組織における認証局システムの構築と運用
― 認証局の構築と運用技術 ―

河北隆生*・岡嶌崇*・富松篤典**

Development and Operation of Certification Authority System for PKI in the Organization

- Development and Operation Technology of Certification Authority System -

Takao KAWAKITA*, Takashi OKAJIMA* and Atsunori TOMIMATSU**

組織での利用を対象としたPKI(公開鍵暗号基盤)の運用に必要不可欠なX.509公開鍵証明書を発行する認証局システムを構築した。本システムの特徴は、UI(ユーザインターフェース)を用いてPKCS#12ファイルをより安全に末端ユーザなどのエンドエンティティへオンライン配布するため、ユーザ設定パスワードに加えてRA(登録局)が発行したチャレンジパスワードの入力、パスワード不正入力回数制限、PKCS#12ファイルの取得回数制限と取得期間制限などの機能を持つことである。このことで、秘密鍵とX.509公開鍵証明書のより安全かつ簡易なオンライン配布を可能とした。

本システムで発行した秘密鍵とX.509公開鍵証明書をPKI対応のWebサーバとブラウザおよび電子メールなどのソフトウェアに組み込んだ。その結果、Web閲覧ではサーバとユーザ認証およびデータ暗号化、電子メールでは送信時の電子署名と暗号化、受信時の復号と署名の検証、データベースシステムなどでは暗号化が可能となった。

本報告では、構築した認証局システム、秘密鍵とX.509公開鍵証明書の発行手順と失効手順、ネットワーク構成について述べ、認証局システムの運用とシステムへの適用について考察する。

1. はじめに

企業などの組織では、ネットワークを構築し、組織内あるいは組織間での情報交換、共有、提供などに利用している。これらの情報には、機密情報や重要なデータなども含まれており、情報漏えい、改ざん、なりすましなどの危険性が指摘されているため、強固な認証や暗号化通信が望まれている。

一方、近年PKI(Public Key Infrastructure、公開鍵暗号基盤)が、標準化されつつある。PKIは、情報の暗号化による機密性、個人や情報の発信元を検証する認証、情報の改ざんなどから保護する完全性、否認防止を提供する[1,2]。そこで、PKIに対応することで上記要求が解決できる。

最近では、PKIに対応した電子メールやWebサーバ、ブラウザ、VPN(Virtual Private Network)、あるいはPKIの運用に必要不可欠なX.509公開鍵証明書(ISOで標準化されたX.509に準拠した公開鍵証明書)を発行する認証局システムなども製品として提供されるようになってきており、今後ますますPKIに対応したシステムが増えると予想される。それに伴い、各組織では、認証局システム運用の必要性が高まると考えられる。

しかし、認証局システム製品は高価であること、X.509公開鍵証明書などの簡易かつ安全な配布方法がまだ十分でないことなどの課題がある。

そこで、筆者らは、組織での利用を対象とした認証局システムを構築した[3,4]。本システムの特徴は、秘密鍵とX.509公開鍵証明書をより安全かつ簡易に末端ユーザなどのエンドエンティティへオンラインで配布可能としたことである。

本報告では、構築した認証局システム、秘密鍵とX.509公開鍵証明書の発行手順と失効手順、ネットワーク構成について述べ、認証局システムの運用とシステムへの適用について考察する。

2. 全体システム概要

公開鍵暗号方式では、秘密鍵と公開鍵と呼ばれる鍵ペアを用いる。この鍵ペアの特性は、一方の鍵で暗号化されたデータは他方の鍵でしか復号できないことである。秘密鍵を所有者のみが保持し、誰でも取得可能な公開鍵の所有者を特定できればデータの暗号化、電子署名、検証を行うことが可能となる。

図1に構築した認証局システム、X.509公開鍵証明書(以下、「証明書」と呼ぶ)の発行とその利用の関係を図示したPKI概要図を示す。信頼されている認証局は、エンドエンティティ(Webなどのサーバや個人などのユーザ、以下「EE」と呼ぶ)から証明書発行依頼を受けると、EEの本人性を確認後、EEの秘密鍵と公開鍵を作成するとともに、その公開鍵に発行者と所有者などの情報を付加して認証局の秘密鍵で電子署名した証明書を作成する。その後、

* 情報デザイン部
** (株) 電盛社
問い合わせ先：tkawakit@kmt-iri.go.jp

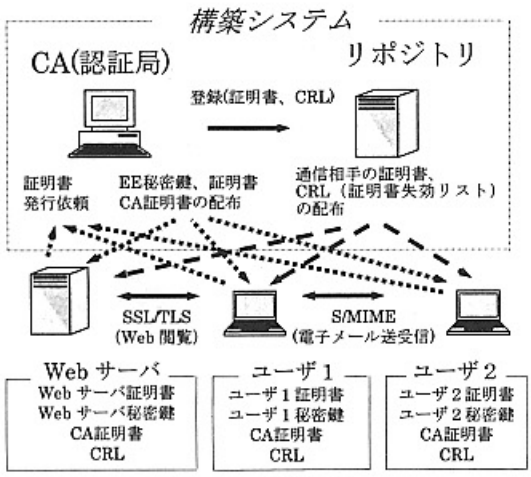

図１　PKI概要図

認証局は、EEの秘密鍵と証明書、認証局の証明書をEEに配布するとともに、証明書をリポジトリ(Repository、配布された証明書などの保管と配布を行うデータベース)に登録する。EEは、通信相手の証明書をリポジトリから取得できる。

つまり、認証局は、証明書を作成することでEEの公開鍵の所有者を保証する。また、EEが、認証局の証明書を所有することで、その認証局を信頼する。EEは、通信相手の証明書の電子署名を検証することで、信頼した認証局が発行した証明書かどうかがわかる。通信相手が、信頼した認証局が発行した証明書に含まれる公開鍵と対応した秘密鍵を持っていることで、通信相手を特定できる。ゆえに認証局は、例えば偽の証明書を発行することがないなどの不正をしない絶対的信頼を持った組織または組織内の部署が運用する必要がある。

秘密鍵が漏えいした場合などは、EEは認証局に証明書失効依頼を行う。認証局は、失効した証明書番号を記載したリストに発行者情報などを付加して秘密鍵で電子署名した証明書失効リスト(Certificate Revocation List、以下、「CRL」と呼ぶ）を作成後、リポジトリへ登録する。EEは、リポジトリからCRLを取得することで、失効した証明書を確認することができる。

Web閲覧の場合、Webサーバとユーザ1のブラウザでは、通信プロトコルであるSSL(Secure Sockets Layer)[5]やTLS(Transport Layer Security)[6]で相互に証明書を交換、検証後、暗号化通信を行う。

電子メール送受信の場合、ユーザ1は、自分の秘密鍵で電子署名後、自分の証明書とともに、ユーザ2の証明書に含まれる公開鍵で暗号化したS/MIME(Secure Multipurpose Internet Mail Extensions)[7,8]形式でユーザ2へ送信する。ユーザ2は、受信したメールを自分の秘密鍵で復号した後、送信者の電子署名の検証、改ざんの有無をユーザ1の証明書に含まれる公開鍵を用いて確認する。なお、電子メール送信相手の証明書は、既にS/MIME形式で電子メールを交換している場合は保持しているが、保持していない場合にはリポジトリから取得することとなる。

PKIに対応したシステムとしては、この他にもIPsec(IP Security Protocol)[9]を利用したVPNなどがある。

3. 認証局システム

3.1 システム概要

構築した認証局システム構成と証明書発行手順を図２に、開発環境を表１に示す。図２においてEEは、厳密にはWebなどのサーバまたは個人などのユーザであるが、ここではサーバの場合サーバ管理者とする。

本システムでは、CA(Certification Authority、認証局)の機能の一部であるEEとのインターフェースをRA(Registration Authority、登録局)、UI(ユーザインターフェース)に分割し、リポジトリを加えた4つのサブシステムから構成される。これらのサブシステムの機能、

表１　開発環境

OS	FreeBSD4.5-RELEASE
開発言語	PHP4.0.6 , perl5.005
暗号化ツール	OpenSSL0.9.6b
Web サーバ	Apache1.3.24+mod_ssl-2.8.8
データベース	PostgreSQL7.1.3
LDAP	OpenLDAP2.0.23

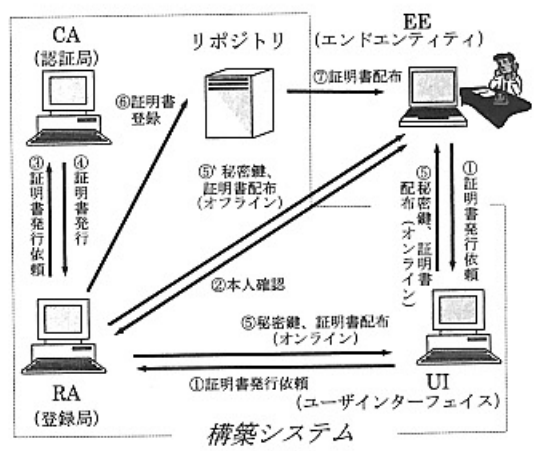

図２　認証局システム構成と証明書発行手順

証明書発行手順、失効手順およびその関係については、3.2、3.3および3.4で述べる。

また、本システムは、暗号化ツールとしてOpenSSLを使用しており、X.509公開鍵証明書フォーマットはVersion3形式、証明書失効リストはVersion1形式、公開鍵暗号方式はRSA、DSA、DHに対応している[10]。

3.2 証明書発行手順

本システムでは、Webサーバなどが使用するサーバ証明書と個人が使用するクライアント証明書を発行できる。ここでは、図２に対応して証明書発行手順と各サブシステムの機能およびその関係を述べる。なお、本報告では、証明書の発行依頼からEEへの証明書配布までの一連の流れを発行、EEへ証明書を渡すことを配布と定義する。

(1)証明書発行依頼(①)

発行依頼は、(a)EEが直接入力する、(b)EEから申請を受けたRAオペレータが入力する場合の2通りの方法がある。EEまたはRAオペレータは、ブラウザを使用してUIへアクセスし、発行依頼を行う。

クライアント証明書発行依頼画面例を図3に示す。入力データは、ユーザ設定パスワードに加えてクライアント証明書の場合は個人情報、サーバ証明書の場合はサーバの情報などである。これらのデータは、UIのデータベースへ登録される。

図3　クライアント証明書発行依頼画面例

(2)EE確認(②)

RAオペレータは、証明書発行依頼データをUIのデータベースからRAのデータベースへオンラインで取得後、EEを確認する。EEの確認方法は、例えば本人への電話、面接、社員証の提示などが考えられる。なお、この時点で情報漏えい防止のため、UIのデータベースからEEのデータを削除する。

(3)秘密鍵、CSRなどの作成

RAオペレータは、EEの秘密鍵、証明書署名要求(Certificate Signing Request、以下「CSR」と呼ぶ)、ランダムなチャレンジパスワード(EEの秘密鍵や証明書を取得するために必要なパスワード)を作成し、RAのデータベースへ登録する。なお、ここで作成されるチャレンジパスワードの利用は、3.3証明書配布で述べる。

(4)証明書の作成(③、④)

CAオペレータは、RAデータベースからCSRを取得後、CAの秘密鍵で電子署名した証明書を作成し、CAのデータベースに保存するとともにRAのデータベースへオンラインで格納する。

(5)証明書配布(⑤、⑤')

RAオペレータは、証明書をEEへ配布するが、本システムでは、(a)UIによるオンライン配布(⑤)、(b)フロッピーディスクなどの媒体によるオフライン配布(⑤')、(c)スマートカードによるオフライン配布(⑤')の3通りの方法が可能である。詳細は、3.3 証明書配布で述べる。

(6)証明書のリポジトリへの登録と配布(⑥、⑦)

RAオペレータは、EEの証明書をリポジトリへ登録する。また、EEは、必要に応じてリポジトリから通信相手の証明書を取得する。

なお、本システムでは、証明書の配布にはLDAP(Lightweight Directory Access Protocol)[11]、CRL配布にはWebサーバで使用されるhttpとhttpsの通信プロトコルを使用した。

3.3 証明書配布

本システムでは、3通りの方法で証明書を配布できる。ここでは、その方法について述べる。

3.3.1 UIによるオンライン配布

RAオペレータは、EEへ電子メールあるいは郵送などで証明書作成完了通知を送付する。証明書作成完了通知例を図4に示す。

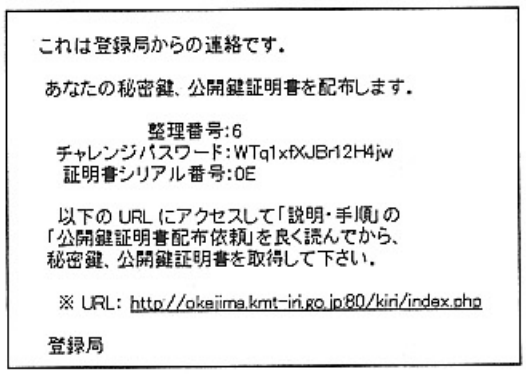

図4　証明書作成完了通知例

RAオペレータは、EEの秘密鍵と証明書、CAの証明書を一つにまとめて暗号化したPKCS#12[12]ファイルを作成するとともに、EEが入力したユーザ設定パスワードとRAが作成したチャレンジパスワードを一方向関数(MD5)で変換後、UIのデータベースへネットワーク経由で格納する。ユーザ設定パスワードとチャレンジパスワードを一方向関数で変換するのは、不正侵入によりこれらのパスワードがUIから不正取得された場合にパスワードの推測を困難にするためである。

EEは、UIへネットワーク経由でアクセス後、ユーザ設定パスワード、チャレンジパスワードに加えてサーバ証明書の場合はホスト名、クライアント証明書の場合は電子メールアドレスなどを入力することでPKCS#12ファイルを取得できる。クライアント証明書配布の入力画面例を図5に示す。

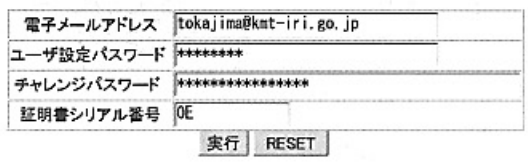

図5　クライアント証明書配布の入力画面例

本システムでは、パスワード不正入力回数制限、PKCS#12ファイルの取得回数制限およびPKCS#12ファイルの取得期間制限の機能がある。これらの機能は、設定した回数パスワードを間違えた場合あるいは設定した回数取得した場合は証明書配布プログラムで、また設定した期限を過ぎた場合にはUNIX(FreeBSD)のcron機能により定

期的にプログラムを起動することで、自動的にUIのデータベースからPKCS#12ファイル、一方向関数(MD5)で変換したユーザ設定パスワードやチャレンジパスワードなどの証明書配布に必要なデータを削除する。このことで安全性をより強化した。

また、EEが取得したPKCS#12ファイルから秘密鍵や証明書などをブラウザや電子メールなどのソフトウェアに取り込む時には、解凍用パスワードとしてユーザ設定パスワードとチャレンジパスワードの組み合わせが必要となる。このことでPKCS#12ファイルが漏えいした場合、解凍パスワードを推測しにくくした。

3.3.2 媒体でのオフライン配布

RAオペレータは、EEへPKCS#12ファイルが納められたフロッピーディスクなどの媒体とともに証明書作成完了通知を郵送や直接手渡すなどのネットワークを使用しない方法で配布する。

3.3.3 スマートカードによるオフライン配布

RAオペレータは、EEの秘密鍵と証明書が格納されたスマートカードとともに証明書作成完了通知をネットワークを使用しない方法で配布する。

今回使用したスマートカードは、USBに接続するALADDIN社製eTokenである[13]。本スマートカードは、Windows98SE以上のPKCS#11[14]に対応するNetscape 、Microsoftが提供する暗号化モジュールであるCryptoAPIに対応するInternet Explorer、Outlook Expressなどが標準で使用可能である。また、スマートカードに格納された秘密鍵と証明書は、パスワードで保護されており、暗号化通信や認証時にスマートカードのパスワードが要求される。本システムでは、ユーザ設定パスワードをスマートカードのパスワードに設定した。今回使用したスマートカードの外観と装着例を図6に示す。

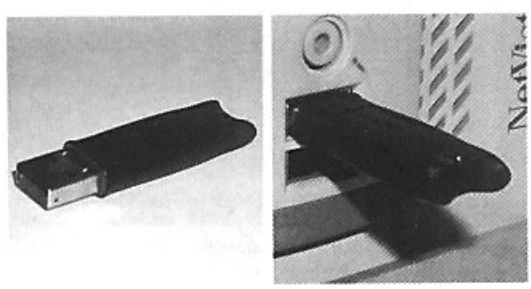

図6 スマートカード
(左) 外観 (右) パソコンのUSBポートへの装着例

3.4 証明書失効手順

(1)失効依頼

EEまたはEEから申請を受けたRAオペレータは、ブラウザを使用してUIへアクセス後、証明書失効依頼を入力する。

(2)失効操作

RAオペレータは、UIから失効依頼データを取得し、CAへ失効を依頼する。CAオペレータは、RAから該当する証明書を取得し、失効処理を行い、失効した証明書番号のリストを作成する。その後、CAオペレータは、そのリストに発行者の情報などを付加後、CAの秘密鍵で電子署名を行うことでCRLを作成し、そのCRLをRAのデータベースに格納する。

(3)リポジトリ操作

RAオペレータは、該当するEEの証明書をリポジトリから削除するとともにCRLを登録する。

なお、CRLには、有効期限があるが、長期間失効処理がないとCRLが更新されないため、本システムではUNIXのcronの機能により定期的にプログラムを起動することでCAがCRLを自動的に作成し、リポジトリへ登録する機能を持たせた。

3.5 ネットワーク構成

本システムでは、次のデータを保持している。

(1)RA : EEの個人情報や秘密鍵など

(2)CA : CAの秘密鍵

(3)UI : EEの個人情報やホスト情報、PKCS#12ファイルやMD5で変換したユーザ設定パスワードとチャレンジパスワード

これらのデータは、非常に重要なデータであるため、外部へ漏えいしないように厳密に管理される必要がある。また、CA、RA、UI、リポジトリが、相互にネットワーク経由でデータ交換できることで効率的な作業が可能となる。そこで、セキュリティを考慮したネットワーク構成とマシン管理が必要となる。EEから証明書発行依頼をオンラインで受け付けるとともに、EEへ秘密鍵と証明書をオンラインで配布する場合のネットワーク構成例を図7に示す。この場合、UIとリポジトリは、EEとの通信が必要なため、公開されたネットワークと隔離されたネットワークの双方に接続された境界マシンに置かれる。また、RAとCAは、隔離されたネットワークに置かれる。

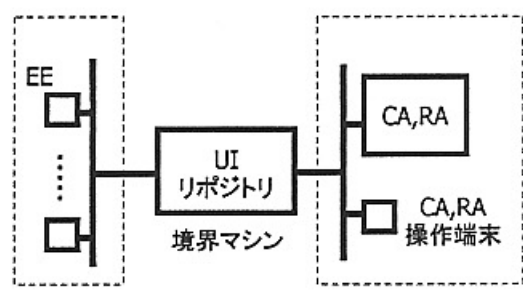

図7 ネットワーク構成例

EEの申請を受けてRAオペレータが証明書発行依頼をオフラインで行うとともに、EEへ秘密鍵と証明書をフロッピーディスクなどの媒体やスマートカードでオフライン配布する場合は、UIを隔離されたネットワークに設置できる。他にも、例えばEEから証明書発行依頼をオンライ

ンで受け付け、EEへ秘密鍵と証明書をオフライン配布する場合は、UIの一部の機能を境界マシンに置くことになる。

また、各マシンでは、例えば次のようなセキュリティ対策も必要となるが、詳細は割愛する。

(1)不要なサービスの停止
(2)サービスに接続できるマシンの限定
(3)パケットフィルタリングなどによる不要なアクセスの禁止
(4)各マシン間のデータ交換や各マシンへのアクセスにPKIによる認証と暗号化の利用
(5)公開されたネットワークと隔離されたネットワーク間のルーティングの禁止

更に、PKIシステムでは、証明書やCRLの有効期限などには正確な時刻を記載することが要求される。そこで、CAやRAには、境界マシンを経由してntp(Network Time Protocol、時刻同期プロトコル)で当センターが運用するstratum 1サーバ(正確な時刻を持つ最上位のマシン)から時刻情報を取得した。

4. 考察

4.1 認証局信頼モデル

筆者らは、PKIソフトウェアとして現在良く利用されているWeb サーバとブラウザの組合せ、電子メールなどを対象として、組織で認証局を運用する場合の認証局信頼モデルを実験などで確認し、検討した。

その結果、複数のroot CA(最上位のCA)を運用する相互認証モデルやブリッジ認証モデルでは、機能しないPKIソフトウェアがあった。また、階層的信頼モデルでは、例えばCRLなどを複数のリポジトリから取得する必要があるなどの手間の問題もあった。そこで、我々は、一つのroot CAから直接EEへ証明書を発行するモデルを採用した。

なお、将来OCSP(Online Certificate Status Protocol、証明書をリアルタイムに検証するプロトコル)などのVA(Validation Authority、検証局)での検証に対応するソフトウェアが増加すれば、さまざまな認証局信頼モデルが簡単に構築できると考えられる。

4.2 UIにおけるサーバ認証

UIのWebサーバには、自己署名したroot CAから発行された秘密鍵とサーバ証明書を組み込んだ。これは、ネットワーク上のデータ送受信を暗号化するためである。

EEが、ブラウザを使用してUIへ証明書発行依頼を行うが、この時EEのブラウザではサーバ認証も行う。この時点では、まだEEのブラウザにCAの証明書が組み込まれていないため、UIのサーバ証明書は信頼できない状況となる。

これを回避するためには、例えば以下の方法が考えられる。

(1) ブラウザに組み込まれたCAから発行された秘密鍵とサーバ証明書のサーバへの組み込み
(2) 自己署名されたCA証明書のオフラインでの配布、ブラウザへの組み込み

前者の場合は、費用負担が発生するが、配布の手間は省かれる。後者の場合は、会社などの組織では、各部署の情報担当者などを経由して配布すれば、手間は少ないと考えられる。ゆえに後者の方法でも組織で運用する場合には、十分機能すると思われる。

4.3 UIによる秘密鍵、証明書のオンライン配布

本システムでの証明書配布は、(a)UIによるオンライン配布、(b)フロッピーディスクなどの媒体によるオフライン配布、(c)スマートカードによるオフライン配布の3通りの方法が可能である。ここでは、本システムの特徴であるオンライン配布について考察する。

オンライン配布では、EEはUIからユーザ設定パスワード、チャレンジパスワード、証明書番号およびサーバ証明書の場合はホスト名、クライアント証明書の場合は電子メールアドレスの入力が必要である。チャレンジパスワード、証明書番号は、証明書作成完了通知によりEEへ通知される。

証明書作成完了通知が他人に漏えいし、その情報を元にPKCS#12ファイルを不正取得を試みる場合を考える。この場合は、ユーザ設定パスワードは本人のみしか知らないため、ブルートフォース攻撃(パスワード総当たり攻撃)が行われる。しかし、パスワード不正入力回数制限により設定回数を越えたパスワードの入力間違いが発生した場合、証明書配布プログラムが自動的にUIのデータベースからPKCS#12ファイル、一方向関数で変換したユーザ設定パスワードやチャレンジパスワードなどの配布に必要なデータを削除するため、PKCS#12ファイルの取得が困難となる。また、本システムでは、PKCS#12ファイルの取得回数制限と取得期間制限を設けており、設定回数取得した場合あるいは設定期限を過ぎた場合、自動的にUIのデータベースから配布に必要なデータが削除される。このため、さらにPKCS#12ファイルの取得が困難になる。

次にUIへの不正侵入が発生した場合を考える。前述したPKCS#12ファイルの取得回数制限と取得期間制限によりUIのデータベースから不必要と考えられる配布データは削除する。このため、発行したPKCS#12ファイルの一部は不正取得されるが、全てのPKCS#12ファイルの不正取得は防げることとなる。

なお、配布時のPKCS#12ファイルの不正取得を防ぐためには、EEは推測されやすいユーザ設定パスワードを設定しないこと、UIマシンのセキュリティ管理を強固にすることが重要となる。

また、媒体によるオフライン配布は、途中に介在する人間によるデータの抜き取りや紛失などが発生する場合も考慮して安全な配送経路を構築する必要がある。しかし、UIによるオンライン配布は、オフライン配布のよう

な配送経路の構築は必要がないため、簡易に配布できると考えられる。

4.4 システムへの適用

本システムで発行した秘密鍵と証明書をシステムへ組み込み、機能を確認した。

Web 閲覧では、Webサーバにはmod_ssl2.8.8を組み込んだApache1.3.24、ブラウザにはInternet Explorer(5.5および6.0)とNetscape(4.78および6.2)を使用した結果、サーバ認証とユーザ認証およびデータ暗号化が可能であった。

電子メールの送受信では、Outlook Express6、Netscape4.78、S/Goma2.12を組み込んだWinbiff2.34を使用した結果、電子メール送信時の電子署名と暗号化、受信時の復号と電子署名の検証が可能であった。

また、データ暗号化を目的としてデータベースソフトウェアであるPostgreSQLやディレクトリサービスソフトウェアであるOpenLDAPなどでも使用した結果、暗号化できることが確認できた。

なお、本システムで発行した秘密鍵と証明書を組み込んだシステムは、当センターで既に複数稼働している。

5. おわりに

本報告では、構築した認証局システム、秘密鍵・X.509公開鍵証明書発行手順と失効手順、ネットワーク構成について述べ、認証局システムの運用とシステムへの適用について考察した。

本システムの特徴は、UIを用いてPKCS#12ファイルをより安全に末端ユーザなどのEEへオンライン配布するため、ユーザ設定パスワードに加えてRAが発行したチャレンジパスワードの入力、パスワード不正入力回数制限、PKCS#12ファイルの取得回数制限と取得期間制限などの機能を持つことである。このことで、秘密鍵とX.509公開鍵証明書のより安全かつ簡易なオンライン配布を可能とした。

本システムで発行した秘密鍵と証明書をPKI対応のWebサーバとブラウザおよび電子メールなどのソフトウェアに組み込んだ結果、Web閲覧ではサーバおよびユーザ認証とデータ暗号化、電子メールでは送信時の電子署名と暗号化および受信時の復号と電子署名の検証、データベースシステムなどでは暗号化が可能となった。

多くの県内企業では、Webサーバや電子メールなどを利用して、社内あるいは関連会社などと情報交換、共有、提供などを行っている。Webサーバやブラウザあるいは電子メールソフトウェアでは、既にPKIに対応したものを利用している場合も多い。その場合は、企業内で認証局システムを運用し、秘密鍵とX.509公開鍵証明書をソフトウェアに組み込むことで容易に強固な認証と暗号化通信が可能となる。

また、PKIでは、CAやRAの運用、証明書などのオフライン配布経路などには人間が介在する。また、末端ユーザなどのEEは、自分の秘密鍵を保持している。これらに関与する全ての人が、自覚を持って運用あるいは管理を行わなければ、PKIは破綻することを認識する必要がある。

本システムでは、鍵更新の機能が不完全である。この機能は、今後改善する予定である。

文　献

1)トム・オースティン,PKI公開鍵基盤 電子署名法時代のセキュリティ入門,東京,日経BP企画 p454,2001

2)カーライル・アダムズ,スティーブ・ロイド,PKI公開鍵インフラストラクチャの概念、標準、展開,東京,ピアソン・エデュケーション,p292, 2000

3)尾方克,富松篤典,河北隆生,中嶋卓雄,認証局の構築と運用実験, 第15回産学官技術交流会講演論文集,p323, 2001

4)岡嶌崇,河北隆生,富松篤典,組織における認証局の開発,第16回産学官技術交流会講演論文集,p54-55,2002

5)Alan O. Freier他, The SSL Protocol Version 3.0, http://wp.netscape.com/eng/ssl3/ssl-toc.html

6)T. Dierks他,The TLS Protocol Version 1.0, RFC2246, 1999

7)B. Ramsdell, Ed., S/MIME Version 3 Certificate Handling, RFC2632, 1999

8)B. Ramsdell, Ed., S/MIME Version 3 Message Specification, RFC2633,1999

9)馬場達也, マスタリングIPsec, 東京, オライリー・ジャパン,p336,2001

10)OpenSSL, http://www.openssl.org/

11)大山実他,8章インターネットでのディレクトリ, X.500ディレクトリ入門,東京,東京電機大学出版局, p131-141, 2001

12)RSA Laboratories, PKCS#12 v1.0:Personal Information ExchangeSyntax,ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf

13)eToken製品概要, アラジンジャパン, http://www.aladdin.co.jp/etoken/index.html

14)RSA Laboratories, PKCS#11 v2.11:Cryptographic Token Interface Standard,ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v211/pkcs-11v2-11r1.pdf