鍵のライフサイクル管理を行う鍵管理プラットフォームの設計と評価

4T－1

田淵 洋介　植田 広樹　関野 公彦
NTT 情報流通プラットフォーム研究所
e-mail: {tabuchi,ueda,sekino}@dsa.isl.ntt.co.jp

1. はじめに

暗号技術を利用したセキュリティシステムの利用機会が増加するに従い，個人が扱わなければならない暗号鍵の数が増加する．安全性の面から，長期間利用しない暗号鍵は使用を一時的に停止させたり，一定期間が過ぎれば別の暗号鍵に更新することが必要である．このような暗号鍵の管理に関する統一的な基盤として ISO/IEC11770-1 では「鍵のライフサイクル管理」という概念が提案されている[1]が，この概念を実装した鍵管理プラットフォームはない[2][3]．

本稿では，ISO/IEC11770-1 で規定されている鍵のライフサイクル管理を実装した鍵管理プラットフォームについて述べる．

2. 求められる要件

鍵管理プラットフォームとは，暗号処理，鍵の保管，鍵情報の管理等を行うためのモジュールとそのインタフェースを提供するものであり，以下の要件を満たす必要がある．

- 一定期間利用した鍵，漏洩した可能性が有る鍵は不正使用を防ぐために利用可能な暗号処理を制限できること(ライフサイクル管理)．
- 電子商取引等，多分野での利用を考慮し，標準的なインタフェースの提供及びアプリケーションのインタオペラビリティが確保できること．
- 同じ鍵を長期間利用する等システム全体の安全性を下げる行為を防ぐため，利用する鍵の有効期限，用途等の鍵情報を集中管理できること．
- モバイル端末など利用者の環境によって暗号処理に充分なリソースが使えない場合があるため，利用者のマシン環境に依存せず，一定以上の性能が保証できること．

3. 設計方針

鍵管理プラットフォームの標準インタフェースとして The Open Group は CDSA(Common Data Security Architecture)と呼ばれるセキュリティアーキテクチャを提案している[4]が，CDSA には鍵のライフサイクル管理に関する規定がない．一方，ISO/IEC11770-1 では鍵のライフサイクル管理の枠組みが規定されているだけで，鍵の状態管理を実現するための実装方法に関する規定はない．本研究では，CDSA 準拠のアーキテクチャを保ちながら，鍵のライフサイクル管理を行う鍵管理プラットフォームを実現する．

4. ISO/IEC 11770-1

ISO/IEC11770-1 では暗号化、ディジタル署名、改竄防止，否認防止等の様々な用途に用いる鍵を管理するための枠組み、鍵の生成・活性化・失活性化・再活性化・削除等を含む鍵のライフサイクル管理、鍵の配布と保管、鍵の証明等の機能について規定している．

このうち，鍵のライフサイクル管理では，鍵は3つの状態を持ち，各状態によって利用範囲が異なることを規定している(図1)．

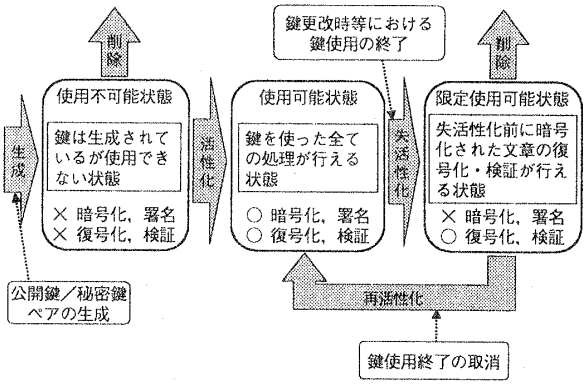

図1　鍵のライフサイクル

Design and evaluation of key management platform for key lifecycle management
Yosuke TABUCHI, Hiroki UEDA,
Kimihiko SEKINO
NTT Information Sharing Platform Laboratories

5. CDSA(Common Data Security Architecture)

The Open Group が定めた CDSA のアーキテクチャを図2に示す．CDSA では，セキュリティシステムの構築に必要な，暗号サービス，安全なデータ保管，証明証関連の処理，キーリカバリ等についての共通インタフェース(CSSM API)を定め，個々のモジュールを動的に利用できる仕様となっている．尚，CDSA は既に幾つかの OS(HP-UX11.0, AIX 等)やアプリケーションに採用されており，現在も The Open Group によって拡張が行われている．

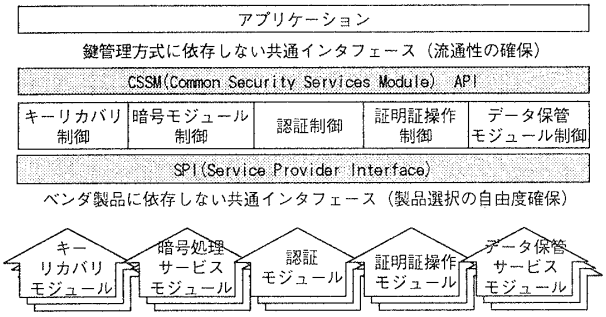

図2 CDSAアーキテクチャ

6. 実装

ライフサイクル管理機能

鍵のライフサイクル管理を行うためには，鍵の属性情報として「鍵の状態」を持たなければならない．そのため，鍵の状態等の属性情報を格納する領域「拡張鍵ヘッダ」を独自に規定した．暗号処理を行う際には拡張鍵ヘッダを参照し，要求された処理が行える状態にあるかを確認する．また，鍵の状態を変更するための API を CSSM API に追加した．

インタオペラビリティ

CDSA に準拠した(鍵のライフサイクル管理を意識しない)アプリケーションから鍵生成要求があった場合は，拡張鍵ヘッダに特定のパターン情報（0パディング）を格納することとし，0パディングされた拡張鍵ヘッダを持つ鍵はライフサイクルとは無関係に暗号処理が行える．これにより，アプリケーションのインタオペラビリティを確保した．

鍵の集中管理

本システムはクライアント／サーバ構成をとる．鍵生成はサーバ側で行い，その際に鍵の利用者，用途，使用期限等をサーバ上のデータベースに保管する．生成された鍵はクライアント(利用者)に配送され，暗号処理等に利用される．これにより，鍵情報を集中管理でき，同じ鍵を長期間利用すること等を未然に防ぐことができるため，システムとしてのセキュリティを保つことが可能となる．

性能保証

暗号処理の中で最もハードウェアのリソースを必要とするのが鍵生成であり，その処理時間は処理を行うハードウェアに大きく依存する．サーバ側で鍵生成を行うことにより一定の処理能力を確保でき，モバイル端末のようにリソースの少ない端末でも本システムが利用可能となる．

7. 評価

拡張鍵ヘッダ

拡張鍵ヘッダの追加による鍵の容量の変化を測定した．拡張鍵ヘッダの大きさはおよそ280Byteであり，暗号アルゴリズム，鍵長には依存しない．拡張鍵ヘッダを含んだ鍵の容量は RSA1024bit の場合 1750Byte(但し，証明証を含んだ PKCS#12 形式)であることからメモリ領域が限られている IC カードへの暗号鍵の格納を考慮した場合でも充分対応できる．

8. まとめ

本稿では，鍵のライフサイクル管理を実装した鍵管理プラットフォームについて述べた．ISO/IEC11770-1 に準拠した鍵のライフサイクル管理，CDSA 準拠のアーキテクチャ，鍵情報の集中管理，鍵生成の性能保証を実現した．今後，IC カードとの連携等について検討する．

参考文献

[1] ISO/IEC 11770-1:1996 (E) Information technology - Security techniques - Key management - Part 1: Framework, 1996

[2] Entrust.,"Entrust key Management Overviews",Entrust Technologies White Paper, April 1996.

[3] IBM Corp.,"Key management framework and key recovery technology", IBM White Paper, February 1997.

[4] The Open Group, "Technical Standard Common Security:CDSA and CSSM",1997