PKIにおけるICカードの適用と評価

5T－4

齋藤 和美 、榊原 裕之 、太田 英憲 、辻 宏郷
三菱電機(株) 情報技術総合研究所

1. はじめに

PKI(公開鍵インフラストラクチャ)の主要構成要素である鍵管理・暗号処理機能を IC カードを用いて実現する場合、IC カードの性能等を考慮し、IC カードへのアクセスやアプリケーションからの利用方法を工夫する必要がある。本稿では、実際の適用事例[1]をもとに、IC カードを PKI システムに適用する際の問題点と解決方法について述べる。また、同方法を実装した結果を示す。

2. PKCS #11 について

今回適用したシステムでは、IC カードへのアクセスには、PKCS #11 API[2]を用いることとした。システムは、IC カードをアクセスする PKCS #11 ライブラリと鍵管理・暗号処理を実現するアプリケーションの二層から構成される。本章では、PKCS #11 の概要とアプリケーションに適用する際の典型的な使用方法について述べる。

2. 1 概要

PKCS #11 は、IC カード等の暗号トークンを用いて暗号処理を行うための API を規定した業界標準規格である。WWW ブラウザにも利用され[3]、現在、多くの IC カード製品においてサポートされている。

2. 2 PKI アプリケーションへの適用例

PKI アプリケーションへ IC カードを適用する例としては、次の三点が考えられる。

① IC カードに格納されている証明証を取得する。
② IC カードに格納されている秘密鍵(Private Key)を用いて、署名を施す。
③ IC カードに格納されている秘密鍵を用いて、データを復号する。

各処理において必要な PKCS #11 の手順としては、次のようなものが典型的な例として考えられる。

①の場合、「ライブラリの初期処理」、「スロットの確認」、「セッションの開始」、「証明証の取得」、「セッションの終了」、「ライブラリの終了処理」となる。

②の場合、「ライブラリの初期処理」、「スロットの確認」、「セッションの開始」、「ログイン」、「署名」、「ログアウト」、「セッションの終了」、「ライブラリの終了処理」となる。

③の場合、「ライブラリの初期処理」、「スロットの確認」、「セッションの開始」、「ログイン」、「データの復号」、「ログアウト」、「セッションの終了」、「ライブラリの終了処理」となる。

3. 実装上の問題点

図 1 に示すように、実際のアプリケーションでは、2. 2 の各処理を組み合わせて構築するので、2.2 の手順通りに実装すると、デバイスとしてファイルを用いた場合と比較して、安全性は向上する反面、速度は低下する。IC カード自体の性能に依存する部分も多いが、IC カードに格納されているデータの読み書きや演算処理の性能を考慮しつつ、より高速化させるように、PKCS #11 の適用方法を検討する必要がある。

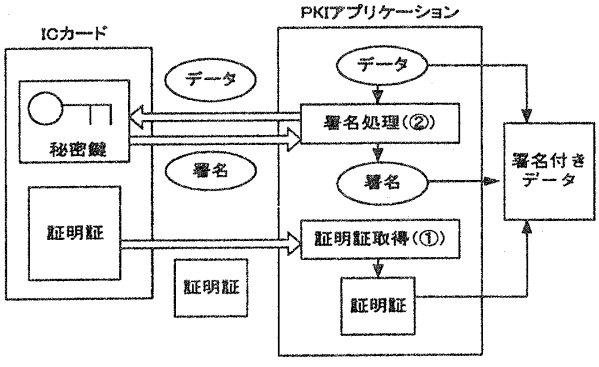

図 1 IC カードを用いた PKI アプリケーション例

4. 高速化手法

今回のシステムでは、PKCS #11 ライブラリから IC カードへアクセスする場合と、アプリケーションが PK

Evaluation of Applying Smart Card to Public Key Infrastructure
Kazumi SAITO, Hiroyuki SAKAKIBARA, Hidenori OHTA, Hirosato TSUJI
Information Technology R&D Center, Mitsubishi Electric Corporation

CS #11 ライブラリを使用する場合の各々について、高速化に対する検討を行った。

4. 1 PKCS #11 ライブラリにおける高速化

IC カード内には、様々なデータが格納されているが、PKCS #11 ライブラリが内部で使用するデータやパスワード認証の不要なデータに関しては、処理の開始時(PKCS #11 の手順では、「セッションの開始」時)に一括して IC カードから読み出して、メモリ領域に保管する。例えば、IC カードにパスワード認証不要な証明証とパスワード認証必要な秘密鍵が格納されている場合、ライブラリは、証明証データ自体と、秘密鍵データの管理情報を IC カードから読み込み、メモリ領域に保管する。アプリケーションから要求があると、ライブラリは、メモリ領域に保管されているデータにアクセスしたり、管理情報によって IC カードのデータ格納領域に直接アクセスする。これにより、比較的時間を費やす IC カードへのアクセス回数が削減する。

4. 2 アプリケーションにおける高速化

PKCS #11 を利用するアプリケーション側では、各処理において必要最低限な手順のみを呼び出すことで、PKCS #11 の効率的な利用方法を実現する。即ち「ライブラリの初期処理」、「スロットの確認」、「セッションの開始」、「ログイン」等は、アプリケーションの初期化時に一括して実行する。また、「ログアウト」、「セッションの終了」、「ライブラリの終了処理」等は、アプリケーションの終了時に実行する(図 2)。

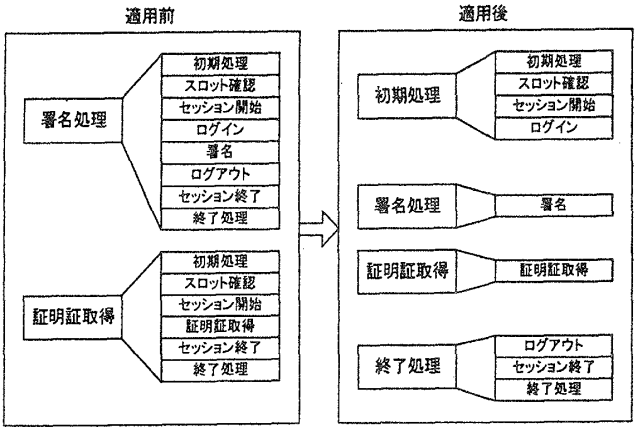

図 2 アプリケーションにおける高速化の例

但し、一括して実行する処理の選択は、そのアプリケーションのセキュリティ要件を考慮し、注意して行う必要がある。例えば、「ログイン」及び「ログアウト」は、ユーザのパスワード認証を行う手順なので、初期化時に実行するよりも、各処理の直前に実行する方が安全性は高い。また、「セッションの開始」及び「セッションの終了」は、IC カードへのアクセス権限を指定可能な手順なので、各処理の中で実行すると細かなアクセス制御が可能である。

5. 適用結果

2.2で述べた処理を全て実行するアプリケーションを作成し、高速化手法の適用前と適用後での処理時間の差分を測定した。結果を表1に示す。(a)は4.1の方式のみ、(b)は4.2の方式のみ、(c)は(a)と(b)を組み合わせた方式を示す。ICカードには、1024ビットの鍵長を持つ証明証1枚と、秘密鍵を格納したものを用いた。

この結果、アプリケーション全体では、最大22秒強の処理時間の短縮に成功した。

(単位:秒)

処理	(a)	(b)	(c)
証明書取得	−2.997	−7.945	−10.525
署名	−2.450	−7.991	−9.924
データ復号	−2.377	−7.945	−9.851
アプリケーション全体	−7.827	−15.916	−22.872

表 1 高速化手法の適用結果

6. おわりに

ICカードをPKIシステムに適用する際の問題点を示し、解決手法について述べた。今後は、より安全性が高く、且つ効率的なICカード高速化手法について検討する予定である。

参考文献

[1] 辻・榊原・齋藤・太田, “PKI 暗号ライブラリにおける IC カードの利用(1)－概要－”, 情報処理学会第 58 回全国大会 2L-04, 1999.

[2] RSA Lab.,“PKCS #11:Cryptographic Token Interface Standard”,Version2.01,1997.

[3] Netscape Communications Corp.,"Implementing PKCS #11 for the Netscape Security Library",http://developer.netscape.com/docs/manuals/security/pkcs/pkcs.htm,1999.