

コーディングで学ぶ MQTT通信 & セキュリティ

ご購入はこちら

角 史生

表1 本稿で紹介するサンプル・コードごとのセキュリティ設定

サンプル・コードの特徴	リスト 番号	セキュリティ設定			備 考
		経路 暗号化	認 証	認可 (権限管理)注2	
全くセキュリティ設定を行わない	リスト1	×	×	—	シンプルなコードであるが最もセキュリティ・レベルが低い実装。盗聴されても問題のないデータを送受信するのに適する
経路暗号化のみ有効化	リスト2	△注1	×	—	通信経路暗号化することで盗聴を防ぐ。接続先サーバの証明書の検証なし
経路暗号化、ユーザ名/パスワードによる認証を有効化	リスト3	△注1	○ 認証用ユーザ名/パスワードによる認証	—	通信経路を暗号化し、認証用ユーザ名/パスワードによる認証を行う。接続先サーバの証明書の検証なし
経路暗号化、クライアント証明書による認証を有効化 (接続先サーバの証明書も検証)	リスト4	○	○ クライアント証明書による認証	—	通信経路を暗号化し、クライアント証明書による認証を行う。サーバ証明書、クライアント証明書の相互検証を実施。設定が複雑
経路暗号化、ユーザ名/パスワードによる認証を有効化 (接続先サーバの証明書も検証)	リスト5	○	○ 認証用ユーザ名/パスワードによる認証	—	通信経路を暗号化し、認証用ユーザ名/パスワードによる認証を行う。接続先サーバの証明書を検証する。設定がやや複雑

注1：接続先サーバの証明書の検証を行っていない。偽装されたサーバ証明書でも検知できず接続してしまう

注2：認可はMQTTブローカ側での設定となる

表2 求められるセキュリティ・レベルに応じたMQTTブローカの使い分け

セキュリティ・レベル	MQTTブローカ	備 考
低いセキュリティ・レベルで良いので手軽にMQTTブローカを使いたい	・パブリックMQTTブローカを選択 - EMQX Public MQTT Server - HiveMQ Free Public MQTT Broker	ユーザ登録、端末の認証が不要で、MQTT over TLSも選択可能
経路暗号化、メッセージの発行/購読を認証された端末のみに制限したい	・Shiftr.io MQTT Cloud Service	端末ごとのPub/Subの制限も可能。GUI版コンソールにより端末の接続状態、メッセージの流れが確認できる
最も高いセキュリティ設定が必要	・HiveMQ CloudServices ・EMQX Serverless MQTT Broker	端末の認証に加え、端末ごとに、どのトピックに対して配信、購読を許可するか細かく設定が可能

MQTT通信におけるセキュリティ・レベルは、マイコン側で選択するオプションとMQTTブローカ側での設定の組み合わせで決まります。

マイコン側で設定可能な項目は、①経路暗号化(なし、あり)、②認証方式(なし、あり)の2点です。認証方式は、ユーザ名/パスワードによる認証方式と、クライアント証明書による認証方式があります。

端末にどのような権限を与えるかを定める認可は、MQTTブローカ側の設定となります。

本章では、表1に示す5種類のプログラム例を示します。また、セキュリティ・レベルに応じたMQTTブローカの使い分けの例を表2に示します。

プログラム動作環境の構築

今回実験用プログラムの動作検証を行ったハードウェアおよびソフトウェアを表3に示します。