5B-2

マルチテナント型アプリケーションの DB データ暗号方式の研究

○及川　和彦†　　佐藤　雅之†　　山足　光義†

三菱電機株式会社　情報技術総合研究所†

1. はじめに

近年、クラウドサービスを提供するシステムとして、1 つのシステムの中に複数の企業をテナントとして同居させ、リソースや運用コストを大幅に低減するマルチテナント型システムが利用されている。マルチテナント型システムでは、複数のテナントが互いのデータにアクセスできないようにするアクセス制御や、万が一システムに侵入された場合においても、データを読めないように暗号化して守るセキュリティ対策が重要である。

マルチテナント型システムとしては、各テナントのデータの分離の仕方により表 1 に示すように大別され、セキュリティ対策の方式はそれぞれ異なるもとになる。

表 1：マルチテナント型のデータ管理方式

管理方式	説明
個別データベース方式	テナント毎に専用の DB を用意して別々に固有データを管理する
個別スキーマ方式	テナント毎に専用のスキーマを用意してスキーマ別に固有データを管理する。
共通スキーマ方式	共通のスキーマを用意して、表に各テナントの識別するテナント ID 列を持たせ、各テナントのデータを同一表で管理する。

③の方式では、図 1 に示すとおり、各テナントのデータが同一テーブルに格納される形態であり、多量のデータを扱うクラウドサービスにおいて効率よくディスクを扱うことができるが、反面、万が一、システムへの侵入があった場合、全てのテナント情報が見えてしまうリスクがある。

テナントユーザ共通属性情報

テナントID	ユーザID	名前	…
A	10001	○○	…
A	10002	△△	…
B	10001	□□	…
B	10002	▽▽	…
C	10001	× ×	…
C	10002	○○	…

テナントユーザ個別属性情報

テナントID	ユーザID	属性ID	属性値
A	10001	1	03-xxxx-xxxx
A	10001	2	a01@aaa.xxx
A	10001	3	2013/03/31
A	10002	1	04-xxx-xxxx
A	10002	4	090-xxxx-xxxx
B	10001	1	06-xxxx-xxxx
B	10001	2	b01@bbb.xxx
B	10001	3	2014/9/1

図 1：共通スキーマ方式でのテーブル格納例

Research on the RDB storing data encryption system of multi-tenant type application

†Kazuhiko Oikawa, Masayuki Sato, Mitsuyoshi Yamatari, Information Technology R&D Center, Mitsubishi Electric Corporation

本論文では、共通スキーマ方式の利点を残したまま、既存のマルチテナント型アプリケーションで扱うデータを暗号化し、セキュリティを確保するマルチテナント型アプリケーションのデータ暗号化の一方式を提案する。

2. 課題と解決方式

既存のマルチテナント型アプリケーションで扱っていた平文データを暗号化する場合は、以下の課題が発生する。

課題 1：暗号化されたデータは一般的にバイナリデータとなるが、バイナリデータのままでは、SQL 文の WHERE 句で扱うことができない。

課題 2：SQL では部分一致検索として、WHERE 句の LIKE 演算子を利用できるが、データが暗号化された場合、部分一致検索ができない。

上記課題を解決するためテナント毎に暗号化鍵を管理し、1 文字単位でデータの暗号化を行うマルチテナント対応データ暗号化方式（図 2）を検討した。

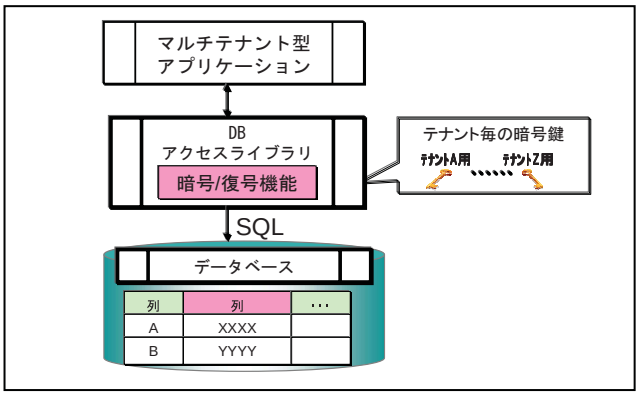

図 2：マルチテナント対応データ暗号化方式

本方式により課題 1 は、暗号化後のデータは BASE64 にエンコードして、文字列型として DB に格納することで対応する。

また、課題 2 は、部分一致検索で利用される暗号化対象項目データは、文字単位で分割し、個々に文字を暗号化・エンコード処理し、それらを連結した文字列データを DB に格納することで対応する。

前述の方式の動作は、既存のマルチテナント型アプリケーションが、暗号/復号機能が追加された DB アクセスライブラリに既存の SQL でアクセスさせる。DB アクセスライブラリ中では、共通スキーマ方式の特徴であるテナントを識別するテナント ID とテナント毎の暗号化/複合化鍵を紐付けて保持する。

DB 上に格納するデータは、DB アクセスライブラリにより文字分割・暗号・エンコードされた状態で DB 保管される。

3. 方式検証

上記、マルチテナント対応データ暗号化方式により、マルチテナント型アプリケーションで扱うデータを平文データから暗号化されたデータに変更したことによる影響を、性能とディスク使用量の観点から検証を行った。

検証を行う上で想定する適用対象のマルチテナント型システムとして、テナント企業の社員情報や ID カード情報を管理する ID 管理サービスを取り上げ、DB アクセスライブラリソフトウェア実装する前に、前述の観点で机上検証を実施した。

［性能検証］

性能検証ではユーザ情報の暗号化対象属性を氏名、カード情報の暗号化対象属性を管理番号とし、データ 1 件当たりの処理性能の机上評価を行った結果を表 2 に示す。

表 2：1 ユーザ分の暗号化処理性能(机上)

机上測定	暗号復号 (AES128)	エンコード (BASE64)	処理合計	備考
1 人のユーザ情報の氏名の文字毎暗号化処理	2μs	4μs	6μs	平均 4 文字の氏名を 1 文字ずつ暗号/復号処理実施
1 人の ID カードの管理番号暗号化処理	1μs	2μs	3μs	16 桁カード管理番号を想定

（AES128 バイト（1 ブロック）処理性能=0.5μs として算出）

評価対象の ID 管理サービスでは、1 テナント当たり 5,000 人までのユーザをサポートしており、サービスレベルとして、各テナントが利用する画面での応答性能は 3 秒以内が要求される。

サポートする 1 テナント当たり最大 5,000 件のユーザ情報、およびカード情報を画面から CSV で一括登録した場合でも、氏名、カードの管理番号の暗号化にかかる処理時間はそれぞれ 3.0 ms、1.5ms となり、机上評価の段階では、サービスレベルに影響を及ぼさないことを確認した。

［ディスク使用量検証］

ディスク使用量の観点として、同じく氏名、カード管理番号について、暗号化によるデータ 1 件当たりのデータ増加量を表 3 に示す。

表 3：1 ユーザ分のデータ増加量(机上)

暗号化対象属性	文字数(想定)	①暗号化前バイト数	②暗号化後バイト数	③エンコード後バイト数	④暗号化による増加バイト数(③-①)
氏名	4 文字	12	64	88	76
カード管理番号	16 文字	16	32	44	28
	1 ユーザ当たり増加バイト数合計				104

評価対象の ID 管理サービスでは、1 サーバ当たりの全テナント含めた登録可能なユーザ数として 10 万人を予定している。氏名やカード管理番号の属性を暗号化した場合でも、データ増加は 10.4M バイトになり、机上評価の段階では、ディスク使用量の観点でも、サービスに影響を及ぼさないことを確認した。

4. おわりに

本報告では、共通スキーマ方式のマルチテナント型アプリケーションを構築し、DB 格納データを暗号化する方式を検討し、性能と、ディスク容量にから検証を行った。本方式では、特定の RDBMS 製品に依存しないで実現することができるため、今後は実アプリケーションへの適用検証により性能面での検証を行う予定である。

参考文献

[1] 小杉,他,「マルチテナント対応データ切替制御に関する研究」, 2011, 第 74 回情報処理学会全国大会論文集,4H-5

[2] 小杉,他,「マルチテナント型テナントアクセス制御方式に関する研究」, 2012, 第 75 回情報処理学会全国大会論文集,3H-1