

SoC FPGAアーキテクチャを用いた ReconfigurableなIoT機器のためのセキュリティ機構の提案

学習駆動コース フィジカルゼミ・室蘭工業大学 灰原 渉

背景

➤ SoC FPGAはIoTに向いている

● SoC FPGA = FPGA + CPU

SoC FPGAは、「内部回路を再構成可能」なFPGAとCPUが繋がれたアーキテクチャ。CPU上のアプリケーションから、FPGAの内部回路を再構成したり、FPGAの内部回路へ入出力したりすることが可能。

● ハードウェアアクセラレーション

ソフトウェアでは時間のかかる処理を、ハードウェアに委託することで省電力かつ高速な演算が可能。用途に合った回路をFPGAに実装すれば、CPUの処理をアクセラレートできる。
(ディープラーニングや画像処理に応用されている。)

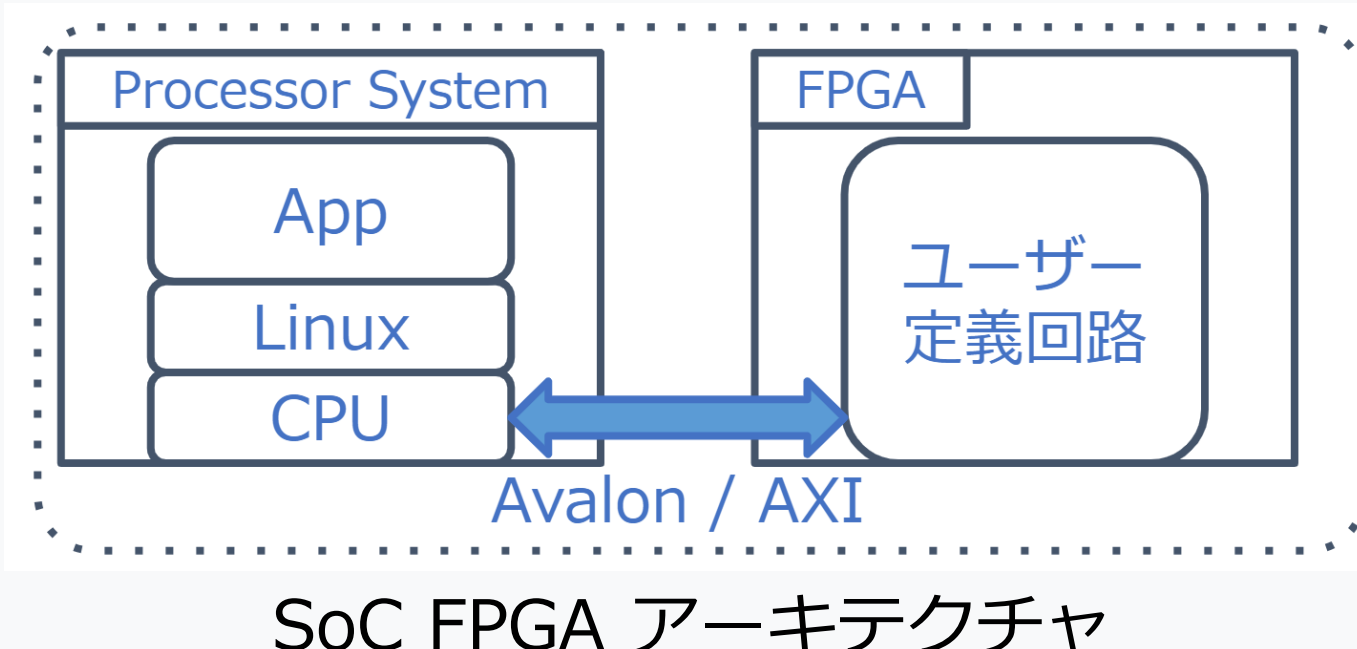
● Reconfigurable System

CPU上のアプリケーションを稼働させたままFPGAを再構成することでCPUからは周辺機器が変わったように見える。またCPUがネットワークに接続されていれば、遠隔地からFPGAを再構成することも可能。

● IoT機器とデータ処理

機械学習や高速なインターネット接続が台頭する今日では、IoT機器が扱うデータ量も増えると予測。しかし、IoT機器で使われることの多い省電力なCPUの処理能力は大きくない。

⇒ SoC FPGAがIoT機器に効率的な演算能力をもたらすのではないか



SoC FPGA アーキテクチャ

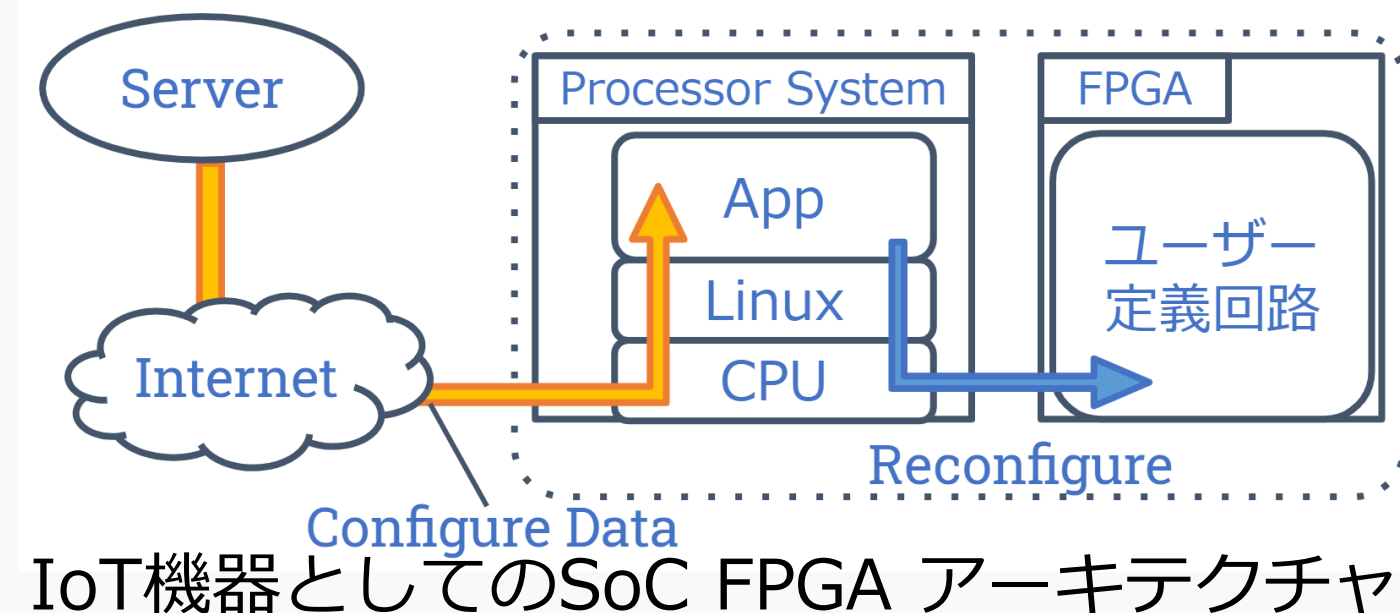
➤ IoT x SoC FPGA のセキュリティ

● 先行研究が少ない

IoT x SoC FPGA のセキュリティについて研究している事例は少ない。

● IoT機器とセキュリティ

IoTシステムは、多数のノードから構成されることが多いことから、IoT機器には単価の安さが求められる。
→ ローエンドなデバイスで十分なセキュリティを実現する必要がある。



IoT機器としてのSoC FPGA アーキテクチャ

脅威の分析 (Reconfigurable System)

➤ 実装する回路を盗聴されるリスク

● Configuration Data の盗聴

サーバーからインターネットを通じて送られるConfiguration Data が盗聴され、実装予定の回路が流出してしまうリスク。

● Configuration Data

どのような回路をFPGA に実装するかが記述されたファイル。

● Configuration Data の暗号化によって対策可能

➤ 攻撃者に回路を実装されるリスク

● Configuration Data の改ざん

サーバーからインターネットを通じて送られるConfiguration Data が改ざんされ、攻撃者によって、FPGAに任意の回路を実装されるリスク。

● Configuration Data の認証によって対策可能

ベンダーの対策 (Intel SoC FPGAの場合)

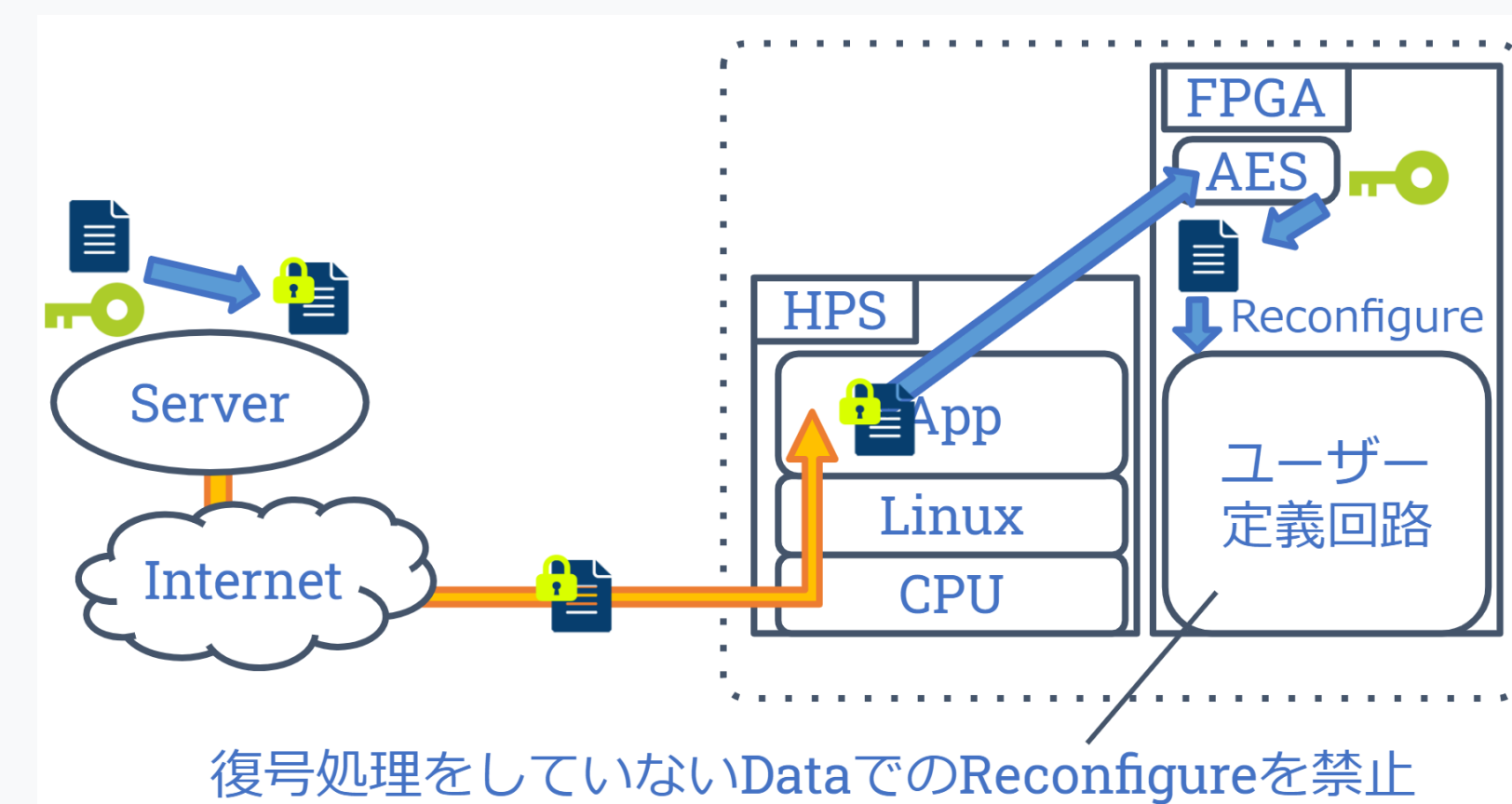
➤ ベンダ提供のAES回路による復号化+認証

● このAESに脆弱性があったら.....?

このAESモジュールによるConfigure Data の復号化と認証によって前述の脅威に対抗している。しかし万が一このAESモジュールに脆弱性があった場合、SoC FPGAは危険に晒されることになる。

● ゼロトラストに向けて

現状のベンダーの対策では、ベンダーのAESモジュールを「信頼」が前提となる。「ゼロトラスト」を実現するためには、ベンダーのAESモジュールに依存しない、別の対策が必要。



ベンダーによる対策

➤ ローエンドなチップでは認証を使えない

● ハイエンドなチップとはセキュリティ機能に大きな差

ローエンドなチップのAESモジュールは認証をサポートしていない。

提案手法 "Sereform"

➤ ベンダーのセキュリティ機能に依存しない

● ベンダーのセキュリティ機能を使わずに実装

- 単一障害点を回避できる
- ベンダーを信用しない「ゼロトラスト」へ

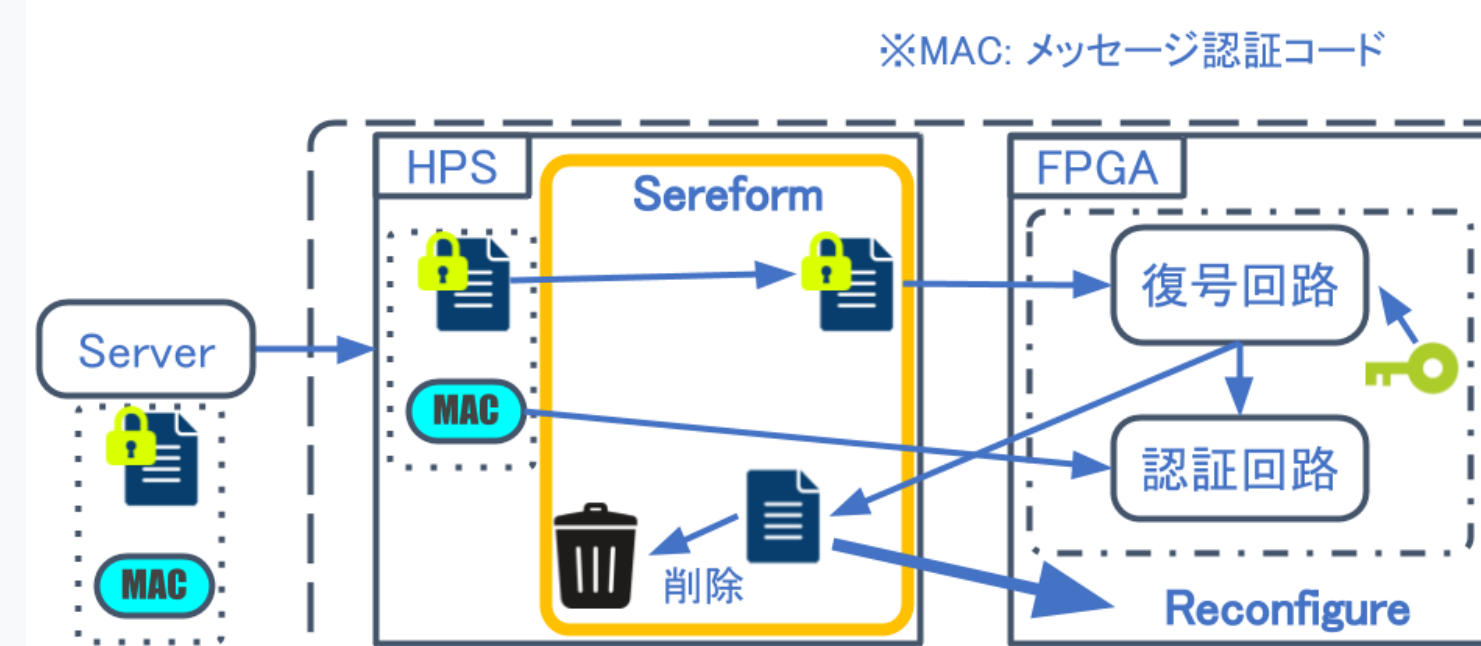
● FPGA実装の暗号化・認証方式をサポート

- ・ 煩雑なLinuxとFPGAとのやりとりを抽象化
→ Appから簡単に操作できる
- ・ 暗号化・認証方式の用意は運用者に任せる
→ Reconfigureで、どの方式を選ぶか選択可
- ・ 今日までのFPGAの研究成果を利用することができる
- ・ Configure Data の暗号化によって対策可能

● Pythonモジュールとして簡単に扱える

- ・ SereformはC言語による実装(163行)
 - ・ FPGAとの通信(/dev/memへのread/write)
 - ・ Reconfigureに関わる操作(デバドラの操作)
- ・ SWIGによりPython向けインターフェース作成
- ・ Pythonから簡単にFPGAを扱えるように！

SereformによるRemote Reconfiguration



More information

追加資料やソースコードを公開予定です。



<https://w-haibara.github.io/sereform>

