制御システムのセキュリティ技術

土岐　賢 Satoshi Doki

キーワード IoT，セキュリティ，耐タンパー性

概　要

近年，急速に広がるIoT（Internet of Things）の浸透で，制御システムに対するセキュリティの脅威が年々増加している。インターネットに接続されるIT機器ではセキュリティ対策が進んでいるが，閉鎖されたネットワークで運用されてきた制御システムでのセキュリティ対策は進んでいない。

性能よりも低コスト・低消費電力が重視される制御システムの末端装置に適用できるセキュリティ技術として，ARM Ltd.が提供しているIoT機器向けCPUのCortex-Mには，オプションとしてTrustZoneと呼ばれるセキュリティ機能がある。TrustZoneは，システムをセキュア領域（注1）とノーマル領域（注2）に分割する機能を，単一のCPU（Central Processing Unit）で安価に提供する。セキュア領域に機密データや処理を置くことで，ノーマル領域からアクセスできなくなり，ノーマル領域で制御するネットワークからのIoT機器へのセキュリティ攻撃を防ぐことができる。また，暗号アクセラレータを使用することで制御装置に適用可能な性能で暗号通信への対応が期待できる。

1　まえがき

近年のIoT（Internet of Things）化によって，従来閉鎖されたネットワーク内で用いられてきた制御装置や，フィールドネットワーク機器が公共のネットワークに接続されるようになり，耐タンパー性の向上が必要となっている。耐タンパー性とは，非正規な手段による内部データの読み取り・書き換えを防ぐ能力である。通信データを暗号化しても，末端の装置が攻撃されて暗号通信で用いられる秘密鍵や認証データが漏えいすると，データを窃盗・改変されるおそれがある。**本稿では，IoT機器向けのセキュリティ技術を用いた暗号通信アプリケーションのセキュア処理化を紹介する。**

2　耐タンパー性を備えたCPU（Central Processing Unit）

2.1　Cortex-Aシリーズ

携帯電話のCPUとして，ARM Ltd.が設計したCortex-Aが広く採用されている。Cortex-AにはTrustZoneと呼ばれるセキュリティ機能が搭載されている。TrustZoneは，システムをセキュア領域とノーマル領域に分離し専用命令で切り替える。セキュア領域とノーマル領域が物理的に分離されないため，安価にセキュア機能を実現し，特別な開発環境が不要となっている。TrustZoneでは，ノーマル領域からセキュア領域へのアクセスを制限することで耐タンパー性を確保している。セキュア領域で扱うものとして携帯電話での例を挙げると，機密データでは電子マネー情報，機密処理では認証・暗号処理，ハードウェア機能では指紋センサがある。

第 1 表　TrustZoneの比較

Cortex-AとCortex-MのTrustZoneの比較を示す。

項目	Cortex-A	Cortex-M
切り替え処理	S/W制御	H/W制御
切り替えレベル	OS	関数
切り替え時間	遅い	速い
メモリの分離	メモリ空間が別	メモリ領域

2.2　Cortex-M23/M33

性能よりも低コストや低消費電力が優先されるIoT機器向けCPUとしてARM Ltd.はCortex-Mを提供している。2016年に発表されたCortex-M23/M33にはTrustZoneが搭載された。Cortex-MのTrustZoneは，非力なCortex-Mでも有効に機能するようにCortex-Aとは異なる仕様となっている。**第 1 表**にTrustZoneの比較を示す。

Cortex-MのTrustZoneでは，セキュリティ状態の切り替えをH/W制御で行うことで，切り替えに必要な処理の簡易化と高速化を行っている。また，セキュリティ状態の切り替えを関数レベルとし，さらにセキュア状態によるメモリの区別をアドレス範囲とすることで，セキュア処理化に必要なプログラミング量を大幅に削減している。

3　セキュア処理対応

3.1　開発環境

Cortex-MのTrustZoneを使用したセキュア処理を通信アプリケーションに適用し，耐タンパー性の確認とセキュア処理化の性能への影響を調査した。**第 2 表**に開発環境を示す。コンパイラとRTOS（リアルタイムOS）は，TrustZone命令に対応済みである。プロトタイプ開発用の開発ボードを用いているため，CPUの動作周波数は低くなっている。

3.2　セキュア処理化

既存のアプリケーションをTrustZoneでセキュア処理化した場合の要点を説明する。

(1) プロジェクトの分離　　統合開発環境では，アプリケーションはプロジェクトで管理される。既存のプロジェクトにはノーマル領域での処理を残し，新たなプロジェクトにセキュア領域に置く処理を移動する。セキュアプロジェクトには，機密処理・機密データ・機密データにアクセスする処理が含まれる。

(2) セキュアプロジェクトの初期化処理　　TrustZone機能の初期化処理を追加する。セキュア領域とするメモリ領域，周辺装置を設定する。

(3) 呼び出し可能関数の登録　　ノーマル領域からの呼び出しを可能とするセキュア領域の関数を登録する。登録外の関数の呼び出しは不正アクセスとなる。

(4) ノーマル領域の関数の呼び出し　　セキュア領域から呼び出すノーマル領域の関数ポインタを登録する。

(5) スレッドの属性設定　　スレッドの属性設定は，ノーマルプロジェクトで行う。セキュア領域の関数を呼び出しているスレッド（リアルタイムOSにおける処理単位）作成時にTrustZone属性を設定する。

第 2 表　Cortex-Mの開発環境

Cortex-Mの開発ボードと開発環境を示す。

機能	仕様
開発ボード	ARM Cortex-M Prototyping System（MPS2＋） Cortex-M33 20MHz
統合開発環境	ARM Keil MDK Ploffessional
コンパイラ	ARM Compiler v6（Clang/LLVM）
デバッガ	μVision v5.26.0
RTOS	CMSIS-RTOS2 RTX5
ミドルウェア	IPv4，mbedTLS，Paho MQTT

3.3　通信アプリケーション

セキュア処理の適用対象として，IoT通信で使用されているTCP/IPスタックをベースとしたMQTT（Message Queuing Telemetry Transport）プロトコルによる通信アプリケーションを用意し，標準的な暗号通信プロトコルであるSSL（Secure Socket Layer）に対応させた。**第 3 表**に暗号通信で使用する暗号スイートを示す。暗号スイートとは，暗号通

第 3 表　暗号スイート

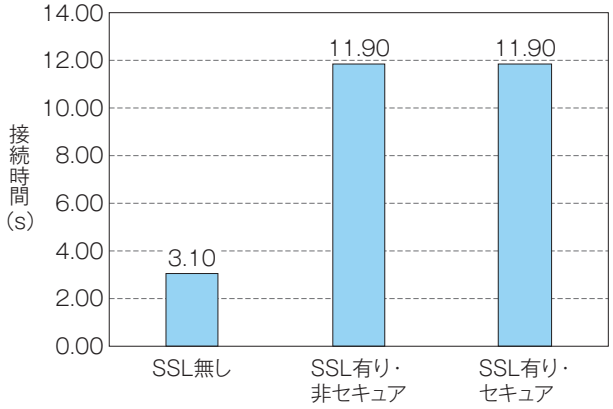
暗号通信における使用する暗号の組み合わせを示す。十分な暗号強度をもつ暗号スイートを選択している。

機能	仕様
プロトコル	TLS（Transport Layer Security）v1.2
鍵交換	ECDHE
公開鍵暗号	RSA 2048bit
共通鍵暗号	AES256-GCM
ハッシュ関数	SHA2-384
公開鍵証明書	X.509 certificate

第 4 表　処理とデータの振り分け

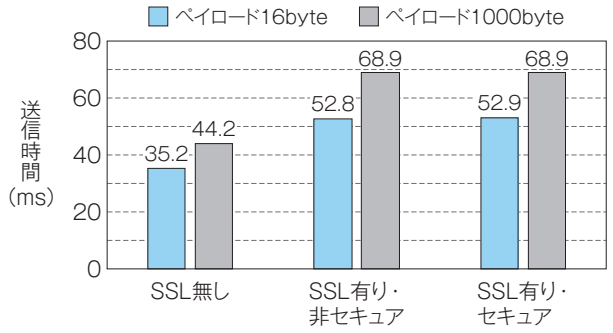
セキュア領域の処理とデータは，ノーマル領域からアクセスできなくなる。

処理／データ	セキュア領域	ノーマル領域
RTOS		○
ネットワーク処理		○
暗号化処理	○	
暗号の秘密鍵	○	
暗号の証明書	○	
暗号化前の送信データ	○	
復号化後の受信データ	○	
暗号化された送受信データ	○	○



第 1 図　MQTTの接続時間

サーバとの接続が完了するまでの時間を示す。暗号化（SSLを有効）で接続時間が大幅に増大している。セキュア処理化の影響は小さい。



第 2 図　MQTTの送信時間

ペイロード（送信サイズ）ごとの送信時間を示す。暗号化（SSLを有効）で送信時間が50%程度増加している。セキュア処理化の影響は小さい。

信に用いる暗号技術の組み合わせである。これに加えて，TrustZoneを使用したセキュア処理対応を行った。**第 4 表**にセキュア領域とノーマル領域での処理とデータの振り分けを示す。ユーザアプリケーション以外で必要になった修正は，MQTTライブラリ内で暗号化処理を呼び出している箇所である。

3.4　評価

ノーマル領域のシリアルポートからセキュア領域内の機密データへのアクセスを試みたが，読み出しができず，不正なアクセスが検出されたことを確認した。

第 1 図にMQTTの接続時間を，**第 2 図**に送信時間を示す。通信を暗号化（SSLを有効）することで，特に接続時間が大幅に増大している。その一方で，TrustZoneによるセキュア処理化の影響は小さいことが分かる。

4　暗号アクセラレータ

暗号処理を高速化する手段として，暗号化をハードウェアで行う暗号アクセラレータがある。Cortex-Aの開発ボードでハードウェアによる暗号アクセラレータの効果を調査した。**第 5 表**にCortex-Aの開発環境を示す。

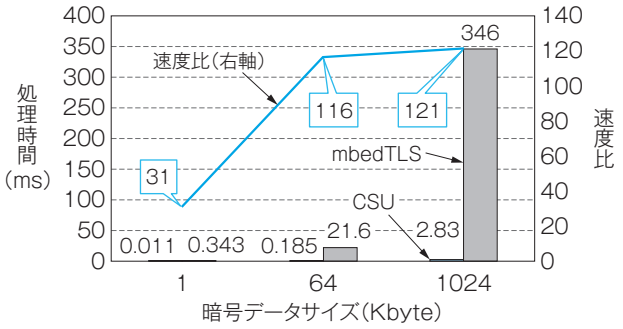
第 3 図に共通鍵暗号の処理時間を，**第 4 図**にハッシュ関数の処理時間を示す。どの場合も暗号アクセラレータを用いることでソフトウェア処理の数十倍の性能となっている。

ARM Ltd.のライセンス供与を受けて，最近各社から出荷され始めたCortex-M23/M33には，いずれも暗号アクセラレータが搭載され，暗号処理の高速化と低消費電力化を実現している。

第５表　Cortex-Aの開発環境

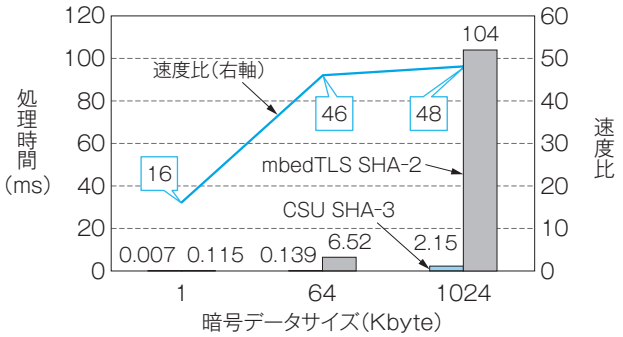
Cortex-Aの開発ボードと開発環境を示す。

機能	仕様
開発ボード	Xilinx Zynq UltraScale＋MPSoC（ZCU102） Cortex-A53 1.5GHz
暗号アクセラレータ	Xilinx Configuration Security Unit（CSU） AES-GCM256，RSA 2048/4096bit， SHA3-384
統合開発環境	Xilinx SDK 2018.1
コンパイラ	GCC 7.2.0
ライブラリ	Xilinx XilSecure（H/W），ARM mbedTLS（S/W）



第３図　共通鍵暗号の処理時間

暗号アクセラレータ（H/W処理）とS/W処理の共通鍵暗号（AES-GCM256）の処理時間の比較を示す。暗号アクセラレータの効果は高い。



第４図　ハッシュ関数の処理時間

暗号アクセラレータ（H/W処理）とS/W処理のハッシュ関数の処理時間の比較を示す。対応するアルゴリズムが異なっているがセキュリティ強度は同等である。

5　むすび

制御システムの末端機器に適用できるセキュリティ技術を紹介した。TrustZoneに対応したCortex-M23/M33のうち，暗号アクセラレータを搭載したものを用いることで，制御装置に適用できる性能で暗号通信対応と耐タンパー性を確保できることを確認した。

・Cortex，TrustZoneは，米国及びその他の国におけるARM Ltd.の登録商標である。
・本論文に記載されている会社名・製品名などは，それぞれの会社の商標又は登録商標である。

（注記）

注1．セキュリティが確保されたメモリ領域
注2．セキュリティが確保されていないメモリ領域

《執筆者紹介》

土岐　賢
Satoshi Doki

製品技術研究所
制御装置の開発に従事