記事
Toshihiko Minamoto · 2020年12月24日 6m read

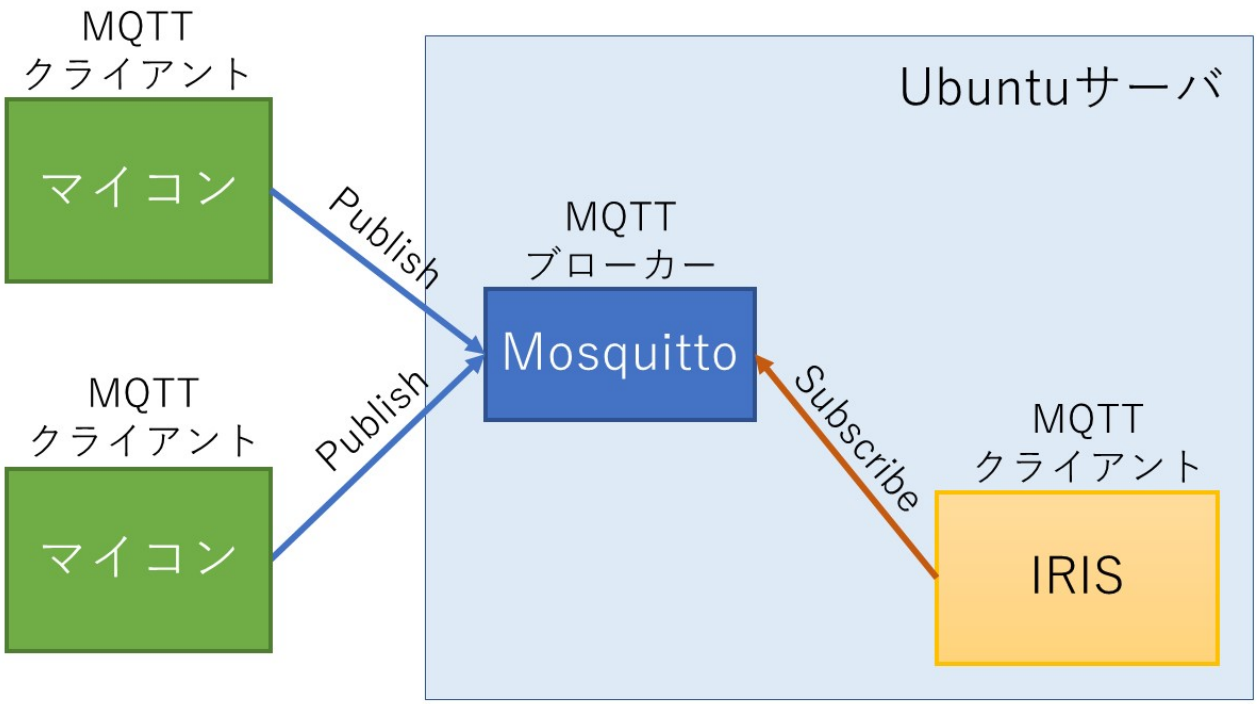
MQTTを使ったIRISとデバイスとの通信

皆さんこんにちは。

IRIS 2020.1からMQTTアダプタが新たに追加されました。MQTTはPublish/Subscribe型のシンプルで軽量なメッセージングプロトコルです。帯域が低いネットワーク環境やArduinoやRaspberry
PIといったワンボードマイコンやシングルボードコンピュータなどでも動作できます。
今回はクラウド上のUbuntuサーバにIRISをインストールし、MQTTアダプタを使いEsp8266というマイコンで取得した温度、湿度、気圧データをIRIS上のデータベースに登録する方法について説明したいと思います。

アーキテクチャ

今回のアーキテクチャは以下の通りです。



データを送受信するマイコンやアプリケーションにMQTTクライアントを組み込み、MQTTクライアントがMQTTブローカーというメッセージを転送するソフトウェアに接続します。

同じMQTTブローカーにつながっているMQTTクライアントにてメッセージを送信したい場合はPublishメッセージを使用し、メッセージと共にトピックと呼ばれるメッセージの種類を表す文字列を送信します。情報を受信したい場合はSubscribeメソッドを使用し、受信したいトピックを指定します。

IRISのインストール

今回、Ubuntu版のIRIS 2020.1 community
editionをインストールしています。お持ちでない方はこちらからダウンロードをお願いします。
インストール方法につきましてはこちらの記事の動画(11分24秒あたりから)をご参照ください。

Mosquittoのインストール

MQTTブローカーとしてMosquittoをインストールします。
これには、Ubuntuのターミナルよりaptというパッケージマネージャを使用します。

```
$ sudo apt-get update

$ sudo apt-get install mosquitto
```

これで、Mosquittoのインストールができます。

Mosquittoの設定

この状態でもMosquittoに接続することは可能ですが、誰でも接続できてしまいますので、パスワードを設定することにします。
まずは -c オプションを指定し、パスワードファイル(/etc/mosquitto/password)を作成します。

```
sudo mosquitto_passwd -c /etc/mosquitto/password <????> <?????>
```

既にあるパスワードファイルに追加する場合は

```
sudo mosquitto_passwd /etc/mosquitto/password <????> <?????>
```

となります。
今回は、マイコン側からアクセスするユーザ「mqttclient」とIRISがアクセスするユーザ「iris」を作成しました。

mosquitto.confに以下の行を追加し、Anonymousユーザのアクセス禁止とパスワードファイルを設定します。

```
      :
 allow_anonymous false
 password_file /etc/mosquitto/password
      :
```

証明書の作成

MQTTクライアントとMQTTブローカの間は暗号化されていないため、認証時のユーザ名、パスワードが盗聴されてしまいます。
そこでサーバ証明書を発行し、通信を暗号化させることにします。
ここでは、自己認証局ならびにサーバの証明書を作成します。作成にはIRISのPKI機能を利用します。

認証局の秘密鍵の作成

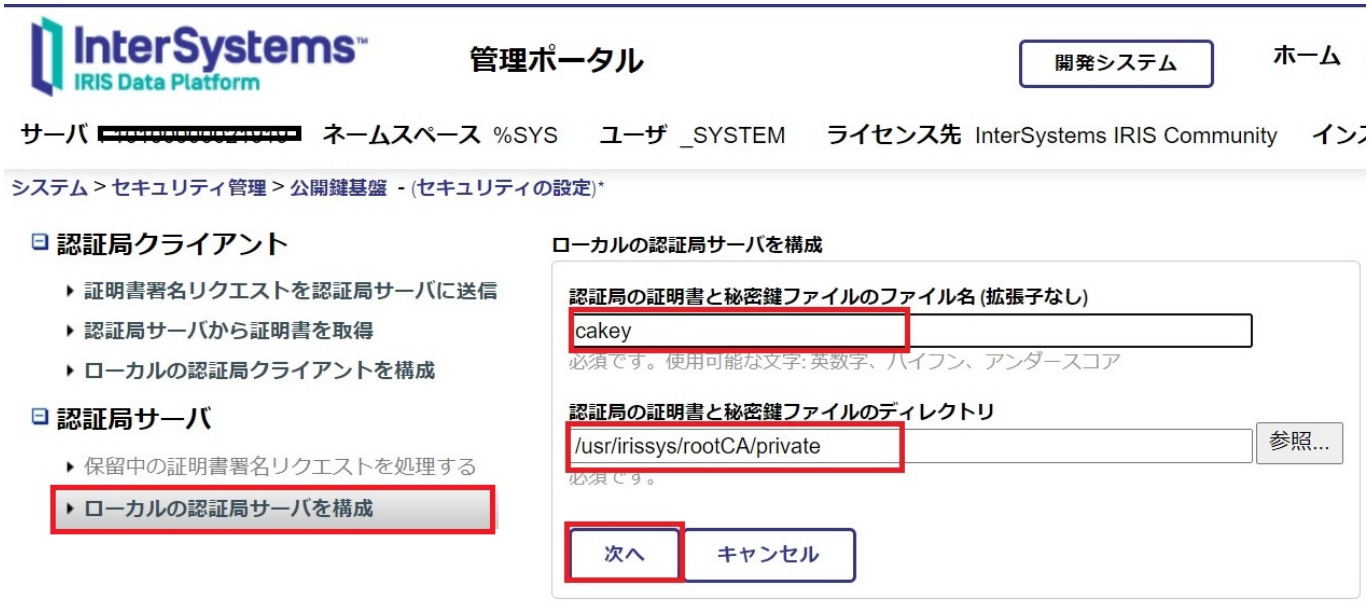
Ubuntuターミナルから認証局の秘密鍵を格納するディレクトリを作成します。

```
$ sudo mkdir -p  /usr/irissys/rootCA/private
```

このとき管理ポータルからアクセスできるよう、ディレクトリのオーナーをirisusrに変更します。IRISインストール時に実行ユーザを変更されている場合はその内容に合わせてオーナーを変更してください。

```
$ sudo chown irisusr:irisusr /usr/irissys/rootCA/private
```

管理ポータルを起動し、「システム管理」「セキュリティ」「公開鍵基盤」をクリックします。
画面左下の「ローカルの認証局サーバを構成」をクリックしますと、以下のように右側に「ローカルの認証局サーバを構成」と書かれた部分が
現れます。ここで認証局のファイル名にファイルの名称、秘密鍵ファイルのディレクトリを入力し「次へ」をクリックします。



ここで、秘密鍵のディレクトリは右にある「参照...」ボタンでディレクトリを選択することも可能です。
また、ファイル名欄には拡張子は入れないでください。

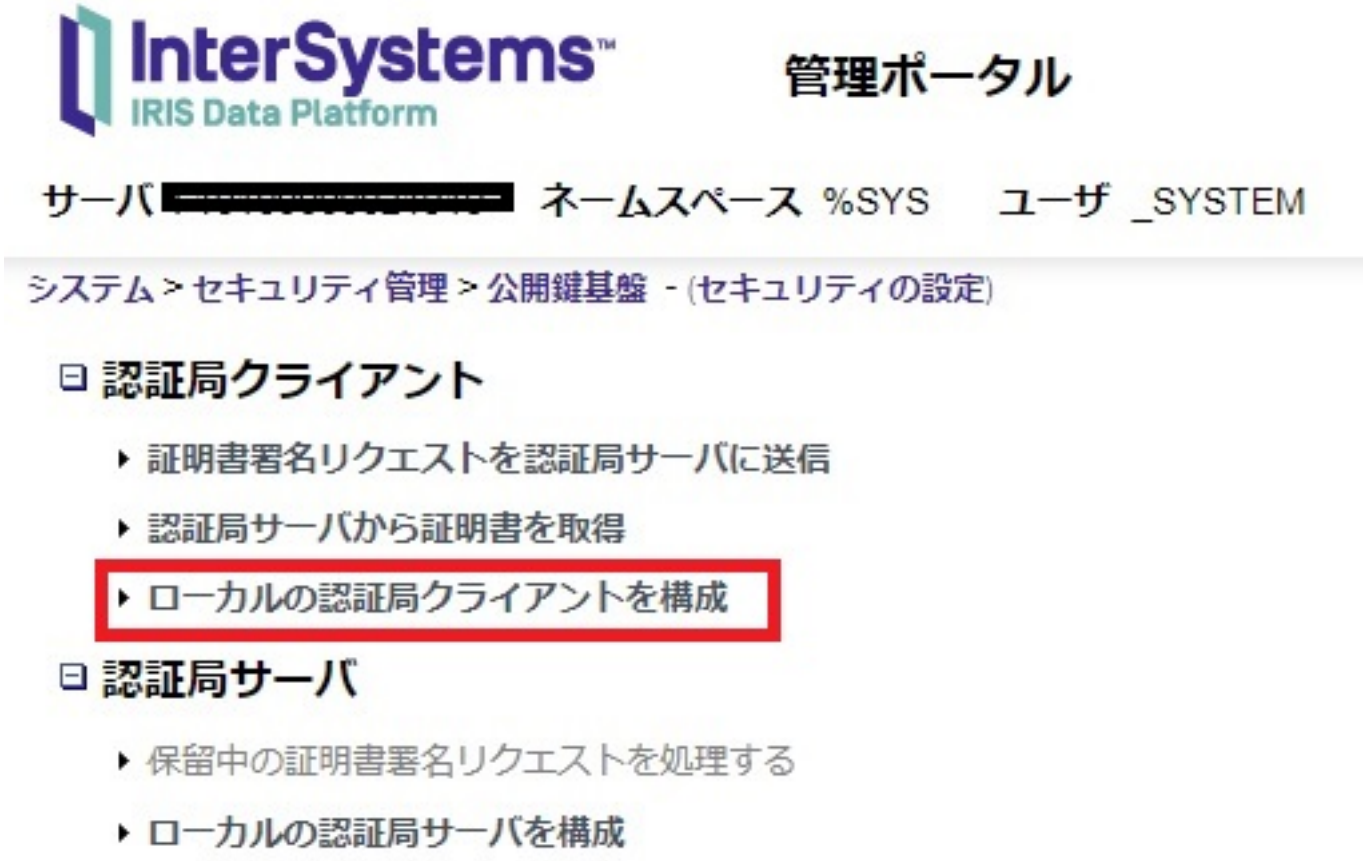
「次へ」をクリックしますと以下の画面となり、秘密鍵のパスワードや、証明書情報、有効期間を入力し「保存」をクリックします。

秘密鍵(cakey.key)、証明書(cakey.cer)が指定されたディレクトリに作成されます。

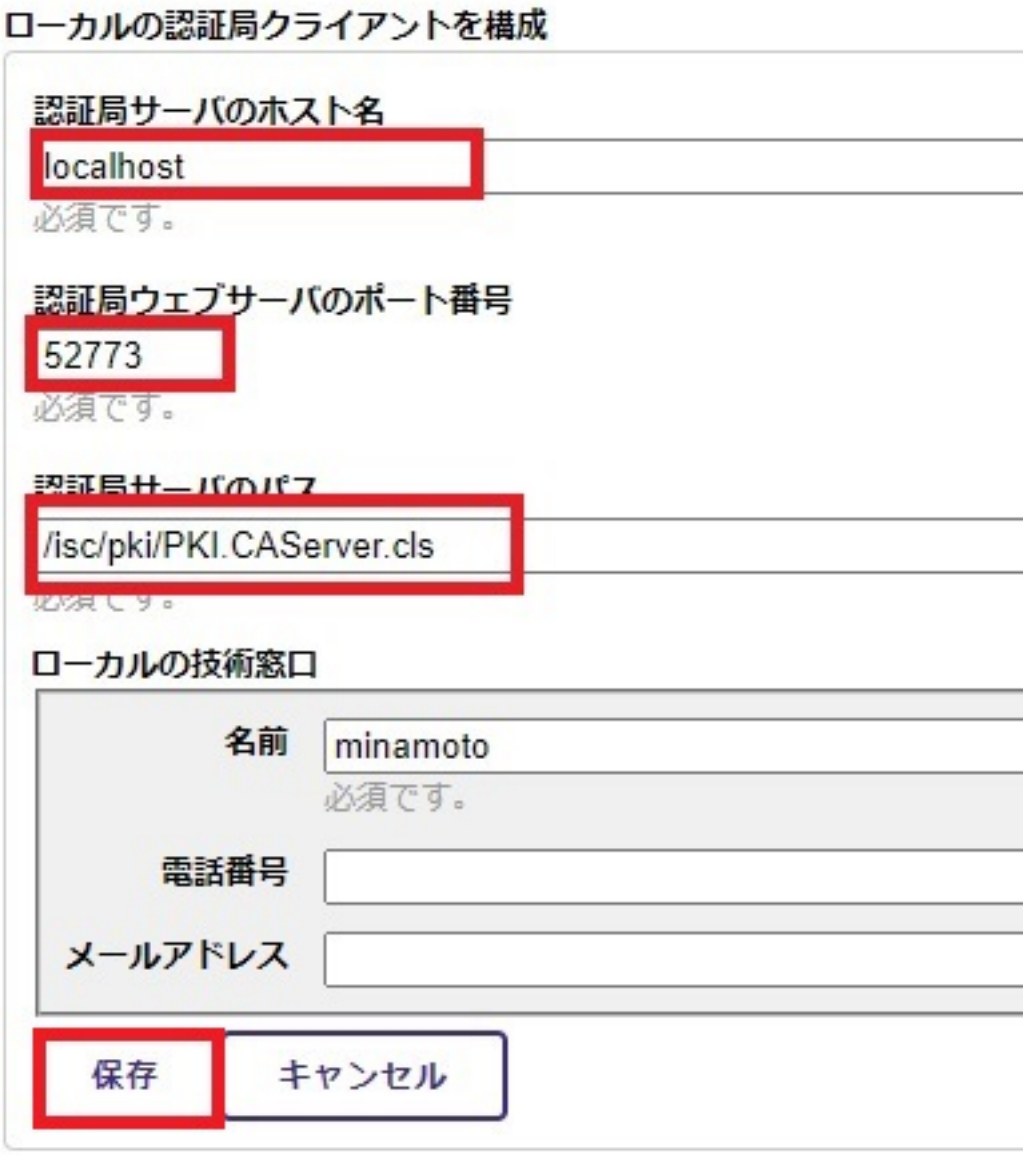
認証局クライアントの設定

先ほど作成した認証局への接続設定を行います。

以下のメニューで「ローカルの認証局クライアントを構成」をクリックします。



以下の画面が表示されます。同じ環境ですので、そのままで「保存」をクリックします。

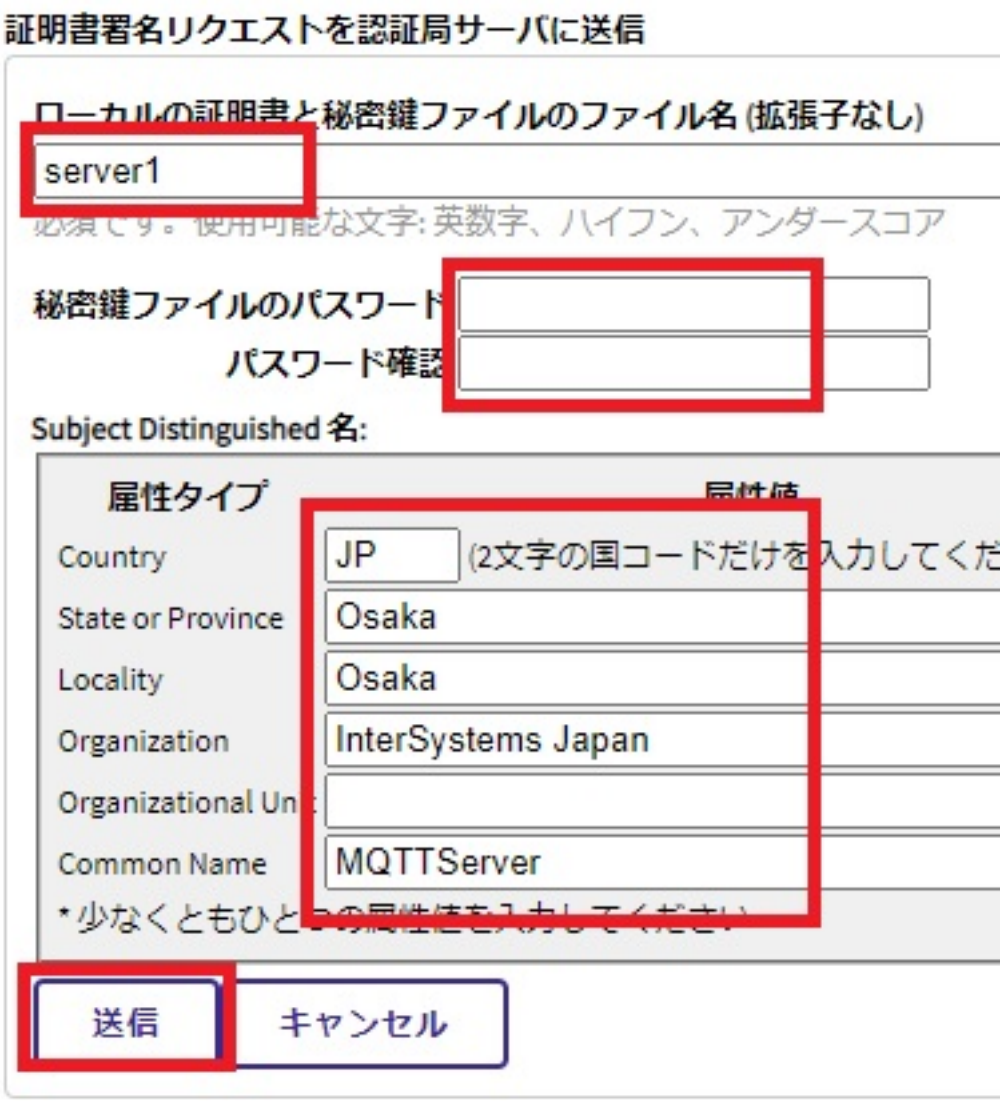


サーバ認証の作成

続いて暗号化通信を行うサーバ証明書を作成します。

以下のメニューで「証明書署名リクエストを認証局サーバに送信」をクリック

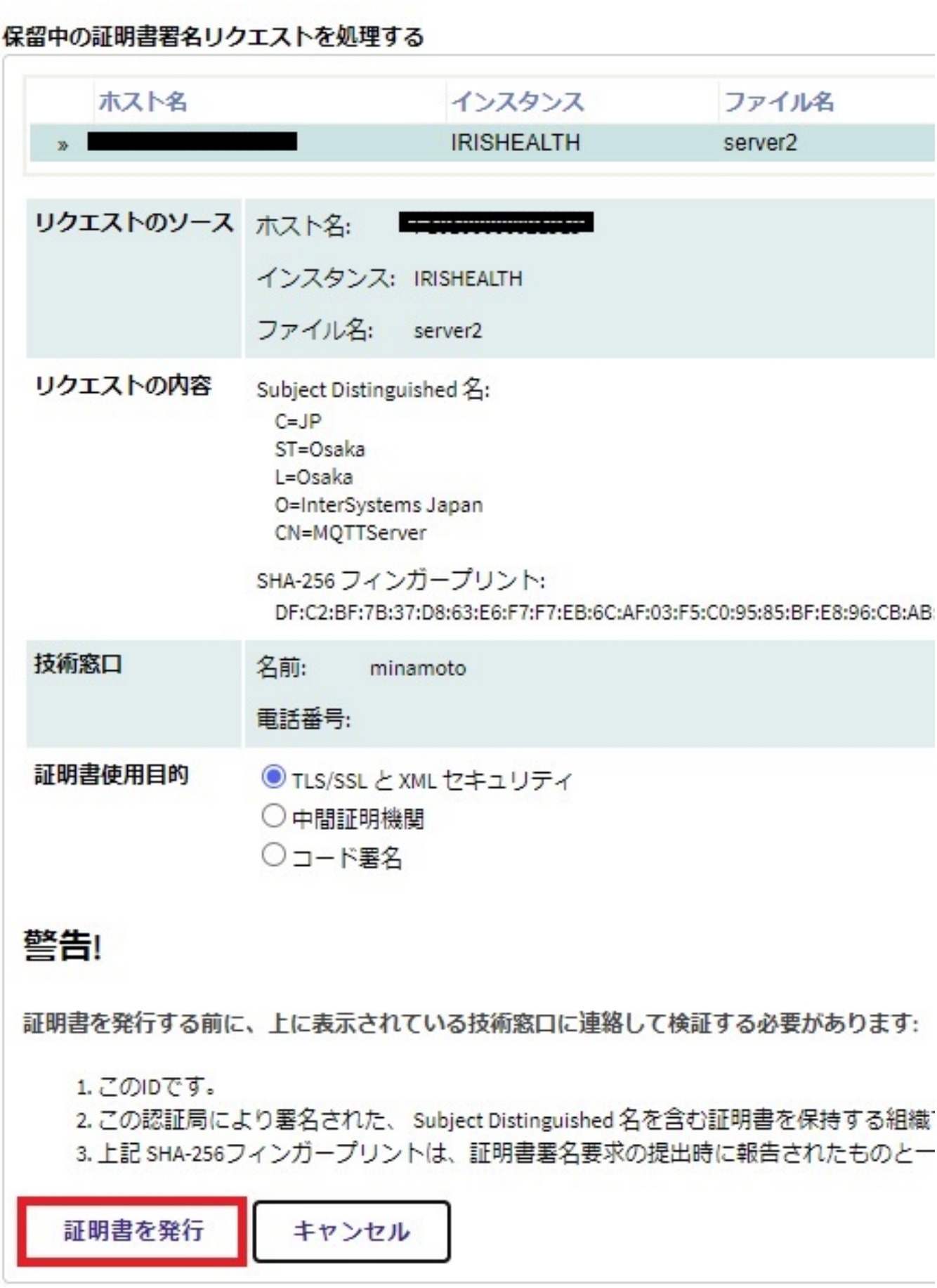
以下の画面が表示されるので、ファイル名、証明情報を入力し、「送信」をクリックします。
パスワードは設定しないでください。



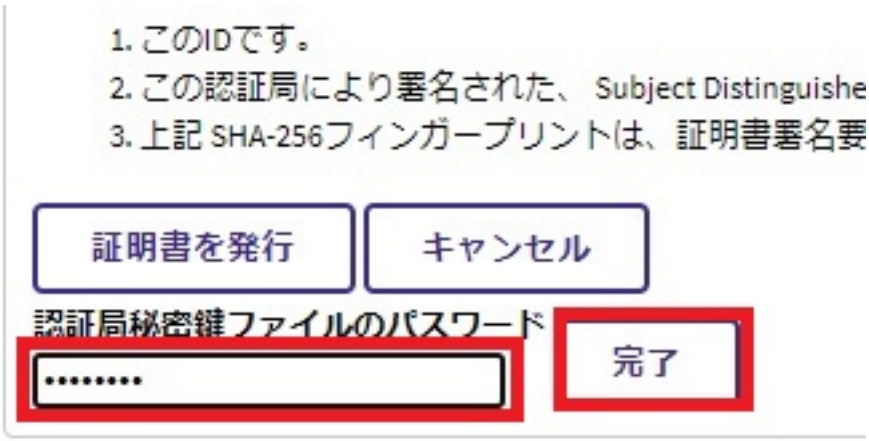
送信をクリックしますと、以下のように「保留中の証明書署名リクエストを処理する」が選択できるようになりますので、それをクリックします。

クリックすると以下の画面が表示されますので、該当するリクエストの「プロセス」をクリックします。

すると証明書の詳細が表示されますので、「証明書を発行」ボタンをクリックします。



さらに認証局のパスワードを入力画面が下に現れますので、パスワードを入力し、「完了」ボタンをクリックします。



作成されたサーバ証明書の取得

作成されたサーバ証明書は以下のメニューにて「認証局サーバから証明書を取得」をクリックします。

以下の画面が表示されますので、「取得」ボタンをクリックしますと/usr/irissys/mgrディレクトリに秘密鍵(<ファイル名>.key)、証明書(<ファイル名>.cer)が作成されます。
これらをMosquittoにコピーします。

```
sudo cp /usr/irissys/mgr/server2.key /etc/mosquitto/certs
sudo cp /usr/irissys/mgr/server2.cer /etc/mosquitto/certs
sudo cp /usr/irissys/rootCA/private/cacert.cer /etc/mosquitto/ca_certificates
```

さらにserver2.keyの権限を変更します。（変更しないとエラーが発生します）

```
$ sudo chmod 600 server2.key
```

mosquitto.confに証明書を設定します。

```
      :
cafile /etc/mosquitto/ca_certificates/cacert.cer
certfile /etc/mosquitto/certs/mqttsrv2.cer
keyfile /etc/mosquitto/certs/mqttsrv2.key
      :
```

さらにmosquitto.confに以下の行を追加し、ポート番号を8883に変更します。

```
port 8883
```

Mosquittoを再起動し設定を反映させます。

```
/etc/init.d/mosquitto restart
```

以上でMosquittoと他のデバイス間の通信を暗号化し、接続に認証を必要とする設定が完了しました。

次回はマイコン側の配線やプログラムについて説明したいと思います。

#IoT #相互運用性 #InterSystems IRIS

ソースURL:
https://jp.community.intersystems.com/post/mqtt%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%9Firis%E3%81%A8%E3%83%87%E3%83%90%E3%82%A4%E3%82%B9%E3%81%A8%E3%81%AE%E9%80%9A%E4%BF%A1