サーバ証明書発行・導入における啓発・評価プロジェクト
オープンドメイン認証局

サーバ証明書インストールマニュアル

Ver1.05
2008 年 8 月 26 日

国立情報学研究所

改訂履歴		
版数	日付	内容
V1.0	2007/05/22	初版（配布開始）
V1.01	2007/10/31	鍵生成時の openssl コマンド引数を OS 別に記述
V1.02	2007/11/30	鍵生成時の 512bit 鍵長に関する注意事項を記述
V1.03	2008/2/21	CSR プロファイルの追加、modssl 中間 CA 証明書インストール方法修正
V1.04	2008/4/24	「1.3 証明書の種類」のテーブルフォーマットを変更
V1.05	2008/8/26	IIS6.0 証明書更新方法の記述

1. はじめに

1.1 サポートする SSL/TLS サーバ

サポートする SSL/TLS サーバは以下となっています。

表 1-1 サポートする SSL/TLS サーバと各サーバの記載章節

SSL/TLS サーバ	記載章節	章節題
各サーバ共通	1	はじめに
	2.1	CSR とは
	2.2	DN のルール
	2.4	鍵ペア生成及び CSR 作成前の準備
Apache-SSL	2.5	Openssl での鍵ペア生成と CSR 作成
	3.2	Apache-SSL
Apache+mod_ssl	2.5	Openssl での鍵ペア生成と CSR 作成
	3.3	Apache (mod_ssl)
Tomcat (4、5.0)、JRE1.4.2	2.6	Java keytool での鍵ペアと CSR 生成
	3.4	Jakarta Tomcat
IIS (Internet Information Server) 5	2.5	Openssl での鍵ペア生成と CSR 作成
	3.1	事前準備
	3.5	IIS5
IIS6	2.5	Openssl での鍵ペア生成と CSR 作成
	3.1	事前準備
	3.6	IIS6

動作確認済みのブラウザについては，プロジェクトウェブサイトをご覧下さい。
https://upki-portal.nii.ac.jp/cerpj/

1.2 Openssl の利用

証明書を申請する際に必要となる鍵の作成や CSR の生成には Tomcat を除く SSL/TLS サーバで openssl を利用します。

OpenSSL Project (http://www.openssl.org) では UNIX で動作するモジュールだけでなく、Windows 版のバイナリモジュールも提供されています。Windows 版の OpenSSL は、OpenSSL binaries (http://www.openssl.org/related/binaries.html) ページで、EXE 形式のインストーラファイルをダウンロードしてインストールすることが可能です。また、Cygwin (http://www.cygwin.com) をインストールすることで UNIX のような操作で openssl を利用することができます。

ほとんどの UNIX システムでは、コンパイル可能なソースコード形態でも提供されています。

Openssl のインストール方法等は OpenSSL Project (http://www.openssl.org) 等のインターネット上のサイトやダウンロードしたファイルに付属しているインストールマニュアルを参照してください。

1.3 証明書の種類

NII が運用するオープンドメイン認証局を利用するために必要となる証明書には以下の種類の証明書が必要となります。次の証明書は、インストールしていただく必要があります。

表 1-2証明書の種類

名称	役割		ファイル名
SecurityCommunicationRootCA1 証明書（中間 CA 証明書）	WebTrust for CA 規準の認定を取得し、主要な Web ブラウザに「信頼できるルート認証機関」として登録されているルート認証局の SECOM Trust.net Root1 CA から発行された中間 CA 証明書です。SECOM Trust.net Root1 CA 自身も WebTrust 認定を取得しています。		scroot1.crt
	https://repo1.secomtrust.net/sppca/NII/ODCA/index.html		
NII オープンドメイン認証局証明書（中間 CA 証明書）	SecurityCommunicationRootCA1 認証局から発行された中間 CA 証明書です。NII オープンドメイン認証局の証明書となり、この認証局からサイト証明書（SSL/TLS 証明書）が発行されます。この証明書は SSL/TLS サーバに中間 CA 証明書として登録する必要があります。		niica.crt
	https://repo1.secomtrust.net/sppca/NII/ODCA/index.html		
SSL/TLS（サイト）証明書	SSL/TLS サーバに対して NII オープンドメイン認証局が発行した証明書です。証明書が有効である場合には、そのサーバが間違いなくサーバ証明書サブジェクト(発行対象)によって運用管理されているものであることが、NII オープンドメイン認証局によって証明されています。	SSL/TLS（サイト）証明書については、所属組織のプロジェクトご担当者にお問い合わせください。	Example.crt

注：上記のファイル名は本文書内の各手順で証明書を表す際に使うファイル名を示しています。

2. 鍵ペアと CSR の生成

2.1 CSR とは

CSR（証明書発行要求：Certificate Signing Request）は証明書を作成するための元となる情報で、その内容には、加入者が管理する SSL/TLS サーバの組織名、Common Name（サーバの FQDN)、公開鍵などの情報が含まれています。NII では、加入者に作成いただいた CSR の内容を元に、証明書を作成します。

2.2 DN のルール

DN（Distinguished Name）とは SSL/TLS サーバの名称を表す識別名です。加入者が管理する SSL/TLS サーバの組織名、Common Name（サーバの FQDN）を一意に示す情報です。
CSR を作る際の DN 情報入力時の注意点を以下に示します。

- SSL/TLS サーバの識別名（DN）に、別紙１に記載の文字は使用できませんので、ご注意ください。
  ※&が含まれる場合は、 半角英語の　and　等に置き換えてください。
- 各項目において、スペースのみの入力は控えてください。
  スペースのみの入力項目がある場合、証明書が発行されません。
- Common Name（コモンネーム）からドメイン名を省略したり、接続する URL に表示されるサーバ名とコモンネームが一致しない場合、ブラウザがサイトへの安全な接続を拒否する場合があります。
- Common Name（コモンネーム）に、プロトコル特定子(http://)、ポート番号、パス名は使用しないでください。また、「*」や「?」のワイルドカード文字や、IP アドレスは使用しないでください。

表 2-1 DN の要素と注意事項

項目	指定内容の説明と注意点	必須	文字数
Country Name	本認証局では必ず「**JP**」と入力してください。	○	JP 固定
State or Province Name	**本認証局では使用しないでください。「.」（ドット）を入力することで省略できます。**	×	省略
Locality Name	本認証局では必ず「**Academe**」と入力してください。	○	Academe 固定
Organization Name	大学名または研究所名等を示します。この情報は、登録担当者の方にお問い合わせください。この項目を省略することは、できません。	○	半角の英数字、別紙１記載文字以外で 64 文字以内
Organizational Unit Name	証明書を使用する部局等の名前になります（この項目は省略可能です)。	△	半角の英数字、別紙１記載文字以外で 64 文字以内
Common Name	サーバの URL に表示されるウェブ・サーバの名前を FQDN で入力してください。例えば、SSL/TLS を行うサイトの URL が https://www.nii.ac.jp/ の場合にはコモンネームの値は「www.nii.ac.jp」となります。	○	FQDN の規格に則った文字で 64 文字以内

2.3 CSR プロファイル

本認証局で作成する CSR プロファイルは以下のとおりです。CSR を作成する前にご確認ください。

基本領域		設定内容	補
Version		Version 1(0)	-
Subject	Country	C=JP（固定値）	1
	Locality	L=Academe（固定値）	1
	Organization	O=″主体者組織名″ ＊ 機関毎に任意に指定 例 ） o= National Institute of Informatics	1
	Organizational Unit	OU=″主体者組織単位名″ ＊ 証明書毎に任意に指定 例）ou= NII Open Domain CA	1
	commonName	CN=″サーバ FQDN″ ＊証明書毎に任意に指定 例）cn=www.nii.ac.jp	1
SubjectPublicKeyInfo		主体者の公開鍵　1024 ビット以上 （ただし、例外を認める）	2
attrobites		原則 Null 値とする （ただし、例外を認める）	3
SignatureAlgorithm		SHA1 with RSAEncryption	

1. 上記指定以外の属性を利用する必要がある場合には事前相談すること。少なくとも ST（state or province name）属性は使用しないこと。また、例えば加入者メールアドレスなど本プロジェクトの確認項目対象外の情報を含めないこと。
2. RSA1024bit 以上とする。鍵長 1024bit 未満の場合には事前に登録局へ相談すること。
3. 任意の属性を含めても構わないが、必ずしも証明書に反映されるわけではない。また、含めた属性によっては受理不能とし、当該属性を除いて証明書発行要求の再生成を登録局から求める場合がある。少なくとも SubjectAltName.rfc822Name 属性は使用しないこと。

2.4 鍵ペア生成及び CSR 作成前の準備

以下の項目を準備しておいてください。

- 乱数生成用ファイル（200KB 程度で少なくとも 3 つ）
- サーバ鍵ペアファイル名
- サーバ鍵ペア用私有鍵パスフレーズ
- サーバ DN（「2.2 DN のルール」に従った DN）
- CSR ファイル名

2.5 Openssl での鍵ペア生成と CSR 作成

Apache-SSL、Apache (mod_ssl)、IIS をお使いの加入者は openssl を利用して下記の手順で鍵ペアと CSR を生成してください。
CSR 作成時には、既存の鍵ペアは使わずに、必ず新たに CSR 作成用に生成した鍵ペアを利用してください。
更新時も同様に、必ず鍵ペアおよび CSR を新たに作成してください。
鍵ペアの鍵長は原則 1024bit 以上としてください。1024bit 未満の鍵を使用しなければならない場合、サーバ証明書申請前に本研究所まで連絡をください。

2.5.1 鍵ペアの生成

1. 鍵ペアを生成するため、お使いのハード・ディスクから、無作為データ元になるファイル (200 KB 程度) を 3 つ選んでください。この手順では、 3 つのファイルの名前を「*randfile1*」、「*randfile2*」、「*randfile3*」として表記します。

2. 鍵ペアの作成を始めるため、次のコマンドを入力してください(お使いのブラウザによっては 2 行以上で表示、印字されるかもしれませんが、実際は 1 行です)。
今回のコマンド例では、 1024 ビットの RSA 鍵ペアを生成し、「*servername.key*」という名前のファイルに保存することを示しています。

注) openssl コマンドの-rand 引数で指定するファイル名が複数の場合、セパレータとして使用する文字が OS により異なります。

■UNIX の場合－セパレータ文字':' (コロン)

```
openssl genrsa -des3 -rand randfile1:randfile2:randfile3  1024 > servername.key
```

■Windows の場合－セパレータ文字が';' (セミコロン)

```
openssl genrsa -des3 -rand randfile1;randfile2;randfile3  1024 > servername.key
```

3. プロンプトが表示されたら、鍵ペアを保護する私有鍵パスフレーズ (以下「鍵ペア用私有鍵パスパスフレーズ」とします) を決めて、入力してください。確認のため、2 度入力してください。(以下のは UNIX の場合の例です。2 の注の通り、Windows の場合はセパレータがセミコロンとなります)

```
$ openssl genrsa -des3 -rand randfile1:randfile2:randfile3 1024 > servername.key
0 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
..............................++++++
..............++++++
unable to write 'random state'
e is 65537 (0x10001)
Enter pass phrase:                                  ← 私有鍵パスフレーズ入力
Verifying - Enter pass phrase:                      ← 私有鍵パスフレーズ再入力
$
```

重要：この鍵ペア用私有鍵パスフレーズは、サーバの再起動時および証明書のインストール等に必要となる重要な情報です。鍵ペア利用期間中は忘れることがないよう、また、情報が他人に漏れることがないよう、安全な方法で管理してください。

4. 作成した鍵ペアのファイルを保存します。バックアップはフロッピーディスク等に保存し、安全な場所に保存してください。鍵ペアの中の私有鍵を利用すれば、お使いのウェブ・サーバが SSL/TLS で保護して送受信したデータを、解読することができてしまいます。従って保存する鍵ペアファイルへのアクセス権は加入者自身と SSL/TLS サーバのプロセス等必要最小限になるよう設定してください。またバックアップを保存したフロッピーディスク等も加入者のみまたは同じ権限のある方のみ利用できる場所へ保管してください。
   また、鍵ペア用私有鍵パスフレーズの管理も、確実に行ってください。鍵ペアファイルの紛失、鍵ペア用私有鍵パスフレーズ忘れ等が発生した場合、証明書のインストールが行えなくなります。この場合、新たに証明書を申請しなおしていただくことになりますので、ご注意ください。

2.5.2 CSR の生成

鍵ペアが作成されたことを確認後、 CSR を生成します。

1. 次のコマンドを入力し、 CSR の生成を開始してください。

```
openssl req -new -key servername.key - out servername.csr
```

と入力し、CSR の生成を開始します。ここで「*servername.key*」は、「2.5.1 鍵ペアの生成」で作成した鍵ペアのファイルの名前です。「*servername.csr*」は、CSR が保存されるファイルの名前です。

※現在利用中の鍵ファイルに上書きしないようご注意ください。

2. プロンプトが表示されたら、「2.5.1 鍵ペアの生成」で入力した鍵ペア用私有鍵パスフレーズを入力してください。

3. プロンプトが表示されたら、証明書内に組み込む SSL/TLS サーバ識別名 (DN)情報を入力してください。
   更新用 CSR 生成時に指定する DN 情報は、前回申請した内容 (現在利用中の証明書の DN) と同一にしてください。サイト名変更等により、DN 情報が変更になる場合は、新規のお申込となります。

```
$ openssl req -new -key servername.key -out servername.csr
Enter pass phrase for servername.key:                    ← 鍵ペア私有鍵パスフレーズ入力
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
Country Name (2 letter code) [AU]:JP                          ← “JP”を入力
State or Province Name (full name) [Some-State]:.             ← 「.」ドットの入力
Locality Name (eg, city) []:Academe                           ← “Academe”を入力
Organization Name (eg, company) [Internet Widgits Pty Ltd]:National Institute of
Informatics
                                                              ← 組織名を入力
Organizational Unit Name (eg, section) []:UPKI                ← 部局名を入力
Common Name (eg, YOUR name) []:www.nii.ac.jp                  ← サーバ名 FQDN を入力
Email Address []:.                                            ← 「.」ドットを入力

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.                                     ← 「.」ドットを入力
An optional company name []:.                                 ← 「.」ドットを入力
$
```

4. 要求された情報の入力が完了すると CSR が生成され、-out 引数で指定した名前のファイル（今回の例では、「***servername.csr***」）に保存されます。ファイルの内容には、次の例のような部分があります。この部分が CSR です。

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBhDCB7gIBADBFMQswCQYDVQQGEwJKUDEQMA4GA1UEBxMHQWNhZGVtZTEMMAoG
A1UEChMDTklJMRYwFAYDVQQDEw13d3cubmlpLmFjLmpwMIGfMA0GCSqGSIb3DQEB
                                    :
                                   例
                                    :
Vzf5ZJsnZ+muqe7gukNqRIW2cb/z8stdaMKYcbulAti7JaKHzPf+J9WapqSAx2iY
UmOE3vq8Ajg=
  -----END CERTIFICATE REQUEST-----
```

尚、CSR 生成時に入力した内容が記載されておりますので、証明書が発行され受け取るまでは、このファイルもバックアップを取って、別途保管するようお勧めします。

5. 次のコマンドを入力し、CSR の内容をご確認ください。

ここで「***servername.csr***」は、CSR を保存したファイルの名前です。

```
$ openssl req -noout -text -in servername.csr
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=JP, L=Academe, O=National Institute of Informatics, OU=UPKI,
CN=www.nii.ac.jp                    ← CSR 生成時に入力した DN と
                                        一致していることを確認してください。
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
```

```
                    00:c9:0e:99:5c:8a:4a:e3:b2:e2:0d:3d:60:4d:30:
                                    :
                                   例
                                    :
                    ca:2e:56:f7:66:bd:01:44:ea:f3:ca:d2:f6:e0:5e:
                    6c:57:4b:65:e4:e7:f7:ca:dd
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: sha1WithRSAEncryption
        88:44:e5:27:06:02:ec:85:6c:29:6a:0f:a3:92:87:4e:e2:f1:
                                    :
                                   例
                                    :
        9c:3c:0b:7e:1c:55:3d:c3:b3:7a:3a:36:d1:f6:3a:97:78:1a:
        c1:cc
$
```

6. ここで生成した CSR は、登録担当者に提出いただきます。

※重要

作成した CSR および鍵ペアのファイルは、必ずバックアップを取っていただき安全な場所に保管してください。また、鍵ペアのファイル生成時に指定した鍵ペア用私有鍵パスフレーズの管理も、確実に行ってください。鍵ペアのファイルの紛失、鍵ペア用私有鍵パスフレーズ忘れ等が発生した場合、証明書のインストールが行えなくなります。この場合は、新たに鍵ペアを生成して証明書を申請しなおしていただくことになりますので、ご注意ください。

2.6 Java keytool での鍵ペアと CSR 生成

2.6.1 Tomcat と JRE のバージョンによる注意点

※Jakarta Tomcat および JRE は、最新版に更新しておくことをおすすめします。

Jakarta Tomcat 4 で PureTLS をインストールしている場合には、JSSE を使うために server.xml ファイルで Factory 要素の SSLImplementation 属性の値を「org.apache.tomcat.util.net.JSSEImplementation」と明示的に指定するか、PureTLS をアンインストールするか、どちらかにしてください。

この文書では、Unix 版の JRE での操作例を掲載します。Windows 版では、実際の表記が異なります。

表 2-2 Unix と Windows の環境による違い

プラットホーム	変数置換	パス名	プロンプト	拡張子	ファイル連結コマンド
Unix	${変数名}	/ディレクトリ名/ファイル名	#や$	Sh	cat
DOS	%変数名%	ドライブ名:¥ディレクトリ名¥ファイル名	ドライブ名>	Bat	type

鍵ストアや鍵ペアの生成には、keytool というコマンドを使用して行います。

これは${JAVA_HOME}/bin ディレクトリにインストールされているはずです（${JAVA_HOME}は、JRE 1.4.2 を導入したディレクトリです)。
鍵ストアや鍵ペアを生成するには、次の手順に従い行なってください。

更新時も、必ず鍵ペアや CSR を新たに生成してください。

2.6.2 DN 情報入力時の注意点

「2.2 DN のルール」を参照してください。

2.6.3 鍵ペアの生成

1. 鍵ペアの生成を始めるため、次のようなコマンドを入力してください。

```
$ keytool -genkey -alias tomcat -keyalg RSA -keysize 1024 -keystore /your/keystore/filename -dname "CN=www.nii.ac.jp, OU=UPKI, O=National Institute of Informatics, L=Academe, C=JP"
```

上のコマンド例では、「*tomcat*」という名前をつけて 1024 ビットの RSA 鍵ペアを生成し、「*/your/keystore/filename*」という名前の鍵ストア・ファイルに、「CN=www.nii.ac.jp, OU=UPKI, O=National Institute of Informatics, L=Academe, C=JP」という DN で保存することを示しています。

ここで、「-dname」の後には、CSR に含む DN を指定します。「=」文字の後に指定する値については、「2.2 DN のルール」を参照してください。

更新用 CSR 生成時に指定するディスティングィッシュネーム(DN)情報は、前回申請した内容（現在利用中の証明書 DN）と同一にしてください。情報の入力にあたっては、次の点にご注意ください。

注意: 現在利用中の鍵ストアの鍵ペアに上書きしないよう、ご注意ください。

2. プロンプトが表示されたら、tomcat の鍵ストア用パスワードを入力してください。

言語環境	プロンプト（出力文字）
英語	Enter keystore password: *changeit*
日本語	キーストアのパスワードを入力してください: *changeit*

上のコマンド例では、「*changeit*」という鍵ストア用パスワードをつけたことを示しています。

注意: 入力した鍵ストア用パスワードは、エコーバックで表示されます。周囲に他人がいる場合、不用意にのぞかれたりしないよう、ご注意ください。

※重要: このパスワードは、証明書のインストールに必要となる重要な情報です。鍵ペア利用期間中は忘れることがないよう、また、他人に漏洩することがないよう、安全な方法で管理してください。

3. プロンプトが表示されたら、私有鍵のパスワードの入力を促されますので、鍵ストアのパスワードと同じにするなら、そのままリターンキーを、鍵ストアのパスワードと異なるパスワードを設定する場合はそのパスワードを入力してください。

言語環境	プロンプト（出力文字）
英語	Enter key password for <***tomcat***> (RETURN if same as keystore password):
日本語	<***tomcat***> の鍵パスワードを入力してください。 (キーストアのパスワードと同じ場合は RETURN を押してください):

4. 生成した鍵ストアを確認するため、次のコマンドを入力してください。

```
$ keytool -list -v - keystore /your/keystore/filename
```

上のコマンド例では、「*/your/keystore/filename*」という名前のファイルに保存した鍵ストアを確認することを示しています。

5. プロンプトが表示されたら、鍵ストア用パスワードを入力してください。

言語環境	プロンプト（出力文字）
英語	Enter keystore password: ***changeit***
日本語	キーストアのパスワードを入力してください: ***changeit***

注意: 入力したパスワードは、エコーバックで表示されます。周囲に他人のいる場合、 不用意にのぞかれたりしないよう、ご注意ください。

6. 鍵ストアのファイルの情報が表示されます。

```
Keystore type: JKS
Keystore provider: SUN
Your keystore contains 1 entry
Alias name: tomcat
Creation date: Mar 9, 2007
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=www.nii.ac.jp, OU=UPKI, O=National Institute of Informatics, L=Academe, C=JP
Issuer: CN=www.nii.ac.jp, OU=UPKI, O=National Institute of Informatics, L=Academe, C=JP
Serial number: 45f167e9
Valid from: Fri Mar 09 22:58:01 JST 2007 until: Thu Jun 07 22:58:01 JST 2007
Certificate fingerprints:
         MD5:  03:41:88:91:A3:44:D7:42:B8:38:6F:7A:C9:5B:5C:64
         SHA1: 1A:6E:CB:66:5E:79:7A:0D:88:17:3B:BB:25:3C:AE:1D:A2:F1:36:05
         Signature algorithm name: SHA1withRSA
         Version: 3
*******************************************
*******************************************
```

7. 鍵ストアのバックアップはフロッピーディスク等に保存し、安全な場所に保存してください。鍵ストアの中の鍵ペアを利用すれば、お使いの SSL/TLS サーバが SSL/TLS で保護して送受信したデータを、解読することができてしまいます。従って保存する鍵ペアへのアクセス権は加入者自身と SSL/TLS サーバのプロセス等必要最小限になるよう設定してください。またバックアップを保存したフロッピーディスク等も加入者のみまたは同じ権限のある方のみ利用できる場所へ保管してください。
   また、鍵ストアのファイル生成時に指定したパスワードの管理も、確実に行ってください。鍵ストアのファイルの紛失、パスワード忘れ等が発生した場合、証明書のインストールが行えなくなります。この場合、新たに証明書を申請しなおしていただくことになりますので、ご注意ください。

2.6.4 CSR の生成

CSR の生成には、keytool コマンドを使用して行います。
これは${JAVA_HOME}/bin ディレクトリにインストールされているはずです（${JAVA_HOME}は、JRE 1.4.2 を導入したディレクトリです)。

CSR を生成するには、次の手順に従い行なってください。更新時も、必ず CSR を新たに生成してください。
鍵ペアが生成できたことを確認後、CSR を生成します。

1. 次のコマンドを入力し、CSR の生成を開始してください。

```
$ keytool -certreq -sigalg SHA1withRSA -alias tomcat -file servername.csr -keystore /your/keystore/filename
```

と入力し、CSR の生成を開始します。ここで上記の文字は以下を示します。

文字列	意味
Tomcat	「2.6.3 鍵ペアの生成」で指定した指定した鍵ペアの名前
/your/keystore/filename	「2.6.3 鍵ペアの生成」で指定した鍵ストアのファイルの名前
Servername.csr	CSR が保存されるファイルの名前

2. プロンプトが表示されたら、鍵ストア用パスワードを入力してください。

```
Enter keystore password: changeit
```

注意: 入力したパスワードは、エコーバックでそのまま端末に表示されます。周囲に他人のいる場合、パスワードを不用意にのぞかれたりしないよう、ご注意ください。

3. CSR が生成され、手順 1 で指定した名前のファイル(今回の例では、「*servername.csr*」)に保存されます。ファイルの内容には、次の例のような部分があります。この部分が CSR です。

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBsjCCARsCAQAwcjELMAkGA1UEBhMCSlAxEDAOBgNVBAcTB0FjYWRlbWUxKjAoBgNVBAoTIU5h
dGlvbmFsIEluc3RpdHV0ZSBvZiBJbmZvcm1hdGljczENMAsGA1UECxMEVVBLSTEWMBQGA1UEAxMN
                                        :
                                       例
                                        :
pO9wBoF1r5fEKl6VCFC0qzfDQsKMKjbAOQJ0C+2wIHR84NBxmse0A6d1ret+Tk/RvM/gX2Mrg+9E
dKyJUY4lfTpLkX1sFyKBimGN8Y/PHsPr6rnL6IeVip63yu1Eiwxf
-----END NEW CERTIFICATE REQUEST-----
```

尚、CSR 生成時に入力した内容が記載されておりますので、証明書が発行され受け取るまでは、この CSR を保存したファイルもバックアップをとって、別途保管するようお勧めいたします。

4. openssl コマンドをインストールされている場合には、次のコマンドの入力で、CSR の内容が確認できます。

```
$ openssl req -noout -text -in servername.csr
```

ここで「***servername.csr***」は、CSR を保存したファイルの名前です。

5. ここで生成した CSR は、登録担当者に提出いただきます。

※重要

鍵ストアと CSR のバックアップはフロッピーディスク等に保存し、安全な場所に保存してください。鍵ストアの中の鍵ペアを利用すれば、お使いの SSL/TLS サーバーが SSL/TLS で保護して送受信したデータを、解読することができてしまいます。従って保存する鍵ペアへのアクセス権は加入者自身と SSL/TLS サーバのプロセス等必要最小限になるよう設定してください。またバックアップを保存したフロッピーディスク等も加入者のみまたは同じ権限のある方のみ利用できる場所へ保管してください。

鍵ストアのファイル生成時に指定したパスワードの管理も、確実に行ってください。鍵ストアのファイルの紛失、パスワード忘れ等が発生した場合、証明書のインストールが行えなくなります。この場合、新たに証明書を申請しなおしていただくことになりますので、ご注意ください。

2.6.5 SSL の有効化

SSL/TLS を有効にするため、次の作業をしていただきます。

1. 次のような Connector 要素を、server.xml ファイルに指定してください。

   - Tomcat 4 の場合

```
<!- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<!--
-->
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
    port="8443" minProcessors="5" maxProcessors="75"
    enableLookups="true"
    acceptCount="100" debug="0" scheme="https" secure="true"
    useURIValidationHack="false" disableUploadTimeout="true">
<Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
    keystoreFile="/your/keystore/filename" keystorePass="changeit"
    clientAuth="false" protocol="TLS"/>
/Connector>
<!--
  -->
```

- Tomcat 5.0　の場合

```
<!- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<!--
-->
<Connector  port="8443"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" disableUploadTimeout="true"
    acceptCount="100" debug="0" scheme="https" secure="true"
    keystoreFile="/your/keystore/filename" keystorePass="changeit"
    clientAuth="false" sslProtocol="TLS"/>
<!--
  -->
```

ここで上記内の文字列は以下を示します。

文字列	意味
8443	Jakarta Tomcat 4 が待ち受けるポートの番号
Changeit	「2.6.3 鍵ペアの生成」で指定した鍵ストアのパスワード
/your/keystore/filename	「2.6.3 鍵ペアの生成」で指定した鍵ストアのファイル名

注意: 鍵ストアのパスワードは、server.xml ファイルに平文で指定します。パスワードや私有鍵が他人に漏洩しないよう、このファイルや conf ディレクトリの許可モードなどにご注意ください。

2. Jakarta Tomcat 4 を(再)起動してください。

```
# $TOMCAT_HOME/bin/catalina.sh run
```

または

```
# $TOMCAT_HOME/bin/startup.sh
```

注意: 待ち受けるポート番号によっては、スーパーユーザー特権の不必要な場合もあります。

3. 「https://コモンネーム:ポート番号/」の URL で閲覧できることを、ウェブ・ブラウザでご確認ください(この時点では、SSL のサーバ証明書が自己署名になっていますので、ブラウザによっては、警告のダイアログなどの出る場合もあります)。

4. Admin サーブレットにアクセスできるよう設定してから、「https://コモンネーム:ポート番号/admin/index.jsp」の URL で、ログイン画面にアクセスできることを、ご確認ください。

5. 適切なユーザ名とパスワードを入力して、Admin サーブレットにログインしてください。

6. 「サービス」の下にある Connector (8443)を選択してから、指定した鍵ストアのファイルが読み込まれていることを、ご確認ください。

ここで「8443」は、Jakarta Tomcat 4 が待ち受けるポートの番号です。

※重要

鍵ストアのパスワードを指定する server.xml ファイルや、そのファイルのある conf ディレクトリのアクセス管理を、必要最小限のユーザ、プロセスのみに限定するよう確実に行ってください。鍵ストアのファイルや、パスワードの紛失等が発生した場合、証明書のインストールが行えなくなります。この場合、新たに鍵ペアを生成して証明書を申請しなおしていただくことになりますので、ご注意ください。

3. 証明書のインストール

3.1 事前準備

3.1.1 証明書の取得

NII オープンドメイン認証局の証明書を利用する際に必要となる証明書の取得方法に関しては、「1.3 証明書の種類」を参照してください。

3.1.2 IIS 利用の場合 openssl による PKCS#12 ファイルの作成

IIS を利用する場合、事前に鍵ペアとサイト証明書（SSL/TLS 証明書）を連結した PKCS#12 ファイルを作る必要があります。PKCS とは RSA Security 社が提唱した仕様で公開鍵／私有鍵や証明書を利用するほとんどのシステムでサポートされています。その中に PKCS#12 があり、鍵ペアと証明書等を PKCS#12 の仕様で決められたフォーマットに従い暗号化された状態で 1 つのファイルとすることができるものです。

以下のコマンドで openssl を使って鍵ペアとサイト証明書（SSL/TLS 証明書）を連結した PKCS#12 ファイルを作ることができます。

```
# openssl pkcs12 -export -inkey servername.key -in Example.crt -out servername.pfx
```

コマンドを実行すると、以下のようなプロンプトが順次表示されます。最初に鍵ペア用私有鍵パスフレーズを入力します。次に PKCS#12 ファイルを開くためのパスフレーズを 2 回入力します。後者を PKCS#12 保護パスフレーズと呼ぶことにします。

```
Enter pass phrase for servername.key:
Enter Export Password:                    ← PKCS#12 保護パスフレーズ入力
Verifying - Enter Export Password:        ← PKCS#12 保護パスフレーズ再入力
```

これで鍵ペアとサイト証明書（SSL/TLS 証明書）を連結した PKCS#12 の「*servername.pfx*」が作成されます。

3.1.3 IIS 利用の場合 Security Communication RootCA1 認証局の自己署名証明書の削除

Windows のバージョンや Windows Update の時期によっては Security Communication RootCA1 認証局の自己署名証明書が登録されています。この自己署名証明書が登録された状態では SSL/TLS の通信の際に Security Communication RootCA1 認証局の中間 CA 証明書が利用されない状態となってしまい、接続不能のブラウザ等がでてしまいます。従って、Security Communication RootCA1 認証局の自己署名証明書を削除する必要があります。

サーバの「コントロールパネル」を開いて「インターネットオプション」を開いて、「コンテンツ」タグを開いた後、「証明書」のボタンより「信頼されたルート証明機関」をクリックしてください。

次の証明書の有無を確認してください。

発行先：Security Communication RootCA1
発行者：Security Communication RootCA1
有効期限：2003/09/30 から 2023/09/30

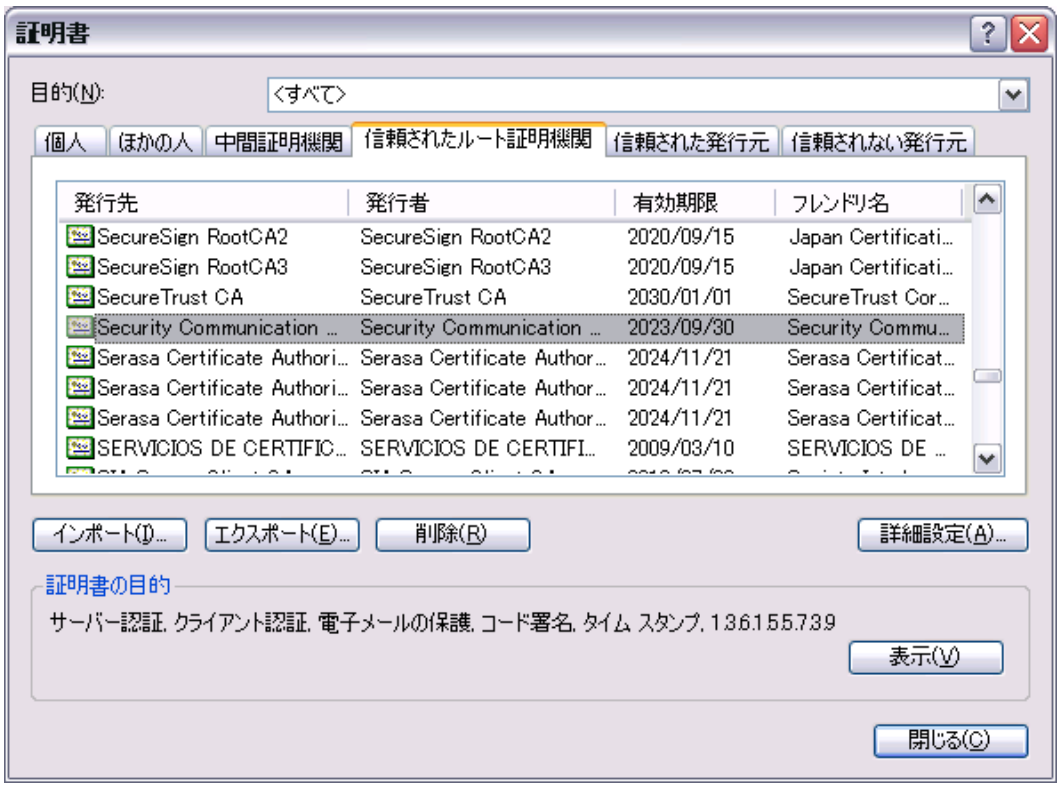


図 3-1 Windows に登録されている Security Communication RootCA1 認証局の自己署名証明書

上記証明書が確認できない場合は削除作業は不要です。
対象の証明書が選択された状態で 「削除(R)」をクリックします。

※誤って別の証明書を削除することがないようご注意ください。
※証明書を削除する前に、エクスポートすることによって証明書をバックアップできます。
削除確認のダイアログがでますので、「はい」を選択して削除完了です。

※重要
本作業の後、WindowsUpdate 等により Security Communication RootCA1 の自己署名証明書を インストールすることのないようにご注意ください。

3.2 Apache-SSL

証明書のインストールにあたり、1.3節で示した 2 種類の証明書を以下の手順に準じてインストールしていただく必要があります。

1. 中間 CA 証明書をインストールします。
2. 中間 CA 証明書をインストール後、サイト証明書（SSL/TLS サーバ証明書）をインストールします。
3. Apache サーバのデーモン・プロセスを再起動します。

以下の SSLCACertificatePath、SSLCertificateFile、SSLCertificateKeyFile 等のエントリは、Apache の立ち上げファイルの中にあります。
デフォルトの立ち上げファイル名は「httpd.conf」です。

立ち上げファイルでは、「#」文字で行内コメントが始まります。以下で使用する各種のエントリの前に「#」文字のないことを確認してください。

3.2.1 中間 CA 証明書のインストール方法

以下の手順に従って、SSLCACertificatePath エントリによって指定されるディレクトリに、中間 CA 証明書をインストールしてください。
※SSLCACertificateFile は設定しないでください。

1. この手順は、SSLCACertificatePath が立ち上げファイルで*<SSLTOP>*/CA に設定されていることを前提としています。
   ここで、*<SSLTOP>*は Apache-SSL で SSL を設定する各種ファイルのトップディレクトリを示しています。異なるパスを使用している場合は、正しいパスに適宜読み替えてください。

2. テキストファイルで中間 CA 証明書を保存していることを確認します。

3. SecurityCommunicationRootCA1 の証明書ファイルをディレクトリ*<SSLTOP>*/CA/に移動します。SecurityCommunicationRootCA1 の中間 CA 証明書ファイルのパス名は、/tmp/*scroot1.crt* としています。

```
# mv /tmp/scroot1.crt <SSLTOP>/CA/a3896b44.0
```

4. NII オープンドメイン認証局の証明書ファイルをディレクトリ*<SSLTOP>*/CA/に移動します。NII オープンドメイン認証局の中間 CA 証明書ファイルのパス名は、/tmp/*niica.crt* としています。

```
# mv /tmp/niica.crt <SSLTOP>/CA/957e1f22.0
```

3.2.2 サイト証明書（SSL/TLS サーバ証明書）のインストール方法

以下の手順に従って、サイト証明書(SSL/TLS サーバ証明書)をインストールしてください。

1. この手順では、SSLCertificateFile が立ち上げファイルで*<SSLTOP>*/conf/ssl.crt/server.crt に設定されていることを、前提としています。また、SSLCertificateKeyFile が立上げファイルで*<SSLTOP>*/conf/ssl.key/server.key に設定されていることを、前提としています。
   ここで、*<SSLTOP>*は、SSL のトップディレクトリです。異なるパスを使用している場合、正しいパスに適宜読み替えてください。またサイト証明書（SSL/TLS サーバ証明書）ファイルのパス名は、/tmp/*Example.crt* としています。

2. サイト証明書（SSL/TLS サーバ証明書）をテキストファイルで保存していることを確認してください。

3. サイト証明書（SSL/TLS サーバ証明書）を、*<SSLTOP>*/conf/ssl.crt/server.crt のファイルに移動してください。

```
mv /tmp/Example.crt <SSLTOP>/conf/ssl.crt/server.crt
```

4. 2.5.1で作成した、サイト証明書（SSL/TLS サーバ証明書）に対応する鍵ペアのファイルを、*<SSLTOP>*/conf/ssl.key/server.key のファイルに移動してください。

5. 立上げファイル内に設定されている SSLCertificateKeyFile が、鍵ペアのファイルを指していることを確認してください。

証明書のインストールは、以上で完了です。

3.3 Apache（mod_ssl）

証明書のインストールにあたり、1.3節で示した 2 種類の証明書を以下の手順に準じてインストールしていただく必要があります。

1. 中間 CA 証明書をインストールします。
2. 中間 CA 証明書をインストール後、サイト証明書（SSL/TLS サーバ証明書）をインストールします。
3. Apache サーバのデーモン・プロセスを再起動します。

以下の SSLCertificateChainFile、SSLCertificateFile、SSLCertificateKeyFile 等のエントリは、Apache の立ち上げファイルの中にあります。
デフォルトの立ち上げファイル名は「httpd.conf」です。

立ち上げファイルでは、「#」文字で行内コメントが始まります。以下で使用する各種のエントリの前に「#」文字のないことを確認してください。

3.3.1 中間 CA 証明書のインストール方法

以下の手順に従って、SSLCertificateChainFile エントリによって指定されるファイルに中間 CA 証明書をインストールしてください。

1. この手順では、SSLCertificateChainFile が立上げファイルで *<SSLTOP>*/conf/ssl.crt/ca.crt ファイルに設定されていることを、前提としています。ここで、*<SSLTOP>*は、Apache(mod_ssl)で SSL を設定する各種ファイルのトップディレクトリを示しています。トップディレクトリの下で異なるパスを使用している場合、正しいパスに適宜読み替えてください。

2. 中間 CA 証明書をテキストファイルで保存していることを確認してください。

3. SecurityCommunicationRootCA1 認証局証明書ファイルをディレクトリ *<SSLTOP>*/conf/ssl.crt/ に移動し、ファイル名をリネームします。SecurityCommunicationRootCA1 認証局の中間 CA 証明書ファイルのパス名は、/tmp/*scroot1.crt* としています。

```
mv /tmp/scroot1.crt <SSLTOP>/conf/ssl.crt/ca.crt
```

4. NII オープンドメイン認証局の中間 CA 証明書ファイルを *<SSLTOP>*/conf/ssl.crt/ca.crt というファイルに追加してください。NII オープンドメイン認証局の中間 CA 証明書ファイルのパス名は、/tmp/*niica.crt* としています。

```
cat /tmp/niica.crt >> <SSLTOP>/conf/ssl.crt/ca.crt
```

3.3.2 サイト証明書（SSL/TLS サーバ証明書）のインストール方法

以下の手順に従って、 サイト証明書（SSL/TLS サーバ証明書）をインストールしてください。

1. この手順では、SSLCertificateFile が立上げファイルで<SSLTOP>/conf/ssl.crt/server.crt に設定されていることを、前提としています。また、SSLCertificateKeyFile が立上げファイルで<SSLTOP>/conf/ssl.key/server.key に設定されていることを、前提としています。
   ここで、<SSLTOP>は、SSL のトップディレクトリです。トップディレクトリの下で異なるパスを使用している場合、正しいパスに適宜読み替えてください。
   またサイト証明書（SSL/TLS サーバ証明書）ファイルのパス名は、/tmp/Example.crt としています。

2. サイト証明書（SSL/TLS サーバ証明書）をテキストファイルで保存していることを確認してください。

3. サイト証明書（SSL/TLS サーバ証明書）を、<SSLTOP>/conf/ssl.crt/server.crt のファイルに移動してください。

```
mv /tmp/Example.crt <SSLTOP>/conf/ssl.crt/server.crt
```

4. 2.5.1で作成した、サイト証明書（SSL/TLS サーバ証明書）に対応する鍵ペアのファイルを、<SSLTOP>/conf/ssl.key/server.key のファイルに保存してください。

5. 立上げファイル内に設定されている SSLCertificateKeyFile が、鍵ペアのファイルを指していることを確認してください。

証明書のインストールは、以上で完了です。

3.4 Jakarta Tomcat

3.4.1 証明書のインストール方法

証明書をインストールするため、次の作業をしていただきます。

1. 次のコマンドを入力し、1.3節で示した中間 CA 証明書とサイト証明書の 3 枚の証明書をつないでください。

```
$ cat Example.crt niica.crt scrootl.crt > combined-chain-and-webcert
```

2. 次のコマンドを入力し、つないだ証明書をインストールしてください。

```
$ keytool -import -alias tomcat -file combined-chain-and-webcert -keystore /your/keystore/filename - trustcacerts
```

ここで上記のコマンド内にある文字列の意味は以下の通りです。

表 3-1 keytool コマンドの文字列の意味

文字列	意味
Tomcat	「2.6.3鍵ペアの生成」で作成した鍵ペアの名前
combined-chain-and-webcert	上記手順 1 でつないだ中間 CA 証明書とサイト証明書のファイルの名前
/your/keystore/filename	「2.6.3鍵ペアの生成」で指定した鍵ストアのファイルの名前

3. プロンプトが表示されたらパスワードを入力してください。

```
Enter keystore password: changeit
```

ここで「***changeit***」は、「2.6.3鍵ペアの生成」で指定した鍵ストアのパスワードです。

注意: 入力したパスワードは、エコーバックでそのまま端末に表示されます。周囲に他人のいる場合、パスワードを不用意にのぞかれたりしないよう、ご注意ください。

4. 次のようなプロンプトの表示される場合は「yes」と入力してください。

```
Top-level certificate in reply:
Owner: OU=Security Communication RootCA1, O=SECOM Trust.net, C=JP
Issuer: CN=http://www.valicert.com//emailAddress=info@valicert.com, OU=ValiCert Class
1 Policy Validation Authority, O=ValiCert, Inc., L=ValiCert Validation Network
Serial number: 10002
Valid from: Tue Mar 16 11:34:42 JST 2005 until: Mon Mar 11 11:34:42 JST 2015
Certificate fingerprints:
  MD5: BC:88:BB:D7:36:2C:63:8C:F3:F6:CB:D9:69:B3:DC:83
  SHA1: AB:A0:A6:0A:3B:A7:D2:2C:0D:CF:77:41:D8:F5:5C:B3:C1:24:1D:F0

... is not trusted. Install reply anyway? [no]: yes
```

5. 次のように表示されることを、ご確認ください。

```
Certificate reply was installed in keystore
```

6. この確認がとれれば不要になりますので、つないだ証明書のファイルは、次のようなコマンドの入力で削除できます。

```
$ rm - f combined-chain-and-webcert
```

※ 重要

証明書のインストール後、鍵ストアのファイルは、必ずバックアップをとり、鍵ペア利用期間中はパスワードの保管場所と別の安全な場所に保管してください。

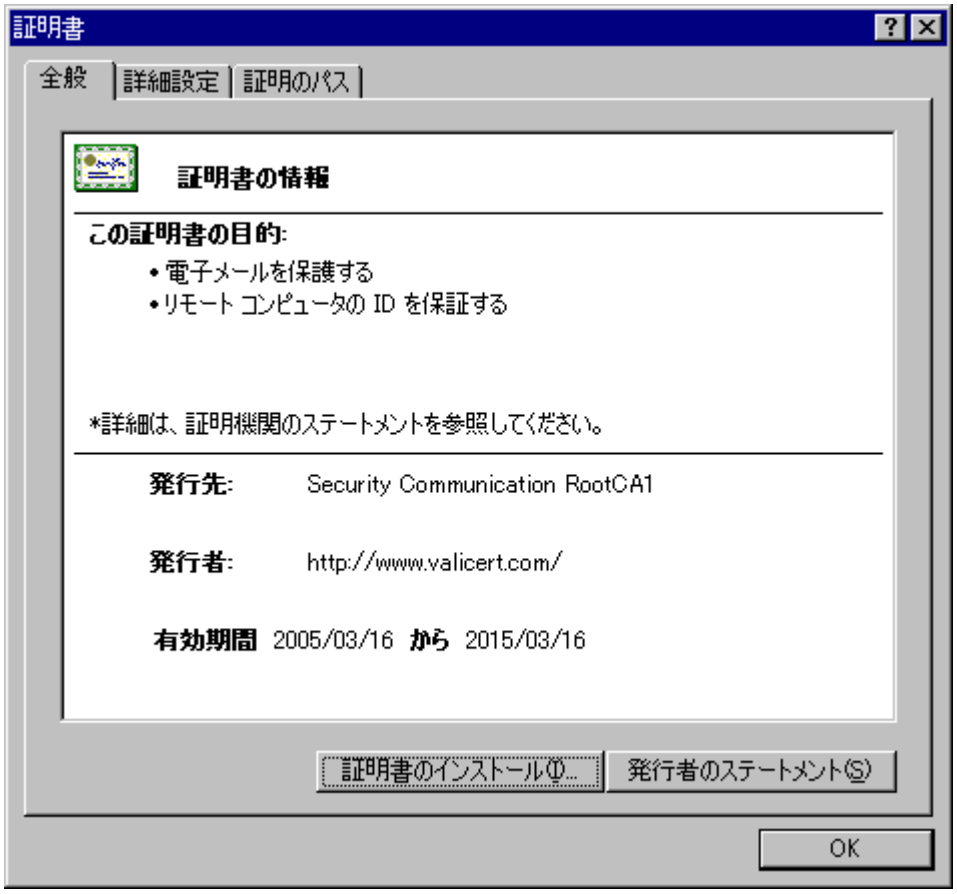
3.5 IIS5

証明書のインストールにあたり、必要な証明書に関しては「1.3 証明書の種類」を参照してください。

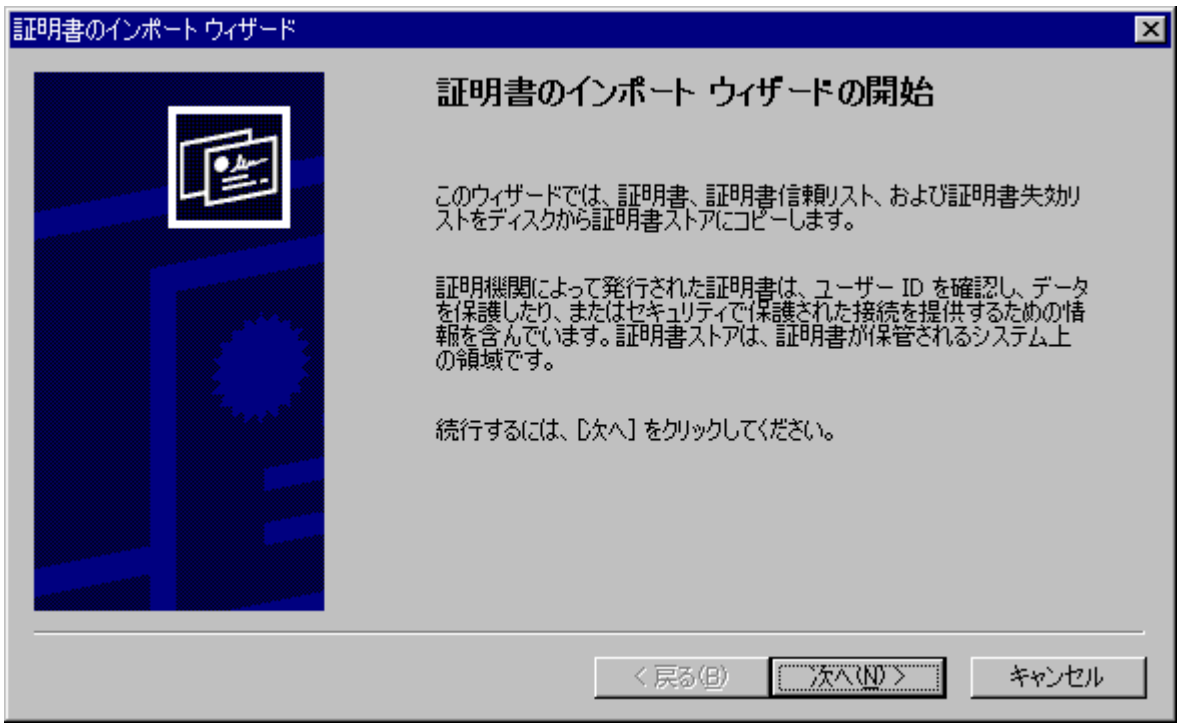
3.5.1 中間 CA 証明書のインストール方法

1. 中間 CA 証明書の SecurityCommunicationRootCA1 認証局証明書（scroot1.crt）と NII オープンドメイン認証局証明書（niica.crt）を順番にインストール作業を行います。

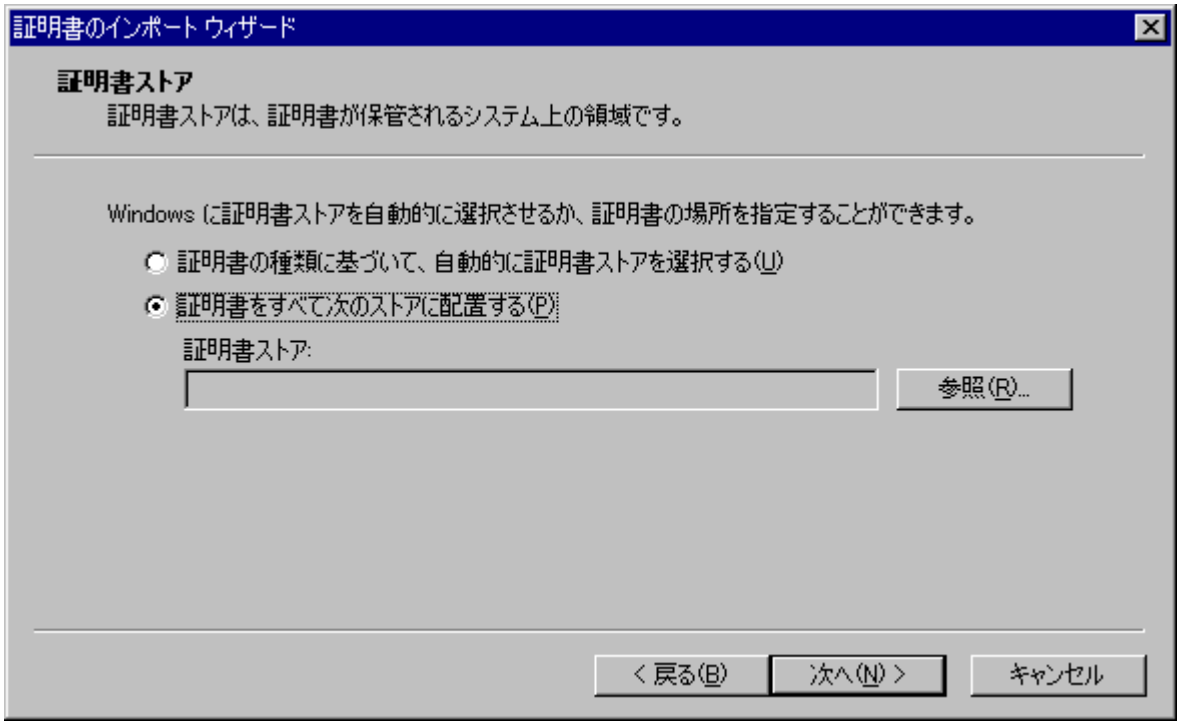
2. 証明書ダイアログボックスが表示されます。「全般」タブの[証明書のインストール]をクリックします。



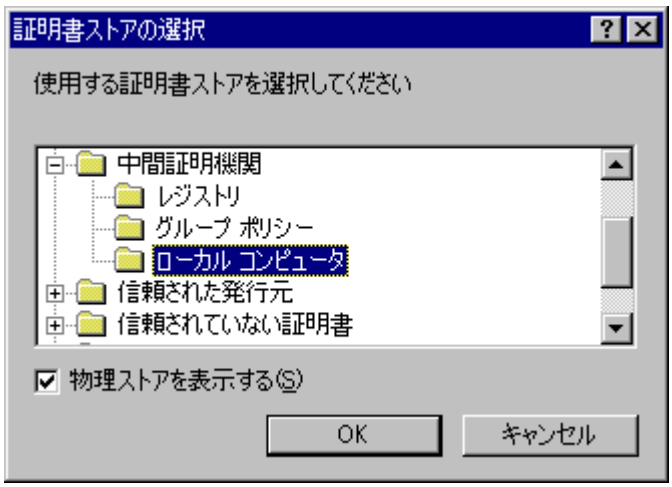
3. 証明書マネージャインポートウィザードが表示されますので、「次へ」をクリックしてください。



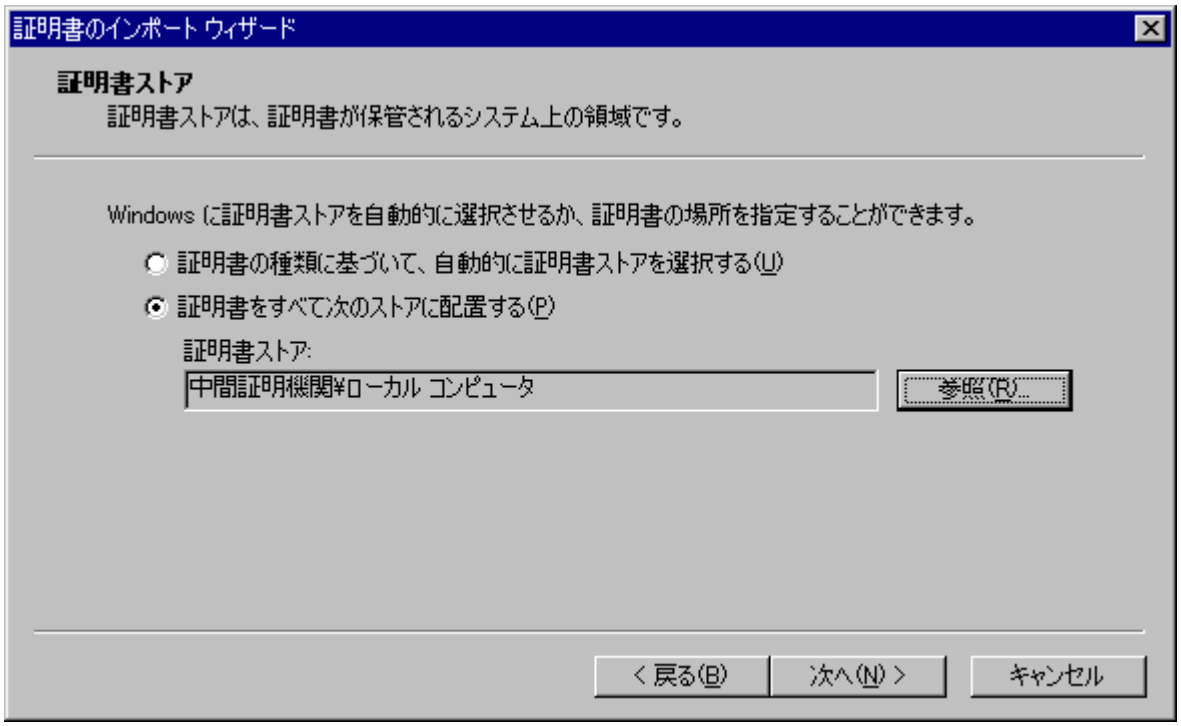
4. ［証明書をすべて次のストアに配置する］を選択し、［参照］をクリックします。



5. 証明書ストア選択ダイアログボックスが表示されますので、［物理ストアを表示する］を選択します。
ダイアログボックス内の項目［中間証明機関］のそばにある［+］マークをクリックして拡張し、［ローカルコンピュータ］を選択し、［OK］をクリックします。



6. 証明書ストアに「中間証明機関¥ローカルコンピュータ」が表示されている事を確認し、「次へ」をクリックしてください。



7. 「証明書のインポートウィザードの完了」が表示されたら、「完了」をクリックしてください。

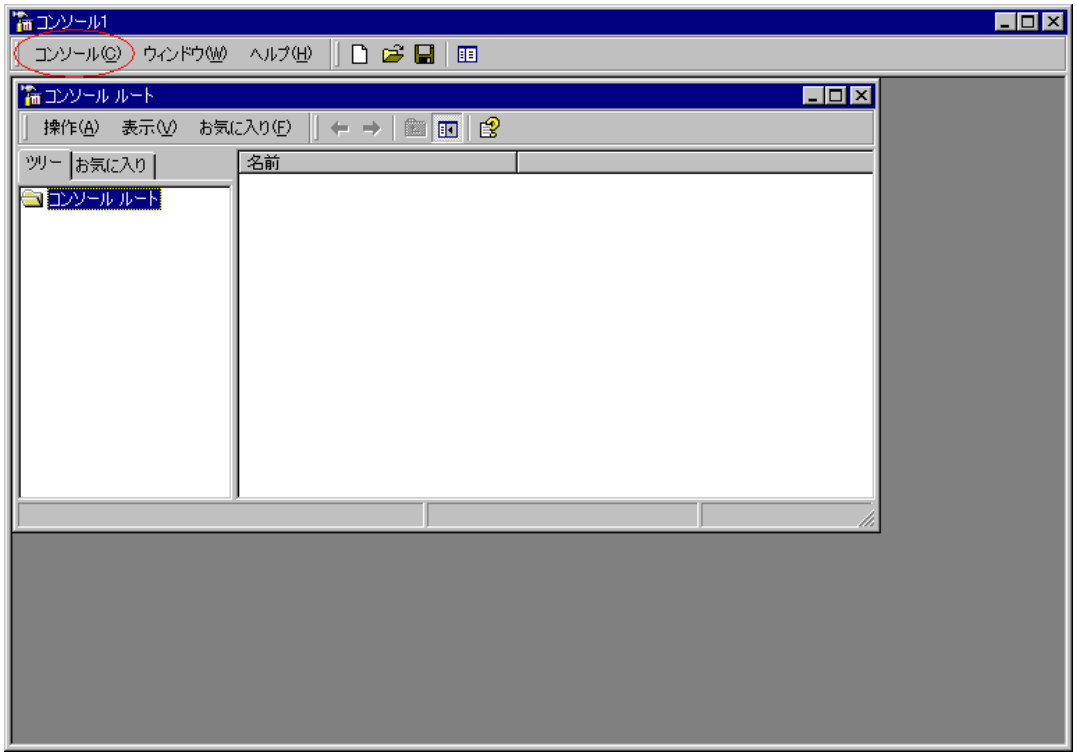
8. 証明書のインポートウィザードが表示されます。[OK]をクリックします。

SecurityCommunicationRootCA1 認証局証明書及び NII オープンドメイン認証局証明書のインストールが終了したら、サイト証明書（SSL/TLS サーバ証明書）をインストールします。

3.5.2 サイト証明書（SSL/TLS サーバ証明書）のインストール方法

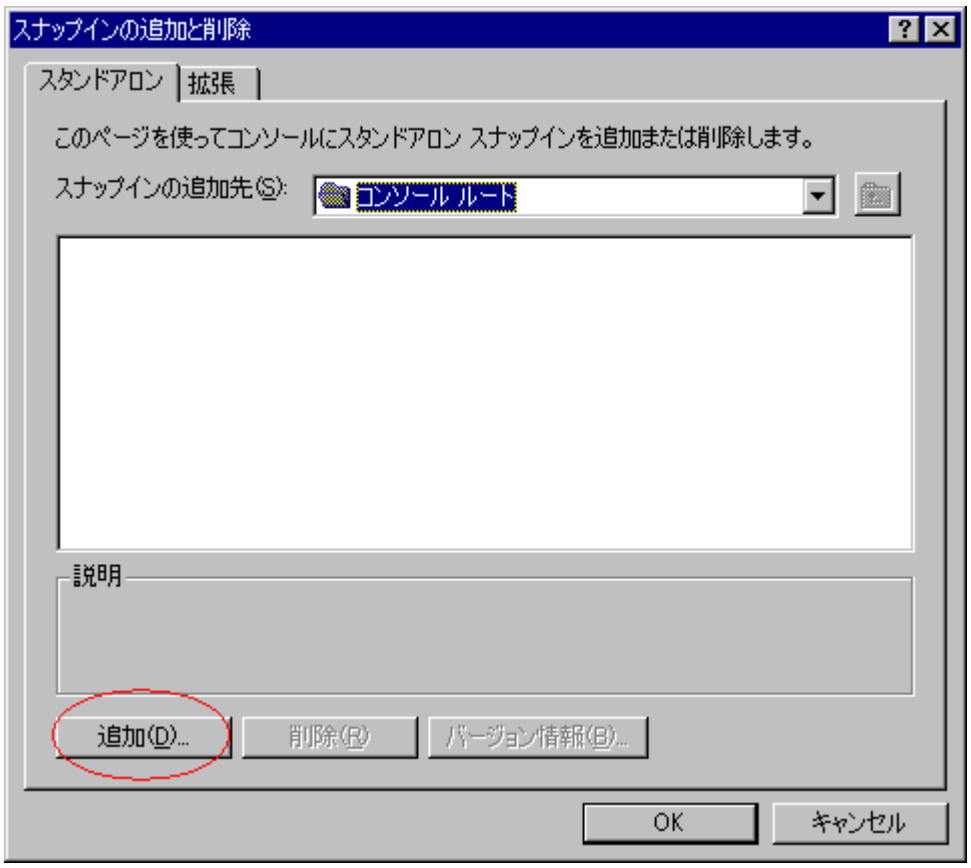
1. 「3.1.2 IIS 利用の場合 openssl による PKCS#12 ファイルの作成」を参照して「***servername.pfx***」ファイルを準備します。

2. 「ファイル名を指定して実行」より「MMC」と入力し MMC コンソールを立ち上げます



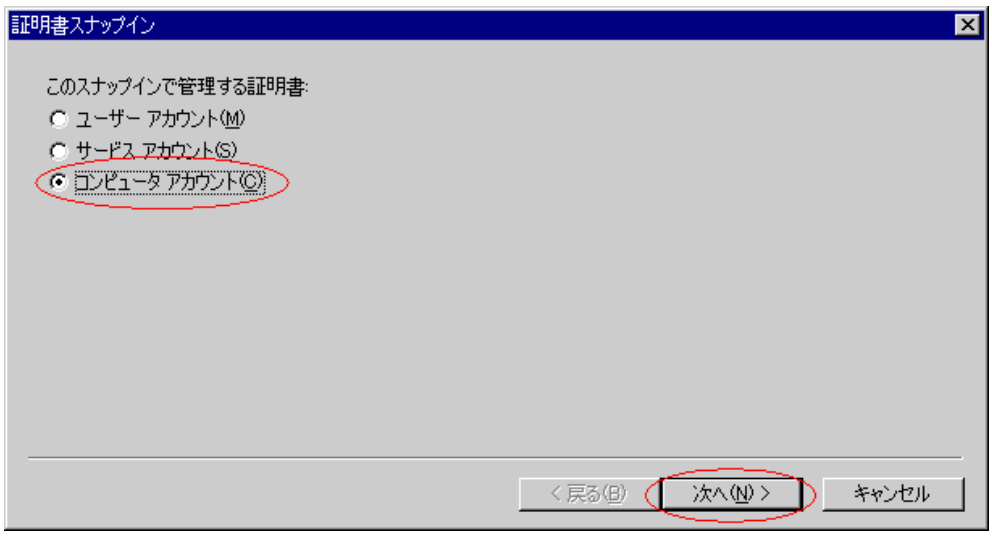
3. MMC コンソールより「コンソール」メニューの「スナップインの追加と削除」を選択します。

4. 追加をクリックします。

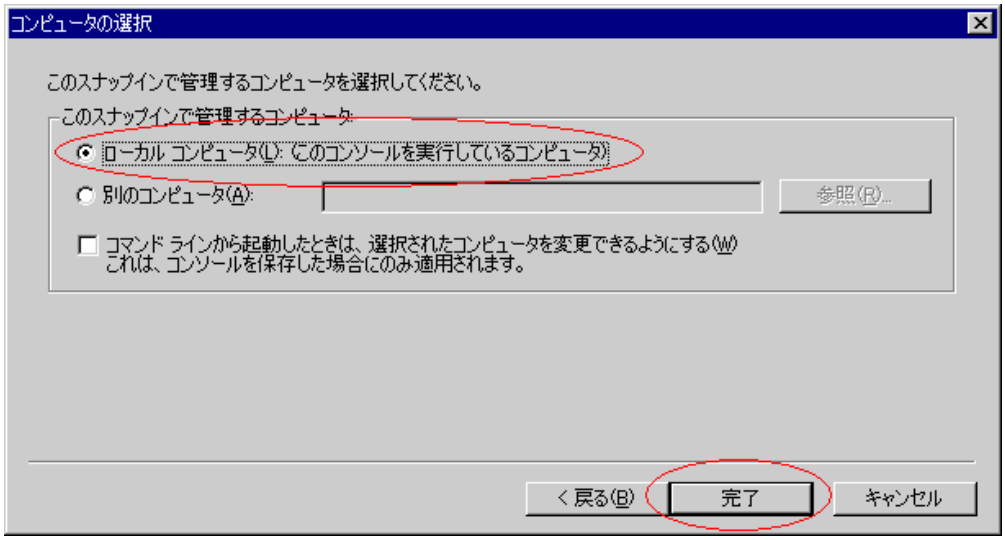


5.　「証明書」を選択し、「追加」をクリックします。

6.　「コンピュータアカウント」を選択し、「次へ」をクリックします。

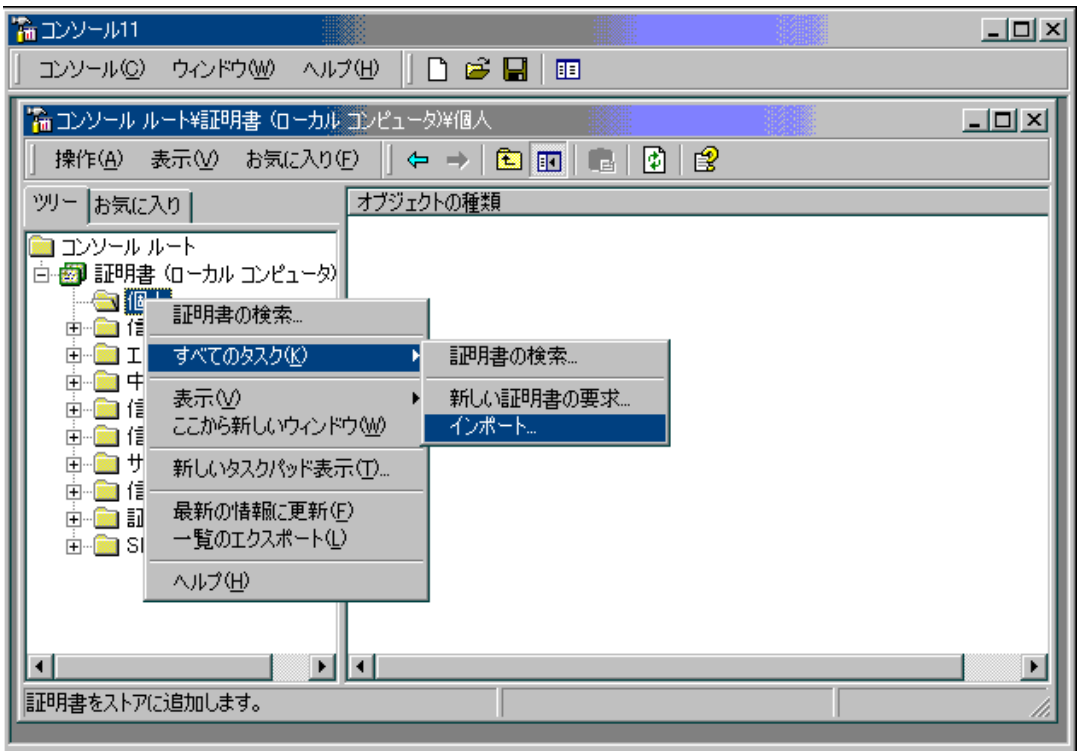


7. 「ローカルコンピュータ」を選択し、「完了」をクリックします。「スタンドアロン スナップインの追加」のウィンドウは「閉じる」にて Close してください。

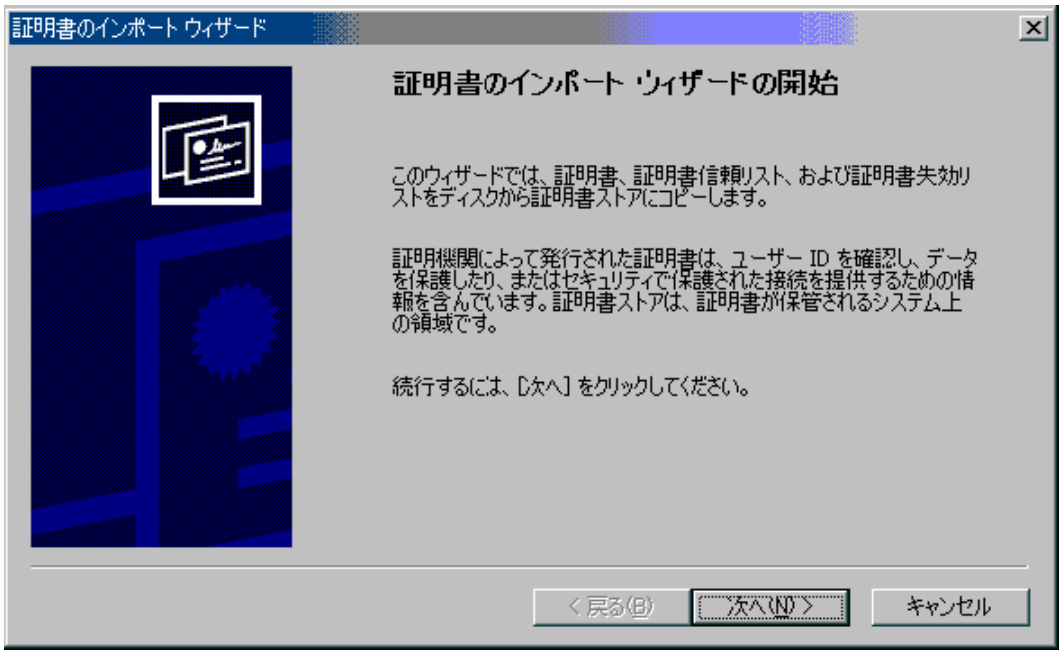


8. 「スナップインの追加と削除」のウィンドウにて「OK」をクリックし、ウィンドウを閉じます。

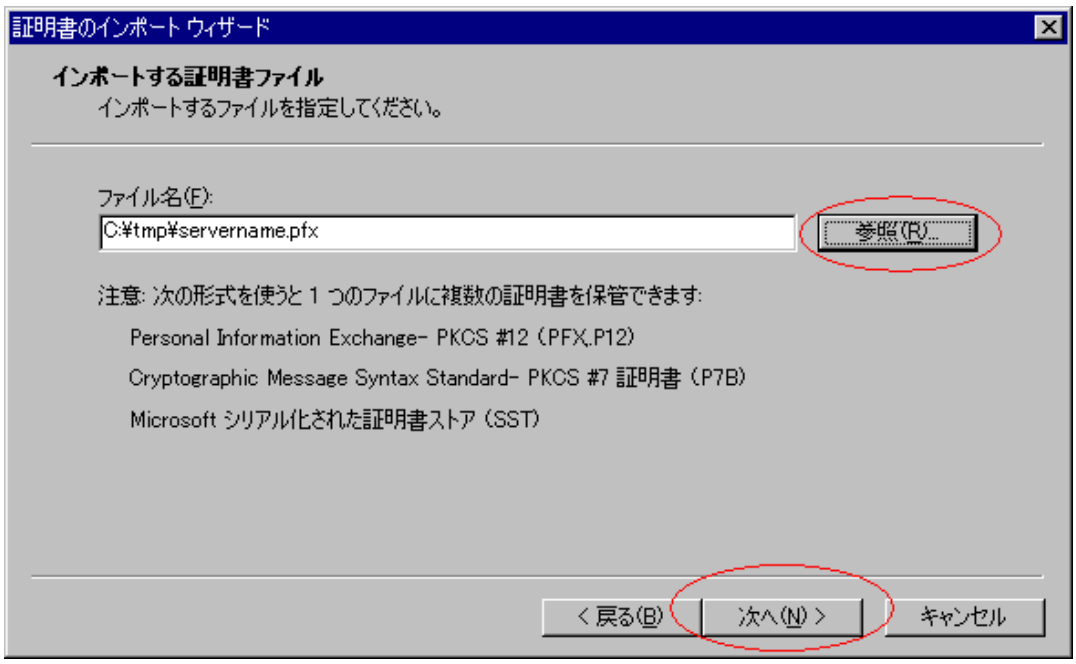
9. 「証明書（ローカルコンピュータ)」の＋を展開し、「個人」ストアをマウスの右ボタンでポイントし、「すべてのタスク」から「インポート」を選択します。



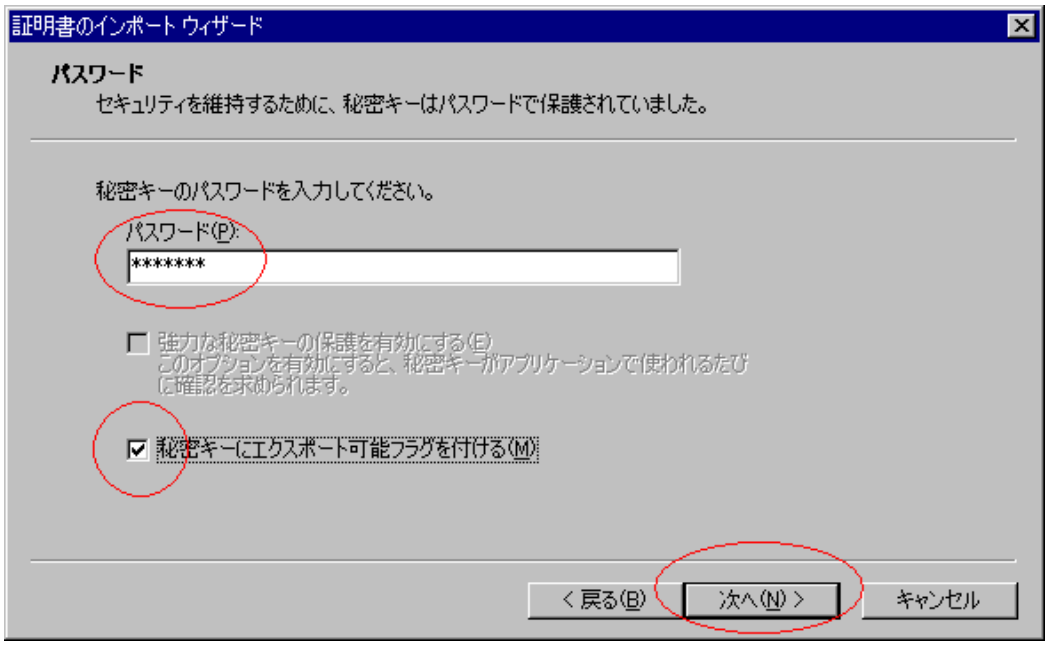
10.　「証明書のインポートウィザード」が起動します。「次へ」をクリックします。



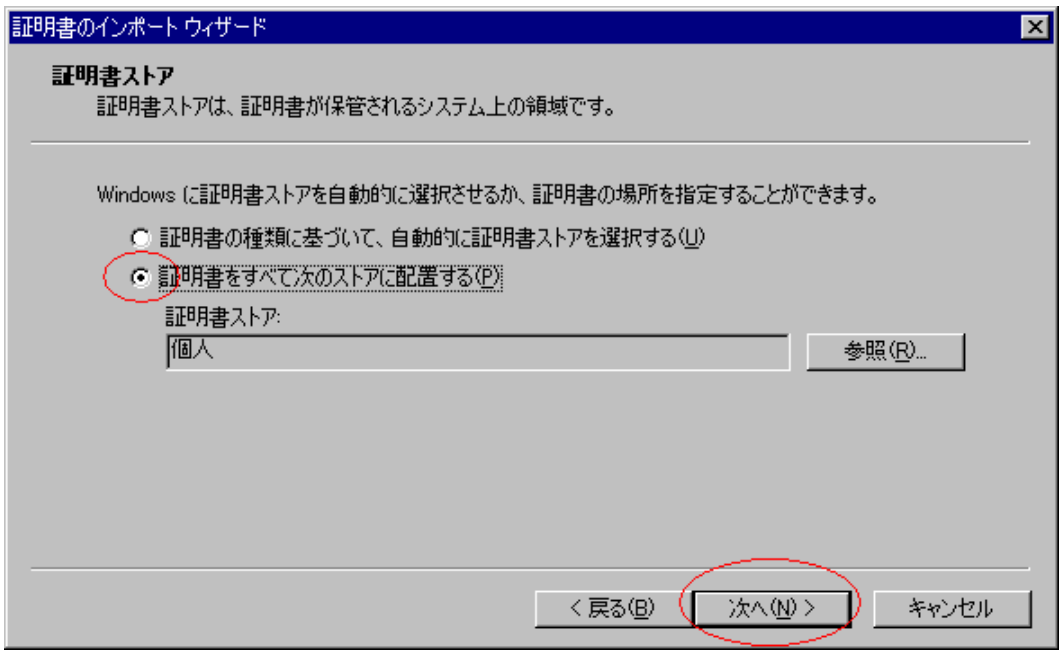
11.　「参照」クリックし、手順 1 で準備した「*servername.pfx*」を指定し、「次へ」をクリックします。



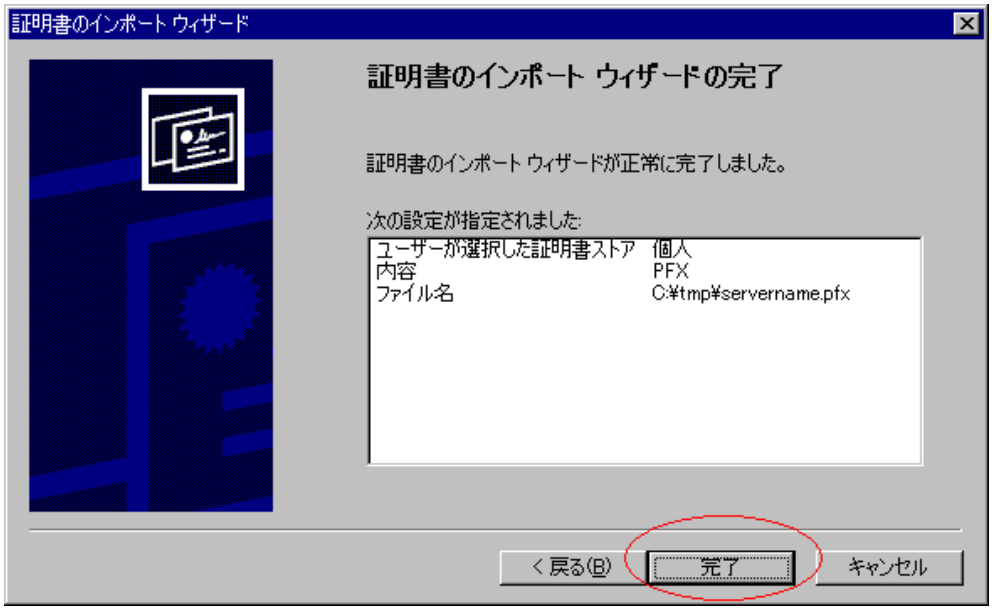
12. PKCS#12 ファイルを作る際に指定した PKCS#12 保護パスフレーズを入力し、「秘密キーにエクスポート可能フラグを付ける」をチェックし、「次へ」をクリックします。



13. 「証明書をすべて次のストアに配置する」を選択し、「次へ」をクリックします。



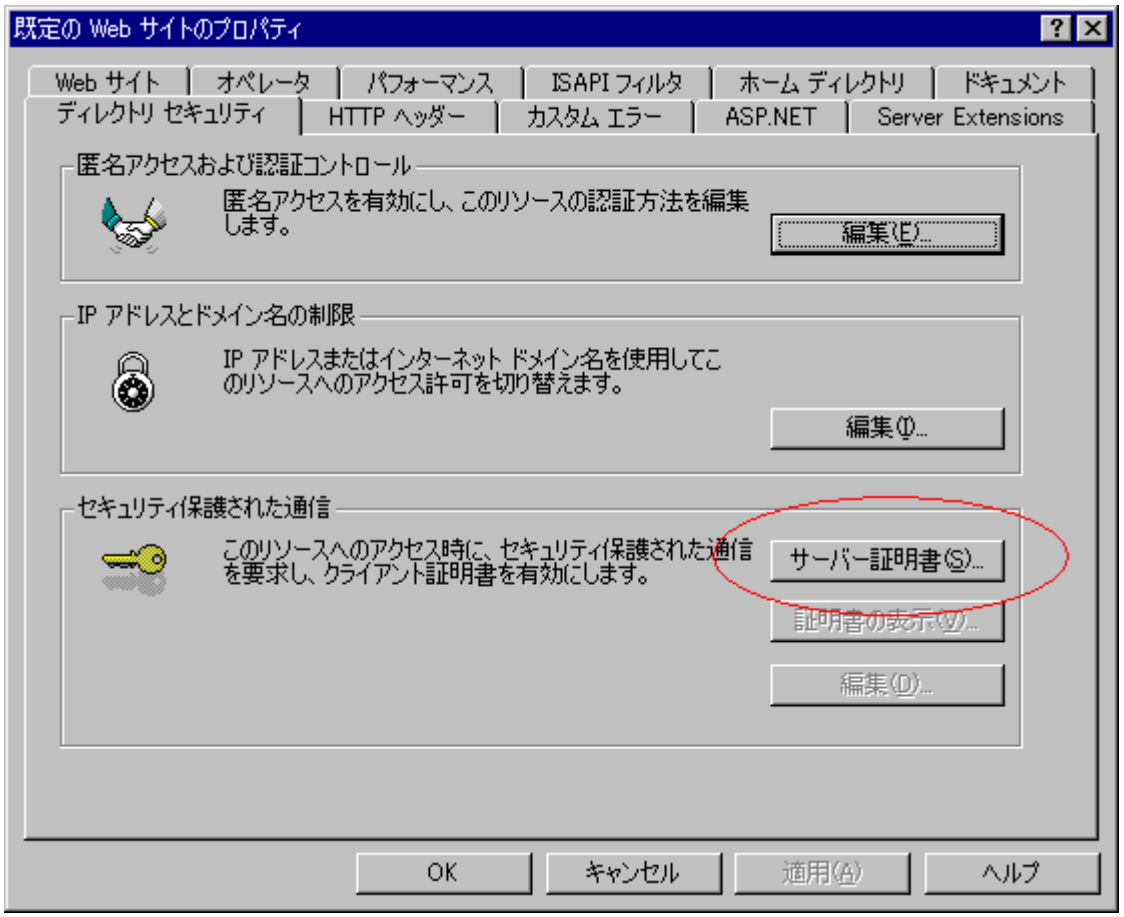
14. 「完了」をクリックします。



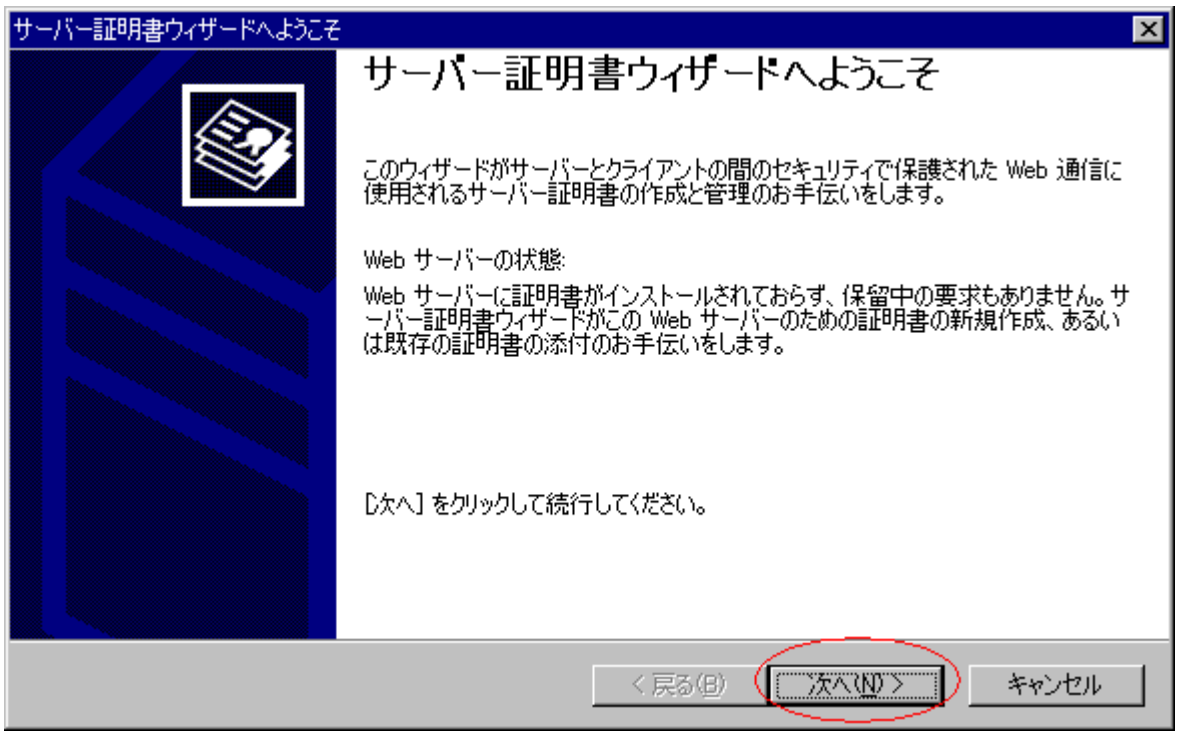
15. 「OK」をクリックします。

16. 「インターネットサービスマネージャ」を起動し、サイトのプロパティを開きます。

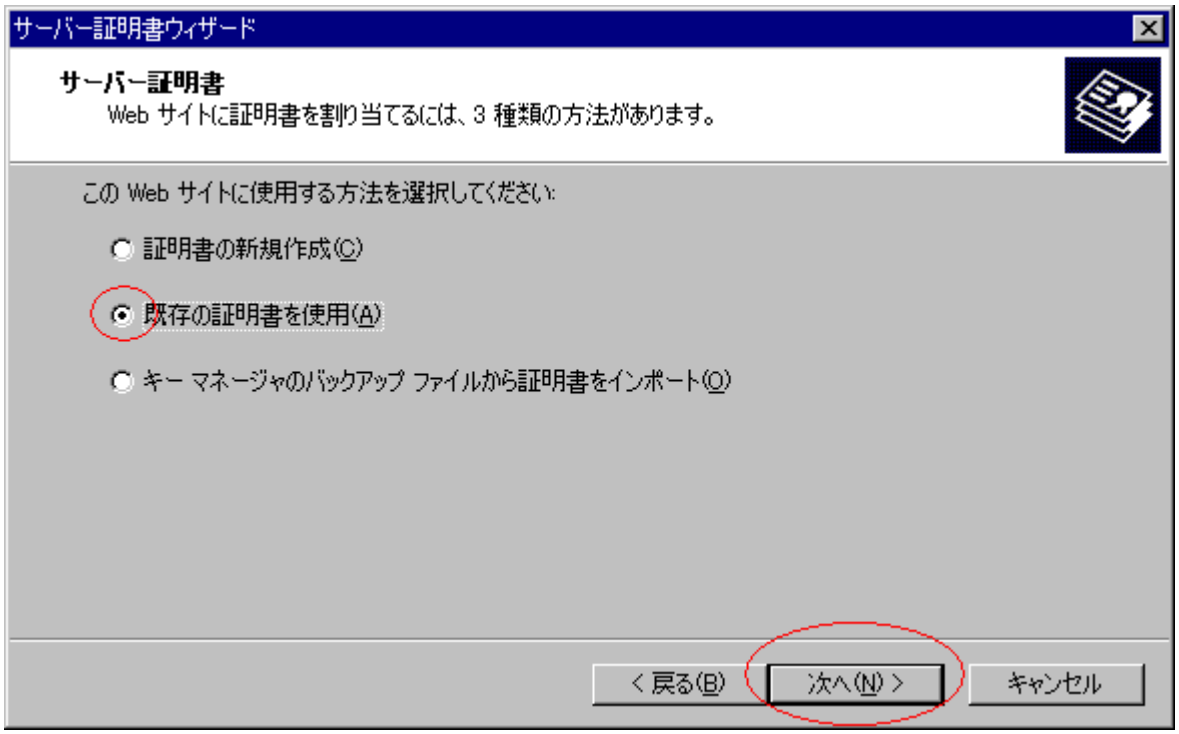
17. 「ディレクトリ セキュリティ」のタグより「セキュリティ保護された通信」の「サーバー証明書」を開きます。



18. 「サーバ証明書ウィザード」が起動します。「次へ」をクリックします。

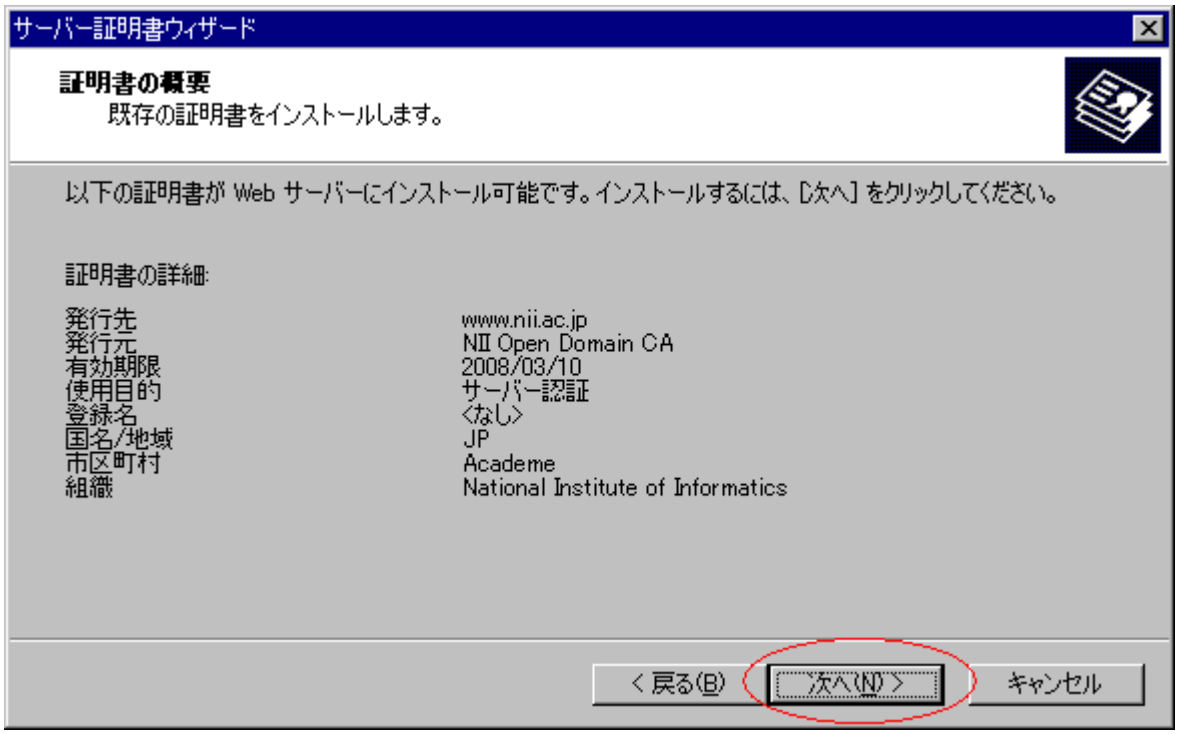


19. 「既存の証明書を使用」を選択し、「次へ」をクリックします。

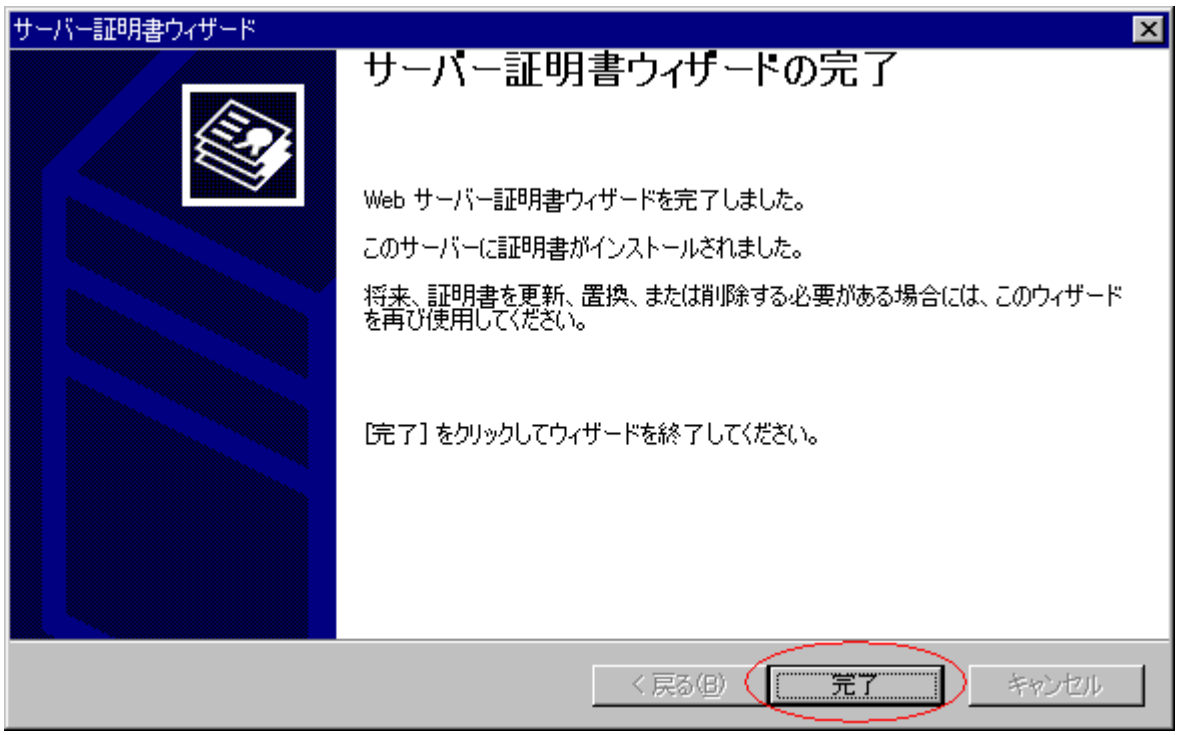


20. SSL/TLS サーバ証明書を選択し、「次へ」をクリックしてください。

21. 証明書情報が表示されます。「次へ」をクリックします。



22. 「完了」クリックします。IIS を再起動してください。



設定したサイト証明書及び秘密鍵はバックアップを取って鍵ペア利用期間中は安全な場所で保管してください。

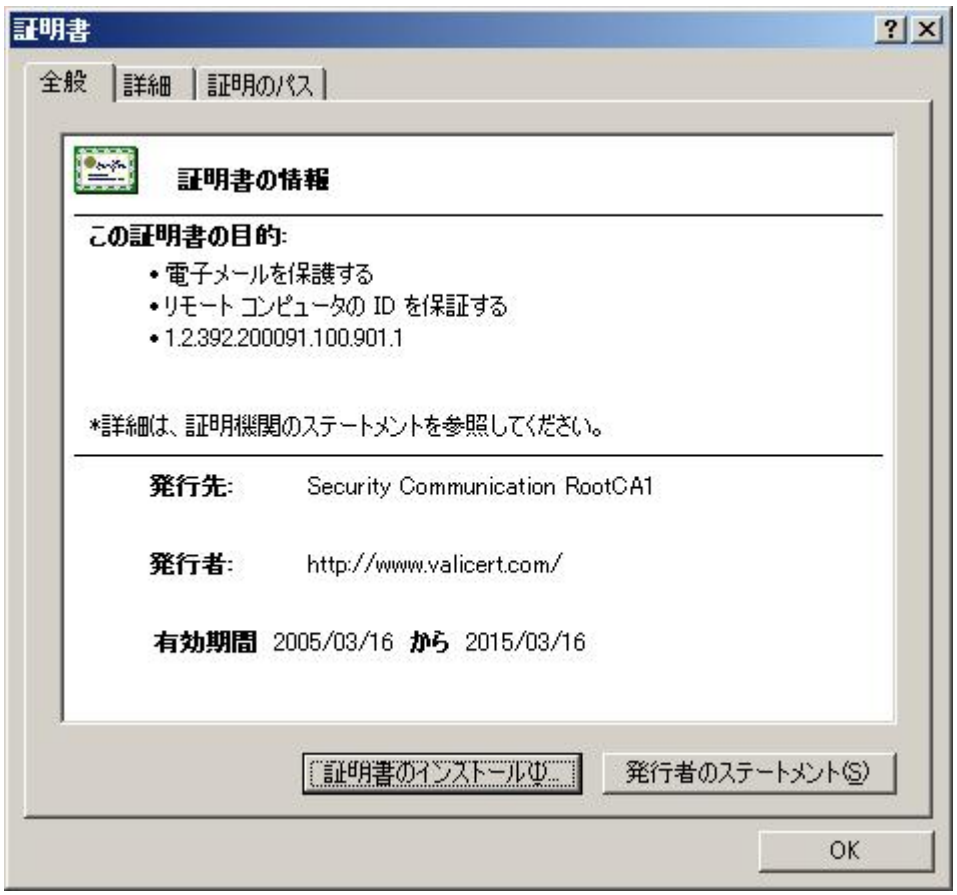
3.6 IIS6

証明書のインストールにあたり、必要な証明書に関しては「1.3 証明書の種類」を参照してください。

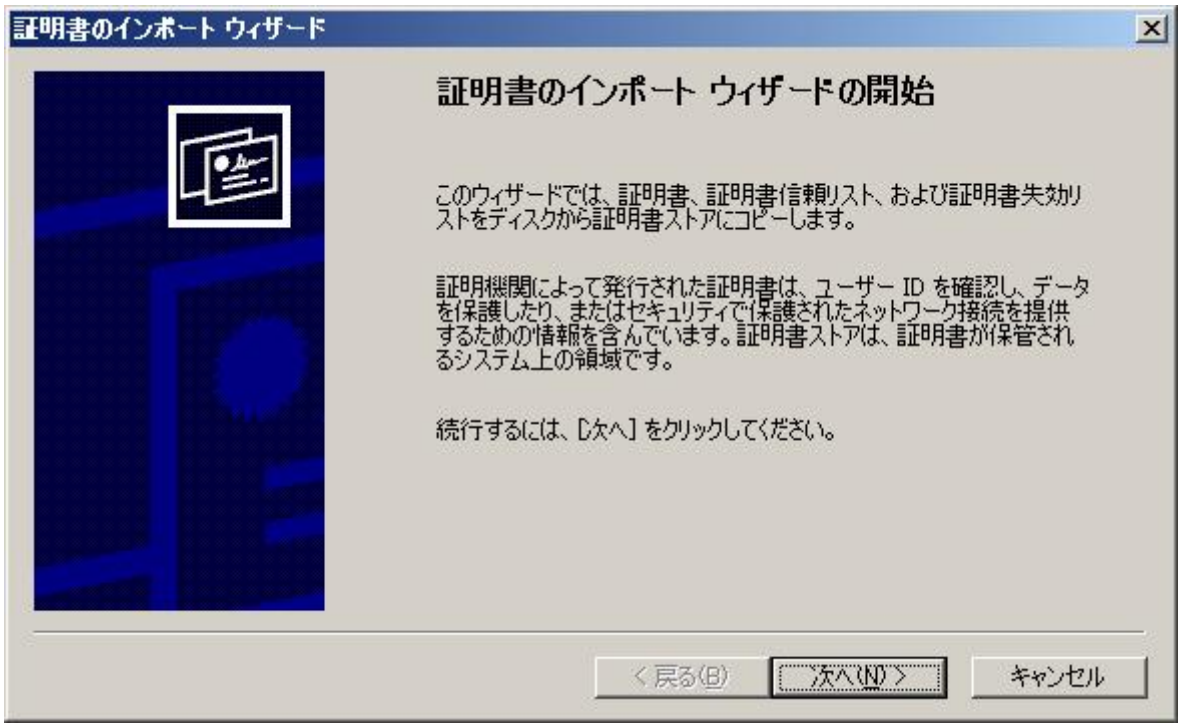
3.6.1 中間 CA 証明書のインストール方法

1. 中間 CA 証明書(SecurityCommunicationRootCA1 認証局証明書:scroot1.crt と NII オープンドメイン認証局証明書：***niica.crt)*** のファイルを開き、以下の作業を繰り返します。

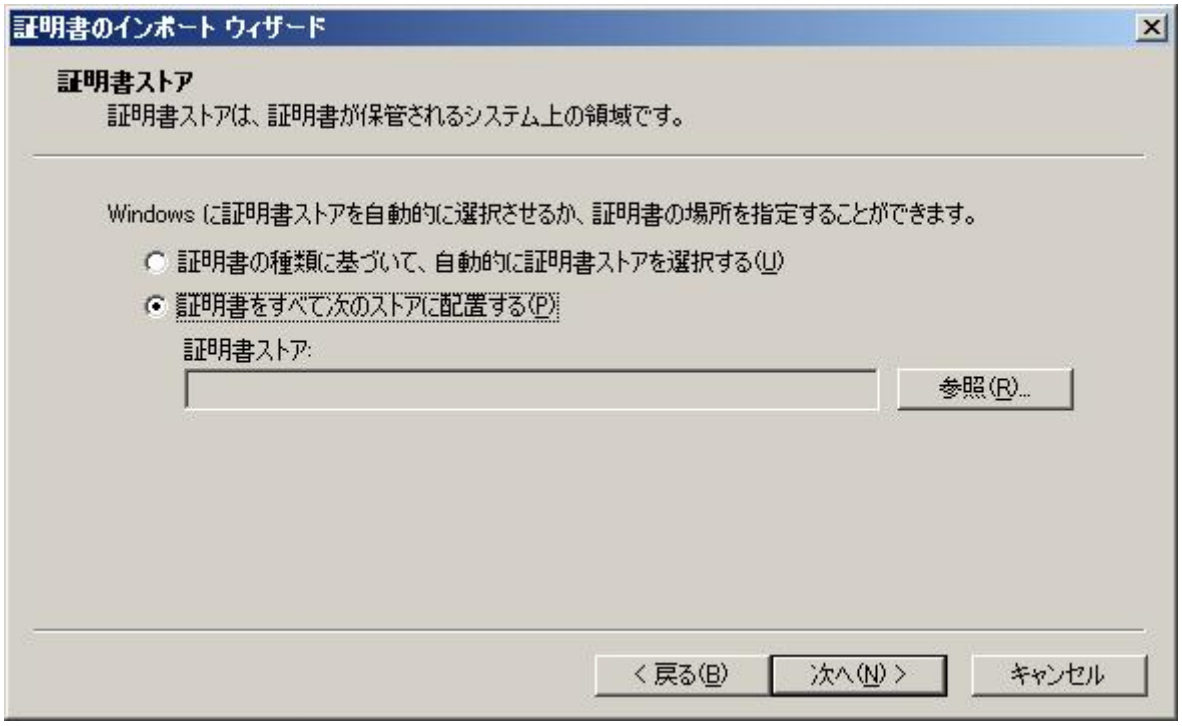
2. 証明書ダイアログボックスが表示されます。「全般」タブの[証明書のインストール]をクリックします。



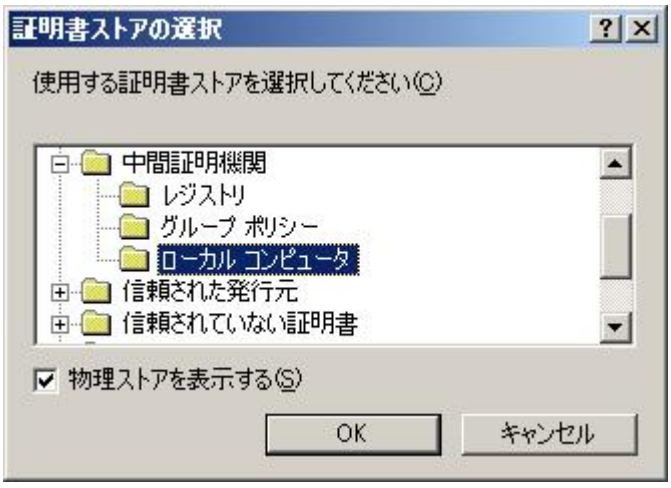
3. 証明書マネージャインポートウィザードが表示されますので、「次へ」をクリックしてください。



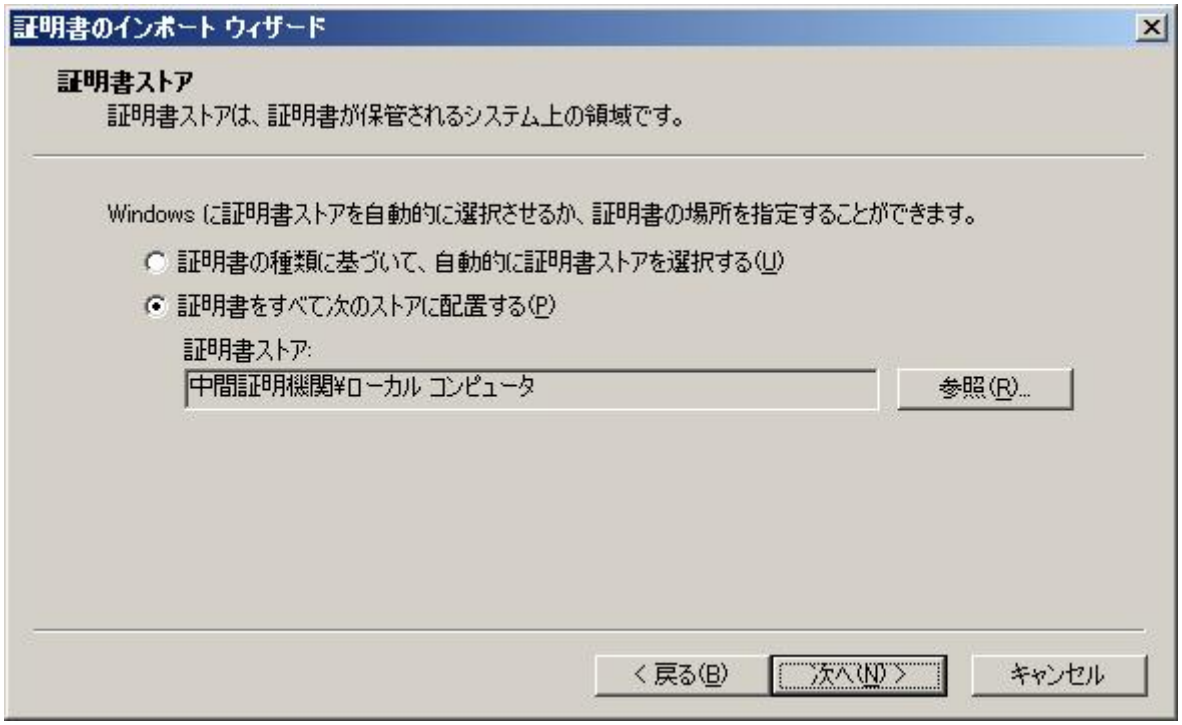
4. ［証明書をすべて次のストアに配置する］を選択し、［参照］をクリックします。



5. 証明書ストア選択ダイアログボックスが表示されますので、［物理ストアを表示する］を選択します。
ダイアログボックス内の項目［中間証明機関］のそばにある［+］マークをクリックして拡張し、［ローカルコンピュータ］を選択し、［OK］をクリックします。

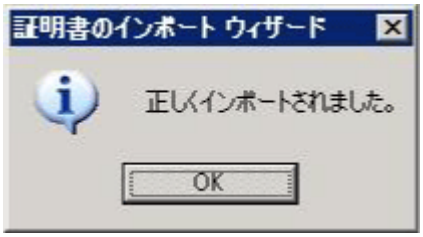


6. 証明書ストアに「中間証明機関¥ローカルコンピュータ」が表示されている事を確認し、「次へ」をクリックしてください。



7. 「証明書のインポートウィザードの完了」が表示されたら、「完了」をクリックしてください。

8. 証明書のインポートウィザードが表示されます。[OK]をクリックします。



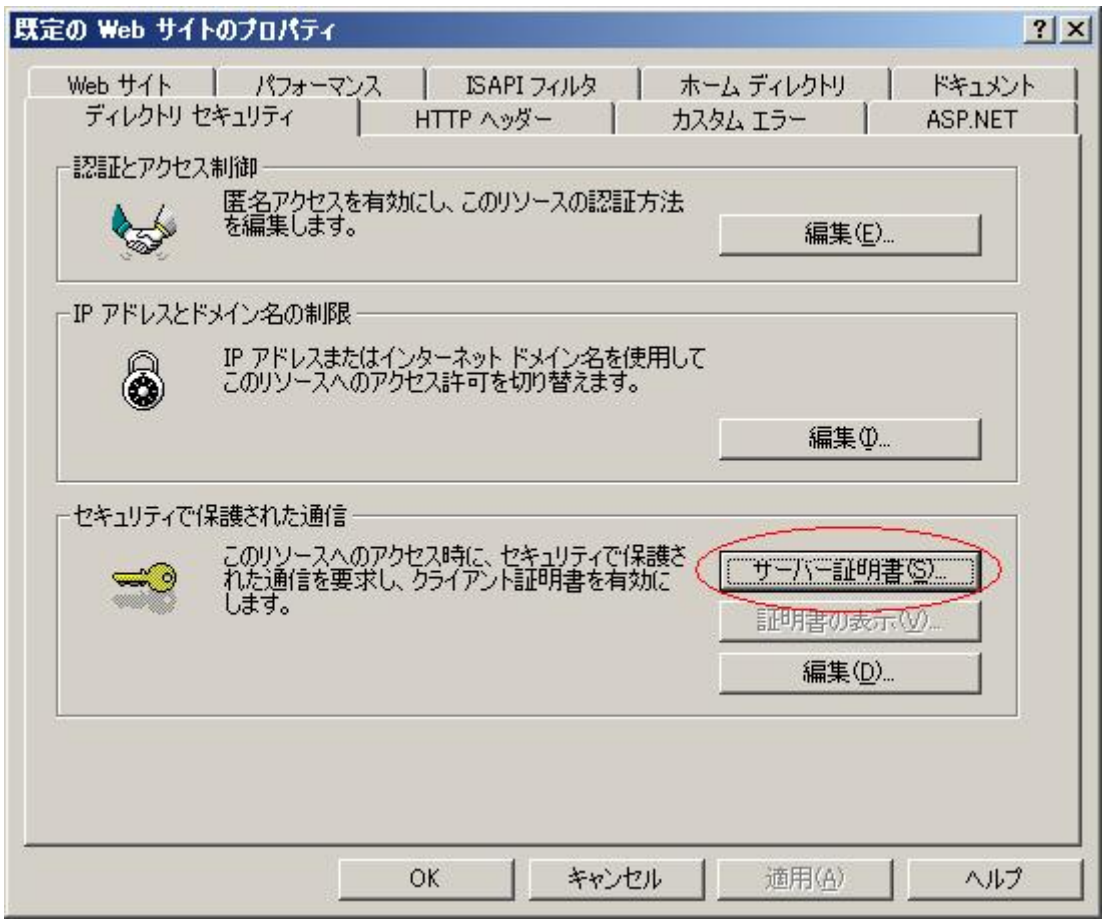
SecurityCommunicationRootCA1 認証局証明書と NII オープンドメイン認証局証明書のインストールが終了したら、サイト証明書（SSL/TLS 証明書）をインストールします。

3.6.2 サイト証明書（SSL/TLS サーバ証明書）のインストール方法

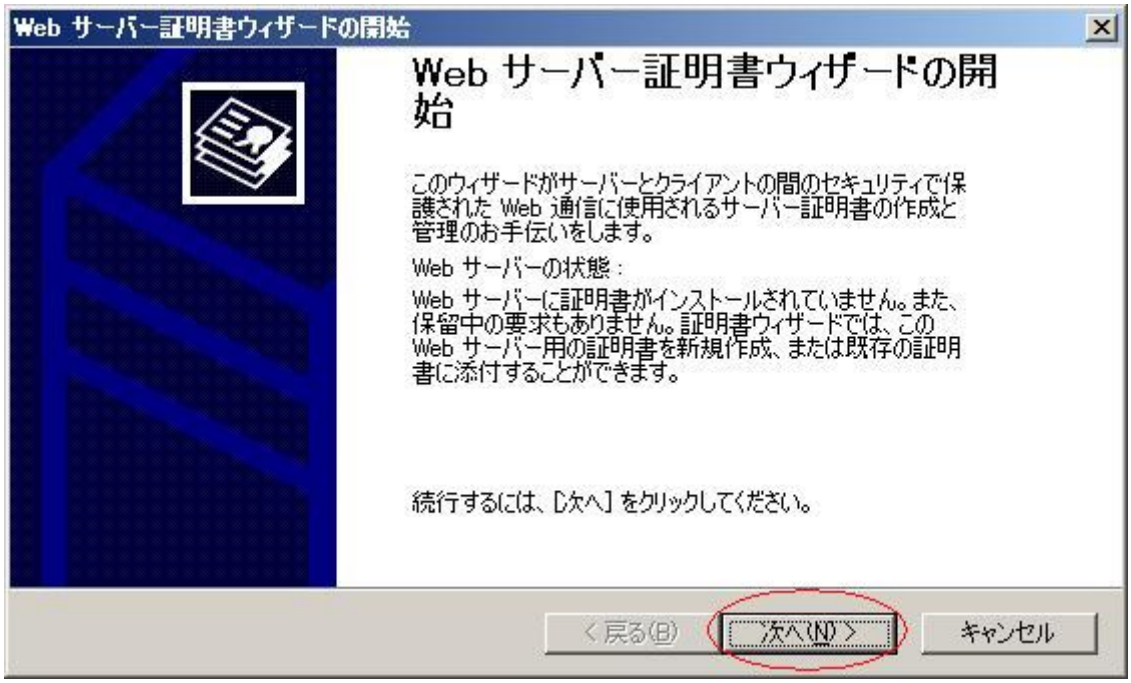
1. 「3.1.2 IIS 利用の場合 openssl による PKCS#12 ファイルの作成」を参照して「*servername.pfx*」ファイルを準備します。

2. 「インターネットインフォメーションサービス（IIS）マネージャ」を起動し、サイトのプロパティを開きます。

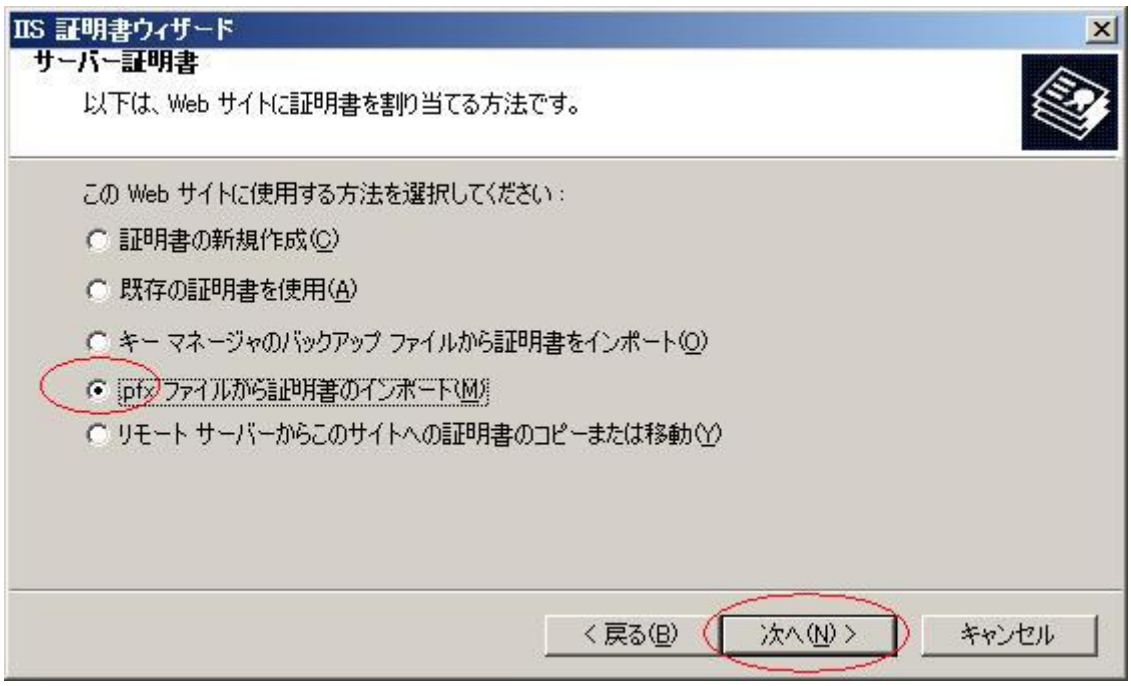
3. 「ディレクトリ セキュリティ」のタグより「セキュリティ保護された通信」の「サーバー証明書」を開きます。



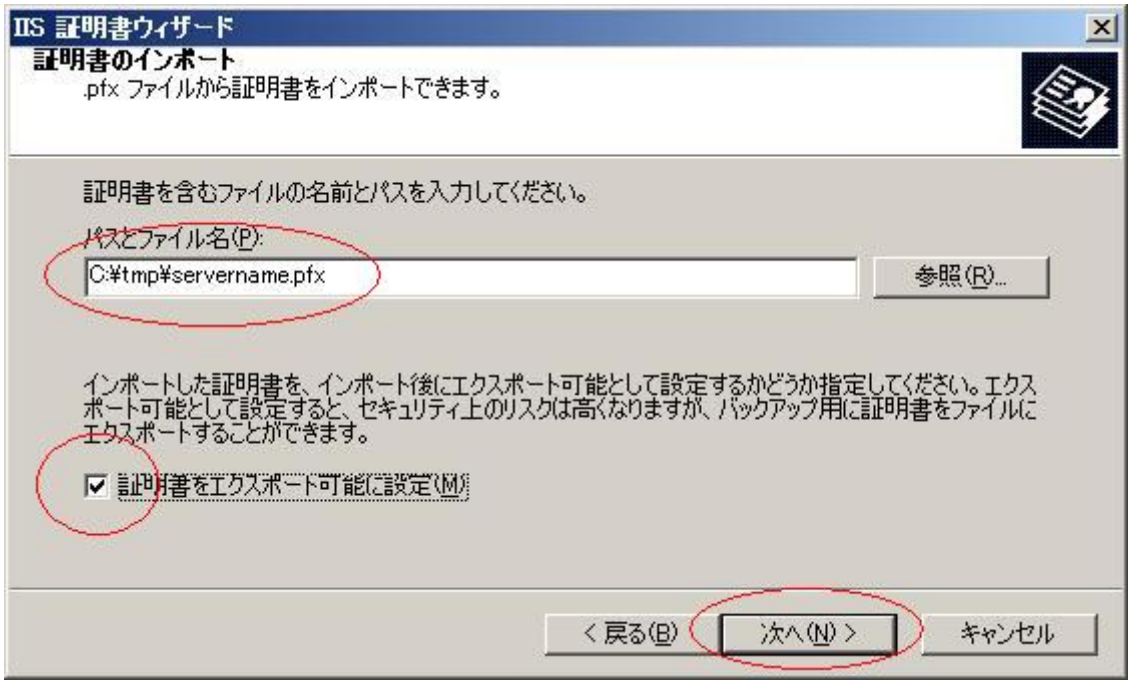
4. 「サーバ証明書ウィザード」が起動します。「次へ」をクリックします。



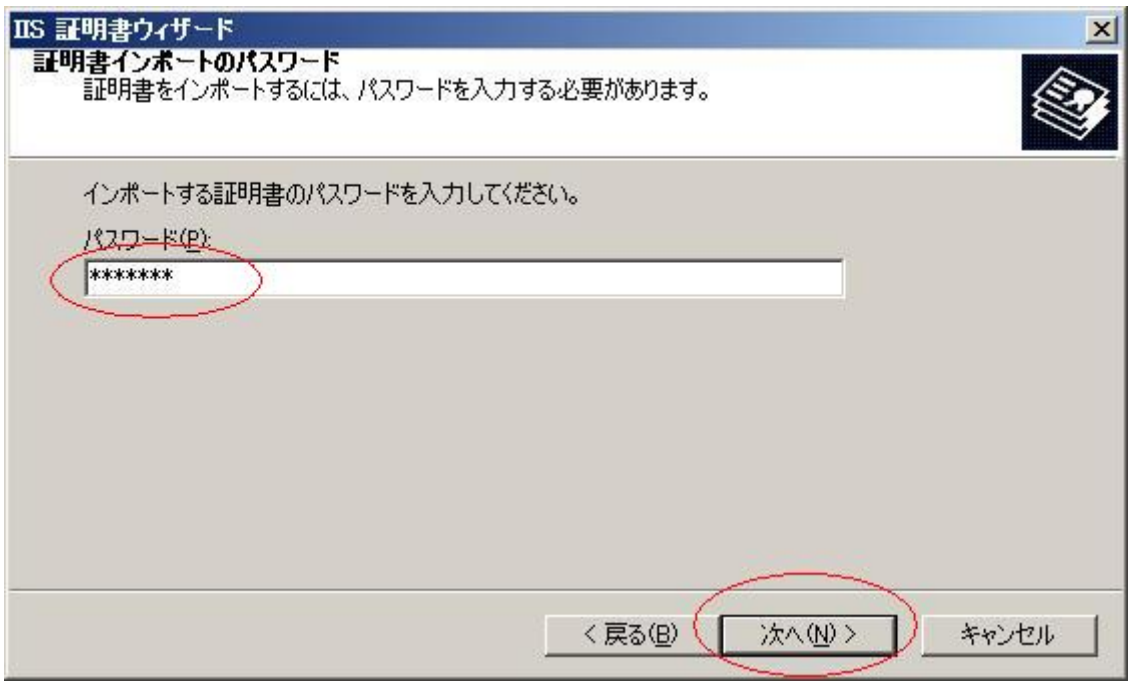
5. 「pfx ファイルから証明書のインポート」をチェックし「次へ」をクリックします。



6. 「参照」クリックし、手順１で準備した「***servername.pfx***」を指定します。「証明書をエクスポート可能に設定」をチェックし「次へ」をクリックします。



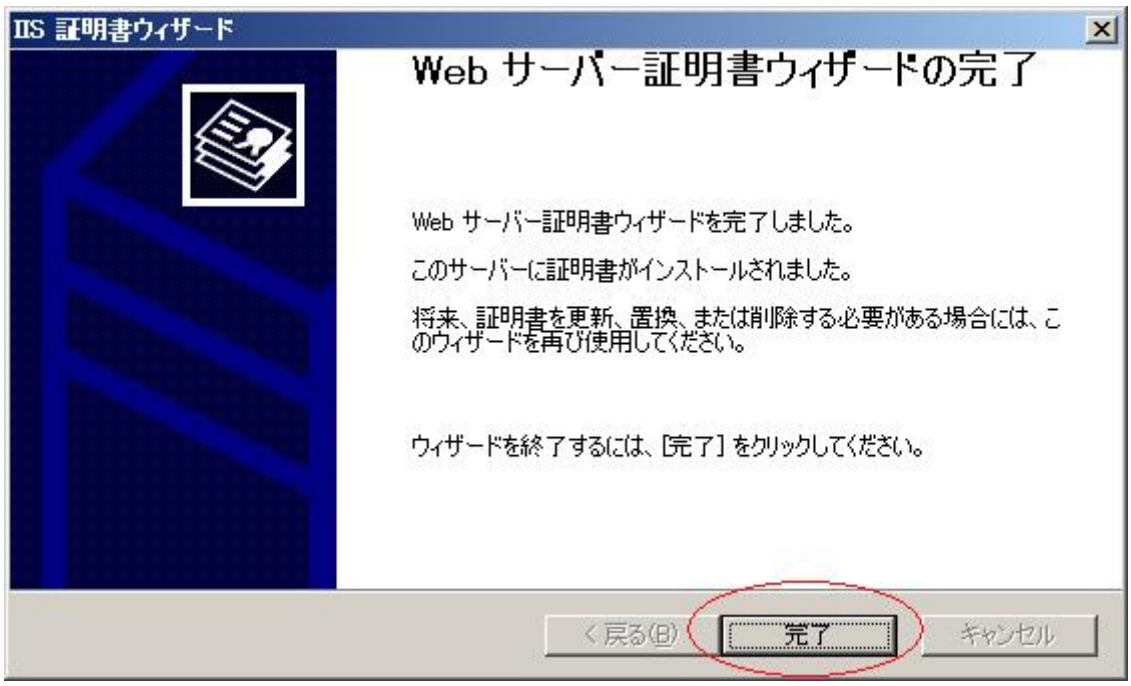
7. PKCS#12 ファイルを作る際に指定した PKCS#12 保護パスフレーズを入力し、「次へ」をクリックします。



8. SSL/TLS サーバの TCP ポート番号を入力し「次へ」をクリックします。

9. 証明書情報が表示されます。「次へ」をクリックします。

10. 「完了」クリックします。IIS を再起動してください。



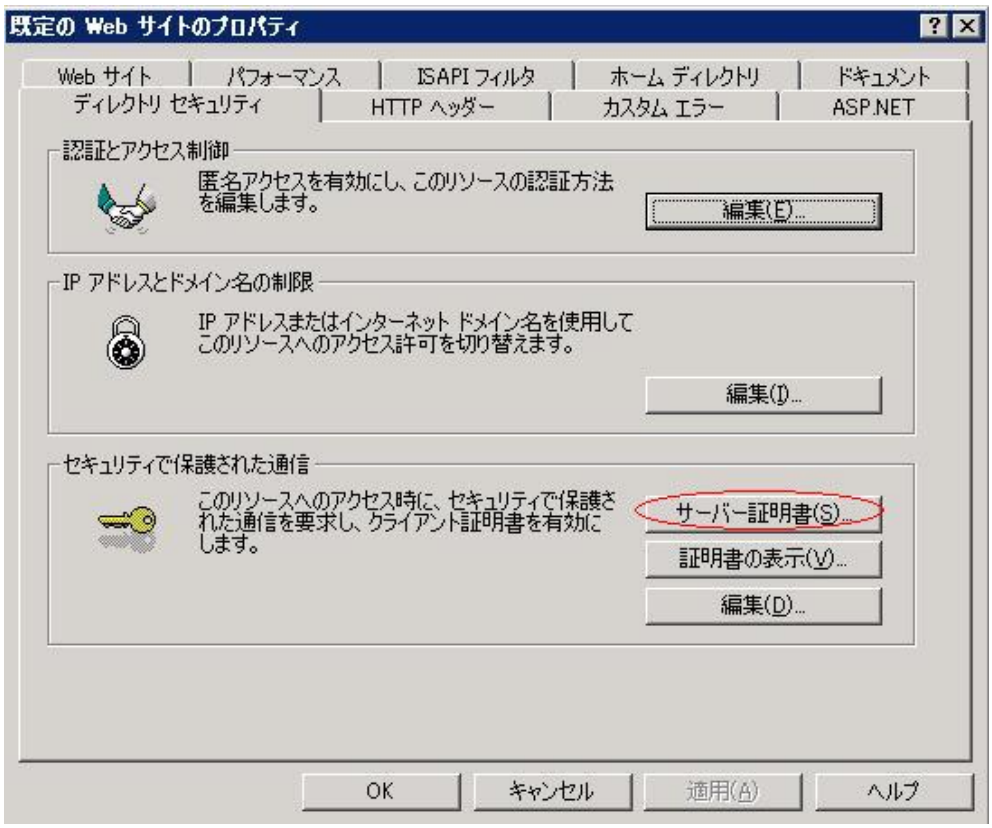
設定したサイト証明書及び秘密鍵はバックアップを取って鍵ペア利用期間中は安全な場所で保管してください。

3.6.3 サイト証明書（SSL/TLS サーバ証明書）の更新時のインストール方法

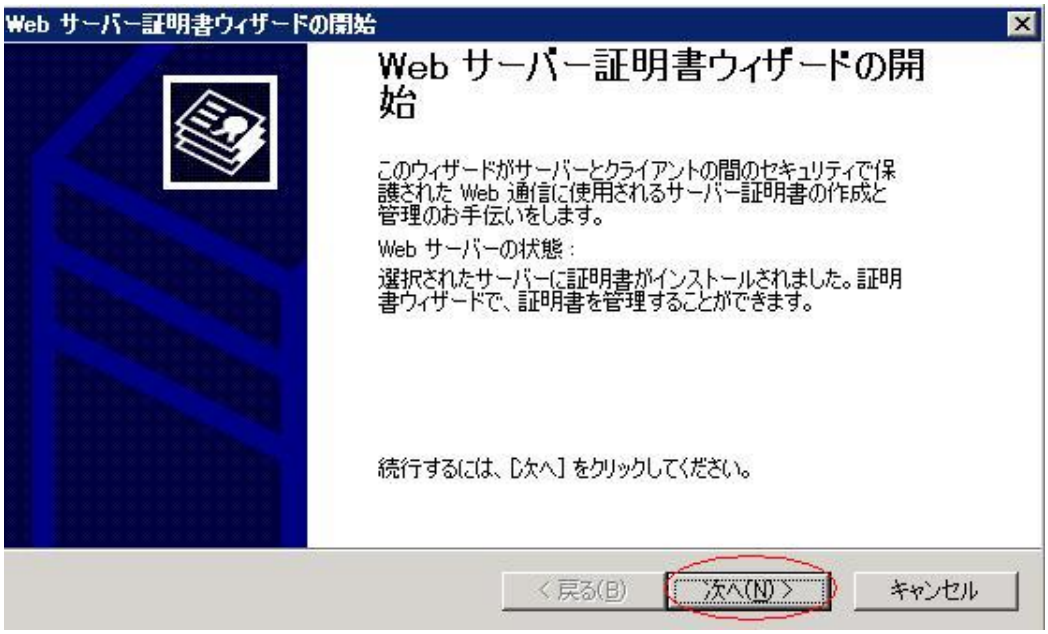
既に対象のサーバに証明書をインストールしている場合は，事前にインストールしている証明書の削除が必要となります。下記に登録された証明書の削除方法を記述します。

1. 「インターネットインフォメーションサービス（IIS）マネージャ」を起動し、サイトのプロパティを開きます。

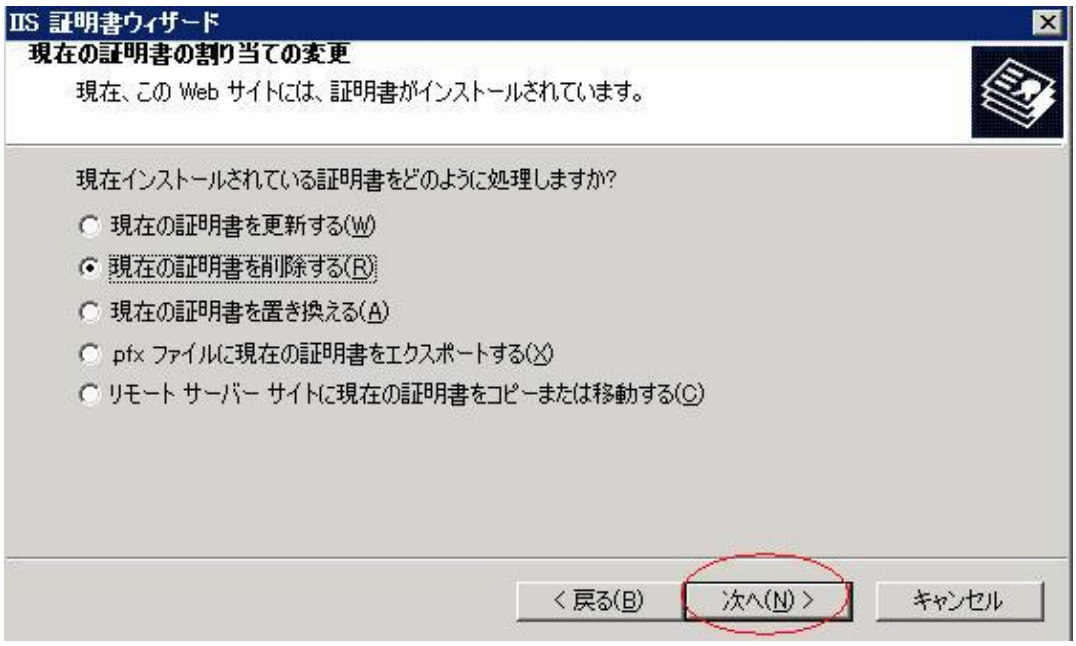
3.1.2



2. 「サーバ証明書ウィザード」が起動します。「次へ」をクリックします。

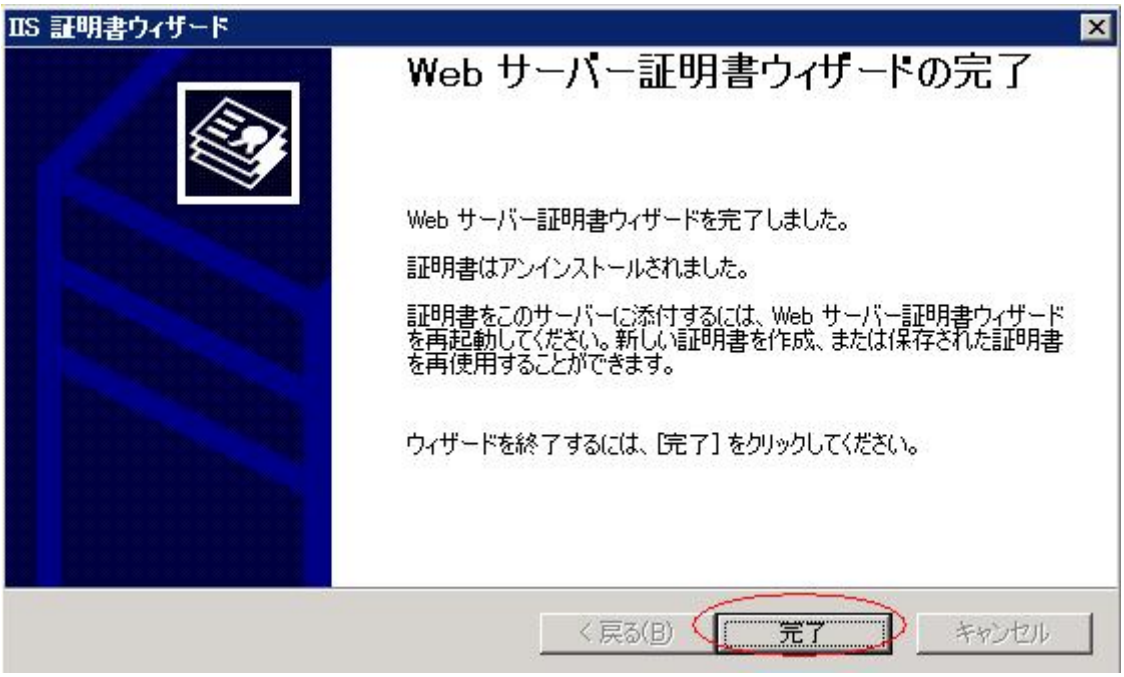


3. 「現在の証明書を削除する」をチェックし「次へ」をクリックします。



4. 現在インストールされている証明書を確認し「次へ」をクリックします。

5. 「完了」をクリックします。



6. 「3.6.2 サイト証明書（SSL/TLS サーバ証明書）のインストール方法」の手順に従い、新しい証明書をインストールする。

DN の禁則文字

番号	禁則文字	文字名称
1	!	exclamation mark
2	″	double quote
3	#	number sign (hash)
4	$	Dollar
5	%	Percent
6	&	ampersand
7	(	open parenthesis
8	)	close parenthesis
9	*	Asterisk
１０	/	oblique stroke
１１	;	SEMICOLON
１２	<	less than
１３	>	greater than
１４	?	Question mark
１５	@	commercial at
１６	[	open square bracket
１７	¥	backslash
１８	]	close square bracket
１９	^	Caret
２０	_	underscore
２１	`	back quote
２２	{	open curly bracket
２３	|	vertical bar
２４	}	close curly bracket
２５	~	Tilde