



UG103.05: IoT Endpoint Security Fundamentals

This document introduces the security concepts that must be considered when implementing an Internet of Things (IoT) system. Using the ioXt Alliance's eight security principles as a structure, the document clearly delineates the solutions Silicon Labs provides to support endpoint security and what you must do outside of the Silicon Labs framework. Where appropriate, Silicon Labs' approach to our own security is offered as an example. This document is designed for product developers and managers.

Silicon Labs' *Fundamentals* series covers topics that project managers, application designers, and developers should understand before beginning to work on an embedded networking solution using Silicon Labs chips, networking stacks such as EmberZNet PRO or Silicon Labs *Bluetooth*®, and associated development tools. The documents can be used as a starting place for anyone needing an introduction to developing wireless networking applications, or who is new to the Silicon Labs development environment.

KEY POINTS

- Based on the ioXt Alliance's Security Pledge
- Describes what manufacturers must do to have a secure product
- Reviews the tools and solutions Silicon Labs provides to support product security
- Provides references to other general and product-specific information

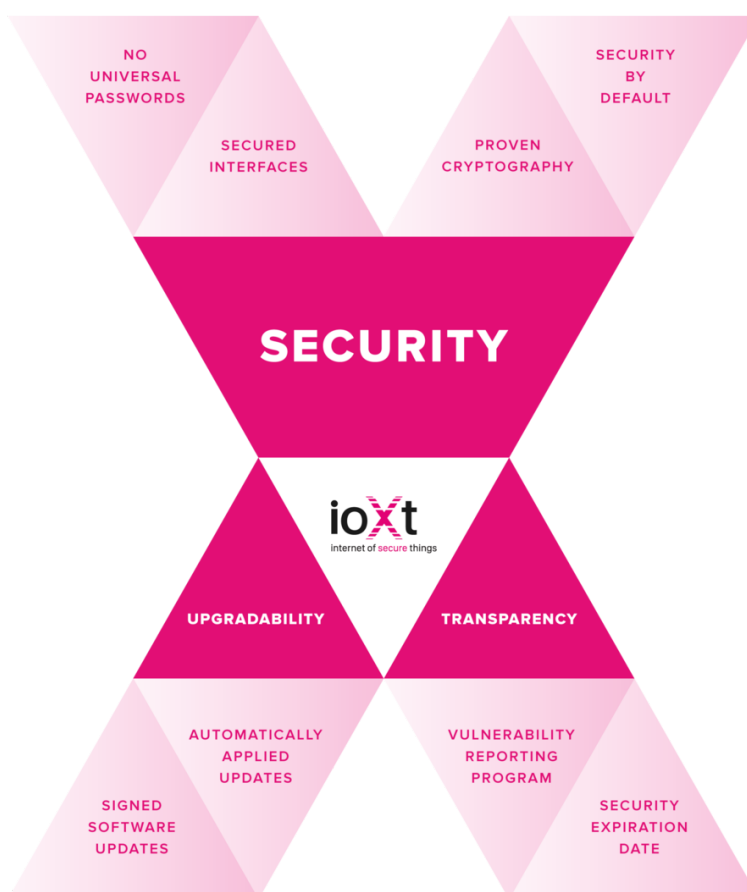
1 Introduction

Securing the IoT is challenging. It is also mission-critical. Threats are continuously evolving, and the demand on product developers to keep up can be burdensome – particularly in low-cost, resource-constrained IoT products. Protecting your product in a connected world is a necessity, as customer data and modern online business models are increasingly targets for costly hacks that jeopardize end-user privacy and corporate brand damage. Silicon Labs is committed to working with the security community, customers, and other experts to bring state-of-the-art technology to help protect your connected portfolio.

Silicon Labs is a member of the ioXt (Internet of Secure Things) Alliance. The ioXt Alliance was formed to bring together wireless carriers, leading consumer product manufacturers, standards groups, compliance labs and government organizations to align baseline security requirements, to set the stage for testing and compatibility certification, and to work together building global standards for the IoT world.



The ioXt alliance has produced the ioXt Security Pledge (<https://www.ioxtalliance.org/s/ioXt-SecurityPledge-booklet-final.pdf>) The pledge covers eight principles in the areas of Security, Upgradability, and Transparency. Silicon Labs has adopted these principles in our own operations as well as in the products we provide. Our approach to these principles is described in this document.



The above image and all pledge language is reproduced from *The ioXt Security Pledge: 8 Principles for Consumer Product Design and Manufacturing to Ensure Security, Upgradability & Transparency* (2019).

2 No Universal Passwords

The product shall not have a universal password; unique security credentials will be required for operation. Universal passwords allow an attacker to easily gain access to any device. Therefore, products shall either have a unique password or require the user to enter a new password immediately upon first use.

It is your responsibility to ensure that your product enforces the creation of a unique password before activation.

Silicon Labs' products are designed to be configured by the manufacturer before being delivered to customers, and therefore passwords are outside of our scope. However, Silicon Labs tools are designed to support the various levels of security provided by the protocol in question. Most protocols offer different security levels, with tradeoffs between security level and other features such as ease of network formation. You need to review and decide on the level required by your application. For example:

- The EmberZNet Pro SDK supports a highly secure centralized trust-center-controlled method that replaces a device's factory-programmed link key with a key that is unique to each device on the network.
- Z-Wave 700 products come with a factory-programmed unique S2 keypair on first power-up, and support SmartStart commissioning through a package QR code containing the public key.
- Bluetooth options range from an unsecured "Just Works" approach to a LE Secure Connections Pairing model. Application designers can implement additional device authentication methods, such as through the companion smartphone app, to help ensure secure pairing even for devices without a user interface.

3 Secured Interfaces

All product interfaces shall be appropriately secured by the manufacturer.

The interfaces to be secured will vary by product configuration. For example, in an NCP topology the NCP interface must be secured. Debug interfaces should always be locked. Wireless interfaces should be secured by using strong pairing and commissioning methods and by enabling encrypted and authenticated transmissions.

While securing the interfaces is in the end your responsibility, Silicon Labs provides the tools to enable that security.

Both Series 1 and Series 2 devices are designed to support securing debug access. For Series 1 devices, that functionality is provided through writing a Debug Lock word to the device. Unlocking the device erases the main application and the key material stored in the Lockbits page. For Series 2 devices, securing debug access is done through the device's Secure Engine. Both allow the developer to lock the debug port itself. See *UG489: Silicon Labs Gecko Bootloader User's Guide for GSDK 4.0 and Higher* or *UG266: Silicon Labs Gecko Bootloader User's Guide for GSDK 3.2 and Lower* for an overview of securing debug access, and *AN1190: EFR32xG21 Secure Debug* for details on the Series 2 implementation. *UG104: Testing and Debugging Applications for the Silicon Labs EFR32MG Platforms* provides an overview of the various application testing stages and the debug access (hardware and software) required in each.

For more information on Wireless interface security in the different protocols, see the following:

- *AN1233: Zigbee Security*
- *UG103.14: Bluetooth® LE Fundamentals* and relevant KBAs
- *AN1037: Apple HomeKit Over Bluetooth®*
- *UG235.03: Architecture of the Silicon Labs Connect Stack v2.x*
- *UG435.03: Architecture of the Silicon Labs Connect Stack v3.x*

4 Proven Cryptography

Product security shall use strong, proven, updatable cryptography using open, peer-reviewed methods and algorithms.

An important aspect of any IoT device is how secure the device is when it communicates with other devices, gateways, or the cloud. This standard mandates using proven cryptographic methods rather than attempting to implement your own.

Developers commonly secure communications such as TCP/IP connections, Bluetooth, Zigbee, or Z-Wave using the standardized and proven cryptographic methods native to the protocol. However, if a microcontroller sends sensitive information over a simple interface such as a UART to another microcontroller, it is important to realize that data should also be secured to prevent someone from snooping the UART line.

Silicon Labs offers a hardware CRYPTO module that provides an efficient acceleration of common cryptographic operations and allows these to be used efficiently with low CPU overhead. The CRYPTO module includes hardware accelerators for the Advanced Encryption Standard (AES), Secure Hash Algorithm SHA-1 and SHA-2 (SHA-224 and SHA-256), and modular multiplication used in ECC (Elliptic Curve Cryptography) and GCM (Galois Counter Mode). The CRYPTO module can autonomously execute and iterate a sequence of instructions to aid software and speed up complex cryptographic functions like ECC, GCM, and CCM (Counter with CBC-MAC).

In addition to the CRYPTO module, Silicon Labs includes mbed TLS as part of the Gecko Platform SDK. mbed TLS is open source software licensed by ARM Limited. It provides an SSL library that makes it easy to use cryptography and SSL/TLS in applications. mbed TLS supports software implementations of all crypto algorithms that are supported by TLS 1.2 as well as a build API that allows hardware drivers to replace the software implementations when cipher accelerators are supported by the platform. Its modular framework allows for subcomponents like the crypto libraries to be incorporated into a design independently of the SSL/TLS components, saving valuable code space and runtime RAM. mbed TLS supports SSLv3 up to TLSv1.2 communication by providing the following:

- TCP/IP communication functions: listen, connect, accept, read/write.
- SSL/TLS communication functions: init, handshake, read/write.
- X.509 functions: CRT, CRL and key handling
- Random number generation
- Hashing
- Encryption/decryption

These functions are split up into logical interfaces. They can be used separately to provide any of the above functions or to mix-and-match into an SSL server/client solution that utilizes a X.509 PKI. Examples of such implementations are provided with the source code. Components or plugins and APIs provide configuration interfaces accessible through the various SDK installations.

For more information, see the latest MCU and Peripheral Software Documentation for the target part at <https://docs.silabs.com>.

5 Security by Default

Product security shall be appropriately enabled by default by the manufacturer.

The state in which a product is shipped is up to the manufacturer. This standard mandates that any security features provided with the product be enabled before shipping. Customers should not have to turn security on; rather they should actively have to disable it. For example, Silicon Labs Z-Wave end-nodes and gateway SDKs ship with S2 cryptography and SmartStart network formation enabled by default.

Silicon Labs believes that product security should be considered during product design, and not as an afterthought. Within development environments, all Silicon Labs application security features may be enabled or disabled as appropriate during application development. Security must also be considered during device design and testing. *AN961: Bringing Up Custom Devices for the EFR32MG and EFR32FG Families* describes the security tokens (keys, certificates, and so on) that can be programmed into a custom device to support various types of security, including that provided by the Gecko Bootloader (see section [6 Signed Software Updates](#)).

6 Signed Software Updates

The product shall only support signed software updates. While it is critical that all products be updatable, it is just as critical that these update images be secured. A manufacturer must cryptographically sign update images to prevent tampering during deployment. The product must not use unsigned updates, as they could be fraudulent.

Silicon Labs development tools support building signed upgrade images and securely updating devices in the field, through the Silicon Labs Gecko Bootloader. The Gecko Bootloader can be configured to perform a variety of functions, from device initialization to firmware upgrades. Key features of the bootloader are:

- Useable across Silicon Labs Gecko microcontroller and wireless microcontroller families
- In-field upgradeable
- Configurable
- Enhanced security features, including:
 - Secure Boot: When Secure Boot is enabled, the bootloader enforces cryptographic signature verification of the application image on every boot, using asymmetric cryptography. This ensures that the application was created and signed by a trusted party.
 - Signed upgrade image file: The Gecko Bootloader supports enforcing cryptographic signature verification of the upgrade image file. This allows the bootloader and application to verify that the application or bootloader upgrade comes from a trusted source before starting the upgrade process, ensuring that the image file was created and signed by a trusted party.
 - Encrypted upgrade image file: The image file can also be encrypted to prevent eavesdroppers from acquiring the plaintext firmware image.

On Series 1 devices, the Gecko Bootloader has a two-stage design, first stage and main stage, where a minimal first stage bootloader is used to upgrade the main bootloader. The first stage bootloader only contains functionality to read from and write to fixed addresses in internal flash. To perform a main bootloader upgrade, the running main bootloader verifies the integrity and authenticity of the bootloader upgrade image file. The running main bootloader then writes the upgrade image to a fixed location in internal flash and issues a reboot into the first stage bootloader. The first stage bootloader verifies the integrity of the main bootloader firmware upgrade image, by computing a CRC32 checksum before copying the upgrade image to the main bootloader location.

On Series 2 devices, the Gecko Bootloader consists only of the main stage bootloader. The main bootloader is upgradable through the hardware peripheral Secure Engine. The Secure Engine provides functionality to install an image to address 0x0 in internal flash, by copying from a configurable location in internal flash. To perform a main bootloader upgrade, the running main bootloader verifies the integrity and authenticity of the bootloader upgrade image file. The running main bootloader then writes the upgrade image to the upgrade location in flash and requests that the Secure Engine install it. The Secure Engine is also capable of verifying the authenticity of the main bootloader update image against a root of trust. The Secure Engine itself is upgradable using the same mechanism.

In summary, Series 2 devices support a hardware root of trust and a Secure Boot process that verifies the authenticity and integrity of Gecko Bootloader, whereas in Series 1 devices, the authenticity and integrity of Gecko Bootloader are assumed trusted and are not explicitly checked.

The Gecko Bootloader can enforce application image security on two levels:

- Secure Boot refers to the verification of the authenticity of the application image in main flash on every boot of the device. When Secure Boot is enabled, the cryptographic signature of the application image in flash is verified on every boot, before the application is allowed to run. Secure Boot is not enabled by default in the example configurations provided by Silicon Labs, but enabling it is highly recommended to ensure the validity and integrity of firmware images.
- Secure Firmware Upgrade refers to the verification of the authenticity of an upgrade image before performing a bootload, and optionally enforcing that upgrade images are encrypted. The Secure Firmware Upgrade process uses symmetric encryption to encrypt the upgrade image, and asymmetric cryptography to sign the upgrade image in order to ensure its integrity and authenticity.

For more information on Silicon Labs' support for software update security, refer to the following:

Bootloaders in general: *UG103.06: Bootloader Fundamentals*

The Gecko Bootloader in general: *UG489: Silicon Labs Gecko Bootloader User's Guide for GSDK 4.0 and Higher* or *UG266: Silicon Labs Gecko Bootloader User's Guide for GSDK 3.2 and Lower*

Using the Gecko Bootloader with specific protocols:

AN1084: Using the Gecko Bootloader with EmberZNet

AN1085: Using the Gecko Bootloader with Silicon Labs Connect

AN1086: Using the Gecko Bootloader with Silicon Labs Bluetooth Applications

Secure Boot on Series 2 devices: *AN1218: Series 2 Secure Boot with RTSL.*

7 Automatically Applied Updates

The manufacturer will act quickly to apply timely security updates. Whenever a security vulnerability is detected, the manufacturer will automatically apply a patch to the product. No user intervention will be required.

It is the manufacturer’s responsibility to develop and implement automatic security updates. The design and methodology of such systems, for example through a Cloud-connected infrastructure or by direct intervention by a service representative, is up to you.

Silicon Labs will notify you of any security-related updates, as described in section 8 [Vulnerability Reporting Program](#). Your responsibility is to evaluate the level of risk that vulnerability poses for your particular product and to integrate the update into your platform as appropriate so that your end users are protected. Updated components might include the protocol libraries, Secure Engine firmware inside the Series 2 family, or an SDK module such as the Gecko Bootloader that enforces secure OTA updates and secure boot functionality.

Silicon Labs recommends the following:

- Subscribe to security updates through our Salesforce portal. To review or change your subscriptions, log in to the portal, click **HOME** to go to the portal home page and then click the **Manage Notifications** tile. Make sure that **Software/Security Advisory Notices & Product Change Notices (PCNs)** is checked, and that you are subscribed at minimum for your platform and protocol. Click **[Save]** to save any changes.

- Do not turn off Simplicity Studio’s update notification. Within Simplicity Studio you can download updates and easily access product release notes.

8 Vulnerability Reporting Program

The manufacturer shall implement a vulnerability reporting program, which will be addressed in a timely manner. All companies that offer Internet-connected devices and services shall provide a public point of contact as part of a vulnerability disclosure policy in order that security researchers and others are able to report issues. Disclosed vulnerabilities should be acted on in a timely manner.

Manufacturers are responsible for implementing their own program. For any individual vulnerability, you will need to weigh the value of transparency with your customers against the risk of malicious use of the information to exploit a vulnerability before it can be addressed. Silicon Labs makes similar decisions about how broadly to report security vulnerabilities discovered in our products.

Silicon Labs customers and security researchers can report security vulnerabilities in Silicon Labs hardware and software products on the Silicon Labs website: <https://www.silabs.com/security/product-security>

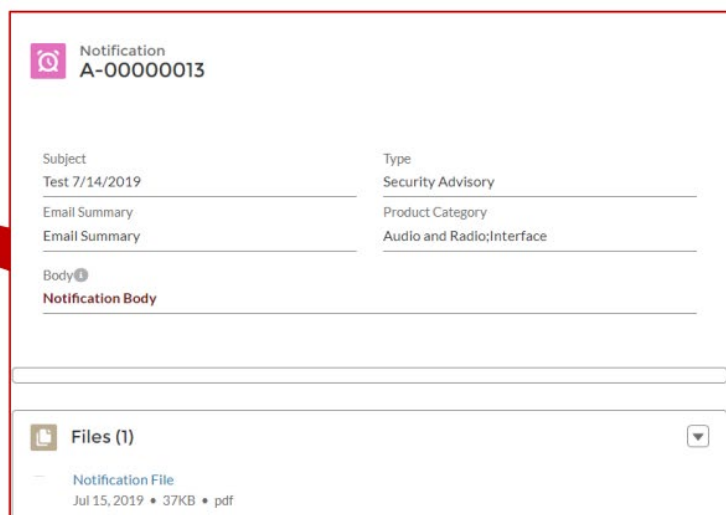
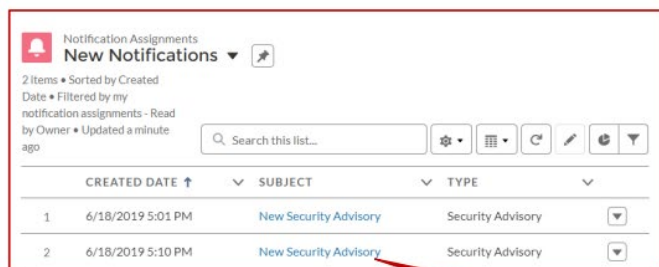
Silicon Labs' Security Vulnerability Disclosure Policy may be found here:

https://www.silabs.com/documents/public/miscellaneous/PS1012-Security_Vulnerability_Disclosure_Policy.pdf

Silicon Labs has a Product Security Incident Response Team (PSIRT) that is dedicated to the case management of reported security vulnerabilities. The PSIRT works with other Silicon Labs groups including Applications, Developers, Sales, and Marketing to assess reported vulnerabilities, perform technical analysis and determine an appropriate response. The key processes for addressing vulnerabilities include:

- Triage: Determines what is needed to reproduce the vulnerability.
- Technical Analysis and Disposition: Confirms the validity of the security vulnerability, its scope, and its impact, and provides a resolution or disposition decision. Silicon Labs scores incidents according to CVSS 3.1 (Common Vulnerability Scoring System): low, medium, high, critical.
- Output: Communicates with our customers. The level and method of disclosure beyond the reporting entity depends on the severity and scope of the vulnerability.

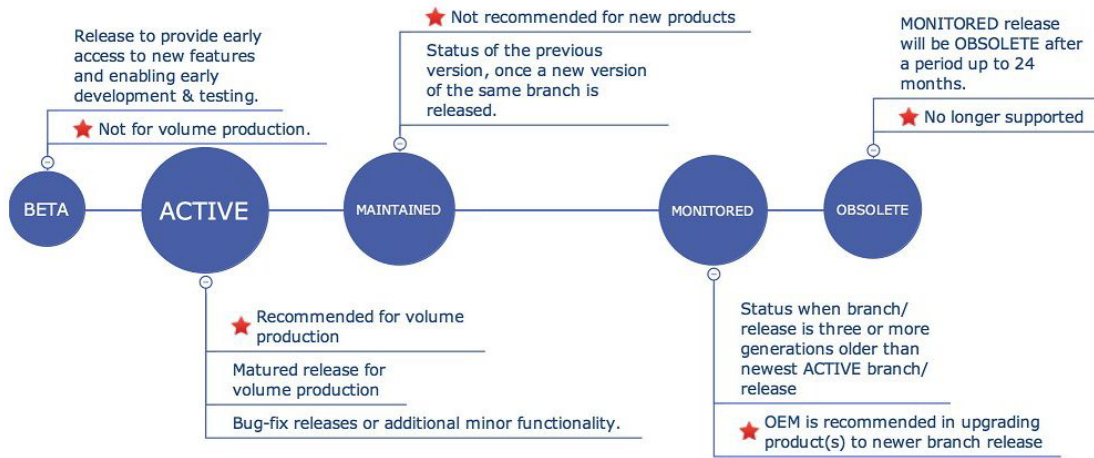
Silicon Labs' provides broad vulnerability reporting to customers subscribed through our Salesforce portal (see [section 7 Automatically Applied Updates](#) for information on how to subscribe). A subscribed customer will see Security Advisory notifications something like the following:



9 Security Expiration Date

The manufacturer shall be transparent about the period of time that security updates will be provided. Like a manufacturer's product warranty, there shall be transparency around the support period of security updates.

Manufacturers should provide details about product support at various stages and publish security expiration dates. Z-Wave's Protocol Lifecycle provides an example.



The Lifecycle details in what phases updates will be applied, and to what product branch. For details on the various phases and how the lifecycle is implemented for specific Z-Wave products, see:

<https://www.silabs.com/products/development-tools/software/z-wave/embedded-sdk/life-cycle>

10 Next Steps

The Silicon Labs Security web page (<https://www.silabs.com/security>) contains links to a variety of general security-related resources. You may wish to bookmark the page, as it will be continually updated with new content, new tools, and new flows.

If you are already in development, we strongly recommend that you implement the standards described here as you develop, test, and release your product to customers.

If you are in the early stages of your product design and have not already selected a device or development environment, we recommend that you include security considerations in your decision. Silicon Labs provides information about the security features of our devices and development environments. Section [11 EFR32 Series 2 Device Security Features](#) highlights the features and their documentation references. In addition, protocol-specific security information is available in the following documents.

- *AN1233: Zigbee Security* (contents were previously in this document under the title Application Security Fundamentals)
- *AN1302: Bluetooth® Low Energy Application Security Design Considerations in SDK v3.x and Higher*
- *AN1329: Using Silicon Labs Secure Vault Features with OpenThread*
- *UG103.14: Bluetooth® LE Fundamentals* and relevant Knowledge Base Articles (KBAs)
- *UG235.03: Architecture of the Silicon Labs Connect Stack v2.x*
- *UG435.03: Architecture of the Silicon Labs Connect Stack v3.x*

11 Series 2 Device Security Features

Protecting IoT devices against security threats is central to a quality product. Silicon Labs offers several security options to help developers build secure devices, secure application software, and secure paths of communication to manage those devices. Silicon Labs' security offerings were significantly enhanced by the introduction of the Series 2 products that included a Secure Engine. The Secure Engine is a tamper-resistant component used to securely store sensitive data and keys, and to execute cryptographic functions and secure services.

On Series 1 devices, the security features are implemented by the TRNG (if available) and CRYPTO peripherals.

On Series 2 devices, the security features are implemented by the Secure Engine and CRYPTOACC (if available). The Secure Engine may be hardware-based, or virtual (software-based). Throughout this document, the following abbreviations are used:

- HSE - Hardware Secure Engine
- VSE - Virtual Secure Engine
- SE - Secure Engine (either HSE or VSE)

Additional security features are provided by Secure Vault. Three levels of Secure Vault feature support are available, depending on the part and SE implementation, as reflected in the following table:

Security Level (1)	SE Support	MCU	Wireless SoC (2)
Secure Vault Base (SVB)	N/A	EFM32JG1, EFM32PG1, EFM32JG12, EFM32PG12, EFM32GG11, EFM32GG12, EFM32TG11	EFR32xG1, EFR32xG12, EFR32xG13, EFR32xG14
Secure Vault Mid (SVM)	VSE (VSE-SVM)	EFM32PG22C	EFR32xG22C, EFR32xG27C (3)
Secure Vault Mid (SVM)	HSE (HSE-SVM)	-	EFR32xG21A, EFR32xG21A (Rev C), EFR32MR21A (Rev C), EFR32xG23A, EFR32xG24A, EFR32xG25A, EFR32xG28A
Secure Vault High (SVH)	HSE only (HSE-SVH)	EFM32PG23B, EFM32PG28B	EFR32xG21B, EFR32xG12B (Rev C) EFR32xG23B, EFR32xG24B, EFR32xG25B, EFR32xG28B

Note:

1. The features of different Secure Vault levels can be found in <https://www.silabs.com/security>.
2. The x is a letter B, F, M, or Z.
3. Unlike other VSE-SVM parts, the EFR32xG27C device has a built-in PUF.

Secure Vault Mid consists of two core security functions:

- Secure Boot: Process where the initial boot phase is executed from an immutable memory (such as ROM) and where code is authenticated before being authorized for execution.
- Secure Debug access control: The ability to lock access to the debug ports for operational security, and to securely unlock them when access is required by an authorized entity.

Secure Vault High offers additional security options:

- Secure Key Storage: Protects cryptographic keys by “wrapping” or encrypting the keys using a root key known only to the HSE-SVH.
- Anti-Tamper protection: A configurable module to protect the device against tamper attacks.
- Device authentication: Functionality that uses a secure device identity certificate along with digital signatures to verify the source or target of device communications.

A Secure Engine Manager and other tools allow users to configure and control their devices both in-house during testing and manufacturing, and after the device is in the field.

Silicon Labs strongly recommends installing the latest SE FW image on Series 2 devices and updating to the latest GSDK to mitigate security vulnerabilities. The latest SE FW image can be found in this Windows folder for GSDK v4.x:

```
C:\Users\>\SimplicityStudio\SDKs\gecko_sdk\util\se_release\public
```

Refer to AN1222: *Production Programming of Series 2 Devices* for guidance on the SE firmware upgrade procedure. The latest SE firmware shipped with Series 2 devices and modules (if available) at the time of this writing are listed in the following table:

Series 2 MCU and Wireless SoC VSE - SVM	Shipped SE Firmware Version
EFM32PG22C	1.2.12
EFR32xG22C	1.2.12
EFR32xG22C (Rev D)	1.2.14
EFR32xG27C	2.2.1

Series 2 Wireless SoC HSE - SVM	Shipped SE Firmware Version
EFR32xG21A	1.2.13
EFR32MR21A (Rev C)	1.2.16
EFR32xG21A (Rev C)	1.2.16
EFR32xG23A	2.1.7
EFR32xG24A	2.1.7
EFR32xG25A	2.2.1
EFR32xG28A	2.2.2

Series 2 MCU and Wireless SoC HSE - SVH	Shipped SE Firmware Version
EFR32xG21B	1.2.13
EFR32xG21B (Rev C)	1.2.16
EFM32PG23B	2.1.7
EFR32xG23B	2.1.7
EFR32xG24B	2.1.7
EFR32xG25B	2.2.1
EFM32PG28B	2.2.2
EFR32xG28B	2.2.2

In support of these products Silicon Labs offers whitepapers, webinars, and documentation. The following table summarizes the key security documents:

Document	Summary	Applicability
AN1190: Series 2 Secure Debug	How to lock and unlock Series 2 debug access, including background information about the Secure Engine	Series 2
AN1218: Series 2 Secure Boot with RTSL	Describes the secure boot process on Series 2 devices using Secure Engine. For information on bootloading with Silicon Labs products, see <i>UG266/UG489</i>	Series 2
AN1247: Anti-Tamper Protection Configuration and Use	How to program, provision, and configure the anti-tamper module	Series 2 with SVH
AN1268: Authenticating Silicon Labs Devices using Device Certificates	How to authenticate a device using secure device certificates and signatures, at any time during the life of the product	Series 2 with SVH
AN1271: Secure Key Storage	How to securely “wrap” keys so they can be stored in non-volatile storage	Series 2 with SVH

Document	Summary	Applicability
AN1222: Production Programming of Series 2 Devices	How to program, provision, and configure security information using Secure Engine during device production	Series 2
AN1303: Programming Series 2 Devices Using the Debug Challenge Interface (DCI) and Serial Wire Debug (SWD)	How to provision and configure Series 2 devices through the DCI and how to program their internal flash memory through the SWD	Series 2
AN1311: Integrating Crypto Functionality Using PSA Crypto Compared to Mbed TLS	How to integrate crypto functionality into applications using Silicon Labs implementation of PSA Crypto compared to Mbed TLS	Series 1 and Series 2
AN1374: Series 2 TrustZone	Describes the basics of TrustZone, secure and privileged programming model, and shows how to upgrade existing application to TrustZone.	Series 2

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



IoT Portfolio
www.silabs.com/IoT



SW/HW
www.silabs.com/simplicity



Quality
www.silabs.com/quality



Support & Community
www.silabs.com/community

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project

Trademark Information

Silicon Laboratories Inc.[®], Silicon Laboratories[®], Silicon Labs[®], SiLabs[®] and the Silicon Labs logo[®], Bluegiga[®], Bluegiga Logo[®], EFM[®], EFM32[®], EFR, Ember[®], Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals[®], WiSeConnect, n-Link, ThreadArch[®], EZLink[®], EZRadio[®], EZRadioPRO[®], Gecko[®], Gecko OS, Gecko OS Studio, Precision32[®], Simplicity Studio[®], Telegesis, the Telegesis Logo[®], USBXpress[®], Zentri, the Zentri logo and Zentri DMS, Z-Wave[®], and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com