

# プライベート CA 構築ツールの利用

## 第 6 版

日本医師会総合政策研究機構

平成 19 年 10 月 31 日

# 目次

1はじめに	4
2インストール	5
2.1セキュリティデバイスの環境設定	5
2.1.1使用可能なセキュリティデバイス	6
2.1.2Aladdin eToken PRO 32K の環境設定	6
2.1.3gemalto Cryptoflex 32K の環境設定	6
3起動と画面説明	9
3.1CA証明書の作成画面	9
3.2メインウィンドウ	12
3.2.1メニューバー	12
3.2.2ツールバー	12
3.2.3タブ	13
3.3各タブウィンドウの動作	13
3.3.1証明書タブウィンドウ	13
3.3.2証明書要求タブウィンドウ	14
3.3.3セキュリティデバイスタブウィンドウ	15
3.3.4認証局タブウィンドウ	15
4基本機能	18
4.1証明書の発行	18
4.2証明書のエクスポート	20
4.3CA証明書のエクスポート	22
4.4日レセ用ユーザDBファイルの生成	22
5その他の機能(参考)	24
5.1証明書要求(CSR)からの証明書の発行	24
5.2セキュリティデバイスの使用	25
5.2.1セキュリティデバイスの準備	25
5.2.2証明書のセキュリティデバイスへのエクスポート	26
5.2.3証明書の確認	27
5.2.4PINのブロック	28
5.3プロファイル	29

## 変更履歴

### 第1版

平成18年9月30日  
初版第一版作成

### 第2版

平成18年12月10日  
プライベート CA 構築ツールのセットアップ部分のみとする

### 第3版

平成19年1月26日  
「2.1.3.3」 「閲覧」の見出しレベルを修正  
「3.2」 userdb ファイルの生成方法の追記  
「4」 章見出しの変更  
「4.1 証明書要求を利用して証明書を利用する」「4.2 プロファイルについて」の追加

### 第4版

平成19年2月7日  
「2.1」 apt-line の確認を追記  
「3.2」 userdb ファイルの生成と更新の節を分離  
EE 証明書の呼称を「サーバ証明書」「クライアント証明書」に統一

### 第5版

平成19年5月9日  
「2.1.2.3」 セキュリティデバイスタブについて追記  
「4.2」 セキュリティデバイスの使用方法の追記

### 第6版

平成19年10月31日  
タイトルを「プライベート CA 構築ツールの利用方法」から「プライベート CA 構築ツールの利用」に変更  
「2」 apt-line の設定を修正  
「2.1」 セキュリティデバイスの環境設定を追加  
glserver、glclient | monsiaj の SSL 接続設定手順を除き、プライベート CA 構築ツールの機能説明のみに変更

# 1 はじめに

この文書は、日医標準レセプトソフト(以下日レセ)の GUI インターフェース(glclient, monsiaj)と glserver 間の通信に SSL を利用するために必要なプライベート CA 構築ツールのセットアップとその利用方法について説明します。

SSL を利用する目的として以下の二つを挙げることができます。

- 通信の暗号化
- 公開鍵証明書を利用した認証(SSL クライアント認証)の利用

これらの機能を利用するには、サーバとクライアントでそれぞれに固有の暗号鍵を所有することが前提となります。特に、クライアントでは認証の際に各ユーザを識別するため、ユーザ ID 毎に異なる暗号鍵が必要となります。

暗号鍵は、ユーザが秘密にしなければならない「私有鍵」と、他のサーバやクライアントに公開可能な「公開鍵」を含みます。公開鍵は通信の際にサーバとクライアントの間で交換されますが、公開鍵が本当に信頼できるものであることを確認できるように、認証局(CA)によるデジタル署名を付与した「エンドエンティティ証明書」を事前に発行しておかなければいけません。また、サーバがクライアント証明書受け取ったときに、対応する日レセ上のユーザと紐付けを行うために、「ユーザデータベース」を用意する必要があります。

以降エンドエンティティ証明書のことを単に「証明書」と記述します。また証明書のうち、サーバの鍵に対して発行されたものを「サーバ証明書」、クライアント(ユーザ)に対して発行されたものを「クライアント証明書」と呼びます。

「プライベート CA 構築ツール」は、glserver と glclient, monsiaj が利用する証明書を発行し、ユーザデータベースをメンテナンスするための機能を提供します。

またこの文書では、コンソールコマンドは一般ユーザで操作する前提で記載しています。管理者権限を利用する場合は、すべて sudo コマンドで実行しています。

## 2 インストール

管理用ユーザで GUI 画面にログインします。コンソールを開いて/etc/apt/sources.list ファイルに下記の内容を追記します。

- 使用している機材の Debian バージョンにより追記する内容が異なります。
- 日レセ導入済みの機材では既に同様の記述がある可能性があります。その場合 /etc/apt/sources.list の編集は必要ありません。

### Debian GNU/Linux 4.0 etch の場合

```
deb http://ftp.orca.med.or.jp/pub/debian/ etch jma
deb-src http://ftp.orca.med.or.jp/pub/debian/ etch jma
```

### Debian GNU/Linux 3.1 Sarge の場合

```
deb http://ftp.orca.med.or.jp/pub/debian/ sarge jma
deb-src http://ftp.orca.med.or.jp/pub/debian/ sarge jma
```

/etc/apt/sources.list 編集後、以下のコマンドを実行しプライベート CA 構築ツールをインストールします。

```
$ sudo aptitude update[Enter]
$ sudo aptitude install jma-certtool[Enter]
```

### 2.1 セキュリティデバイスの環境設定

プライベート CA 構築ツールは、セキュリティデバイスに対応しています。

セキュリティデバイスとは証明書、秘密鍵を安全に格納する認証デバイスです。秘密鍵の使用をセキュリティデバイス内部で実行する機能を持ち、秘密鍵を外部に漏らす心配がありません。また証明書の格納も可能なため SSL に必要な情報を 1 つのデバイスに集約することができます。

以下、セキュリティデバイスを使用するための環境設定について説明します。

**セキュリティデバイスを使用しない場合は設定の必要はありません**

### 2.1.1 使用可能なセキュリティデバイス

現在、プライベート CA 構築ツールから使用可能なセキュリティデバイスは以下の二つです。またデバイス使用のためそれぞれ環境設定が必要になります。

#### Aladdin eToken PRO 32K

- USB トークンタイプ
- 製品仕様
  - [http://www.aladdin.co.jp/etoken/etoken\\_line\\_01.html](http://www.aladdin.co.jp/etoken/etoken_line_01.html)

#### gemalto Cryptoflex 32K

- IC カードタイプ
  - カードリーダーとして axalto Reflex USB v.3 を使用
- 製品仕様
  - [http://www.cryptoflex.com/Products/cards\\_32k.html](http://www.cryptoflex.com/Products/cards_32k.html)

### 2.1.2 Aladdin eToken PRO 32K の環境設定

以下のコマンドで、管理ユーザを scard グループに追加します。

```
$ sudo adduser <管理ユーザ名> scard[Enter]
```

設定後、一端ログアウトしてログインし直します。セキュリティデバイスを機材に接続し、以下のコマンドを実行します。「Aladdin eToken PRO」という表示があれば正しく設定されています。

```
$ opensc-tool -l[Enter]
Readers known about:
Nr.   Driver      Name
0     opentct     Aladdin eToken PRO <= この部分を確認
1     opentct     OpenCT reader (detached)
2     opentct     OpenCT reader (detached)
3     opentct     OpenCT reader (detached)
4     opentct     OpenCT reader (detached)
```

### 2.1.3 gemalto Cryptoflex 32K の環境設定

`/usr/lib/pcsc/drivers/ifd-ccid.bundle/Contents/Info.plist` を編集します。

```

...
<key>ifdVendorID</key>
<array>
    <string>0x04E6</string> <= この行を追加
    <string>0x08E6</string>
    <string>0x08E6</string>
...
<key>ifdProductID</key>
<array>
    <string>0x511c</string> <= この行を追加
    <string>0x3437</string>
    <string>0x3438</string>
...
<key>ifdFriendlyName</key>
<array>
    <string>Reflex USB v.3</string> <= この行を追加
    <string>Gemplus GemPC Twin</string>
    <string>Gemplus GemPC Key</string>

```

この設定ファイルでは <array> の順序が意味を持つので、追記する箇所は各キーごとに同じ位置にしなければなりません。そのためそれぞれ <array> の内容の最初に追記しています。

/usr/lib/pcsc/drivers/ifd-ccid.bundle/Contents/Info.plist の編集後、以下のコマンドを実行します。

```
$ sudo /etc/init.d/pcscd restart[Enter]
```

確認のためセキュリティデバイスを機材に接続し、opensc-tool コマンドを実行します。「Reflex USB v.3」という表示があれば正しく設定されています。

```
$ opensc-tool -l[Enter]
Readers known about:
Nr.    Driver    Name
0      opencnt   Reflex USB v.3 <= この部分を確認
1      opencnt   OpenCT reader (detached)
2      opencnt   OpenCT reader (detached)
3      opencnt   OpenCT reader (detached)
4      opencnt   OpenCT reader (detached)
```

### 3 起動と画面説明

コンソールを開いて以下のコマンドを実行し、プライベート CA 構築ツールを起動します。

```
$ jma-certtool [Enter]
```

root ユーザのパスワードが要求されるので入力します (図 1:パスワード入力画面)。

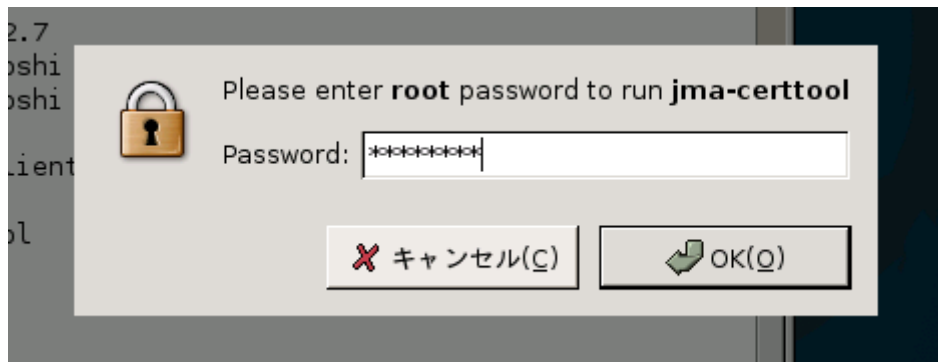


図 1:パスワード入力画面

パスワード入力に成功すると、CA 証明書の作成画面またはメインウィンドウが表示されます。

#### 3.1 CA 証明書の作成画面

初回起動時や CA 証明書が作成されていない場合に表示される画面です。(図 2:CA 証明書作成画面)

この画面で CA 証明書の作成を行います。

The image shows a 'New CA' dialog box with the following fields and options:

- 新しいCAの名前: myca1
- 国名(必須): JP
- 都道府県名: (empty)
- 市町村名: (empty)
- 組織名(必須): test
- 部署名: (empty)
- コモンネーム(必須): test-ca
- Eメールアドレス: (empty)
- シリアル番号: (empty)
- 鍵アルゴリズム:  RSA,  DSA
- 鍵長(ビット数):  512,  1024,  2048,  4096
- ダイジェスト:  MD5,  SHA1,  SHA256
- 証明書の有効期限: 3650 日
- CRLのデフォルト有効期限: 7 日
- CA鍵のパスワード: (empty)
- CA鍵のパスワード(確認): (empty)
- CA証明書の拡張領域の編集: (button)
- キャンセル(C): (button)
- OK(O): (button)

図 2:CA 証明書作成画面

CA 証明書の各設定項目を入力します。

それぞれの項目について以下に簡単に説明します。CA 証明書を作成するために必要な項目は、項目名に(必須)と記載しています。

**新しいCAの名前(必須)**

CAの名称を入力します。

**国名(必須)**

半角アルファベットでJPと入力されています。この項目は変更しないでください。

**都道府県名**

お住いの都道府県名をローマ字で入力します。(例: Tokyo)

**市町村名**

お住いの市町村名をローマ字で入力します。(例: Bunkyo-ku, Mitaka City)

**組織名(必須)**

CAを利用する組織名(医院名など)をローマ字で入力します。(例: Nichi-i Clinic)

**部署名**

組織に複数の部署がある場合は部署名をローマ字で入力します。(例: Keiri)

**コモンネーム(必須)**

CA証明書の名前をローマ字で入力します。(例: Nichi-i Clinic CA)

**E メールアドレス**

CAの管理を行うためのメールアドレスを入力します。(例: admin@example.or.jp)

**シリアル番号**

シリアル番号を半角の数字で入力します。

**鍵アルゴリズム**

鍵のアルゴリズムを指定します。デフォルト値はRSA です。

**鍵長(ビット数)**

鍵長を指定します。デフォルト値は2048 です。

**ダイジェスト**

ダイジェストのアルゴリズムを指定します。デフォルト値はSHA1 です。

**証明書の有効期限**

作成するCAの有効期間を日数で指定します。デフォルト値は3650 日です。

**CRL のデフォルト有効期限**

CAが発行するCRL(失効リスト)の有効期間を日数で指定します。デフォルト値は7 日です。

**CA 鍵のパスワード**

CAの私有鍵にパスワードを設定する場合は値を入力します。数字とアルファベットが利用可能です。設定すると証明書の発行時にパスワードが必要となります。

**CA 鍵のパスワード(確認)**

入力確認のため、「CA 鍵のパスワード」と同じ値を入力します。

**CA 証明書の拡張領域の編集**

通常は特に設定を変更する必要はありません。

必要な項目を入力して「OK」をクリックするとCA証明書が作成され、メインウィンドウが表示されます。

## 3.2 メインウィンドウ

プライベート CA 構築ツールの基本画面です。この画面から全ての操作を行います。

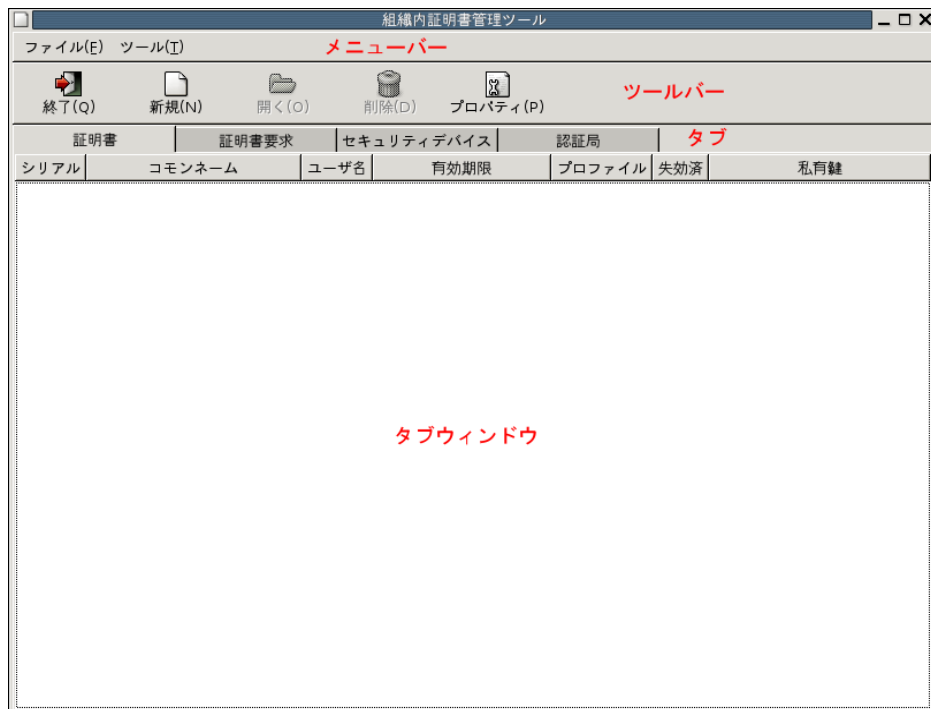


図 3:メインウィンドウ

### 3.2.1 メニューバー

メニューバーで利用可能な項目とそれぞれの用途は以下のとおりです。

#### ファイル

- 新規 CA：新しい CA を構築します
- CA を開く：既存の CA を選択します
- CA の削除：既存の CA を削除します
- 終了：このツールを終了します

#### ツール

- 日レセ用のユーザ DB を作成する：日レセ SSL 認証で利用するためのユーザ DB ファイル userdb を作成します

### 3.2.2 ツールバー

ツールバーに表示されているボタンは、選択されたタブによって動作が異なります。それぞれの用途と動作は以下のとおりです。

#### 終了

このツールを終了します

## 新規

開いているタブによって動作が変わります

- 証明書タブ：新しい証明書の作成を開始します
- 証明書要求タブ：新しい証明書要求の作成を開始します
- 認証局タブ：新しいCAの作成を開始します

## 開く

既存の証明書要求ファイルを取り込む際に利用します

## 削除

選択されているアイテムを削除します

## プロパティ

選択されているアイテムのプロパティを表示します

### 3.2.3 タブ

プライベート CA 構築ツールには4つのタブが用意されており、タブをクリックすることでメインウィンドウが切り替わり、利用できる情報が変わります。個々のタブで扱うことができる情報は以下の通りです。

#### 証明書

CAで発行された証明書の情報を管理します

#### 証明書要求

CAで利用する証明書要求の情報を管理します

#### セキュリティデバイス

CAの機材に接続されているセキュリティデバイスの情報を表示します

#### 認証局

CA自身の情報を管理します

## 3.3 各タブウィンドウの動作

### 3.3.1 証明書タブウィンドウ

プライベート CA 構築ツールを起動した直後はこのタブが開きます。他のタブウィンドウが開いているときに証明書タブをクリックした場合にもアクティブになります。

プライベート CA 構築ツールで作成した証明書の管理全般を行います。

証明書		証明書要求		セキュリティデバイス		認証局	
シリアル	コモンネーム	ユーザ名	有効期限	プロファイル	失効済	私有鍵	
2	www.example.com		2008-04-25T08:40:18Z	orca-server		✓	
3	client1	oruser	2008-04-25T08:42:48Z	orca-client		✓	

図 4: 証明書タブウィンドウの画面

証明書タブウィンドウ（図 4: 証明書タブウィンドウの画面）ではプライベート CA 構築ツールを利用して発行されたすべての証明書について、以下の情報が表示されます。

### シリアル

証明書のシリアル番号が表示されます。発行順に連番が付与されます

### コモンネーム

証明書に入力したコモンネームが表示されます

### ユーザ名

glserver が認証に利用するユーザ名が表示されます。クリックすることで内容の変更が可能です

### 有効期限

証明書の有効期限が西暦で記載されています

### プロファイル

発行に利用した証明書プロファイル名が記載されています。証明書プロファイルの詳細は認証局タブウィンドウで確認できます

### 失効済

証明書が失効されている場合にチェックマークが表示されます

### 私有鍵

証明書に対応する私有鍵が保存されている場合にチェックマークが表示されます

## 3.3.2 証明書要求タブウィンドウ

証明書要求タブをクリックするとアクティブになるウィンドウです。

証明書要求の管理全般と CA 証明書による署名を行います。

証明書	証明書要求	セキュリティデバイス	認証局	
コモンネーム	鍵アルゴリズム	要求発行日	署名済	私有鍵
www.example2.com	RSA/2048	2007-04-26T17:44:58+09:00		✓

図 5: 証明書要求タブウィンドウの画面

証明書要求タブウィンドウで（図 5: 証明書要求タブウィンドウの画面）は、新規作成あるいはインポートされたすべての証明書要求について、以下の情報が表示されます。

### コモンネーム

コモンネームが表示されます

### 鍵アルゴリズム

鍵アルゴリズムが表示されます

### 要求発行日

証明書要求がインポートされた日時、あるいは証明書要求タブウィンドウ上で作成された日時が表示されます

### 署名の有無

CA 証明書によって署名されている場合（証明書が発行されている場合）はチェックマークが表示されます

### 私有鍵

証明書要求の対になる私有鍵が存在する場合はチェックマークが表示されます

## 3.3.3 セキュリティデバイスタブウィンドウ

セキュリティデバイスタブをクリックするとアクティブになるウィンドウです。

デバイスの設定や状態を管理します。



図 6:セキュリティデバイスタブウィンドウの画面

以下のボタンが用意されています。

#### リフレッシュ

「利用できるデバイス」ウィンドウの表示を更新します

#### デバイスの初期化

「利用できるデバイス」ウィンドウで指定したデバイスを初期化します

#### 設定

デバイスの共通設定を行う画面が表示されます

## 3.3.4 認証局タブウィンドウ

認証局タブをクリックするとアクティブになるウィンドウです。

CA 証明書や CRL（失効リスト）、証明書タブから発行する証明書のプロファイルなどを管理します。

証明書	証明書要求	セキュリティデバイス	認証局
CA証明書の情報			
CA証明書のエクスポート(PEM)	サブジェクト	/C=JP/O=test/CN=test-ca	
CA証明書のエクスポート(DER)	有効期限開始日時	2007-05-29T08:27:26Z	
CRLのエクスポート(PEM)	有効期限終了日時	2017-05-26T08:27:26Z	
CRLのエクスポート(DER)	フィンガープリント(MD5)	54:c4:3e:50:57:46:bc:76:19:90:18:a4:4a:a	
閲覧	フィンガープリント(SHA1)	40:30:dc:64:45:9b:be:47:fa:a9:72:39:e8:1	
CA証明書を表示する	証明書プロファイル		
最新のCRLを表示する	名前	サマリー	
認証局の管理	orca-client	日医標準レセプトソフトクライアント証明書	
新しいCRLを発行する	orca-server	日医標準レセプトソフトサーバ証明書	
CAのパスワードを変更する			
CA鍵のCSRを作成する			
CA証明書をインポートする			

図 7: 認証局タブウィンドウの画面

以下のボタンが用意されています。

#### エクスポート

- CA 証明書のエクスポート(PEM) : PEM 形式の CA 証明書をエクスポートします
- CA 証明書のエクスポート(DER) : DER 形式の CA 証明書をエクスポートします
- CRL のエクスポート(PEM) : PEM 形式の CRL をエクスポートします
- CRL のエクスポート(DER) : DER 形式の CRL をエクスポートします

#### 閲覧

- CA 証明書を表示する : CA 証明書の詳細情報が表示されます
- 最新の CRL を表示する : 最新の CRL の詳細情報が表示されます

#### 認証局の管理

- 新しい CRL を発行する : CRL を発行します
- CA のパスワードを変更する : 現在のパスワードから新しいパスワードに変更できます
- CA 鍵の CSR を作成する : 現在利用している CA 鍵を利用して新しい証明書要求 (CSR) を作成し、エクスポートします
- CA 証明書をインポートする : 「CA 鍵の CSR を作成する」で作成した証明書要求から生成された新たな CA 証明書をインポートします

#### CA 証明書の情報

現在の CA 証明書のサブジェクト、有効期間とフィンガープリントを表示します

## 証明書プロファイル

CAが署名に利用できる証明書プロファイルのリストを表示します。「名前」欄がCAによる署名時に選択できる名称です

## 4 基本機能

### 4.1 証明書の発行

glserver や glclient、monsiaj で利用する証明書を発行します。

「証明書タブウィンドウ」を開き、「新規」ボタンをクリックします（図 8:証明書新規発行ボタン）。

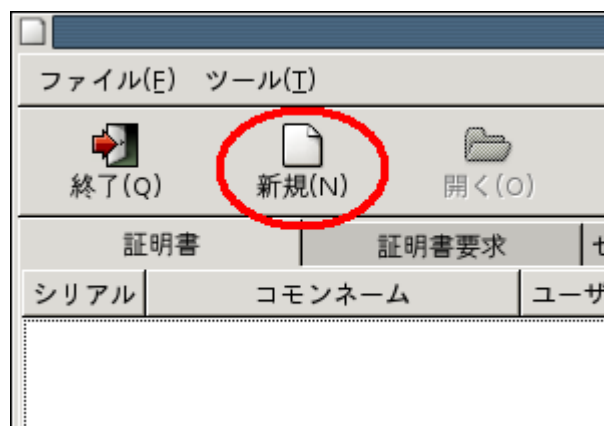


図 8:証明書新規発行ボタン

「証明書要求の編集」画面（図 9:証明書要求の作成）が表示されたら、コモンネームに適切な名前を入力します。セキュリティデバイスを利用する場合を除き、既に入力されている部分は変更する必要はありません。

図 9: 証明書要求の作成

#### サーバ証明書のコモンネームの設定

glserver が稼働する機材の FQDN あるいは IP アドレスを指定します。ここで設定する値は glclient、あるいは monsiaj が接続する際に指定するサーバ名、あるいは IP アドレスと一致している必要があります。

**glserver に設置する証明書のコモンネームが正しく設定されていない場合、glclient が接続できないため注意してください。**

#### クライアント証明書のコモンネームの設定

適切なユーザ ID を指定します。ここで指定するユーザ ID は証明書を識別する時に利用されますが、glserver に接続する際のユーザ名と異なっても構いません。

#### セキュリティデバイス用証明書を発行する場合

セキュリティデバイスは現在クライアント証明書、また glclient のみに対応しています。また、セキュリティデバイスとして Aladdin eToken を利用する場合、「**鍵長 (ビット数)**」を 1024 に変更してください。

「CA による署名」ウィンドウ (図 10: CA による署名画面) が表示されたら、以下の情報を入力して「OK」をクリックします。

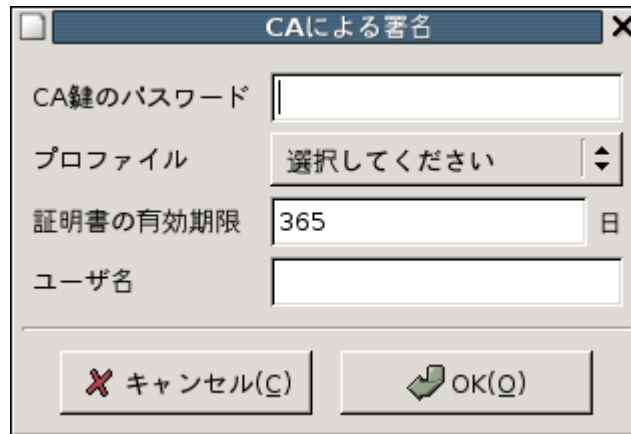


図 10:CAによる署名画面

#### CA 鍵のパスワード

「CA 証明書の作成」画面で設定したパスワードを入力します。

#### プロファイル

サーバ証明書はプルダウンメニューから「orca-server」を選択します。

クライアント証明書はプルダウンメニューから「orca-client」を選択します。

#### 証明書の有効期限

作成する証明書の有効期限を設定します。デフォルト値は365日です。証明書の有効期限のデフォルト値は認証局タブウィンドウから各証明書のプロパティを選択して確認・修正が可能です。

#### ユーザ名

日レセ用ユーザDBファイルの生成時のユーザ名として使用します。

クライアント証明書では、必ず glserver に接続するユーザ名を指定します。

また他の証明書が使用しているユーザ名は指定しないでください。

サーバ証明書では指定しません。

証明書タブウィンドウにエントリが追加されれば証明書の発行は成功です。

## 4.2 証明書のエクスポート

エクスポートしたい証明書のエントリの上で画面を右クリックします。

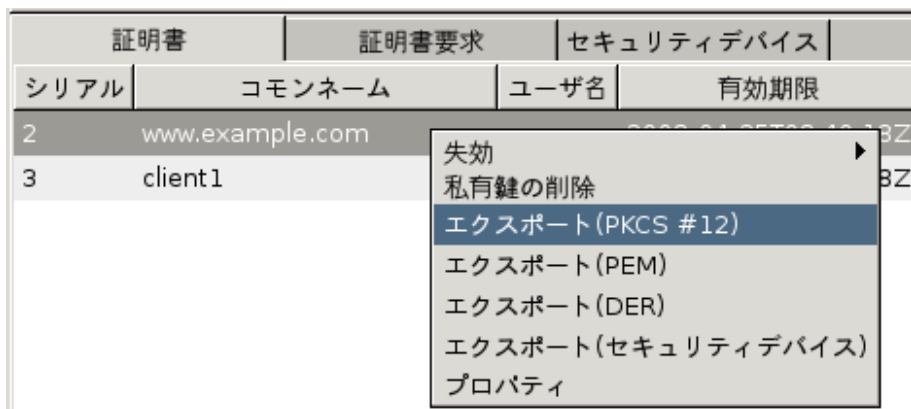


図 11:証明書の操作ポップアップウィンドウ

日レセで利用するサーバおよびクライアント証明書は、私有鍵と証明書がひとつのファイルに含まれる PKCS#12 形式を利用します。

ポップアップウィンドウ（図 11:証明書の操作ポップアップウィンドウ）の「エクスポート(PKCS#12)」を選択するとパスワード入力画面(図 12:PKCS#12 パスワード入力画面)が表示されます。サーバ証明書、クライアント証明書でそれぞれ以下の操作を実施します。

#### サーバ証明書のパスワード

パスワード欄は空のまま「OK」をクリックします

#### クライアント証明書のパスワード

パスワード欄に適切なパスワードを入力します。mosiajでは、空のパスワードで作成された証明書を使用することができないためパスワードの設定が必要です。

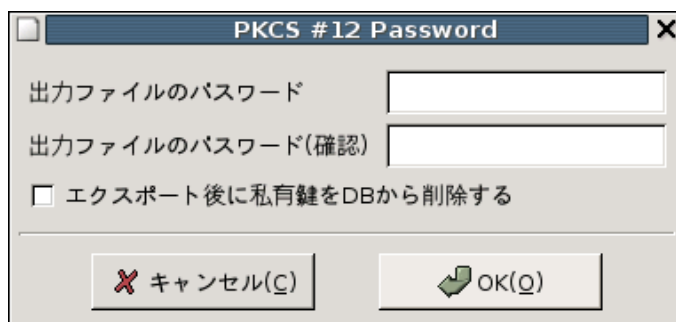


図 12:PKCS#12パスワード入力画面

パスワード欄を空とした場合、「空のパスワードを設定しますか?」と警告画面が表示されますので、「OK」をクリックします。（図 13:空のパスワード警告画面）

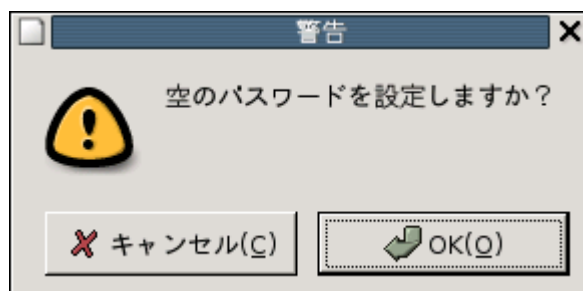


図 13:空のパスワード警告画面

「出力ファイル保存」画面（図 14:ファイルの保存画面）が表示されるので保存先を決定

し「OK」ボタンを押します。デフォルトでは保存されるファイル名は<コモンネーム>.p12 となります。

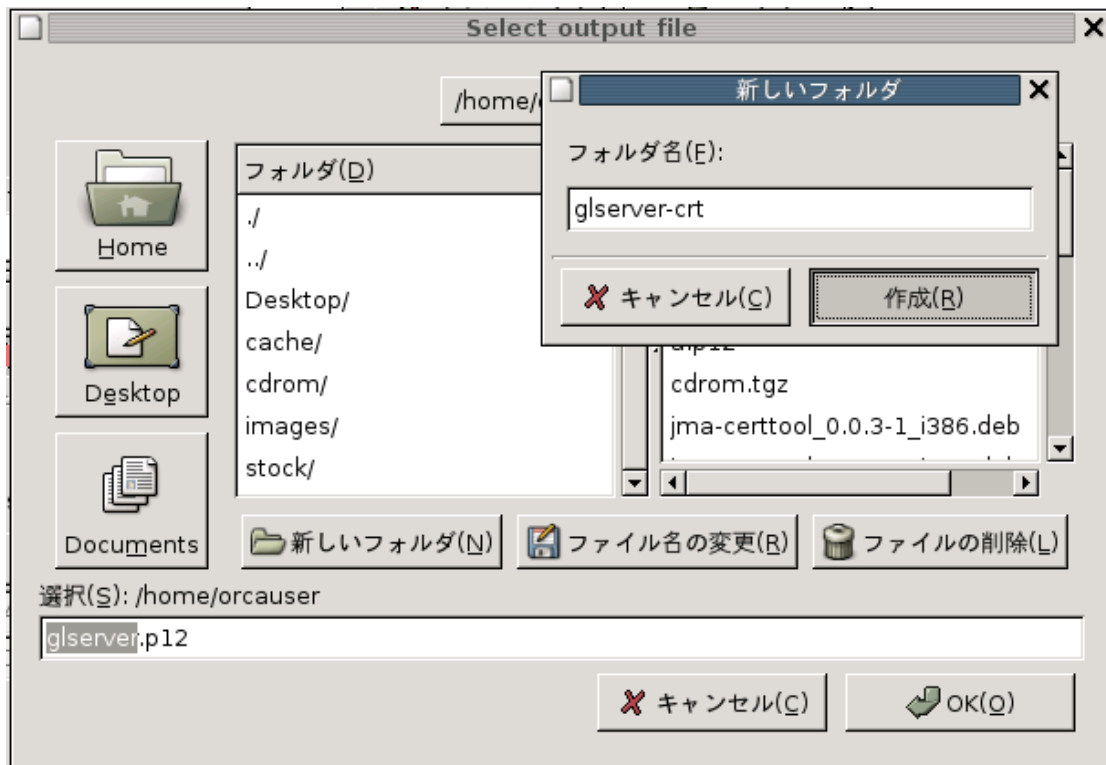


図 14: ファイルの保存画面

### 4.3 CA 証明書のエクスポート

「認証局」タブをクリックして「認証局タブウィンドウ」に移動します。

「CA 証明書のエクスポート(Pem)」ボタンをクリックすると「出力ファイル保存」画面が表示されるので保存先を決定し「OK」ボタンを押します。デフォルトでは保存されるファイル名は<CAのコモンネーム>.pem となります。

### 4.4 日レセ用ユーザDBファイルの生成

証明書タブの証明書エントリのうち、以下の条件を満たすものを認証可能な証明書とするユーザDBファイルを生成します。

- 「ユーザ名」が設定されている
- 有効期限が切れていない

プライベート CA 構築ツールの「ツール」メニューを開き、「日レセ用のユーザDBを生成する」を選択します（図 15: 日レセ用ユーザDBの作成）。

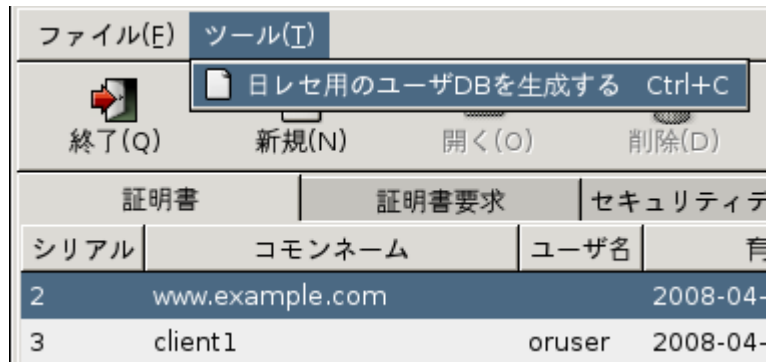


図 15: 日レセ用ユーザDBの作成

「ファイルの保存」画面が表示されるので、適当な場所(例えばホームディレクトリ等)に保存します。保存したユーザDB ファイルを日レセサーバに設定します。

ユーザの追加、削除、変更があった場合は、先に対象の証明書の「ユーザ名」を変更し、再度ユーザDB ファイルの生成を実行します。証明書の「ユーザ名」の変更は、証明書タブを開いて対象の証明書エントリの「ユーザ名」欄をクリックして編集することで行います。

- ユーザの証明書の有効期限が切れるとユーザDB ファイルに有効なユーザとして登録されている場合でも、日レセに接続できなくなります
- 同じユーザ名を複数の証明書に設定した場合、証明書のどれか一つだけが接続可能となり、その他の証明書では接続できなくなります

## 5 その他の機能(参考)

### 5.1 証明書要求 (CSR) からの証明書の発行

証明書を作成するにあたり、自分で私有鍵および証明書要求を作成し、署名のみをプライベート CA 構築ツールで行いたい場合があります。そのような場合は「証明書要求タブウィンドウ」から証明書要求をインポートして証明書を発行します。

プライベート CA 構築ツールを起動して「証明書要求タブウィンドウ」に移動し、「開く」ボタンをクリックします (図 16: 証明書要求のインポート)。

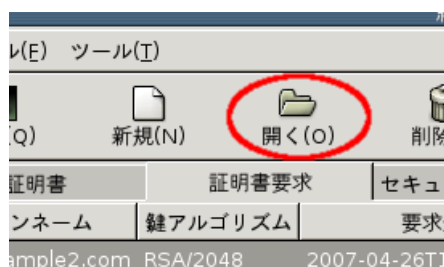


図 16: 証明書要求のインポート

「入力ファイルの選択」画面が表示されたら、証明書要求を選択して「OK」をクリックします。証明書要求が正しい形式であれば「証明書要求タブウィンドウ」に新しいエントリが追加されます。

証明書要求	セキュリティデバイス	認証局
名前	鍵アルゴリズム	要求発行日
sample2.com	RSA/2048	2007-04-26T17:44:58+09:00
署名済		
署名 (証明書発行)		
プロパティ		
削除		

図 17: 証明書要求への署名

証明書要求に追加されたエントリ上で右クリックして表示されるウィンドウで「署名 (証明書発行)」を選択します (図 17: 証明書要求への署名)。

「CA による署名」ウィンドウが表示されたら、以下の情報を入力して「OK」をクリックします。

#### CA 鍵のパスワード

「CA 証明書の作成」画面で設定したパスワードを入力します

## プロフィール

プルダウンメニューから利用するプロフィールを選択します

## 証明書の有効期限

作成する証明書の有効期限を設定します。デフォルト値は 365 日です

## ユーザ名

クライアント証明書を作成する場合は glserver に接続するユーザ名を指定します

証明書要求のエントリ上で「署名済」にチェックが入ったら（図 18:署名されていることを確認する方法）署名は完了です。

証明書要求	セキュリティデバイス	認証局
鍵アルゴリズム	要求発行日	署名済
n RSA/2048	2007-04-26T17:44:58+09:00	✓

図 18: 署名されていることを確認する方法

証明書をエクスポートする場合は、「証明書タブウィンドウ」に移動して、対応する証明書のエントリ上で右クリックし、「エクスポート (PEM)」あるいは「エクスポート (DER)」を選択します。

## 5.2 セキュリティデバイスの使用

この作業は「4.1 証明書の発行」の作業が完了した後に行います。

任意の第三者によるリモートからの glserver への接続を防止するために、クライアント証明書をセキュリティデバイスに格納し、セキュリティデバイス接続時のみ glclient 機から接続させるような運用を行いたい場合があります。

### 5.2.1 セキュリティデバイスの準備

セキュリティデバイスを、プライベート CA 構築ツールが稼動している機材に接続します。

プライベート CA 構築ツールを起動して「セキュリティデバイスタブウィンドウ」に移動します。

セキュリティデバイスが正しく認識されている場合「利用できるデバイス」ウィンドウにエントリが追加されます。初めて利用する場合など「初期化」にチェックが入っていなければ、エントリを選択して「デバイスの初期化」ボタンをクリック、またはエントリを右クリックして表示されるメニューから「デバイスの初期化」を選択してデバイスの初期化を行います（図 19:デバイスの初期化）。



図 19:デバイスの初期化

セキュリティデバイスのPINを設定する画面が表示されますので、任意の4~8文字の文字列を入力します（図 20:PINの設定）。

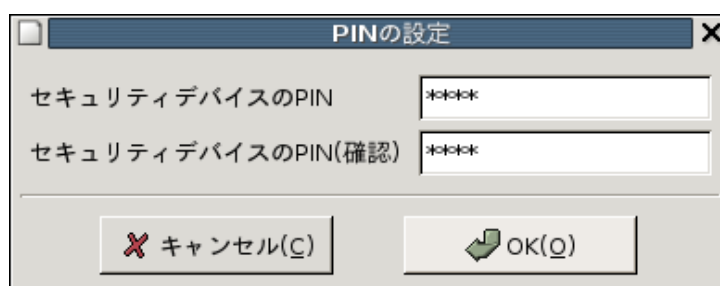


図 20:PINの設定

初期化に成功すると初期化完了ダイアログ(図 21:初期化完了ダイアログ)が表示されます。

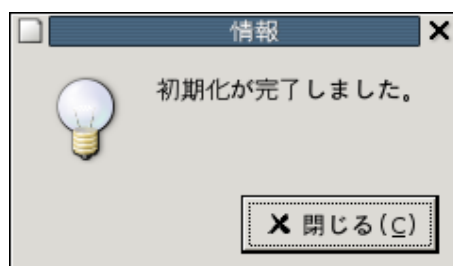


図 21:初期化完了ダイアログ

デバイスに入っている証明書を消去する場合も同様に「デバイスの初期化」を行ってください。

## 5.2.2 証明書のセキュリティデバイスへのエクスポート

「証明書タブウィンドウ」に移動し、セキュリティデバイスに格納したいクライアント証明書のエントリの上で右クリックします。（図 22:セキュリティデバイスへの証明書インポート）



図 22:セキュリティデバイスへの証明書インポート

ポップアップウィンドウの「エクスポート (セキュリティデバイス)」を選択するとセキュリティデバイスのPINを入力する画面が表示されます。セキュリティデバイスの初期化時に設定したPINを入力して証明書をエクスポートします。

エクスポートに成功するとエクスポート成功ダイアログが表示されます(図 23:エクスポート成功ダイアログ)。

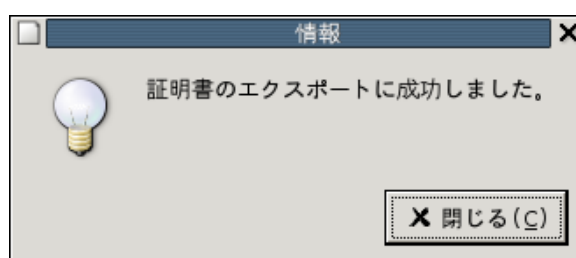


図 23:エクスポート成功ダイアログ

※複数のセキュリティデバイスを接続している場合は、セキュリティデバイスのPIN入力画面が表示される前にどのデバイスを利用するか確認する画面が表示されます。

### 5.2.3 証明書の確認

「セキュリティデバイスタブウィンドウ」に移動します。証明書をインポートしたセキュリティデバイスのエントリの上で右クリックして表示されるメニューから「プロパティ」を選択します(図 24:デバイスタブの右クリックメニュー)。



図 24: デバイスタブの右クリックメニュー

セキュリティデバイスにエクスポートされた証明書の詳細情報が表示されますので、内容を確認します（図 1: パスワード入力画面）。

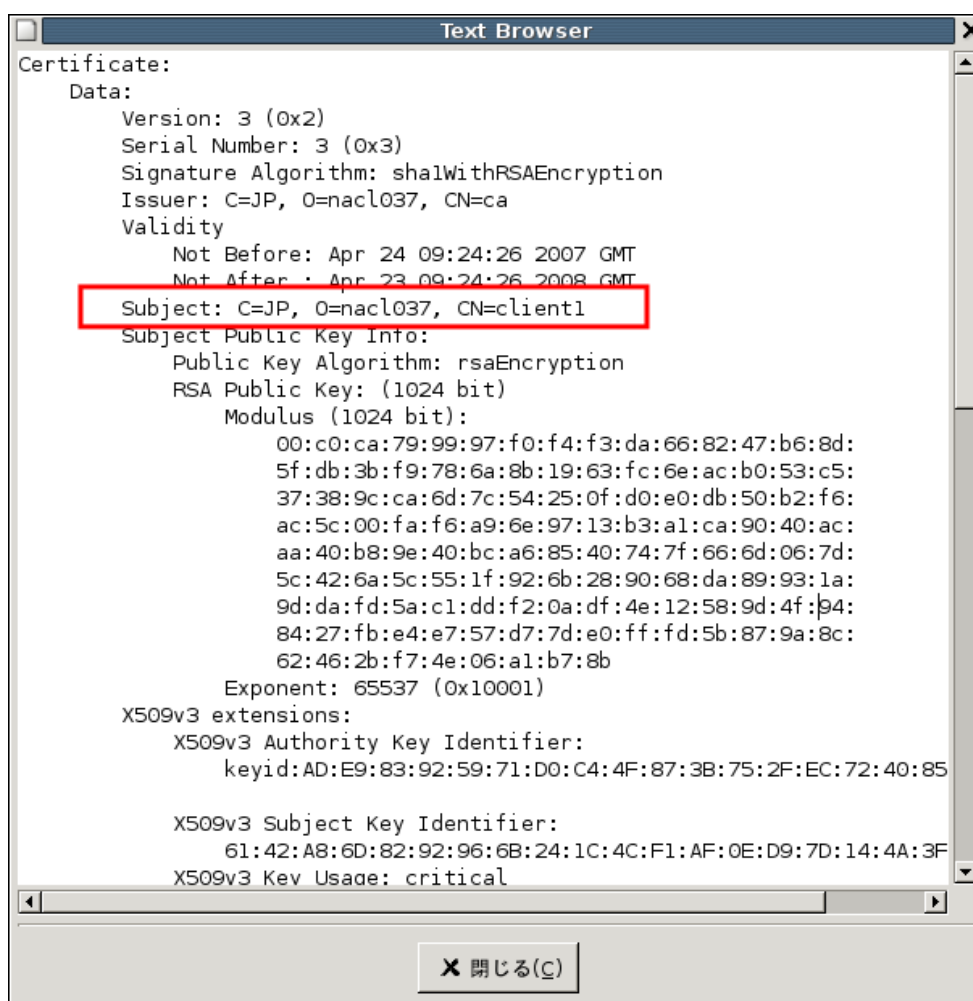


図 25: プロパティ画面

確認が終わったら画面を閉じ、デバイスを機材から外します。

## 5.2.4 PINのブロック

セキュリティデバイスの初期化や証明書のエクスポートの際に、PINの入力を行います

が、PIN入力を連続して失敗するとPINがブロックする場合があります。Aladdin eToken PRO 32Kでは4回、Cryptoflex 32Kでは3回連続でPIN入力失敗するとブロックが発生します。PINがブロックすると以下の操作が実行できなくなります。

- 証明書のエクスポート
- デバイスの初期化(Aladdin eToken PRO 32K の場合のみ)
- glclient でのセキュリティデバイスの利用

### 5.3 プロファイル

「証明書管理ツールのセットアップ」と同時にサーバ証明書プロファイル (orca-server) とクライアント証明書プロファイル (orca-client) が自動生成されます。詳細は「認証局タブウィンドウ」で確認、修正が可能です。通常お使いの場合は特に変更する必要はありません。