があったため、パッケージソフトウェアであるD'Amoの検討、検証が開始された。
- D'Amoはデータベースサーバにインストールするだけで暗号化を実装することができるため、早急な実装という命題をクリアすることができた。
- データベースアプリケーションの修正作業が必要ないため、当初想定されていた導入コストよりも大幅に下げることができた。
- D'Amoの管理コンソールは1台で複数のデータベースにインストールされているD'Amoを一元的に管理することができるため、1台の管理者端末から複数のデータベース（MS-SQL、Oracle）を管理することで、運用コストを下げることができた。
- D'Amoによるアクセスコントロール機能により、データベース管理者であっても暗号化されているカラムを読むことはできないため、メンテナンス時におけるデータ取り扱いのリスクを大幅に削減することができた。
- 管理ツールが使いやすいため、簡単に運用することができた。

事例4: 金融におけるPCI DSS対応

DB	・Oracle 11g
暗号化対象	・クレジットカード番号
テーブルサイズ	
データ移行時間	
暗号化製品	・Thales nShield Solo, Oracle Advanced Security Option
暗号化方式	・表領域単位
暗号鍵管理	・HSM（Thales nSheild Solo）
アクセス制御	
通信経路暗号化	
導入までの期間	・2ヶ月
導入費用概算	

導入のポイント

・FIPS140-2 Level3に準拠したHSMによる暗号鍵管理、HSMによるPCI-DSS要件３への準拠、アプリケーションに対して透過的に暗号化ができ、DBMSのセキュリティ機能とHSMのシームレスな連携が可能。

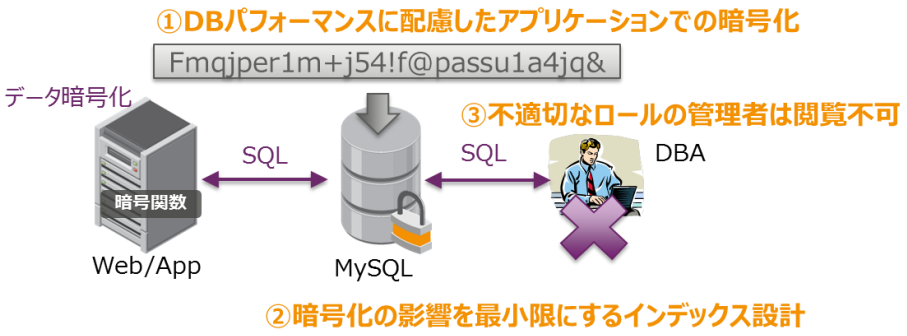


事例5: 「ポイントシステムASPサービス」における個人情報保護対策

DB	·MySQL
暗号化対象	·個人情報
テーブルサイズ	·160万件
データ移行時間	
暗号化製品	·Blowfish暗号のアプリケーション組み込み
暗号化方式	·カラム単位
暗号鍵管理	
アクセス制御	
通信経路暗号化	
導入までの期間	
導入費用概算	

導入のポイント

·MySQL自体は暗号機能が標準でもたないため、アプリケーション側に暗号機能を実装、結果データベースへのパフォーマンスを配慮した設計が実現。



事例5: 「ポイントシステムASPサービス」における個人情報保護対策

- MySQLに個人情報を持つユーザにて暗号化の要件が上がったが、MySQL自体に暗号化機能は標準ではなかった
- 暗号化しても引き続きASP利用ユーザ側でのデータ分析に影響がないような設計が求められた。
- システム再構築のタイミングでの暗号化実装であったため、アプリケーション側への暗号化機能実装（暗号アルゴリズム：Blowfish）が採用された。
- 通常暗号カラムについてはインデックスは役にたたないため、CRM分析が必要なカラムについてはデータを分割（住所であれば都道府県市町村以降とカラムを分ける）することで、フルテーブルスキャンやインデックス利用不可といったデメリットを排除した。

DB暗号化市場の動き

- PCI DSSを中心としたECサイト等での対応増
  - コストに応じた暗号化手法の選択とそのリスク掌握が必要
  - レベル1加盟店ではトークナイゼーション適用も視野に
  - TSPによる中小企業でのPCI DSS適用加速

- 個人情報保護「高度な暗号化」への適合
  - 漏洩に対するリスクの回避
  - 何を持って高度な暗号化とするかはグレー（法第20条）
    - 電子政府推奨暗号リスト又はＩＳＯ/ＩＥＣ18033に掲げられている暗号アルゴリズムによって、記録媒体内の個人情報の保存先として利用可能な全領域が自動的に暗号化されること。
    - 暗号化された情報及びその暗号化された情報を復号させる復号鍵の管理が適切にされていること。
  - 十分な強度の暗号アルゴリズムの採用・暗号鍵に対する物理的保護・鍵への厳格なアクセス制御・管理者への管理・監査導入は必須

- DBもクラウドへ
  - AzureやAWSを始めクラウド上でDBサービスの提供が活発化
  - 混在環境でのデータ保護をどうするか
  - 暗号機能の提供レベル（ユーザではなくクラウド側依存）
  - 暗号鍵のコントロールをオンプレ側で実行・管理するサービスも