MONTSUQI - PostgreSQL SSL 接続

目次

montsuqi から他のマシンの PostgreSQL へ TCP/IP で接続している通信を SSL による暗号化通信にします。

標準的な日医標準レセプトソフトの構成では主サーバの dbredirector から従サーバへの PostgreSQL への接続が対象となります。

Debian GNU/Linux 4.0 Etch の PostgreSQL 8.1 を対象とします。

Debian GNU/Linux 4.0 etch 日医標準レセプトソフトインストール手順書 2 版の手順に従ってインストールした構成を想定しています。また IP アドレスは以下のアドレスを仮定していますので、順次読みかえてください。

・主サーバの IP アドレス 192.168.1.11

・従サーバの IP アドレス 192.168.1.12

各サーバの作業はそれぞれ以下の背景色で示しています。

主サーバ(192.168.1.11)での作業

従サーバ(192.168.1.12)での作業

ここからの作業は sudo が使用できるユーザで実行されることを前提にしています。sudo 時に聞かれるパスワードは省略していますので、随時自分のパスワードを入力してください。

PostgreSQL の設定変更

リモートマシンから接続される(従サーバ 192.168.1.12 の)PostgreSQL の設定を変更します。

/etc/postgresql/8.1/main/postgresql.conf を変更します。

Debian GNU/Linux 4.0 etch 日医標準レセプトソフトインストール手順書 2 版に従っていれば変更済みなはずです。

```
# gedit /etc/postgresql/8.1/main/postgresql.conf
```

```
#listen_addresses = 'localhost'
↓
listen_addresses = '*'
```

さらに SSL が有効であることを確認します。

```
ssl = true
```

続けてインストール手順書に沿って追加変更した項目(host)を SSL 対応(hostssl)に変更します。

```
# gedit /etc/postgresql/8.1/main/pg_hba.conf
```

```
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
host    orca        orca      192.168.1.11/32       password
↓
hostssl orca        orca      192.168.1.11/32       password
```

hostssl に変更するときに SSL 以外のリモート接続のエントリーが残っていないか確認します。

証明書の用意

PostgreSQL の SSL 接続をするためには、以下のサーバファイルを使用します。

ファイル	内容
server.crt	サーバ証明書
server.key	サーバの秘密鍵
root.crt	CA 証明書(クライアント証明書の検証用)
root.crl	失効された証明書リスト

サーバ認証 (正しいサーバであることを証明書により検証) では、server.crt と server.key が必要です。SSL 対応するためには、最低限この 2 つを用意します。

クライアント認証(クライアントを証明書により検証)する場合は、root.crt が必要です。

root.crl は必要なときのみ用意します。

証明書ファイルは $PGDATA/ に置く必要があります。

デフォルトでは、/var/lib/postgresql/8.1/main/になります。

Debian パッケージでは、pg_createcluster を実行したときに、あらかじめ証明書が用意されています。

```
$ sudo ls -l /var/lib/postgresql/8.1/main
```

```
-rw------- 1 postgres postgres    4 Oct 18  2007 PG_VERSION
drwx------ 6 postgres postgres 4096 Jun 16 10:17 base
～～～～～～～～～～～～～～～ 略 ～～～～～～～～～～～～～～～～～～～～～～～～～～～～
lrwxrwxrwx 1 root   root     31 Aug  6 13:31 root.crt -> /etc/postgresql-common/root.crt
lrwxrwxrwx 1 root   root     36 Aug  6 13:31 server.crt -> /etc/ssl/certs/ssl-cert-snakeoil.pem
lrwxrwxrwx 1 root   root     38 Aug  6 13:31 server.key -> /etc/ssl/private/ssl-cert-snakeoil.key
```

/etc/postgresql-common/root.crt は説明が書かれたダミーファイルです。

server.crt と server.key はそれぞれ/etc/postgresql-common/root.crt と /etc/ssl/private/ssl-cert-snakeoil.key に実体が置かれています。

用意された server.crt, server.key は有効期限が短く、またサーバ証明書内のホスト名が一致していないと接続出来ないので、証明書を新規に作成します。

以後の説明では、実体を /etc/postgresql/8.1/main/ に置き /var/lib/postgresql/8.1/main へシンボリックリンクを張るようにします。

証明書の作成方法はいくつかありますが、コマンドラインから自己署名証明書を作成する方法とプライベート CA 構築ツールを使用して作成する方法を説明します。

PostgreSQL への接続が主サーバから従サーバの 1 台のみであれば、自己署名証明書でも構いませんが、複数台 PostgreSQL へ接続するような場合には、クライアントの検証が CA 証明書を使用して出来るため、プライベート CA 構築ツールを使用した方がよいでしょう。

コマンドラインから自己署名証明書を作成する方法

ここでは組織内で使用されることを前提に自己署名証明書を作成します。

まず /etc/postgresql/8.1/main/に移動し証明書秘密鍵を作成します。

```
$ cd /etc/postgresql/8.1/main/
$ sudo openssl genrsa -out server.key 1024
```

```
Password:
Generating RSA private key, 1024 bit long modulus
.........++++++
.....................++++++
e is 65537 (0x10001)
```

有効期限を決めて自己署名証明書を作成します。ここでは 1096 日を指定してます。

```
$ sudo openssl req -new -key server.key -x509 -days 1096 -out server.crt
```

実行するとメッセージが出て質問が出るので答えていきます。

重要なのは 「Common Name」で外部からアクセスするためのホスト名か IP アドレスを入れます(主サーバから psql -h ホスト名 でアクセス出来るようなホスト名です)。

ここでは従サーバのアドレス(192.168.1.12)を入力しています。

```
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
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []: 192.168.1.12
Email Address []:
```

これで server.key と server.crt という 2 つのファイルが作成されました。

証明書の設置に進んでください。

プライベート CA 構築ツール(jma-certtool)を使用する方法

プライベート CA 構築ツールの利用方法を参考に jma-certtool をインストールし CA を作成してください。glserver-glclient で SSL を使用するために CA を作成してあれば、その CA がそのまま使用できます。

jma-certtool は主サーバで起動されることを想定しています。

まず、PostgreSQL サーバ用の証明書を作成します。

jma-certtool を起動し証明書タブが表示されている状態で「新規」ボタンをクリックします。

```
$ jma-certtool
```

証明書要求の編集画面が開きますので、コモンネームに PostgreSQL が外部からアクセスするためのホスト名か IP アドレスを入れます(主サーバから psql -h ホスト名 でアクセス出来るようなホスト名です)。

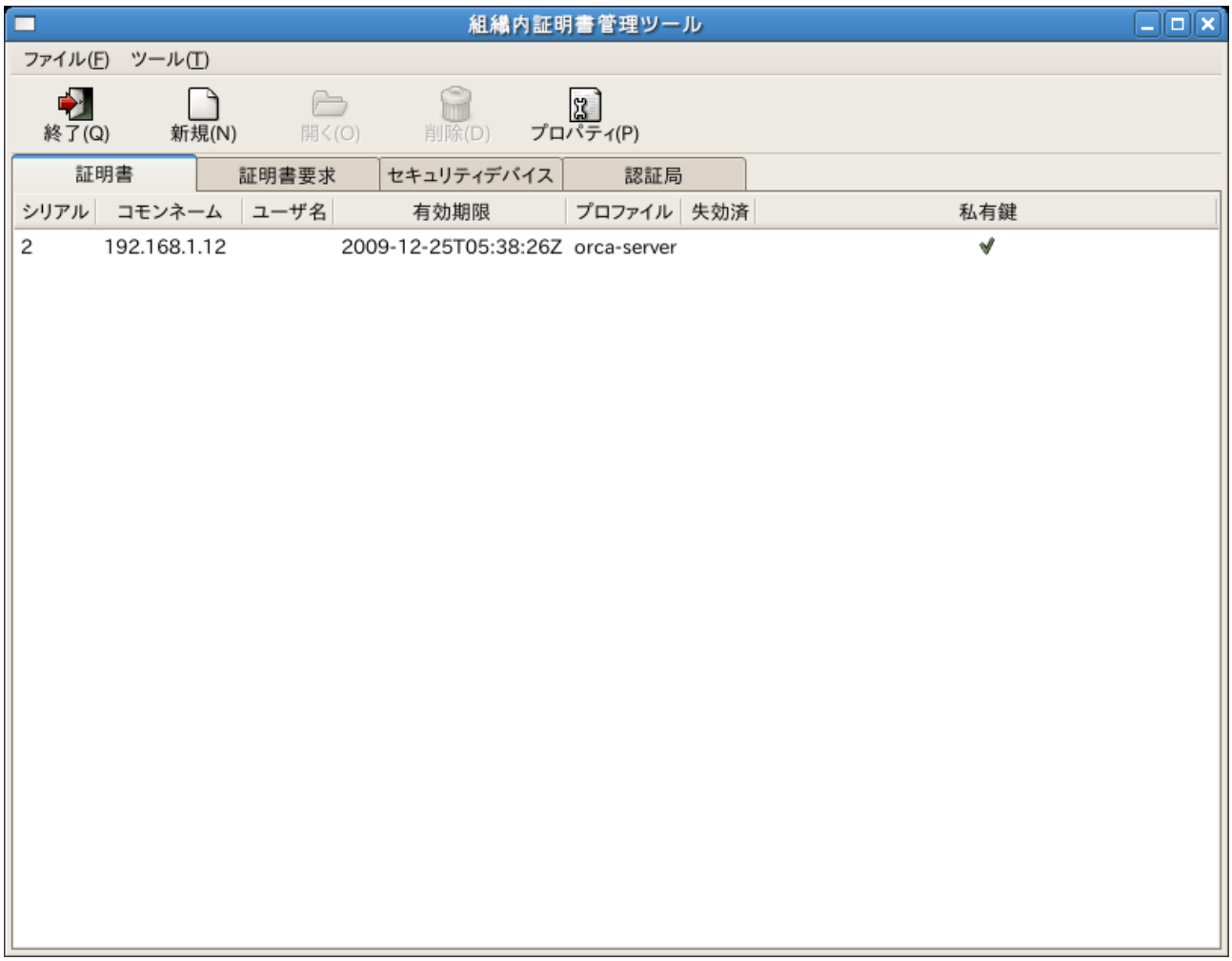
「OK」をクリックします。

CA による署名画面が開きます。

CA 鍵のパスワードを入力し、プロファイルは「orca-server」にし、証明書の有効期限に
を入れます。ここでは 1096 日としました。ユーザ名は空欄にします。

「OK」をクリックします。

証明書欄に作成された証明書が発行されていますので、右クリックします。



エクスポート(PKCS #12)を選択します。

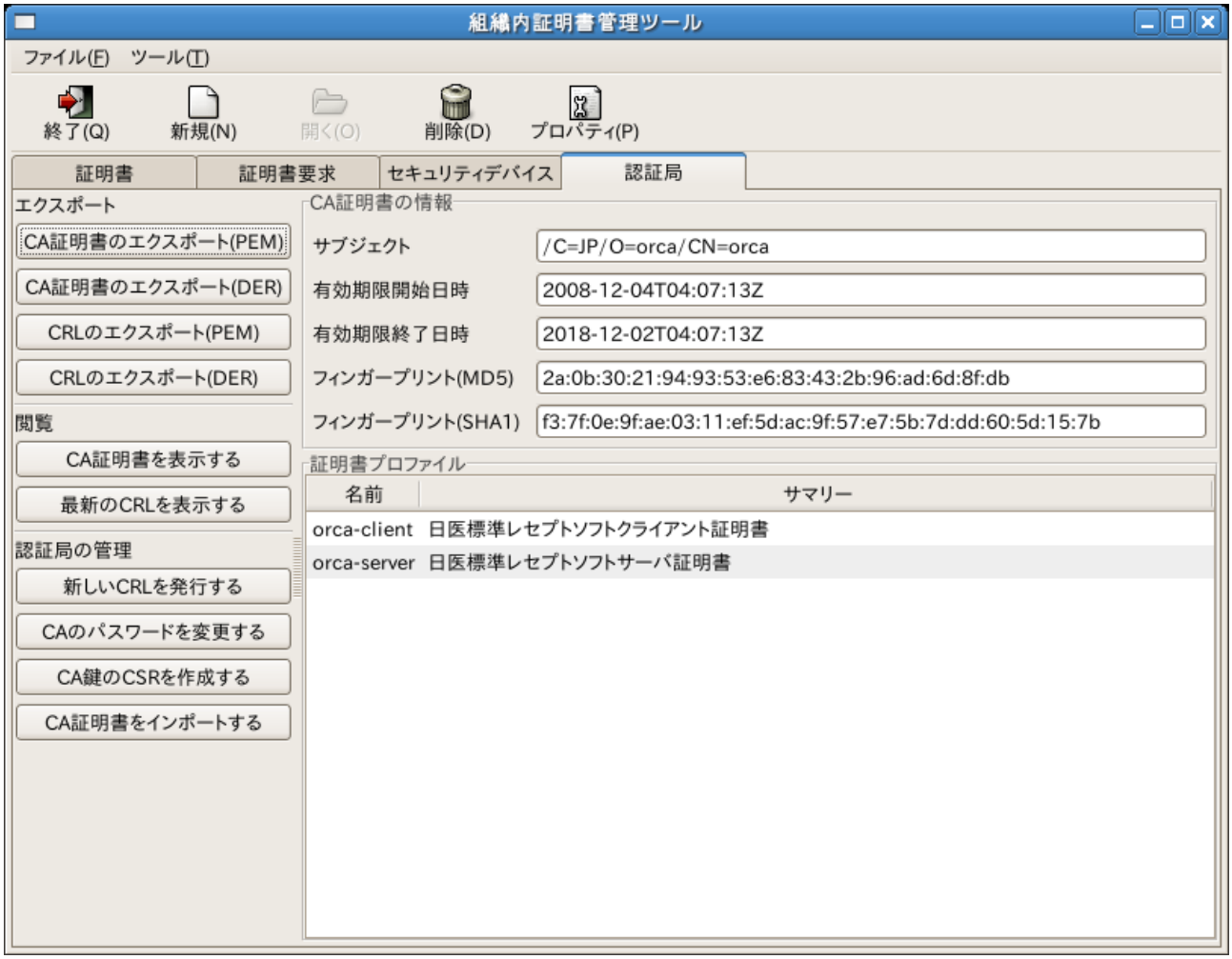
パスワード入力画面が表示されるので、パスワード欄は空欄のまま「OK」をクリックします。

「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。

出力ファイル名の選択画面になりますので 192-168-1-12.p12 というようなファイル名で OK を押して保存します。

続けて、CA 証明書もエクスポートしておきます。

「認証局」タブをクリックし「CA 証明証のエクスポート(PEM)」をクリックします。



root.crt というファイル名で保存しておきます。

PostgreSQL では #pkcs 12 形式はそのままでは使用できないので、変換する必要があります。

jma-certtool を終了し、エクスポートされたファイルがあるディレクトリに移動しコマンドラインから以下のコマンドを入力します。

```
$ sudo openssl pkcs12 -in 192-168-1-12.p12 -clcerts -nokeys -out server.crt
$ sudo openssl pkcs12 -in 192-168-1-12.p12 -nodes -nocerts -out server.key
```

以上で server.crt と server.key というファイルが作成されています。

変換した後の 192-168-1-12.p12 は必要がないので削除します。

```
$ sudo rm  192-168-1-12.p12
```

server.crt と server.key を従サーバ(192.168.1.12)の/etc/postgresql/8.1/main/ にコピーします。USB メモリ等を使用してコピーしても構いません。

ここではアーカイブしてから従サーバのホームディレクトリに一旦コピーします。

```
$ sudo tar czf server.tar.gz server.crt server.key
$ scp server.tar.gz 192.168.1.12:~/
```

従サーバにログインし /etc/postgresql/8.1/main/ に展開します。

```
$ sudo tar xzf ~/server.tar.gz -C /etc/postgresql/8.1/main
```

証明書の設置

postgres ユーザで読めるようにし、秘密鍵は postgres ユーザのみが読めるようにパーミッションを変更します。

```
$ cd /etc/postgresql/8.1/main/
$ sudo chown postgres.postgres server.crt
$ sudo chown postgres.postgres server.key
$ sudo chmod 400 server.key
$ sudo chmod 444 server.crt
```

/var/lib/postgresql/8.1/main/ にシンボリックリンクを張ります。

```
$ sudo ln -sf  /etc/postgresql/8.1/main/server.crt
              \ /var/lib/postgresql/8.1/main/server.crt
$ sudo ln -sf  /etc/postgresql/8.1/main/server.key
              \ /var/lib/postgresql/8.1/main/server.key
```

サーバの証明書の準備が整ったのでサーバを再起動します。

```
$ sudo /etc/init.d/postgresql-8.1 restart
```

以上で PostgreSQL の設定が終わりました。

PostgreSQL サーバの検証

従サーバの PostgreSQL が SSL 接続のみ許可した状態で起動されています。psql は SSL 接続が可能なので、このまま主サーバから従サーバへ psql でアクセス可能です。

主サーバ(192.168.1.11)のマシンから実行

```
$ sudo su orca
$ psql -h 192.168.1.12 orca
```

```
Welcome to psql 8.1.13, the PostgreSQL interactive terminal.


Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit


SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)


orca=> \q
```

SSL connection と出て SSL で接続していることがわかります。

ただし、このままではサーバの証明書を検証していない状態です。

クライアント側で用意するファイルはアプリケーションによって異なりますが、psql や libpq を使用したプログラム、MONTSUQI の SSL 接続では $HOME/.postgresql の以下のファイルを使用します。

ファイル	内容
$HOME/.postgresql/postgresql.crt	クライアント証明書
$HOME/.postgresql/postgresql.key	クライアント秘密鍵
$HOME/.postgresql/root.crt	CA 証明書(サーバ証明書の検証用)
$HOME/.postgresql/root.crl	失効された証明書リスト

サーバの検証をするには、 $HOME/.postgresql/root.crt ファイルを用意します。

まず $HOME/.postgresql を作成します。

```
$ sudo su orca                          # すでに orca ユーザの場合は不要
$ mkdir /home/orca/.postgresql
```

root.crt を用意する前に、接続出来ていたので、まず正しくない証明書の場合は接続出来ないことを確認してみます。

従サーバの証明書ではない証明書を root.crt にコピーします。

```
# 主サーバの ssl-cert-snakeoil.pem をコピー
# 従サーバが使用している証明書ではないので、間違っている証明書
$ cp /etc/ssl/certs/ssl-cert-snakeoil.pem /home/orca/.postgresql/root.crt
```

従サーバへ接続

```
$ psql -h 192.168.1.12
psql: SSL error: certificate verify failed
```

と出て接続エラーとなります。

サーバが正しくないときには、接続されないことが確認できました。

次に正しい証明書をセットします。

「コマンドラインから自己署名証明書を作成する方法」で作成した場合は、従サーバの /etc/postgresql/8.1/main/server.crt を 主サーバの/home/orca/.postgresql/root.crt にコピーします。

ここでは、scp を使ってコピーします。USB メモリ等を使用してコピーしても構いません。

```
$ scp 192.168.1.12:/etc/postgresql/8.1/main/server.crt
    \  /home/orca/.postgresql/root.crt
```

「プライベート CA 構築ツール(jma-certtool)を使用する方法」で作成した場合は、あらかじめエクスポートしておいた root.crt をコピーします。

```
$ cp root.crt /home/orca/.postgresql/root.crt
```

正しい証明書がセットされていれば接続出来るようになります。

```
$ psql -h 192.168.1.12 orca
```

```
Password for user orca:
Welcome to psql 8.1.13, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

orca=> \q
```

以上でサーバ証明書でサーバの検証が出来るようになりました。

日レセの設定

主サーバの/etc/jma-receipt/dbgroup.inc を設定して dbredirector を SSL 接続にします。

従サーバへの接続設定に sslmode "require" という行を追加します。

```
$ sudo gedit /etc/jma-receipt/dbgroup.inc
```

```
db_group "log" {
  priority 100;
  type "PostgreSQL";
  port "192.168.1.12";
  name "orca";
  user  "orca";
  password "orca123";
  sslmode "require";     # <------------------------------ 追加
  file "/var/lib/jma-receipt/dbredirector/orca.log";
  redirect_port "localhost";
};
```

PostgreSQL の設定と証明書の準備が出来ていれば、これで SSL で接続されるようになります。日レセサーバを再起動して動作を確認してください。

```
$ sudo /etc/init.d/jma-receipt restart
```

クライアント認証

これまでの作業でサーバ証明では正しいサーバであることを検証し、認証はパスワード認証を利用しています。

証明書によるクライアント認証をすることも可能です。

コマンドラインから自己署名証明書を作成する方法

主サーバ側で証明書を作成します。

```
$ cd /home/orca/.postgresql
$ openssl genrsa -out postgresql.key 1024
$ openssl req -new -key postgresql.key -x509 -days 1096 -out postgresql.crt
```

```
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
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:
```

クライアント側の証明書は CN はなんでも構いません。

postgresql.key は秘密鍵なのでパーミッションを設定し他の人から見えないようにします。

```
$ chmod 400 postgresql.key
```

postgresql.crt をサーバ側の root.crt に追加します。

ここでは scp を使用します。まず主サーバから従サーバへファイルをコピーします。

```
$ scp postgresql.crt 192.168.1.12:~/
```

今度は従サーバ側で主サーバからコピーされたファイルを所定の位置にセットします。

```
$ sudo cp ~/postgresql.crt /etc/postgresql/8.1/main/root.crt
$ sudo ln -sf  /etc/postgresql/8.1/main/root.crt
     \ /var/lib/postgresql/8.1/main/root.crt
```

以上で証明書の準備が整いました。PostgreSQL の再起動に進んでください。

プライベート CA 構築ツール(jma-certtool)を使用する方法

jma-certtool で作成する場合は CA 証明書を作成しているので、作成した root.crt がクライアント証明書の検証にも使用できます。

主サーバに設置した /home/orca/.postgresql/root.crt ファイルを従サーバに scp 等でコピーし設置します。

主サーバから従サーバにコピーします。

```
$ scp root.crt /home/orca/.postgresql/root.crt 192.168.1.12:~/
```

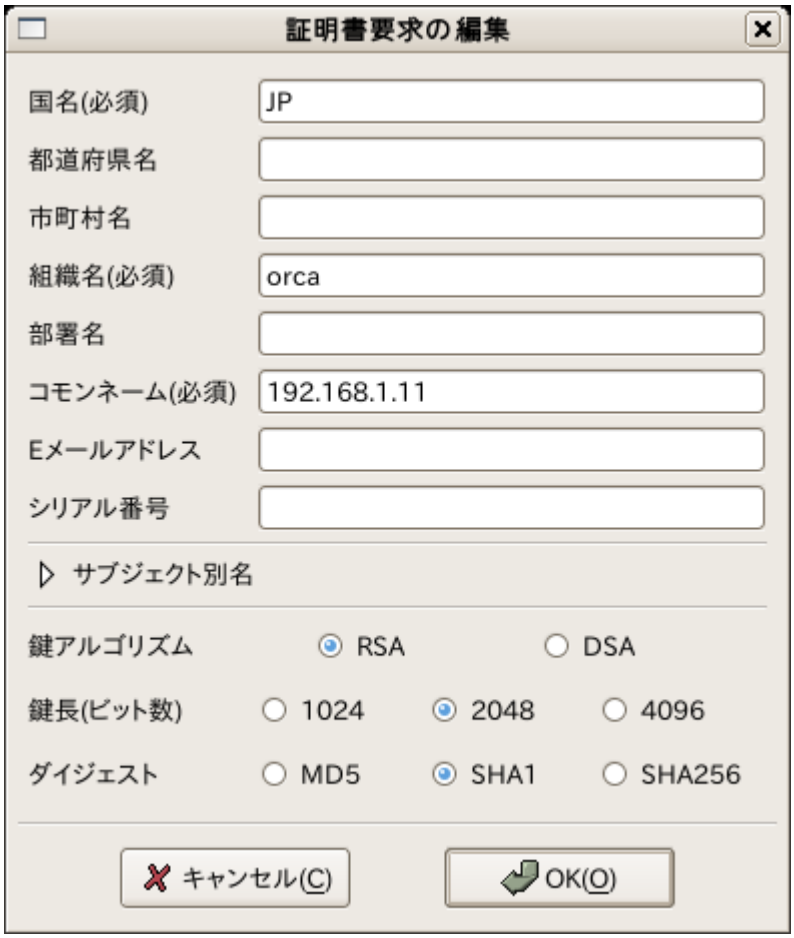
従サーバで root.crt を設置します。

```
$ sudo cp root.crt /etc/postgresql/8.1/main/root.crt
$ sudo ln -sf  /etc/postgresql/8.1/main/root.crt
     \ /var/lib/postgresql/8.1/main/root.crt
```

続けてクライアント証明書を発行します。

```
$ jma-certtool
```

サーバ証明書と同じように証明書を発行します。サーバとは違いコモンネームはチェックされないので、識別出来るような名前を入力します。



「CA による署名」画面でも同様に入力します。

プロファイルは orca-client は glclient 用なので、orca-server で代用します。

作成されたらサーバ証明書と同様にエクスポート(PKCS #12)をします。

jma-certtool を終了し、エクスポートされたファイルのディレクトリに移動しコマンドラインから以下を実行します。

```
$ sudo openssl pkcs12 -in 192-168-1-11.p12 -clcerts -nokeys -out
                        \ /home/orca/.postgresql/postgresql.crt
$ sudo openssl pkcs12 -in 192-168-1-11.p12 -nodes -nocerts -out
                        \ /home/orca/.postgresql/postgresql.key
```

変換後の 192-168-1-11.p12 は必要がないので削除します。

```
$ sudo rm  192-168-1-11.p12
```

orca ユーザ権限に変更し秘密鍵は他の人から見えないようにします。

```
$ sudo chown orca.orca /home/orca/.postgresql/postgresql.crt
$ sudo chown orca.orca /home/orca/.postgresql/postgresql.key
$ chmod 400 postgresql.key
```

PostgreSQL の再起動

従サーバの PostgreSQL を再起動します。

```
$ sudo /etc/init.d/postgresql restart
```

主サーバから接続してみます。

```
$ psql -U orca -h 192.168.1.12 orca
```

```
Password for user orca:
Welcome to psql 8.1.13, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

orca=> \q
```

接続できるようになりました。

インストール手順書により設定していたため、パスワードも聞かれますが、
/var/lib/postgresql/8.1/main/root.crt が 設定されたサーバにはクライアントに
postgresql.crt と postgresql.key というファイルがなければ接続出来ないようになります。

証明書で認証しているためパスワードが不要という場合は pg_hba.conf ファイル
で該当の認証を trust とします。

orca ユーザの psql で接続出来るようになっていれば、日レセではそのままクライアント認証も使用できるようになります。