HSM（Hardware Security Module）の試作と評価

2M－4

竹原 明　中野 初美　中川路 哲男
三菱電機（株）　情報技術総合研究所

1．はじめに

公開鍵暗号技術を用いて電子署名や暗号化を実現するための基盤技術であるPKI（Public Key Infrastructure）ではデータの暗号化や電子署名などで使用される暗号鍵の保管には高度な安全性が要求される。そのため、システムの安全上特に重要な暗号鍵の管理には、暗号鍵をハードウェアで安全に管理するHSMが求められるようになってきている。今回、HSMに業界標準のAPIを実装して他のアプリケーションとの接続を可能とするとともに、複数のHSMによる負荷分散機能を実装した。本稿ではHSMの概要および評価結果を報告する。

2．HSMの概要

2．1　特長

本HSMはPCIボード上に実装しており、以下の特長を持つ。

① 公開鍵対の生成、暗号処理を内部に隠蔽し、ボードの抜き取りなどの不正アクセスを検出して保持している鍵を瞬時に消去する耐タンパー機構を備えている。これはFIPS(Federal Information Processing Standards) 140-1 Level3に相当する。

② 業界標準規格となっているPKCS#11準拠のAPIを提供することにより、他のPKCS#11準拠の鍵格納デバイス（暗号モジュール）との互換性を実現している。

2．2　PKCS#11ライブラリ

本来PKCS#11はマルチスロット対応であり、使用するスロットを指定して実行するAPIであるが、本HSMではPKCS#11ライブラリが複数のHSMを管理し、内部で処理を分散させることで、アプリケーションから複数のHSMを意識することなく、負荷分散処理を可能にした。

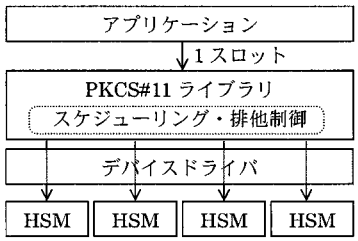

図2　PKCS#11ライブラリによる負荷分散

3．評価

3．1　秘密鍵演算性能

表1　測定環境

OS	CPU
WindowsNT4.0	PentiumⅢ 500MHz
Solaris7	UltraSPARC-Ⅱ 450MHz

まず、HSMを使用する主目的である電子署名生成や復号で実行されるRSA秘密鍵演算の性能を表1に示すプラットフォーム上でHSM1台を接続して測定した。表2中の処理時間はホスト上のPKCS#11ライブラリの署名関数実行開始から終了までの時間の測定値である。秘密鍵演算処理時間の大部分はHSMでの処理時間であるためSolaris7、WindowsNTともほぼ同じ結果が得られた。

表2　RSA秘密鍵演算処理時間

	1024ビット	2048ビット
処理時間(msec)	170	1,050

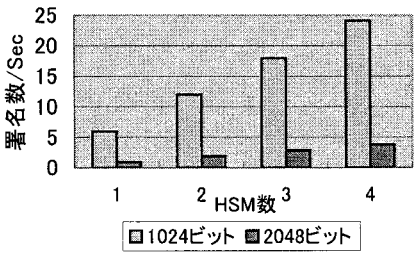

図3　署名生成負荷分散性能

A prototype of Hardware Security Module and its evaluation
Akira Takehara, Hatsumi Nakano, Tetsuo Nakakawaji
Information Technology R&D Center,
Mitsubishi Electric Corporation

次に複数の HSM を接続し、マルチスレッドで署名生成を繰返し実行した結果、図 3 のようにほぼ HSM 数に比例した結果が得られた。

3．2 SSL サーバとの接続性能

SSL サーバではコネクション確立の際の CPU 負荷が大きいことが問題になるため、秘密鍵の管理に加えて CPU 負荷の分散のために SSL サーバに本 HSM を接続して評価を行った。

評価は図 4 に示すように Solaris 上の iPlanet Web Server（以下 iWS）に本 HSM 用の PKCS#11 ライブラリを接続し、4 クライアントから同時に繰り返しアクセスした場合の性能とサーバ CPU 負荷を測定した。iWS の秘密鍵は 1024 ビット鍵を使用した。

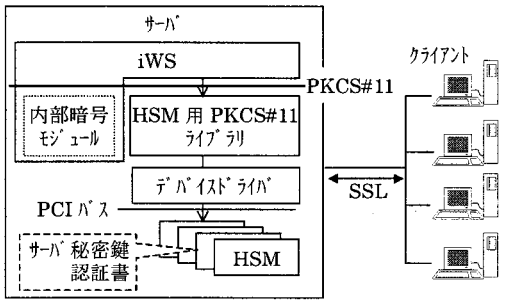

図 4　iWS と HSM の接続構成図

図 5 はクライアントでの測定時間から算出した平均性能である。測定時間にはコネクション確立後の 10KB のコンテンツのダウンロード時間を含む。

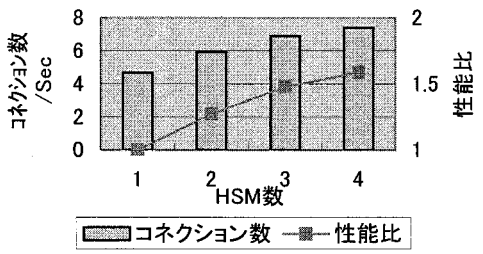

図 5　iWS 接続時の負荷分散性能

図 6 は 4 クライアントがそれぞれ 100 回コネクションを繰り返し確立した場合のサーバ上で測定した CPU 負荷であり、HSM 未使用時は実行開始から 21 秒まで、HSM4 台使用時は 56 秒までが実行時の CPU 負荷を示している。

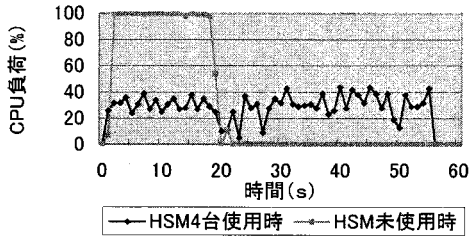

図 6　iWS 接続時のサーバ CPU 負荷

4．考察

署名生成のみを繰り返し実行する場合には HSM 数にほぼ比例して処理性能が向上しており、十分な効果が確認できた。認証局などのように主に署名生成に使用されるシステムにおいては HSM を複数使用することにより高速化が期待できる。

iWS と接続した場合には HSM が 4 台で 1.6 倍程度の向上しか得られなかったが、コネクション確立の際にハッシュや MAC の生成が頻繁に実行されており、これらの処理がホスト側のライブラリで実行されているのが原因だと推測される。また、iWS の秘密鍵に 2048 ビット鍵を使用した場合には相対的に HSM の処理時間が長くなるため、複数枚使用時の効果が大きく現れると思われる。サーバ側の CPU 負荷については処理時間が異なるため単純に比較できないが、HSM を使用しない場合にはクライアント 4 台で負荷が 100%に達しているため、クライアント数がさらに多くなるにつれ、HSM 使用による負荷分散効果が大きくなると考えられる。

5．まとめ

本稿では、試作した HSM の概要および評価結果について述べた。今後、負荷分散時の性能解析、性能改善や適用システムの検討を行う。

参考文献

[1] National Institute of Standards and Technology , “Security requirements for cryptographic modules”, 1994

[2] RSA Laboratories, “PKCS#11 Cryptographic Token Interface Standard”, 1997