PKI（Public Key Infrastructure：公開鍵暗号基盤）の活用

1　認証局ソフトウェアで証明書を発行する

認証局ソフトウェア（Easy Cert）で認証局を構築する手順を示す。この「Easy Cert」は名古屋工業大学電気情報工学科の岩田研究室で開発された暗号ライブラリをベースにして開発された認証局ソフトウェアである。証明書と失効リストの発行を主眼にしており、登録局やリポジトリの要素は省略されている。

① Easy Cert のダウンロード

岩田研究室の Web サイトからダウンロードする。

http://mars.elcom.nitech.ac.jp/security/download.htmi

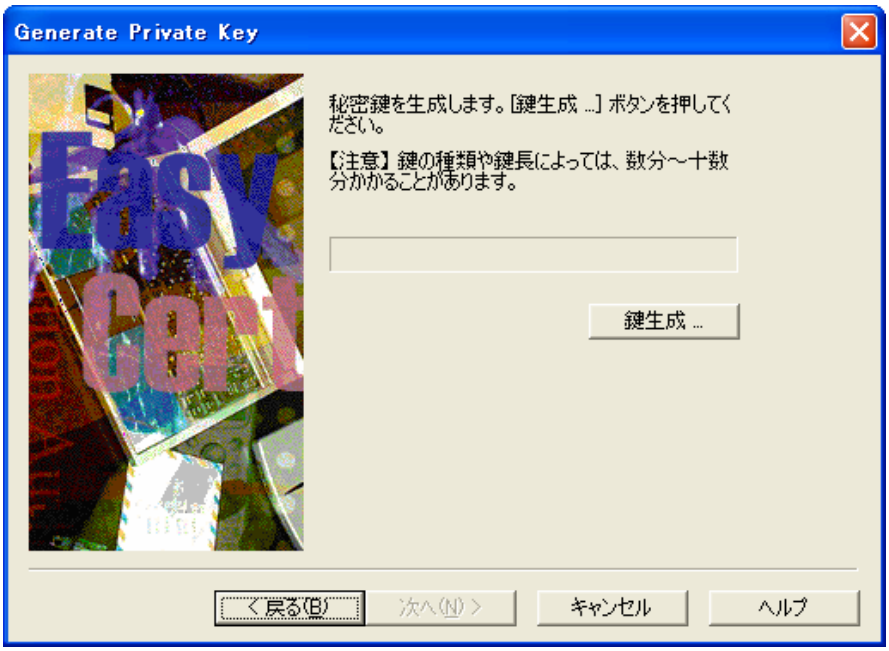
ダウンロード可能な最新バージョン 0.91 Beta2　（EasyCertSetup091b2e.lzh）

ダウンロードしたファイルは圧縮されているため、解凍する必要がある。解凍後、setup.exe.を実行するとインストールが始まる。

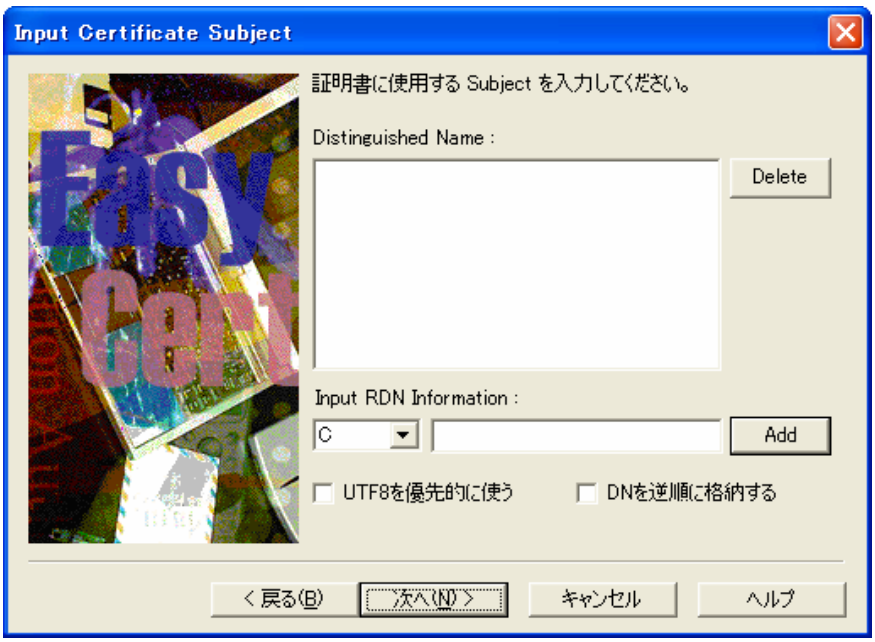
次に、構築するＣＡの公開鍵と秘密鍵の鍵ペアに関する指定を行う。ここでは、「証明書と秘密鍵を新規に作成する。」を選択する。証明書の電子署名はここで生成した秘密鍵で行うことになる。

Easy Cert は輸出規制に対応しているので RSA 暗号の鍵長が制限されている。本来であれば、安全性の観点からは 1,024 ビットを選ぶのが望ましい。ハッシュ関数については、MD5 は衝突を起こす可能性があることが報告されているため、「SHA1」を選択する。

次に鍵の生成を行う。「鍵生成」ボタンをクリックしてしばらく待つと鍵生成が完了する。



証明書のサブジェクトとなる識別名を入力する。



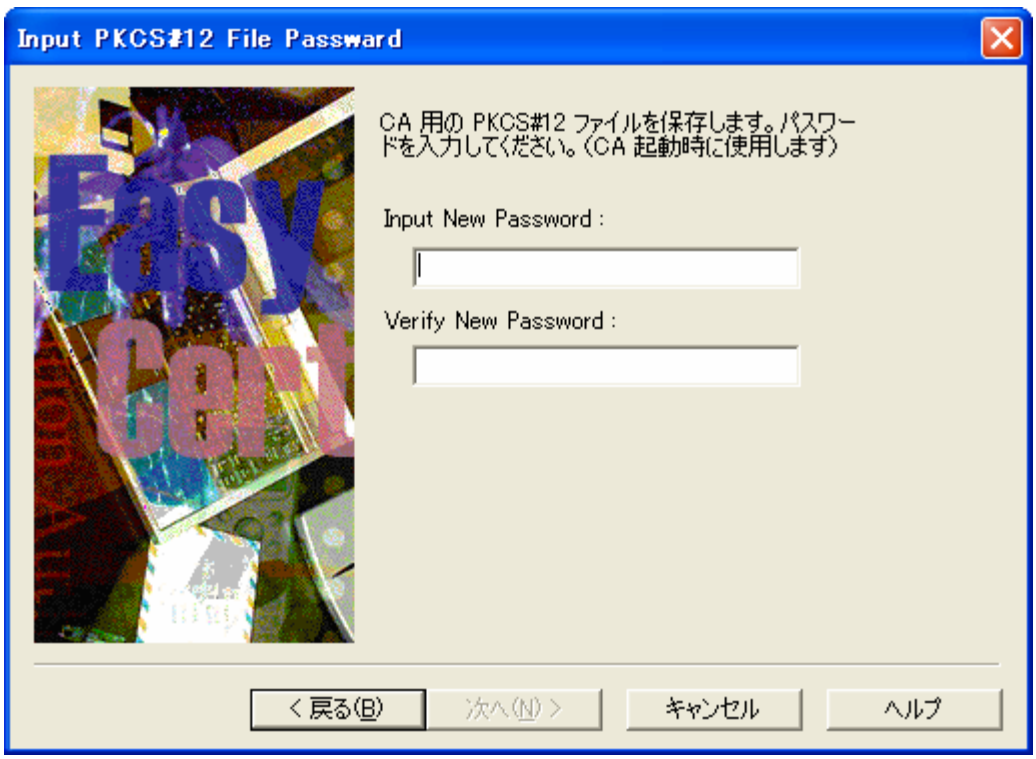
証明書の DN（Distinguished Name）フィールドの項目名と概要

項目名	説　　明
C	国（Country）の名称。必ず半角 2 文字のアルファベットを入力する。
ST	州名（State）の名称。入力する必要はない。
L	位置（Location）の名称。これも特に入力の必要はない。
O	組織（Organization）の名称。半角 64 文字まで入力可。日本語入力も可能。
OU	下位組織（Organization Unit）の名称。半角 64 文字まで入力可。日本語入力も可能。
CN	一般的な名前（Common Name）。半角 64 文字まで入力可。日本語入力も可能。
EMAIL	電子メールアドレス（EMAIL）。半角 64 文字まで入力可能。

続いて、証明所に記載される有効期間と失効リスト（CRL）を発行する間隔をそれぞれ日数で指定する。（いずれの項目もデフォルト値）

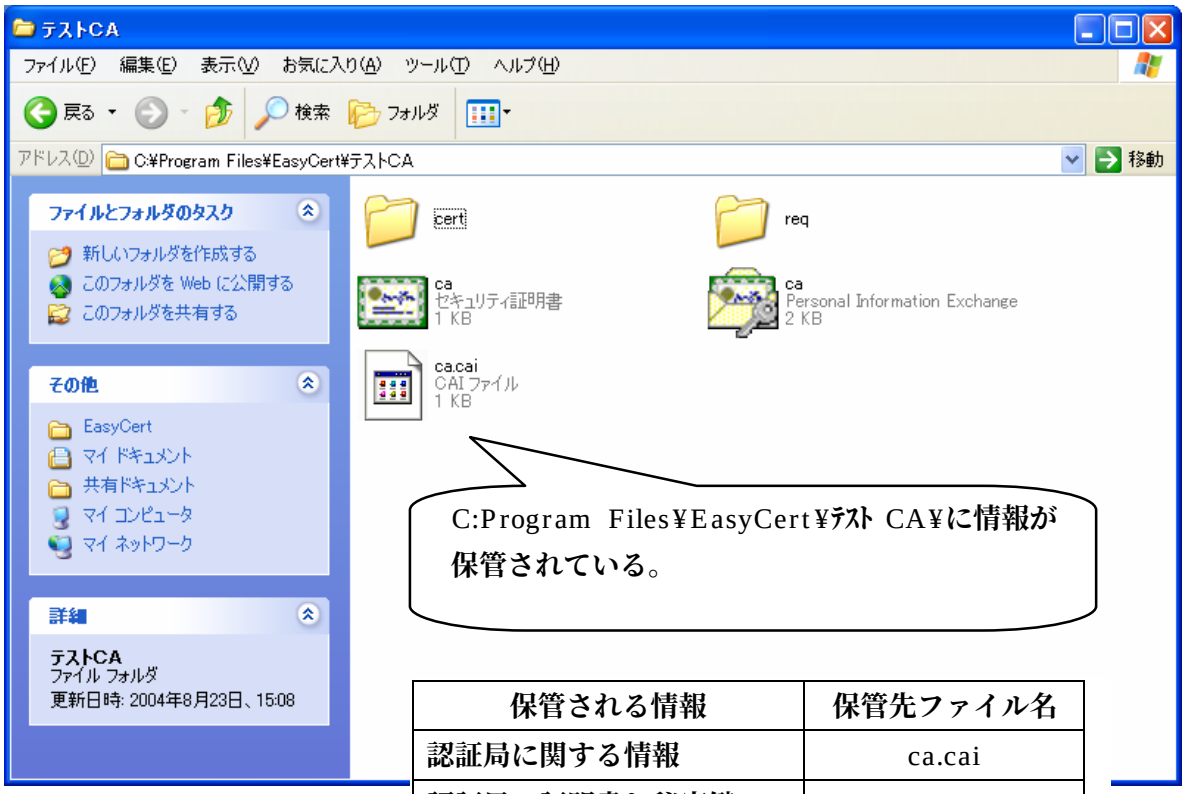
続いて、必要な証明書の拡張情報を指定する。「詳細設定」ボタンをクリックすると、拡張情報の値を詳細に指定できる。ここもデフォルトのままで、「次へ」をクリックする。

最後に、PKCS#12 ファイルのパスワードを設定する。PKCS#12 形式は作成した証明書と秘密鍵をペアでファイルに保存できる。その場合、秘密鍵などを保護する必要があるので、パスワードを使って暗号化してから保存する。このパスワードは、Easy Cert 起動時や「ファイル」メニューの「ＣＡを開く」コマンド実行時などに確認されることになる。

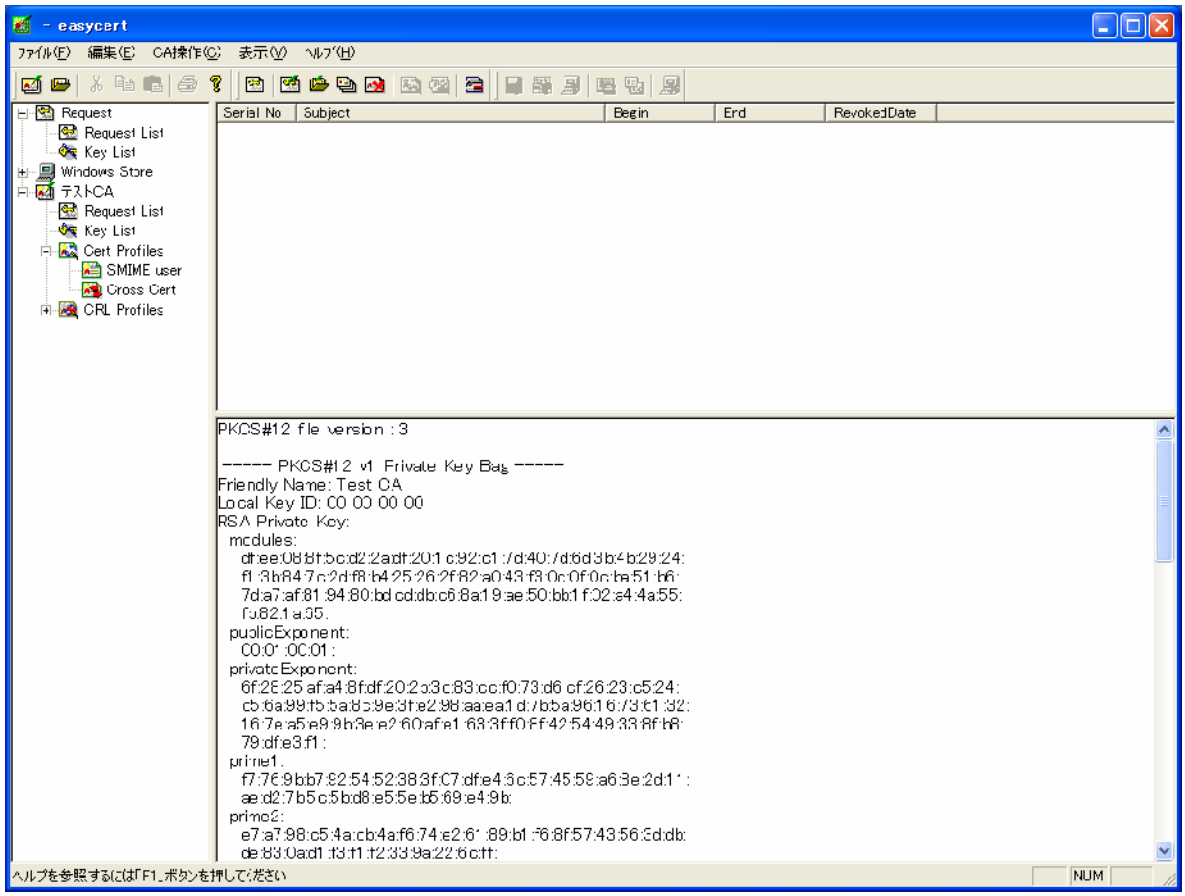


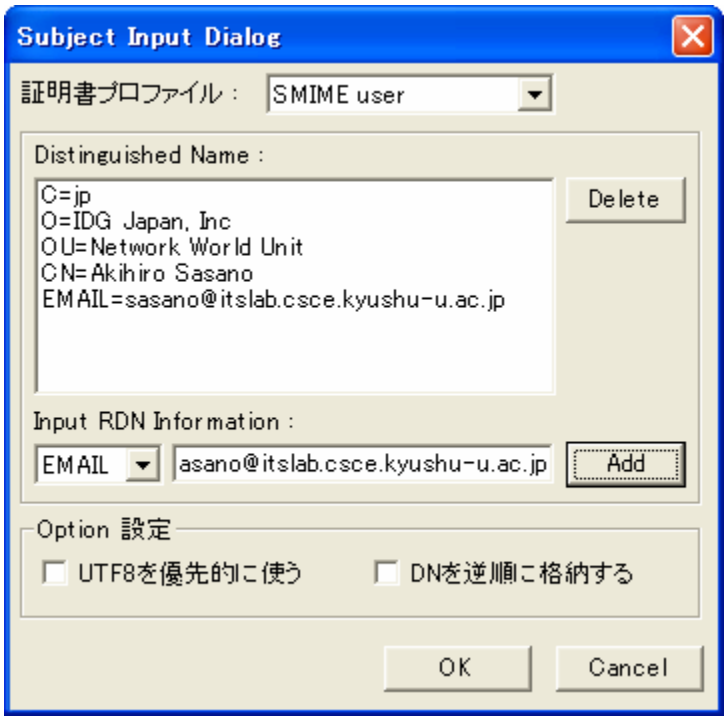
以上で認証局の構築が完了するので、「完了」ボタンをクリックして作業を終了する。

作成したファイルは、＜インストール先＞￥＜認証局の名前＞ディレクトリに保管される。

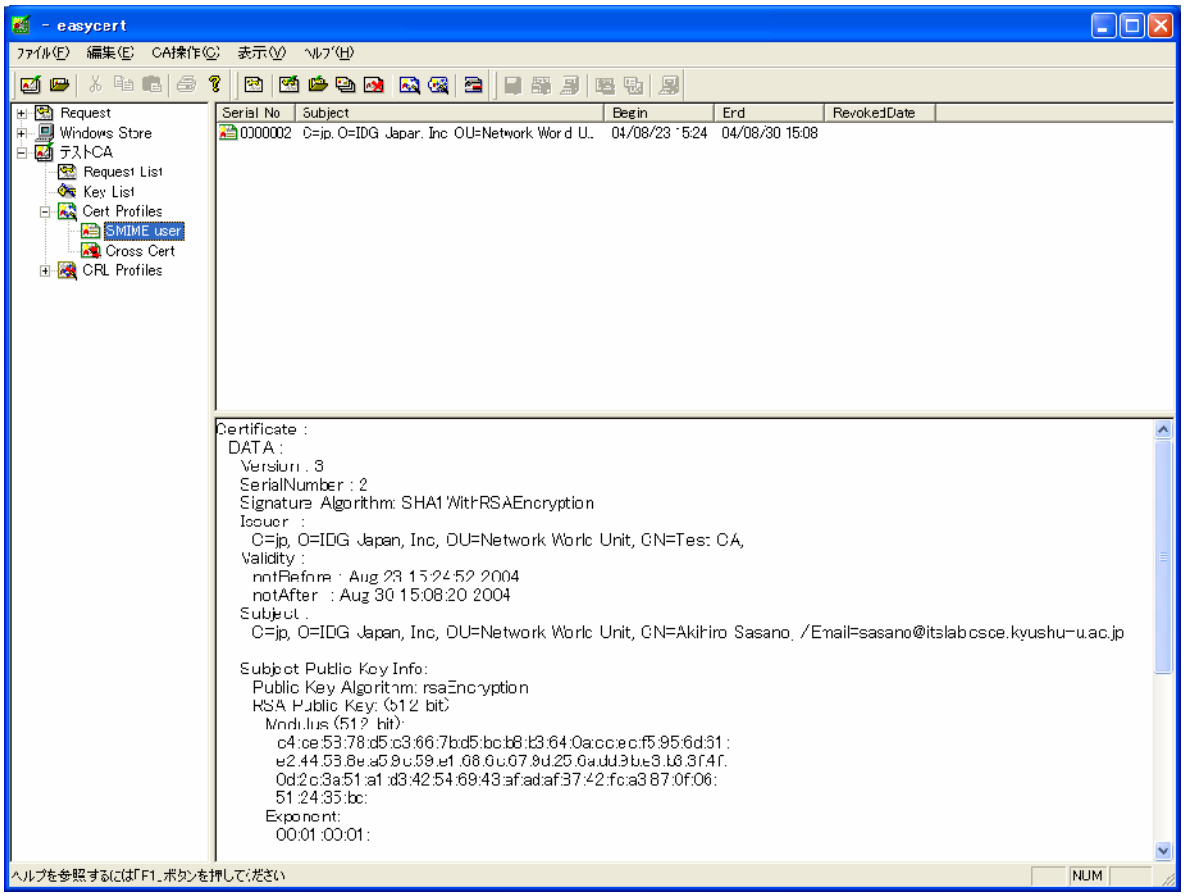


保管される情報	保管先ファイル名
認証局に関する情報	ca.cai
認証局の証明書と秘密鍵のペアの PKCS#12	ca.p12
認証局の証明書	ca.cer



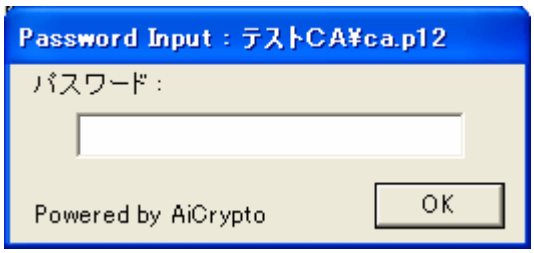

Subject Input Dialog
証明書プロファイル：
SMIME user
Distinguished Name：
C=jp
O=IDG Japan, Inc
OU=Network World Unit
CN=Akihiro Sasano
EMAIL=sasano@itslab.csce.kyushu-u.ac.jp
Delete
Input RDN Information：
EMAIL
asano@itslab.csce.kyushu-u.ac.jp
Add
Option 設定
UTF8を優先的に使う
DNを逆順に格納する
OK
Cancel


Password Input : Save Private Key
新規パスワード：
******
新規パスワード(確認用)：
******
Powered by AiCrypto
OK

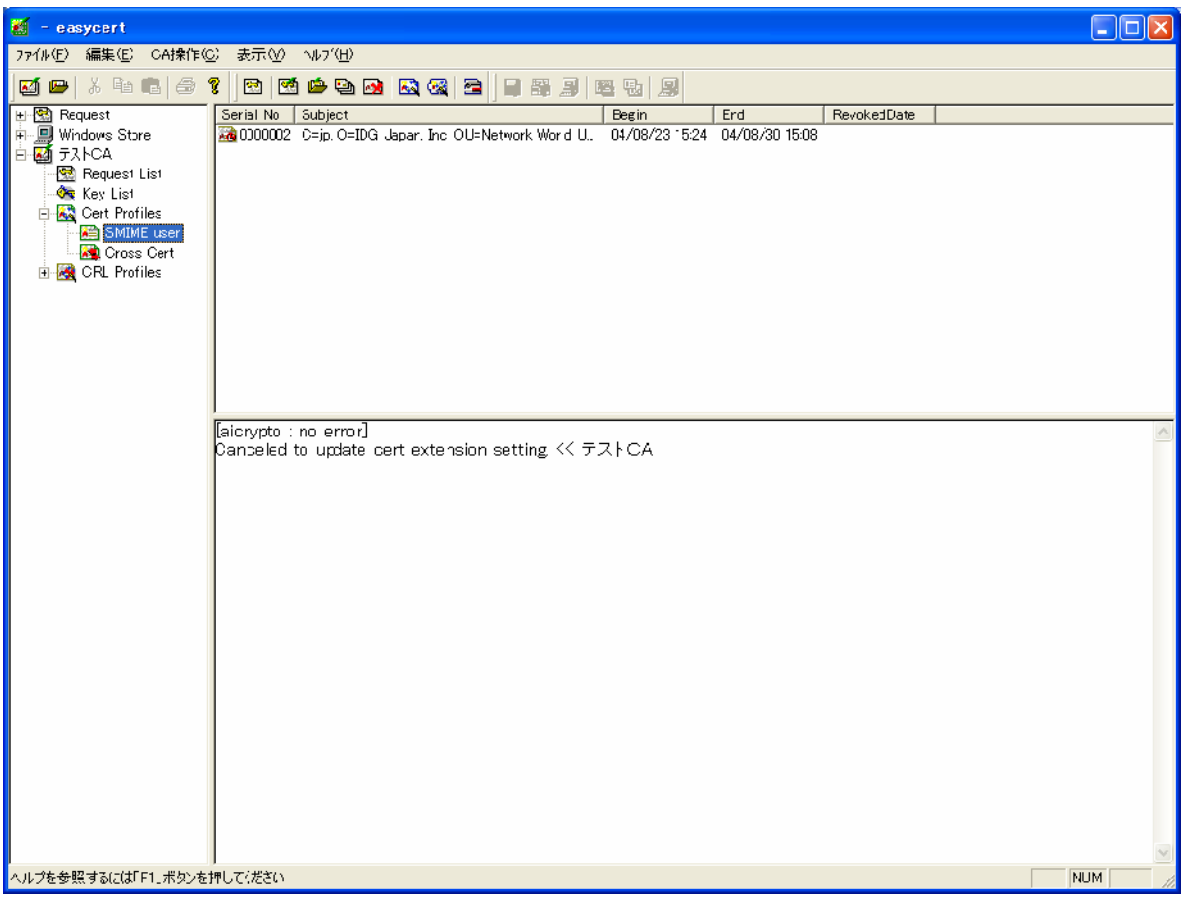

easycert
ファイル(F) 編集(E) CA操作(C) 表示(V) ヘルプ(H)
Request
Windows Store
テストCA
Request List
Key List
Cert Profiles
SMIME user
Cross Cert
CRL Profiles
Serial No
Subject
Begin
End
RevokedDate
0000002 C=jp, O=IDG Japan, Inc, OU=Network Word U.. 04/08/23 15:24 04/08/30 15:08
Certificate :
DATA :
Version : 3
SerialNumber : 2
Signature Algorithm: SHA1WithRSAEncryption
Issuer :
C=jp, O=IDG Japan, Inc, OU=Network World Unit, CN=Test CA,
Validity :
notBefore : Aug 23 15:24:52 2004
notAfter : Aug 30 15:08:20 2004
Subject :
C=jp, O=IDG Japan, Inc, OU=Network World Unit, CN=Akihiro Sasano, /Email=sasano@itslab.csce.kyushu-u.ac.jp
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (512 bit)
Modulus (512 bit):
c4:ce:53:78:d5:c3:66:7b:d5:bc:b8:b3:64:0a:cc:ec:f5:95:6d:31:
e2:44:53:8e:a5:9c:59:e1:08:6c:07:9d:25:6a:dd:3b:e3:b8:3f:4f:
0d:2c:3a:51:a1:d3:42:54:69:43:af:ad:af:37:42:fc:a3:87:0f:06:
51:24:35:bc:
Exponent:
00:01:00:01:
ヘルプを参照するには「F1」ボタンを押してください
NUM

2　認証局を開いて証明書プロファイルを設定する

認証局に関する操作は、起動画面の「ファイル」の「CA を開く」から行う。「CA を開く」では、認証局に関する情報を得るために「ca.cai」を選択すると、その認証局の PKCS#12 ファイルのパスワードを聞かれる。パスワード欄を空白のまま「OK」ボタンをクリックするとキャンセルできる。認証局の表示が不要な場合には、「CA を閉じる」を選択する。



発行する証明書のプロファイルを設定する。ツリービューで「SMIME user」のコンテキストメニューの「プロファイルの設定」から証明書の基本領域を設定する。

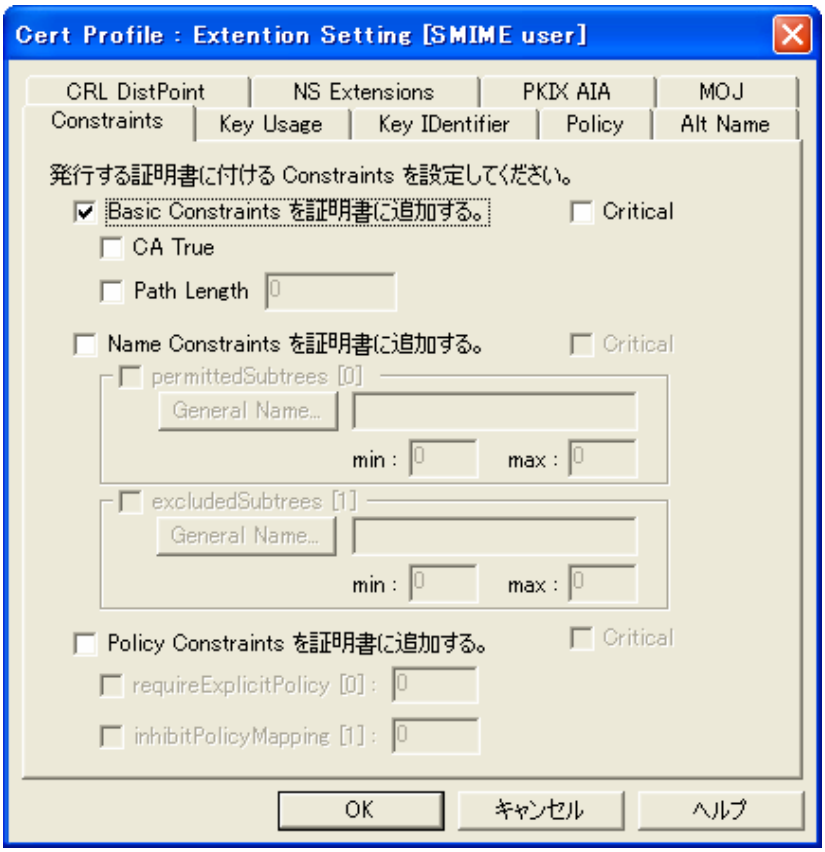


「拡張情報設定」では、証明書拡張領域の設定を行う。ここで設定した内容が、これ以降に発行する証明書に反映される。基本領域として、証明書の有効期限や証明書のバージョンなどの項目を設定する。デフォルトでは有効期間が 7 日となっている。

①　基本制限（Basic Constrains）

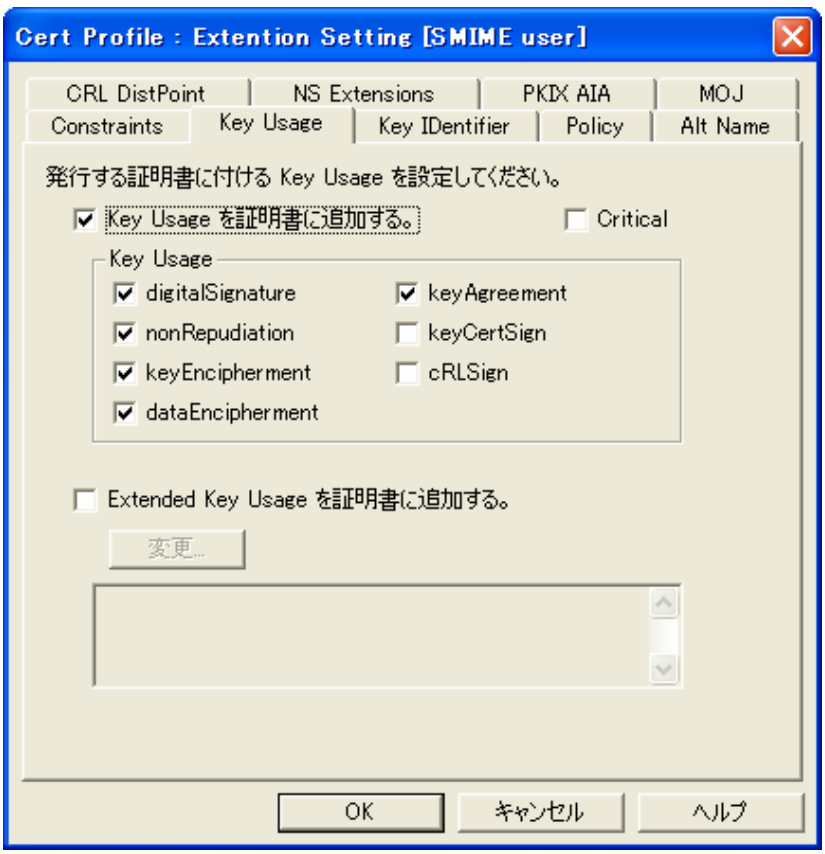
「Constraints」タブで指定する。認証局か否かを示す。「Basic Constrains」はどのような場合にもほ

ほ必須と考えてよい。メニューの「Basic Constrains を証明書に追加する。」をチェックする。



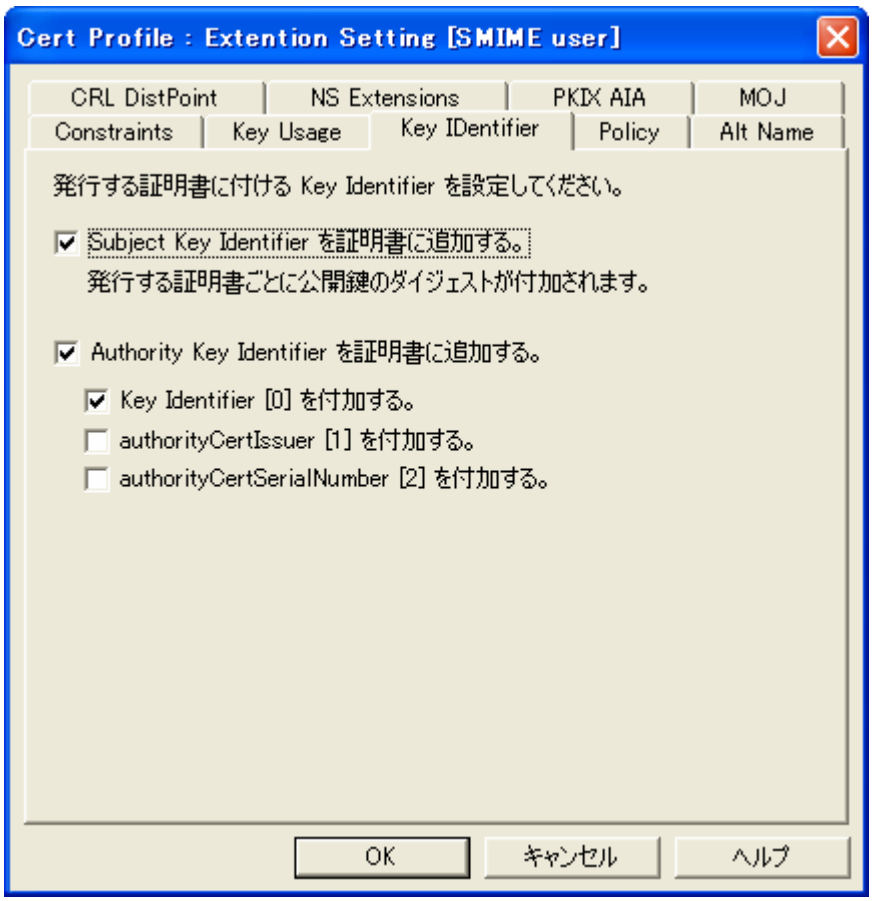
② 鍵使用法（Key Usage）

「Key Usage」タブで、許可する鍵の用途を指定する。RSA 暗号の鍵を電子メールに使う場合には、「digitalSignature」（電子署名）と「nonRequdiation」（否認防止）、「keyEncipherment」（鍵の暗号化）をチェックしておく。



③　サブジェクトキーID（Subject Key Identifier）と発行元キーID（Authority Key Identifier）

サブジェクトキーID は「Key IDentifier」画面で指定する。サブジェクト ID キーとは、サブジェクトの公開鍵の識別子、つまり証明書に含まれている公開鍵を識別するための値で、発行元キーID は証明書を発行した認証局の公開鍵の識別子である。公開鍵の識別子には、公開鍵のハッシュ値などを使う。発行元キーID では発行元 DN とシリアル番号の組み合わせを使うこともある。



④　項目の重要性（Critical）

証明書の拡張領域のそれぞれの項目には、「重要」か「非重要」かを指定する。「重要」となっている項目については、アプリケーションはその項目を仕様に基づいて解釈し、処理しなければならない。アプリケーションが解釈および処理できない項目が「重要」であった場合、アプリケーションはその証明書を無効なものとして扱う。

⑤　サブジェクトの固定値

「CA 操作」メニューの「CA のプロパティ」で、証明書のサブジェクトの一部に標準で使用する値を決める。デフォルトでは認証局証明書のサブジェクトから、C、O、OUの値が採用されている。

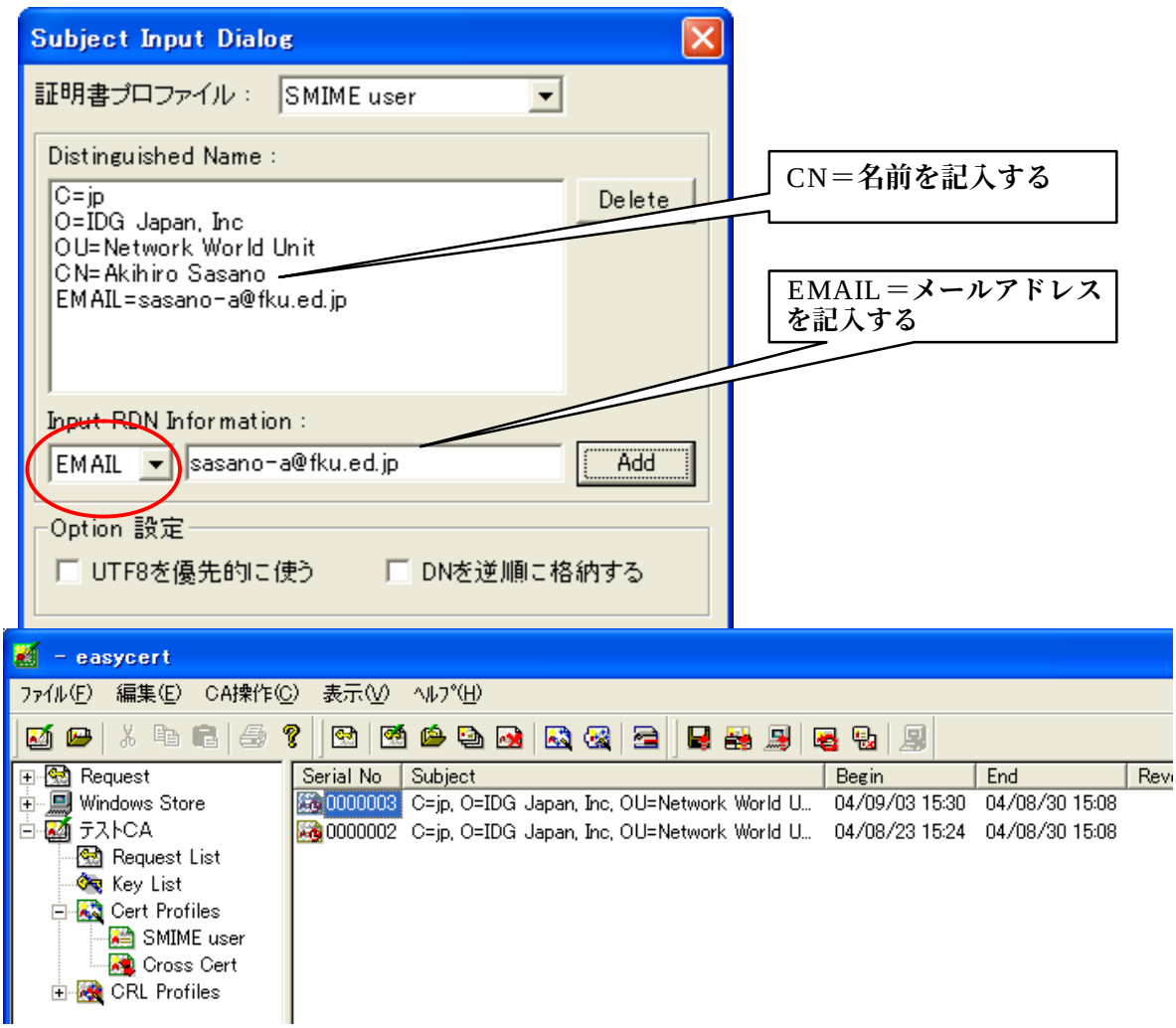
⑥　プロファイルの追加

「CA 操作」メニューの「プロファイルの追加」を選択すると、新たに証明書プロファイルを定義したり追加したりできる。

3　証明書を発行して利用者に配布する

利用者のそれぞれが使う秘密鍵と公開鍵のペアを生成し、証明書を作成する。作成した証明書はファイルに保管して利用者ごとに証明書を発行して渡す。

サブジェクトを入力する。標準の DN があらかじめ設定されているので、「CN」に名前を、「EMAIL」にメールアドレスを追加する。（CA 操作→証明書発行）



作成した証明書は、利用者に配布し、目的のアプリケーションで行うことになる。そのために、証明書を扱うための標準的なフォーマットのファイルに保存して受け渡しをする。Easy Cert では、証明書は拡張子が.cer であるようなファイルに、秘密鍵と証明書をペアでファイルにする場合は PKCS#12 と呼ばれるフォーマットに従ったファイルにそれぞれ保管可能である。

Easy Cert では、PKCS#12 に保存する場合、秘密鍵と証明書とその発行元の証明書をいっしょに保存している。これにより、目的のアプリケーションで証明書の信頼性を確認できる。

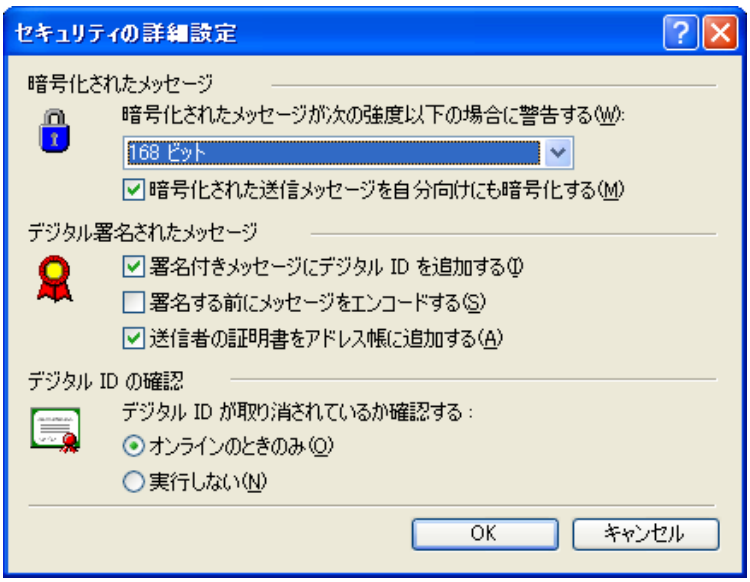
ツリービューで「SMIME user」を選択すると、右上のペインに作成した証明書が一覧表示される。証明書を選択し、コンテキストメニューを開くと「保存」と「PKCS#12 で保存」コマンドがある。「保存」では、証明書をファイルに保存する。送信相手の証明書は「保存」でファイルを作成し、自分の証明書は「PKCS#12 で保存」を使って秘密鍵と証明書がペアになったファイルを作成する。作成には秘密鍵を保護しているパスワードと PKCS#12 を保護するパスワードが必要になる。

4　Microsoft Outlook Express で証明書を使う

証明書と秘密鍵をセットで保存したファイル（拡張子が「.p12」となっている PKCS#12 ファイル）と、証明書のみを保存したファイル（拡張子が「.cer」となっている証明書ファイル）を上で作成したこの２つのファイルを電子メールソフトウェアで使うことで、電子署名や暗号化が可能となる。

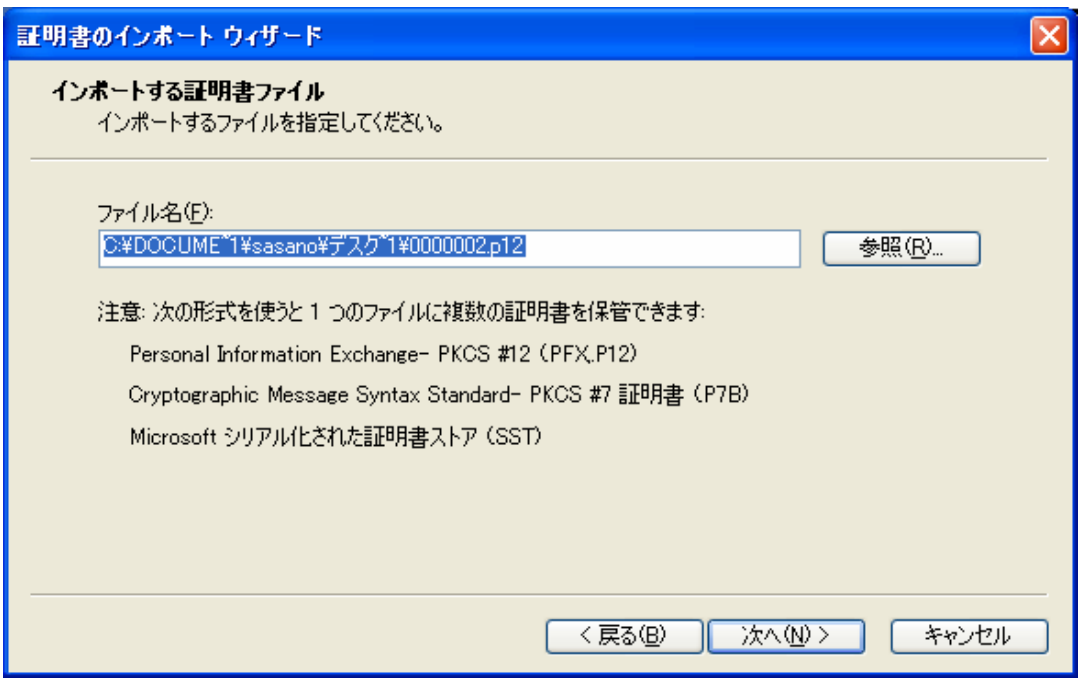
Outlook Express では、電子署名や暗号化に関る設定は「ツール」メニューの「オプション」で「セキュリティ」のタブを開いて行う。詳細な設定は、「セキュリティ」タブの「詳細設定」から行う。

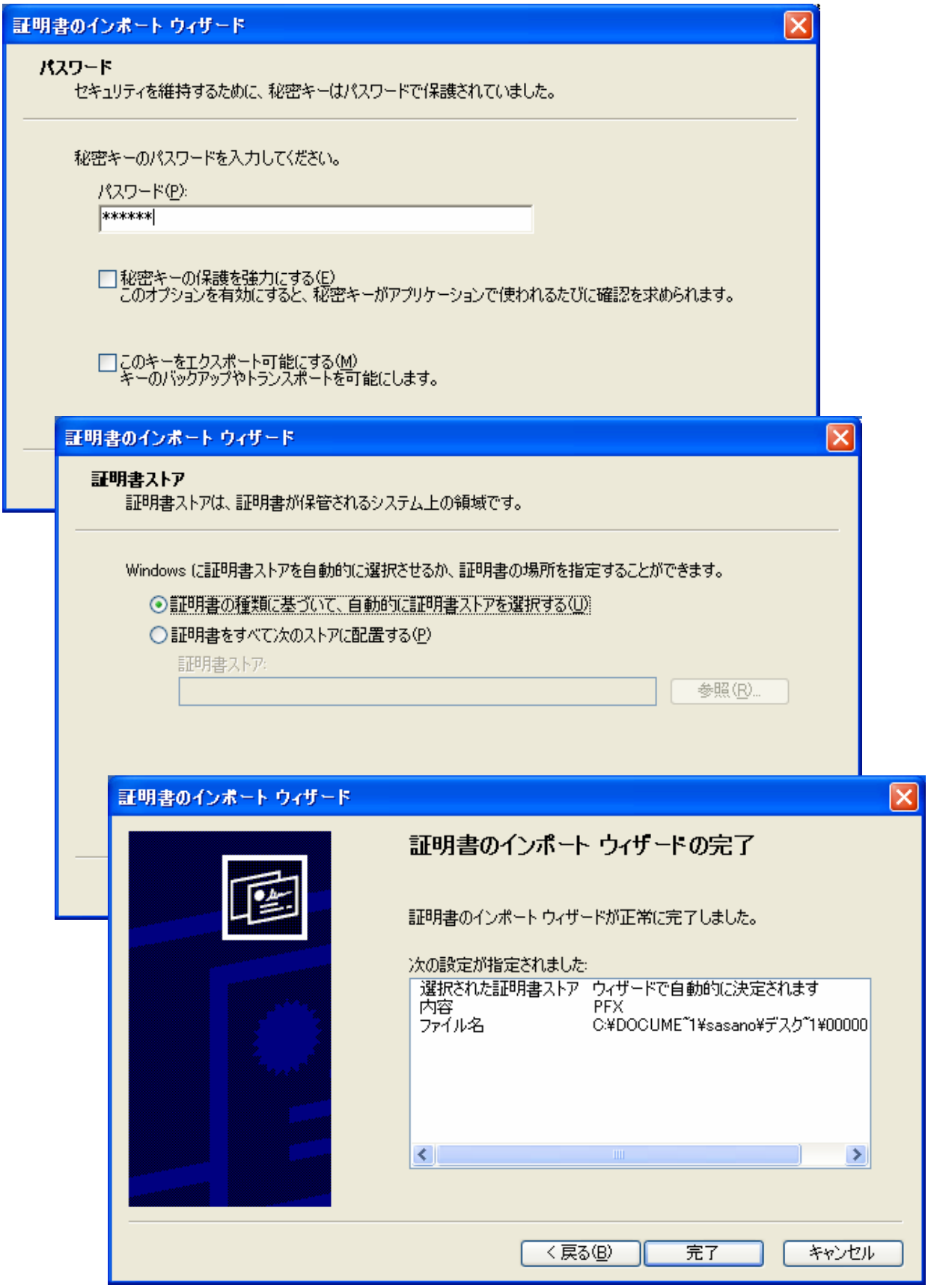
「セキュリティの詳細設定」画面では、標準で電子署名や暗号化を行うかどうかを指定する。ここでは、暗号の強度や電子署名に関るプロトコルのオプション、証明書の有効性の確認などを設定する。



①　ウィザードに従って PKCS#12 ファイルをインポート

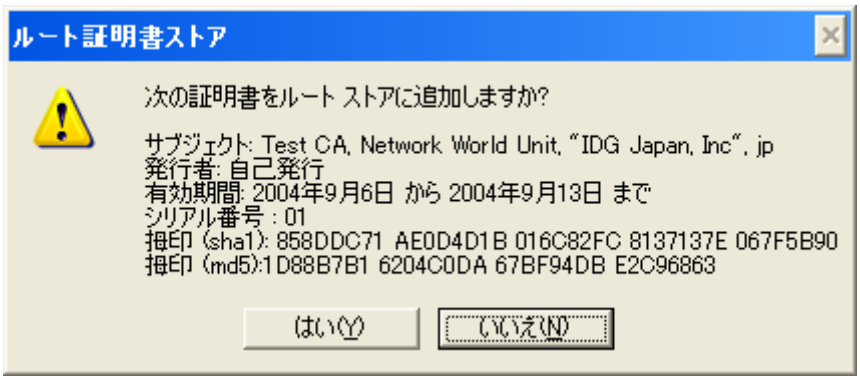
PKCS#12 ファイルをダブルクリックすると、「証明書のインポートウィザード」が開始される。このウィザードに従って操作を進めると、Windows が標準で持つ証明書の保管場所である「証明書ストア」に証明書が保管され、Outlook Express でも秘密鍵や証明書が利用可能になる。





途中で「秘密キーのパスワード」の入力が要求されるので、PKCS#12 ファイルを保護する際に用いたパスワードを入力する。「証明書のインポートウィザードの完了」画面で「完了」をクリックすると、認証局の証明書をルート証明書ストアに追加することを確認するメッセージが表示される。「はい」をクリ

ックするとインポートは完了する。



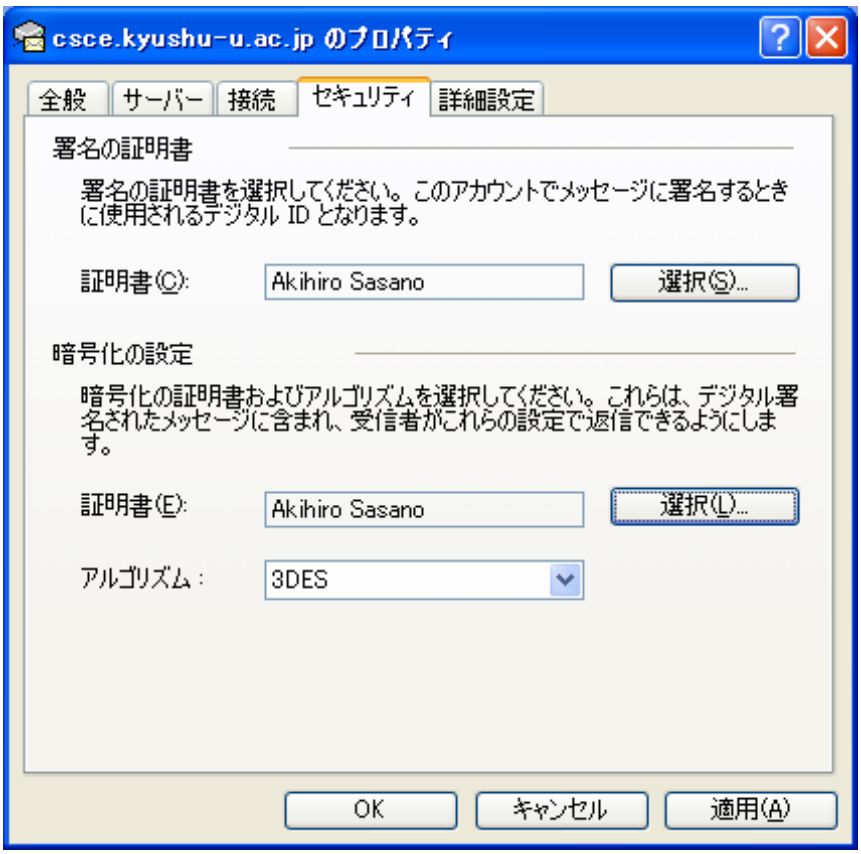
このメッセージは、PKI において、ルート証明書を信頼の基点として信頼性を検証するため、自己署名の証明書をインポートする際に、信頼できるものであるか否かを確認するために表示されるものである。利用者は、表示された「拇印（sha1)」または「拇印（md5)」を基に、自分の信頼性を確認しなければならない。拇印とは、証明書のバイナリデータを入力値としたハッシュ値のことである。自己署名の証明書の場合には、拇印を参照して、証明書が信頼できるものかどうかを確認する。ハッシュ関数には sha1 を使うケースと md5 を使うケースがあるので注意が必要である。Windows では、証明書ファイルのコンテキストメニューの「開く」コマンドで表示される「証明書」ダイアログの「詳細」ページで確認できる。

また、拇印は何らかの安全な経路を介して認証局の管理者から利用者に伝える必要がある。保護された Web サイトなどに記載するなどの方法が採られることもあるが、それ自身が同じルート証明書を基点にしている場合にはこの方法も意味がないため注意する。

② メールアカウントの設定を「プロパティ」画面で行う

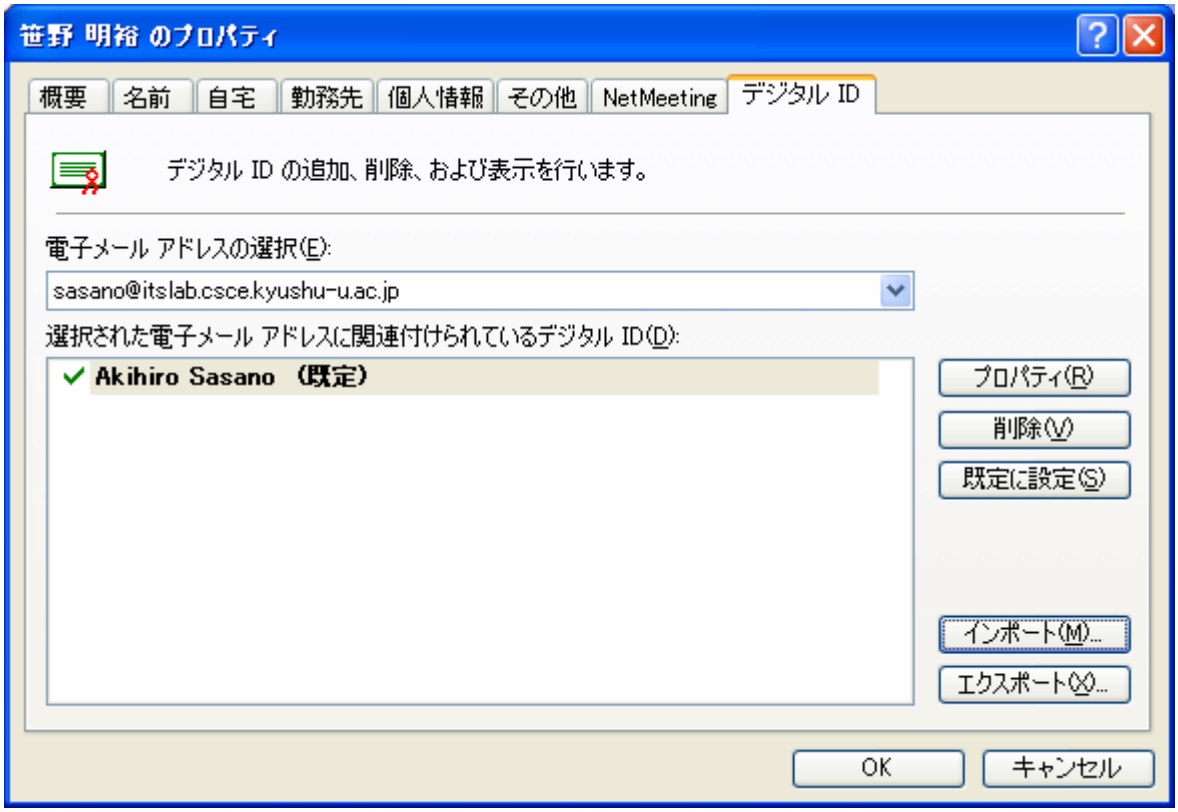
電子署名に使う証明書の指定と、暗号化に使う証明書の指定は、メールアカウントの「プロパティ」の「セキュリティ」タブで行う。

「選択」ボタンをクリックすると利用可能な証明書が表示されるので、この中から選択する。このとき、メールアカウントの電子メールアドレスと証明書の電子メールアドレスが一致している必要がある。また、「暗号化に使う証明書」が、自分宛の暗号化に使われる証明書となる。



③　「アドレス帳」から証明書をインポート

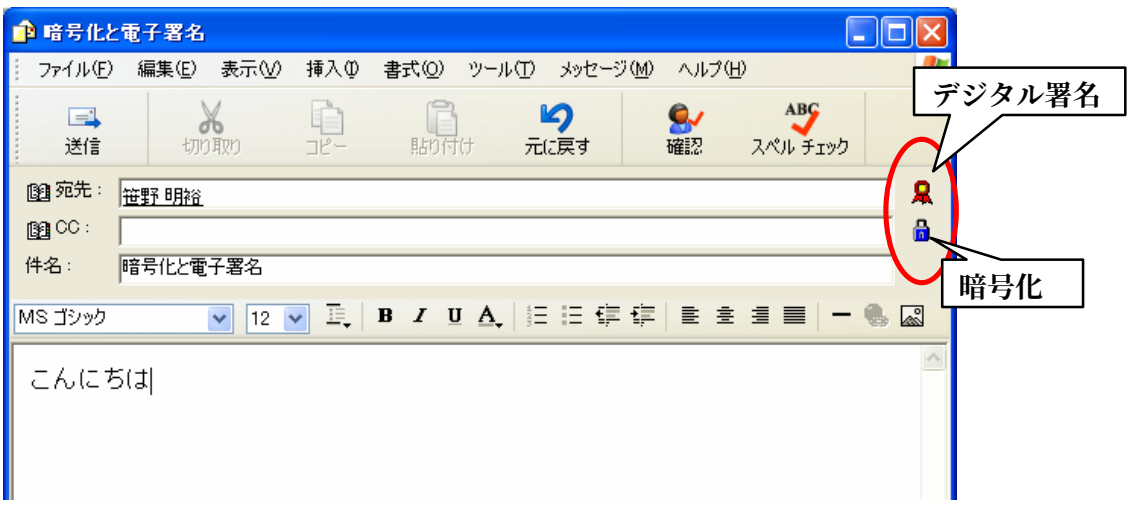
暗号化する相手は連絡先としてアドレス帳に登録されている必要がある。「ツール」メニューの「アドレス帳」から証明書をインポートして、メールの送受信に証明書を使えるように設定する。「アドレス帳」を選択し、右ペイントで通信相手を右クリックして「プロパティ」を開き、「デジタル ID」タブの「インポート」ボタンをクリックして、作成した証明書ファイルを選択する。



ここでいう「デジタル ID」とは証明書のことである。ここでも、証明書に指定されている電子メールアドレスと選択する電子メールアドレスが一致している必要がある。

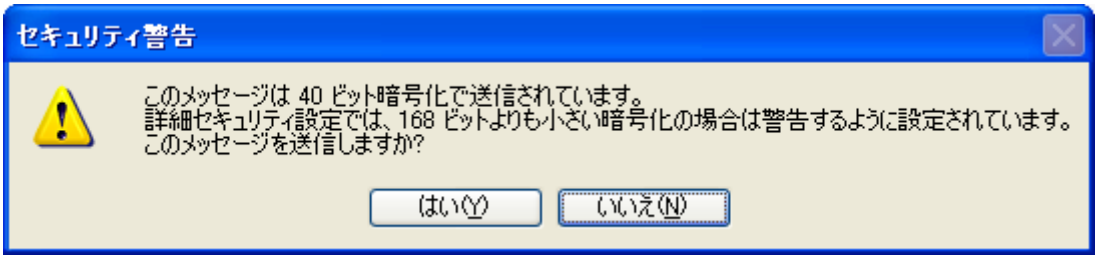
④　メッセージの暗号化

インポートした証明書を使って電子メールメッセージを暗号化して送信する。メールの作成画面で連絡先を選択し、「ツール」メニューで「暗号化」と「デジタル署名」を選択する。



「デジタル署名」にはリボンのアイコンが、「暗号化」には錠前のアイコンが対応する。なお、Outlook Express では電子署名を「デジタル署名」と呼んでいる。一般的には、電子的に行われる署名一般を「電子署名」と呼び、「電子署名」のうち、PKI の枠組みにおいてなされるものは「デジタル署名」と呼んで使い分けられる傾向がある。

「送信」をクリックすると、下図のような「セキュリティ警告」が表示される。暗号化アルゴリズムに 40 ビットの RC2 が利用されることになる。



この警告メッセージが表示される理由については、Outlook Express が暗号化アルゴリズムを決定するにあたって、相手からの電子署名のデータから、利用可能な暗号化アルゴリズムの通知を参照している。しかし、電子署名されたメールを一度も受信していない時点では、40 ビットの RC2 を利用することになる。これと「ツール」メニューの「オプション」コマンドの「セキュリティ」タブの「セキュリティの詳細設定」で指定されている「暗号化されたメッセージが次の強度以下の場合に警告する」で設定されているビット長とを比較して、40 ビットの RC2 では指定に満たないため安全性が不十分と判断される。

5　PKI の構成要素

PKI を大別すると次の４つの要素で構成される。

① 利用者

PKI の枠組みの中では「エンドエンティティ」と呼ばれる。エンドエンティティは、公開鍵と対応づけたられた、公開鍵の持ち主である人間であったり、サービスを提供するサーバのサービスそのものであったりする。

② 登録機関

証明書を発行するにあたって、利用者の本人性を確認する役割を担う。

③ 証明書発行機関

この機関自身が証明書を持ち、公開鍵とその持ち主の情報に電子署名を行って証明書を発行する。また、失効した証明書のシリアル番号のリストに電子署名を行い、証明書失効リストを発行する。

④ リポジトリ

証明書や証明書失効リストを格納し、それらを公開、配布することを目的として設置される。通常 LDAP ディレクトリサーバが用いられる。

認証局は、登録機関、証明書発行機関、リポジトリの役割を一括してサービスを提供したり、登録局やリポジトリを外部に持ち、証明書発行機関のみサービスを提供したりといった形態を取る。この３つの要素の管理と運営が認証局の業務である。認証局は CPS（Certification Practice Statement）という形で運営方針などを規定する文章を公開し、利用者に明示する義務がある。

6　3種類の暗号化アルゴリズム

PKI（公開鍵基盤）は、暗号化アルゴリズムとして公開鍵暗号のみを用いて成立しているのではなく、次に示す3種類のアルゴリズムを組み合わせて使っている。

① ハッシュ関数（一方向性関数）

入力データを基にして一定の長さのデータを作る処理を行う。出力データからもとの入力データを算出することはできない。

② 共通鍵暗号（秘密鍵暗号）

暗号化と復号化に同じ鍵を使用する。この鍵を「共通鍵」または「秘密鍵」と呼ぶ。

③ 公開鍵暗号

暗号化に使う鍵と複号化に使う鍵が異なり、一方の鍵を公開してももう一方の鍵を割り出すことはできない。公開する鍵を「公開鍵」、秘密にする鍵を「秘密鍵」または「私有鍵」「非公開鍵」と呼ぶ。上記の共通鍵暗号における共通鍵も「秘密鍵」と呼ぶことが多い。

公開鍵証明書などに使われている電子署名では、ハッシュ関数と公開鍵暗号を組み合わせて使う。署名したいデータのハッシュ値を、秘密鍵を使った公開鍵暗号で暗号化して相手に渡す。

電子メールでメッセージを暗号化する場合などは、共通鍵暗号と公開鍵暗号を組み合わせて使う。共通鍵暗号の処理の高速性を生かして、メッセージを公開鍵暗号ではなく共通鍵暗号で暗号化し、その鍵を公開鍵暗号で暗号化し、暗号化したメッセージとともに通信相手に送るという手法が取られている。