Vol.2021-OS-153 No.6
2021/7/20

DRAM/NVMM と FPGA を統合した セキュアで高信頼なエッジシステムの提案

黄 文康[1]　李 彦志[1]　石綿 陽一[2]　菅谷 みどり[1]

概要：現在，FPGA は低消費電力・高性能なハードウェアアクセラレータとして，エッジコンピューテングシステムでの応用が期待されている．エッジコンピュータはユーザの近距離にあることから，プライバシーに関わる情報も扱うことが期待されるが，情報セキュリティについては十分な議論がなされていない．そこで本研究では，エッジコンピューティングに集約されたデータを，低消費電力・高性能 FPGA により暗号化し，不揮発性メモリに保存することで，統合的にデータの応答性と安全性を実現するシステムを提案する．また，FPGA の暗号化性能を評価し，消費電力が同一レベルのプロセッサより高速・低消費電力で暗号化が可能であることがわかった．

キーワード： FPGA，エッジコンピューティング，暗号化，DCPM

Proposal of a secure and highly reliable Edge system that integrates DRAM / NVMM and FPGA

Wenkang Huang[1]　Yanzhi Li[1]　Yoichi Ishiwata[2]　Midori Sugaya[1]

Abstract: Currently, FPGAs are expected to be applied in edge computing systems as low power consumption and high-performance hardware accelerators.　Since edge computers are close to users, they are expected to handle information related to privacy, but information security has not been fully discussed.　Therefore, in this research, we have created a system that integrates data responsiveness and security by encrypting data aggregated in edge computing with a low power consumption, high-performance FPGA and storing it in non-volatile memory.　suggest.　We also evaluated the encryption performance of FPGA and found that it is possible to encrypt at higher speed and lower power consumption than processors with the same level of power consumption.

Keywords: FPGA，Edge computing，Encrypt，DCPM

1. はじめに

Society5.0 は IoT，ロボット，人工知能などの技術が融合し，人に対して高度なサービスとして提供されることが期待される[1]．これらを実現するために，エッジコンピューティング技術が想定できる．エッジコンピューティングはユーザの近距離に位置し，高性能で応答性が期待される処理に適用される[2,3,[4]．しかし，エッジを構成するハードウエア・ソフトウエアとして何がふさわしいかについては，まだ議論が十分ではない．

そこで，本研究では，実現のための具体例として，「人に優しいロボット」をアプリケーションとし，これらを複数動作させることを想定した場合に，必要となるエッジの構成や要件について議論し，必要となる仕組みについて検討を行った．

我々は，Society5.0 で実用化が期待されるアプリケーションとして，人の気持ちに柔軟に寄り添う革新的なロボットの実用化を検討した．2017 年に提案された，人の気持ちに寄り添う声がけロボットでは [5]，人の生体信号（心拍変動値）から無意識な人の気持ちをモデル化しこれをもとに，ロボットが相手の人の気持ちに応じた振る舞いを提案した[5]．これらのサービスをリハビリや，病床時に行うことで，人の気持ちに応じた自然なコミュニケーションが可能となる．こうした技術は，近年さらに広まっており，生体データをもとにしたロボットの表情の変化[7]や，パーソナルスペースの調整[6]，などに活用されることで，より円滑なコミュニケーションが可能であることを示した．　鈴木らは，こうした生体信号から人の感情モデルや精神モデルを深層学習で構築し，それを病理診断などに活かす手法を提案した[8]．こうしたサービスで扱われるデータには重大な特徴がある．

1 芝浦工業大学 Shibaura Institute of Technology
2 株式会社 Ales, Ales K．K.

ⓒ2021 Information Processing Society of Japan

クレジットカードや，個人情報といった個人の財産に関わる個人情報のみならず，生体データといった人の精神や状態といった私生活に関わるプライバシー(privacy)情報が，人の生活を支援するために大量に利用される．個人情報に関連するセキュリティ問題は多くの研究がなされている一方，こうしたプライバシー情報の保護や安全な利用方法については十分な検討がなされていない． Society5.0 のサービスでは今後，様々なセンサにより人のよりプライバシーに関わる情報が収集され，AI の活用によりより人の意図や気持ち，精神状態にそったサービスを実現されることが期待される中で，我々はこれらのデータの保護を行いつつ，有効なサービスを提供するといった技術の高度な融合を検討する必要がある．

プライバシーに関わる情報を保護しつつ，サービスに対して有効に扱うためには，これらの情報のセキュリティを考慮しつつ，従来のサーバに必要とされる応答性と信頼性を保証する必要がある．情報のセキュリティとは，システムにおいてデジタル化された情報を守ることを目的とし，その情報の CIA である機密性(Confidentiality)，完全性(Integrity),可用性(Availability)を守ることである．これらは，一般的には暗号化技術を用いてユーザ・アクセス認証，データの暗号化などにより支援される．一方，これらの暗号化はオーバーヘッドが大きく，サービスの応答性を維持しつつ，オーバーヘッドを低減させる方法を検討することが求められる．

本研究では，プライバシーに関わる情報を保護しつつ，データの応答性と信頼性を高めるための新しいシステムとして，DRAM/NVMM と FPGA を統合したセキュアで高信頼なエッジシステムの提案を行う．エッジコンピューティングでは，デバイスに近いエッジ側に存在するセンサノードに対し，近距離でのリアルタイム応答を目的とし，大量のセンサノードに対し応答性を重視した設計が提案される[9]．クラウドとクライアントマシンであるロボットの間に，豊富な計算資源を持つエッジサーバを配置することで，クラウドサーバに集中していた負荷の分散や，ネットワーク通信による遅延や不安定性の解消をすることが可能となる[9]．先の例に示したように，人の内部状態を利用する主なサービスはパーソナルなサービスであり，ロボットなどが利用する高度なリソースを，近距離に配置することは合理的な選択である． 我々は，さらに，エッジサーバが本来満たすべき要求である応答性，信頼性を担保するために，まず，memcached[10]を NVMM に適応させた fogcached[11] を用いたシステムを構成するものとした．これにより，応答性，信頼性を向上させる．さらに，データの機密性のセキュリティを確保するために，FPGA に注目した．現在，FPGA は低消費電力・高性能なハードウェアアクセラレータとして，エッジコンピューテングシステムでの応用が期待されている．本研究では，高度に秘匿性が求められる情報のセキュリティを実現するために用いることを検討する．FPGA は自由にセキュリティを考慮した回路を作成し，情報セキュリティにおける機密性を確保することができる．また，FPGA は一方向での処理に強く，暗号化は FPGA のメリットである性能を生かしつつ，目的となる情報セキュリティを両立させることができると考えられる．

本ミドルウエアを構成するために，すでに fogcached[11]や，ロボットサービスを実現するアーキテクチャである，ROS[12][13]は実装が存在しているが，これらを統合する仕組み，また，セキュリティを実現するための FPGA の統合，暗号化性能は十分に評価されていない．そこで，本研究ではまず，次世代サービスに向けた ROS と FPGA サーバを含めた統合の全体アーキテクチャを提案する．次に，FPGA の暗号化システムの提案と評価を実施した．その結果，消費電力が同一レベルのプロセッサより 53%短時間で暗号化できることを確認した．

本論文の構成は，次の通りである．まず 2 節にて関連研究について述べる．次に 3 節にて，データの応答性と安全性を実現するシステムを提案する．そして，4 節に予備実験と予備実験結果． FPGA で暗号化を加速する方法を提案した．次に，5 節では，提案したシステムを評価するため，実験を行なった．実験はこのシステムは短時間で暗号化できることが確認した．さらにこの結果に基づき，第 6 節にて，まとめと今後の課題．

2. 関連研究

ROS（Robot Operating System）はロボットアプリケーション向けのミドルウエアである[12][13]．ROS は Publisher / Subscriber という通信フレームワークを提供し，統一的な通信を可能としている．近年，多くのロボットなどは，ROS システムを用いて研究されている[9]．

小沢らは Intel 社が提供する不揮発性メインメモリである DCPM を利用し，エッジサーバにおいて，応答性と信頼性を両立させる fogcached を提案した[11]．fogcached は，memcached[15]を拡張し，RAM と NVMM(DCPM)からなるハイブリッド型メインメモリ上で In-Memory Key-Value Store を展開し，エッジサーバの応答性を向上させた．

FPGA（Field Programmable Gate Array）は PLD の一種であり，製造後に内部倫理回路を定義し変更できる集積回路として，近年は安全性やセキュリティについても様々な提案がなされている[16][17]．しかし，暗号化アプリケーションとしての利用はまだ十分ではない．

FPGA の計算力を利用するためには，幾つか方法がある．一つは，CPU の演算処理の補助である．これにより，CPU の負荷を軽減する．実現には， FPGA を CPU に繋ぐ必要があることから， PCIe か MMIO で接続する方法をとることが一般的に行われる．XILINX と ALTERA は，この二つのインターフェースに対応する FPGA を提供している．また，

ⓒ2021 Information Processing Society of Japan

FaaS（FPGA-as-a-Service）と呼ばれる，クラウド上で FPGA を使用する方法も提案されている． Amazon 社からは, EC2 F1 インスタンスが提供されている[14]．しかし，FPGA はクラウドやエッジサーバでの応用方法は十分提案されていない．モナメッドらは，OpenSSL 専用のアクセラレータとして FPGA の実装を示した[18]．しかしモナメッドらの実装は組み込み FPGA での実装であり，アプリケーション向けではない． 大川らでは FPGA をネットワーク専用アクセラレータとして利用することを提案した[9][19][20]．大川らの提案では，FPGA で ROS のネットワークトラフィックを加速し，応答性を向上することができる利点がある．しかし，これらの研究は主に FPGA 自身を発展させる研究であり，具体的な利用を想定したアプリケーションをエッジとクラウドに連携させて，利用するための仕組みや研究はまだ十分ではない．

3. 提案

3.1 目的

本研究では，DRAM/NVMM と FPGA を統合したセキュアで高信頼なエッジシステムの提案を行うことを目的とする．目的を実現するために，まず，ROS を統一的なミドルウエアと定め，この通信を基礎とした汎用的なシステムを構想する．本システムにおいては，近距離サーバとして応答性と信頼性を向上させるために fogcached によるデータの送受信および，NVMM の利用による信頼性の向上を実現する[9]． さらに，暗号化を行うミドルウェアを部分の設計と実装について検討する．ミドルウェアは FPGA を用いて暗号化を行う部分を構成する．生体データなど，プライバシーを含むデータを保存する前に，本ミドルウェアによる暗号化を行う．

全体アーキテクチャについてミドルウエアでのサービス統合にあたり，本研究では ROS の仕組みを拡張する．実現にあたり，検討した内容を次の(A),(B),(C) にまとめた．

(A) プライバシーデータの暗号化時の影響が不明．

(B) 高性能 FPGA と CPU 暗号化性能比較が十分ではない．

(C) プライバシーデータをエッジへ保存する方法の検討．

本三つの課題に対し，本研究では，プライバシーデータである生体データを高速に暗号化して保存し，エッジサーバの高速性と信頼性を実現する方法を検討した．

本研究では FPGA を用いる生体データを暗号化保存するシステムについて，まず，提案(1)として 課題(A)に対し，暗号化アプリケーションがある場合，システムのスループットの影響を評価する．提案(2)として課題(B)(C)に対し，FPGA と CPU は暗号化性能を比較し，生体データを暗号化保存するシステムとして設計・実装し，評価を実施した．

3.2 提案システム

本提案は ROS[15]をミドルウエアの基礎として用いたシステムとし, FPGA を統合したセキュアで高信頼なエッジミドルウエアである．データのセキュリティと高性能・低消費電力の実現のための FPGA を用いる．提案システムの全体図を図 1 に示す．データの通信図は図 2 に示す．

提案システムでは，クライアントアプリケーション，エッジサーバの構成とする．エッジサーバには，データベース，エンコーディング処理のためのサーバを配置し，これらのサーバがエッジ内で ROS のインターフェイスで通信し，コンポーネントとして連携して動作する．クライアント側のアプリケーションは ROS アプリケーションを使用して構築する．クライアントとエッジでは ROS アプリケーションとして SLAM(Simultaneous Localization and Mapping)と生体センサから取得される情報を用いたサービスを提供するものとした． SLAM はセンサデータから周辺の空間の地図の作成とロボットの位置の特定を行う技術であり，多くのロボットで用いられている．このことから，汎用的なアプリケーションとして利用を想定した．エッジゲートは ROS のデータを処理する．また，エッジサーバは，エンコーディングサーバ(FPGA)，高信頼データ保存と高速な応答のための KVS を配置した Edge Server，と Database サーバで構成する．Edge Server では， fogcached をインターフェイスとして，ROS の通信をサポートした fogcached-ros [9]を用いる．これにより DRAM/NVMM(DCPM)を利用し，高い応答性と，データの信頼性を保証する．NVMM のデータのうち，長期保存に適したデータは，データベースへ移行する．

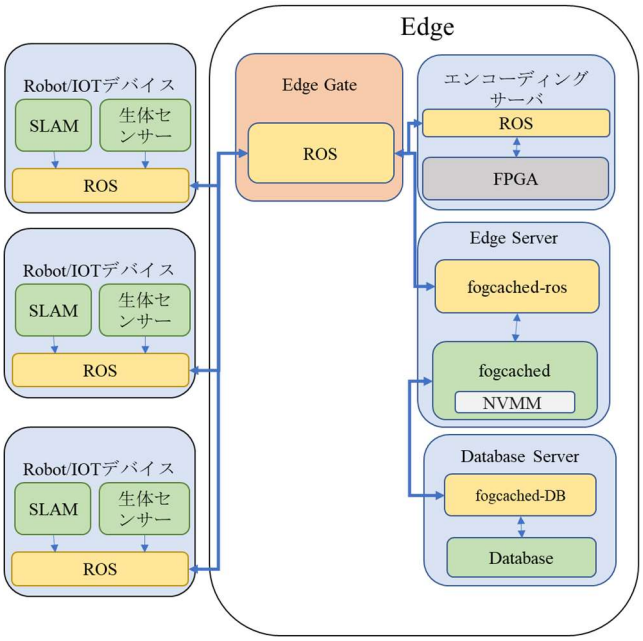

図 1 システムの全体図

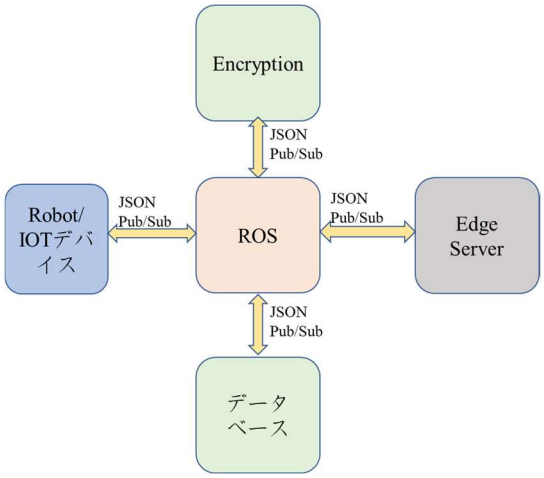

図 2 データの通信図

3.3 ROS (Robot Operating System)

ROS (Robot Operating System)[12][13]は分散通信フレームを提供するミドルウエアで，通信モジュールを使用して，モジュール間に P2P ネットワーク接続と実装するアーキテクチャである．複数ノードの通信支援を行っているが，リアルタイム性能などは考慮していない．本研究では ROS の通信システムをついて，プライバシー情報である生体データの伝送方法を設計するものとした．

ROS には，(a)node, (b)master, (c)message, (d)topic, (e)publish, (f)subscribe，という用語がある．(a)Node は，ROS クライアントライブラリを使用し，他の Node と通信するプロセスである．SLAM をはじめとしたアプリケーションは通常多くの node で構成される．(b) Master は ROS 全体の実行の中心となり， node の登録，サービス，topic 名の登録や，パラメータサーバの維持を行う (c)message は通信のデータ，(d)topic は，共有されるデータであり，それぞれ独立している．(e),(f)について，ROS は Pub/Sub 通 信を行う．

3.4 FPGA (Field Programmable Gate Array)

本研究では FPGA (Field Programmable Gate Array)を用いる．FPGA はプログラマブルデバイスであり，従来の論理回路やゲートアレイと比べ構造が異なる．今回使用した FPGA は PYNQ-Z1 である．PYNQ はオープンソースフレームワークである．PYNQ を利用して，ロジック回路を設計しなくても，PYNQ-Z1 に載せた機能を十分に利用できる利点がある．高位合成 HLS（High-level Synthesis）は，C/C++ コードから FPGA の回路変換するプロセスとした．本研究では，Zynq を使用してプログラマブルロジックとマイクロプロセッサを統合し，FPGA 暗号化サーバを構築するものとした．このように，FPGA は，暗号化アルゴリズムで高速に計算するのに適していることから，本研究では FPGA を使用して暗号化システムを構築する．

3.5 システムの設計

本研究で提案するシステムにおいては ROS を拡張し，エンコーディングサーバを通じてデータを暗号化し，エッジサーバにデータを fogcached-ros を通じてキャッシュし，データベースサーバと暗号化データを保存するものとした．次に基本的な構成の詳細について述べる．

- クライアントアプリケーション: ROS システムに接続したロボットとする．ロボットにはプライバシー情報である個人の生体データを収集する心拍・脳波センサと，SLAM で用いる LIDAR が装備されていることを想定する． クライアントアプリケーションは，pub/sub 通信を通じて ROS がインストールされたエッジサーバと通信し，データを転送する．
- エッジサーバ: fogcached-ros [9]をインターフェイスとして，NVMM(DCPM)を利用できるものとする．本サーバ上では，データキャッシングおよびデータ処理サービスを提供する．エッジサーバとなる IoT や各種端末デバイスの近くにエッジサーバを置くことで，データの保存と処理を行う．

本システムにおいては， ROS と Node の通信は，publisher/subscriber を基礎として行う． Topic はデータを区分して， データを共有メモリに非同期通信を行う．fogcached[11]と ROS の接続部分は，fogcached-ros を用いる．これにより，ROS システム全体の応答性能と信頼性の向上を実現させる．

生体データと SLAM データは， エッジサーバ の fogcached-ros を通じて，fogcached （DRAM・DCPM）にキャッシュされる． データは fogcached に保存する．

同時に Topic 中のプライバシーデータはエンコーディングサーバに転送して暗号化する．暗号化されたデータは，再度 ROS に転送され，Edge Gate とエッジサーバを通じて，データサーバに送信される．また，データの暗号化後は，fogcached-ros の中の暗号化されていないデータを削除する．データサンプルを図 3 に示す．図 4 は ROS Topic とデータフローを示す．

```
[{"Time": "2021-2-14 15:22:17"},
  {"Real altitude": "-21．85"},
  {"Pressure": "100930"},
  {"Humidity": "37．00"},
  {"Temperature": "20．13"},
  {"Luminance": "473"},
  {"MQ3": "453"},
  {"db": "41"},
  {"heart": ” 486"}]
```

図 3 データフローのデータサンプル

ⓒ2021 Information Processing Society of Japan

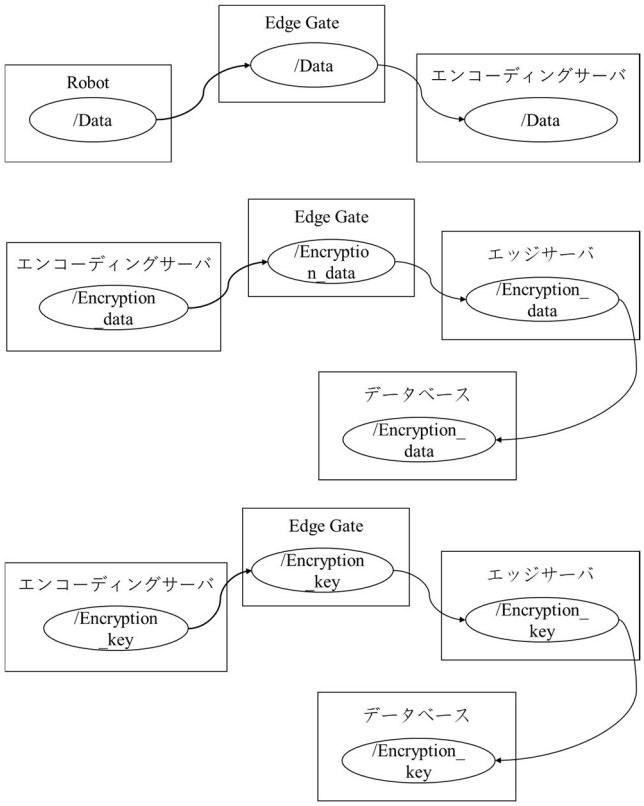

図 4 暗号化のデータフロー

4. 予備実験

4.1 目的

関連研究に示したように，本研究ではプライバシー情報である生体データを，保存する前に暗号化することを目的としている[22]．本提案システムに示したように，暗号化されたデータは，データのセキュリティを確保するためにROS を通じて DCPM へ保存することを提案システムで述べた．

本予備実験では，暗号化した場合のシステムのスループットと，暗号化していない場合のシステムのスループットを比較する．これにより，システムのスループットの低下を調査し，システムの有効性を議論する．予備実験を行うための実験システムを設計した．システムは Raspberry Pi 3B を用いた．

4.2 概要

実験に使用した機材，またコンピュータのリソースと実行した ROS アプリケーションを表 1 に示した．

OS	Memory	PC	Environment
Ubuntu 18.04	1GB	Raspberry Pi 3B	暗号化アプリケーション
Ubuntu 18.04	1GB	Raspberry Pi 3B	

表 1 コンピュータのリソースとタスク

ロボットと生体センサにプライベートデータを生成し，サーバに送信する．また，システムのスループットに対して，暗号化アプリケーションを検証する．予備実験の設計図を図 5 に示した．

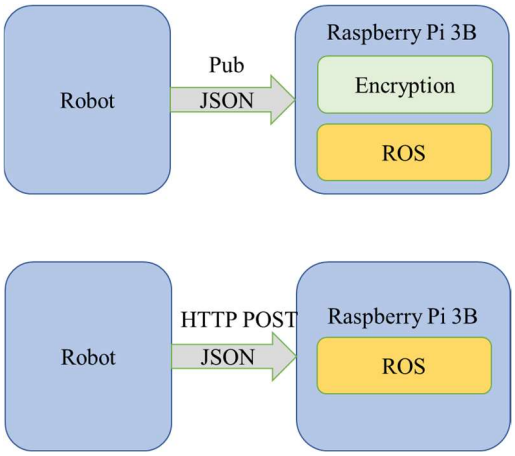

図 5 予備実験の設計

4.3 評価と結果

予備実験の結果を表 2 と図 6 に示した．高 CPU 負荷のスループットと低 CPU 負荷のスループットを比較した． 高 CPU 負荷は，暗号化アプリケーションを実行している．低 CPU 負荷は，システムはアイドル状態，暗号化アプリケーションを実行していない．

表 2 から，高 CPU 負荷のスループットは，低 CPU スループットの負荷と比べて 170%遅い結果となった．

また，高 CPU 負荷の最大値は，低 CPU 負荷の最小値よりも小さくなった．このことから，システムが暗号化作業を完了することを支援するために，対応する暗号化モジュールを導入する必要があると考えられる．専用の暗号化モジュールを導入することにより，システムの負荷が軽減すると考えられる．

	高 CPU 負荷 スループット(KB/S)	低 CPU 負荷 スループット (KB/S)
平均値	0.52713	1.42458
標準偏差	0.011514633	0.011566792
最小値	0.5151	1.4067
最大値	0.5498	1.4431

表 2 予備実験結果

ⓒ2021 Information Processing Society of Japan

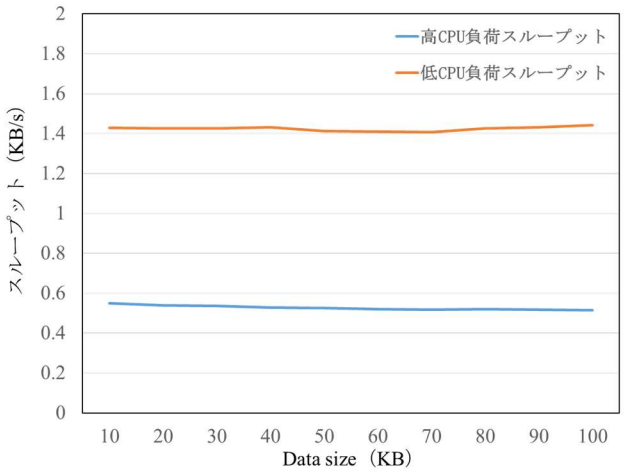

図 6 予備実験結果

5. FPGA による暗号化性能の評価

5.1 目的

予備試験では，システムスループットの評価を高 CPU 負荷と低 CPU 負荷にて実施し，いずれもシステムのスループットが低下することを確認した．このことから， CPU の負荷軽減のために，暗号化モジュールを導入する．

5.2 AES による暗号化方式

暗号化技術は，大まかに二つの種類に分類できる．一つ目は，同じ鍵を持っていれば誰でも復号できる対称暗号と，二つ目は暗号用鍵復号用鍵が，別々である非対称暗号である．今回の提案は「人にやさしいロボット」を想定した基盤システムである．それを実現するため，情報を複数ノード間での共有を考慮することがあると考えた．

共通鍵暗号方式は送信側と，受信側は平等であることから，今回想定した複数のエッジが互いに送信する場合に適している．また，AES 暗号したデータは fogcached に保存する．AES 暗号化方式は，鍵を共有することで他のデバイスとデータを共有することができる．エッジの特徴においては，データは，エッジ間を遷移することが想定されることから，保存されたデータは，復号せずに別のエッジに送付することも必要となる．

FPGA は単一で，何回も重複した作業に優れている．暗号化用アルゴリズムはこのような重複した計算が多く，FPGA で AES 暗号化することは合理的であると考えられる．また，AES-ECB の暗号化方式は並列コンピューティングに適している．これらにより， AES-ECB 暗号化方式を使用するものとした．AES のコードの記述例を図 7 に示した．このコードでは，AES-ECB のキーをセットアップした後，AES 暗号化処理を行っている．SubBytes 関数はデータを暗号化関数．ShiftRows 関数は左にシフトする．MixColumns 関数は AES 状態行列のマトリックスを混合する関数．Cipher はメインの関数，データを処理する．キー入力して，SubBytes 関数にデータを入力し、マトリックスをセットする．そして、ShiftRows 関数はデータを左にシフトして、MixColumns 関数はマトリックスにデータを混合して、データを AES 暗号化する．

```
static void SubBytes(state_t* state)
{
  uint8_t i, j;
  for (i = 0; i < 4; ++i)
  {
    for (j = 0; j < 4; ++j)
    {
      (*state)[j][i] = getSBoxValue((*state)[j][i]);
    }
  }
}
static void ShiftRows(state_t* state)
{
  uint8_t temp;
  temp           = (*state)[0][1];
  (*state)[0][1] = (*state)[1][1];
  (*state)[1][1] = (*state)[2][1];
  (*state)[2][1] = (*state)[3][1];
  (*state)[3][1] = temp;
  temp           = (*state)[0][2];
  (*state)[0][2] = (*state)[2][2];
  (*state)[2][2] = temp;
  temp           = (*state)[1][2];
  (*state)[1][2] = (*state)[3][2];
  (*state)[3][2] = temp;
  temp           = (*state)[0][3];
  (*state)[0][3] = (*state)[3][3];
  (*state)[3][3] = (*state)[2][3];
  (*state)[2][3] = (*state)[1][3];
  (*state)[1][3] = temp;
}
static void MixColumns(state_t* state)
{
  uint8_t i;
  uint8_t Tmp, Tm, t;
  for (i = 0; i < 4; ++i)
  {
    t   = (*state)[i][0];
    Tmp = (*state)[i][0] ^ (*state)[i][1] ^ (*state)[i][2] ^ (*state)[i][
3] ;
    Tm  = (*state)[i][0] ^ (*state)[i][1] ; Tm = xtime(Tm);  (*state)[i
][0] ^= Tm ^ Tmp ;
    Tm  = (*state)[i][1] ^ (*state)[i][2] ; Tm = xtime(Tm);  (*state)[i
][1] ^= Tm ^ Tmp ;
    Tm  = (*state)[i][2] ^ (*state)[i][3] ; Tm = xtime(Tm);  (*state)[i
][2] ^= Tm ^ Tmp ;
    Tm  = (*state)[i][3] ^ t ;              Tm = xtime(Tm);  (*state)[i]
[3] ^= Tm ^ Tmp ;
  }
}

static void Cipher(state_t* state, const uint8_t* RoundKey)
{
  uint8_t round = 0;
  AddRoundKey(0, state, RoundKey);
  for (round = 1; ; ++round)
  {
    SubBytes(state);
    ShiftRows(state);
    if (round == Nr) {
      break;
    }
    MixColumns(state);
    AddRoundKey(round, state, RoundKey);
  }
  AddRoundKey(Nr, state, RoundKey);
}
```

図 7 AES のコード

ⓒ2021 Information Processing Society of Japan

5.3 FPGA と CPU の利用

予備試験では，高 CPU 負荷のスループットと低 CPU 負荷のスループットを比較した．予備試験の結果より，高 CPU 負荷のスループットが低いことがわかっている．このことから，暗号化モジュールを導入する必要があると考えた．本実験は FPGA で暗号化モジュールについてプライベートデータを暗号化して保存するシステムを提案する．使用したエンコーディングサーバは Raspberry Pi 3B と PYNQ-Z1 とした．今回はセキュアな処理を FPGA での暗号化により実現する方法を設計・実装し，評価する．使用するコンピュータのリソースを表 3 に示す．実験の設計を図 8 に示した．

OS	Memory	Hardware
Ubuntu 18.04	1 GB	Raspberry Pi 3B
Ubuntu 18.04	512MB	PYNQ Z1

表 3 コンピュータのリソース

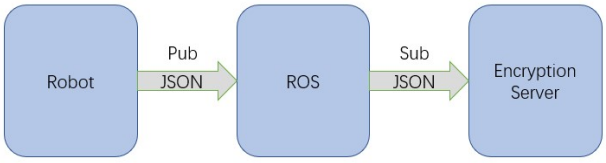

図 8 実験の設計

5.4 評価方法

評価においては，ROS システムを介してデータを topic として作成し，publish する．次に,データを ROS に介して暗号化サーバに subscribe する．FPGA と CPU をテストすることにより,同一データ量で暗号化の速度,また，暗号化時の FPGA と CPU の消費電力量を比較する．

5.5 評価結果

(1) 暗号化速度

各データ量の暗号化速度を図 9 に示す. FPGA の暗号化速度は 250.15(KB/s)，Raspberry Pi 3B の CPU は 471.92(KB/s)かかった．平均，標準偏差，最大，最小は表 4 に示す．

	Raspberry Pi 3B	FPGA
平均値(KB/s)	250.15	471.92
標準偏差	3.95	5.34
最小値(KB/s)	242.93	460.27
最大値(KB/s)	255.15	480.15

表 4 暗号化速度の結果

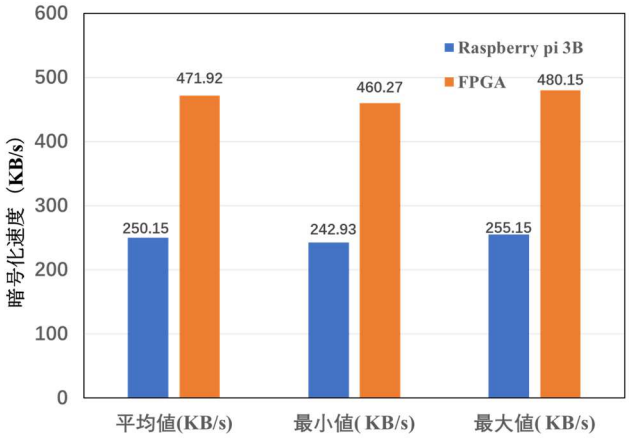

図 9 暗号化速度の結果

表 4 と図 9 の結果から，FPGA の暗号化速度は Raspberry Pi 3B に比べて 1.8 倍速いことがわかった．

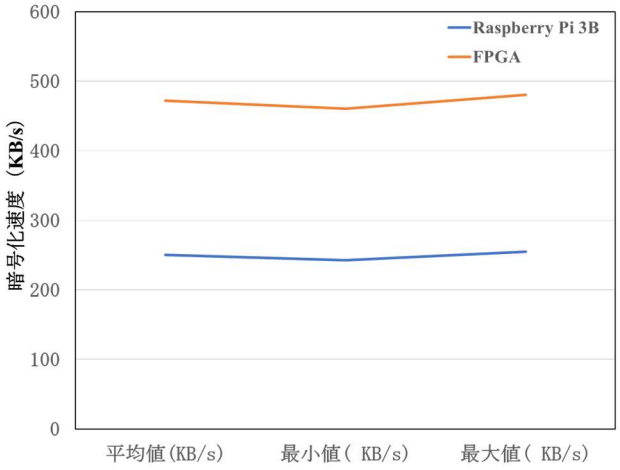

図 10 各データ量の暗号化速度

次に，Raspberry Pi 3B と FPGA について，異なるデータ量で比較した時の暗号化速度の変化ついて図 10 にまとめた．結果より，Raspberry Pi 3B と比較して，FPGA システムの暗号化速度が 1.8 倍速いことがわかった． また，標準偏差から FPGA システムの暗号化速度の変動が 1.35 倍大きいことがわかった． このことから，CPU と FPGA 間のデータ転送は I / O 操作であるため，メモリコピーにとして遅延があることがわかった．ばらつきのある遅延がスタッキングし，システムの標準偏差は大きくなる． ただし，遅延が発生する場合もあるが，FPGA の最悪の速度は Raspberry Pi 3B よりも優れていることがわかった． このことから，FPGA が Raspberry Pi 3B よりも暗号化に適していると考えられる．

(2)消費電力の比較

Raspberry Pi 3B と FPGA の暗号化時の消費電力を比較した結果を図 11 に示した．図 12 の結果から，FPGA システムが暗号化するときに消費する電力が 3.4 倍であり少な

ⓒ2021 Information Processing Society of Japan

いことがわかる．同時に，FPGA が大量のデータを暗号化処理している場合，消費電力の増加は Raspberry Pi 3B よりも 380%低くなる．FPGA は低消費電力で大量のデータを暗号化できる利点があることから，こうした要求に対応することができると考えられる．

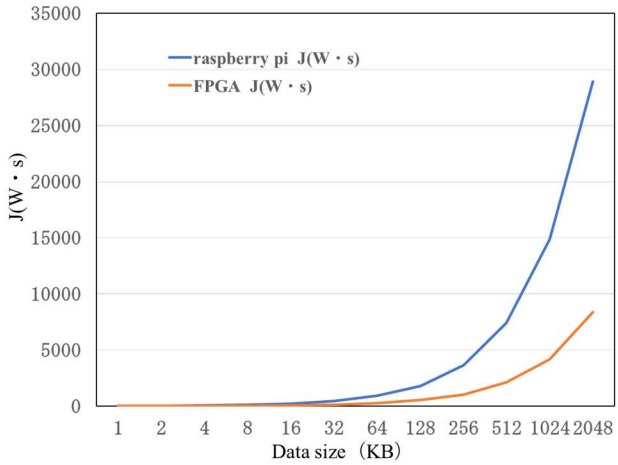

図 11　FPGA と Raspberry Pi 3B の消費電力

6. まとめと今後の課題

本論文では，ROS ベースの FPGA/NVMM を統合した高信頼なエッジミドルウエアの提案を行った．また，その一部である個人情報の安全性を重視した生体データをエッジに保存するシステムを提案した．提案では，FPGA を用いる生体データを暗号化保存する方法を実装・評価を行った．評価の結果では，FPGA の方は CPU より暗号化の性能が優れている．消費電力の面からも，FPGA の方は消費電力が低いことが分かった．

今後の課題として，システムの暗号化速度を高め，複数の FPGA の FIC（Flow-in-Cloud）[23][24]を使って，システムを実行することを考える．FIC（Flow-in-Cloud）はマルチテナントを実現して複数の FPGA 間を直接シリアルリンクの FPGA システム．複数の FPGA を使用して直接シリアルリンクすることにより，FPGA のリソースをよりよく利用することができる．また，システムにロボット SLAM を追加することを考えられる．そして，このシステムにディープラーニングを追加することを考えられる．

謝辞
本研究は JST, CREST, JPMJCR19K1 の支援を受けたものです．ここに感謝いたします．

参考文献

[1] Society5.0：https://www8.cao.go.jp/cstp/society5_0/
[2] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322-2358, Fourthquarter 2017, doi: 10.1109/COMST.2017.2745201.
[3] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.
[4] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657-1681, thirdquarter 2017, doi: 10.1109/COMST.2017.2705720.
[5] Teppei Ito, Reiji Yoshida, Yoshito Tobe, Midori Sugaya, Supportive Voice-Casting Robots using Bio-Estimated Emotion for Rehabilitation, The 15th International Conference on Intelligent Environments 2019, June 24-27, 2019, Rabat, Morocco
[6] Suzuki, Muhammad Nur Adilin Mohd Anuardi, Peeraya Sripian, Nobuto Matsuhira, Midori Sugaya, Multi-user Robot Impression with a Virtual Agent and Features Modification According to Real-time Emotion from Physiological Signals, The 29th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN 2020), Virtual Conference, Aug. 31st – Sep. 4th, 2020.
[7] Peeraya Sripian, Yuya Kurono, Reiji Yoshida, Midori Sugaya, “Study of Empathy on Robot Expression Based on Emotion Estimated from Facial Expression and Biological Signals”, Proceedings of 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2019), Le Meridien, Windsor Place, New Deli, India, Oct, 14th-18th, 2019.
[8] 鈴木圭，松原良太，菅谷みどり，脳波指標と心拍変動指標による感情推定モデルの構築とその評価，BioX,CNR 研究会 BioX 一般セッション，オンライン 2021 年 3 月 2 日
[9] Koki Higashi, Yoichi Ishiwata, Takeshi Ohkawa and Midori Sugaya, fogcached-ros: Hybrid main memory KVS server, Asia Pacific Conference on Robot IoT System Development and Platform (APRIS2020), 202011, online
[10] “memcached” http://memcached.org/ (参照 2020-08-04)
[11] Kouki Ozawa, Takahiro Hirofuchi, Ryousei Takano, Midori Sugaya, “fogcached: DRAM-NVM Hybrid Memory-Based KVS Server for Edge Computing”, Edge Computing – EDGE 2020 – 4th International Conference, Lecture Notes in Computer Science 12407, Springer, 2020, pp. 50-62
[12] Lum,. S. . UtilizingRobotOperatingSystem(ROS)inRobotVision and Control . National Technical Reports Library U . S Department of Commerce. 2015.
[13] “ROS”, https://www.ros.org/ (参照 2020-08-04)
[14] Amazon Web Services: “Amazon EC2 F1 Instance”, https://aws.amazon.com/j p/ec2/instance-types/f1/.
[15] Y. Nitta, S. Tamura, H. Yugen and H. Takase, "ZytleBot: FPGA Integrated Development Platform for ROS Based Autonomous Mobile Robot," 2019 International Conference on Field-Programmable Technology (ICFPT), Tianjin, China, 2019, pp. 445-448, doi: 10.1109/ICFPT47387.2019.00089.
[16] S. K. R, S. R, M. A. M, P. K. M. S and R. M, "Design of High Speed AES System for Efficient Data Encryption and Decryption System using FPGA," 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Msyuru, India, 2018, pp. 1279-1282, doi: 10.1109/ICEECCOT43722.2018.9001535.
[17] K. Kumar, K. R. Ramkumar and A. Kaur, "A Design Implementation and Comparative Analysis of Advanced Encryption Standard (AES) Algorithm on FPGA," 2020 8th International Conference on Reliability, Infocom Technologies

ⓒ2021 Information Processing Society of Japan

and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 182-185, doi: 10. 1109/ICRITO48877. 2020. 9198033.
[18] https://ieeexplore. ieee. org/document/5416065
[19] Takeshi Ohkawa, Kazushi Yamashina, Hitomi Kimura, Kanemitsu Ootsu, Takashi Yokota: FPGA components for integrating FPGAs into robot systems, IEICE Transactions on Information and Systems. E101. D. 363-375. 10. 1587/transinf. 2017RCP0011. (2018)
[20] Yuhei Sugata, Takeshi Ohkawa,Kanemitsu Ootsu Takashi Yokota: Acceleration of publish/subscribe messaging in ROS-compliant FPGA component, Proc. of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies. ACM, 2017.
[21]"Intel®OptaneTMPersistentMemory". https://www. intel. com/content/www/us/en/architecture-and-technology/optane-dc-persistent-memory. html (参照 2020-08-04)
[22] 2014 年度情報セキュリティ事象被害状況調査 https://www. ipa. go. jp/security/fy26/reports/isec-survey/index. html
[23] 弘中和衛, 山倉美穂, 天野英晴, "マルチ FPGA システムにおける部分再構成の実際", 信学技報, vol. 120, no. 36, RECONF2020-16, pp. 85-90, 2020 年 5 月.
[24] Kazusa Musha, Tomohiro Kudoh and Hideharu Amano, "Deep Learning on High Performance FPGA Switching Boards: Flow-in-Cloud", Proc. of the International Simposium on Applied Reconfigurable Computing (ARC), 2018.

ⓒ2021 Information Processing Society of Japan