Protecting Your IP in a Cloud-Connected World

Presenter: Moung Heum Eum

Protecting Your IP in a Cloud-Connected World

Abstract

The rapid explosion of cloud-enabled applications and businesses is generating a larger connectivity footprint. This means your IP is now facing a bigger and more global threat from hackers, counterfeiters and competitors. We will discuss threats and how to counter them using a hybrid solution comprised of both hardware and software.

Protect Your IP

Can an attacker steal your code?

- Either online, after deployment or in supply chain

Can an attacker [ab]use your services?

- How do you know the device that “calls home” is legitimate

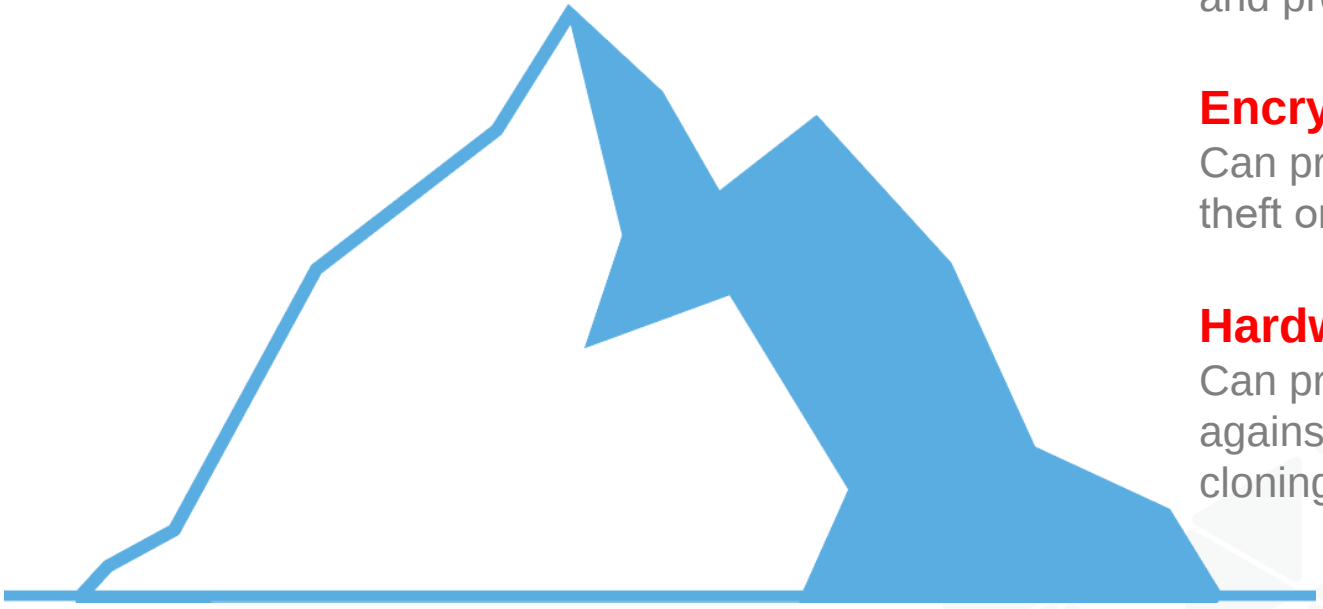
Can an attacker steal/fake your data?

- How can you trust readings from your devices

Can an attacker steal your business?

- Fake devices, or supply chain fraud

Security is Hard



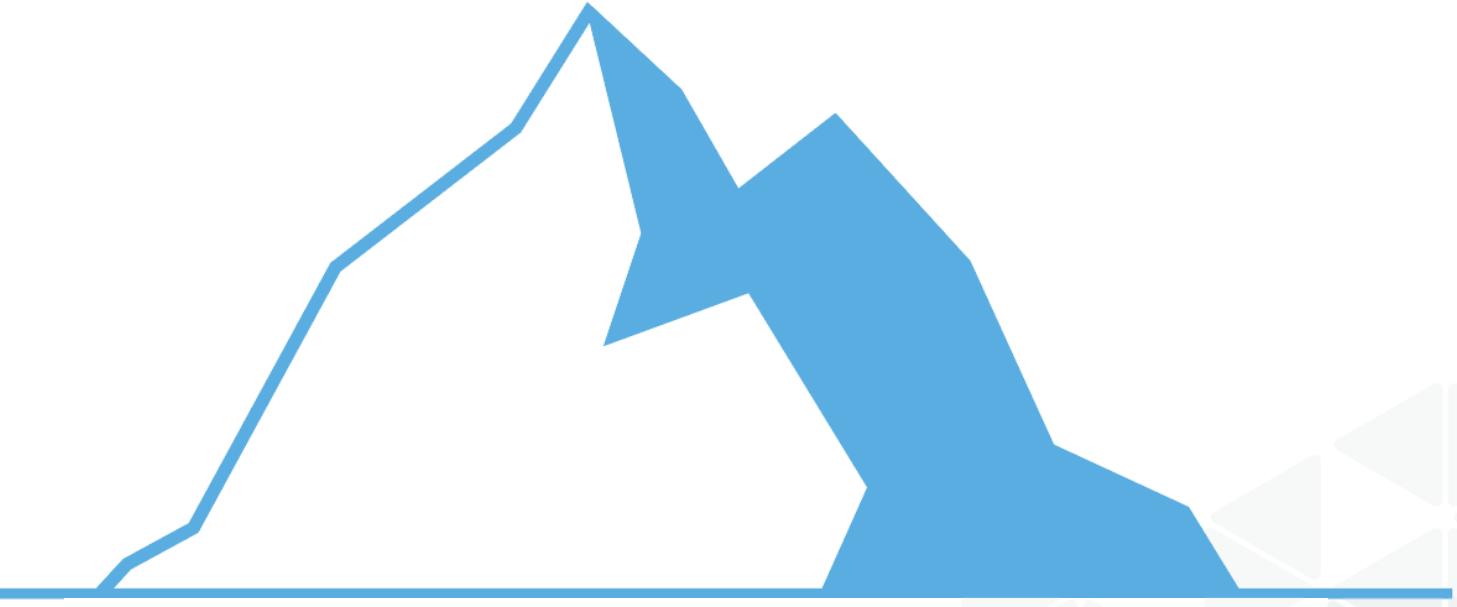
Code Isolation
Is critical to reduce attack surface and protect software on device

Encryption / Signing
Can protect critical data against theft or manipulation “on wire”

Hardware Security
Can protect code and keys against theft and devices against cloning

“Good Enough” Security?

Security Has a Lot of Hidden Baggage



Security Has a Lot of Hidden Baggage

Security Features
Encryption, isolation, …

Secure Provisioning
Flashing, update, attestation

Secure Foundation
Secure boot, power management, secure interrupts, debug, data flash, MPU control, tamper detection, …

Security is a Speciality

There is a lot “below the surface” to get right, and even “simple” features like encryption need a lot of care to make secure.

Trustonic provides a platform approach to reduce project risk.

With the right tools, everyone can build secure devices

- We focus on security
- You focus on your application

Trustonic Expertise

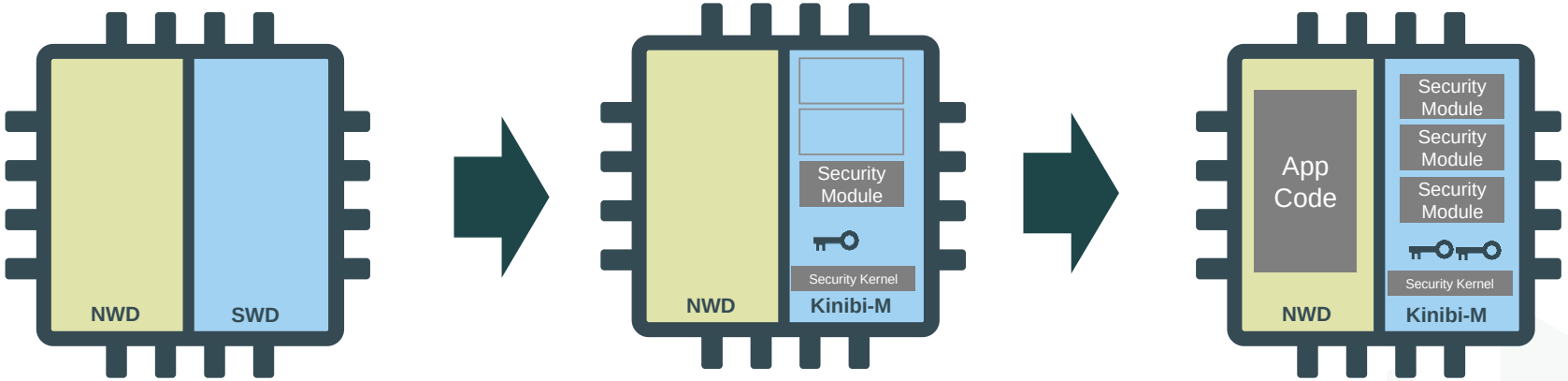
ARM/Gemalto Joint Venture

Initially focused on exploiting TrustZone™ technology in A-Class CPUs

2 Billion devices run our security foundation today

Recently added support for ARM® M-Class devices

SAML11 KPH



SAML11 with TrustZone™ provides hardware security features

- Normal and secure world split
- Technology for secure boot and MPU isolation
- Crypto, tamper detect (etc.)

Trustonic adds a security foundation

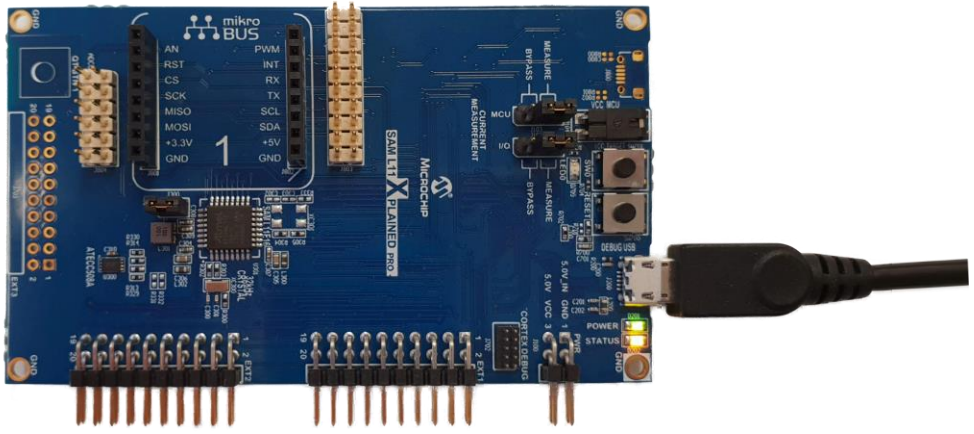
- Isolation for secure S/W and secure H/W access
- Simple APIs for secure fns.
- Unique device key

You add application code

- You can add normal world and secure functions
- Use provided high level APIs
- And/or direct access to hardware

Getting Started

Off the shelf SAML11 Xplained Pro



Free SDK for development

https://www.trustonic.com/microchip/iot-developer-kit/

(Use SAML11 KPH for production devices)

TRUSTONIC

MICROCHIP

Security Based on “Trusted Modules”

Modules are:

- Code written in C
- Have a single entry point for calls and interrupts
- Own a set of private resources: PINS, data, interrupts, flash...

Modules can:

- Securely identifies their callers, limiting exposure to attack
- And can call other secure modules
  - Encryption, attestation, transactional data flash (key/value store)
  - Secure power management, UART (printf) etc.
  - …or any you care to write

Example Module: Encrypt Using Stored Key

Standard pattern for calls to any module (global platform)

```
__attribute__((section (".sessionEntry"))) TEE_Result invoke_entry(uint32_t commandID, TEEC_Operation *operation) {
switch (commandID)
{
  case CMD_ENCRYPT: {
    TEEC_Operation op;
    uint8_t key[16];
    REQUIRE_TEEC_PARAMETERS(TEEC_MEMREF_TEMP_INPUT, TEEC_MEMREF_TEMP_OUTPUT, TEEC_NONE, TEEC_NONE);
```

Call storage module to get our key (modules can only read their own keys/data)

```
    // Read the key from secure data flash (all storage is private to each module)
    op.paramTypes = TEEC_PARAMETERS(TEEC_VALUE_INPUT, TEEC_MEMREF_TEMP_OUTPUT, TEEC_NONE, TEEC_NONE);
    op.params[0].value = 123; // key ID=123
    op.params[1].tmpref = {key, 16};

    TEEC_Result res = TEE_InvokeCommand(KM_SECURE_STORAGE, TEE_CMD_STORAGE_READ_ITEM, &op, 0);
    if (TEE_SUCCESS != res) return res;
```

Call crypto module to encrypt message and return to caller

```
    op.paramTypes = TEEC_PARAMETERS(TEEC_MEMREF_TEMP_OUTPUT, TEEC_MEMREF_TEMP_INPUT, TEEC_MEMREF_TEMP_INPUT, TEEC_NONE);
    op.params[0].tmpref = operation->params[1].tmpref, // output
    op.params[1].tmpref = operation->params[0].tmpref, // input
    op.params[2].tmpref = {key, 16} // key
    res = TEE_InvokeCommand(KM_CRYPTO, TEE_CMD_ENCRYPT_MESSAGE, &encrypt_operation, 0);
    operation->params[1].tmpref.size = op.params[0].tmpref.size; // copy back size
    return res;
```

The Power of Uniqueness

Every MPU has a unique key embedded in it

This can be used to bootstrap secure communication

- Prove a device is genuine
- Connect to a cloud service (AWS, Google, ….)
- Perform “Just in time key (or license) provision”

The pre-installed key can avoid the need for additional secure key provision in factory, or for a separate secure element

… and can protect against cloning (as clones won’t have key)

See samples included in SDK

Protecting Code IP

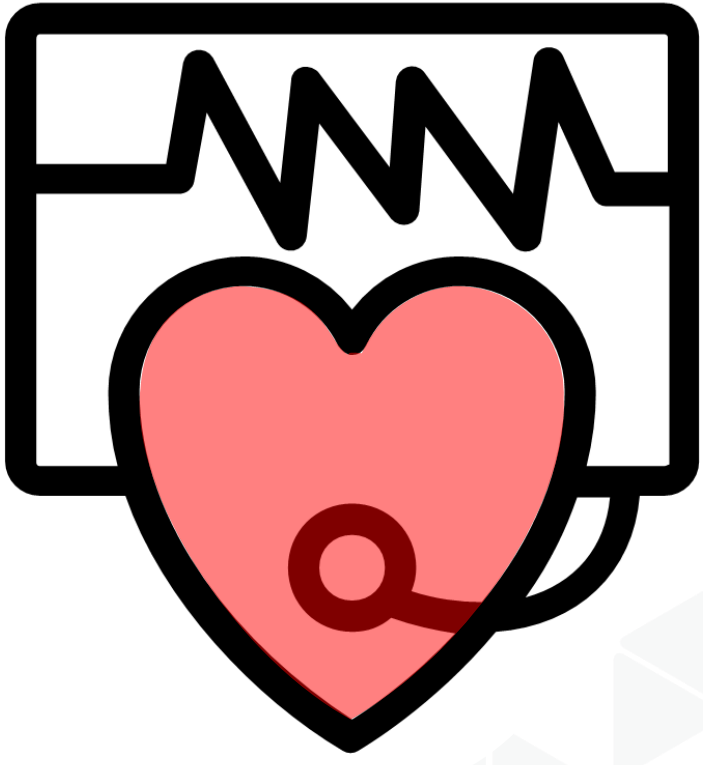
The *Production* SDK

- Provides secure install and update of modules on SAML11 KPH devices, while preserving security properties

- This prevents module code from being read, even by someone later in production line

This provides protection for code against reverse engineering / theft and allows secure parts to be used in insecure factories

Example



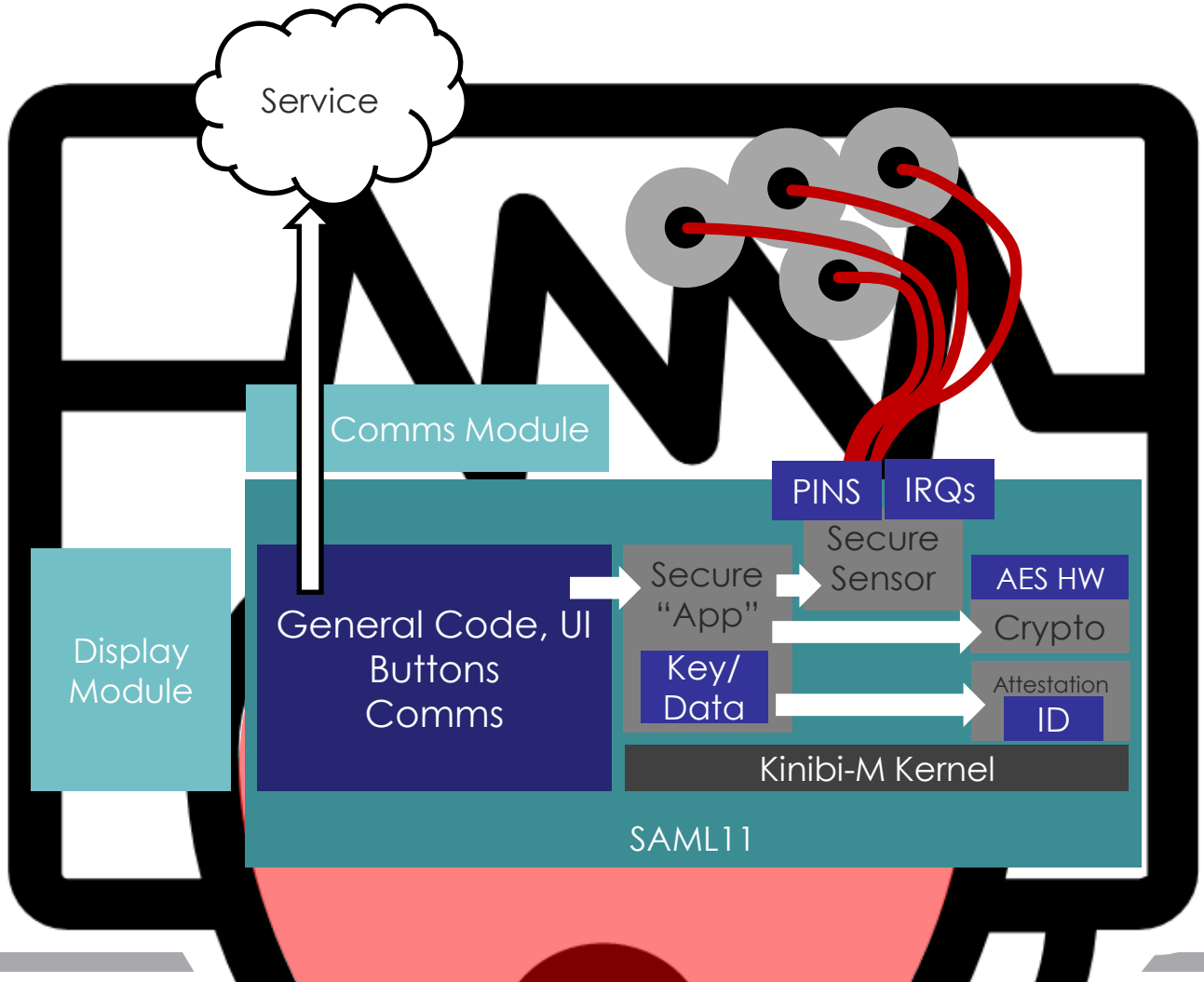
Example

Isolation minimizes security cost

- Doesn't affect "main" code
- Secure modules can be reused
- Hard to attack or exploit bugs

Attestation adds value

- Demonstrate integrity of readings
- Enable end to end encryption for privacy



SAML11 KPH is the Path to IP Protection

Hardware security with secure provisioning prevents code theft

Key management enables services to identify legitimate devices and prevent abuse by rogue ones

Data can be trusted as the foundations are secure

Your business is protected, and you can focus on adding value

Thank You

Visit

Visit https://www.trustonic.com/microchip
for information and SDK access