なぜ XP_CRYPT & SQL Shield を選ぶのか？

プロジェクトマネージャーの見解

パート I：必要性の定義。SQL サーバの防衛がかけている部分。
　フィールドレベルデータの保護
　保存プロシージャやスクリプトの保護

パート II：コードソリューション―SQL コードを確実に保護する方法
　スクリプトの保護
パート III：データソリューション―SQL データの保護を強化する方法
パート IV：ケースの作成―購入 vs 構築

パート I 必要性の定義。SQL サーバの防衛がかけている部分。

フィールドレベルデータの保護

以外にも、SQL サーバではフィールドレベルでデータは暗号化されない。データベースにログインした時点でデータへのアクセスが得られる。しかし、現実的にもしファイルシステムにアクセスがあれば、データベースファイルを簡単にコピーすることができる。つまり、アクセス者がシステムアドミニストレーターに認証されている SQL サーバにペーストすることができ、データへのアクセスを完了することができるのである。

つまり、SQL サーバのセキュリティは実際的に、ファイルシステムに入り込まれない限り、強力であるということになる。これは、エクスプロイットやハッカーが溢れている現状で、特に重要機密データ（例えばクレジットカード番号や健康情報など）の保護に責任をもっているものとして多大な信頼であるといえる。

保存プロシージャとスクリプトの保護

開発者は SQL サーバーで、データベースにロジックをコードすることができる。このロジックは保存プロシージャ、トリガーおよびユーザー定義機能として保存される。このロジックを暗号化するには 2 つの理由がある。

一つ目は、知有権問題である。仮にスクリプトのロジックを見ることができるとすれば、それは、ソースコードをみることができるのと同じである。つまり、プ

ロジェクトの極秘内部動向を知られることとなり、リバースエンジニアリングをより簡単にすることになるのである。

二つ目は、保存プロシージャを見られた場合、編集されることは簡単である。つまり、保存プロシージャを書き換えられ、データベースに影響を及ぼす特別なロジックを挿入されることがある。どういうことかというと、データの削除、データベースの破壊、もしくはさらに深刻な場合には盗まれる可能性があるということである。つまり、特別なトークン信号が「提示」されると、極秘医療データが書換えもしくは読み出しされたり、おそらくｅーコマースなどで、商品購入がある度にデビット利用するよりむしろ、特定人物の口座に現金振込みが行われたりするようなことが起きるのである。

多くの開発者は多数のアイテムについて、データベースフィールドレベルで暗号化されるべきであると主張するであろう。これらのアイテムにはクレジッドカード番号、社会保障番号、その他医療情報などの個人データが含まれる。

恐ろしいように聞こえるかもしれないが、これら 2 つの問題には幸運にも解決法がある。

パート II：コードソリューション―SQL コードを確実に保護する方法

スクリプトの保護

SQL サーバーのヘルプファイルを開けるとすぐに、SQL サーバーで保存プロシージャおよびスクリプトの暗号化ができることがわかる。しかし安心する前に、ヘルプファイルには載っていないが、5 分あればウェブを探して、いくつもある「マイクロソフト暗号化」された保存プロシージャの暗号解読プログラムのうちのひとつを無料でダウンロードできる。つまり、SQL サーバーのネイティブ暗号を用いて暗号化したにもかかわらず、多少の経験があるどんなハッカーでも SQL コードに入り込み、やりたい放題することができるということである。

どうすればよいのか？幸運にも SQL Shield では、暗号解読のハッキングプログラムがない（周知の範囲で）保存プロシージャの暗号化を行うことができる。つまり、ハッカーに暗号化されたスクリプトを見られた場合、ハッカーが出回っているハッキングツールキットを何度用いたとしても、SQL スクリプトコードを解読することはできない。つまり、安全である。

これらを踏まえて、データの保護は非常に重大な問題となってきている。

パート III：データソリューション—SQL データ保護の強化

これらはデータを暗号化するのに使用できる多数の異なるアルゴリズムである。XP_CRYPT は RSA(非対称アルゴリズム)、AES、トリプル DES,DESX および RC4（対称アルゴリズム）を含む。必要に応じてアルゴリズムを選ぶことができる。

しかし、対称アルゴリズムに比べて非対称アルゴリズムによる暗号化は比較的時間がかかることを記しておく。

暗号化データフィールドは、簡単にデータベースにコードルーティンを挿入するプログラムである XP_CRYPT GUI を使う XP_CRYPT を用いては簡単である。XP_CRYPT GUI は大多数の仕事を自動化する。すべてのインターフェースを加え、コードをサポートし、データベースに適用する。それぞれ特有のキーを用いて、簡単に複数のアルゴリズムを加えることができる。より強力なアルゴリズムからの追加オーバーヘッドを考えると、フィールドの長さ、タイプおよびセキュリティーの重要度に応じて異なるフィールドについて、異なるタイプの暗号を適用することが賢明であるかもしれない。

5 分以内に、データベースにあるフィールドを暗号化することができる。プログラムでは、以下のことが実行できる。

1）パスワードの管理のため、テーブルの作成
2）自身で選択する暗号化されたフィールド表示の作成
（クリアーテキストは自身で削除する）
3）暗号化したフィールドを解読するビューの作成
4）ユーザーセッションのために暗号解読キーを有効にし、
記録をとる手順

下に示すイメージでは、データベースにフィールドレベルでの暗号化を加えるときに使用される画面を表示している。

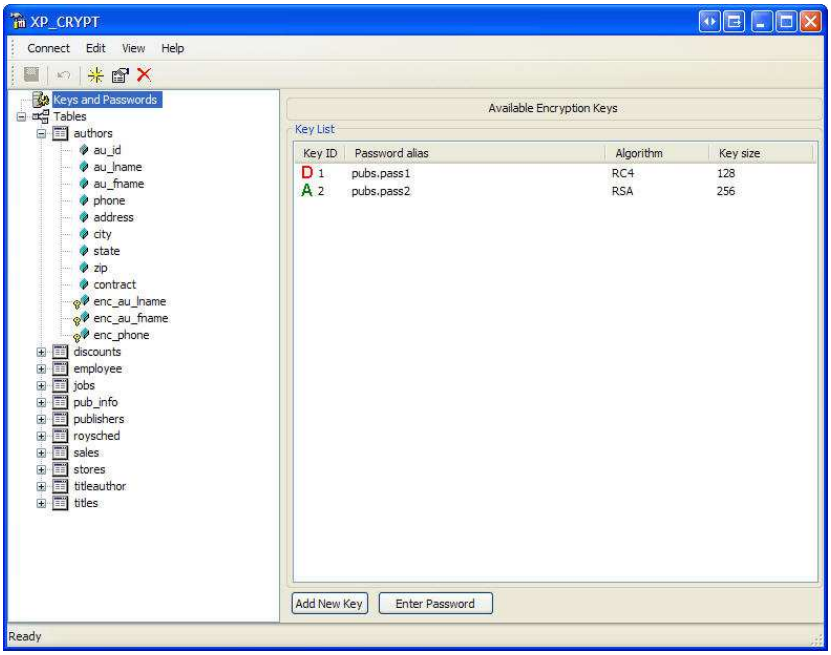

図 1—ここでは、XP_CRYPT GUI を用いてデータベースに複数タイプの暗号を追加する。

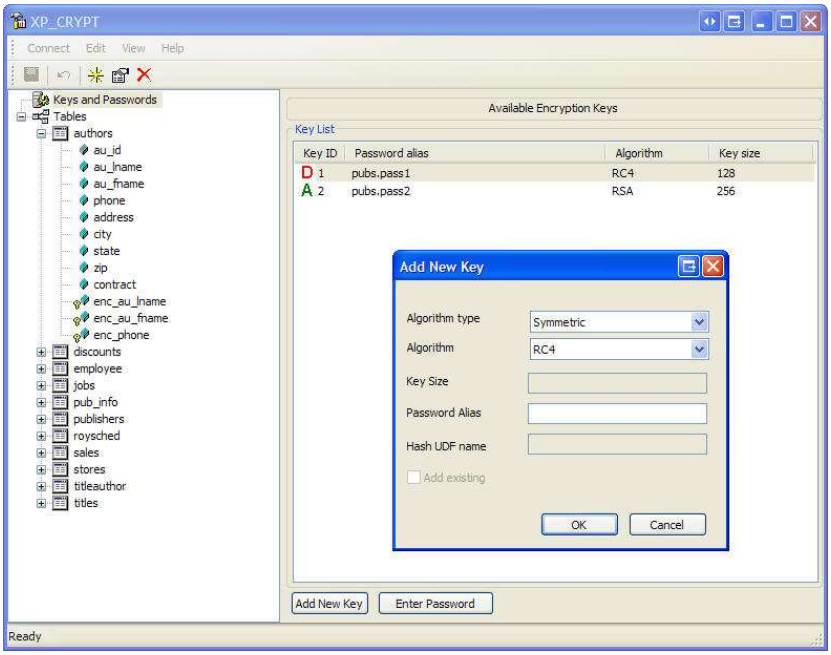

図 2—　ここでは既定の暗号キーの属性を設定する。対称、非対称暗号を含む異なるアルゴリズムがサポートされている。

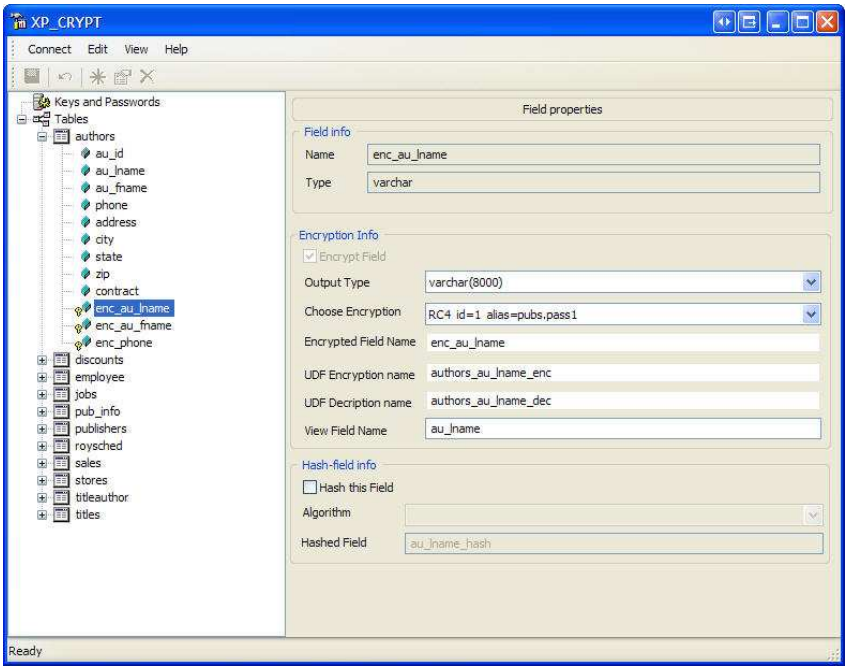

図 3—ここでは、フィールドに暗号化アルゴリズムを適用し、出力タイプ、暗号タイプおよび XP_CRYPT GUI が作成するプロシージャのフィールド名を選択できる。

パート IV：ケースの作成—購入 VS 構築

ツールキットの評価や、購入するか構築するかどうかの判断をする必要がある時がある。

1) 拡張された保存プロシージャには一切バグがいないことが不可欠である。状態の悪い保存プロシージャをしばしばデータベースから呼び出すと、データベースが凍ったり、崩壊する可能性がある。XP_CRYPT が市場商品であり、多数のユーザーがいることは、このリスクを軽減する。

2) 自身でソリューションを構築する利点は、ソースコードを得られることである。幸運にも、ソースコードのあるヴァージョンの XP_CRYPT を購入することができる。

3) 時は金なり。複数のアルゴリズムをサポートし、よくテストされたものを自身で構築するには、最終的に高くつく。加えて、XP_CRYPT GUI を用いればすべてが速く進む。

4) データにはいくらの価値があるのか？もしくは、データが完全体でなくなるまたは、暗号解読できない場合に起こる最悪の出来事はなにか？そのような問題はバグが発生したときに起こりうる。証明されたソリューションに何

千ドルを費やして安心を得ることは、保存プロシージャもしくは自身で書いた暗号コードにバグがあるか心配するより、いくらかよいものである。
5）これは、証明されたソリューションである。XP_CRYPT はすでに国政府機関、金融会社、医療機関や世界中の大学施設で使用されている。
6）ライセンス許可―XP_CRYPT では複数の使用許可オプションがある。XP_CRYPT の金額提示は競合価格のごく一部であるので、非常に妥当な金額である。ライセンスオプションに含まれるのは：

a. シングルライセンスはひとつのサーバー用
b. サイトライセンスを購入すると、ひとつの会社内すべてのサーバーがカバーされる。
c. 再配布可能ライセンスは保護したいソリューションを再販売するソフトウェア製造者向けで、ソフトウェア成果物に製造者の技術を組み込むことができる。

自身のソリューションを書くことの是非について考える時すぐに、ソースコード付属の手頃なソリューションが最良の手段であることがわかる。幸運にも XP_CRYPT は素晴らしい解決策であり、手頃な価格で販売されている。