Application Report
SLAA715–August 2016

System-Level Tamper Protection Using MSP MCUs

Bhargavi Nisarga, Eric Peeters — *TI Microcontrollers – MSP MCUs*

ABSTRACT

Security in embedded systems is a topic that is gaining prominence as embedded systems and products are being deployed everywhere to be used in our everyday routines. Security concerns for embedded system developers and its users scales from adversaries having remote to physical access of the system. Increasing security for remote access includes incorporating secure data communication and secure software and firmware updates to the system; for example, leveraging industry-accepted cryptographic algorithms and secure communication protocols. This application report focuses on security concerns with adversaries having physical access to the system or product, understanding the need for system-level tamper protection, and how the security impact can be mitigated using system-level tamper detection and response functions.

Implementing system-level tamper detection involves identifying security assets in the system and defining a trust line boundary around it; and any attempt to invade the trust line (for example, an electric-meter box) is considered a tamper attempt that must be detected. The detection must be followed with appropriate responses or actions that are taken to improve the security of the assets. Antitamper mechanisms must be carefully implemented to not significantly impact the cost and the power of the overall system solution. This application report describes the features supported by ultra-low-power MSP microcontrollers (MCUs) to enable a possible implementation of system-level tamper functions to achieve the aforementioned purpose.

Contents

List of Figures

List of Tables

1 Introduction

Tamper refers to intentional alteration or manipulation to the system such that it compromises the secrets in the system or enables unauthorized operation of the system. Designing systems that are absolutely tamper proof is often not possible due to the increased cost involved to implement counter-measures against various known and potentially unknown attacks (for example, due to constantly improving technology that increases adversaries' capabilities to carry out an attack, either in terms of time taken or reduced cost to perform a successful attack).

1.1 Tamper Definitions

As defined in Reference [1], the following set of enabling techniques is used by a system-designer to implement some level of tamper protection to address or counter physical attacks to the system:

- **Tamper detection:** refers to monitoring tamper sensors or inputs in the system and identifying/qualifying any abnormalities to indicate a possible tamper event in the system. Examples include: an ambient light sensor or pressure sensor within the product enclosure monitored to check if product enclosure is open or closed, or a vibration sensor to detect any drilling actions in the system.
- **Tamper response:** refers to the action taken by the system upon indication of a tamper event. Tamper response typically takes actions such that the security assets in the system are not compromised (for example,, disable read out or manipulation of critical information in the system) and the system is not misused/modified. Examples include: making the product nonoperational, entering a safe mode by disabling critical operations within the product. It can also involve indication of tamper event to external systems; for example, displaying a message on the product screen asking user to take product to nearest authorized party for further evaluation.
- **Tamper evidence:** refers to marking or logging the occurrence of a tamper event in the system. Typically tamper evident systems require an irreversible change in the system that can be observable upon further investigation. Tamper evidence may include physical evidence at the product level (for example, broken seal), tamper log with details about the time and source of tamper, and other information providing details upon which further actions can be taken.
- **Tamper resistance:** refers to the ability of the system to detect and defend against a threat that has the objective to compromise the system or the data processed by the system. It is a measure of time, skill, tools and the knowledge of the threat (attacker) needed to perform a successful tamper attack.

Embedded systems can have varying levels of tamper protection requirements and tamper protection must be designed based on what needs to be achieved in the system. For example, some systems may only need to be tamper evident with no explicit need for tamper response or resistance.

2 Understanding the Need for Tamper Protection

2.1 Types of Adversary Access to the System

From a system access perspective, the hacker poses mainly three types of threats: *network* (remote), *board* (close proximity), and *chip* threats (see Figure 1).

The *network* threats comprise any communication that would allow the hacker to not be present at the location or in close vicinity of the device (for example, the hacker could be seating in a room on the other side of the planet).

The *board* threats comprise any PCB (printed card board) access and use of one of the chip wired interface. Typically the debug interface (for example, JTAG), the power supply (for example, power analysis attack or glitch attack), any of the serial interfaces (for example, UART allows read of the memory).

The *chip* threats comprise attacks that require depackaging of the device to get access to the internal layers and elements (for example, reverse engineering, memory read using electron microscopy, sea of gate analysis using emission microscopy, or focused ion beam (FIB) to modify or bypass security sensors). This last threat type is typically harder and more expensive to counteract but also costs more for the hacker to carry out (for example, special equipment and training is necessary).

Physical attacks refer to board and chip threats. This document focuses on board threats.

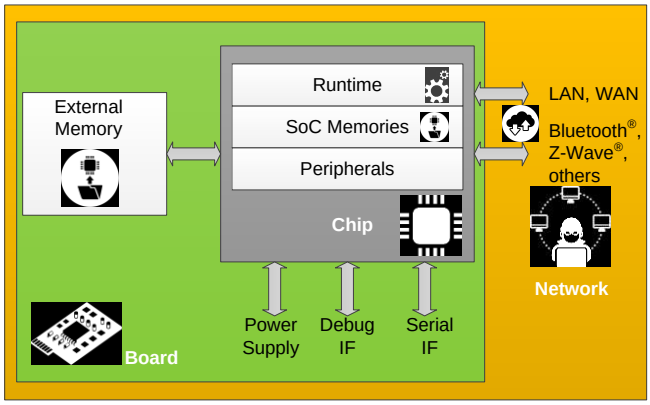

Figure 1. Threat Types in an Embedded System

2.2 Security Concerns With Physical Attacks (Board and Chip Threats)

Tamper protection refers to measures implemented in a system to mitigate security risks due to physical tamper attempts. The objectives of physical attacks at system- or chip-level include:

- Compromising confidentiality of the system (for example, access code IP and data, or keys).
- Compromising integrity of the system (for example, modify code, data, or keys stored in the device or in external memories to control the system).
- Compromising availability of the system (for example, disrupt normal operation by making system unavailable). This is also known as a denial-of-service attack.
- Exploiting weakness in the system such that it can be used for remote hacks of units deployed in field (that is, link between physical and remote attacks).

The various attacks with physical access to the product can be broadly classified as:

- **Noninvasive attacks:** Attacks that do not include physical intrusion or damage to the product enclosure or device package at system- or chip-level, respectively. Typical noninvasive attacks include:
  - **Side-channel attacks:** Attacks that are based on observing the system behavior while it performs cryptographic or secure operations, (for example, execution time, power consumption, or behavior in the presence of faults) to retrieve keys or passwords used in the system. Typical side-channel analysis based attacks include:
    - Timing analysis attacks
    - EMA (electromagnetic analysis)
    - Power analysis, either simple power analysis (SPA) or differential power analysis (DPA)
  - **Fault Injection attacks:** Attacks that alter environmental and operating conditions that cause the device or system to malfunction in a way that compromises the security (for example, skip a critical CPU instruction or erase bits to defeat device debug lock or other device programming security features). Common methods include altering:
    - Voltage (for example, introduce glitches to the power supply or increase or decrease supply voltage beyond operating limits.
    - Temperature (for example, heat or cool device beyond operating limits)
    - Clock or timing (for example, external crystal attacks, send a clock pulse that is too short)
  - **Software attacks:** Attacks launched through any communication interface with the device. Commonly attacked device communication interfaces are:
    - Device debug interface (for example, locking JTAG/SBW access to the device with embedded memories holding security assets)

- Other device programming interfaces (for example, bootloader, vendor proprietary interface, or protocol)
- Any data communication interface (for example, external memory interface, serial interfaces (I^2C, UART, SPI), (EMIF, Quad SPI) wireless interface)

- **Invasive attacks:** Attacks that involve physical intrusion at a system or chip level.
  - At the system level, this would mean physical intrusion into the product enclosure or the tamper-protected enclosure (trusted boundary) in the system (for example, the PCB tamper mesh).
  - At the chip level, this would mean physical intrusion into the device package. Chip-level physical intrusion is not in the scope of this document and is, therefore, not discussed in further detail here.

2.3 Need for Securing Physical Access of a System

Tamper protection at the system level is most commonly implemented in payment card industry (PCI) products (for example, credit card readers) and, in these applications, the tamper protection needs to conform to specific PCI standards. However, the tamper protection concept should not be limited to just these applications and must be considered in other application spaces that are at a potential risk of being attacked at physical access level.

With many applications becoming internet-connected, the need to secure remote connections is becoming critical; however, system designers should also consider that any information gathered at a physical access level can enable remote hacks that were otherwise more difficult to carry out. Therefore, it is very important to secure physical accesses to the product as well. Not all security measures may apply to a system and must be considered based on the likelihood and the impacts of successful security exploitation in terms of reputational, financial, operational, safety, or health damage.

2.4 Difference Between Chip-Level and System-Level Tamper Protection

With MCUs having embedded memories, the secrets in the system are typically on-chip (for example, code, data, and keys). However, designing MCUs with chip-level tamper protection is expensive (for example, a shield on the MCU die may involve additional masks in device fabrication process), and this feature is normally supported on MCUs in the secure-MCU type of product. To alleviate the chip-level tamper cost, but still implement sufficient security against physical threat accesses, system-level tamper protection should be considered. This provides tamper protection at a system level, which can potentially include components other than the MCU that are protected by the system-level tamper protection measures. Other components include sensor controllers, backup batteries, external crystals, and external memories. System-level tamper protection can also offer flexibility in setting tamper conditions and thresholds, depending on the environment or surroundings where the product is deployed.

3 Designing System-Level Tamper Protection for Example Embedded System

This section describes a typical MCU-based embedded system and the various steps involved in designing example tamper protection in the system.

CAUTION

This is an example flow that describes identifying security assets, threats, attacks of concern, and corresponding measures for the typical embedded system considered. This document does not cover all security assets, threats, or exposure points applicable to the example system, nor does it describe all necessary security measures that must be considered to secure the system. It is the responsibility of the system designer to properly analyze their system and to consider all of the measures that are needed to sufficiently secure their system.

3.1 Example Embedded System

Figure 2 shows a typical microcontroller-based embedded system.

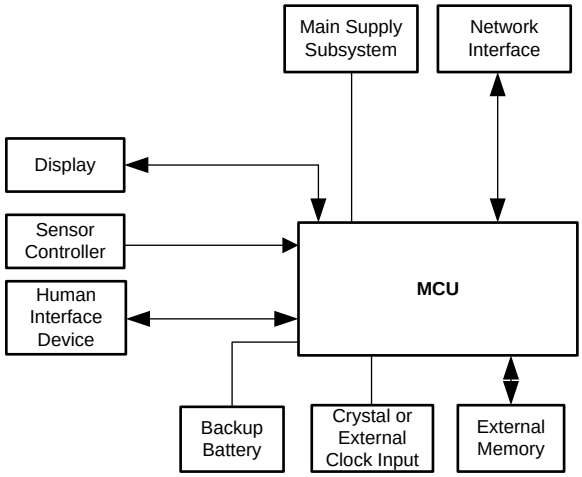

Figure 2. Example Embedded System

In this example embedded system, the MCU interfaces to the sensors and sensor controllers, processes the sensor information, and transmits the processed data through external data communication interfaces (for example, a network interface as shown). There can be external human interface devices (for example, keypad, mouse, or touch buttons) and indicators (for example, displays, LEDs, audio/buzzers) in the system that the MCU interfaces to. Many systems use on-chip clocking options, but some systems need precise and accurate clocking and, therefore, use external oscillators [for example, a crystal oscillator (XTAL)] or external clock inputs.

Backup batteries are common in systems that need the MCU or parts of the MCU to function when the main supply to the system is not available (for example, for real-time clock operation). Many MCU-based embedded systems use on-chip memories to store keys, code, and data. However, some systems have external memories to store additional information (for example, some configuration information, additional keys, code, or data used by the system such as display data, or a previous or golden firmware image in case of firmware or software updates).

As an embedded system designer wanting to improve the security of this system, the first step is to identify the security assets and threats in the system.

3.2 Identifying Security Assets and Potential Threats

To define tamper protection measures, it's important to identify the security assets in the system and the potential threats with attackers having physical access to the system. This in turn helps define a *tamper-protected enclosure* (or trusted boundary) in the system and corresponding measures needed to mitigate security risks.

Figure 3 shows some of the example assets (in blue) in the embedded system considered.

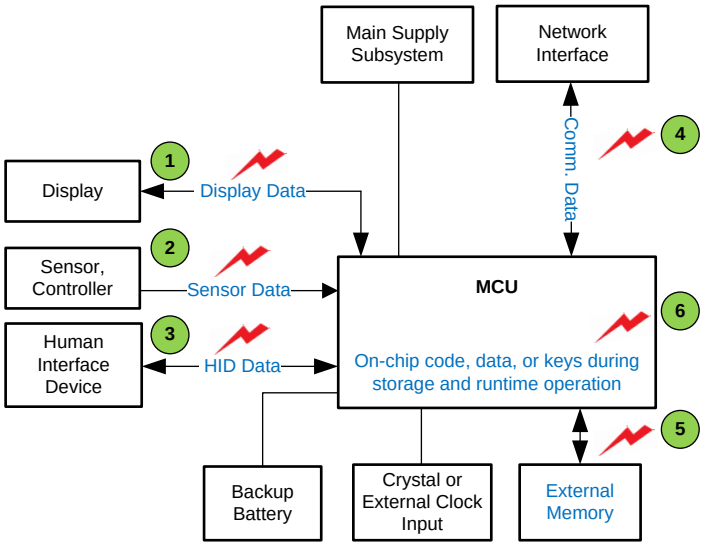

Figure 3. Typical Assets in Example Embedded System

Table 1 lists example threats to the respective assets identified in Figure 3.

Table 1. Example Threats to the Assets Defined in the Embedded System

Asset ID	Example Assets	Potential Threats
1	Display data	The system must not be compromised such that it displays wrong or sensitive information on the display interfaces (for example, electronic shelf displays made to display wrong price or sensitive information).
2	Sensor or actuator data	The sensor or actuator data arriving to the MCU must not be tampered with such that wrong information is propagated to the MCU and eventually the network. Conversely, the commands going from the MCU to the actuator must not be tampered with.
3	HID data	The inputs from the human interface devices must not be tampered with such that the system is made to operate in an unintended way (for example, invalid inputs or invalid combination of inputs to the MCU).
4	Communication data	The network interface communication must be kept confidential and not prone to eavesdropping or spoofing or man-in-the-middle attacks.
5	External memory contents	The external memory contents must be kept confidential and must not be tampered with such that unintended code or data is accessed or used by the MCU in the system
6	On-chip code, data, or keys	The MCU must keep the on-chip assets (for example, code or data stored and processed, sensitive keys, or device runtime operation) secure such that confidential information is not leaked and any vulnerabilities in the system is not used to compromise the MCU operation or information communicated to the network.

3.3 Attacks of Concern

At a physical access level, the potential threats to the embedded system can be achieved through physical attacks. Table 2 maps example attacks to the assets and threats identified in Table 1.

NOTE: Invasive chip-level attacks are not considered in this document.

Table 2. Example Attacks of Concern in the Embedded System

Asset ID	Example Assets	Example Attacks of concern
1	Display data	• Eavesdropping on data sent to or received from MCU • Data modification or spoofing of data sent to or received from MCU (for example, man-in-the-middle attacks) • Replay attacks (for example, replay valid inputs from HID)
2	Sensor or actuator data	
3	HID data	
4	Communication data	
5	External memory contents	
6	On-chip code, data, or keys	• Fault injection attacks to alter the device environmental or operational conditions with the intent to operate the MCU or other components in an unintended way such that it compromises device security (for example, open device debug lock, or to modify on-chip code or data contents to leak further information or operate in an unintentional way). Fault injection attacks include: – Overtemperature and undertemperature conditions – Overvoltage and undervoltage conditions – Overclocking, underclocking, and glitches or short clock pulses applied to the device • Noninvasive side-channel attacks (for example, timing attacks, EMA, or DPA) to retrieve keys or passwords used on-chip.

3.4 Tamper Protection Measures

Defining a system boundary or box that identifies the limit between an assumed trusted world and the nontrusted world is typically the first level of defense used in designing tamper protection. This helps partition the system to maintain components with key security assets within the tamper-protected enclosure and appropriately define tamper protection around this boundary.

Figure 4 shows an example tamper-protected enclosure defined to the embedded system in discussion. The tamper-protected enclosure is typically a product enclosure or PCB mesh enclosing all or some of the components in the system.

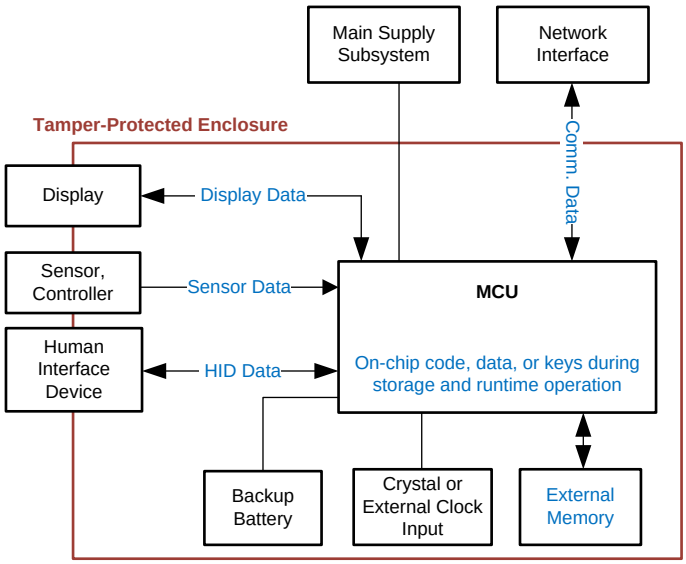

Figure 4. Tamper-Protected Enclosure Applied to Example Embedded System

End-product requirements may also drive the tamper-protected enclosure definition in the system. In this example, the main supply subsystem (for example, with need to replace or recharge batteries in the end product) and the network interface component (for example, with the need to support multiple network interface options as plug-ins at the end-product level) are external to the tamper-protected enclosure.

The components within the tamper-protected enclosure are protected by this first line of defense and any additional security measure implemented (for example, securing display, sensor, HID, and external memory data communication with the device with cryptographic and protocol level measures) to counter the threats listed in Table 2 further boost the security of the components within the boundary.

Tamper protection flow in the system includes tamper monitoring or detection followed by tamper evidence or response actions. The security measures to implement tamper protection are numerous and, although it is not intended to provide all of them here, Figure 5 shows some of the typical detection and monitoring mechanisms as well as the main evidence and response actions.

NOTE: The example options listed in Figure 5 for tamper monitoring and tamper evidence and response actions is not the complete set needed to secure an embedded system. It is the responsibility of the system designer to properly access and to consider any and all measures needed to sufficiently secure their system.

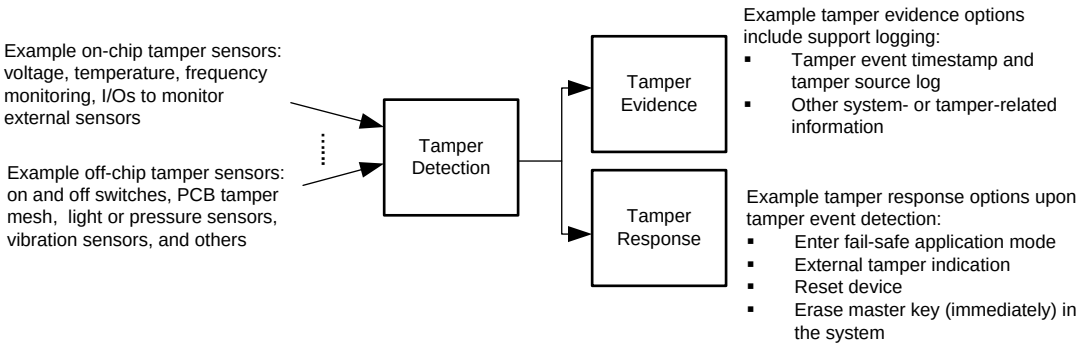

Figure 5. Tamper Protection Flow

Table 3 provides further details about these tamper protection measures applicable to the example embedded system. The measures correlate to monitoring of the tamper-protected enclosure including the environmental and operating conditions of the system. Table 3 also describes the capabilities and features of MSP MCUs in efficiently implementing these measures.

NOTE: Although not part of tamper protection measures for this example system, the designer should consider secure data communication for interfacing to network PHY component that is external to the tamper-protected enclosure (see Figure 4). Cryptographic functions enable data confidentiality and authenticity to counter eavesdropping and spoofing attacks of communication data. Protocol level measures should be considered for replay attacks. Many MSP devices have an on-chip AES hardware accelerator that can be used for implementing crypto functions. For devices that do not have the required crypto hardware accelerators, TI offers software cryptographic libraries that are described in *C Implementation of Cryptographic Algorithms*. Side-channel counter measures should be considered for timing, EMA, DPA, and other attacks.

Table 3. Example Tamper Protection Measures

Tamper Protection	Hardware or Software Measure	Measure Implementation Details	Comments
Tamper-protected enclosure monitoring	On and off switches at the product enclosure level. Can also include external boundary monitors like light or pressure sensors and vibration sensors.	On and off switches are tied to tamper input pins of the MCU.	Ability to detect change in status of on and off switches in the lowest MCU power mode enables low-power monitoring of tamper-protected enclosure. Most MSP MCUs have interrupt-capable I/O pins that can be enabled in the lowest MCU power mode (for example, LPMx.5 shutdown mode) to achieve this.
	Active tamper mesh forming a PCB-based enclosure that is continuously sending and receiving random sequence of data along the PCB traces and monitoring for any interruptions or short circuit along the path	With no hardware active tamper mesh support, the implementation requires: (i) Generating random sequence of data (ii) Transmitting (outputting) the random data through a GPIO and along the PCB mesh trace (iii) Receiving the transmitted data on through a GPIO on the other end of the PCB mesh trace (iv) Comparing the output and input sequence with expected delay (introduced by the PCB mesh trace) to ensure mesh is secure.	PCB tamper mesh for active tamper monitoring is typically an expensive measure at system-level and more common in PCI (payment card industry applications). As of this writing, a TI design that supports active tamper mesh is being developed and is scheduled for release soon.
Overtemperature and undertemperature monitoring	On-chip or off-chip temperature sensor	Sample the temperature sensor using on-chip ADC every so often to check if the temperature of the system is within bounds. If using off-chip temperature sensor, then this sensor must be placed within the tamper-protected enclosure defined for the system.	Because temperature does not instantly fluctuate, the temperature samples can be spaced in time (for example, every 0.5 second) such that power needed to monitor this condition is optimized. Most MSP MCUs have on-chip temperature sensors and ADCs, and architecture enables configuring the ADC to sample every so often using an interval timer. Also, most MSP ADCs have window comparator feature that automatically determines whether or not the ADC measurement is within the predefined minimum and maximum bounds and accordingly interrupts the CPU, thus enabling further power optimization.
Overvoltage and undervoltage monitoring	On-chip or off-chip voltage monitors or supervisors	Monitor supply voltage condition such that when the supply voltage drops below or rises above the predefined voltage boundary conditions, the MCU has sufficient time to take appropriate actions to secure assets within the system (for example, erase keys, data logging, or notify external system of tamper event). Overvoltage and undervoltage conditions may include short pulses or glitches in voltage or continuously elevated or reduced voltage conditions. If using off-chip voltage monitors and supervisors, these components should be placed within the tamper-protected enclosure defined for the system.	The MSP430™ F5xx/6xx, FRAM, and MSP432™ MCUs have on-chip supply voltage supervisor circuitry that can be used for detecting overvoltage or undervoltage conditions. Upon subsequent power up, the reset reason can be checked and appropriate actions at application level can be taken. The on-chip supervisors are typically very low power and, if used, should be enabled all the time for tamper monitoring. External components are needed for overvoltage monitoring (for example, TPS3700) or suppression (for example, transients suppressors).

Table 3. Example Tamper Protection Measures (continued)

Tamper Protection	Hardware or Software Measure	Measure Implementation Details	Comments
Overclocking and underclocking monitoring	On-chip or off-chip clock monitors	If using an external crystal, it must be placed within the tamper-protected enclosure defined for the system. If an external digital clock is input to the system, then, the clock can be monitored against on-chip reference clocks to ensure it is within expected bounds.	Most MSP devices that support an external crystal interface have a crystal fault indication that can be used for monitoring the crystal faults. When the external clock interface is used in crystal-bypass mode, a software routine using on-chip clocks and interval timers can be used to monitor the external clock frequency to ensure its within bounds. This check can be enabled only when the external digital clock is being used in the system to conserve power.
Firmware integrity check	Software routine	Firmware routines using cryptographic algorithms to check integrity of all/parts of on-chip/off-chip firmware	The software cryptographic library for SHA can be used for implementing the integrity check routine on MSP MCUs (see Secure Hash Standard).
Secure device access	Device debug lockout, secure bootloader	Although the MCU device is placed within the tamper-protected enclosure, the device debug-lockout should be enabled as good practice. If bootloader operation is not needed, then bootloader should be disabled in the device. If bootloading when in-field is needed, then security measures to maintain firmware image confidentiality and authenticity during firmware updates should be considered.	All MSP devices support debug lock-out feature which should be enabled. MSP432 supports encrypted and authenticated bootloading. For more details, see *Configuring Security and Bootloader (BSL) on MSP432P4xx*. Crypto-bootloader is a custom bootloader solution for MSP430 FR58/9xx devices that supports encrypted and authenticated firmware updates on FRAM devices. For more details, see *Crypto-Bootloader - Secure In-Field Firmware Updates for Ultra-Low Power MCUs*.
Keys management	Secure storage and handling of keys	Using a master key to encrypt all other keys on-chip and off-chip keeps keys encrypted in the system when not used. The master key is now the main security asset and must be secured in the system. Upon tamper detection, some applications require erasing this master key instantly so that the rest of the keys in the system are not compromised.	The MSP430 FR58/9xx devices and MSP432 devices support IP encapsulation and IP protection features respectively that can be used to store the master key such its resilient to software vulnerabilities present external to the IP zones. Ability to instantly erase master key on FRAM devices is an additional advantage. Also, FRAM devices have flexible keys update capability (no need to erase a memory segment before updating the key value).

4 Certification

The main security certifications schemes are: Common Criteria (applicable to various markets and applications with a focus on smart-card and e-passport), Payment Card Industry (PCI-DSS and PTS, Finance industry) and FIPS 140-2 (applicable to various markets and applications). There are certainly more but these aforementioned three certification schemes are internationally recognized and used. These schemes all work in the same way: first a document providing the threats, assets, and security claim is prepared by the system designer, then the security of the measures is assessed by an independent testing lab (quantified by points or levels) and finally a report is issued (this process may take a few iterations between the designer and the reviewer at the testing lab). The certification requirements for each market are constantly evolving (as hacking techniques evolve), and hence there are regular revisions of the certification specifications.

The Payment Card Industry Data Security Standard (PCI DSS) is a proprietary information security standard for organizations that handle branded credit cards from the major card schemes including American Express, Discover, JCB International, MasterCard and Visa Inc. At the board and chip level, one the main requirement of PCI-DSS certification is to be able to detect and react to any tamper event by rendering the chip inoperable.

The Common Criteria for Information Technology Security Evaluation (abbreviated as Common Criteria or CC) is an international standard (ISO/IEC 15408) for computer security certification. Common Criteria provides assurance that the process of specification, implementation and evaluation of a computer security product has been conducted in a rigorous and standard and repeatable manner. The process starts usually from a Protection Profile (or PP) that defines the threats, assets and objectives of measures are.

The Federal Information Processing Standard (FIPS) Publication 140-2, (FIPS PUB 140-2), is a U.S. government computer security standard used to accredit cryptographic modules and protocols (hardware and software). Tampering of the system is considered starting from simple evidence (level 2) to robust detection (high probability) and resistance (level 3 and 4).

5 Summary

The increase in data and control between all types of devices, large and small, has driven security from a set of specific markets to a context that impacts all markets. Security of electronic systems has become paramount; so much so that it is now reported on in general news and has even become the subject of popular culture. Attacks requiring physical access to the system have proven to be very efficient and very cost effective for all sorts of hackers.

Whether it is for security reasons or for safety reasons, tamper evidence and resistance has become an effective way to limit the exposure to such attacks. This document provides guidance the help anyone using an MSP MCU to analyze their system and start thinking about the potential threats and measures that would be needed to make the application more secure.

6 References

1. *Configuring Security and Bootloader (BSL) on MSP432P4xx*
2. *Crypto-Bootloader – Secure in-field firmware updates for ultra-low power MCUs*
3. *C Implementation of Cryptographic Algorithms*
4. *Encyclopedia of Cryptography and Security, 2nd Edition*, Henk C.A. van Tilborg, Sushil Jajodia
5. *MSP430x5xx and MSP430x6xx Family User's Guide*
6. *MSP430FR58xx, MSP430FR59xx, MSP430FR68xx, and MSP430FR69xx Family User's Guide*
7. *MSP432P4xx Family Technical Reference Manual*

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265