

開発成果報告書

ソフトウェア開発編 2章

ソフトウェア開発総括報告

2. ソフトウェア開発総括報告

PKIのアプリケーションとして、Webブラウザ、電子メールソフトウェアなどがあるが、これらの既存の製品では、ブリッジCAなどの新たな役割機能を導入したGPKI環境において、パス構築、証明書検証機構などの点で十分に対応できていない。本プロジェクトは、電子メールソフトウェアに焦点を当て、電子メールにセキュリティを付与するS/MIME形式の電子メールメッセージを運用する際に、GPKIにて利用される公開鍵証明書を正しく取り扱うことができるように技術調査し、クライアントソフトウェアの機能を拡張するプラグインソフトウェアを開発するものである。公開するプラグインソフトウェアは、整備されたAPI群を持ち、今後のGPKI対応のアプリケーション開発を容易にする。

また、広く普及しているWindows環境における、電子メールのクライアントソフトウェアとして、多くの市場シェアを占めているMicrosoft社のOutlook ExpressあるいはOutlookにて、GPKI環境において公開鍵証明書を正しく扱えるようにする。

2.1. はじめに

政府は、行政の効率化や国民負担の軽減を目標に行政手続きを電子化する「電子政府」の基盤を構築することを目指している。電子政府の構築は、デジタル経済・社会の一つのモデルであり、その中で実施される情報セキュリティ確保のための対策もまた、広く民間の範となるべきものであり、我が国のネットワーク全体の安全性・信頼性を高め、更に、具体的に進みつつある同様な取り組みと連携していくことにより国際的な貢献につながることを期待される。そのセキュアな情報流通の基盤として、政府認証基盤(GPKI : Government Public Key Infrastructure)が構築されている。

従来、国民等から行政機関に対する申請・届出等や行政機関から国民等への結果の通知等は、署名又は記名押印した書面に行われるのが通常であった。しかし、インターネットを利用してこのやり取りを行う場合には、申請・届出等や結果の通知等が本当にその名義人(申請者や行政機関の処分権者)によって作成されたものか、申請書や通知文書の内容が改ざんされていないかを確認できなければならない。これを確認できるようにするための行政機関側の仕組みが GPKI である。

GPKI に基づいて各種行政手続きが電子化されるわけだが、本研究開発では、情報伝達のために欠かすことの出来ない手段である電子メールに焦点を当てている。

電子メールでは、インターネットにおけるデータを交換するための規約“ MIME”を利用している。これに、PKI (Public Key Infrastructure) と暗号技術を電子メールに適用し、秘匿性、認証、真正性を与えるものが“S/MIME”であり、IETF にて標準仕様の策定が行われ、いくつかの電子メールソフトウェアにて実装され利用可能となっている。しかし、既存の S/MIME 対応の電子メールソフトウェアは GPKI を想定しておらず、GPKI にて利用される公開鍵証明書に十全には対応できていない。本テーマでは GPKI にて発行される公開鍵証明書を正しく取り扱い、S/MIME 形式の電子メールメッセージのやり取りが正しく行われるよう、クライアントソフトウェアの機能を拡張するプラグインソフトウェアを開発した。

プラグインソフトウェアの実現としては、広く普及している、Microsoft Windows において利用可能な Win32 対応のアプリケーション・ライブラリと、電子メールソフトウェアとして多くの市場シェアを占めている Microsoft 社の Outlook もしくは Outlook Express のいずれか一方を対象としたアドインモジュールを目標とした。

2.2. 研究開発の目標と内容

2.2.1. 研究開発の目標

2.2.1.1. 目標

本研究開発では、相互認証をさらに拡張したブリッジCAを配したGPKI環境における公開鍵証明書を正当に扱い、S/MIME形式の電子メールメッセージの取り扱いを可能とする、クライアントソフトウェアの機能を拡張するプラグインソフトウェアを開発する。具体的には、開発されたソフトウェアによって以下の2つのことが可能となることを目標とする。

(1) C言語APIの提供

電子メールソフトウェア開発者がS/MIME未対応の電子メールソフトウェア対し、GPKI環境においてS/MIMEメールメッセージの送受信が可能となるような開発を容易とする外部アプリケーションインターフェイスの提供

(2) Microsoft Outlook あるいは Outlook Express への GPKI 対応の拡張

電子メールのクライアントソフトウェアとして、広く普及しているWindows環境にて、多くの市場シェアを占めているMicrosoft社のOutlook、Outlook Expressがある。これらは、すでにS/MIMEには対応しているが、2.2.1.2に示すような要件をみたしておらず、証明書検証機構が正しく動作しない。そこで、Outlook もしくは Outlook Express にて、S/MIME 利用時にGPKIに対応した、証明書パス構築、検証機構を与える拡張を行う。なお、本年度は Outlook Express に対する拡張機能の開発のみを行った。

2.2.1.2. GPKI 対応の要件

既存のPKIアプリケーションが対応できていないGPKIの特性について以下に述べる。

(1) 信頼モデル

相互認証、ブリッジ認証のような信頼のモデルを導入したことにより、以下の機能が必要となる。

A) 証明書パス構築に関して

相互認証証明書を介したパス構築、および単一のトラストアンカによる証明書パスの検証を可能とすること。

B) X.509 拡張領域の解釈

証明書ポリシ (certificatePolicies)
ポリシマッピング (policyMapping)
ポリシ制約 (policyConstraints)

C) 相互認証証明書

リポジトリに格納された次のような形式の CrossCertificatePair 型のデータを解釈する。

```
CertificatePair ::= SEQUENCE {  
  issuedToThisCA [0] Certificate OPTIONAL,  
  issuedByThisCA [1] Certificate OPTIONAL}
```

(2) 鍵更新

自己署名証明書を持つ認証局の秘密鍵を更新することで、一時期に新しい世代の公開鍵証明書と古い世代の公開鍵証明書が存在することになる。新しい世代の鍵ペアと古い世代の鍵ペアの関係を保証し、更新しても既存の有効な証明書による証明書パスを構築できるようにするため、以下のような4種の証明書を持つこととなる。

- 旧世代秘密鍵で署名した旧世代公開鍵の証明書 (oldWithOld)
- 新世代秘密鍵で署名した旧世代公開鍵の証明書 (oldWithNew)
- 旧世代秘密鍵で署名した新世代公開鍵の証明書 (newWithOld)
- 新世代秘密鍵で署名した新世代公開鍵の証明書 (newWithNew)

これら4種の証明書がリポジトリの caCertificate 属性値として別々に登録されている。

(3) リポジトリアクセス

証明書検証時、パス上の証明書や相互認証証明書ペア、ならびにそれら証明書の発行元の CRL、ARL を、LDAP プロトコルを用いリアルタイムにオンラインでリポジトリから取得する。

(4) OCSP、証明書検証サーバ

RFC2560 に基づく OCSP レスポンド、およびその拡張である証明書検証サーバが導入されている。

証明書検証サーバの拡張の特徴は、

- リクエスト
拡張領域 singleRequestExtensions に検証対象者の証明書を設定する。
- レスポンス
拡張領域 singleExtensions に独自のステータスコードを返す。

(5) 拡張領域の特性

証明書の拡張領域では、

- certificatePolicies の critical 値が TRUE である。
- authorityKeyIdentifier、subjectKeyIdentifier、authorityInfoAccess がある。

CRL の拡張領域では、

- issuingDistributionPoint が必須で、critical 値が TRUE である。
- reasonCode、authorityKeyIdentifier、cRLNumber が必須である。

これらは標準仕様のものであるが、既存のアプリケーションでは、ほとんどがサポートができていないようである。

2.2.1.3. 相互接続性

S/MIMEの仕様については1998年にVersion 2のRFCが公開され、続く、1999年にはVersion 3が公開されている。現在は、移行の過渡期であると思われるが、Version 3の仕様は膨大であり、各実装とも、機能の取捨選択が行われている。

現状、S/MIMEを実装した電子メールソフトウェアとして、前述のOutlook Express、Outlook以外にもNetscape社のNetscape Messengerなどがあり、広く流通し利用されている。また、これら製品はVersion 2のRFC以前のInternet Draftの段階で実装され出荷されている。

これらのことから各製品に実装の相違も考えられるため、本開発での成果物についてこれら製品との相互接続性を持つよう留意するとともに、実験フェーズにて相互接続性検証を行う。

2.2.2. 研究開発の内容

S/MIMEはIETF(The Internet Engineering Task Force)の作業部会であるS/MIME Working Groupにて討議され、RFCが策定されている。本研究開発にて開発するソフトウェアはそれらRFCに準ずる形で以下の4つのコンポーネントより構成される。

(1) CMS 処理部

RFC2630「Cryptographic Message Syntax」(略称CMS)に相当する機能を実現する。つまり、S/MIMEで暗号、電子署名を交換するためのデータ形式である、CMSの生成、検証を行う。

(2) 証明書管理部

RFC2631「S/MIME Version 3 Certificate Handling」に相応する。つまり、証明書や失効情報をローカルのデータベースに管理するとともに、リモートのリポジトリにアクセスし、証明書パス構築、検証を行う。

(3) S/MIME メールメッセージ処理部

RFC2633「S/MIME Version 3 Message Specification」に相当する機能を実現する。つまり、電子メール送信時において、CMS処理部で生成したデータをMIME形式の電子メールメッセージとして成型する機能。ならびに、電子メール受信時にいて、受信した電子メールを解析し、CMSのデータ、証明書データを採取する機能を有する。

(4) 外部アプリケーションインターフェイス部

上記の3つのコンポーネントの上位に位置し、それらの機能を前節で述べた「C言語API」および「Outlook ExpressへのGPKI対応の拡張」として外部のアプリケーションに提供するための2つのモジュールである。

§ 利用するライブラリについて

前述のコンポーネントのうち、「CMS処理部」「証明書管理部」に関しては、それぞれ、以下のようなGetronics Government Solutions社が開発したオープンソースのライブラリを流用し拡張することにより実現する。

- CMS 処理部
S/MIME Freeware Library (略称 **SFL**)
- 証明書管理部

Certificate Management Library (略称 **CML**)

- 両コンポーネントに共通

ASN.1 エンコード・デコードライブラリ”**Enhanced SNACC**”

これらは、米国においても、GPKIと同様な連邦認証基盤(FPKI)の構築が進められているが、そこでの検証実験では、リファレンスアプリケーションとしてS/MIMEを想定しており、政府の委託を受けた Getronics Government Solutions 社が開発したライブラリのソースコードを公開しているものである。

また、暗号アルゴリズムを実装したライブラリとしては、名古屋工業大学で開発が行なわれている”**AiCrypto**”を利用する。このライブラリは、純国産でソースコードを公開されており、本研究開発に必要な、共通鍵暗号 (DES、Triple-DES、RC2)、公開鍵暗号 (RSA、DSA、ECDSA) や一方行ハッシュ関数 (MD2、MD5、SHA1、HMAC) が実装されている。

その他に、世界的に見ても実績のある以下のフリーライセンスのライブラリを利用した。

ディレクトリアクセスのために、LDAP APIを実装した **Netscape Directory SDK for C**、あるいは **OpenLDAP** を利用する。

データベースマネジメントライブラリ Sleepycat Software”**Berkeley DB**”

2.2.2.1. CMS 処理部

以下のような機能を実現する。

- (5) 暗号アルゴリズム処理

PKI の礎である公開鍵暗号、ハッシュ関数に加え、S/MIMEにおける暗号化に必要な共通鍵暗号を処理する部位である。

- (6) IC カードアクセス

IC カードの標準インターフェイスを用いて IC カードから、操作者の公開鍵証明書などを取り出し、内包されている秘密鍵を用いて電子署名を行い、あるいは、暗号の復号を行う。

- (7) CMS 生成、検証

暗号化においては envelopedData を、電子署名においては signedData を生成する。また、入力となった CMS データを解析し復号あるいは電子署名の検証を行う。

§ ライブラリ SFL の拡張

SFL では図 2-1 のようなレイヤ構成にて、暗号アルゴリズム処理部分を抽象化することにより、任意の暗号ライブラリを組み込み可能としている。本開発では、以下の 2 種の暗号ライブラリを適用させるための CTIL (暗号ライブラリ抽象化モジュール) を新規開発した。

- ソフトウェア暗号ライブラリ AiCrypto
- IC カードリーダーコントロールライブラリ PKCS#11



図 2-1 ライブラリ SFL の構造

2.2.2.2. 証明書管理部

(8) 証明書ストア管理

証明書や失効リスト (CRL / ARL) をローカルのハードディスクにてデータベースとして管理し、蓄積、保管を行う。

(9) リポジトリアクセス

証明書や CRL / ARL が格納されたりポジトリに LDAP プロトコルでアクセスし、操作時に必要な証明書を収集する。獲得した証明書 / CRL / ARL は証明書管理データベースを用いて保管することも可能とする。

(10) OCSP アクセス

OCSP レスポングに証明書の検証を要求し、そのレスポンスを正しく解釈する。

(11) パス構築、証明書検証機能

上記 3 機能の上位に位置し、証明書、失効情報を収集し、証明書パスを構築するとともに、証明書の有効性の検証を行う。

§ ライブラリ CML の拡張

- 証明書ストア管理部の置換

CML 付属の証明書、CRL データベースでは、アプリケーションとして必要な十分な情報を保持できず、また、秘密鍵の管理機能が欠落しているため、証明書ストアとして機能不足である。

そのため、データベースエンジンとして BerkeleyDB を使うデータベースを新たに定義し、証明

書ストアとして必要な情報を保持しアクセスを可能とする。

- OCSP アクセス機能の追加

CML には OCSP リクエストとしての機能が欠落しているため、OCSP リクエスト、レスポンスの解釈機能を追加する。

2.2.2.3. S/MIME メールメッセージ処理部

「2.2.2.1CMS 処理部」の上位に位置し、電子メールソフトウェアとして基本的な機能である MIME 解析、BASE64 エンコード・デコード、日本語文字コード処理を持ち、RFC2633「S/MIME Version 3 Message Specification」に準じたメールメッセージを生成および解釈する。

2.2.2.4. 外部アプリケーションインターフェイス部

外部アプリケーションに対し以下の2種のインターフェイスを提供する

(1) C 言語 API

「2.2.2.3S/MIME メールメッセージ処理部」と同一のモジュールで実現し、「2.2.2.2 証明書管理部」にエンドユーザに利便性を与える GUI を付与する。API として提供する関数を以下に示す。

- SGS_GetVersion
バージョンを返す。
- SGS_DisplayAbout
アバウトダイアログを表示する。
- SGS_Init
セッションを初期化する。
- SGS_Release
セッションを開放する。
- SGS_ShowOption
動作環境を示すダイアログを表示する。
- SGS_Encrypt
通常の電子メールメッセージを入力として、暗号化および電子署名をし、S/MIME 形式の電子メールメッセージを出力します。状況に応じた GUI を持つ。
- SGS_Decrypt
受信した電子メールメッセージを入力として、復号および電子署名の検証を結果を出力します。状況に応じた GUI を持つ。
- SGS_SMimeCaManager
認証局の証明書管理画面を表示する。
- SGS_SmimeCertManager
証明書管理画面を表示する。
- SGS_SmimeEnvironment
セキュリティの設定画面を表示する。

- SGS_SMimeGenerateCSR
証明書署名要求のデータ(PKCS#10)を生成し、状況に応じて認証局への電子メールメッセージを出力する。
- SGS_GetDecryptProcess
最新の復号処理の結果を返す。
- SGS_GetLastErrorCode
最新のエラーを返す。

§ コンポーネントの統合

各コンポーネントを統合し、「C言語API」を実現する、モジュール構成を図 2-2 モジュール構成に示す。

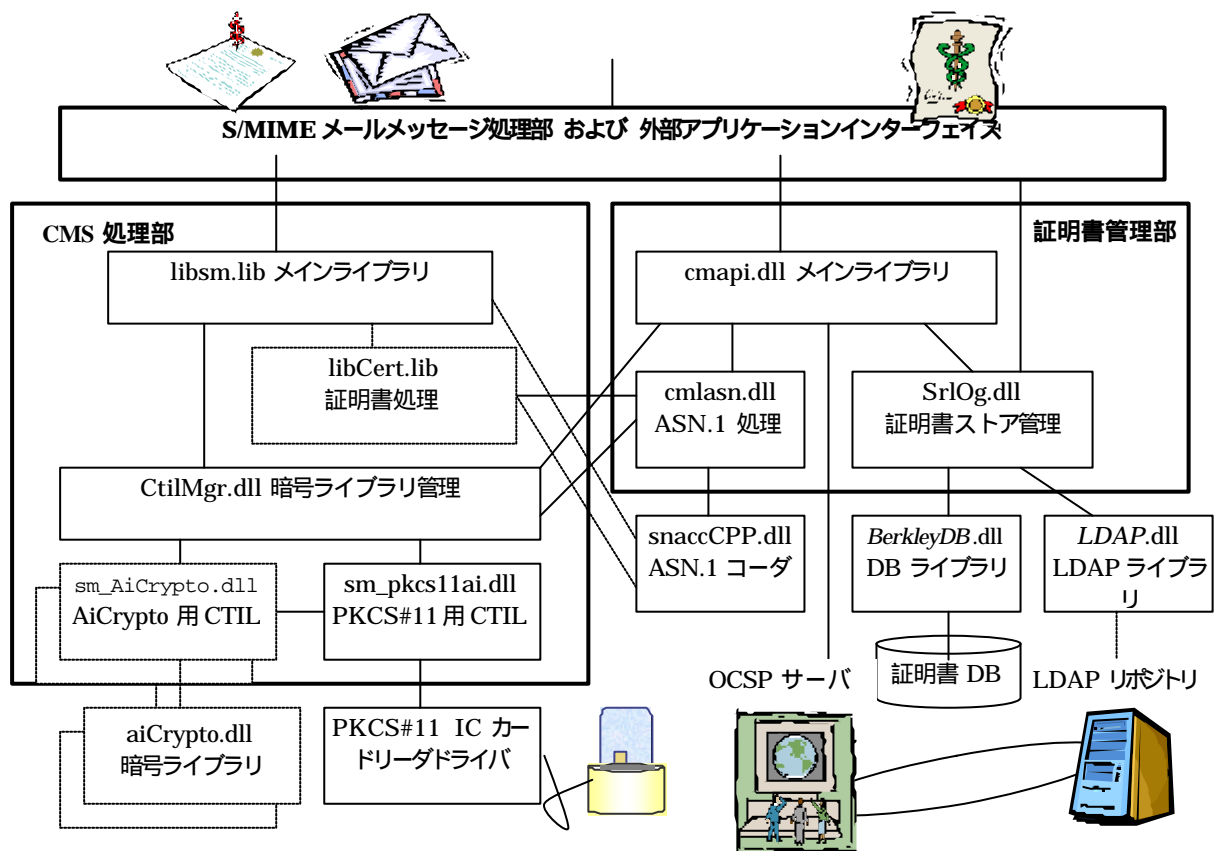


図 2-2 モジュール構成

(2) Outlook Express への GPKI 対応の拡張

Microsoft windows ではセキュリティのプラットフォームとして CryptoAPI が用意されている。Outlook Express など Windows 標準のアプリケーションは CryptoAPI を通して、証明書管理や暗号、電子署名などの機能を実現している。

また、証明書の失効検証について、CryptoAPI では図 2-3 CryptoAPI における失効検証機構のように標準の失効検証ルーチン以外に開発者が、“Revocation Provider” と呼ばれる任意の失効検証モジュールを追加することが可能となっている。アプリケーションが失効確認のための関数「CertVerifyRevocation」を呼び出すと、登録されている“Revocation Provider”が動作し、検証結果をアプリケーションに返すわけである。

本研究開発ではこの機構を利用し、GPKIの公開鍵証明書を正当に取り扱うための“Revocation Provider”を開発する。これにより、Outlook ExpressがGPKI対応したと考える。このとき、

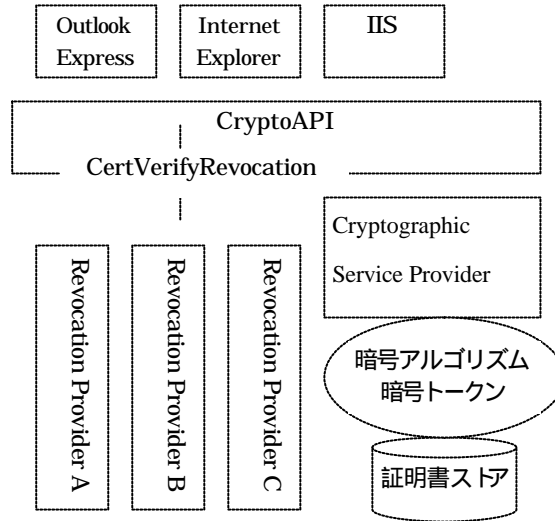


図 2-3 CryptAPIにおける失効検証機構

以下に各コンポーネントを統合し機能を実現しているかを図 2-4 に示す。

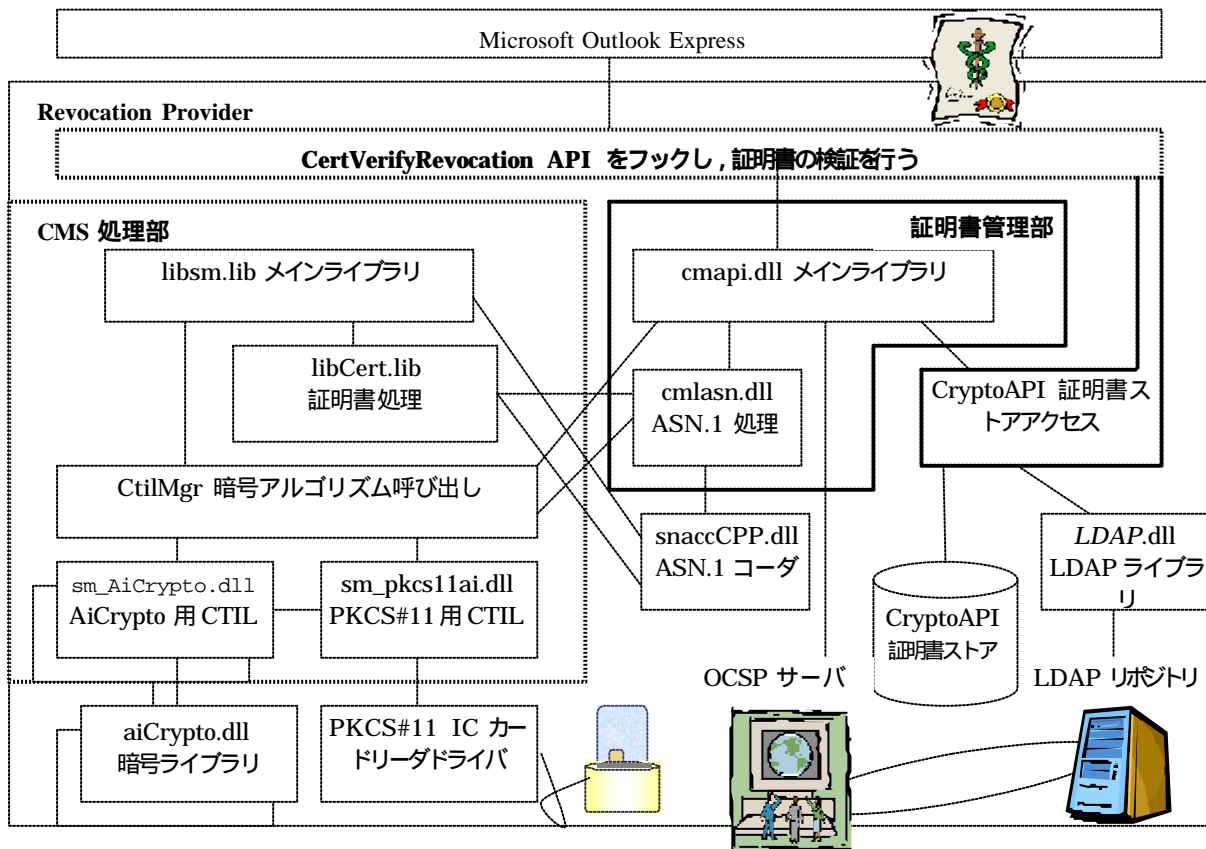


図 2-4 Microsoft Outlook Express 対応のモジュール(Revocation Provider)構成

2.3. 本年度の活動状況

2.3.1. 調査活動

本研究開発での、ソフトウェアの開発に関連する以下の事項について調査を行った。

2.3.1.1. 電子メールクライアント関連調査

プラグインを開発する対象となる電子メールソフトウェア Microsoft 社の電子メールクライアントである Outlook および Outlook Express の外部アプリケーション インターフェイスについて調査し、本開発に反映した。

2.3.1.2. OCSP 関連調査

IETF においては、OCSP (Online Certificate Status Protocol) の仕様が RFC 2560 で規定されて以降も、関連技術の検討が該当部会 (Security Area PKIX-WG) において進行中である。この動向についても継続的に調査するとともに、これを実装した製品の状況について調査し、本開発に反映した。

2.3.1.3. IC カードリーダー調査

調達仕様書によるとインターフェイスとして PKCS#11 が実装された IC カードリーダーが GPKI にて利用されるわけであるが、その他に IC カードリーダーのインターフェイス PC/SC と JavaCard についても視野にいれ、これらの実装状況とソフトウェアへの適応方法について調査し、本開発に反映した。

2.3.1.4. GPKI 仕様の調査

総務省から発行されている文献に基づいて GPKI における標準プロトコルの利用方法ならびにその拡張機能について調査し、本開発に反映した。

2.3.2. ソフトウェア開発

2.3.2.1. CMS 処理部におけるライブラリ SFL の拡張

2つの CTIL (暗号ライブラリ抽象化モジュール) 新規開発し、以下の暗号ライブラリを利用可能とした。

- ソフトウェア暗号ライブラリ AiCrypto
- IC カードリーダーコントロールライブラリ PKCS#11

また、既存製品との相互接続性の検証の結果、encryptedContent(平文を共通鍵暗号で暗号化した部分)のエンコード形式など、いくつか、互換性のないケースが存在することがわかり、互換性を保持するよう「CMS生成、検証」部を修正した。また、共通鍵暗号での暗号化、復号の処理が冗長で処理時間が妥当性を欠いていたので、これを修正し、妥当な処理時間とした。

2.3.2.2. 証明書管理部におけるライブラリ CML の拡張

(1) 証明書ストア管理

独自のデータベースによる管理を可能とし、メールアドレスと証明書との関連付け、秘密鍵の管理機能などアプリケーションの機能に柔軟性を与えたるものとした。

(2) OCSP アクセス機能

OCSP クライアントのリクエストの形式には電子署名付きリクエストと電子署名のないリクエストの2つの形式があるが、本年度の開発では電子署名なしのリクエストのみをサポートした。

2.3.2.3. S/MIME メッセージ処理部

ほぼ必要と考えられる一通りの機能を開発した。

懸念されることがあるとすれば、文字コード変換機能にある。Windows 環境では、Shift-JIS コードにて日本語文字が取り扱われているが、電子メールの世界では、JIS を基本とする。そのため、本開発でも Shift-JIS から JIS への変換機能を取り入れているがその逆はない。外部アプリケーションにとって、電子メール受信時に復号結果の本文を Shift-JIS に変換してから出力することが必要になるものもあるかもしれない。

2.3.2.4. 外部アプリケーションインターフェイス部

(1) C 言語 API

本年度、「2.2.2.4 外部アプリケーションインターフェイス部」に記載した一連の関数を実装した。これによって、電子メールソフトウェアの開発者が新たに S/MIME プロトコルを実装することなく、利用できる、必要な機能が全て提供できると考える。

(2) Outlook Express "GPKI 対応プラグイン"

「2.2.2.4 外部アプリケーションインターフェイス部」に記載した Revocation Provider を開発し、証明書検証時に、ディレクトリサーバから認証局証明書、相互認証証明書ペア、CRL、ARL の取得、OCSP 検証の機能を実現した。

2.3.3. 検証実験

以下に大別する 2 通りの検証実験を通し、本研究開発の主たる成果物であるソフトウェアの有効性が検証できた。

2.3.3.1. C 言語 API の S/MIME 機能検証実験

C 言語 API をサポートした(株)オレンジソフトの電子メールソフト"Winbiff"との連携による S/MIME 機能について検証実験を行った。検証項目を大別すると以下の 3 項目となる。

- CMS の生成・解析
- 証明書、CRL の処理
- S/MIME メッセージの生成・解析
- RSA 鍵ペア生成と PKCS#10 生成、自己署名証明書生成

2.3.3.2. C 言語 API と Outlook Express "GPKI 対応プラグイン" の模擬 GPKI 環境での検証実験

GPKI における環境を想定した以下のサーバ群を持つ模擬実験環境を構築し、研究開発の成果物である 2 種類の実行単位について、それぞれ、動作の妥当性を検証した。

- 府省、民間の模擬環境でのサーバ
認証局
リポジトリ
OCSP
メールサーバ
- 総務省ブリッジ CA の模擬環境でのサーバ
ブリッジ CA
リポジトリ
また、テストデータとして公開鍵証明書は「政府認証基盤相互運用性仕様書」に沿って作成し、

作成した証明書、CRLのプロファイルに従い正しく動作することを検証した。

2.3.3.3. S/MIME 機能の相互接続性の検証

C 言語 API をサポートした電子メールソフト "Winbiff" と、他の市販の S/MIME をサポートした 2 つの電子メールソフトウェアとの間で、S/MIME に係る相互接続性の検証を行い、開発にフィードバックした。

- C 言語 API を利用した "Winbiff"
- " GPKI 対応プラグイン " をアドオンした Outlook Express
- Outlook
- Netscape 社の Netscape Messenger

2.4. 外部発表及び成果物

2.5. 外部発表

本研究開発の内容を含んだ発表として以下を行った。

澤野「PKI、GPKI の電子メールでの利用」を日本 PKI フォーラム「第 8 回 PKI セミナー」にて発表

科学技術交流財団「PKI 利用推進研究会」にて発表予定

2.6. 今後の課題

2.6.1. 普及

一般的に、インターネットに対するセキュリティの意識は高まっているようだが、未だ、電子メールに関するセキュリティの意識は低く、守秘や真正性が要求される場面でも、セキュアではない電子メールにて情報の伝達が行われている。そのギャップを埋めるものとして、セキュアな電子メール環境の普及が望まれる。オープンなコミュニケーションである電子メールであるため、多様な電子メールソフトウェアが使われている。中には、API の公開などプラグインインターフェイスを持つ電子メールソフトウェアがある。本開発を継続し、それらへの積極的な対応を図り、普及の一助としたい。考えられる電子メールソフトウェアとしては、

- Microsoft 社 Outlook
 - Mozilla
 - QUALCOMM 社 Eudora
 - シェアウェア Becky!
 - シェアウェア 鶴亀メール
- などである。

2.6.2. IC カード

本研究開発では、PKCS#11 ドライバを通して IC カードを取り扱うことを前提としていたが、市場に流通している製品は Windows 環境においては、標準の PC/SC インターフェイスを前提として提供されており、PKCS#11 に関して、未実装の関数があるなど、仕様のサポート状況にはばらつきがあり、検証実験では単一の機種を対象とした。継続し開発し、利用可能な IC カードリーダを増やしたいが、その際には、PC/SC ドライバでの利用も検討事項としたい。

本テーマでは、IC カード内に利用者の秘密鍵、公開鍵証明書が作成されている上での利用を前提とした。今後、IC カードのドライバを通した、鍵生成、証明書署名要求データの生成機能の開発が課題となる。

2.6.3. 証明書検証サーバ

GPKI で実際に利用されている証明書検証サーバは標準のOCSP の仕様に独自の拡張をおこなったものであるが、一般の開発者が容易に検証実行を行える体制にはない。そのため、本開発ではOCSP サーバとしてOCSP バージョン1であるRFC2560に準拠した製品を対象として開発し検証を行った。継続する年度の開発にて、証明書検証サーバのprotocolsに則した開発を行い、検証用の模擬サーバの開発、あるいは実環境での検証を目標としたい。

2.6.4. C言語API

本開発では、強力な証明書検証機構を持ったAPIを開発できた。しかし、本開発ではクライアントサイドの電子メールソフトウェアを対象として、S/MIME対応のための関数のみを実装した。しかし、PKI、GPKI対応アプリケーションは電子メールに限らない。また、クライアントサイドに限らず、この証明書検証機構は有効と思われる。電子メールの範囲にとどまらない汎用的な開発の追加は有効であると考えられる。

2.6.5. CryptoAPI の Revocation Provider

この失効検証機構はCryptoAPIに組み込まれるもので、CryptoAPIを利用する全てのアプリケーションで利用可能なものである。例えば、SSL 証明書の検証、MS Office文章やマクロの電子署名などでも、この機構が利用される。本テーマでは電子メールソフトウェアを対象として開発したが、これら他のアプリケーションでの有効な利用方法についても検討してみたい。

2.6.6. S/MIMEでの暗号アルゴリズム

本年度の開発ではS/MIMEにおける電子署名の公開鍵アルゴリズムとしてRSAのみを対象とし、検証実験においてもRSAのみについて実験を行った。DSS署名の生成に関し、利用可能性の検証が残っている。

暗号化の公開鍵アルゴリズムとしては、Diffie-Hellmanには対応していない。今後の課題である。

2.7. まとめ

本年度はGPKIにおける要件を満たし、市販の流通する電子メールソフトウェアがGPKI環境にて十全に機能できることを可能とした。これをインターネットに公開し、PKI、GPKI普及の一助としたい。

また、対象とした電子メールソフトウェアの普及率もトップクラスのものであるが、電子メールというコミュニケーションのツールを対象とした開発ゆえ、双方のツールにて、同様の機能を持つことが前提である。サポートする電子メールソフトの追加、セキュリティデバイスの追加、プラグインの汎用化などとともに、実環境での動作検証も視野にいれ、継続する年度の開発を提案したい。

2.8. 参考文献

- 総務省行政管理局政府認証基盤(GPKI)基本問題専門部会承認「政府認証基盤相互運用性仕様書」
<http://www.gpki.go.jp/session/index.html>
- RSA Laboratories PKCS 仕様
<http://www.rsasecurity.com/rsalabs/pkcs/>
- Cryptographic Message Syntax
<http://www.ietf.org/rfc/rfc2630.txt>
- S/MIME Version 3 Certificate Handling RFC2632
<http://www.ietf.org/rfc/rfc2632.txt>
- S/MIME Version 3 Message Specification RFC2633
<http://www.ietf.org/rfc/rfc2632.txt>
- PKI Online Certificate Status Protocol – OCSP
<http://www.ietf.org/rfc/rfc2560.txt>
- S/MIME Freeware Library (SFL)
http://www.getronicsgov.com/hot/sfl_home.htm
- データベースマネジメントライブラリ：Sleepycat Software “Berkeley DB”
<http://www.sleepycat.com/docs/index.html>
- 名古屋工業大学電気情報工学科岩田研究室 暗号ライブラリ“AiCrypto”
<http://mars.elcom.nitech.ac.jp/security/aicrypto.html>