Application Using Secure Element and TrustZone on SAML11

This application demonstrates a security use case on SAML11 by combining Arm TrustZone and secure element ATECC508.

TrustZone
TrustZone provides the flexibility for hardware isolation of memories and peripherals, therefore reinforcing the ability of Intellectual Properties (IP) and Data protection. SAML11 provides up to six regions for the Flash, up to two regions for Data Flash, up to two regions for SRAM and the ability to assign peripherals, I/O pins, interrupts to secure or non-secure application.

ATECC508
The Microchip ATECC508A integrates ECDH (Elliptic Curve Diffie Hellman) security protocol an ultra-secure method to provide key agreement for encryption/decryption, along with ECDSA (Elliptic Curve Digital Signature Algorithm) sign-verify authentication for the Internet of Things (IoT) market including home automation, industrial networking, accessory and consumable authentication, medical, mobile and more.

For more information please visit:
GitHub

Or contact
jpiwek@arroweurope.com

Use case Diagram

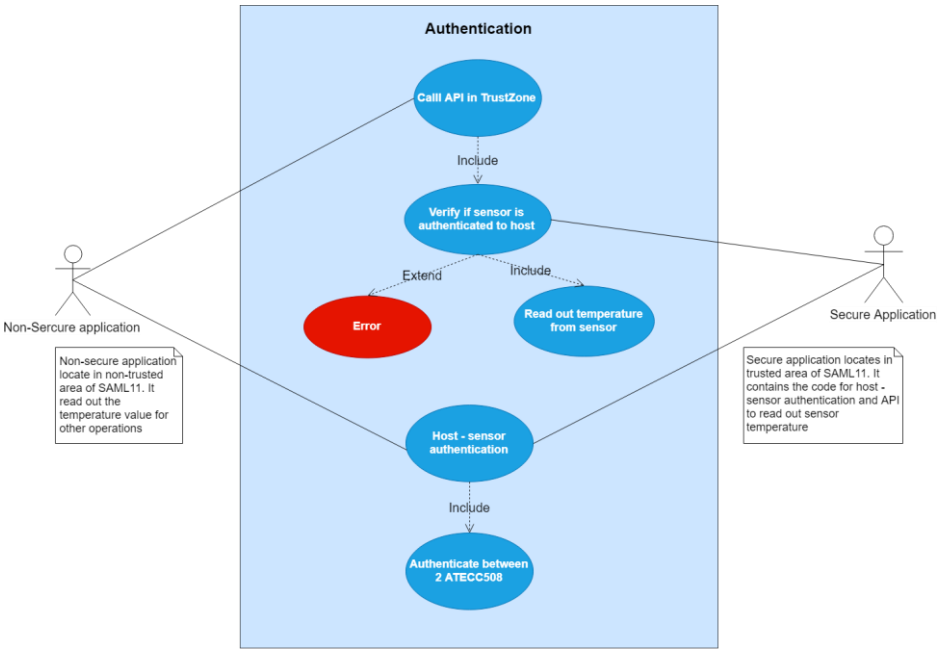

Figure 1: Use case diagram

Description

Inside SAML11, there are two running application, which are the secure and non-secure one. Each of them locates in trusted and non-trusted area respectively. The secure application contains the secure element driver, temperature sensor driver (simulated by a random function), and code for sensor-host authentication process. When the non-secure application tries to call the secure API, which returns the temperature value from the sensor, the secure application checks if the sensor is already authenticated to the host. If authentication was successfully, secure application allows the API to be called. Otherwise, an error message is prompted.

Features/Benefits

> IP protection
> Software isolation
> Authentication
> Cryptography

Key Components

> SAML11
> Arm TrustZone
> ATECC508

```
Secure Hello world !
Non-Secure Hello World !
Verify if sensor node is authenticated
Not authenticated
Try to read the temperature value without sensor node authentication
Function is not executed
Press SW0 to start Authenticate
Authentication between sensor and host
Authentication in progress
Host init complete
Serial Number of host
0x01, 0x23, 0x3f, 0xa1, 0x15, 0x47, 0x1e, 0x63,
0xee,

Random from host
0x1a, 0xae, 0x61, 0xf7, 0x1d, 0x0a, 0xc0, 0x35,
0x30, 0x7b, 0xb9, 0x9c, 0x55, 0x54, 0xbc, 0x75,
0xda, 0x01, 0x16, 0xda, 0xc9, 0x88, 0x7b, 0x02,
0x14, 0x77, 0x89, 0xea, 0xe8, 0x22, 0x04, 0x87,

Serial Number of sensor
0x01, 0x23, 0x65, 0x33, 0x90, 0xda, 0x75, 0xda,
0xee,

MAC from sensor
0xe4, 0x0d, 0x9d, 0x1a, 0xb2, 0x81, 0x9b, 0xf5,
0xde, 0x7c, 0xae, 0x87, 0x47, 0x5f, 0x6f, 0x8d,
0x34, 0x89, 0x9f, 0x0b, 0xdb, 0x23, 0x8c, 0xf5,
0x15, 0xd3, 0x41, 0x3e, 0x81, 0xef, 0x69, 0x23,

Host verifying MAC from sensor
Authenticated by host

Verify if sensor node is authenticated
Authenticated
Read out temperature value
temperature: 18
Read out temperature value
temperature: 18
Read out temperature value
temperature: 17
```

Figure 2: Debugging Log

The non-secure application is initialized by the secure application. The non-secure application can access to the API by, at first, executing of the sensor host authentication. The secure application carries out the authentication process and return the status. If the authentication is successful, the non-secure application is allowed to access the API.

Possible Application

> IP protection
> Authentication
> Anti-counterfeit