経済産業省補助事業

平成１５年度情報セキュリティ対策推進事業

（電子商取引（ＥＣ）技術基盤の相互運用性に関する調査研究）

PKI の国際的相互接続実証実験報告書

平成１６年３月

（財）日本情報処理開発協会

1. はじめに

インターネットの急速な普及により電子商取引やオープンなネットワークを経由しての情報交換が広く行なわれるようになってきたが、公開鍵基盤（PKI）は取引相手の確認や情報の改ざん防止等、インターネット上での安全、確実な情報交換を実現する必須の技術として、ますますその重要性を増しているといえる。

国内では電子政府／電子自治体の構築が着実に進んでいる。政府・自治体での省庁への各種申請・届出等が電子証明書を用いて行なえるようになりつつあるほか、平成 16 年 1 月には公的個人認証サービスが開始され、市民ひとりひとりへの電子証明書の配布が開始されるなど、オンラインによる行政サービスの基盤は順調に整備されていている。ビジネス分野においては、企業内イントラでの利用が進むとともに、PKI を利用した時刻認証や公証サービス、属性認証へのニーズの増大、WEB サービスでの実装等、実ビジネスでのアプリケーションやソリューションへのニーズが高まってきている。既に PKI は本格的な普及の段階を迎えたと言える。

一方海外に眼を移すと、例えばアジア圏で特に IT に関する先進的な取り組みを行なっているシンガポール、香港（中国）などでは、PKI を利用した電子政府および電子商取引においてめざましい進展を遂げている。また、欧米の IT 先進国では国民 ID カードの活用、PKI をキーアプリケーションに包含したサービスの提供が進むとともに、国家間、官民間、コミュニティ間等様々なスキームでの相互運用が行なわれている。このように各地域での PKI 普及が新しい段階を向かえつつある今日、各ドメイン間の PKI 相互接続の必要性はますます重要度を増してきている。

平成 13 年以降、日本 PKI フォーラムはアジア圏の各国・地域とともに PKI の相互接続に関する課題を検討してきており、CA 相互接続インターフェース仕様、パス検証ガイドライン、署名用トークンインターフェース仕様の開発を行なってきたが、平成 15 年 3 月には PKI 相互接続に関する仕様を策定することができた。今後はこの仕様を実際のアプリケーションに広く適用してもらうことが、アジア圏での PKI 相互接続の実現に向けた大きな原動力となるものと考える。

本事業は、このような目的の下、経済産業省の支援により、平成 14 年度までに韓国、シンガポール、チャイニーズ台北、香港チャイナとともに開発した CA 間の相互接続インターフェース仕様、パス検証テストガイドライン、署名用トークンインターフェース仕様について、更にタイを実験対象国として追加して仕様の有効性を確認し、また課題となっていた事項の解決を図り内容を更に充実させるとともに、CA 間の相互接続インターフェース仕様のチェックリスト化及びパス検証テスト用テストデータ作成支援ツール等を開発して、各仕様のユーザビリティ向上を行なったものである。

2 プロジェクト概要

2.1　期間

平成 15 年 9 月 24 日～平成 16 年 3 月 7 日まで

2.2　体制

本実証実験の参加国／地域と推進組織を「表 2.1　体制」に示す。

表 2.1　体制

参加国／地域	組織
日本	日本 PKI フォーラム
韓国	Korea PKI Forum
シンガポール	PKI Forum Singapore
チャイニーズ台北	Chinese Taipei PKI Forum
タイ	Government Information Technology Services（以降 GITS と表記）

2.3　活動概要

顕在化した技術的な課題の解決を含む相互接続基盤の整備にあたり、以下の活動を実施した。

- 電子署名に関連する法律、政省令及び認証局の認可・認定に関わる文献調査を行い、国／地域による差分を抽出した。
- 実アプリケーションの実現について、ターゲットとして PAA（Pan Asia e-commerce Alliance）を適用するに当たっての法制度面、技術面、運用面での課題を調査した。実証実験による検証等、課題解決のための方策について考察を行った。
- 日本、韓国、シンガポール、チャイニーズ台北、タイの認証局間で相互接続が可能となる、認証局間相互接続及びそのテストに関する標準、証明書検証に関する標準、アプリケーションインターフェースに関する標準案を作成した。
- 各標準案について Korea PKI Forum、PKI Forum Singapore、Chinese Taipei PKI Forum、及び日本 PKI フォーラムの参加団体のメンバを含む各国／地域技術者と調整した。
- 認証局間相互接続及びそのテストに関する標準案について GITS、及び日本 PKI フォーラムの参加団体のメンバを含む各国技術者と調整した。
- 各標準案の有効性を、実証実験環境にて検証した。
- 実証実験の成果を標準案にフィードバックし、標準を作成した。

3 標準作成

3.1　概要

本標準作成作業は、日本、韓国、シンガポール、チャイニーズ台北、香港チャイナ、タイの認証局間で相互接続が可能となる、認証局間相互接続及びそのテストに関する標準、証明書検証に関する標準、アプリケーションインターフェースに関する標準を策定した。

アプリケーションインターフェースに関する標準では、実アプリケーションの実現について、ターゲットとして PAA（Pan Asia e-commerce Alliance）を適用するに当たっての法制度面、技術面、運用面での課題を調査する。実証実験による検証等、課題解決のための方策について考察を行った。

3.2　認証局間相互接続及びそのテストに関する標準

相互接続基盤の整備に際し、認証局間相互接続及びそのテストに関する標準を定めた。

3.3　証明書検証に関する標準

認証局の設計担当者及びソフトウェア開発者が、アプリケーションにおける証明書検証方法の設計作業において必要となる検証要件及び評価作業におけるテスト項目を検討する際に使用するパス検証に関する標準を定めた。前年度作成した標準に対しテスト検証テストパターンの追加を行うと共に、パス検証テストツールを整備することにより実験を効率的に行う方法について追加した。

3.4　アプリケーションインターフェースに関する標準

相互接続基盤の整備に際し、アプリケーションに関する標準としてアプリケーションインターフェース仕様を定めた。

4 開発作業

4.1　アプリケーションインターフェース確認機能

トークンに格納された公開鍵証明書及び秘密鍵を利用可能とするための機能等を具備するソフトウェアとして、アプリケーションインターフェース確認機能を開発した。

4.2　パス検証テストツール

パス検証テストガイドラインに則ったテストを支援するソフトウェアとして、以下のシステムを開発した。

(1) パス検証テストデータローダー

リポジトリの汎用データフォーマットである LDIF ファイルを入力としてテストケースを管理するデータベースへ登録するコマンドラインプログラムである。これにより、前年度実験で利用したデータの再利用や改変が容易可能

となった。なお、証明書、CRL および証明書発行要求のプロファイル準拠性のチェック機能を含む。

(2) パス検証テストデータジェネレータ

データベースで管理されているテスト情報に基づき、汎用リポジトリデータフォーマットである LDIF を生成するためのコマンドラインプログラムである。これによりテストに必要な鍵ペア、証明書、証明書失効リストなどの情報を含んだ LDIF ファイルを一括して自動生成することができる。

5 タイ認証局との認証局間相互接続テストガイドラインの実証実験

利用者に、一連のテストを「認証局間相互接続テストガイドライン」に従って実施することにより、「認証局間相互接続テストガイドライン」が、利用者にとって、テスト実施に必要な事前チェックの要件やテスト手順、及び実施結果の正確な確認方法についての必要十分な内容であることを確認した。

6 パス検証テストガイドラインの実証実験

「パス検証テストツールの開発」で開発されたツールを用いることにより「パス検証テストガイドラインの有効性の検証」の実証実験がより効率的に行えるようになったか、作業負担が軽減されたかを検証した。

7 アプリケーションインターフェースの実証実験

各国／地域の PKCS#11 ライブラリの機能的な差異及び PKCS#11 ライブラリとアプリケーションの開発言語の差異を吸収するための機能を開発し、各国／地域の PKCS#11 ライブラリと証明書トークンが正しく処理できるかについて実験を行った。

8 検証結果

検証作業を通じて、「認証局間相互接続テストガイドライン」に従ってテストを実施することで、利用者は目的とする認証局間の相互接続についての確認を正しく行うことができることが明らかになった。「認証局間相互接続テストガイドライン」の定める仕様とテスト手順に基づいて構築した認証局間の相互接続関係の中では、双方の認証ドメインでエンドエンティティの検証が正しく機能することが確認され、PKI の利用環境として正しく相互接続関係を構築することができたと判断される。また利用者であるテスト主体の認証局が不正な証明書を発行した場合や、データ環境に不備がある状態に陥った場合には、テストによって正しく検出された。これらのことから「認証局間相互接続テストガイドライン」を使用して相互接続関係の構築を検証することは、利用者にとって

有効であることが確認されたものと判断する。

9 全体考察（まとめ）

9.1 成果

9.1.1 認証局間相互接続及びそのテストに関する成果

(1) テストガイドライン作成による簡易で安全な認証局間相互接続テスト手法の確立

平成 14 年度までの実証実験では証明書及び証明書失効リストプロファイル及び PKI コンポーネントのインターフェースは決められていたが、実証実験を進めるためのノウハウや実験参加国／地域で議論された内容は必ずしも整理・明文化されているわけではなかった。

そこで、今年度相互接続のノウハウを持たない国／地域を含んだ相互接続においても、安全でスムーズな関係構築を可能とするために、今年度これまでの認証局間相互接続実証実験を標準テストパターンとしてブラッシュアップし、「認証局間相互接続テストガイドライン」を作成した。

本ガイドラインは、相互接続時の事前確認項目、テストの具体的な作業手順及び結果確認の方法を明記したものであり、認証局間固有の運用及び使用しているアプリケーションに依存するテスト項目・手順等を除き、ドキュメントの可読性及び作業の効率的な流れに注意し設計したものである。

テストガイドライン構築の成果は主に以下の 3 点である。

(a) ノウハウのない利用者でも安全な相互接続を実現

これまでの実証実験で、異なるドメイン間の相互接続については実験参加国の国内で実績が無く、本実証実験で初めて実施するという場合が少なくなかった。そのような場合、実験参加国である CA 運用者及び証明書検証者は相互接続のノウハウを持たず、テスト手順自体が相互接続作業の具体的な作業手順そのものであった。ノウハウを持たない作業者にとっては、必要十分な作業手順やチェック項目を漏れなく設定すること自体が困難である。

ガイドラインの作成により、テスト手順に従えば相互接続関係がスムーズに構築でき、また結果の確認を恣意的でなく行うことが出来るようになり、安全な相互接続を実現できるようになったと言える。

(b) テスト作業の効率化によるテスト期間の短縮

これまでの実証実験のノウハウの蓄積から、(i)相互接続を行う場合に陥りやすいポイントやパターンを整理し、事前確認項目として明確化し、(ii)テスト環境の変更を極力少なくするテスト手順を構築することにより、テスト作業の効率化を図ることができテスト期間を短縮することが出来た。

テスト実施期間としては、トラブルシューティング、各認証局におけるセキュリティ確保のための運用ポリシの定める手順及びオペレーションの冗長性等の時間を除けば、シミュレーションセンターの環境で行ったように最短１日で消化できることが検証できた。

(c) ガイドライン利用者の可読性の向上

コンテンツ提供形態を平成 14 年度までのプレーンなドキュメントスタイルからインタラクティブな Web インターフェースに移行し、関連する PKI ドキュメントを整理してハイパーリンクを行った。これによりガイドライン利用者の可読性を大きく向上することができた。

(2) 実運用に向けた認証局間相互接続仕様の確立

これまでの韓国・シンガポール・チャイニーズ台北・香港チャイナ、及びタイの合計 5 カ国／地域と相互接続実証実験を行い、仕様としても平成 14 年度のものをそのまま適用することができた。

これはすなわち、本仕様が一定の完成度に達したことを示しており、他国のヒアリング結果においても高い評価を得ていることから、国際的に通用する相互接続仕様として確立できたといえる。

また、今年度は実証実験と実運用との相違点を注記することにより、実運用により適用しやすくなったと言える。

9.1.2 パス検証に関する成果

(1) パス検証ガイドラインの確立

パス検証ガイドラインとは、認証局の設計担当者及びソフトウェア開発者がアプリケーションにおける証明書検証方法の設計作業を行う際に必要となる検証要件、評価作業におけるテスト項目を検討する際に使用する標準である。

基本的には X.509 や RFC3280 の標準において記述されているパス検証ロジックを対象としたテスト項目となっているが、昨年度までの実験で策定した 証明書および証明書失効リストプロファイル（以下「プロファイル」）に準拠していることを前提としたテスト内容となっている。

パス検証ガイドラインは平成 14 年度、第一版が公開されたが、その後、国内外の実証実験参加メンバより 3 つ指摘事項があった。相互接続モデルによってテストパターンを分類しているが、テストパターンに関してはプロファイルに必要な接続モデルのみにすべきであるという意見。1 つは失効モデル、サービスモデルという分類は有益であるが、現行のテストパターンは、重複してい

るものが多くテストが冗長であり整理が必要であるという意見。もう 1 つは、最後は、テストレベルにおいてどこまでを満足する必要があるのか、その規範が提示されていないために誤解を招きやすいという意見であった。

以上の課題を解決するためにパス検証ガイドラインで定義されるテストパターンの見直しを行い、失効モデル、サービスモデルについて、トラストモデルを中心とするようなテスト体系に変更した。また、テストレベルは標準テストとオプションテストの 2 種類のみとした。これにより、本ガイドラインの利用者が何をすべきか判断が容易になった。

パス検証ガイドラインはドキュメントのみならず、テストデータを証明書、証明書失効リストのファイルのみでなく、データベースのデータファイルとしても配布することが可能であり、テスト環境の構築／再現が容易で再利用性の高いデータとなっている。

このように本ガイドラインは、平成 15 年度の改訂を以って課題として残された問題をクリアしアプリケーションが相互接続環境において必要なパス検証機能を実装しているかどうかを評価するためのフレームワークとして利用することができる有効なガイドラインとすることができた。

(2) テストツールを用いたパス検証テストの効率化

平成 15 年度実験では、パス検証テストに用いる証明書や証明書失効リストの発行およびリポジトリの提供をテストツールを用いて行った。

パス検証実験に用いられる認証局やリポジトリ等のサーバ環境については Apache や OpenLDAP 等のオープンソースソフトウェアで構成される 1 台のテストツールサーバにより、複数のリポジトリを提供しており、パス検証実験に必要なリポジトリ環境等の構築にフリーウェアのみで構成されている。

テストパターン設計およびテストデータの生成においては、テストツールサーバ上のウェブサーバを介して、日本とチャイニーズ台北が協調作業によりテストパターンの設計やデータのデータベース投入を効率的に行えた。

テストツールを用いたテストデータの設計、テストデータとなる証明書、証明書失効リストおよびリポジトリ設定ファイル(LDIF)をバッチ処理により一括自動生成でき、効率的にテストデータおよびリポジトリを構築することができた。

また、平成 15 年度の成果であるテストデータを利用できることにより、パス検証実験の再実験や新規参加国の実験の際に、実験環境の再構築が格段に容

易になり、構成の変更、テストケースの変更にも柔軟に対応可能であり再利用性も 1 つの大きな成果となっている。

9.1.3 アプリケーションインターフェースに関する成果

(1) アプリケーションサービスの国際的利用における共通仕様の確立

昨今の PKI 推進活動は国の内外を問わず盛んであり、具体的な活動としてはアメリカにおける OASIS（旧 PKI フォーラム）、EU 圏における EESSI、国内における GPKI、LGPKI、CALS/EC、公的個人認証などが挙げられる。これらの状況において、PKI を活用する PKI アプリケーションの観点で見た場合、特定の仕様或いは特定の製品に依存した PKI アプリケーションとなることも多々見受けられる。このような場合、ある PKI アプリケーションサービスを利用するには利用者が PKI アプリケーションの要求を満たす署名・暗号機能をインストールする等の作業が必要となり、PKI アプリケーションごとにクライアント環境を構築する必要があるなどの煩雑さを増す要因となる。

1 国内で使用する場合にはこのような問題がある。また国際間で 1 つの PKI アプリケーションサービスを利用しようとした場合、クライアントで準備する必要がある署名・暗号機能は輸出規制の対象となり、特定の製品に依存する PKI アプリケーションを構築することは実用的ではない。また、PKI アプリケーションが各国で使用可能な署名・暗号機能を使用するには、署名・暗号機能の仕様が統一されていないことから、PKI アプリケーションの構築が不可能となるという問題がある。

これらの問題は、国際間における PKI アプリケーションの普及を妨げる要因であり、国際間における PKI アプリケーションを普及させるにはこれらの問題解決を行うことが最大の課題であると言っても過言ではない。国際的な標準に基づいた上で、最低限の共通ルールを設定する必要がある。

平成 14 年度の本事業において、PKCS#11 に関する署名機能について共通仕様を策定した。本年度はこれを拡張し、署名機能の改善を行うと共に暗号化機能、鍵管理機能を追加することでアプリケーションからの用途を拡大した。

現在、PKI を活用するアプリケーションと PKI コンポーネントとの間のインターフェースに関して世界的な標準として使用されているものには、Windows の CryptoAPI、Java の JCA/JCE、鍵管理に関する PKCS#11 等がある。

CryptoAPI や Java は世界中で共通に使用することが可能であり、CryptoAPI では InternetExplorer のバージョン、Java では JRE（Java Runtime Environment）のバージョンを合わせることで共通な動作をさせることが可能である。

しかし、PKCS#11 については、仕様がスタンダードとして広まっているものの、ライブラリとしての実装は各国各社における製品化、或いは GNU 等におけるフリーのライブラリ化のようにそれぞれバラバラに実装を進めているのが現状であり、国際的に共通のアプリケーションを利用しようとした場合、アプリケーションと PKCS#11 ライブラリの間のインターフェースに関して以下の問題が発生する可能性がある。

- PKCS#11 ライブラリのファイル名がそれぞれ異なるため、アプリケーションから PKCS#11 ライブラリを認識させることが困難である。
- スタンダードな仕様に沿っていても、実装の際の様々な事情により詳細機能で個別差が発生し、期待どおりの動作結果を得られない可能性がある。

今回のアプリケーションインターフェース仕様の策定は、これらの問題を解消するものであり、実証実験により仕様の有効性を検証することができた。これにより、アジア圏において PKCS#11 を CryptoAPI や Java と同様に共通アプリケーションサービスで利用することが可能となり、CryptoAPI や Java と並んでアプリケーションサービスの共通利用におけるセキュリティプラットフォームの選択肢の 1 つとすることが可能となった。

一方、今回のアプリケーションインターフェース仕様の中では、署名付与・検証機能、暗号化・復号機能、鍵の安全な管理のためのラップ・アンラップ機能、1 台のクライアント PC を複数人で利用する場合を想定したマルチスロット／鍵管理機能を盛り込んだ。

実ビジネスにおけるアプリケーションの運用では、PKI に関する機能として以下の機能が必要とされている。

1. 安全な鍵管理機能
2. 送信する電子文書への電子署名付与機能
3. 送信された電子署名の検証機能
4. 電子署名と合わせて送信された署名者の証明書の有効性検証
5. 電子文書の暗号化・復号機能
6. 暗号化通信機能
7. 利用者認証機能
8. アプリケーション環境の正当性保証機能

これらの機能のうち、PKCS#11 がサポートする機能は 1.～3.と 5.であり、PKCS#11 に関する今回のアプリケーションインターフェース仕様で全て網羅されている。

更に、マルチスロット／鍵管理機能を盛り込むことでアプリケーション利用

時の利便性を向上させ、より実用的な仕様とすることができた。
　以上により、PKCS#11 に関しアプリケーションサービスの国際的利用における共通仕様を確立することができた。

(2) 平成 14 年度の署名処理の課題解決
　平成 14 年度に策定した署名用トークンインターフェースでは、署名アルゴリズムとして「CKM_RSA_PKCS」を使用するというものであり、これは署名対象データを PKCS#1 形式で暗号化するものであった。これは、署名対象データからハッシュ値を作成し、署名者の秘密鍵で暗号化するという本来あるべき署名付与動作とは違うものであり、以下の点で課題が残った。
・ 署名者の成りすましが可能である。
・ 署名データの改ざんを検出することができない。

　署名アルゴリズムとして「CKM_RSA_PKCS」を採用した背景としては、SmartCard を使用する場合、署名値を作成する演算が SmartCard 内で行われるため、SmartCard のメモリ容量の制限からハッシュ値を暗号化する複数の処理を一括して行うことに無理が生じる危険性があったためである。
　本年度はこれを改善し、署名に関するインターフェースを共通化するのみにとどまらず、署名本来の機能を実現させることができた。
　改善点は次のとおりである。
・ 署名処理をハッシュ値作成とハッシュ値の暗号化の 2 つの処理に分割する。
・ ハッシュ値作成については「CKM_SHA1_PKCS」のアルゴリズムパラメータを使用し、現在、多く用いられている SHA1 アルゴリズムによりハッシュ値を作成する。
・ ハッシュ値の暗号化については、平成 14 年度と同様に「CKM_RSA_PKCS」のアルゴリズムパラメータを使用し、署名者の秘密鍵を使用してハッシュ値を暗号化する。

　以上により、SmartCard のメモリ容量が厳しい環境でも、SHA1withRSA で署名値を作成する場合と同等の結果を得ることができる仕様を確立することができた。