Tech Talks LIVE Schedule – Presentation will begin shortly

Tuesday, December 21	Secure IoT Products with Custom Part Manufacturing Services (CPMS)

Respond to the poll to enter to win a Thunderboard Sense 2

Recording and slides will be posted to:
www.silabs.com/training

We will begin in: 0:00

SILICON LABS

tech talks

WELCOME

Secure IoT Products with Custom Part Manufacturing Service (CPMS)

Mike Glazebrook

The Leader in Short Range IoT Wireless Connectivity

100%
Revenue Based on IoT

Bluetooth® Multiprotocol
Proprietary 100s of Technologies THREAD WiFi
Wi-SUN zigbee Z-WAVE

Breadth and Depth of Wireless IoT Protocols

#1
Share in Mesh

1st
To Market with Multiprotocol, BLE Mesh, BLE 5.1

Innovation
Performance, Power, CoEx, Xpress, Modules

ember	2012	Software ZigBee SoC
ENERGY micro	2013	Low-power 32-bit MCUs
bluegiga	2015	BT Smart Modules
telegesis	2015	ZigBee/Thread Modules
Micrium	2016	Software RTOS
ZENTRI	2017	Cloud Connected Wi-Fi
Z-WAVE	2018	Smart Home Protocol
REDPINE SIGNALS	2020	Ultra Low Power Wi-Fi

Use Case – #1 Over Manufacturing/Grey Market Manufacturing

Chamberlain 3-Button Garage Door Remote Control

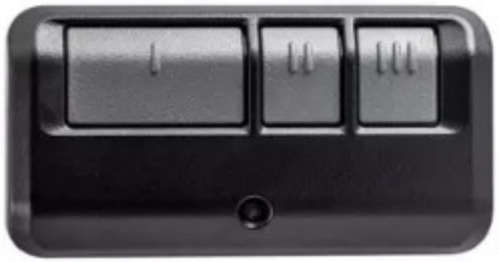
THE HOME DEPOT

$29.98

Garage Door Transmitter For 893MAX Function Digital LED Display

Alibaba.com

Ready to Ship



$2.99-$4.90

+$0.96 (Shipping)

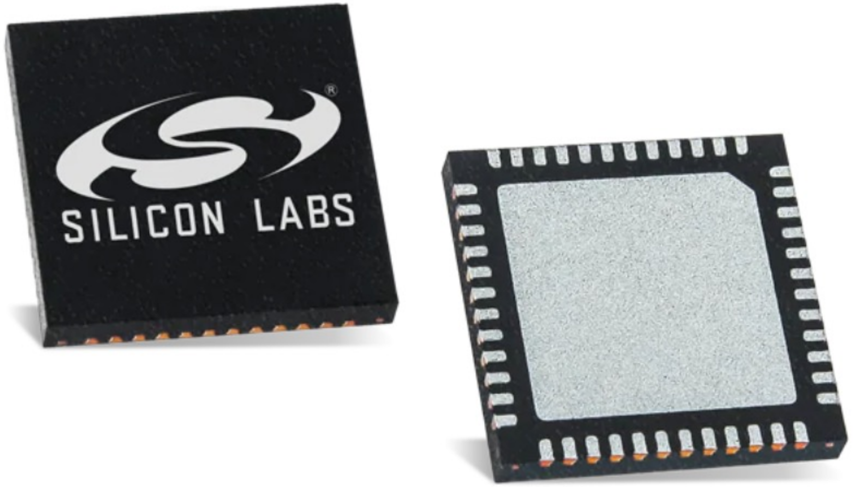
50 Pieces (MOQ)

Use Case - #2 Mutual App and Device Authentication

- What is preventing your mobile app or cloud back end from accepting a 3rd party or malicious device onto your network?
- What is preventing your edge node device from releasing sensitive customer data to a 3rd party mobile app?

CPMS

- **What is CPMS?**
  - CPMS (Custom Part Manufacturing Service) is a service offered by Silicon Labs that allows you to order custom parts that have your firmware and security settings programmed into them before they are sent to the CM
- **Why is this important?**
  - IoT security is complex, and it's easy to accidentally leave a system vulnerable. CPMS provides a "checklist" of easily enabled security features
  - IoT devices are at their most vulnerable during production. CPMS allows you to secure your parts from the moment they're programmed
- **Where is it?**
  - https://cpms.silabs.com/

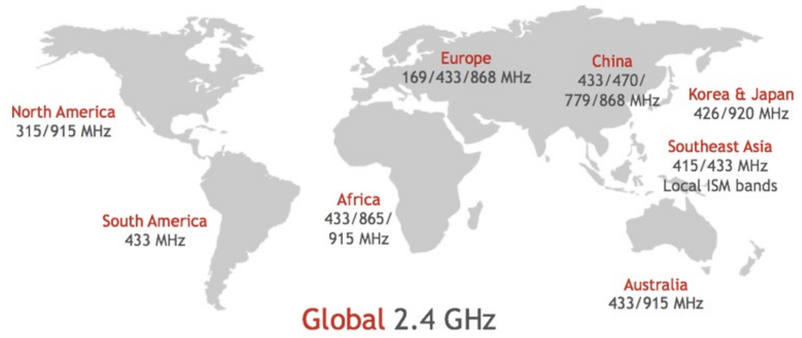


Futureproof Your Design and Start Building Products Today

- **Question:**
  - I'm developing new products today. How does Matter affect my development path?
- **Answer:**
  - The new application protocol will complement existing technologies
  - Start building products today using existing technologies like Zigbee or Thread
  - Update your product in the future using secure over the air updates
  - Use larger memory variant ICs and Modules since memory requirements are not fully defined today
  - Join project Connected Home Over IP

Introducing the Wireless Gecko Series 2 Platform

Bluetooth Proprietary THREAD WiFi zigbee ZWAVE



- **Optimized for IoT Protocols**
  - Zigbee, Thread, Bluetooth, Z-Wave and Wi-Fi
  - Multiband and multiprotocol portfolio
- **High performance and integration**
  - Arm Cortex-M33 processor core
  - Up to 125 dBm link budget with fully integrated PA/LNA
- **Ultra-low power**
  - Very low active current (27 µA/MHz)
  - Low sleep current (1.4 µA)
- **Dedicated security core**
  - Hardware crypto
  - Secure Boot
  - Secure Debug Access
  - True random number generator (TRNG)

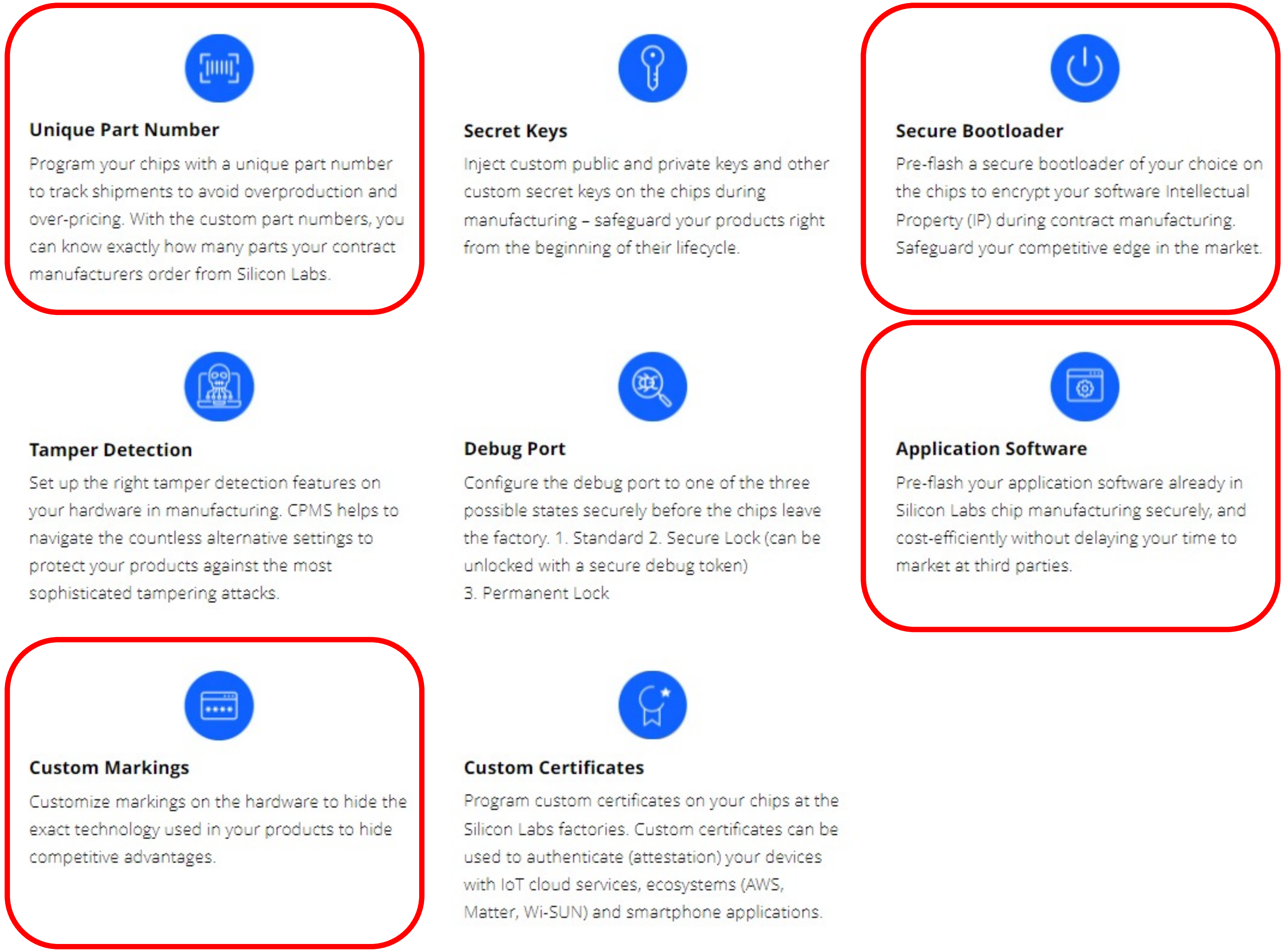
Application Optimized for the IoT

SecureVault™

Base	Mid	High	Feature
✓	✓	✓	True Random Number Generator
✓	✓	✓	Crypto Engine
✓	✓	✓	Secure Application Boot
—	VSE/HSE	HSE	Secure Engine
—	✓	✓	Secure Boot with RTSL
—	✓	✓	Secure Debug with Lock/Unlock
—	Optional	✓	DPA Countermeasures
—	—	✓	Anti-Tamper
—	—	✓	Secure Attestation
—	—	✓	Secure Key Management
—	—	✓	Advanced Crypto

https://www.silabs.com/support/training/discover-the-security-features-of-secure-vault

CPMS - Customization Options



Unique Part Number

Program your chips with a unique part number to track shipments to avoid overproduction and over-pricing. With the custom part numbers, you can know exactly how many parts your contract manufacturers order from Silicon Labs.

Secret Keys

Inject custom public and private keys and other custom secret keys on the chips during manufacturing – safeguard your products right from the beginning of their lifecycle.

Secure Bootloader

Pre-flash a secure bootloader of your choice on the chips to encrypt your software Intellectual Property (IP) during contract manufacturing. Safeguard your competitive edge in the market.

Tamper Detection

Set up the right tamper detection features on your hardware in manufacturing. CPMS helps to navigate the countless alternative settings to protect your products against the most sophisticated tampering attacks.

Debug Port

Configure the debug port to one of the three possible states securely before the chips leave the factory. 1. Standard 2. Secure Lock (can be unlocked with a secure debug token)
3. Permanent Lock

Application Software

Pre-flash your application software already in Silicon Labs chip manufacturing securely, and cost-efficiently without delaying your time to market at third parties.

Custom Markings

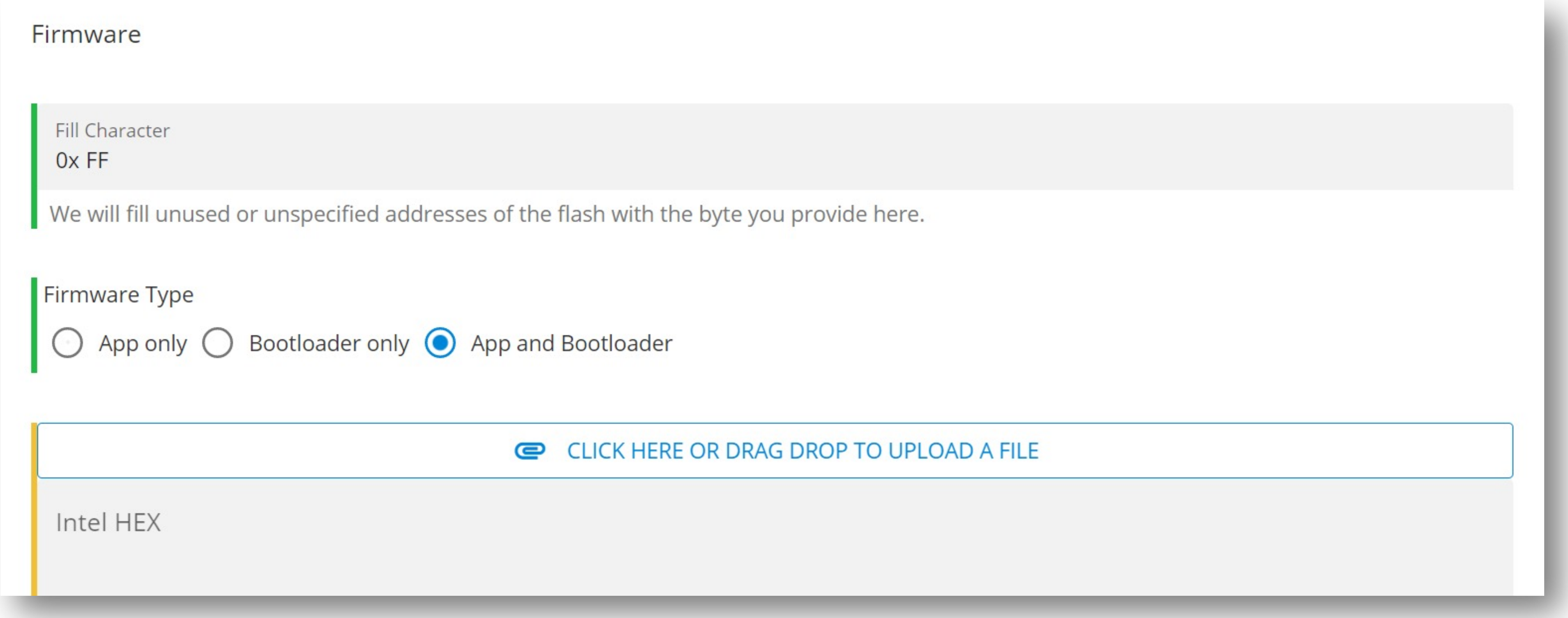
Customize markings on the hardware to hide the exact technology used in your products to hide competitive advantages.

Custom Certificates

Program custom certificates on your chips at the Silicon Labs factories. Custom certificates can be used to authenticate (attestation) your devices with IoT cloud services, ecosystems (AWS, Matter, Wi-SUN) and smartphone applications.

Flash Programming

- CPMS allows you to program your application and/or bootloader into the device before it is sent to the CM
- The Fill character can be specified to aid in detecting memory corruption



Firmware

Fill Character
0x FF

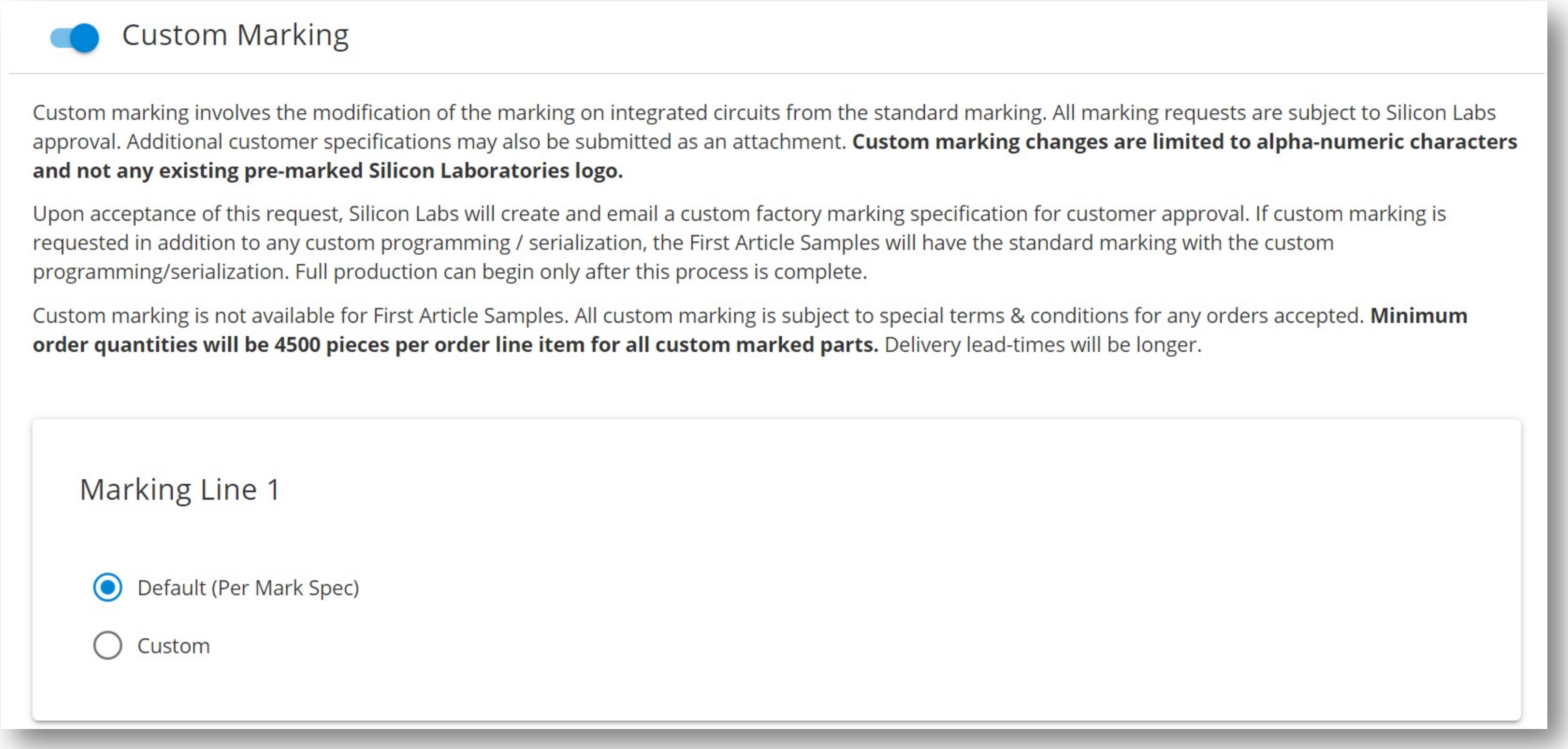
We will fill unused or unspecified addresses of the flash with the byte you provide here.

Firmware Type

○ App only ○ Bootloader only ● App and Bootloader

Intel HEX

Custom Marking



Custom Marking

Custom marking involves the modification of the marking on integrated circuits from the standard marking. All marking requests are subject to Silicon Labs approval. Additional customer specifications may also be submitted as an attachment. **Custom marking changes are limited to alpha-numeric characters and not any existing pre-marked Silicon Laboratories logo.**

Upon acceptance of this request, Silicon Labs will create and email a custom factory marking specification for customer approval. If custom marking is requested in addition to any custom programming / serialization, the First Article Samples will have the standard marking with the custom programming/serialization. Full production can begin only after this process is complete.

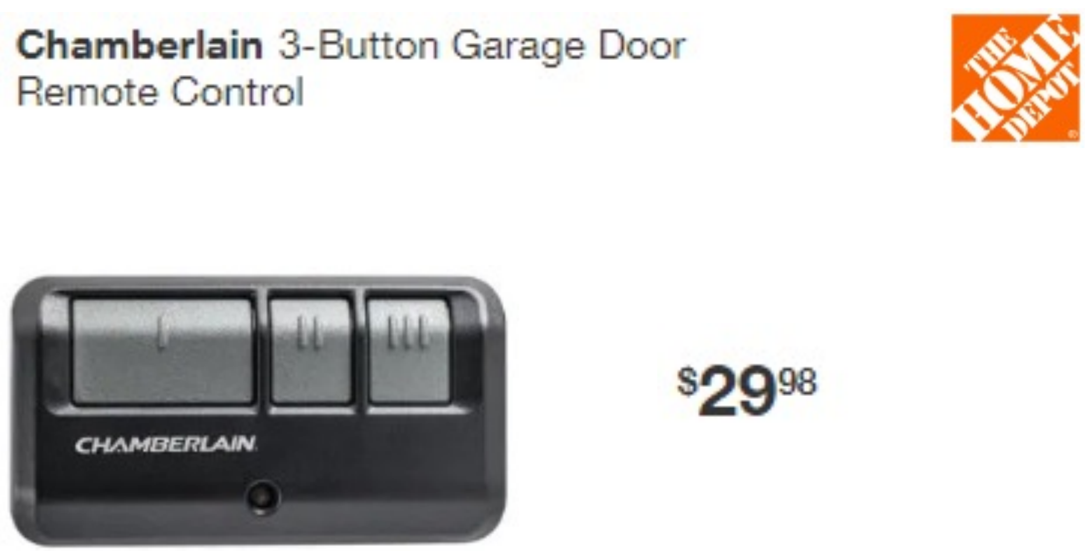
Custom marking is not available for First Article Samples. All custom marking is subject to special terms & conditions for any orders accepted. **Minimum order quantities will be 4500 pieces per order line item for all custom marked parts.** Delivery lead-times will be longer.

Marking Line 1

● Default (Per Mark Spec)

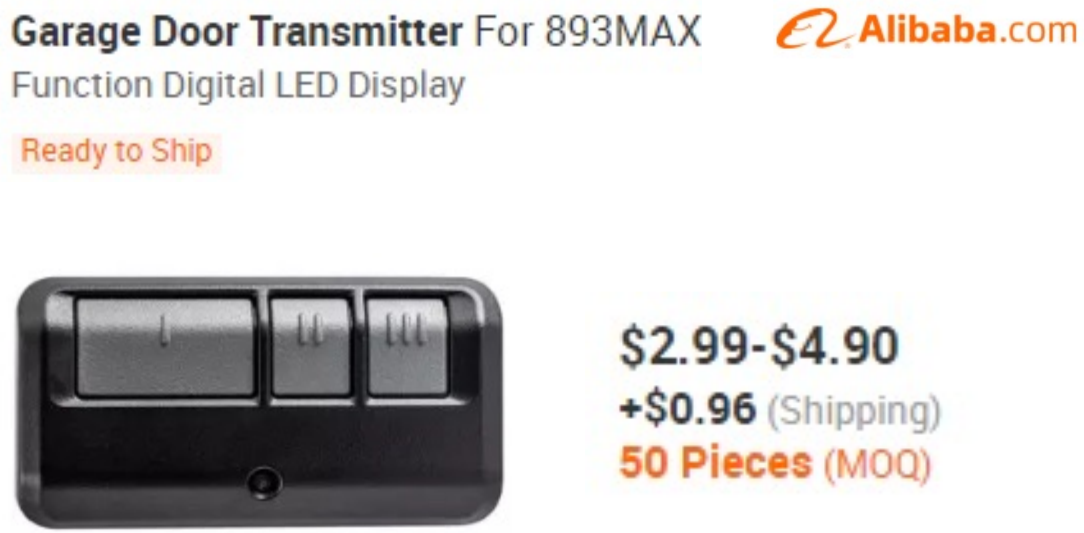
○ Custom

Use Case: Over Manufacturing – Addressable by custom Part Numbers



Chamberlain 3-Button Garage Door Remote Control

$29.98



Garage Door Transmitter For 893MAX Function Digital LED Display

Ready to Ship

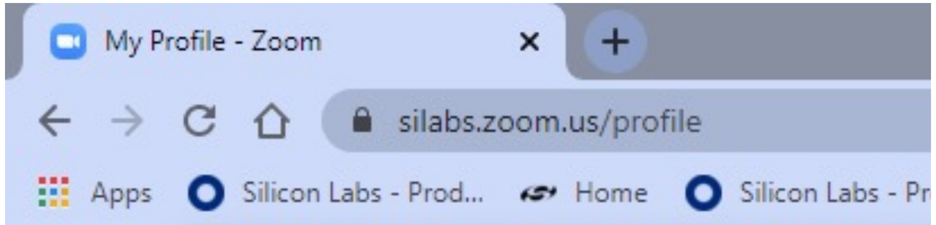
$2.99-$4.90

+$0.96 (Shipping)

50 Pieces (MOQ)

Authentication = Trust

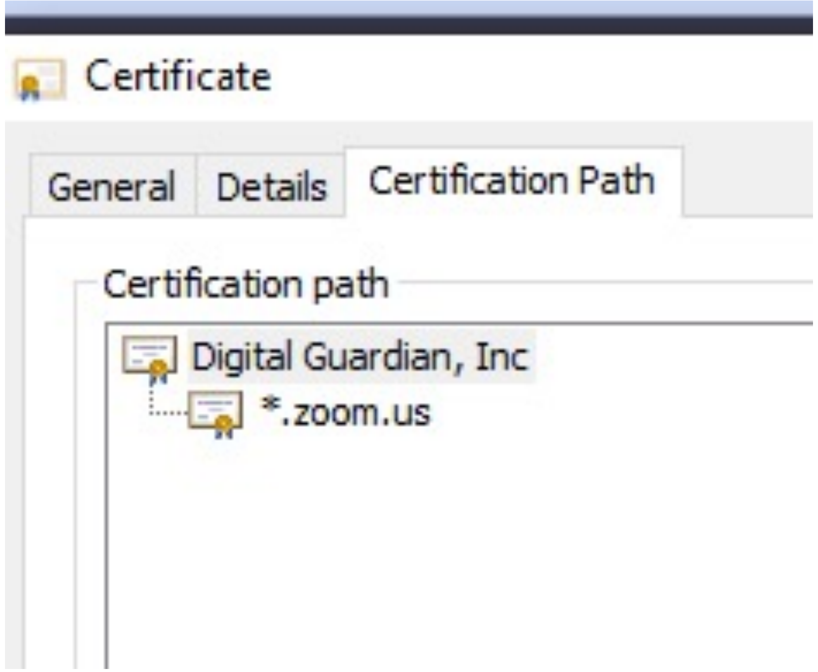
Certificate-Based Authentication



My Profile - Zoom

silabs.zoom.us/profile

Apps Silicon Labs - Prod... Home Silicon Labs - Pr



Certificate

General Details Certification Path

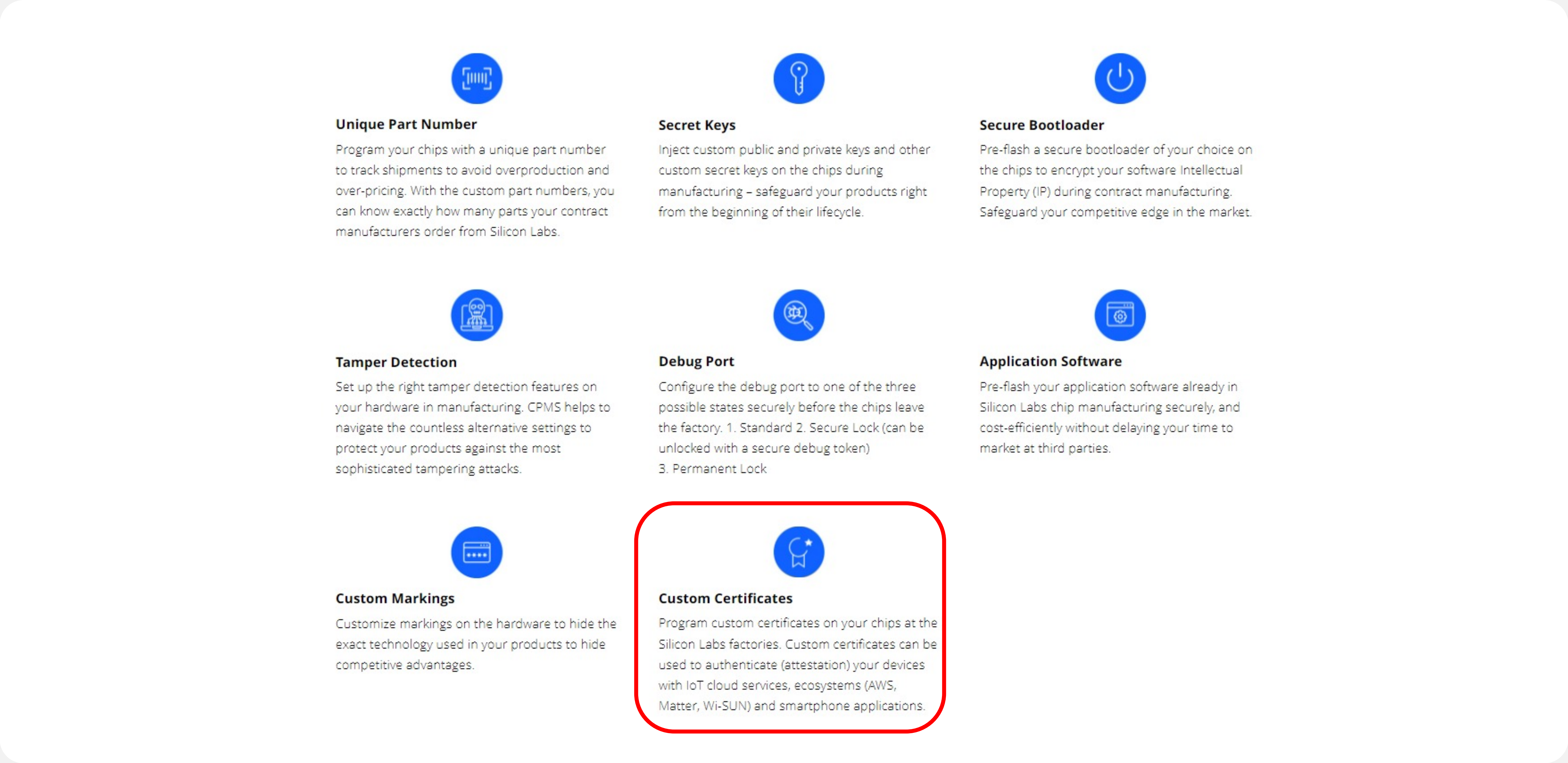
Certification path

Digital Guardian, Inc

*.zoom.us

- HTTPS uses certificate-based authentication (“lock” icon in Google Chrome)
- Chrome trusts the root certificate in the zoom.us certificate chain

Customization Options



Unique Part Number

Program your chips with a unique part number to track shipments to avoid overproduction and over-pricing. With the custom part numbers, you can know exactly how many parts your contract manufacturers order from Silicon Labs.

Secret Keys

Inject custom public and private keys and other custom secret keys on the chips during manufacturing – safeguard your products right from the beginning of their lifecycle.

Secure Bootloader

Pre-flash a secure bootloader of your choice on the chips to encrypt your software Intellectual Property (IP) during contract manufacturing. Safeguard your competitive edge in the market.

Tamper Detection

Set up the right tamper detection features on your hardware in manufacturing. CPMS helps to navigate the countless alternative settings to protect your products against the most sophisticated tampering attacks.

Debug Port

Configure the debug port to one of the three possible states securely before the chips leave the factory. 1. Standard 2. Secure Lock (can be unlocked with a secure debug token)
3. Permanent Lock

Application Software

Pre-flash your application software already in Silicon Labs chip manufacturing securely, and cost-efficiently without delaying your time to market at third parties.

Custom Markings

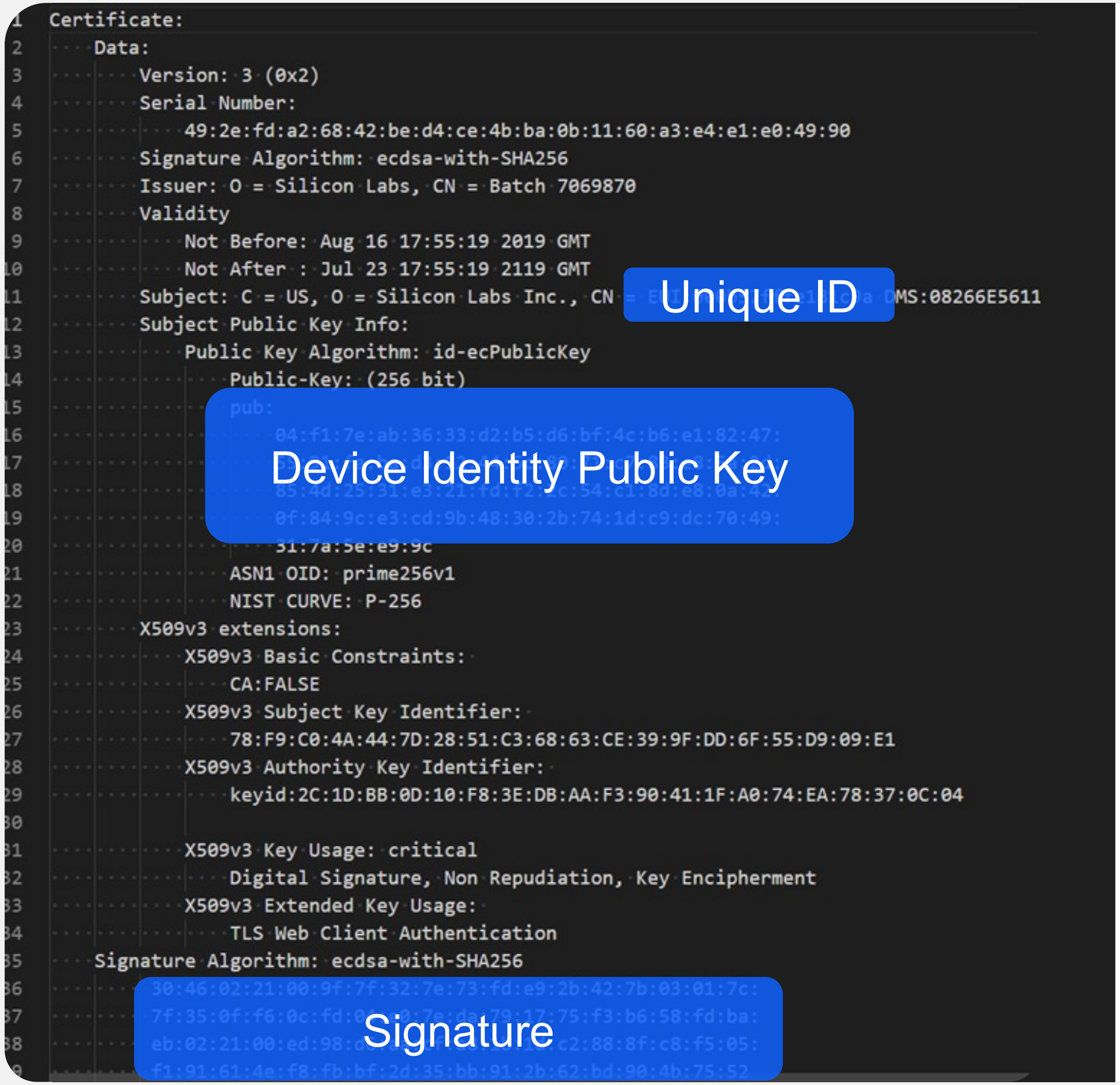
Customize markings on the hardware to hide the exact technology used in your products to hide competitive advantages.

Custom Certificates

Program custom certificates on your chips at the Silicon Labs factories. Custom certificates can be used to authenticate (attestation) your devices with IoT cloud services, ecosystems (AWS, Matter, Wi-SUN) and smartphone applications.

Elements of a Secure Identity

Requirements for a Secure Identity



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            49:2e:fd:a2:68:42:be:d4:ce:4b:ba:0b:11:60:a3:e4:e1:e0:49:90
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Silicon Labs, CN = Batch 7069870
        Validity
            Not Before: Aug 16 17:55:19 2019 GMT
            Not After : Jul 23 17:55:19 2119 GMT
        Subject: C = US, O = Silicon Labs Inc., CN = [illegible] DMS:08266E5611
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:f1:7e:ab:36:33:d2:b5:d6:bf:4c:b6:e1:82:47:
                    [illegible]
                    85:4d:25:31:e3:21:fd:f2:2c:54:c1:8d:e8:0a:42:
                    0f:84:9c:e3:cd:9b:48:30:2b:74:1d:c9:dc:70:49:
                    31:7a:5e:e9:9c
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                78:F9:C0:4A:44:7D:28:51:C3:68:63:CE:39:9F:DD:6F:55:D9:09:E1
            X509v3 Authority Key Identifier:
                keyid:2C:1D:BB:0D:10:F8:3E:DB:AA:F3:90:41:1F:A0:74:EA:78:37:0C:04

            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:9f:7f:32:7e:73:fd:e9:2b:42:7b:03:01:7c:
         7f:35:0f:f6:0c:fd:[illegible]:7e:da:79:17:75:f3:b6:58:fd:ba:
         eb:02:21:00:ed:98:[illegible]:c2:88:8f:c8:f5:05:
         f1:91:61:4e:f8:fb:bf:2d:35:bb:91:2b:62:bd:90:4b:75:52
```

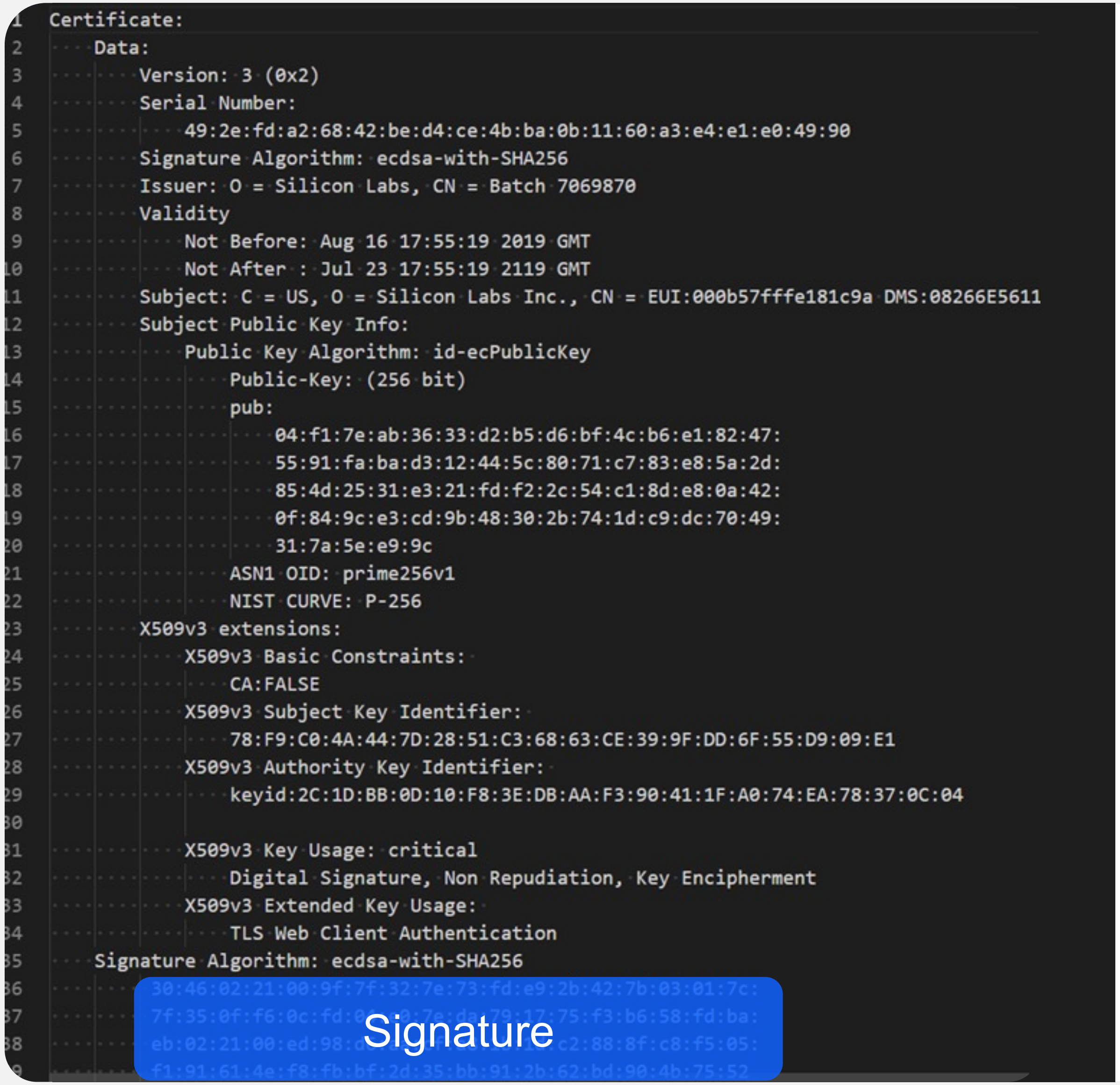
Unique ID

Device Identity Public Key

Signature

- **A Secure Identity should be:**
  - Unique for each instance of the product
  - Hard to fake
  - Hard to steal

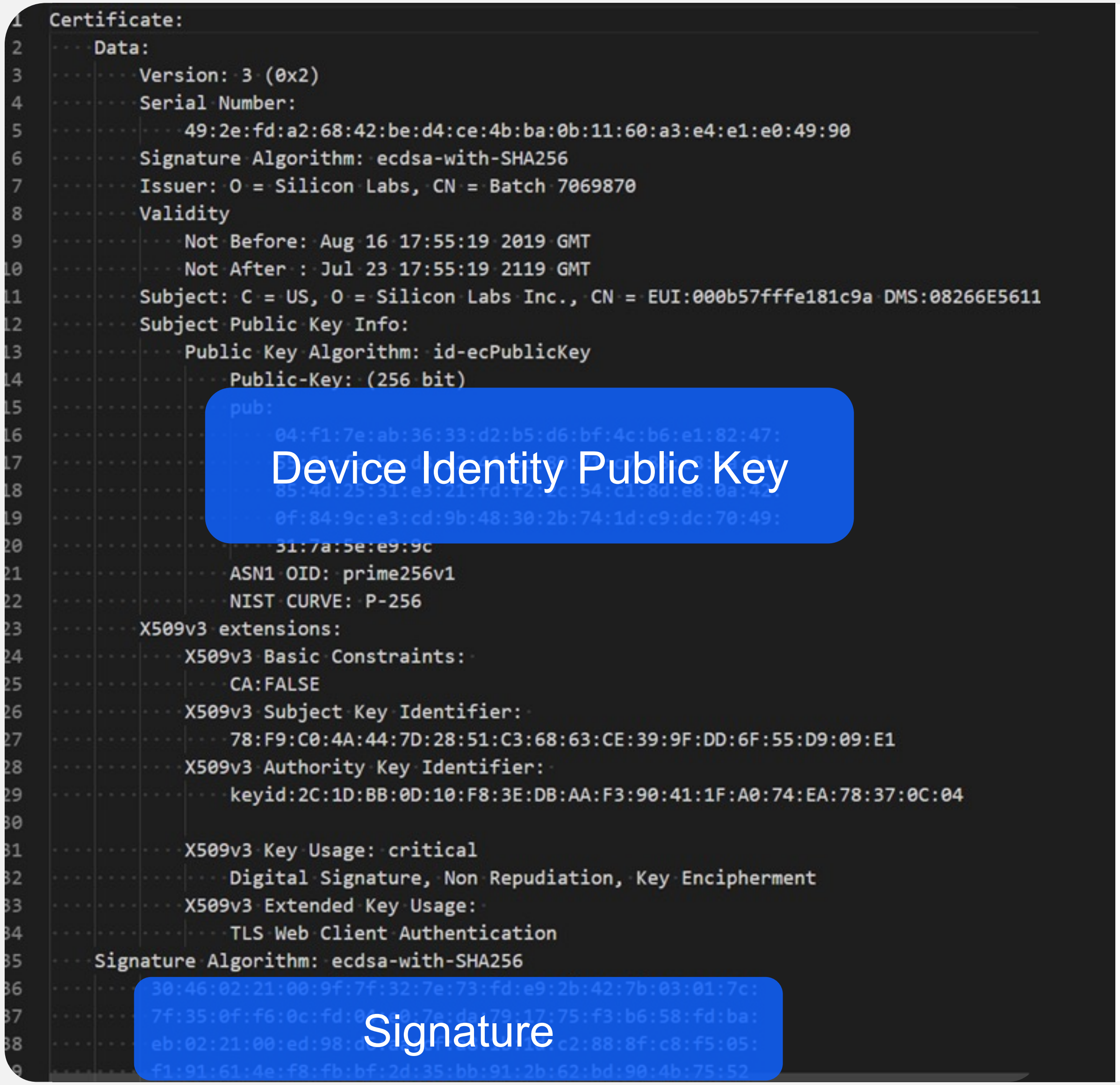
What a Device Certificate Looks Like (1)



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            49:2e:fd:a2:68:42:be:d4:ce:4b:ba:0b:11:60:a3:e4:e1:e0:49:90
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Silicon Labs, CN = Batch 7069870
        Validity
            Not Before: Aug 16 17:55:19 2019 GMT
            Not After : Jul 23 17:55:19 2119 GMT
        Subject: C = US, O = Silicon Labs Inc., CN = EUI:000b57fffe181c9a DMS:08266E5611
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:f1:7e:ab:36:33:d2:b5:d6:bf:4c:b6:e1:82:47:
                    55:91:fa:ba:d3:12:44:5c:80:71:c7:83:e8:5a:2d:
                    85:4d:25:31:e3:21:fd:f2:2c:54:c1:8d:e8:0a:42:
                    0f:84:9c:e3:cd:9b:48:30:2b:74:1d:c9:dc:70:49:
                    31:7a:5e:e9:9c
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                78:F9:C0:4A:44:7D:28:51:C3:68:63:CE:39:9F:DD:6F:55:D9:09:E1
            X509v3 Authority Key Identifier:
                keyid:2C:1D:BB:0D:10:F8:3E:DB:AA:F3:90:41:1F:A0:74:EA:78:37:0C:04

            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:9f:7f:32:7e:73:fd:e9:2b:42:7b:03:01:7c:
         7f:35:0f:f6:0c:fd:[illegible]:7e:da:79:17:75:f3:b6:58:fd:ba:
         eb:02:21:00:ed:98:[illegible]:c2:88:8f:c8:f5:05:
         f1:91:61:4e:f8:fb:bf:2d:35:bb:91:2b:62:bd:90:4b:75:52
```

- **Common attributes of a Device Certificate**
    - **Signature** of the Device Certificate

What a Device Certificate Looks Like (2)



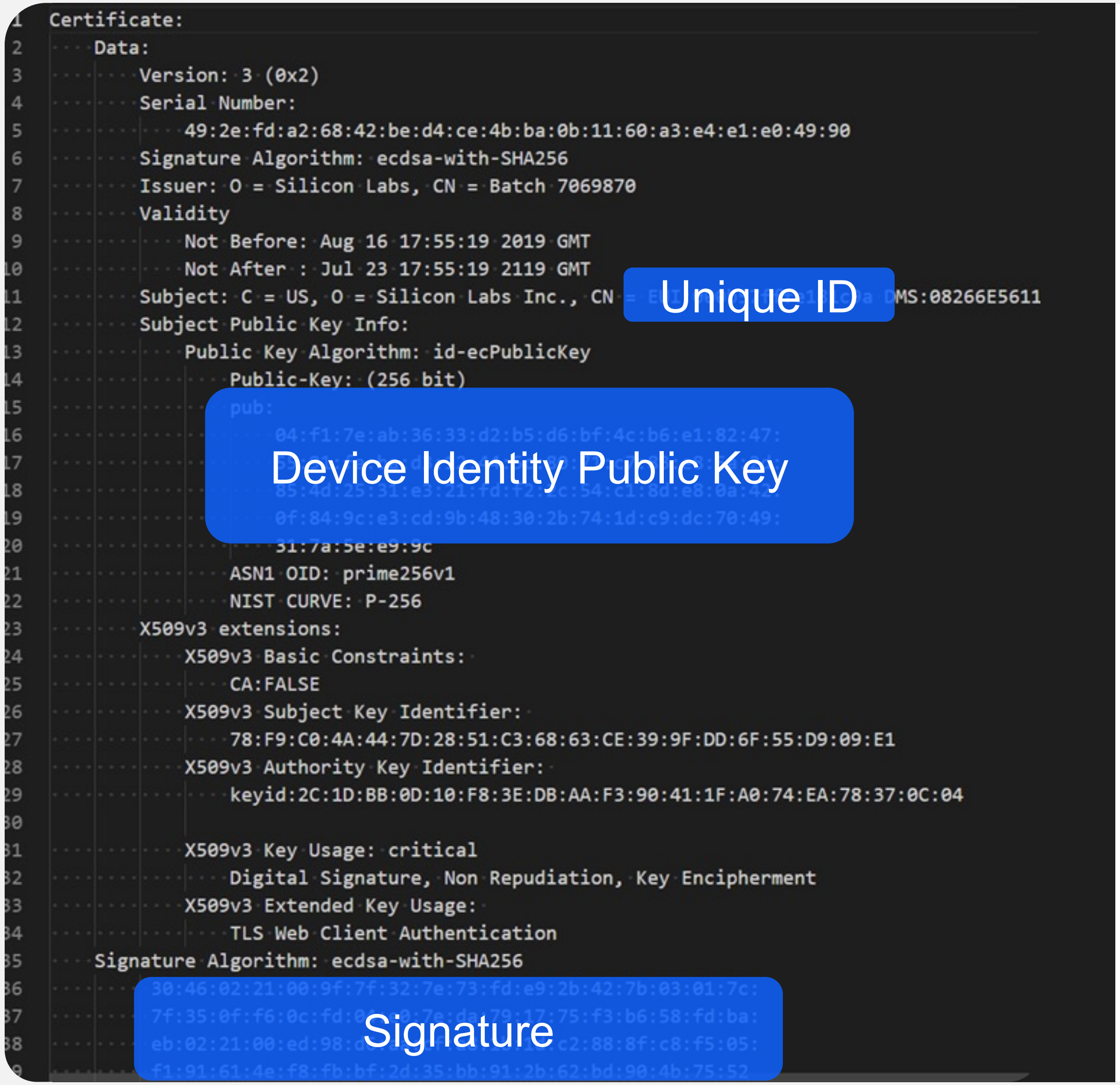
```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            49:2e:fd:a2:68:42:be:d4:ce:4b:ba:0b:11:60:a3:e4:e1:e0:49:90
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Silicon Labs, CN = Batch 7069870
        Validity
            Not Before: Aug 16 17:55:19 2019 GMT
            Not After : Jul 23 17:55:19 2119 GMT
        Subject: C = US, O = Silicon Labs Inc., CN = EUI:000b57fffe181c9a DMS:08266E5611
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:f1:7e:ab:36:33:d2:b5:d6:bf:4c:b6:e1:82:47:
                    [illegible]
                    85:4d:25:31:e3:21:fd:f2:2c:54:c1:8d:e8:0a:42:
                    0f:84:9c:e3:cd:9b:48:30:2b:74:1d:c9:dc:70:49:
                    31:7a:5e:e9:9c
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints: 
                CA:FALSE
            X509v3 Subject Key Identifier: 
                78:F9:C0:4A:44:7D:28:51:C3:68:63:CE:39:9F:DD:6F:55:D9:09:E1
            X509v3 Authority Key Identifier: 
                keyid:2C:1D:BB:0D:10:F8:3E:DB:AA:F3:90:41:1F:A0:74:EA:78:37:0C:04

            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage: 
                TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:9f:7f:32:7e:73:fd:e9:2b:42:7b:03:01:7c:
         7f:35:0f:f6:0c:fd:[illegible]:7e:da:79:17:75:f3:b6:58:fd:ba:
         eb:02:21:00:ed:98:[illegible]:c2:88:8f:c8:f5:05:
         f1:91:61:4e:f8:fb:bf:2d:35:bb:91:2b:62:bd:90:4b:75:52
```

Device Identity Public Key

Signature

- **Common attributes of a Device Certificate**
  - **Signature** of the Device Certificate
  - **Device Identity Public Key**

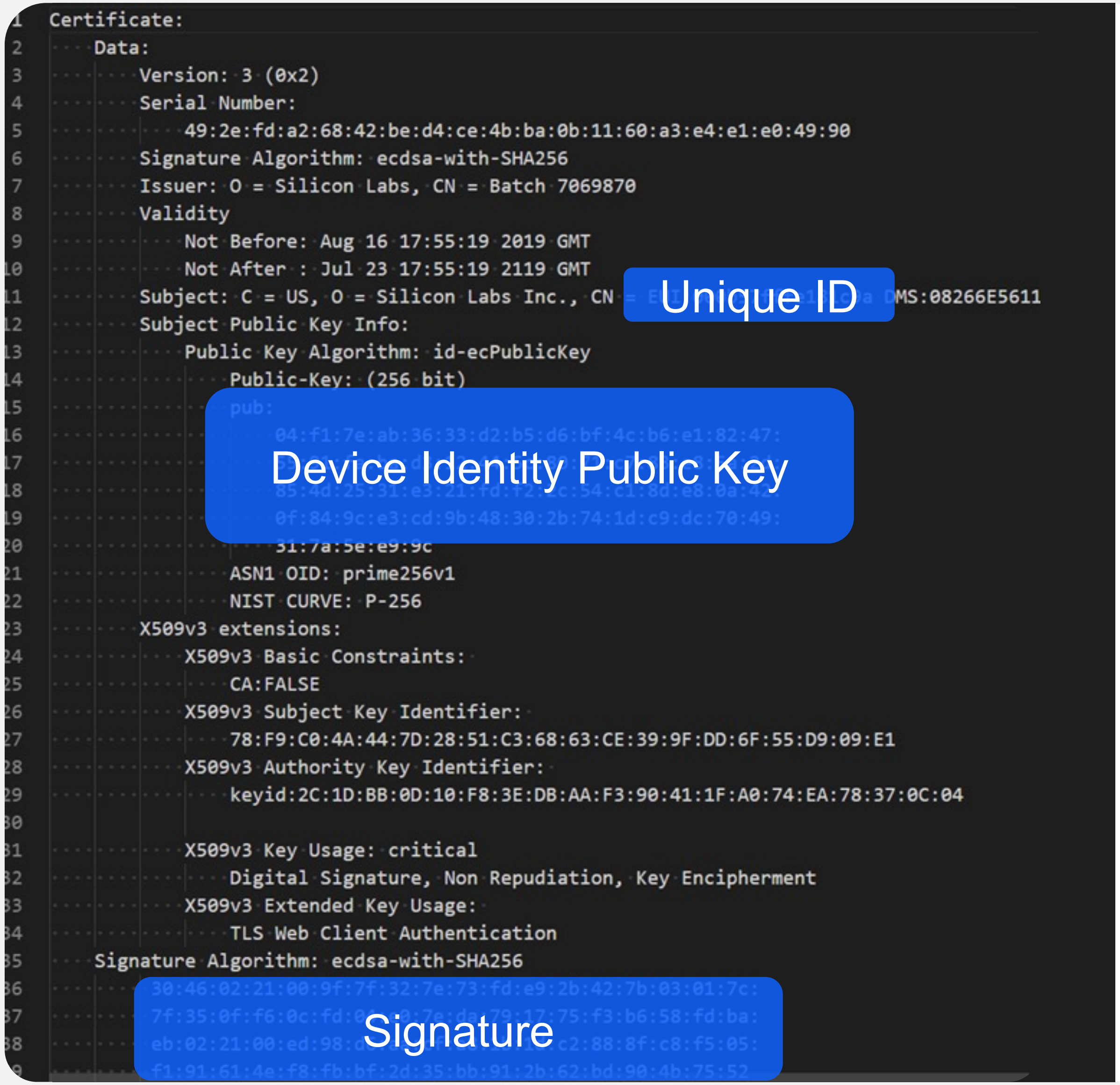
What a Device Certificate Looks Like (3)



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            49:2e:fd:a2:68:42:be:d4:ce:4b:ba:0b:11:60:a3:e4:e1:e0:49:90
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Silicon Labs, CN = Batch 7069870
        Validity
            Not Before: Aug 16 17:55:19 2019 GMT
            Not After : Jul 23 17:55:19 2119 GMT
        Subject: C = US, O = Silicon Labs Inc., CN = [illegible] DMS:08266E5611
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:f1:7e:ab:36:33:d2:b5:d6:bf:4c:b6:e1:82:47:
                    [illegible]
                    85:4d:25:31:e3:21:fd:f2:2c:54:c1:8d:e8:0a:42:
                    0f:84:9c:e3:cd:9b:48:30:2b:74:1d:c9:dc:70:49:
                    31:7a:5e:e9:9c
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                78:F9:C0:4A:44:7D:28:51:C3:68:63:CE:39:9F:DD:6F:55:D9:09:E1
            X509v3 Authority Key Identifier:
                keyid:2C:1D:BB:0D:10:F8:3E:DB:AA:F3:90:41:1F:A0:74:EA:78:37:0C:04

            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:9f:7f:32:7e:73:fd:e9:2b:42:7b:03:01:7c:
         7f:35:0f:f6:0c:fd:[illegible]:7e:da:79:17:75:f3:b6:58:fd:ba:
         eb:02:21:00:ed:98:[illegible]:c2:88:8f:c8:f5:05:
         f1:91:61:4e:f8:fb:bf:2d:35:bb:91:2b:62:bd:90:4b:75:52
```

- **Common attributes of a Device Certificate**
  - **Signature** of the Device Certificate
  - **Device Identity Public Key**
  - **Unique ID**
  - (optional) Custom information

What a Device Certificate Looks Like (4)

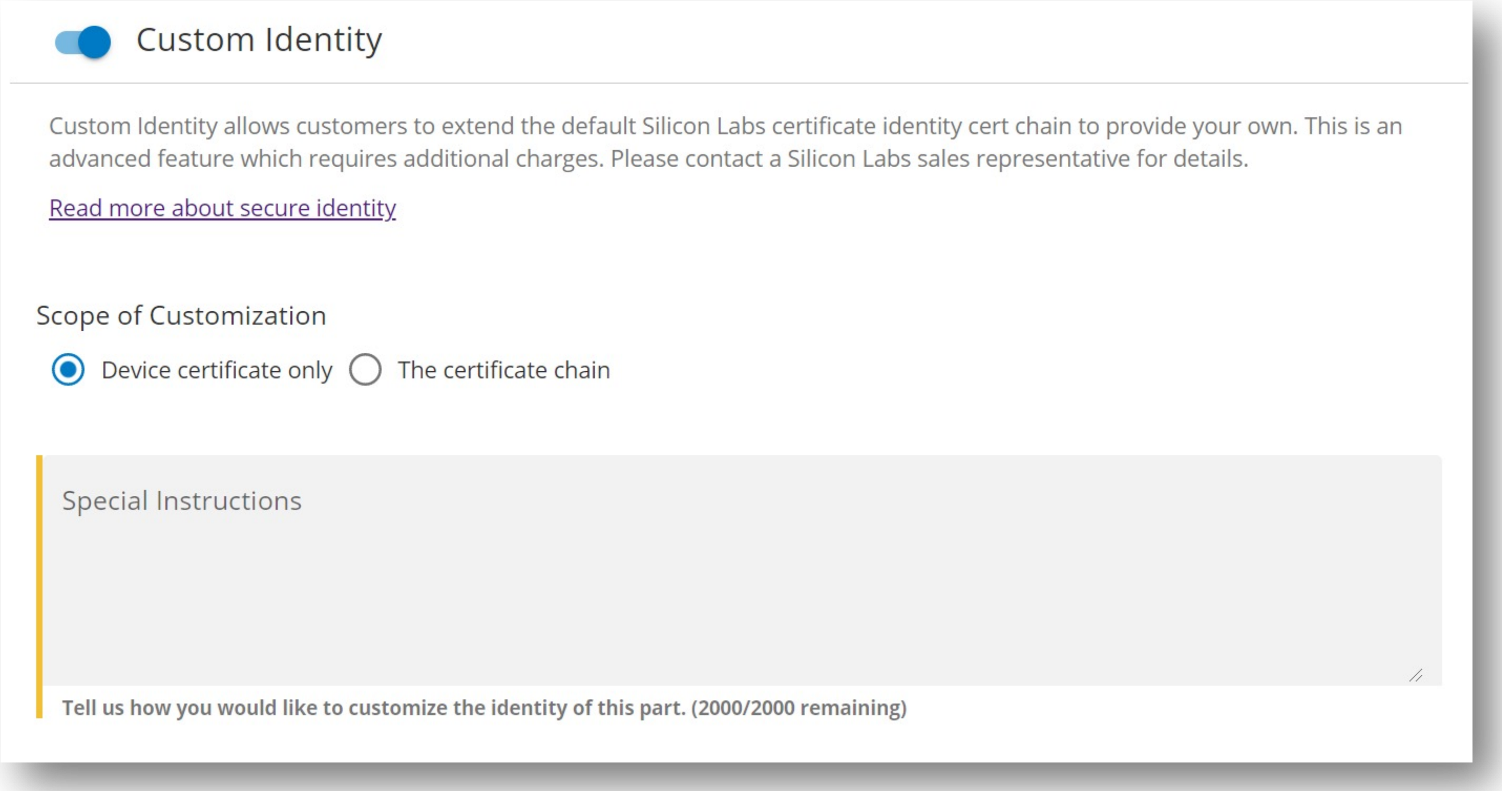


```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            49:2e:fd:a2:68:42:be:d4:ce:4b:ba:0b:11:60:a3:e4:e1:e0:49:90
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Silicon Labs, CN = Batch 7069870
        Validity
            Not Before: Aug 16 17:55:19 2019 GMT
            Not After : Jul 23 17:55:19 2119 GMT
        Subject: C = US, O = Silicon Labs Inc., CN = [illegible] DMS:08266E5611
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:f1:7e:ab:36:33:d2:b5:d6:bf:4c:b6:e1:82:47:
                    [illegible]
                    85:4d:25:31:e3:21:fd:f2:2c:54:c1:8d:e8:0a:42:
                    0f:84:9c:e3:cd:9b:48:30:2b:74:1d:c9:dc:70:49:
                    31:7a:5e:e9:9c
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                78:F9:C0:4A:44:7D:28:51:C3:68:63:CE:39:9F:DD:6F:55:D9:09:E1
            X509v3 Authority Key Identifier:
                keyid:2C:1D:BB:0D:10:F8:3E:DB:AA:F3:90:41:1F:A0:74:EA:78:37:0C:04

            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:9f:7f:32:7e:73:fd:e9:2b:42:7b:03:01:7c:
         7f:35:0f:f6:0c:fd:[illegible]:7e:da:79:17:75:f3:b6:58:fd:ba:
         eb:02:21:00:ed:98:[illegible]:c2:88:8f:c8:f5:05:
         f1:91:61:4e:f8:fb:bf:2d:35:bb:91:2b:62:bd:90:4b:75:52
```

- **Note that the Device Identity Private key isn't in the Device Certificate**
  - The **Private** key is securely stored inside the device, ideally in secure key storage

Custom Identity

- CPMS allows you to specify how to incorporate your own certificate chains into the Silicon Labs cert chain
- Cert chain implementations vary by use case, so certificate field details should be provided in the "Special Instructions" section



Custom Identity

Custom Identity allows customers to extend the default Silicon Labs certificate identity cert chain to provide your own. This is an advanced feature which requires additional charges. Please contact a Silicon Labs sales representative for details.

Read more about secure identity

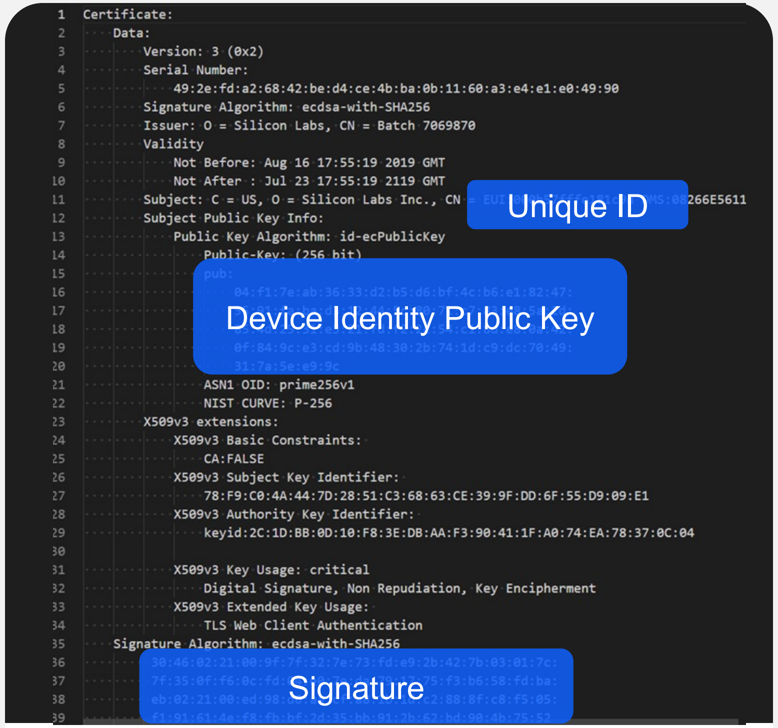
Scope of Customization

● Device certificate only ○ The certificate chain

Special Instructions

Tell us how you would like to customize the identity of this part. (2000/2000 remaining)

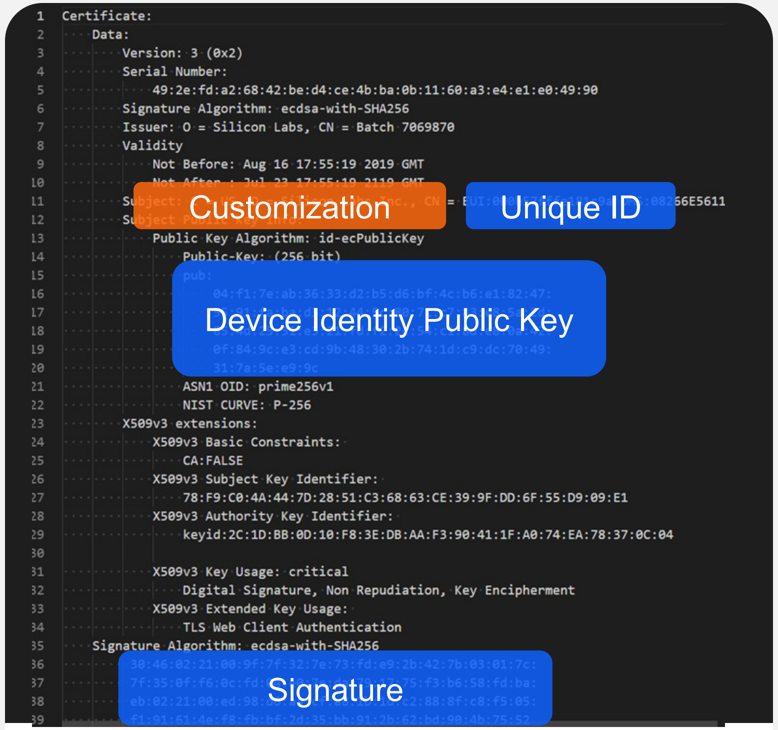
Use Cases for Standard and Customized Device Certificates



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            49:2e:fd:a2:68:42:be:d4:ce:4b:ba:0b:11:60:a3:e4:e1:e0:49:90
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Silicon Labs, CN = Batch 7069870
        Validity
            Not Before: Aug 16 17:55:19 2019 GMT
            Not After : Jul 23 17:55:19 2119 GMT
        Subject: C = US, O = Silicon Labs Inc., CN
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:f1:7e:ab:36:33:d2:b5:d6:bf:4c:b6:e1:82:47:
                    0f:84:9c:e3:cd:9b:48:30:2b:74:1d:c9:dc:70:49:
                    31:7a:5e:e9:9c
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                78:F9:C0:4A:44:7D:28:51:C3:68:63:CE:39:9F:DD:6F:55:D9:09:E1
            X509v3 Authority Key Identifier:
                keyid:2C:1D:BB:0D:10:F8:3E:DB:AA:F3:90:41:1F:A0:74:EA:78:37:0C:04

            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:9f:7f:32:7e:73:fd:e9:2b:42:7b:03:01:7c:
         f1:91:61:4e:f8:fb:bf:2d:35:bb:91:2b:62:bd:90:4b:75:52
```

STANDARD DEVICE CERTIFICATES

Protects against counterfeit **components**



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            49:2e:fd:a2:68:42:be:d4:ce:4b:ba:0b:11:60:a3:e4:e1:e0:49:90
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Silicon Labs, CN = Batch 7069870
        Validity
            Not Before: Aug 16 17:55:19 2019 GMT
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:f1:7e:ab:36:33:d2:b5:d6:bf:4c:b6:e1:82:47:
                    0f:84:9c:e3:cd:9b:48:30:2b:74:1d:c9:dc:70:49:
                    31:7a:5e:e9:9c
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                78:F9:C0:4A:44:7D:28:51:C3:68:63:CE:39:9F:DD:6F:55:D9:09:E1
            X509v3 Authority Key Identifier:
                keyid:2C:1D:BB:0D:10:F8:3E:DB:AA:F3:90:41:1F:A0:74:EA:78:37:0C:04

            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:9f:7f:32:7e:73:fd:e9:2b:42:7b:03:01:7c:
         f1:91:61:4e:f8:fb:bf:2d:35:bb:91:2b:62:bd:90:4b:75:52
```

CUSTOMIZED DEVICE CERTIFICATES

Protects against counterfeit **products**

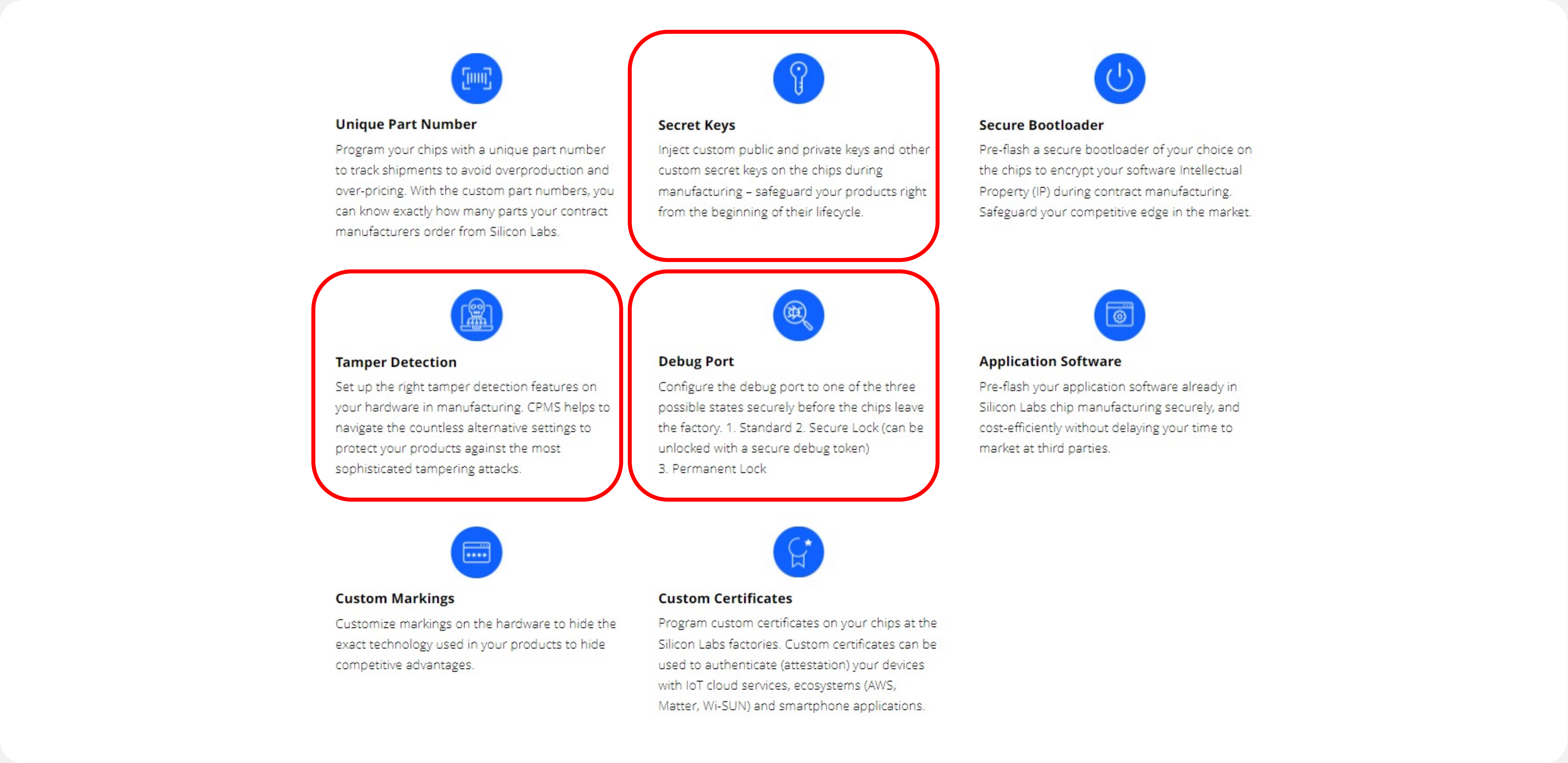
Protects against impersonation attacks

Supports streamlined commissioning

Use Case - Mutual App and Device Authentication using Public Key Certificates

- **Protects against counterfeit products and malicious apps**
- **An example of a Smartphone authenticating a Device**
  - Start by providing the certificate
  - Is the certificate authentic?
  - Is the certificate related to this device?
- **An example of a Device authenticating a Smartphone application or user**
  - Start by providing the certificate
  - Is the certificate authentic?
  - Is the certificate related to this app or user?

Customization Options



Unique Part Number

Program your chips with a unique part number to track shipments to avoid overproduction and over-pricing. With the custom part numbers, you can know exactly how many parts your contract manufacturers order from Silicon Labs.

Secret Keys

Inject custom public and private keys and other custom secret keys on the chips during manufacturing – safeguard your products right from the beginning of their lifecycle.

Secure Bootloader

Pre-flash a secure bootloader of your choice on the chips to encrypt your software Intellectual Property (IP) during contract manufacturing. Safeguard your competitive edge in the market.

Tamper Detection

Set up the right tamper detection features on your hardware in manufacturing. CPMS helps to navigate the countless alternative settings to protect your products against the most sophisticated tampering attacks.

Debug Port

Configure the debug port to one of the three possible states securely before the chips leave the factory. 1. Standard 2. Secure Lock (can be unlocked with a secure debug token)
3. Permanent Lock

Application Software

Pre-flash your application software already in Silicon Labs chip manufacturing securely, and cost-efficiently without delaying your time to market at third parties.

Custom Markings

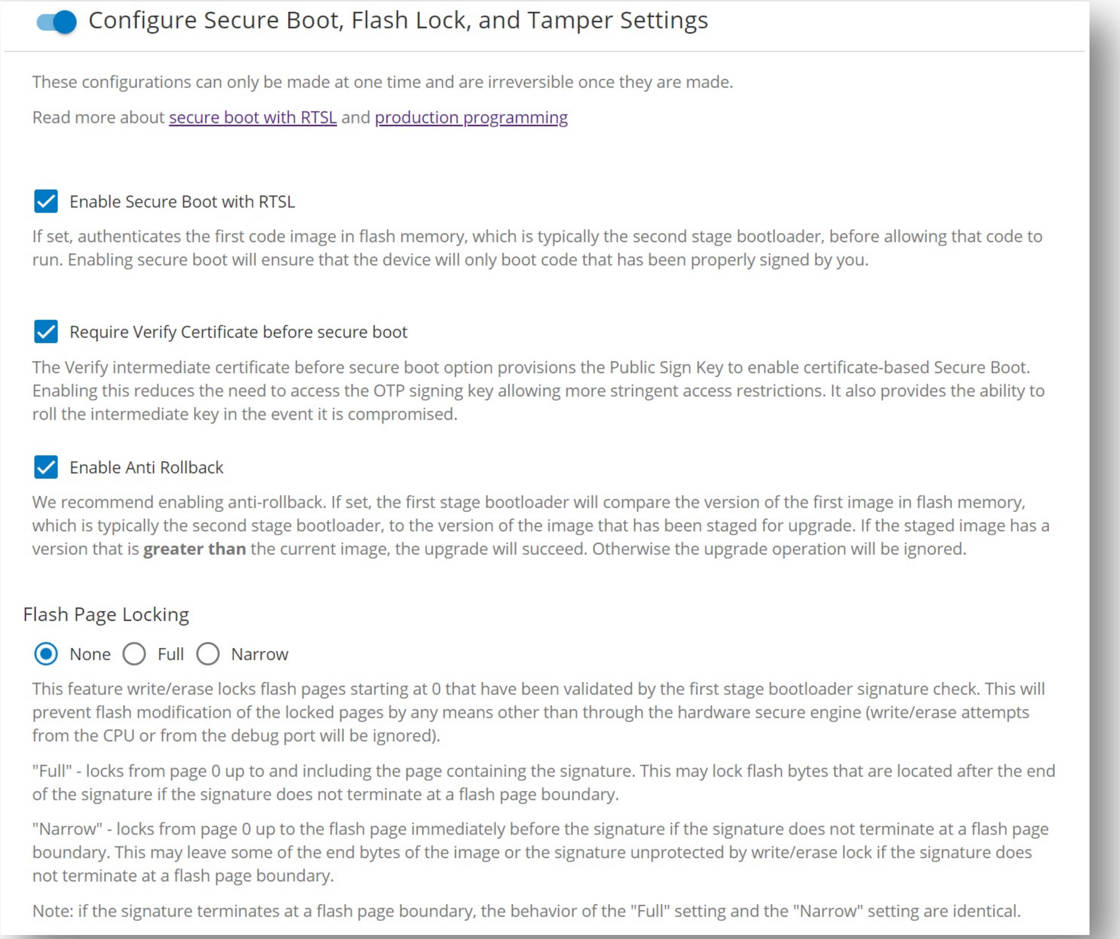
Customize markings on the hardware to hide the exact technology used in your products to hide competitive advantages.

Custom Certificates

Program custom certificates on your chips at the Silicon Labs factories. Custom certificates can be used to authenticate (attestation) your devices with IoT cloud services, ecosystems (AWS, Matter, Wi-SUN) and smartphone applications.

Initialize OTP Settings

- **CPMS allows you to configure OTP security settings. Since these settings are One Time Programmable, once set, they cannot be cleared**
  - *Enable Secure Boot* requires that any code on the device must have a valid signature or certificate in order to run. This ensures that only approved code runs on the device.
  - *Require Verify Certificate before secure boot* requires that certificates be used in the Secure Boot chain, rather than direct signing. This reduces the need to access the private key corresponding to the signing public key on the device.
  - *Enable Anti Rollback* prevents applications from "updating" to older (potentially vulnerable) versions of the firmware
  - *Flash Page Locking* prevents applications from writing to certain flash pages



Configure Secure Boot, Flash Lock, and Tamper Settings

These configurations can only be made at one time and are irreversible once they are made.

Read more about secure boot with RTSL and production programming

☑ Enable Secure Boot with RTSL

If set, authenticates the first code image in flash memory, which is typically the second stage bootloader, before allowing that code to run. Enabling secure boot will ensure that the device will only boot code that has been properly signed by you.

☑ Require Verify Certificate before secure boot

The Verify intermediate certificate before secure boot option provisions the Public Sign Key to enable certificate-based Secure Boot. Enabling this reduces the need to access the OTP signing key allowing more stringent access restrictions. It also provides the ability to roll the intermediate key in the event it is compromised.

☑ Enable Anti Rollback

We recommend enabling anti-rollback. If set, the first stage bootloader will compare the version of the first image in flash memory, which is typically the second stage bootloader, to the version of the image that has been staged for upgrade. If the staged image has a version that is **greater than** the current image, the upgrade will succeed. Otherwise the upgrade operation will be ignored.

Flash Page Locking

● None ○ Full ○ Narrow

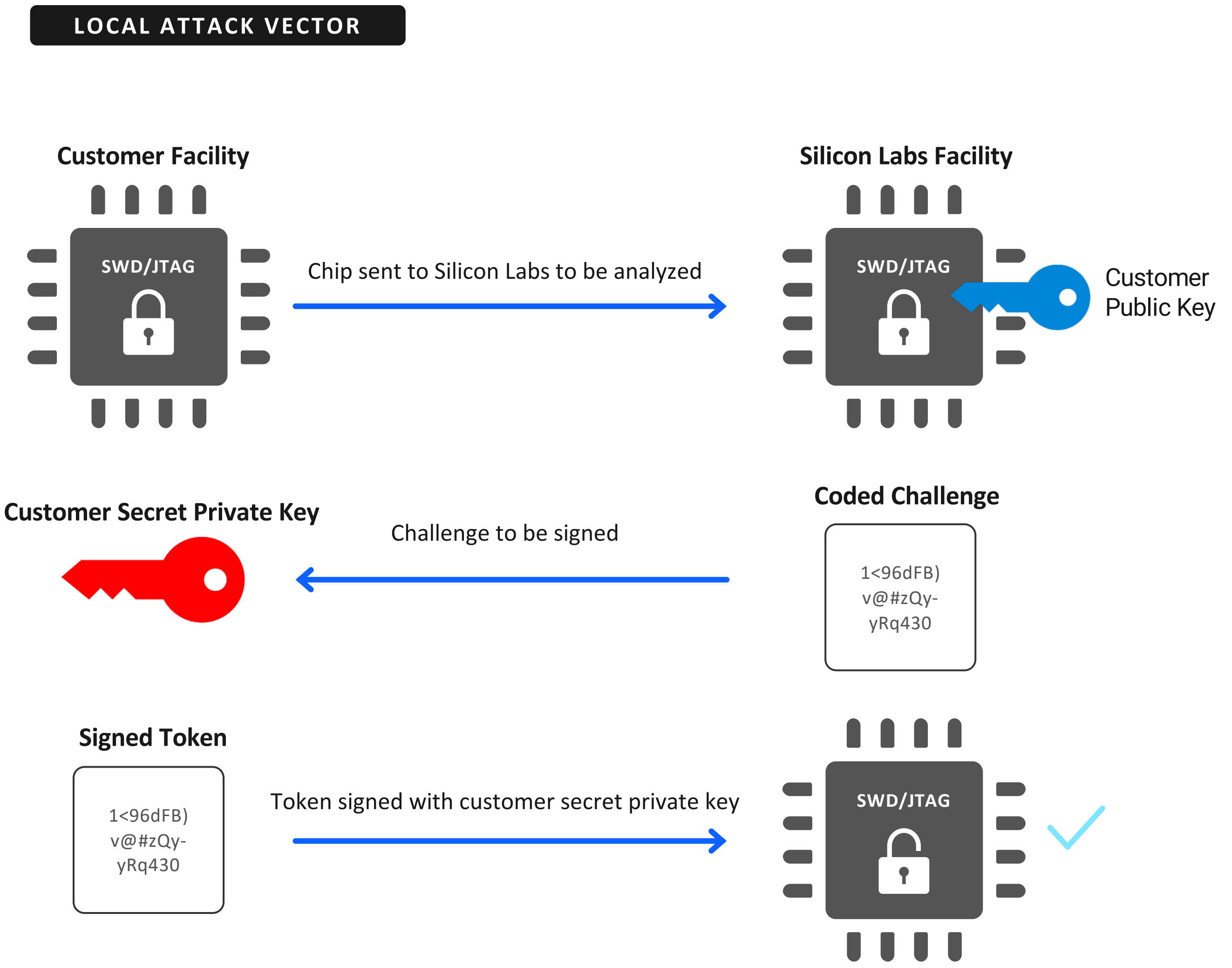
This feature write/erase locks flash pages starting at 0 that have been validated by the first stage bootloader signature check. This will prevent flash modification of the locked pages by any means other than through the hardware secure engine (write/erase attempts from the CPU or from the debug port will be ignored).

"Full" - locks from page 0 up to and including the page containing the signature. This may lock flash bytes that are located after the end of the signature if the signature does not terminate at a flash page boundary.

"Narrow" - locks from page 0 up to the flash page immediately before the signature if the signature does not terminate at a flash page boundary. This may leave some of the end bytes of the image or the signature unprotected by write/erase lock if the signature does not terminate at a flash page boundary.

Note: if the signature terminates at a flash page boundary, the behavior of the "Full" setting and the "Narrow" setting are identical.

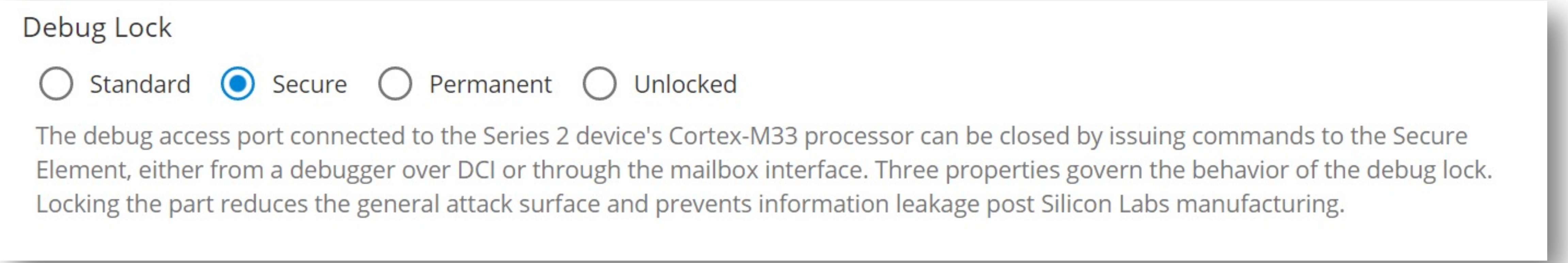
Secure Debug



- **Vulnerabilities**
  - Unlocked ports are a significant security vulnerability
  - Unlocking debug ports typically wipes the memory to protect IP but this limits device failure analysis capabilities
- **Secure Debug**
  - Lock the emulation port and use optional cryptographic tokens to unlock it allowing memory to remain intact

Debug Lock

- **CPMS allows you to select the state of the debug lock when the part is shipped to the CM**
- **Series 2 devices have 4 options for the debug lock:**
  - Permanent – the debug port is locked and cannot be unlocked
  - Standard – the debug port is locked, but it can be unlocked with a full flash erase
  - Secure – the debug port is locked, but it can be unlocked with a full flash erase or with a debug unlock token. The debug unlock token is verified with a public key stored in the device, and it only unlocks the debug port until the next reset
  - Unlocked – the debug port is unlocked



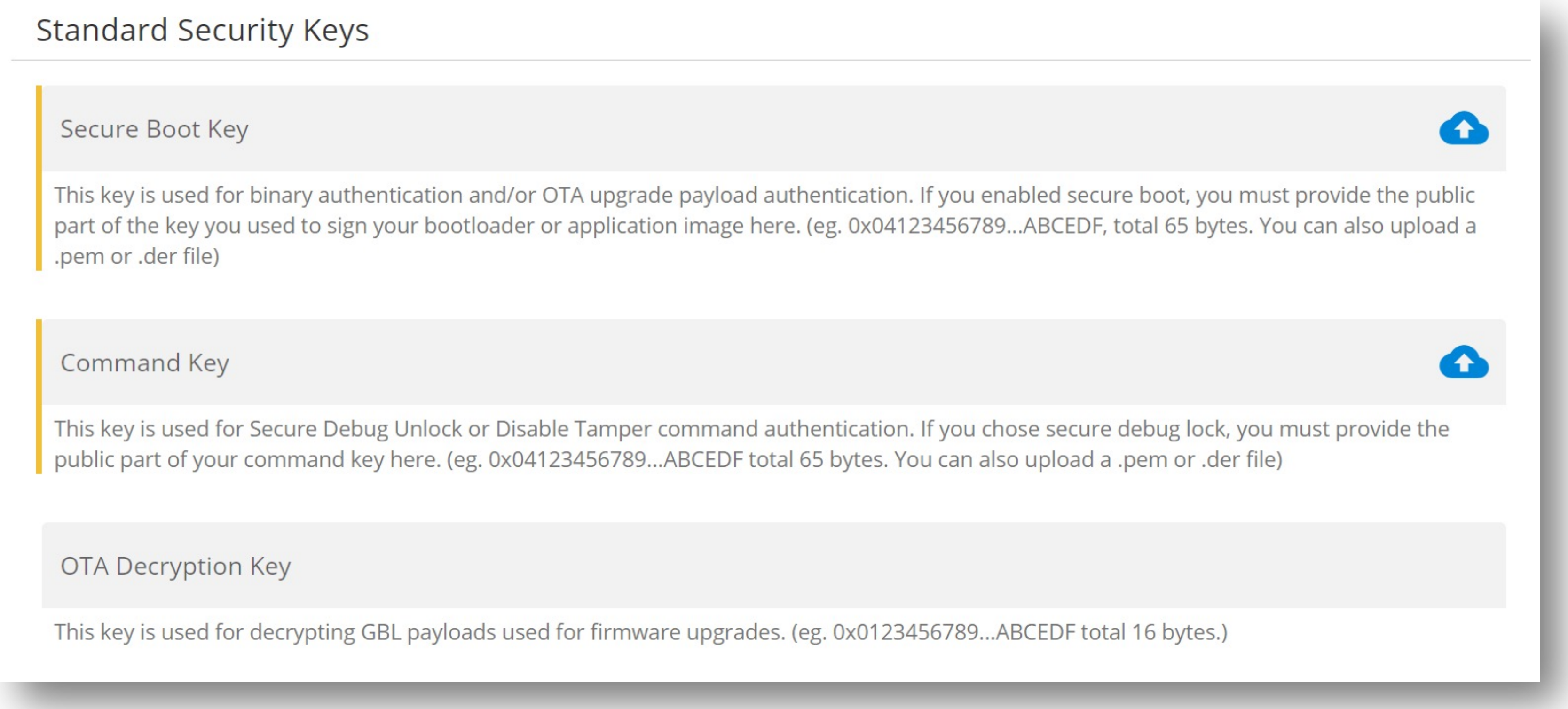
Debug Lock

○ Standard ● Secure ○ Permanent ○ Unlocked

The debug access port connected to the Series 2 device's Cortex-M33 processor can be closed by issuing commands to the Secure Element, either from a debugger over DCI or through the mailbox interface. Three properties govern the behavior of the debug lock. Locking the part reduces the general attack surface and prevents information leakage post Silicon Labs manufacturing.

Standard Security Keys

- **CPMS allows you to provision standard security keys into the device**
  - The *Secure Boot Key* is a public key used as the root of trust during the secure boot process to authenticate the firmware
  - The *Command Key* is a public key used to validate Secure Debug tokens
  - The *OTA Decryption Key* is a symmetric key used for decrypting GBL firmware upgrades



Standard Security Keys

Secure Boot Key

This key is used for binary authentication and/or OTA upgrade payload authentication. If you enabled secure boot, you must provide the public part of the key you used to sign your bootloader or application image here. (eg. 0x04123456789...ABCEDF, total 65 bytes. You can also upload a .pem or .der file)

Command Key

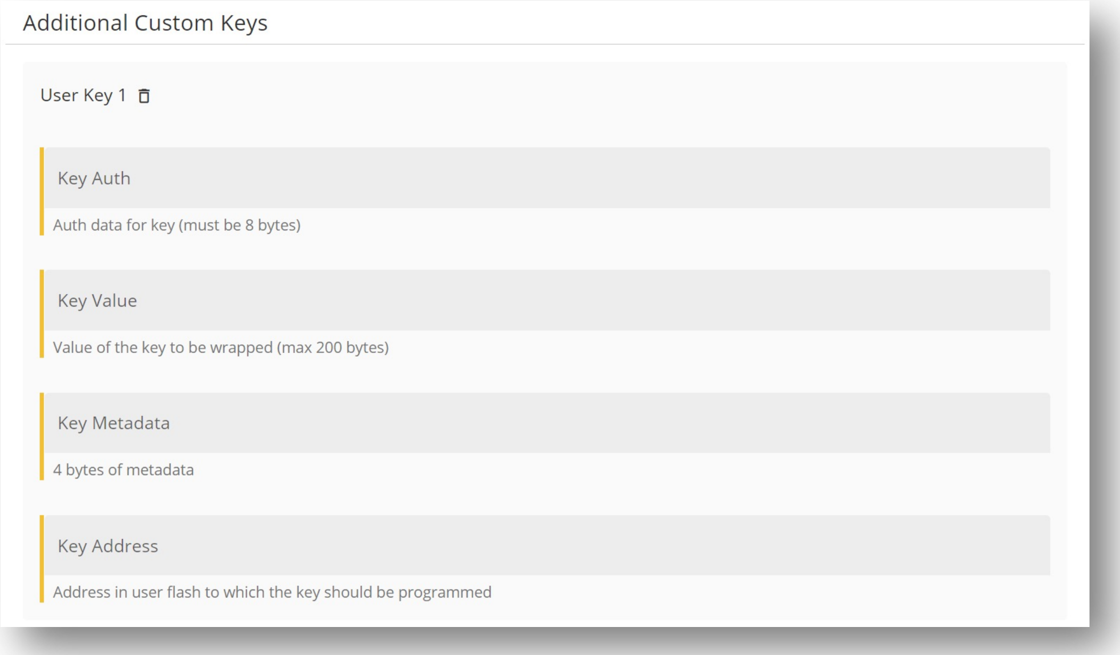
This key is used for Secure Debug Unlock or Disable Tamper command authentication. If you chose secure debug lock, you must provide the public part of your command key here. (eg. 0x04123456789...ABCEDF total 65 bytes. You can also upload a .pem or .der file)

OTA Decryption Key

This key is used for decrypting GBL payloads used for firmware upgrades. (eg. 0x0123456789...ABCEDF total 16 bytes.)

Custom Keys

- **In addition to the standard Security Keys, CPMS allows you to provision custom keys**
- **These custom keys will be wrapped by the Secure Element, then stored at a specified address in user flash**
- **To provision a custom key, you must provide:**
  - *Key Value* – the value of the key to be wrapped
  - *Key Address* – the address where the wrapped key will be stored
  - *Key Metadata* – a 32-bit key specification used by the SE (this value can be generated from a key descriptor using sli_se_key_to_keyspec)
  - *Key Auth* – an 8-byte password used to allow access to the wrapped key



Additional Custom Keys

User Key 1

Key Auth

Auth data for key (must be 8 bytes)

Key Value

Value of the key to be wrapped (max 200 bytes)

Key Metadata

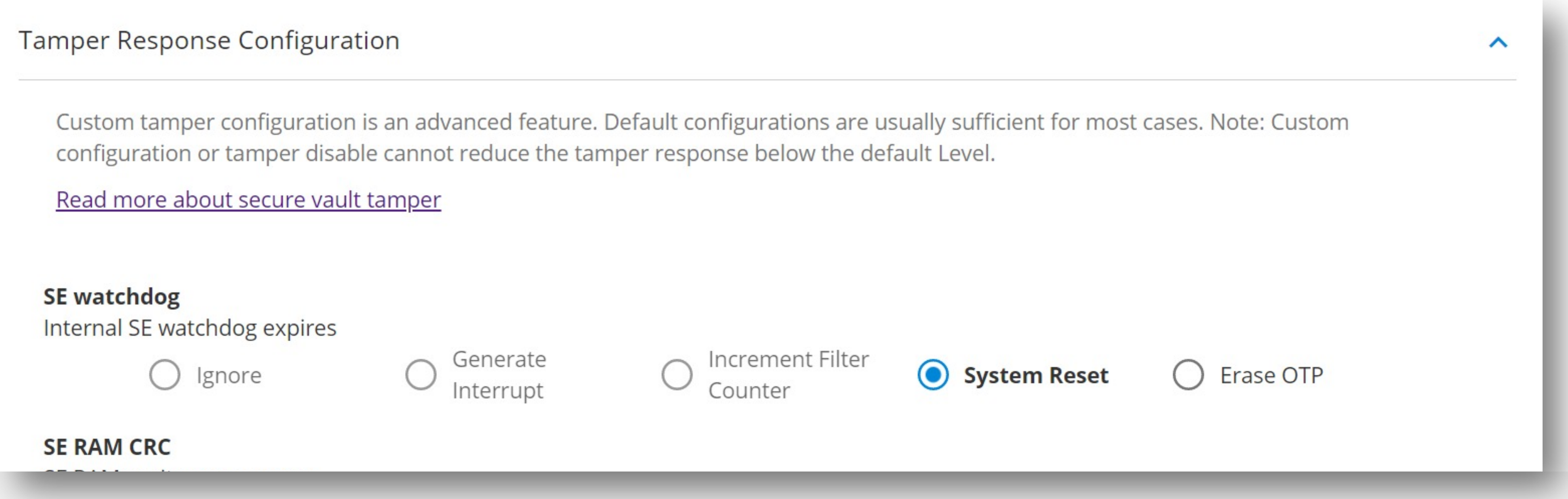
4 bytes of metadata

Key Address

Address in user flash to which the key should be programmed

Tamper Response Configuration

- **CPMS allows you to configure responses for 27 tamper sources**
- **When a tamper source is triggered, the device can choose to either:**
  - Ignore it
  - Generate an Interrupt
  - Increment the Filter Counter
  - Trigger a System Reset
  - Erase the OTP memory (note that this will make the device and all wrapped secrets unrecoverable. After this response, the device will no longer be able to boot.)



Tamper Response Configuration

Custom tamper configuration is an advanced feature. Default configurations are usually sufficient for most cases. Note: Custom configuration or tamper disable cannot reduce the tamper response below the default Level.

Read more about secure vault tamper

SE watchdog

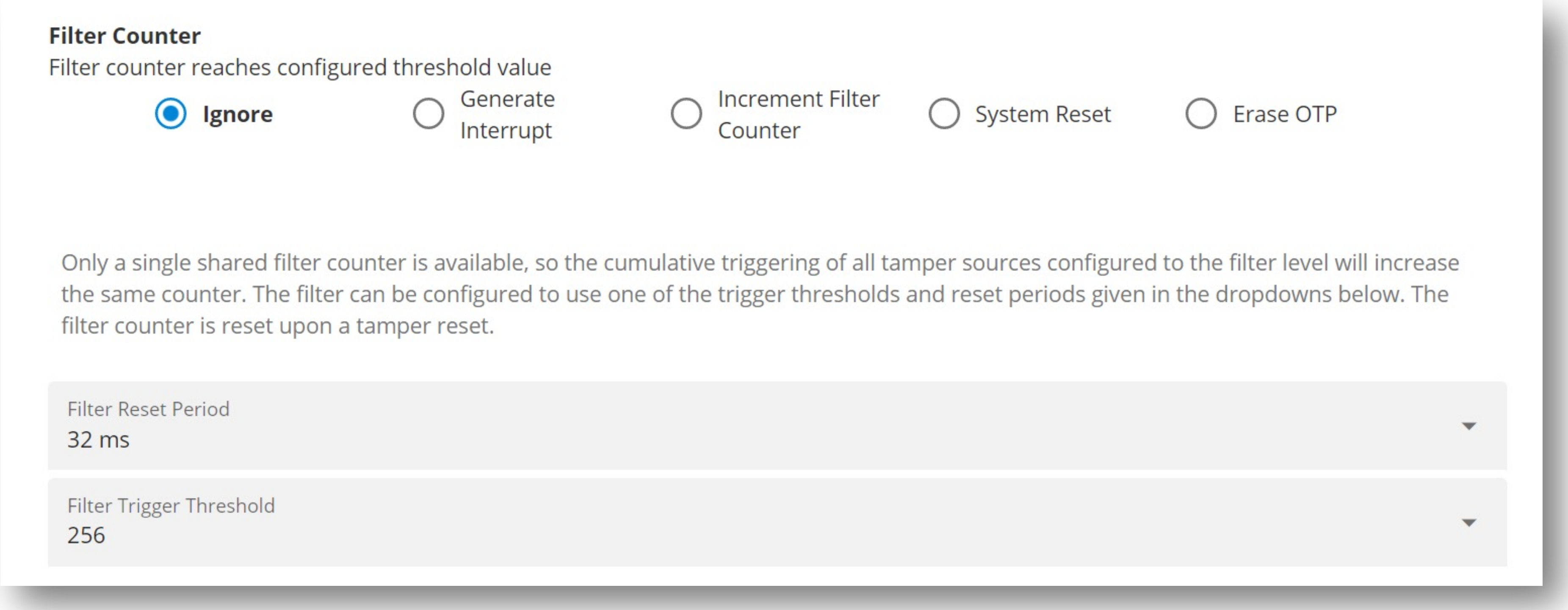
Internal SE watchdog expires

○ Ignore ○ Generate Interrupt ○ Increment Filter Counter ● **System Reset** ○ Erase OTP

SE RAM CRC

Tamper Response Configuration – Filter Counter

- Every tamper source has the option to increment the “Filter Counter”
- The Counter resets to 0 at a pre-defined period
- Once the Counter reaches a pre-determined Trigger Threshold, the Filter Counter tamper source is triggered
- Both the Reset Period and the Trigger Threshold can be configured in CPMS



Filter Counter

Filter counter reaches configured threshold value

● **Ignore** ○ Generate Interrupt ○ Increment Filter Counter ○ System Reset ○ Erase OTP

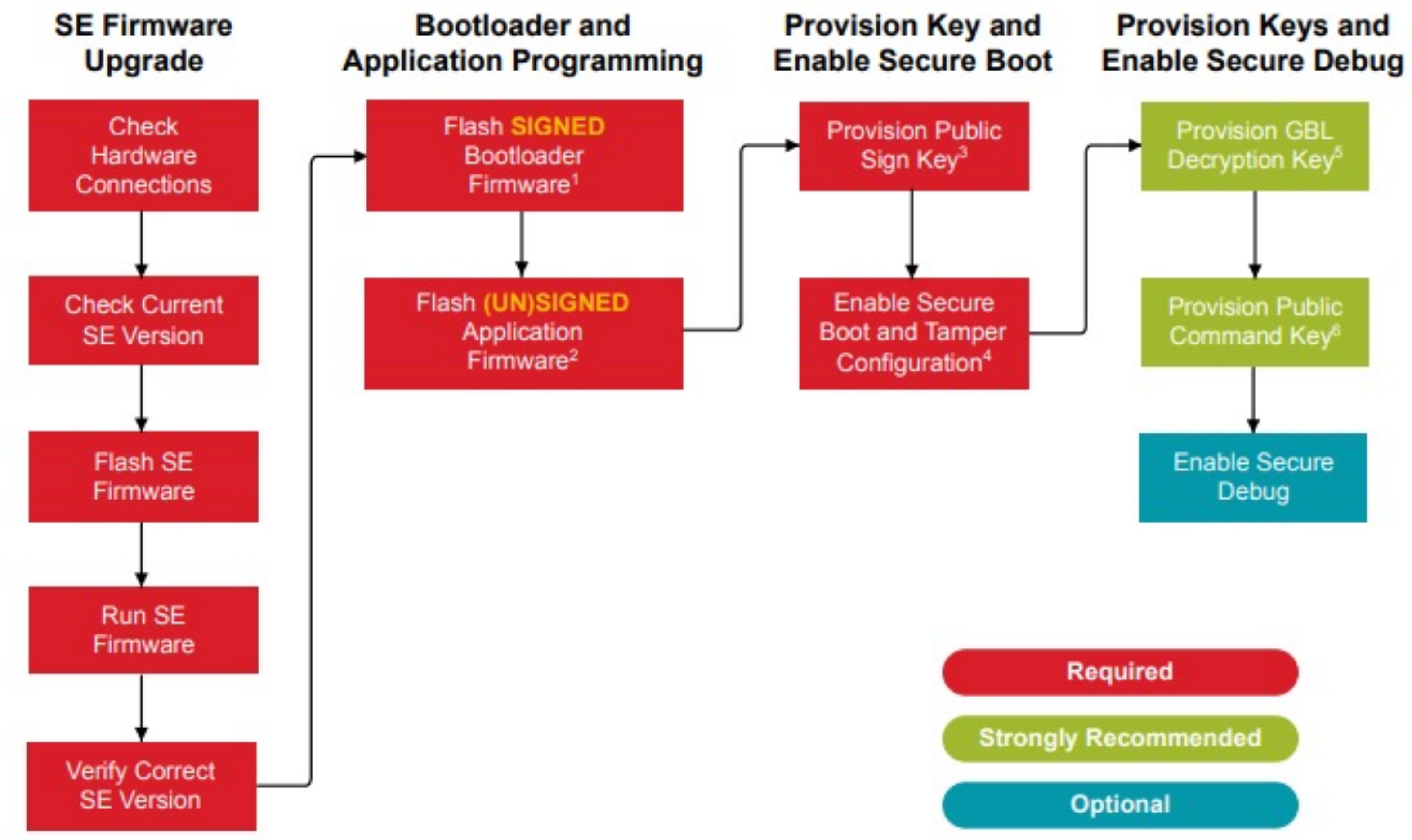
Only a single shared filter counter is available, so the cumulative triggering of all tamper sources configured to the filter level will increase the same counter. The filter can be configured to use one of the trigger thresholds and reset periods given in the dropdowns below. The filter counter is reset upon a tamper reset.

Filter Reset Period
32 ms

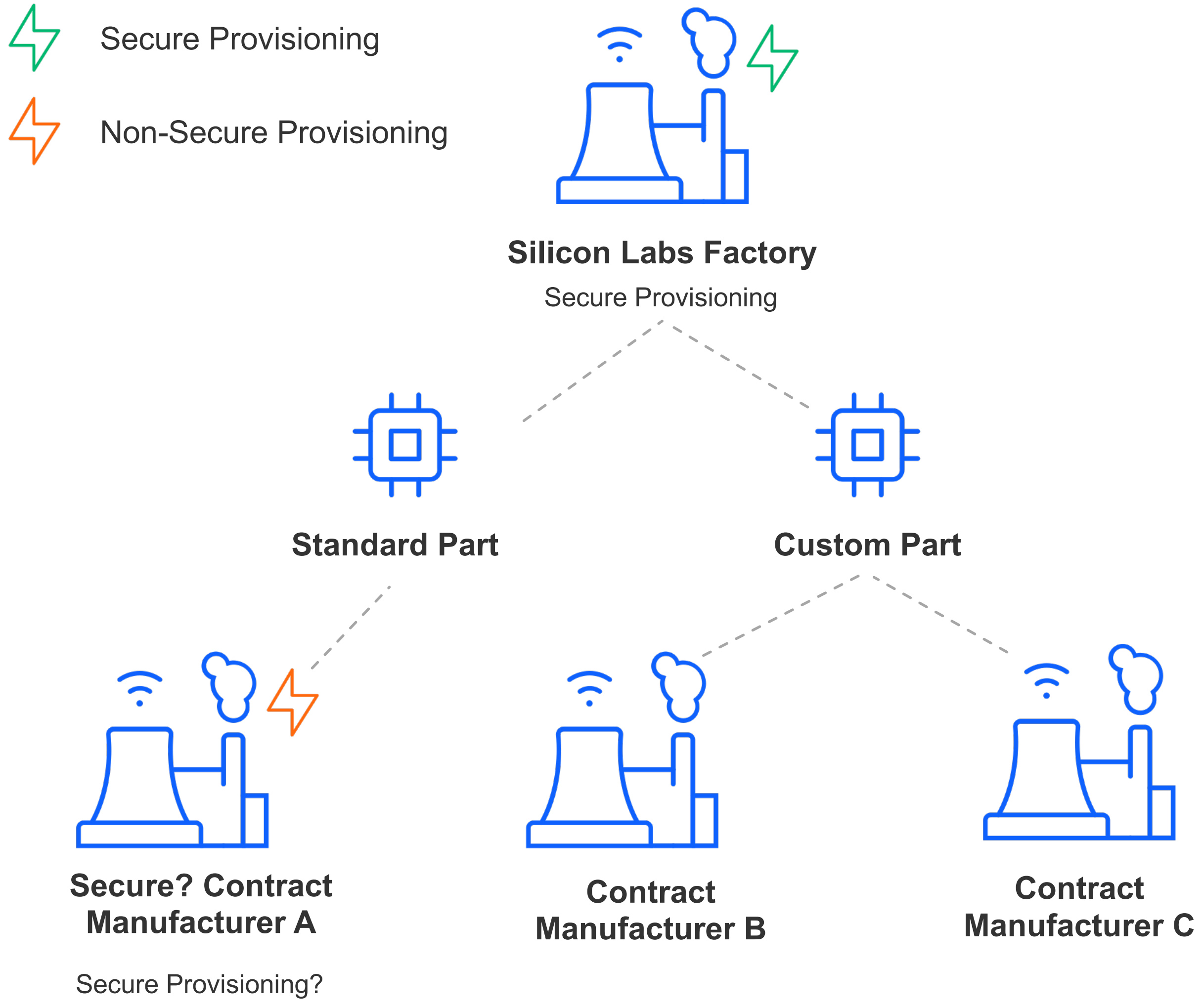
Filter Trigger Threshold
256

Why use CPMS?

- **Manufacturing processes around programming and provisioning are getting more and more complex**

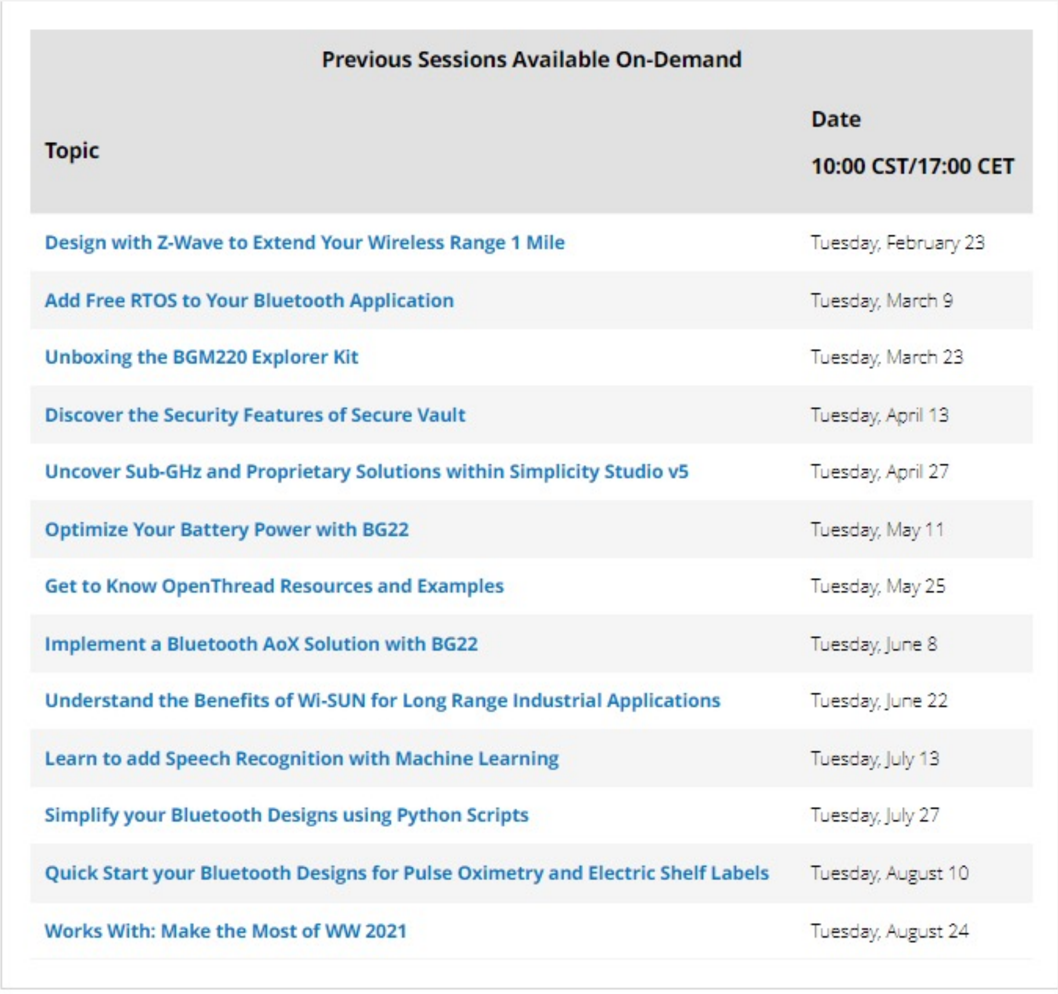

SE Firmware Upgrade
Check Hardware Connections
Check Current SE Version
Flash SE Firmware
Run SE Firmware
Verify Correct SE Version
Bootloader and Application Programming
Flash SIGNED Bootloader Firmware1
Flash (UN)SIGNED Application Firmware2
Provision Key and Enable Secure Boot
Provision Public Sign Key3
Enable Secure Boot and Tamper Configuration4
Provision Keys and Enable Secure Debug
Provision GBL Decryption Key5
Provision Public Command Key6
Enable Secure Debug
Required
Strongly Recommended
Optional

Why use CPMS?



- Available for Series 1 and Series 2 EFRx parts
- Easy to use web user interface
- User Private/Public Key Injection
- Security Settings:
  - Secure Debug Locked
  - Secure Boot Enable
  - Tamper Options Set
  - Anti-rollback Set
- Bootloader pre-flashed for protection of Software IP
- Secure Identity (Certificates) Injection
- Flash Programming

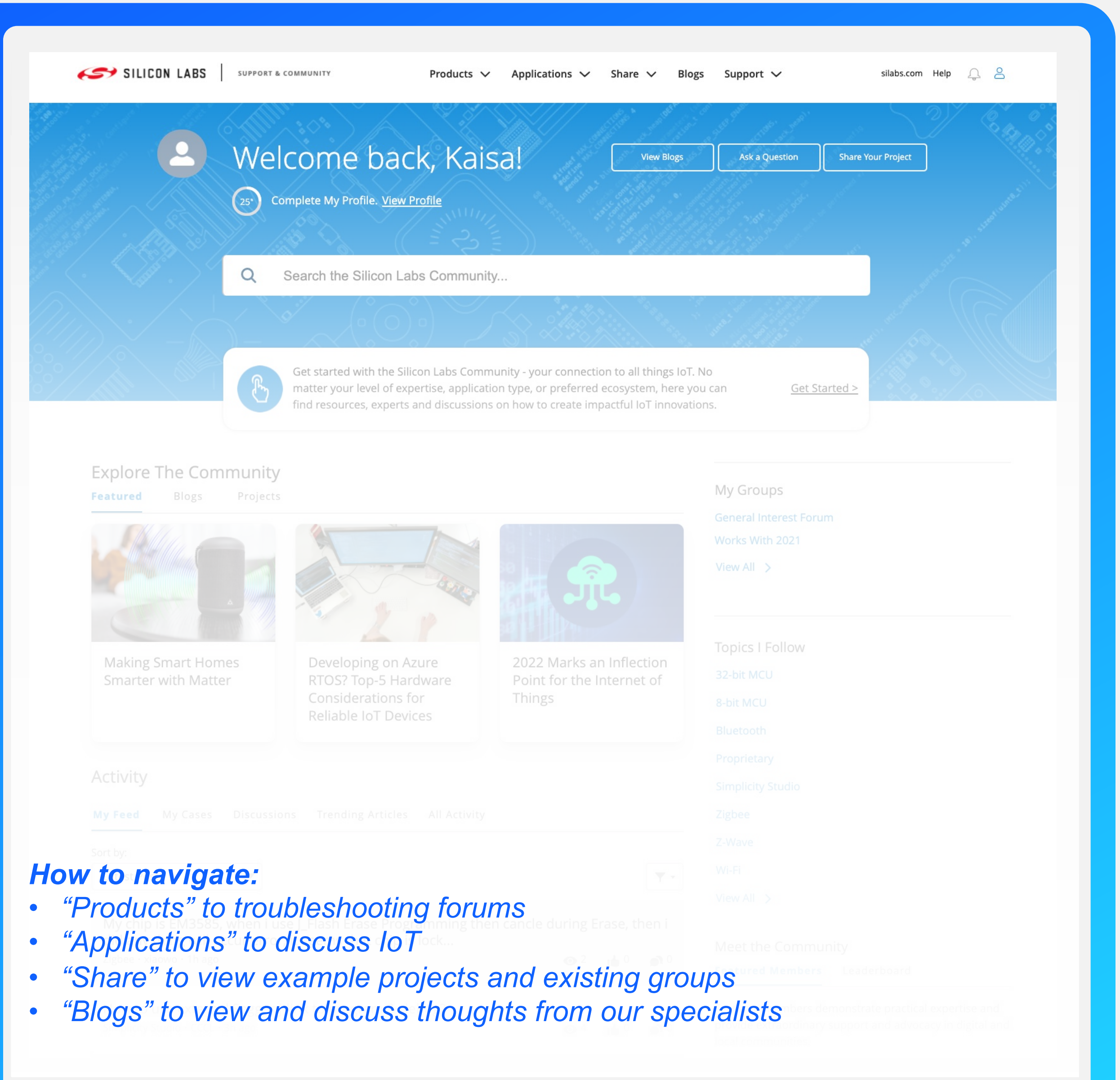
Miss a previous Tech Talk? Watch on Demand



Previous Sessions Available On-Demand	
Topic	Date 10:00 CST/17:00 CET
Design with Z-Wave to Extend Your Wireless Range 1 Mile	Tuesday, February 23
Add Free RTOS to Your Bluetooth Application	Tuesday, March 9
Unboxing the BGM220 Explorer Kit	Tuesday, March 23
Discover the Security Features of Secure Vault	Tuesday, April 13
Uncover Sub-GHz and Proprietary Solutions within Simplicity Studio v5	Tuesday, April 27
Optimize Your Battery Power with BG22	Tuesday, May 11
Get to Know OpenThread Resources and Examples	Tuesday, May 25
Implement a Bluetooth AoX Solution with BG22	Tuesday, June 8
Understand the Benefits of Wi-SUN for Long Range Industrial Applications	Tuesday, June 22
Learn to add Speech Recognition with Machine Learning	Tuesday, July 13
Simplify your Bluetooth Designs using Python Scripts	Tuesday, July 27
Quick Start your Bluetooth Designs for Pulse Oximetry and Electric Shelf Labels	Tuesday, August 10
Works With: Make the Most of WW 2021	Tuesday, August 24

https://www.silabs.com/about-us/events/wireless-connectivity-tech-talks-2021

Continue discussion in our community!



Welcome back, Kaisa!

Complete My Profile. View Profile

Search the Silicon Labs Community...

Get started with the Silicon Labs Community - your connection to all things IoT. No matter your level of expertise, application type, or preferred ecosystem, here you can find resources, experts and discussions on how to create impactful IoT innovations. Get Started >

Making Smart Homes Smarter with Matter

Developing on Azure RTOS? Top-5 Hardware Considerations for Reliable IoT Devices

2022 Marks an Inflection Point for the Internet of Things

How to navigate:

- *"Products" to troubleshooting forums*
- *"Applications" to discuss IoT*
- *"Share" to view example projects and existing groups*
- *"Blogs" to view and discuss thoughts from our specialists*

Stay tuned for our next Tech Talk series in 2022!

SILICON LABS

tech talks

Q&A

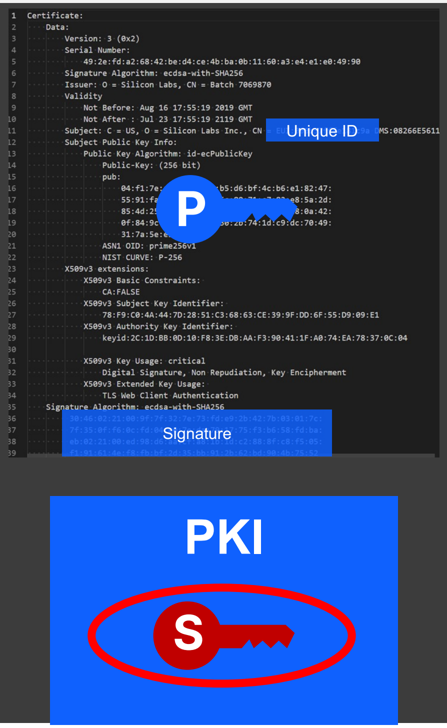
SILICON LABS

tech talks

THANK YOU

Why – Protecting the Keys

SILICON LABS
S
P
Unique ID
Signature



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            49:2e:fd:a2:68:42:be:d4:ce:4b:ba:0b:11:60:a3:e4:e1:e0:49:90
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: O = Silicon Labs, CN = Batch 7069870
        Validity
            Not Before: Aug 16 17:55:19 2019 GMT
            Not After : Jul 23 17:55:19 2119 GMT
        Subject: C = US, O = Silicon Labs Inc., CN                         MS:08266E5611
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:f1:7e:           b5:d6:bf:4c:b6:e1:82:47:
                    55:91:fa                         e8:5a:2d:
                    85:4d:25                          8:0a:42:
                    0f:84:9c            0:2b:74:1d:c9:dc:70:49:
                    31:7a:5e:e
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                78:F9:C0:4A:44:7D:28:51:C3:68:63:CE:39:9F:DD:6F:55:D9:09:E1
            X509v3 Authority Key Identifier:
                keyid:2C:1D:BB:0D:10:F8:3E:DB:AA:F3:90:41:1F:A0:74:EA:78:37:0C:04

            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:46:02:21:00:9f:7f:32:7e:73:fd:e9:2b:42:7b:03:01:7c:
         7f:35:0f:f6:0c:fd:04          17:75:f3:b6:58:fd:ba:
         eb:02:21:00:ed:98:d6:          1d:c2:88:8f:c8:f5:05:
         f1:91:61:4e:f8:fb:bf:2d:35:bb:91:2b:62:bd:90:4b:75:52
```

PROTECTING KEYS ON THE DEVICE

Use Secure Key Storage

Use TrustZone

Use obfuscation techniques

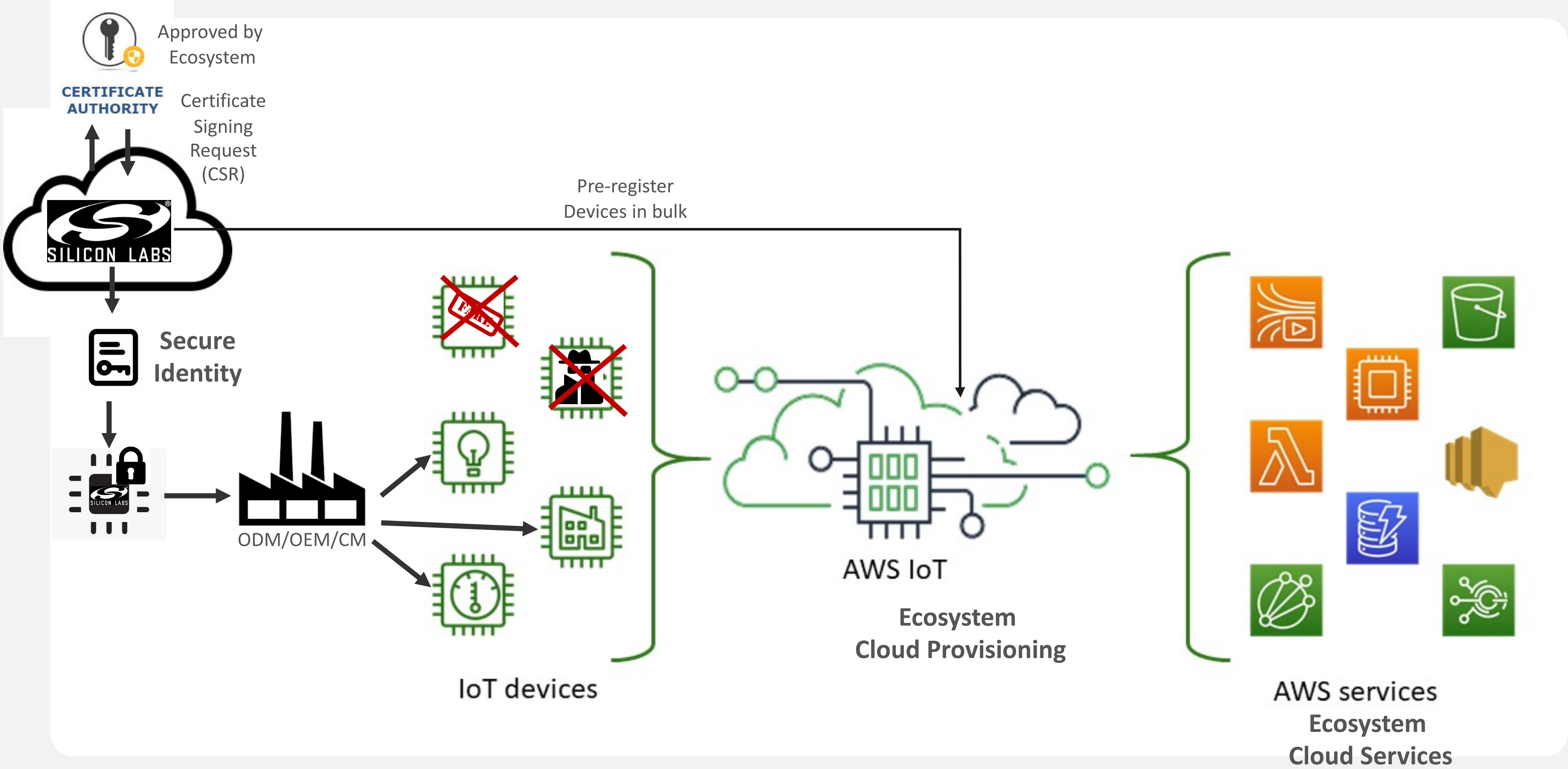
PROTECTING KEYS IN THE PKI

Use a Hardware Security Module

Physical security

Access controls and policies

Securing Ecosystems requires a Secure Identity to Authenticate Devices



Approved by Ecosystem

CERTIFICATE AUTHORITY

Certificate Signing Request (CSR)

Pre-register Devices in bulk

Secure Identity

ODM/OEM/CM

IoT devices

AWS IoT

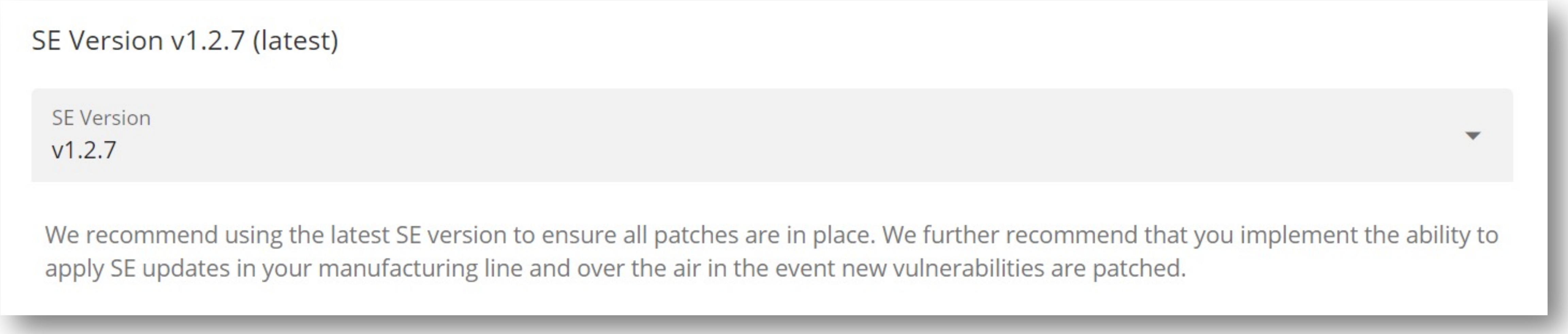
Ecosystem Cloud Provisioning

AWS services

Ecosystem Cloud Services

SE Version

- **CPMS allows you to select the Secure Element firmware version that is programmed into your custom parts**
  - We recommend using the latest SE version to ensure all patches are in place



SE Version v1.2.7 (latest)

SE Version: v1.2.7

We recommend using the latest SE version to ensure all patches are in place. We further recommend that you implement the ability to apply SE updates in your manufacturing line and over the air in the event new vulnerabilities are patched.