第4章 ラズパイPico 2 W - PC間で実際に通信して不正なメッセージを検出できるか実験する

フルスクラッチ実装で理解する公開鍵暗号のセキュア通信

宮田 賢一

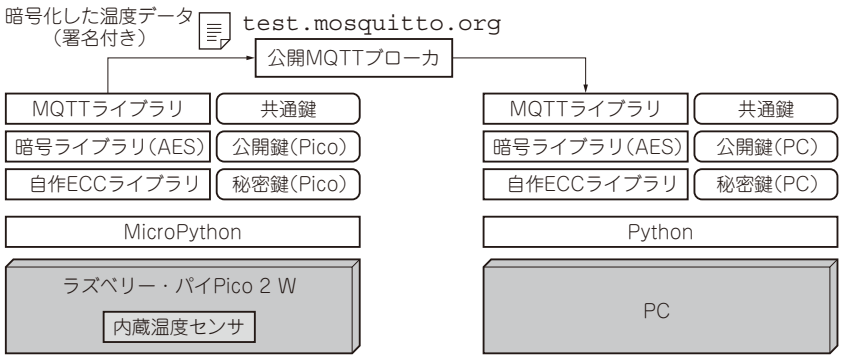

図1
システムの全体構成

表1　実験に使用する機器

項　目	内　容
マイコン	ラズベリー・パイ Pico 2 W（以下 Pico 2 W）
センサ	Pico 2 W 内蔵の温度センサ
送信プロトコル	MQTT
暗号化アルゴリズム	ECC（署名），AES（データ）
実装言語	MicroPython 1.25.0（Pico 2 W）， Python 3.8 以降（PC）

● 本稿で行うこと

RSAと楕円曲線暗号（以下，ECC）の署名処理を比較すると，サイズでも実行時間でも，ECCの方が小さくて済みます（コラム1）．マイコン上で署名処理を行うにはECCが適していそうです．

そこで本稿では，マイコン・ボード上でどこまで実用的なセキュリティ機構をECCで構築できるかを検証します．特に，IoTで多用されるMQTTプロトコルを通じて，デバイスとクラウド（PC）間の通信を保護するユースケースを，実験を通じて体験していきます．ラズベリー・パイ Pico 2 W（以下，Pico 2 W）をデバイスとして用いて，Pico 2 Wの温度センサで取得した温度データを盗聴されないように，セキュリティ保護をかけてPCへ送る実験を行います．

本実験は原理検証を目的とします．そこでクラウド・サービスではなくマイコンと同じLAN内に配置したPCをクラウドに見立てるものとします．

実験構成

本実験で使用する機器を**表1**に示し，システム構成を**図1**に示します．

Pico 2 WとPCとの間の通信データに署名をするためにECCを使用します．今回はECCの実装を確認するために，ECCの処理を自前で実装しました．

データの暗号化にはECCではなく共通鍵暗号の1つであるAES（Advanced Encryption Standard）を使用しています．データの暗号化にECCを使うこともちろん可能です．しかし，公開鍵を用いてある程度長いメッセージを暗号化するのは計算量が多く重い処理であるため，データ本体の暗号化には共通鍵による軽い暗号アルゴリズムを用いることがよく行われています．

設定する暗号

本実験では次のポイントを試します．

▶ポイント①…共通鍵の安全な共有

温度データを共通鍵を用いて暗号化します．共通鍵は事前に生成してPico 2 WとPCが持ち合うのではなく，楕円曲線暗号版のディフィー・ヘルマン鍵共有（以下，DH鍵共有）（コラム2）という方式を用いて共