Abstract

AI can be deployed to IoT devices to enable autonomous decisions based on pre-trained inference models. Yet the inconsistent level of security practices in IoT devices remains a long-standing issue. Deploying AI to IoT devices, especially constrained devices, needs a holistic approach to safeguard valuable model IP, potentially confidential input data such as audio or biometrics, and inference results vulnerable to adversarial attacks or spoofing. However, AI and security are often silos in both service delivery and device integration organizations. This white paper examines the problem space and specific needs around deploying AI models as secure IoT workloads on Arm based devices and how Linaro’s Confidential AI Project aims to address issues raised through the use of open source standards-based solutions.

Contents

Introduction - the Challenges of Securely Deploying AI in IoT 2
• What Makes AI Different as an IoT Workload?

The AI Security Problem Space 3

Unmet Needs for Deploying AI Securely in IoT 4
• AI Model Integrity
• AI Data Integrity
• AI Developer Experience

Sidebar: Open Source Standards-based Solutions 5

Introducing the Confidential AI Project 6
• A Set of Project Goals to Benefit the Ecosystem
• The Route to Confidential and Secure AI
  - Secure Model
  - Secure Data & Results
  - Developer Tools and Workflow
• Deliverables and Milestones

References 8

How to find out more and participate 9

About Linaro 9
Introduction - the Challenges of Securely Deploying AI in IoT

What Makes AI Different as an IoT Workload?

Deploying AI into smart devices means that those devices can analyze data from their surroundings and make autonomous decisions based on that data. This represents a huge new opportunity across traditional and new automation segments. The performance of deployed AI models can bring significant competitive advantage to OEMs and service providers, albeit at the expense of significant new R&D cost or technology licensing both in model development and also in training data [1].

It’s important to safeguard competitive advantage and investment in AI models to defend against IP leakage or in some cases outright IP theft and the rise of potential clone products. Models trained on private data sets are also at risk of reverse engineering to identify at least some of the training data. Finally, the autonomous nature of smart devices unfortunately means bad actors can benefit from compromising that autonomy for example by developing adversarial attacks [2].

A smart monitoring device deployment into a high capital industrial environment might be trained to carry out condition monitoring to minimize plant downtime due to preventative maintenance cycles but also to detect misuse, tampering or modification for warranty or rental contract enforcement. In this case, as well as the value in developing the monitoring product, the data could be commercially sensitive if it gives plant usage patterns, and an unscrupulous end customer might want to circumvent the monitoring in order to use it outside of conditions specified in the warranty or rental contract [3].

Security threat mitigation in all these cases is more important than ever. Failure to address traditional security threats helps enable the AI/ML-specific attacks in both the software and physical domains. It’s also an issue that the skillsets of security engineers and data scientists typically do not overlap.

AI models and device platform code are also typically developed by separate organizations or teams and updated at a different cadence. This is a difference with conventional IoT devices where the final image is likely to be built by the OEM. This split responsibility and cadence creates new challenges for build, packaging and authentication as ultimately an AI model may be downloaded over-the-air (OTA) into a device where the OEM code is considered to be in the untrusted domain.

What Makes AI Different as an IoT Workload?

<table>
<thead>
<tr>
<th>Model</th>
<th>Data</th>
<th>Supply Chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Often very high value in terms of R&amp;D investment and also training data</td>
<td>Applications often use confidential/sensitive data inputs such as speech, video, biometrics</td>
<td>Complex secure development flow for AI framework, model and platform software</td>
</tr>
<tr>
<td>Threat models can include reverse engineering both model and details of training set</td>
<td>Outputs are high value ‘inference’ results which often directly guide decision logic</td>
<td>Separate AI model ownership and lifecycle management for e.g. OTA</td>
</tr>
<tr>
<td>Leaking model access and knowledge can permit adversarial example development</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The AI Security Problem Space

A service provider wanting to deploy AI models securely on IoT devices faces questions about trust in the supporting platform and also the range of threat models. Initial deployments may be able to rely on a close trust relationship between the OEM and service provider, possibly through close collaboration and a number of 1:1 legal agreements. In response to the need to scale deployments to a range of devices and OEMs, it will become necessary to separate the untrusted code from the OEM and high value model IP in a device. This may not only be for concerns related to the OEM or supply chain, but also concerns that model IP and data may be accessible via debug tactics either directly or by modifying the OEM code on deployed devices.

In addition to a direct attack on the model IP, there are threats related to the security of the input data. For example, how to secure confidential data on a remote device from theft via compromised hardware or insecure code. As well as direct theft of data on the device, what if an adversary could take over the datapath and drive the model for their own purposes? In this case it may be possible for an adversary to input noise or adversarial data or probe the workings of the model. Finally, since inference outputs typically drive decision making logic, an adversary (or a bug) could cause the output to be replaced with misleading results. These two final cases mean there is a requirement to secure the inference outputs from theft or tampering.

An end-to-end development flow from model training to IoT device deployment must interface smoothly with device key provisioning and image signing. The development flow must not be cumbersome or fragile as AI model updates are likely to be pushed frequently - far more frequently than the supporting platform software. Standardized tooling for configuration, component packaging and image signing need to be integrated into a DevSecOps workflow that can be shared between service provider and OEM whilst preserving trust isolation and ensuring the integrity of the AI model throughout the supply chain.

All the above concerns must be addressed using platform security best practices by selecting cryptographic algorithms and approaches which fit within the constraints of a resource constrained IoT device.
Unmet Needs for Deploying AI Securely in IoT

AI Model Integrity

The requirements are to protect model IP not only from threat of direct theft, but also to limit the possibility of the model IP leaking and helping competitors. Ultimately this protects the IP business by mitigating the risk of a “clone army” of competitors. In addition, it protects the service business by limiting the potential for bad actors to generate adversarial data that compromises the service in some way.

In short, AI model integrity needs to be assured when models are deployed in remote devices. To do this, it’s necessary to safeguard the model throughout its lifecycle. This also gives the benefits that good security practice always brings, for example protection and effective defence-in-depth against widespread hacks against devices and the services layered on top of them.

On the device itself, this means running inference in a secure environment - secure boot, key provisioning, credential management, trusted storage and mandating isolated inference execution in a trusted execution environment. It also needs to take into account extensible support for hardware AI accelerators within the secure environment such as the Arm Ethos-U55.

AI Data Integrity

High-value data is the driving force behind high-value AI services, regardless of whether that data is in the form of inference outputs driving local autonomous decisions, biometric or other sensitive input data, or data which enables the AI models to function. AI data integrity assurance within remote devices means confirming that inputs come from expected sensor sources, securing the end-to-end data pipeline, and trusting that inference results have not been tampered with.

The key steps for data integrity are:

- secure communication for restricted sensor access
- a pipeline to read data securely, run inference, and post-process inferred results
- mutual (client + server) authentication using registered credentials for trusted communication and signature verification

This is summarized in the diagram below.
AI Developer Experience

Making the developer experience at the heart of any transformation is key to a sustainable and scalable business. Often a proof-of-concept is followed by a declaration of victory by the leadership. Meanwhile the developers painfully hand-crank a Rube Goldberg machine to deploy images to the field and scaling to other platforms and services must be relegated to a nice-to-have because of the overhead in maintaining the initial deployment.

Standardization of configuration for package dependencies, builds, OTA deployments and at run-time is a long overdue feature for IoT devices, especially those which are MCU based. In the complex world that requires security and AI, the developer experience for building and deploying ML models needs to minimize any necessity to stitch together arbitrary service provider and vendor tooling for e.g. provisioning and credential management, have platform specifics taken into account under the hood and provide ease of deployment and time to market benefits.

AI developers should be abstracted from having to make security decisions about crypto algorithm selections, since these are strongly affected by both constrained device capabilities and up-to-date guidance on best security practice.

The business benefits are faster deployment of new features, faster deployment of fixes.

Shorter cycle to bring new platforms online, improved business agility to grow features, and improved business resilience with respect to fixing issues.

This is effectively taking the AI DevOps process [4] and adding security standards at all stages to give a true DevSecOps flow.

Open Source Standards-based Solutions

The open-source ecosystem in partnership with Linaro and Arm maintains a number of standards and references for secure platforms and general software configuration on the Arm architecture which form the foundation for the Confidential AI Project.

These include:

PSA Certified
The PSA Certified program (co-founded by Arm and six other expert security companies in 2019) is a freely available security framework (including threat models and security analyses, documents [5] specifying security requirements and Application Programming Interfaces, open source reference implementation and test suites) and an associated security evaluation and certification program. Its mission is raising the security bar so that every connected device at least has a foundation of security best practice. Together, all the resources, and the ecosystem of over 50 partners, are enabling consistent design-in at the right level of security.
https://www.psacertified.org/

Trusted Firmware
This open governance project hosted by Linaro provides a reference implementation of secure software for Armv8-A, Armv9-A, Armv8-M and dual-core Armv7-M. It provides SoC developers and OEMs with a reference trusted code base complying with the relevant Arm specifications. The Trusted Firmware codebase is the preferred implementation of Arm specifications, allowing quick and easy porting to modern chips and platforms. This forms the foundations of a Trusted Execution Environment (TEE) on application processors, or the Secure Processing Environment (SPE) of microcontrollers.
https://www.trustedfirmware.org/

Arm Project Centauri
Project Centauri is the latest industry-wide strategy for driving rapid, exponential IoT growth on Arm-based microcontrollers. It brings security, harmony, and homogeneity to billions of IoT endpoint devices by combining foundational standards, security initiatives, and the extensive Arm Cortex-M software ecosystem. Project Centauri draws upon and includes Arm’s rich portfolio of Cortex-M software, bringing together complementary initiatives such as Open-CMSIS-Packs and PSA security standards under a single MCU software strategy.
https://www.arm.com/solutions/iot/project-centauri

Open CMSIS Pack
Software compatibility for component reuse has long been a challenge in the microcontroller space, especially for the IoT, which is much more diverse at the hardware level compared to PCs or the data center. Open-CMSIS-Pack is a project within Linaro which will remove this complexity, delivering a standard for software component packaging and related foundation tools for validation, distribution, integration, management, and maintenance. The project is currently hosted and managed as an incubation project by Linaro in partnership with Arm, NXP and ST.
https://www.open-cmsis-pack.org/
Introducing the Confidential AI Project

A Set of Project Goals to Benefit the Ecosystem

From the above review of the problem space, we can identify the following high-level goals as benefits to all organizations tasked with deploying IoT devices for an AI-enabled service:

- Deployed AI model integrity assured in remote devices
  - Protects investment in either model development or 3rd party model licensing
  - Manages model security and lifecycle independent from platform
- AI data integrity assured within remote devices
  - Assures that inputs come from expected sensor sources
  - Data pipeline confidentiality assurance
  - Provides trust that inference results have not been tampered with
- Best developer experience for building and deploying ML models
  - Minimizes necessity to stitch together CSP and SiP tooling
  - Platform specifics are taken into account under the hood
  - Provides ease of deployment and time to market benefits

The Route to Confidential and Secure AI

Linaro’s Confidential AI Project integrates a number of strands of ongoing technology development in Linaro for the Arm architecture. These include: key provisioning, credential management, Root-of-Trust, secure sensor data pipeline, TLS authentication and AI inference execution as a secure service. It also references the key standards and references for secure platforms and general software configuration on the Arm architecture (see Sidebar: Open Source Standards-based Solutions).

Secure Model

The Confidential AI Project instantiates a secure Application Root-of-Trust (ARoT) service running a Tensorflow Lite Micro model using Trusted Firmware-M. Trusted Firmware-M (TF-M) implements the Secure Processing Environment (SPE) for Armv8-M, Armv8.1-M architectures and dual-core platforms. There are plans to extend to uTVM as well for more model flexibility. The inference model is accessed via a pseudo driver running in non-secure space.
Secure Data & Results

Data security starts with access to sensor data and secure sensor communication over I2C. Sensor data processing is handled via a secure datapath isolated from the non-secure platform code. This is also run as a secure ARoT service with a Pseudo driver managing calls from the non-secure platform code. Results data are hashed and signed on the secure side. A reference implementation avoids AI developers making bad security choices: which algorithms are up to date today, etc., order of operation, key sizes and types, etc.

If secure data storage is required to be stored off-chip, data can be encrypted using a device-bound encryption key derived from a hardware unique key (HUK).

Developer Tools and Workflow

Work on the project includes proof-of-concepts for new features and reference implementations for existing standards. These are implemented as command-line tools which can be used as part of an automated flow and also integrated into IDEs. This approach of reference code driven interoperability has been demonstrated by the Open-CMSIS-Pack project (see sidebar).

Linaro has started developing tools that include certificates and key generation utilities. These tools can be integrated into development environments for the IoT platform OS and OTA agent flow. They encapsulate knowledge of constrained device capabilities and up-to-date guidance on best security practice (Diagram: X.509 Certificates and Key Generation Workflow).
Deliverables and Milestones

Linaro is currently working on a proof of concept to provide a flexible Secure Data Pipeline to make this workflow easier to implement on Arm devices on top of TF-M for Linaro LITE members.

- Proof of Concept - Inference running in trusted processing environment
- Add specifics to support one or more service provider IoT platform OS, OTA agent, model framework and tooling e.g.
  - TFLite Micro, microTVM, Pytorch, ONNX (as a model format)
  - Update agent integration
  - RTOS
  - Tools
- MVP (Dev Summit October 2022)
  - Platform security
  - Secure accelerated inference
  - Secure pipeline and data

How to Find Out More and Participate

Talk to us about how the direction of Confidential AI project and how it relates to your business

There are multiple ways to participate in the Confidential AI project. Founding Membership is open to those interested in directly influencing the direction of the project to ensure it delivers the solutions they need. By becoming a member, your engineers get to work with Linaro’s team of experts and other industry leaders on scoping and steering the solution.

For more information on membership, contact us on confidential_ai@linaro.org

---

**X.509 Certificate and Key Generation Workflow**

- Secure ARoT Key Derivation Service
  - HUK
  - Key Derivation Algorithm
    - HKDF
    - HKDF
  - “info”
  - Signing Key
  - mTLS Auth Key

- Certificate Authority (LinaroCA)
  - CA Key
  - Certificate Registry
  - X.509 Certificate Signing Request

- Certificate
  - mTLS Auth Certificate
  - CA Signature
About Linaro

Linaro leads collaboration in the Arm ecosystem and helps companies work with the latest open-source technology. The company has engineers working on more than 70 open-source projects, developing and optimizing software and tools, ensuring smooth product roll outs, and reducing maintenance costs.

Work happens across a wide range of technologies including artificial intelligence, automotive, datacenter & cloud, edge & fog computing, high performance computing, IoT & embedded and mobile. Linaro is distribution neutral: it wants to provide the best software foundations to everyone by working upstream, and to reduce costly and unnecessary fragmentation. The effectiveness of the Linaro approach has been demonstrated by Linaro consistently being listed as one of the top ten company contributors, worldwide, to Linux kernels since 3.10. To ensure commercial quality software, Linaro’s work includes comprehensive test and validation on member hardware platforms. The full scope of Linaro engineering work is open to all online.

To find out more, please visit https://www.linaro.org

References

[1] The cost of training machines is becoming a problem
   The Economist Technology Quarterly June 13 2020

   Security Documentation Microsoft March 31 2021

[3] Things to Know About Heavy Equipment Warranties
   Gocodes Blog (undated)

   Accenture April 22 2021

   Arm Platform Security Model 1.0
   ARM DEN 0128 May 27 2021