

# Security use case - TrustRAM

# Application Using TrustRAM and TrustZone on SAML11

This application demonstrates the enhancing security features on the SAML11 microcontrollers for security, which are Trusted RAM and TrustZone.



### **TrustZone**

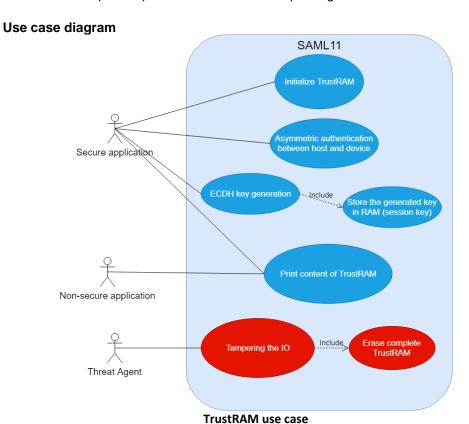
TrustZone provides the flexibility for hardware isolation of memories and peripherals, therefore reinforcing the ability of Intellectual Properties (IP) and Data protection. SAML11 provides up to six regions for the Flash, up to two regions for Data Flash, up to two regions for SRAM and the ability to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

For more information please visit: <u>GitHub</u>

Or contact jpiwek@arroweurope.com

#### **TrustRAM**

Trusted RAM implements 256 bytes of secure memory with address and data scrambling by user-defined key. Trusted RAM is also equipped with chip-level tamper detection and rapid tamper erase to resist micro-probing attacks.



## **Description**

Inside SAML11, there are two running applications which are the secure and non-secure application. Each of them locates in trusted and non-trusted area respectively. The secure application initializes the TrustRAM, executes the authentication between the two secure elements on the host and remote side, generates a secret key. This secret key is stored in the trusted RAM as a session key for later usage.

#### Features/Benefits

- > IP protection
- > Software isolation
- > Secure storage
- > Tamper detection

# **Key Components**

- > SAML11
- > Arm TrustZone
- > TrustRAM

Content inside TrustRAM before and after being tampered

The non-secure application is initialized by the secure application. Non-secure application is not allowed to call any functions directly from the secure application but through a predefined function table. In this case, it is the function to print the content of the TrustRAM.

When a tamper attempt is detected, the content in the TrustRAM will be automatically erased so the sensitive data, it is the session key in this case, is not exposed.

### **Possible Application**

- > Session keys storage
- > Sensitive data storage

