データベース統合におけるセキュリティ対策

Security Countermeasure in Database Integration

八津川　直伸

要　約　一般的に企業の情報システムでは，部門毎に独自のデータを管理するなど，マスターデータが複数存在する状態に陥っている．企業経営の観点からは ROI（Return On Investment）改善のため，マスターデータを統合し，同時にセキュリティ管理を充実したいという動機がある．一方，サイバー攻撃など外部からの不正アクセスや内部犯罪による情報漏洩事故が頻発している．攻撃技術と対策技術はイタチごっこであるため，企業システム防衛にあたっては，新たな方法による攻撃を受けても被害を最小限に抑える仕組みを備えることが肝要である．

特にデータベース統合の局面では，異なるセキュリティ機能やセキュリティポリシーを持つ個別データベースを統合するため，統合データベース全体としてのセキュリティを如何に維持するかが課題となる．

本稿ではまず，世の中の基準やガイドラインに照らし，望まれ得る一般的なデータベースのセキュリティ要件を述べ，さらに仮想的データベース統合に特有なセキュリティ要件を整理した．これらをデータベース統合の際のセキュリティ要件策定に適用することにより，不正アクセスや，内部犯罪，誤操作に起因する機密情報や個人情報の漏洩・破壊，ならびに障害，災害による企業情報の損壊・滅失による業務継続不能といったセキュリティ上の脅威やリスクを低減できる．

Abstract　In general, the corporate information system often falls into the existence of multiple sets of master data as a result of a manner where individual department controls own data. From the viewpoint of business management, for the ROI (Return On Investment) improvement, the corporation has the motives for integrating multiple sets of master data and enhancing security management at the same time. On the other hand, the information leakage incidents caused by the illegal accesses from outside such as the cyberattack, and crimes inside company have occurred frequently. Because attack techniques and defense techniques play never-ending cat-and-mouse game between them, in the defense of the corporate system, it is important to establish the mechanism to minimize the damage even if the system is attacked by a new technique.

Especially, in integration of individual databases with different security features and policies, it is also important how to maintain the security of the integrated database as a whole.

This paper firstly discusses the security requirements of general database desirable according to security standards and guidelines in the world, next, finds the security requirements specific to the virtual database integration. Through the application of proposed requirements to actual security design during database integrations, it is possible to reduce the security threats and risks such as the leakage and destruction of confidential or personal information caused by unauthorized access, internal crimes, or erroneous operation, including unsuccessful business continuation due to destruction or loss of corporate information caused by system failures or disasters.

1. はじめに

多くの企業は，かつてメインフレームで構成されていた情報システムのダウンサイジングに伴う分散化やM&Aによる企業再編等によって，部門ごとに独自にデータを管理することとなり，マスターデータが複数存在する状態に陥っている．情報の保管場所は企業内外の各所に分散し，複数のデータベースに同一のデータが存在（レコード重複）したり，同一のデータであってもコード体系が異なったり，あるいはデータ形式も統一されていない等の弊害が生じている．このような状況を放置することは低い業務効率と高コスト状態が継続することとなり，企業経営においてはROI（Return On Investment）の悪化につながる．そのため，マスターデータを統合して一つにしようという動機がある．しかし，企業システムが利用する内外の全てのデータを一気に統合するには，多額のコストと時間が必要であるため，段階的にデータ統合を進めようと考える企業が一般的である．その過渡的な方法の一つとして仮想的にデータベースを統合する手法がある．

日本ユニシスでは，企業内に分散して存在する既存データベースを統合する技術として，PostgreSQLを活用した仮想データ統合基盤（PostgreSQL Federation Database System，以下PGFDBSと称する）[1]を研究・開発した．一般的に，このような統合基盤は連邦型データベース（Federated Database System，以下FDBS）と呼ばれる．

FDBSは，分散する既存のデータベースに存在するデータを利用可能にする仮想的なデータベース統合技術であるが，異なるセキュリティ機能やセキュリティポリシーを持つ個別データベースを統合するので，統合データベース全体としてのセキュリティを如何に維持するかが課題となる．本稿ではこのFDBSを題材として仮想的なデータベース統合の際のセキュリティ要件や留意点について述べる．

2. 仮想データベース統合システムFDBSの概要

FDBSは既存のデータベース（DB）システムとアプリケーションの間に位置し，複数の異種データベースを仮想的に統合し，分散するデータ群に対する透過的かつリアルタイムなデータ参照と更新を可能とする．FDBSは既存システムのデータベース群に対するクライアントの位置づけであり，既存システムに対する変更は不要という長所を有する．

利用者は，バックエンドにある既存の個別DBのテーブルをFDBSの仮想スキーマ上に一元的に定義されたテーブルとして参照可能である．これによって，利用者はあたかも単一データベースであるFDBS上に存在するテーブルのように問い合わせをすることができる．図1に，FDBSのアーキテクチャの概略を示す．

このような仮想データベース統合は，複数の異なるプラットフォームやデータベースによる基幹業務システムを持つ中規模以上の企業や，M&A等企業合併による業務システム統合の課題を抱える企業に適用が推奨される．

FDBSは有用な統合技術ではあるが，その構築・運用にあたっては単一のデータベースセキュリティに加え，留意すべき点がいくつかある．次章以降では，まずデータベースのセキュリティ維持に必要な一般的セキュリティ要件を挙げ，次いでデータベース統合に特有なセキュリティ要件に言及する．

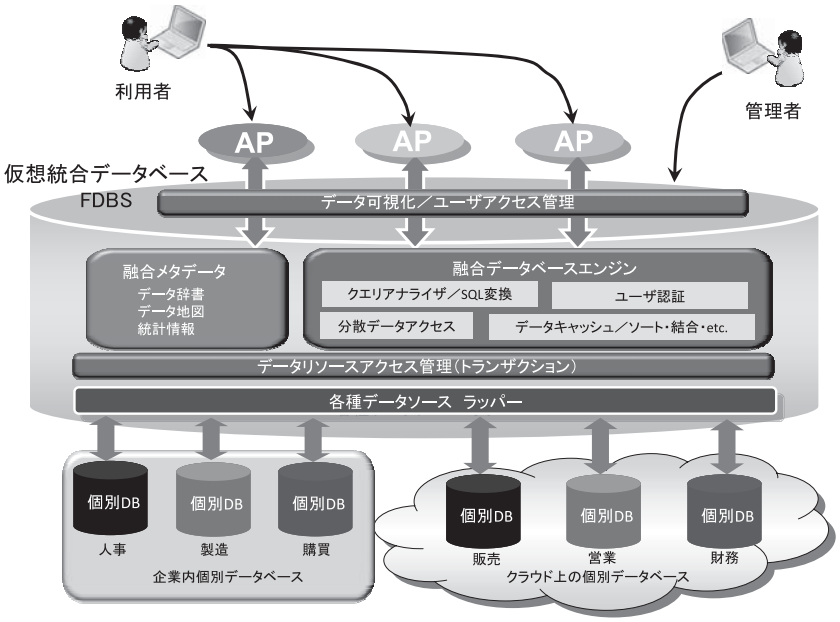

図 1　FDBS アーキテクチャの概要

3.　セキュリティ対策の検討ステップ

一般的な情報システムの基本的なセキュリティ対策のステップを以下と図 2 に挙げる.

1)　調査・状況認識ステップ

まず，前提条件を確認し，保護すべきデータやシステム構成要素，およびシステムに関わる利用者等を正確に把握する.

2)　脅威分析検討ステップ

ステップ 1)で状況を正確に把握した後，システムの安全性の観点から，保護すべきデータやシステムに対する脅威を洗い出し，リスク分析を行い，その後，対策としてのセキュリティ要件を定める.

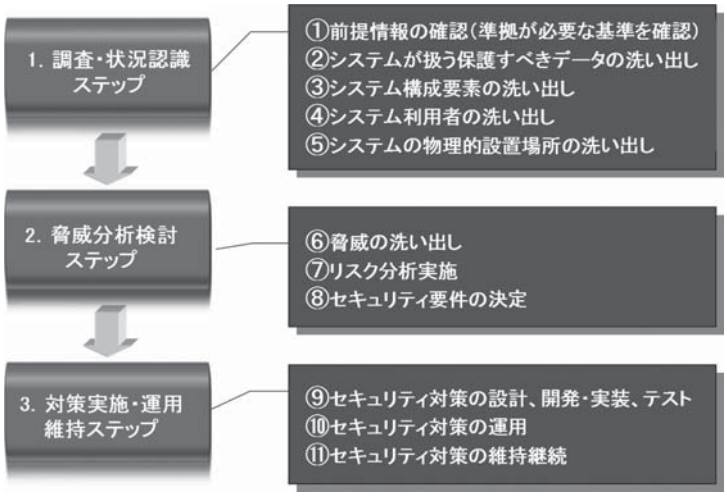

図 2　基本的なセキュリティ対策のステップ

3） 対策（セキュリティ設計/開発）実施・運用維持ステップ

ステップ 2)で定めたセキュリティ要件に対して，具体的な対策を決定し，システム設計に反映する．本番稼働後は，セキュリティ対策の適切な運用を維持継続する．

一般のシステムと同様に FDBS においても，具体的な業務プロセスとシステムの全容が見えないと脅威の識別とリスク分析ができず，正確なセキュリティ対策は定められない．そこで本稿では，国際標準でもある国内セキュリティ規格の JIS Q 27000 シリーズ[2][3]，各省庁が定めるセキュリティに関する基準[4][5][6][7]，省庁関連団体が定める安全対策基準等関連ガイドライン[8][9]，クレジット業界等民間団体が定める基準[10]~[15]等々，世の中に存在する各種基準・指針・ガイドラインに鑑み，FDBS のような仮想データベース統合において必要と想定され得る望ましいセキュリティのあり方について示唆する．

4. 想定する脅威とセキュリティ要件

脅威としては，システム外部あるいは内部からの不正アクセスによる機密情報や個人情報の漏洩，およびシステム破壊やデータ破壊が挙げられる．さらに悪意ではないが，誤操作等の事故や災害等によってシステムやデータそのものが滅失し，業務継続不能に陥ることもある（図 3）．

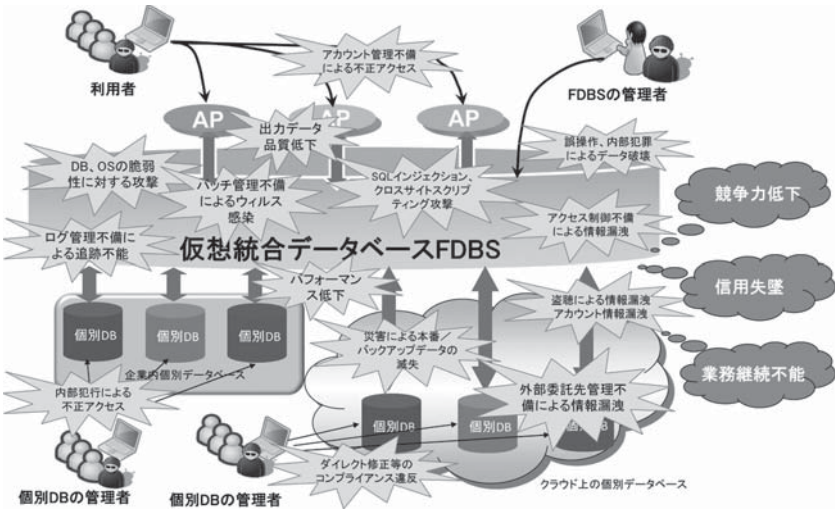

図 3　FDBS 上で想定される脅威とリスク

これらの脅威から想定されるリスクを整理すると以下の 4 点に集約される．

1） 機密情報漏洩による企業競争力低下

2） 個人情報漏洩による企業の信用失墜

3） データ破壊による業務継続不能

4） 企業情報の損壊，滅失による業務継続不能

本章では，企業の存続を脅かすこれらのセキュリティ上の脅威やリスクへの対策を述べる．これらの対策は，FDBS および分散する個別 DB の双方に必要となる．

4.1　アカウント管理（ID 管理）

システムのセキュリティ強化や対策は，一意のアカウント環境があることが大前提であり，データベースにおいても全ての利用者のアカウントとアクセス権限が適切に管理されることが要求される．管理者 ID/ユーザ ID の管理手続きを定め（表 1），アクセス権限管理に資する各種照会機能（表 2）を備えることが必要である．

表 1　管理者 ID/ユーザ ID の管理手続き

項番	要件
1	管理者ID/ユーザID の権限付与承認者と登録実施者を分離する．
2	アクセスコントロール方針に基づき，保護資産ごとにアクセスレベルを定める．
3	アクセス権限の付与申請・承認・発行・削除等に係る手続きを定める．
4	管理者ID/ユーザID の登録・変更・削除の管理者を定め，一元管理する．
5	職務，職位に照らして必要最小限のアクセス権限を付与し，不要になった際は速やかに削除する．
6	管理者ID/ユーザID 及びアクセス権限の設定状況について，定期的な自主点検(監査)を行う． ・利用者のアクセス権を定期的及び人事異動や組織改編時にレビューする ・管理者ID/ユーザIDの共用は禁止する
7	退職者・転出者の管理者ID/ユーザID は速やかに削除する．
8	長期間使用されていない管理者ID/ユーザID は削除する．

表 2　アカウント照会機能

項番	要件
1	ユーザID一覧の照会／出力
2	ユーザ権限（ユーザに割り当てられたロール）の照会／出力
3	ファイルごとのアクセス権限照会／出力
4	ユーザID設定オプション内容照会／出力（保有権限，最終使用時刻等）
5	パスワードオプション設定内容照会／出力（桁数，変更期間，連続誤入力許容回数等） 詳細は4.2.1項参照
6	アクセス権限に係る変更履歴の照会／出力
7	ロール一覧，ロール権限内容の照会／出力

4.2　アクセス制御

データベースへの正当な利用者のアクセスを確実にし，認可されていないアクセスを防止する．これを保証するために，本節で述べる利用者の識別と認証・認可機能が最低限必要である．

4.2.1　パスワード認証の強化

パスワードの安全性向上のため表 3 に示す機能が望まれる．

表 3　パスワード安全性向上

項番	要件
1	システムの全ユーザに対して認証機能を提供する．
2	ユーザIDは一人につき一個付与する．
3	入力されたパスワードは非表示（アスタリスク等）にする．
4	認証エラーメッセージとして，不要な情報を表示しない．
5	同一管理者ID/ユーザIDによる同時ログインを抑止する．
6	初期パスワードは初回ログイン時に強制変更させる．
7	一定回数連続した認証失敗時にはアカウントをロックする．
8	パスワードの有効期限を設定し強制変更させる．
9	パスワードの最低利用日数を設定する．
10	パスワード履歴を管理し繰返し利用を制限する．
11	安全なパスワード再設定（失念対策など）機能を備える．
12	パスワード文字列の複雑性実現機能を備える． a）パスワードの桁数を設定できること b）パスワードの文字種（英・数・記号）を設定できること c）安易な文字列（1234，password等）設定を禁止できること d）パスワードの選択及び利用時に，利用者が遵守すべきセキュリティ要求事項（安全なパスワードを選択する，定期的に変更する等）を利用者に提示できること

4. 2. 2　不要なアクセス権限は与えない

データベースへアクセスするアカウントユーザ（アプリケーションであることが多い）には，認証とともにその利用形態に応じた最低限の権限を付与する．データベースへのアクセス形態（読み出し，書き込み等）に応じた各種権限の組み合わせを持つ複数のアカウントを設け，それぞれの場面に適したアカウントを使い分ける．具体的には，読み出し，変更，追加，削除，テーブル構成変更，インデックス作成，ストアドプロシジャー実行等の各権限を適切にアカウントに割り当てる等，データベースへのアクセス権限管理（ロール管理）を厳密に行うことが望ましい．

4. 2. 3　タイムアウト機能を備える

以下のいずれかの動作をするタイムアウト機能を備えるべきである．

1）　無通信状態で一定時間経過後，利用者セッションを切断する．

2）　無操作状態で一定時間経過後，データベース管理端末等を保護する（ロック機能等）．

4. 2. 4　クロスサイト・スクリプティング，SQL インジェクション対策

インターネットに接続したシステム（主に Web アプリケーション）の場合，クロスサイト・スクリプティング（Cross Site Scripting：XSS）[*1] や SQL インジェクション攻撃による情報漏洩やセッションの乗っ取りが多発している．2011 年末までに IPA（独立行政法人 情報処理推進機構）に届出があった全ての Web サイトの脆弱性の 63％が XSS および SQL インジェクションであり，また 2011 年に限ると 90％以上が XSS である[16]．本項では XSS および SQL インジェクションの必須対策を述べる．また，図 4 にはデータフロー上の各対策の位置を示した．

1）　XSS 対策

Web アプリケーションにおいて，クライアントからの入力を別の処理系への出力とする場合，不適正な出力によりシステムがセキュリティ上問題のある動作を起こすことを防ぐため，表 4 のように特殊な文字/文字列を完全に排除しクライアントからの入力を無害化する．

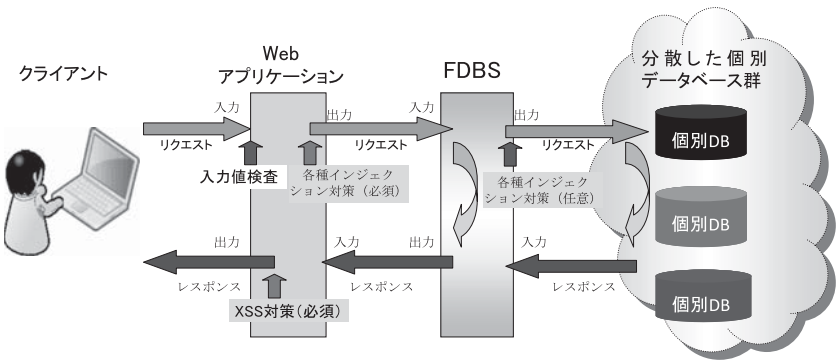

図 4　アプリケーション，データベースに対する入出力チェック

表4　クロスサイト・スクリプティング（Cross Site Scripting：XSS）対策

項番	要件
1	HTMLへ出力する場合，予期しないスクリプトの動作（の割り込み）を防ぐため，「<」「>」「&」「”」「’」の文字は，それぞれ「<」「>」「&」「"」「'」とエスケープ処理し，文字として扱われるように変換して出力する．
2	ダウンロードやファイル閲覧のため，ユーザの入力データをファイルアクセス時の引数（ファイル名）として利用する場合，PATHトラバーサル対策として，「..」や「/」（あるいは「¥」）等の文字をエスケープ処理してファイルを呼び出す．
3	クライアントからの入力データが，そのままHTTPヘッダに埋め込まれる（URLのリダイレクト先やCookie）アプリケーションでは，HTTPレスポンス分割対策として，入力に「CR」や「LF」が含まれている場合はそれらを排除する．

2)　SQL インジェクション対策

クライアントから入力されるデータを受け取る Web アプリケーションは，HTTP リクエスト（ヘッダー，ボディ）が改ざんされている可能性を考慮して，毎回必ず，表5に挙げる方法で全ての入力データの正当性をチェックする．これにより，システムの誤動作や，データ汚染・破壊を防止する．なお，FDBS から個別 DB への出力のインジェクション対策は，Web アプリケーションの SQL インジェクション対策が不完全な場合を想定しており，必須ではない．

表5　SQL インジェクション対策

項番	要件
1	入力値検査において業務要件上の許容範囲内で，文字種，不要な文字（群），サイズ等について，必要最小限に受け入れ範囲を絞る．その上で，項番２以下の処理を行う（サイズはバッファ・オーバーフロー対策）．
2	入力されたデータの検証(基本的には，排除または拒否)を，業務的な仕様等一定のルールの下で行う．置換が要求される場合もあるが，文字数が変わる等コードが煩雑になるので基本的に範囲外のデータは排除または拒否する．
3	業務上，排除または拒否ができず受け入れる必要があるデータについては，項番４，５のように，出力時にデータを検証（基本的には，無害化）する．
4	SQLによりデータベースをアクセスする場合は，SQLインジェクション対策として，バインド変数を使ったアクセス（Java：Prepared Statement，.NET：PrameterizedQuery）を利用する．バインド変数が利用できない場合（order by句，where条件全体，対象tableが可変となる等）は，SQL文を発行する際，SQLとしての特殊文字（「’」「%」「_」等）をエスケープ処理する．なお，LDAPやXML等RDB以外のデータソース類にアクセスする場合でも，それらのインターフェイスで使用する特殊文字についてエスケープ処理する必要がある（LDAPインジェクション，XPATHインジェクション対策）．
5	項番４と同様にクライアントからの入力データをOSコマンドの呼び出しの引数として使用する場合は，OSコマンドインジェクション対策として，呼び出し（出力）の際に，そのシステムで特別な意味をもつ文字をエスケープ処理して引き渡す．通常は，このサニタイジングによる対策以前に，汎用的にシェル，コマンドプロンプト等を呼び出すことを禁じ，想定した動作以外が起きないようインターフェイスを設計する．

4.2.5　認証機能の強化

システム，データベースの機密度に応じて以下のように認証機能を強化することが望ましい．

1)　パスワードの盗聴対策強化（OTP[*2]，CHAP[*3] 等）

2)　二要素認証によるユーザ認証精度の向上（トークン所持とパスワード等）[*4]

3)　生体認証による認証精度の向上（指紋，手形，網膜パターン，静脈パターン，顔認証，筆記パターン，キーストローク等）

4.2.6　その他の留意点

データベースにアクセスするアプリケーションを設計する際には，特に以下の点を考慮することが望ましい．

1） ソースコード内やスクリプト内へのパスワード記述禁止

データベースにアクセスするアプリケーションやスクリプト内にDBアカウントのパスワードを埋め込んだりすると，それらのソースコードが漏洩したときに不正アクセスの脅威が生じる．サーバの設定誤りやプログラムの脆弱性によりASPやCGIなどのサーバサイドスクリプトのソースファイルが漏洩する可能性があるので，ソースコードやスクリプトファイル中にDBアカウントのパスワード情報を埋め込んではならない．

2） システム関連情報の隠蔽

アプリケーションやデータベースに関する重要なシステム関連情報は暗号化やレジストリ（Windowsの場合）に保存するなどして容易に第三者に渡らないように隠蔽する．

3） システム領域とデータ領域の分割

オペレーティングシステムやアプリケーションのシステム領域とデータベースのデータ領域は別パーティションに構成するなど分割することが望ましい．これにより，OSやアプリケーションの障害によるデータ破壊がデータベース領域に及ぶことを避ける．

4.3 ログ管理

ログは，データベースに係る各種プロセス実行の記録であり，これによってデータベースの稼働状態や処理の実行状態，障害や異常の発生状況の把握，不正行為の追跡等が可能となる．これを実現するために必要な機能を本節に挙げる．

4.3.1 ログの収集

FDBSおよび個別DB上の情報にアクセスし，閲覧・更新・削除などを行った操作記録としてのログを収集・記録する（表6）．

表6 ログ収集機能

項番	要件
1	閲覧・更新・削除などを行った操作を記録できること．
2	正確な時刻を記録するため外部から正確な時刻を取り込めること．
3	以下の基本的なログ情報を収集できること． ・利用者ID ・利用（アクセス）日時 ・情報資産（データ）へのアクセスの成功と失敗（試みと回数）の記録 ・アクセス対象の情報資産名 ・情報資産に対する操作内容 ・各種デバイスの取り付け，取り外し ・データベースおよびシステムの起動および停止 ・管理者IDの新規作成，変更，アクセス権限設定の記録 ・特権的ユーティリティの使用状況 ・重要プログラムの使用状況
4	ログは自動出力され一元管理できること．

4.3.2 ログの保全

採取されたログが容易に改ざんされたり削除されると，不正行為の追跡等ができなくなるので，表7のような適切な保全措置が必要である．

表 7　ログ保全機能

項番	要件
1	収集後のログが不正に利用されることがないように，アクセス制御，暗号化，改ざん防止，改ざん検出等の機能を有すること．
2	収集したログは統計的な分析ができ，グラフ等に結果を出力できる機能を有すること．
3	ログを収集・長期保存・バックアップができ，必要に応じて閲覧できる機能を有すること．
4	ログの検索・集計結果は，CSV 形式又はPDF形式等適切な形式で出力する機能を有すること．

4. 3. 3　ログ管理の運用

FDBS のログおよび分散する個別 DB のログを統合管理できるよう，表 8 を考慮することが望ましい．

表 8　ログ管理の運用

項番	要件
1	どのようなログを収集するかを決める．
2	収集するログの内容を決める．
3	ログの保存場所と保存方法を決める．
4	ログを保管するサーバは安全な場所に設置する．
5	ログ管理サーバは冗長化する．
6	ログ収集時にデータベース機能本来の動作速度を著しく低下させないこと．
7	ログの保存期間，ローテーションを決める．
8	収集したログを定期的に点検・分析し，結果に応じて必要な情報セキュリティ対策を講じるプロセス（セキュリティ侵害時の対応など）を定める．
9	疑わしい事情があったとき，管理者に通知する機能を備える．
10	外部媒体に保存する場合は，追記・更新不可の記録媒体へ記録し，記録媒体は施錠保管する．
11	収集したログは，バックアップを取得する等，ファイルの破損，消去等から保護するための措置を講ずる．
12	ログのバックアップは，十分離れた場所に保管する．

4. 4　バックアップ

分散する個別 DB の故障や破壊または災害の発生の際，全ての重要なデータおよびソフトウェアの回復を確実にするために，適切なバックアップ対策を備える．FDBS 自体は仮想 DB のためデータの実体は持たないが，FDBS の設定情報，定義情報，および制御情報としてのトランザクションログ等はバックアップする必要がある．本節にその機能要件を挙げる．

4. 4. 1　バックアップ機能とその運用

各個別 DB においては，表 9 に示すバックアップの機能要件を備え，表 10 の運用要件に従い，適切にバックアップ運用することが望ましい．バックアップ運用は，手順書を備えることが肝要である．また，バックアップしたデータがリストアできなければ意味が無いので，リストアテストは定期的に行うことが望ましい．

表 9　バックアップ機能

項番	要件
1	計画した時間にリカバリに必要な情報を保存期間と保存条件を定義して保存（バックアップ）できること．
2	定期的に保存データに問題が無いか確認できること．
3	保存データの書き換え保護が可能なこと．
4	定期的なリストアテストが実施できること．
5	バックアップの対象として，業務データ，プログラム，各種ログ，運用操作記録等が指定できること．

表 10　バックアップ運用

項番	要件
1	バックアップ及びリストアの運用手順書（バックアップ手順書，リストア手順書）を作成する．
2	バックアップおよびリストアは，作業誤りによる事故防止のため運用手順書に基づき実施する．
3	ログもバックアップ対象とし，そのアーカイブは，日次/月次のバッチ処理にて実施する．
4	外部記憶媒体への保存時は，情報の重要度に応じて暗号化により保護する．
5	運用作業の教育・訓練を実施する．
6	障害に備え，運用手順書に則り，正常にリストアされることを確認する．
	運用手順書は以下の点を考慮して作成する． a）バックアップが必要な情報のレベルを明確化する． b）バックアップ情報の正確で完全な記録及び文書化したデータ復旧手順を作成する． c）バックアップの範囲（例えば，フルバックアップ，差分バックアップ），及びバックアップの頻度は，組織の業務上の要求事項，関連する情報のセキュリティ要求事項，及びその情報の組織の事業継続に対しての重要度を考慮して決定する． d）バックアップ情報に対して，適切なレベルの物理的及び環境的保護を実施する．主事業所で媒体に適用している管理策は，バックアップ情報の保管場所にも適用する． e）バックアップに用いる媒体は，必要になった場合の緊急利用について信頼できることを確認する． f）復旧手順は有効であること，及び回復のための運用手順で定められた時間内に完了できることを点検し確認する． g）バックアップ情報の暗号化は情報の機密性を考慮して決定する．

4.4.2　ディザスタリカバリ対応

東日本大震災のような大規模災害の場合，本番システムに近い場所にデータをバックアップしても無意味である．以下の点に留意する．

1）　個々のシステムにおけるバックアップの取り決めは，事業継続計画の要求事項を満たすことを確実にするために，定めに従って検査することが望ましい．

2）　バックアップすべき情報は，災害による被害から免れるために，十分離れた場所に保管する．

3）　重要なシステムにおけるバックアップの取り決めは，災害に際してシステム全体を復旧させるために必要となる，システム情報，アプリケーション及びデータのすべてを対象とすることが望ましい．

4.5　サービス/システムの監視

データベース障害の未然防止や障害発生時の的確な切り分けおよび復旧のため，以下のようなデータベースに係るシステム，サービス，リソースの監視機能が望まれる．

1）　プロセス，CPU 負荷，メモリ/スワップ使用量，ディスク空き容量

2）　データベース接続可否，接続ユーザセッション数

3）　データベース領域使用率

4）　キャッシュ使用率

5）　テーブルのオープン/ロック状況，スレッド使用状況

6）　レプリケーション時のステータス

4.6　パッチ/ウィルス対策

データベースを含むシステムは定期的に脆弱性をチェックし，必要に応じてパッチを適用したり，ウィルス定義ファイルを更新する[17]．

なお，セキュリティパッチやウィルス定義ファイルの適用およびアプリケーションのアップ

デートは，これを行うことにより実運用環境を変更することになるということを意識する必要がある．実運用環境の変更であるため，ベンダーから提供されたセキュリティパッチ等を適用する前に，検証環境での動作確認が必要である．以下の手順・機能を備え対処することが望まれる．

1) パッチの評価，本番適用への手順を備える．
2) 検証環境で検証した後に本番環境にリリースする．
3) パッチ/ウィルス対策ソフト/定義ファイルの配布，更新，適用状況の確認機能を備える．

4.7 情報の秘匿

データベースのセキュリティを検討する上で，情報の秘匿（暗号化）は情報漏洩に対する根本的な対策であり，適切に実装し運用することで，漏洩リスクの低減あるいは漏洩を防止できる．なお，データベース上に格納される情報に対する参照，更新，挿入，削除といったデータ保護施策はアクセス制御の機能（4.2節）であって本節で述べる暗号化機能の範疇ではない．

情報の秘匿によって防御する脅威は以下のとおりである．

1) データベースを内蔵するハードディスク等の抜き取りによる情報漏洩
2) バックアップメディア盗難による情報漏洩
3) 通信経路の盗聴による情報漏洩

但し，暗号化しても暗号鍵の盗難による情報漏洩や，特権ユーザなど正規の権限を有するユーザの不正行為による情報漏洩の可能性は残る．これらの脅威は暗号化では防止できず，前者に対しては，4.2節で述べた適切なアクセス制御，後者に対しては5.5節に記述の内部統制プロセスにより対処する．

4.7.1 データベースの暗号化

データベースの暗号化の方法には，「HDDの暗号化」，「DBテーブル暗号化」，「DBカラム暗号化」，「バックアップデバイス暗号化」など[13]がある*5．機密性の高い情報資産には，適切な暗号化が求められる．世の中には様々なデータベース暗号化製品があるが，リスクに応じた強度，品質を考慮し，また暗号アルゴリズムについては電子政府推奨暗号化リスト[19]より選択するのが望ましい．なお，暗号化には負荷がかかるのでパフォーマンスには留意のこと（5.4節参照）．

4.7.2 暗号鍵の管理

データベース暗号化の鍵管理は重要である．正当な権限を持つ利用者だけが鍵にアクセスできるよう適切な制御が必要である．また鍵自体のバックアップやその管理，鍵の世代管理も必要である．ライフサイクル全般に亘り，全ての暗号鍵の生成，配布，保管，執行，紛失，更新，廃棄に係わる管理が可能でなければならない．鍵管理の主な要件は以下のとおりである．

1) ソフトウェア側で提供される鍵の保護オプション（パスワード保護など）は必ず利用する．
2) 1)の保護オプションで保護される場合，パスワード盗難が情報漏洩に直結するため，パスワードを媒体に記載してはならない．

3） 保護された暗号鍵は，外部メディアに安全にバックアップを取得し，保管する．

4） 暗号鍵の管理権限については，複数人の管理者で職務分離できることが望ましい．

2012年現在では，専用ハードウェアで暗号鍵を管理するハードウェアセキュリティモジュール（以下HSM）が利用できる．HSMは，FIPS140-2[*6]にて定義されるような耐タンパー性を備えた暗号化管理ハードウェアである．暗号化の実行および暗号鍵のライフサイクル管理を行う．外部からの鍵盗難や改ざんを防御し，鍵管理者に対する職務分離を実現させる機能を有するなど，データベースの暗号化処理および暗号鍵管理に利用され，データベース処理のパフォーマンス向上と併せ，暗号化データのアクセス管理を容易にする設備[13]である．

4. 7. 3　通信の暗号化

データベースと外部システム間，データベースとデータベース間の通信経路は盗聴による情報漏洩の可能性があるため，次に挙げる暗号化が必要である．

1） 統合データベース全体を構成する以下の各種通信経路を暗号化する（図5）．

・利用者～アプリケーション間
・FDBS～アプリケーション間
・FDBS～FDBS管理者間
・FDBS～分散した個別DB間
・分散した個別DB～個別管理者端末間
・分散した個別DB～鍵管理デバイス間

2） 各種通信経路は，標準化された暗号化方式（SSL，TLS，SSH），VPN，またはデータベースのシステム仕様が規定する暗号方式で秘匿する．

3） 通信経路の暗号化/復号のパフォーマンスを向上させるために，例えば，外付け機器（SSLアクセラレータ等）を活用する．

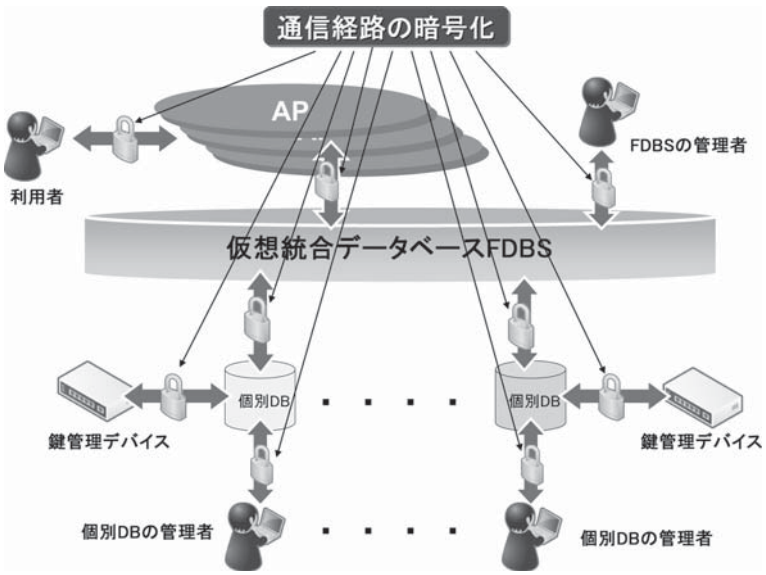

図5　通信経路の暗号化

4.7.4　トークナイゼーション

かつてデータベース暗号化の課題であったパフォーマンス低下や導入/利用時の非透過性等の問題は様々なツールにより克服されてきた．しかしクレジットカード番号を始めとする暗号化された機密情報は，運用される中で業務アプリケーションや運用者に対し未だ復号された状態で利用される．この復号された状態で発生する情報漏洩や，セキュリティ監査に対する負担を軽減する対策として現れたのがトークナイゼーション[10][15]と呼ばれる技術である．トークナイゼーション自体はクレジットカード番号を始めとする機密情報の一部を無作為なデータ（トークン）で置き換えることにより，データを匿名化する技術である．この技術により，クレジットカード決済システムにおいて，クレジット業界のグローバルセキュリティ基準 PCI DSS[11]の要求基準を比較的低コストでクリアすることが可能である．既存のアプリケーションやデータベースにトークナイゼーション機能を容易に組み込む製品もある．

5.　データベース統合に特有なセキュリティ対策

本章では，4 章で述べた一般的なセキュリティ要件に加え，FDBS のような仮想データベース統合に特有なセキュリティ対策について述べる．

5.1　情報漏洩対策（機密性維持）

FDBS からの情報漏洩の危険性はアクセス制御の煩雑さに起因する．分散したデータベースは，それら個々に OS の設定やパスワード管理を行うため，アカウント管理，アクセス制御，ログ管理が複雑になって，システム全体としてみたとき不正アクセスの穴ができやすい．特に分散する DB 上の同一機密レベルの情報へのアクセス権限が異なっている場合，FDBS への低いアクセス権限で権限以上のデータが参照・更新できる場合がある（図 6）．このような場合，FDBS のアクセス制御レベル（アクセス制御の頑強さ）は，各個別 DB の最もアクセス制御レベルが低いものと同一となる．

対策としては，分散する各個別 DB へのアクセス権限と FDBS へのアクセス権限を同一とすることが必要である．

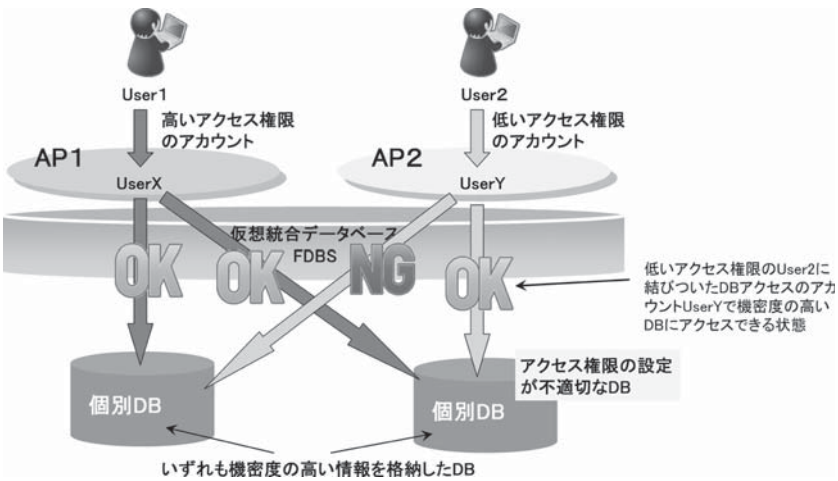

図 6　アクセス制御レベルの低下問題

5. 2　障害対応（可用性維持）

例えば，A，B，C という三つの分散 DB からデータを収集し，マージして利用者に表示するような場合，B が故障でダウンしていると，処理が先に進まないというように，一つの DB の障害が FDBS 全体の動作に支障を来すことがある．特に FDBS のクライアントである重要なアプリケーションがそれほど重要でない DB をアクセスする場合，重要でない DB（＝可用性の低い DB）の障害によって重要なアプリケーションが停止し，可用性の目標を達成できなくなることもある[18]．また，分散した DB のいくつかが冗長化構成を採っていない場合，FDBS 全体としては稼働率が低下し，システム全体としては可用性が低下することになる．

対策にあたり，以下を考慮する．

1）　分散する各個別 DB に対しても冗長化対策を施す．

　データベースの不測の事態に備えるには，バックアップを実施するとともに，冗長化によりシステムを二重化しておくことが有効である．冗長化はコスト増に繋がるが，障害による業務停止の損失（リスク）が大きい企業の基幹システムや決済系システムなどでは必須である．クラスタ化や負荷分散装置などにより冗長化を図り，統合データベース全体としてフォールトトレランシを考慮することが望ましい．

2）　一つの個別 DB が故障していてもアプリケーションが停止しないような考慮をする．

　但し，この場合は，FDBS の出力の信頼性（品質）が低下（情報の一部欠落）することに留意する．なお，重要でない個別 DB の可用性を他の個別 DB と同一のレベルに向上させるか否かは，システム全体の経済比較による．

冗長化には様々なソリューションがあるので，3 章で述べたセキュリティ対策ステップにおけるリスク分析によって導出される具体的信頼性要求基準値の MTBF[*7]，MTTR[*8]，RTO[*9]，RPO[*10] 等を考慮しながらそれらの利用を検討する．

稼働率は MTBF/（MTBF+MTTR）で表され，この稼働率を上げるために一般的に MTTR を小さくする方法が採られる．MTTR を短くするには，デュアルシステム化またはデュプレックスシステム化という二つの方法があるが，前者の方が稼働率は高くなる．

1）　デュアルシステム化

　常時 2 系列並列で互いに監視し合いながら稼働し，一方が故障した場合にも全タスクを他方のみで引き受け，システム運用の続行が可能なシステム．片方に障害が生じた際も，もう片方で処理を続行しながら復旧にあたることができる．

2）　デュプレックスシステム化

　システムを 2 系統用意して，普段は片方で処理を行い，もう片方は障害発生に備えて待機させておく方式．平常時に処理を行うシステムを主系または本番系などと呼び，主系に障害が生じたとき肩代わりするシステムを従系または待機系，予備系などと呼ぶ．待機や切り替えの仕方により，ホットスタンバイ，ウォームスタンバイ，コールドスタンバイなどの種類がある．信頼性はデュアルシステムよりも劣るが，比較的安価に実現でき，銀行のオンラインシステムなどで採用されている．

5. 3　データ品質維持（正確性，完全性，正当性維持）

分散する DB 上のデータ品質にばらつきがある場合，仮想統合して表示するデータ全体の品

質は，元データの最も低いレベルの品質となることは当然である．また統合表示を行う際，クレンジング不足によるデータの重複表示も品質低下の要因である．対策として，以下が必要である．

1) 分散する各個別 DB のデータ品質を統一する．
2) 同一データは適切にクレンジングする．

なお，金融商品取引法（以下 J-SOX 法と称する）に基づく財務報告の信頼性を確保するための IT の統制として，以下の要件があることに留意する[5]．

1) 正当性：取引が組織の意思・意図にそって承認され，行われること
2) 完全性：記録した取引に漏れ，重複がないこと
3) 正確性：発生した取引が財務や科目分類などの主要なデータ項目に正しく記録されること

5. 4　パフォーマンス（可用性維持）

FDBS は分散する複数の既存データベースを統合するため，利用者から見て応答速度の劣化が課題となる．そのため可用性の観点からも負荷分散など以下の点に留意する．

1) ネットワークのボトルネック対応

FDBS のような仮想データベース統合の場合，クエリはネットワークを経由して分散した各データベースに送信され，それらのレスポンスを統合して最終出力とするため，単一のデータベースへのクエリに比べネットワークの遅延が加算される．従って，ネットワークの帯域はある程度大きく確保しないと，レスポンスタイムにおいて実用性に影響がでる．

2) 暗号化の負荷

暗号化はアプリケーションで行うか基盤で行うかの検討が必要である．特に，遵守しなければならない基準やポリシーにおいて，アプリケーションでの暗号化を求めている場合，セキュア・プログラミングガイド等に反映し，確実に暗号化が行われるようにしなければならない．基盤で実施する場合は，ハードウェアやソフトウェアの選択時に，システムが求めるパフォーマンスを保証できるかを検証する必要がある．データベース暗号化アプライアンス製品の利用もパフォーマンス向上に有効である．

3) ログ採取の負荷

対象とするログの種類，ログ採取レベル等によりパフォーマンスに大きな影響をおよぼすため，そのログの必要性を考慮し，採取するログやそのレベル（詳細度）を決定する必要がある．また 4. 3. 3 項で述べたように，ログのアーカイブや保管場所等の運用を考慮した収集方法も合わせて検討する．

4) データベース監視の負荷

データベースの操作に係る監視はデータベース自体に負荷を及ぼす．これを回避するために，ネットワーク上を流れる SQL パケットをキャプチャし，何時，誰が，何処から，どのアプリケーションを使って，どの SQL 文を実行し，どのような結果情報を取得したか等を監視・記録できるアプライアンス製品がある．これはデータベース自体になんら負荷を与えることがないので，パフォーマンス低下防止施策の一つとして有効である．

5. 5　内部統制維持（会社法，J-SOX 法対応）

企業，特に上場会社においては法制上，財務報告に係る虚偽表示リスクを一定限度に抑える統制が必要である．特に FDBS のような仮想統合データベースの出力を財務報告に使用する場合，分散する全ての個別 DB も内部統制要件を満足する必要がある[5][6][17]．

4 章でデータベースの一般的なセキュリティ要件を述べたが，セキュリティ対策全般に亘って忘れてはならないのが悪意の内部犯罪や過失等への対応である．これらは，J-SOX 法に基づく監査等においても，監査人から指摘され易いポイントとなっている．「2010 年 情報セキュリティインシデントに関する調査報告書～個人情報漏えい編～」[20]によると，個人情報の漏えい原因比率（漏えい人数比率）は，「管理ミス」22.3%，「盗難」10.2%，「内部犯罪・内部不正行為」8.4%，「不正な情報持ち出し」6.3%と，過失や内部犯罪が依然大きな比率を占めている．対策として考慮すべきポイントを以下に述べる．

1）　特権を有するシステム利用者（管理者 ID）の統制を厳密にする

システム利用者の中でも特権を有するシステム管理者等は，あらゆるファイルを閲覧，変更することが可能であり，その結果，特権を有する一人の人間が悪意を持てば，正規の権限での不正行為が実行可能である．また，操作ログ等をも削除できるので，不正行為の発見も困難となる．悪意がなくても，不注意でデータを変更してしまう可能性もある．従って，特権を有するシステム利用者に対しては表 11 の統制が必要である．

表 11　特権を有するシステム利用者（管理者 ID）の統制

項番	要件
1	重要な処理における権限を分離（職権分離）する．例えば，ユーザ ID の権限付与承認者と登録実施者を分離し，それぞれの職権毎に異なるユーザ IDを付与する．
2	特権的アクセス権限の設定は，責任者の承認に基づいて設定し，付与の理由および権限者は適切かを確認する．
3	特権的アクセス権限の必要性の見直しを定期的および人事異動や組織改編時に行い，不要な管理者IDは確実に削除する．
4	特権的アクセス権限の使用状況は厳格に管理する．例えば，ログ等（4.3節参照）により定期監査を実施したり，管理者IDの共用を禁止する．

2）　開発と本番運用の分離

どんなにシステムのセキュリティ対策を万全に行っても，不正なプログラムや誤ったデータを容易に本番環境に載せることができるのであれば，全てのセキュリティ対策は水泡に帰する．そこで，プログラムやデータの修正，OS，データベース等に対する各種パッチの本番リリースにあたっては表 12 を考慮する．

表 12　開発と本番運用の分離

項番	要件
1	間違った，あるいは不正なプログラムやデータが責任者の承認なしに本番環境にリリースされないこと．
2	変更管理プロセスにおいて内部の意図的な不正または意図的なプロセスの迂回等を検知できること．例えば，データベースのダイレクト修正作業など．
3	本番データをテストに使用する際は，4.7.4項で述べたトークナイゼーション技術の利用やデータにマスキングを施すなどして意図しない情報漏洩に備えること．

3)　誤った行為あるいは不正行為を検知，追跡できるような管理プロセスを備える

企業システムに係る処理が適切に実行されているかをモニタリングすることは，不正行為の検知のみならず予防，抑制にも繋がる．また，デジタルフォレンジックを正しく行うためにも必要である．備えるべき管理プロセスを表 13 に挙げる．

表 13　誤った行為あるいは不正行為を検知，追跡できる管理プロセスの整備

項番	要件
1	ログシート，処理報告，日誌等の作業記録は管理者の検証を受け，一定期間保管する．
2	監視ログを分析，レビューし，報告する運用手順を確立する．特に注意すべき監視項目の例を以下に示す． ・失敗アクセス ・長期間使用されていないユーザID の再使用 ・特権IDの割当てと使用状況 ・外部からの不正なネットワークアクセス等
3	以下のようなログの保全（不正消去，改ざん防止）機能を備える． ・ログファイル，作業記録へのアクセス制御 ・追記・更新不可な記録媒体への記録等

4)　外部委託先の管理

分散する個別 DB の構築・運用の一部を第三者の外部業者に委託している場合，そこでのセキュリティ事故の管理責任は委託元企業にある[4][6][9]．また委託先のシステム運用は内部統制の評価の範囲にも含まれる[5]．従って，委託元は委託先に委託した業務システムのセキュリティ管理の整備および運用状況を把握し，また適切に評価しなければならず，表 14 の対策が必要である．

表 14　外部委託先の管理

項番	要件
1	委託業務につき，サービスレベルを定義し，委託先企業とサービスレベル契約（SLA）を結ぶ[7]．
2	委託先から業務報告を定期的に受け，それらを検証する．
3	サービスレベルが維持されていることを確認するため，外部の専門機関による外部委託先の監査結果を確認する．
4	必要に応じ委託先へ立ち入り検査を行う．

5. 6　統合初期段階の留意点

非機能要件であるセキュリティ対策は，利便性の低下，開発費用の増大，運用効率の低下などの弊害を招くことが多く，データベース統合プロジェクトにおいてもセキュリティ対策の実施が後回しにされることも多いと想定される．その結果，統合作業の後半，例えば詳細設計以降において明らかになったセキュリティ要件への対応のために，基本設計のやり直し等の手戻りが発生することも予想される．

このような事態を避けるためには，マイナス志向に陥りやすいセキュリティ対策検討の際，最終的にどのようなシステム構成になったとしても，3 章に述べた基本的なセキュリティ対策のステップを踏まえ，データベース統合の初期段階において，予め必要なアクセス制御要件や内部統制要件を可能な限り盛り込んでおくことが肝要である．

6. お わ り に

一般的にIT事故発生のリスクが不明確な場合，適切なセキュリティ対策投資の判断が困難な状況も多々あるかと思われる．しかし想定外の事故は発生するものである．

2010年頃から，ビッグデータを扱うクラウドコンピューティングと親和性が高いキーバリュー型データストアが注目されている．その例としてGoogle検索サービスに使用されているBigtableやAmazonクラウドのSimpleDBがある．これらは膨大な量のコンテンツやインデックスを高速に検索するためのデータストアとして開発された，いわゆる非SQL系のキーバリューストレージである．リレーショナルデータベース（Relational Database，RDB）と異なり，データの検索やテーブルの結合ができないが，高度なレプリケーション機能やディザスタリカバリ機能を備えるなど高い可用性を持ち，GoogleやAmazon以外にも多数のプロジェクトで使用されている．しかしその実態は，分散配置されたストレージサーバ群に過ぎないので，これらに対しては，冗長化する，厳密なアクセス制御をする，機密情報を秘匿（通信，ストレージ）するなど，FDBSと同様な考慮が必要である．

本稿ではデータベースのセキュリティを網羅的に俯瞰したので，実システム構築時においてはリスク分析結果に応じて本稿の内容を適宜利用できる．特に情報漏洩にセンシティブなシステムの開発・運用における要求定義や基本設計の際の参考としていただければ幸いである．

最後に本稿の執筆において協力を頂いた関係各位に感謝の意を表する．

＊1 XSSは，Webアプリケーションの脆弱性を悪用して利用者を狙う攻撃手法の一つである．Webサイトを閲覧した利用者のブラウザ内で悪意のあるスクリプトが実行され，偽情報の表示や，情報の漏えい，セッションの乗っ取り，フィッシング詐欺等の被害を引き起こす代表的な攻撃手法である．

＊2 One Time Password（ワンタイムパスワード）の略．一回限り有効なパスワード認証システムのことでパスワードの盗聴対策に用いられる．

＊3 Challenge Handshake Authentication Protocolの略（チャップ）．クライアントがサーバからのチャレンジに対するレスポンスを生成し，それをパスワードとしてサーバに返信する認証プロトコルのことで，パスワードの盗聴対策として用いられる．

＊4 フィッシング詐欺手法の高度化により，二要素認証をも無力化する中間者攻撃の被害が拡大している．このような攻撃への対策技術として日本ユニシスが開発した特許技術[21]があるが，インターネットに接続するシステムにおいてはこのような対策を考慮することも必要である．

＊5 日本ユニシスが研究・開発したPGFDBSにおいては，DBアクセスの透過性維持のためPGFDBS自体では暗号化に関与せず，データベース暗号化は個別DB毎に考慮することとしている．

＊6 NIST（National Institute of Standards and Technology：米国標準技術局）が制定した，暗号モジュールに関するセキュリティ要件の仕様を規定する米国連邦標準規格（Federal Information Processing Standard）．

＊7 Mean Time Between Failure（平均故障間隔）の略．

＊8 Mean Time To Repair（平均修理時間）の略．

＊9 Recovery Time Objective（目標復旧時間）の略で，障害発生からどのくらいの時間でデータを復旧できるかという目標を表す．

＊10 Recovery Point Objective（目標復旧地点）の略で，障害発生からどのくらい古いデータを復旧するかというデータ損失の最大許容範囲を表し，災害復旧の際の重要な指標となる．

参考文献 ［1］ 中山 陽太郎,「PostageSQLを活用した仮想データ統合基盤の実現」, ユニシス技報, 日本ユニシス，通巻111号，2012年3月

［2］「JIS Q 27001：情報技術―セキュリティ技術―情報セキュリティマネジメントシステム―要求事項」，日本規格協会，2006年5月

［3］「JIS Q 27002：情報技術―セキュリティ技術―情報セキュリティマネジメントの実

践のための規範」，日本規格協会，2006 年 5 月
［4］「情報セキュリティ管理基準」，経済産業省，2003 年 4 月
［5］「財務報告に係る内部統制の評価および監査の基準並びに財務報告に係る内部統制の評価および監査に関する実施基準の設定について（意見書）」，金融庁企業会計審議会，2007 年 2 月
［6］「システム管理基準 追補版（財務報告にかかる IT 統制ガイダンス）」，経済産業省，2007 年 3 月
［7］「SaaS 向け SLA ガイドライン」，経済産業省，第 1 版，2008 年 1 月
［8］「金融機関等コンピュータシステムの安全対策基準・解説書」，金融情報システムセンター，第 8 版，2011 年 3 月
［9］「金融機関等のシステム監査指針」，金融情報システムセンター，第 3 版，2007 年 3 月
［10］「Payment Card Industry Data Security Standard (PCI DSS) Ver2.0 Information Supplement: PCI DSS Tokenization Guidelines」，Payment Card Industry Security Standards Council，2011 年 8 月
［11］「Payment Card Industry (PCI) Data Security Standard」，Payment Card Industry Security Standards Council，Ver2.0，2010 年 10 月
［12］「データベースセキュリティガイドライン」，データベース・セキュリティ・コンソーシアム，第 2.0 版，2009 年 2 月
［13］「データベース暗号化ガイドライン」，データベース・セキュリティ・コンソーシアム DB 暗号化 WG，第 1.0 版，2011 年 11 月
［14］「統合ログ管理サービスガイドライン」，データベース・セキュリティ・コンソーシアム統合ログ WG，第 1.0 版，2010 年 12 月
［15］「緊急提言：オンラインサービスにおけるデータベースと機密情報の保護」，データベース・セキュリティ・コンソーシアム，2011 年 6 月
［16］「ソフトウェア等の脆弱性関連情報に関する届出状況［2011 年第 4 四半期（10 月～12 月）］」，独立行政法人情報処理推進機構，2012 年 1 月
［17］ 八津川　直伸，石野　貴子，「重大な脅威に対するセキュリティ設計手法の考察」，ユニシス技報，日本ユニシス，Vol.28 No.3，通巻 98 号，2008 年 11 月
［18］ 小田　圭二，「44 のアンチパターンに学ぶ DB システム」，翔泳社，2009 年 11 月
［19］「電子政府推奨暗号リスト」，総務省，経済産業省，2003 年 2 月
［20］「2010 年情報セキュリティインシデントに関する調査報告～個人情報漏えい編～」，NPO 日本ネットワークセキュリティ協会，第 1.4 版，2011 年 8 月
［21］ 八津川　直伸，信岡　弘光，「アクセス制御システム，認証サーバシステムおよびアクセス制御プログラム」，特許公報“特許第 4698751 号”，日本国特許庁，2011 年 3 月 11 日

執筆者紹介　八津川　直伸（Naonobu Yatsukawa）

1980 年福井大学工学部電子工学科卒業．同年日本電信電話公社入社，専用回線遠隔試験システムの施設設計，日本縦貫光ファイバー伝送システムの導入計画等を担当．1985 年日本ユニシス株式会社（当時，日本ユニバック株式会社）入社．電気通信事業の企画推進，汎用機の通信制御装置ソフトウェア（DCP/TELCON シリーズ）の開発および保守，セキュリティ商品の開発および保守，法務実務，J-SOX 対応支援，システム構築支援等を経て，ネット取引の安心・安全を実現するセキュリティの研究に従事．現在，総合技術研究所インキュベーションラボに所属．CISSP.