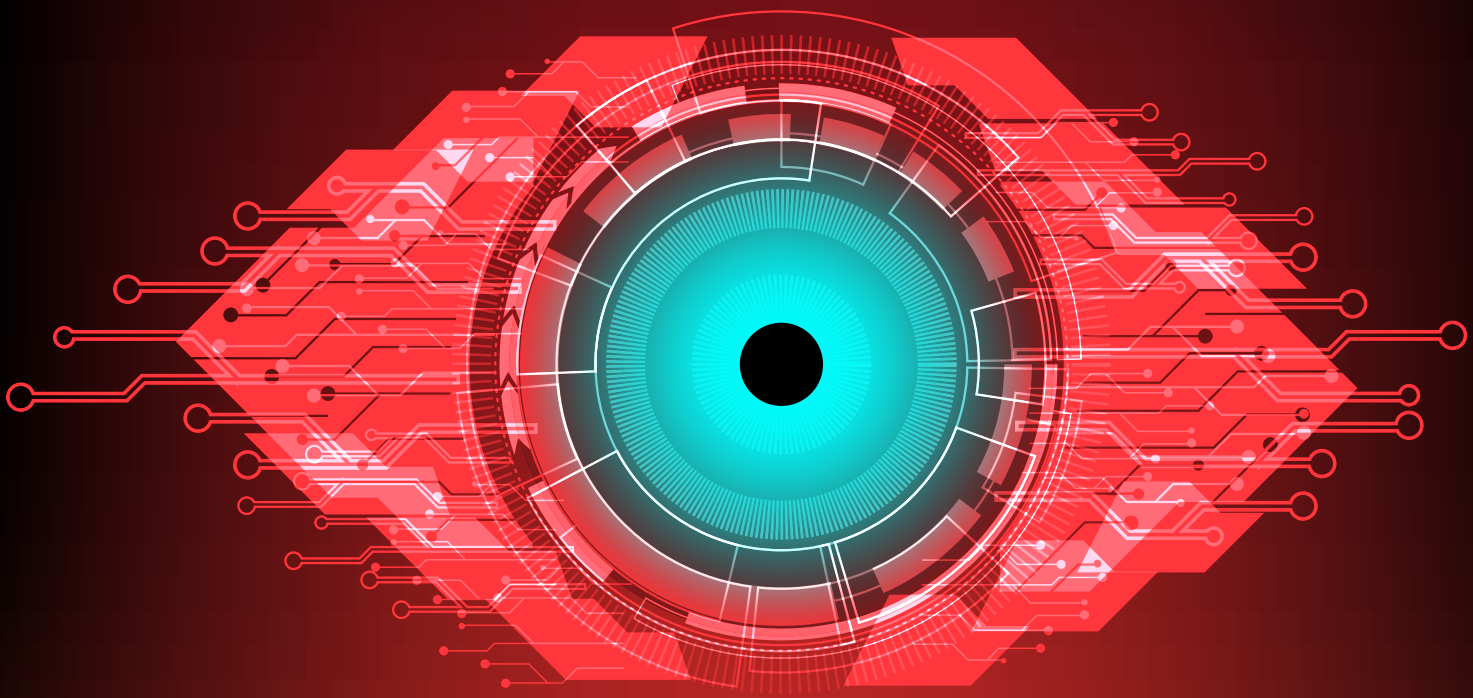
Intel Processor-Based Embedded Systems Cybersecurity



AUTHOR: **BORIS BAER**
R&D MANAGER

AITECH GROUP
JULY 2019

www.rugged.com

Contents

1. Introduction

In the emerging world of embedded systems, as autonomous platforms (UAVs, SUAVs, etc.) are being used for defense and aerospace, there is a growing interest in advanced means to secure these systems and related information from menacing intruders.

This article provides an overview of possible security threats for embedded systems utilized in defense/aerospace and also covers available techniques for securing these systems, and the information that they contain.

The defense and aerospace sector is subject to a variety of security threats through all phases of the lifecycle of an embedded system. This starts with cyberattacks to gain access to vendors' servers and steal design information, then continues with efforts to take control of the embedded systems or retrieve classified information from the aircraft, UAVs, SUAVs, etc. during their infield deployment, and in the case of a crashed UAV or loss of control, extract critical information from or reverse-engineer/clone different hardware and software elements of the system.

Aitech developed the AiSecure™ Cybersecurity Architecture to provide a comprehensive solution for cybersecurity threats for users of Aitech Secured Single Board Computers. This paper summarizes most cybersecurity threats and protection techniques with special attention to the solution introduced by Aitech, based on a combination of hardware and firmware tools.

2. Cybersecurity Protection Techniques

2.1. Authenticating Software

In order to ensure the security of embedded systems, care should be taken to authenticate all system firmware and software.

The BIOS, OS Loader and OS can be authenticated using a Trusted Platform Module (TPM) with Intel's Trusted Execution Technology (TXT). TXT uses the TPM to generate cryptographic hashes of the code prior to loading to verify that it has not been altered or corrupted.

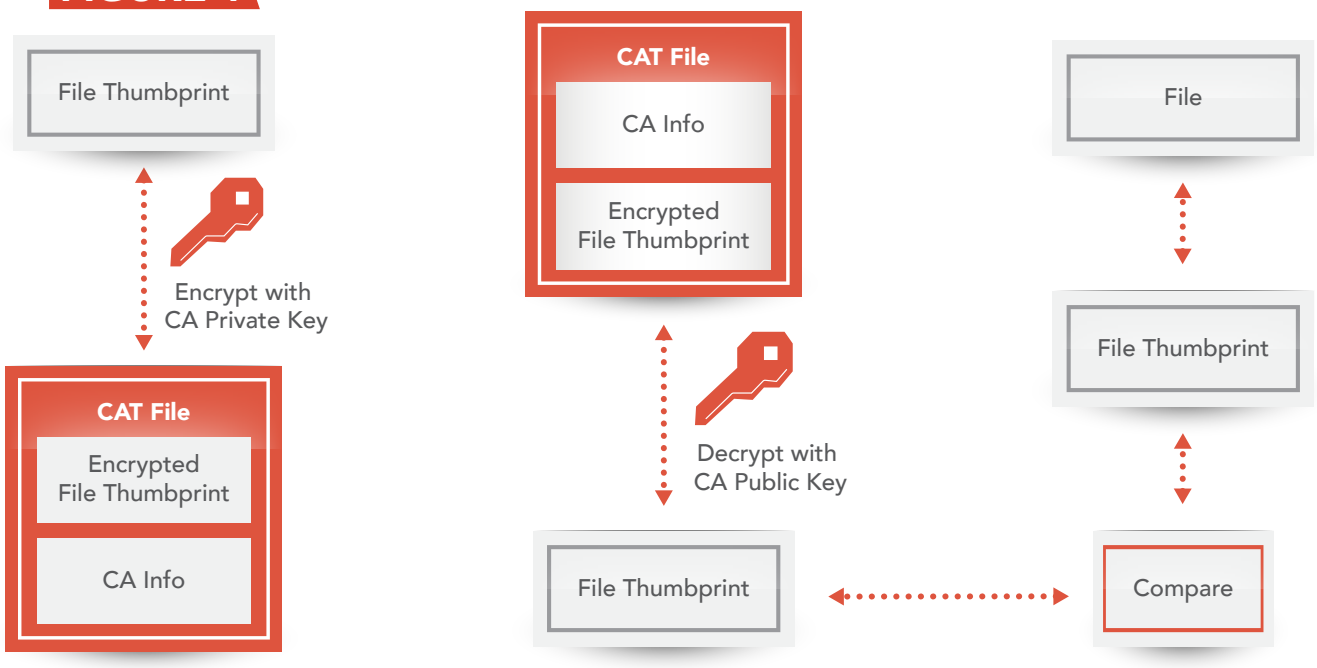
In addition to authenticating the BIOS, it is also important to verify that all additional software installed in the embedded system is authentic and has not been tampered with. Authentication of software can be performed using digital signatures, which can identify the software vendor and show that the files have not been modified after signing/publication.

We can look, for example, at the process through which a Windows OS device driver is verified using a digital signature. (See Figure 1)

The driver package is signed by a Certificate Authority (CA) that has been approved by Microsoft. The CA signs the driver by generating a thumbprint (cryptographic hash) and encrypting the thumbprint using the CA's private key. The encrypted thumbprint is then embedded into the driver's CAT file.

To verify the driver, the encrypted thumbprint is extracted from the CAT file before the driver is installed and it is decrypted using the CA's public key. The decrypted thumbprint is then compared with a locally-generated thumbprint of the driver file.

FIGURE 1



2.2. Protection of Network Interfaces

Ethernet interface is the most obvious source of cybersecurity threats. There are many ways to compromise embedded systems through network interfaces. Connectivity means potential targeting of opportunity for malicious actors. Every system module connected on a network can create a potential vulnerability that unauthorized users can exploit to obtain confidential sensitive data or worse, disrupt the safe operation of a system. Among cybersecurity threats that can find their way to endanger embedded computer systems, we can list viruses, Trojan horses, spyware, DOS and DDOS attacks, rootkits, SQL injection attacks and MITM (Man In The Middle) attacks (DNS spoofing, IP spoofing, ARP spoofing, SSL hijacking). It is important to protect the network interfaces of embedded systems from cybersecurity threats.

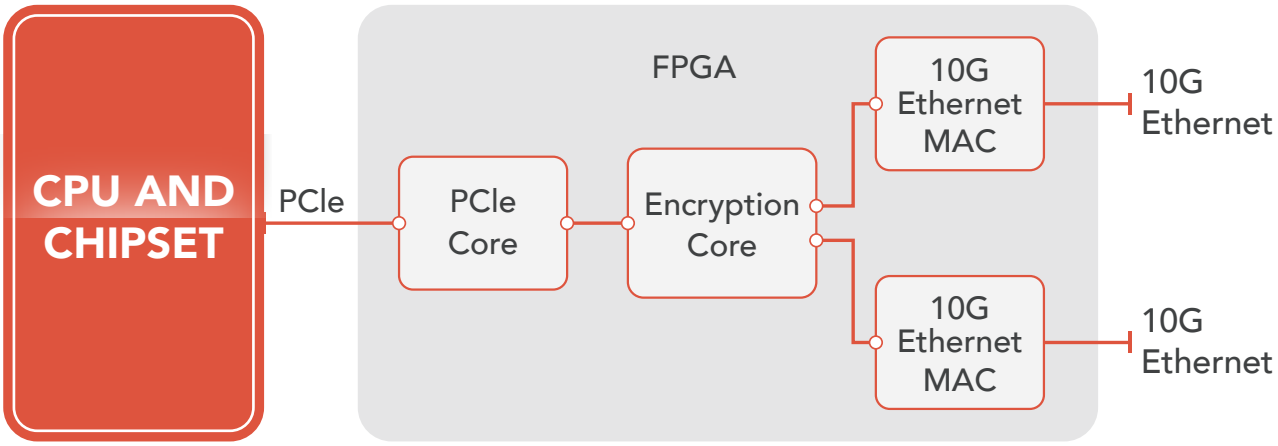
There are software-based security layers that provide protection, mainly based on authentication. Two key examples of these security layers are the security protocol suites MAC Security Standard (MACsec: IEEE 802.1AE) and Internet Protocol Security (IPsec). They can be built into the network software driver and stack to ensure that end-to-end communication cannot be disrupted, hacked or tapped into.

The MACsec standard strengthens network security by identifying unauthorized local area network (LAN) connections and excluding them from communication within the network. The protocol authenticates nodes through a secure exchange of randomly-generated keys, ensuring data can be transmitted and received only by MACsec-configured nodes, and provides optional point-to-point, Layer 2 encryption between devices on a virtual or physical LAN.

IPSec provides similar protection for a wide area network (WAN). It works on IP packets at Layer 3 (as opposed to Ethernet frames at Layer 2, like MACsec).

Most available network protection measures are based on software solutions. The drawback of this approach is that the data analysis and decisions are made after the data received from the network is copied to the system memory of the embedded computer. Protecting the network in hardware before malicious data penetrates into the processor and system memory is much more effective and safer. The network controller implemented in the FPGA (See Figure 2) provides a high level of user control over the security algorithm. Ethernet port monitoring, ingress Ethernet packet filtering, threat recognition, data encryption/decryption, building of white and black list of IP and MAC addresses, log attempts by unauthorized devices to communicate through the port are only part of a small list of cyber protection capabilities that can be implemented using a powerful, highly integrated FPGA, like the Xilinx Zynq UltraScale+ MPSoC.

FIGURE 2



2.3. Non-Authorized External Media Recognition

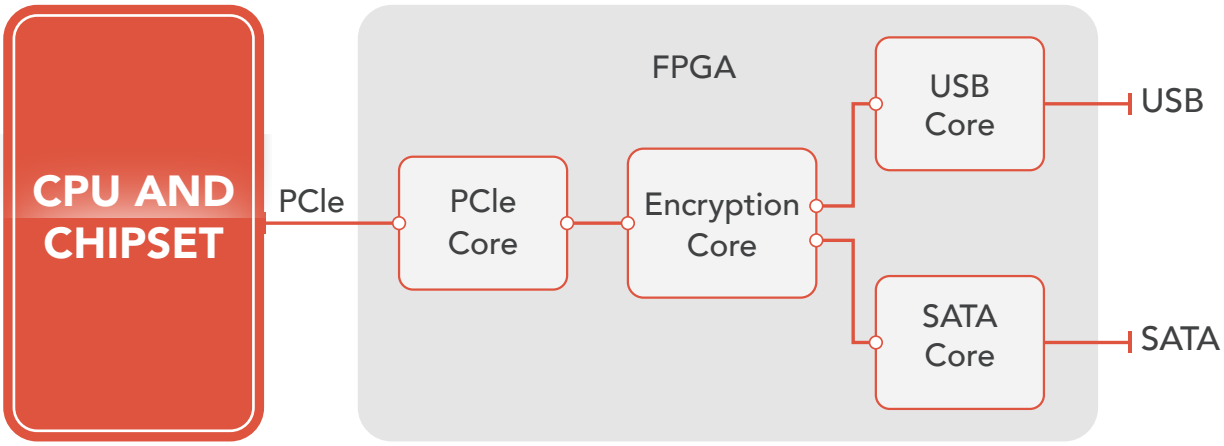
Although we normally think of malware as being transmitted via the internet, transmission via removable storage media is also possible. Allowing external storage media to be used also introduces the risk of critical information being taken off site.

For these reasons, it is often desirable to prevent the usage of external storage media on machines used for the development and control of critical systems.

The easiest methods use software. Windows OS, for example, has several ways to restrict usage of external storage devices. Group policies can be configured to prevent read and write access to various types of removable media. Group policies can also be configured to prevent usage of external media by blocking users from using removable storage devices at the driver level.

An alternate method relies on hardware rather than the OS. USB and SATA controllers are normally embedded within the CPU chipset and drivers are supplied by the OS vendor, leaving the end user with limited control. In order to prevent unauthorized media from introducing malware and leaving the site with sensitive data, a custom USB or SATA interface could be implemented using an FPGA (See Figure 3). The FPGA encrypts and decrypts all read and writes to the ports without relying on software. In case a non-authorized device is installed in one of the ports, the data on the device will pass through the decryption mechanism and will appear to contain only random data (any data on the device will be "decrypted" and appear to be garbage). Alternately, if critical data is copied to the device, the data will be encrypted and become unreadable on other machines. The port-monitoring function embedded in the interface cores will record unauthorized media attachments, issue system alarms or lock the system from further use.

FIGURE 3



2.4. Protecting FPGAs

With the ever-expanding usage of FPGAs in embedded systems, FPGAs often form the core of any system. This rise in both the usage and importance of FPGAs in a system makes protecting the IP contained in FPGAs as important as protecting the data processed by the FPGA.

There are a range of threats to FPGA design security, each threat with its own implications. Some are threats to the financial interests of a company, while others can threaten personal or even national security.

The most widely-used threats are:

- Reverse engineering reconstructs the FPGA design by looking at the layout, devices used, downloading the firmware, and analyzing the interaction between devices.
- Tampering occurs when an external agent attempts to gain unauthorized access to an electronic system and can try to extract or modify operating data or firmware in a system in an attempt to compromise it or shut it down.
- Offenders can attack an FPGA design by decoding its bitstream during loading from flash to RAM after applying power.

Other threats to be aware of include replacing all or part of an FPGA bitstream or microprocessor program with its own, either as a part of a reverse engineering program or as an attempt to compromise the system or the attached infrastructure; performing a side channel attack to use operational characteristics of the design – for example, timing or power – to retrieve keys, learn how to insert faults, or gain insight into the design; and issuing fault insertion to cause a circuit to malfunction in an attempt to force the circuit into a test or debug mode, an invalid state, or to output secret data by introducing glitches.

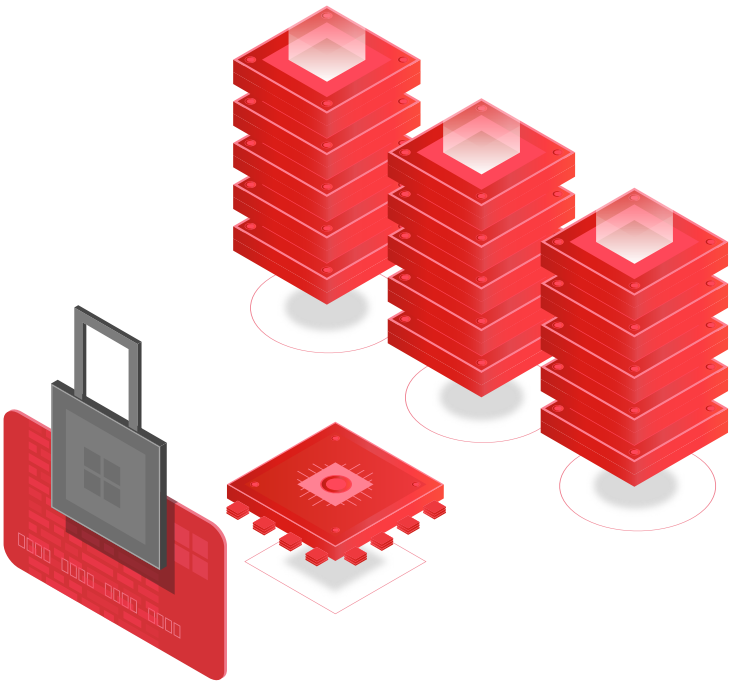
Xilinx Zynq UltraScale+ MPSoC FPGA offers a spectrum of security solutions to designers, including Device DNA, bitstream encryption, encryption key storage, bitstream authentication to prevent spoofing and Trojan horse attacks, Readback/JTAG Disable, Readback CRC to detect FPGA code changes, Hashed Message Authentication (HMAC) bitstream authentication, Keyclear and IPROG to delete encryption keys and erase configuration data.

2.5. Prevention of BIOS Tampering & Hardware Cloning

The system BIOS is responsible for embedded computer boot. After power up or reset, it initializes system resources, including memory, PCI system, Graphics, USB, SATA controllers, etc. The BIOS is also responsible for operating system loading. This core software operates below the operating system and therefore is not protected by OS security products. Offenders can plant malware and spyware into the BIOS, which will remain operating and undetected even when main storage devices are wiped. In addition, many commercial BIOS have remote management backdoors based on Intel vPRO technology. The AMT (Active Management Technology) allows remote access to the computer for management and security tasks when an OS is not operational.

Although most BIOS have protections to prevent unauthorized modifications, like secure boot, password protection, authentication, etc., the offenders are able to bypass these to reflash the BIOS and implant their malicious code.

The most obvious methods to protect BIOS include disabling vPRO backdoors and using the most updated BIOS where all known vulnerabilities were resolved. These measures provide enhanced protection, but don't prevent BIOS reflash. In order to provide comprehensive BIOS protection, Aitech designed specially tailored distributed firmware, combining hardware, software and programmable logic system components that prevent any unauthorized BIOS from running on our Secure Single Board Computers. This creates a tight connection between various system elements on the board. A special mechanism performs continuous verification of BIOS authenticity and integrity and checks full fit between all system elements. In addition to BIOS protection, this mechanism ensures the computer hardware doesn't go through unauthorized modification or cloning.



2.6. Protection of Storage

Write Protection

Write protection can be valuable in multiple ways, in both workstations and embedded systems. We can use write protection to protect software from being altered or corrupted and can also use it to prevent sensitive data from being written to non-volatile storage.

Write filtering can be implemented using software. For example, the Windows embedded OS offers the Enhanced Write Filter (EWF) tool, which places an overlay layer over the selected volume and writes changes only to the overlay. This makes the volume appear to be writeable to applications without actually altering the contents of the volume. EWF allows the user to commit the changes in the overlay to the volume or to simply discard the changes at reboot if a commit is not performed.

Data Encryption

Encryption is another tool that can be used to protect data in both workstations and embedded systems. Regardless of other security measures that may be taken, without encryption, someone with physical access to your system can simply remove a storage device and insert it into another system to gain full access to its contents.

Encrypting your data will provide protection. When the data has been encrypted, it will appear to be random garbage and will not be accessible without the encryption key. Encryption keys can be generated from passwords or can be stored on a USB drive or TPM (Trusted Platform Module).

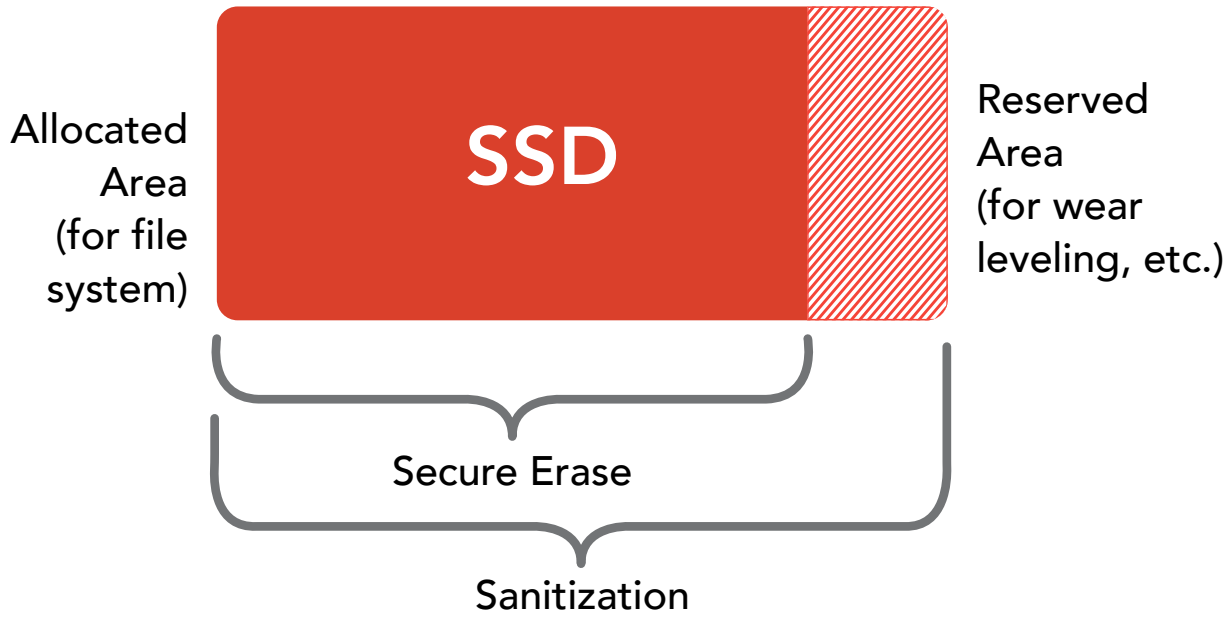
Encryption can be applied to specific files and directories on an individual basis or to a full disk as well as a BIOS device, so that the entire contents of the flash memory will be encrypted.

Full-disk encryption is performed transparently – data is automatically encrypted/decrypted during writes/reads to the disk. Files are accessible when the key is provided and the encrypted volume is mounted (typically as if it is a physical drive). After the volume is dismounted, the data will be inaccessible without remounting the volume with the correct encryption key.

Even if your data is encrypted, there are several attacks that may be used to attempt to bypass the encryption. These attacks include malware that is designed to capture the keys and passwords entered by the user, or reading the encryption keys from system memory.

Another attack relies on physical access to the machine rather than malware. This attack takes advantage of the fact that the contents of RAM can remain intact for a brief period after power is removed and attempts to read the keys from system memory by performing a cold boot (reboot without clearing of RAM) and quickly rebooting from a drive that will dump the contents of RAM to non-volatile memory, which can later be searched for the encryption key.

FIGURE 4



Secure Erase/Sanitization

With the wide-spread usage of high-capacity flash memory devices in both embedded systems and workstations, it is important to understand the available methods of erasing flash storage.

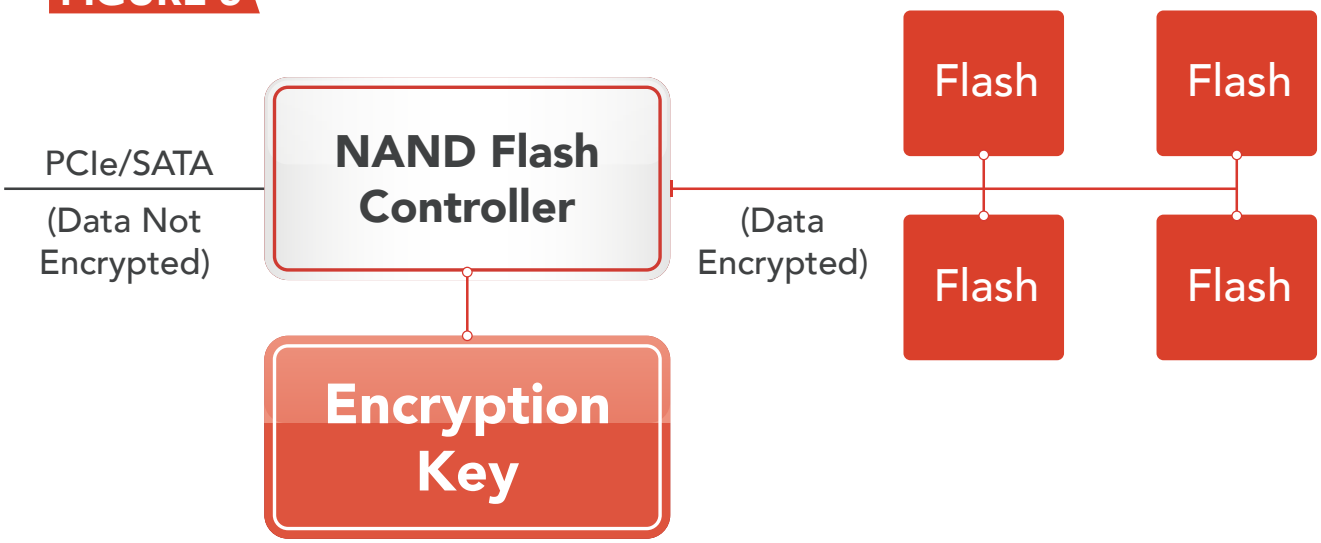
Secure erase, a feature supported by most SSDs, clears data from blocks that are allocated as volume sectors.

Sanitization is a more powerful tool, which clears data from all flash memory devices on the SSD (not only the blocks allocated as volume sectors, but also reserved areas used for functions such as wear leveling) (See Figure 4).

Usually, performing multiple write and erase operations directly on flash devices makes data recovery effectively impossible.

For SSDs with integrated/automatic encryption, deleting the encryption key is sufficient to securely erase the disk. The encrypted data would remain on the disk, but would be inaccessible without the encryption key (See Figure 5).

FIGURE 5



Physical destruction of flash storage can be accomplished by applying a high voltage directly to the flash devices for 3-5 seconds. This will physically destroy the devices and make them no longer usable. No data can be recovered.

Physical destruction of the flash storage may be used for an extra layer of security when discarding old hardware (after first erasing the disk as described above) and can also be used along with tamper detection circuitry to perform quick destruction of data in an embedded system that falls into the wrong hands.

For example, a tamper detection mechanism could connect a capacitor bank to the flash devices to destroy the devices within seconds of a tamper detection event.

2.7. System Tamper Protection

Tamper attack prevention techniques described in the previous sections make it more difficult to initiate an attack on the embedded system. In the event that an attack is launched, despite any employed prevention techniques, attack detection techniques detect the attack as soon as possible. The elapsed time interval between the launch of an attack and its detection (the detection latency) represents a period of vulnerability and needs to be kept as low as possible.

Once an attack is detected, the embedded system needs to take appropriate action. Attack recovery refers to techniques used to ensure that the attack is countered and that the system returns to secure operation. Attack recovery techniques could include locking up the system and rendering it useless for further operation, zeroing out sensitive data in memory.

Other recovery techniques include, for example, displaying a security warning and rebooting the system without full or partial cleanup of system data.

The design of attack recovery schemes involves trade-offs between the level of security and the inconvenience caused to users in the usage of the system after an attack.

3. AiSecure™ Cybersecurity Architecture

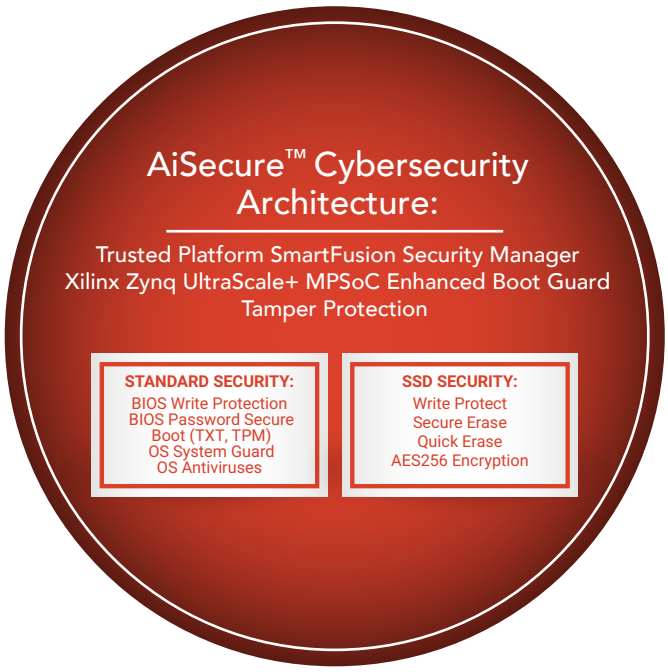
The AiSecure Cybersecurity Architecture is a set of hardware and firmware techniques designed to support a secured environment on Intel-based Single Board Computers (SBCs). AiSecure is based on capabilities and intellectual property developed by Aitech and deployed on our SBCs based on Intel processors, including the cybersecurity-focused C877 – a 3U VPX SBC with a 12-core Xeon D processor.

This unique cybersecurity architecture comprises trusted hardware and firmware and provides tools to maintain a highly secured and trusted system. This architecture supports a wide range of user-defined security policies, including system element integrity, access permissions and authentication of system resources and external interfaces.

Security mechanisms defining AiSecure protect embedded systems against all types of cybersecurity threats described in the previous sections of this white paper: theft of control, theft of intellectual property, theft of secrets and cloning.

AiSecure is based on a combination of standard security tools available on most Intel-based embedded systems (TXT, TPM, Secure Boot, BIOS password, BIOS write protection) with Solid State Drive (SSD) protection (SSD write protect, Secure Erase, Quick Erase) and unique hardware and firmware design that provides high-level security coverage and great users flexibility: Trusted Platform, Embedded Security Manager, Xilinx Zynq UltraScale+ MPSoC FPGA, Enhanced Boot Guard and Anti-tamper protection (See Figure 6).

FIGURE 6



AiSecure™ Cybersecurity Architecture Elements

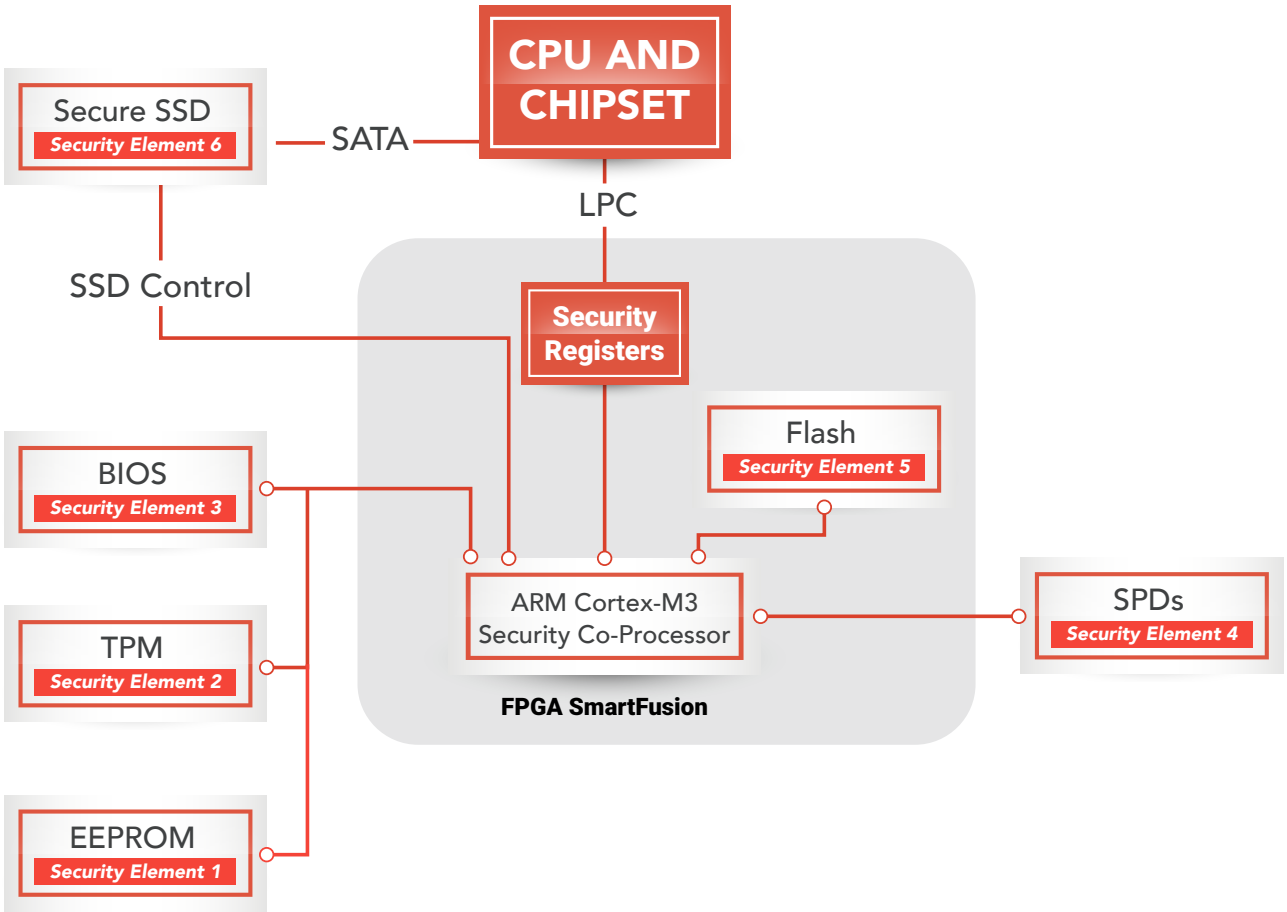
SmartFusion ARM as security controller

The ARM Cortex-M3 embedded in the main C877 SmartFusion FPGA has been utilized as an Intelligent Platform Management Controller (IPMC) and board security manager. The main C877 FPGA was designed to be in the center of the SBC's architecture and is capable of supporting most of the board's vulnerable non-volatile resources: BIOS, EEPROMs, FPGA configuration files, SSD. The security manager performs continuous monitoring of integrity of major system elements during all phases of C877 operation. It intercommunicates with the main board's CPU to maintain trusted platform and plays a key role in applying anti-tamper policies.

Trusted Platform & Boot Guard

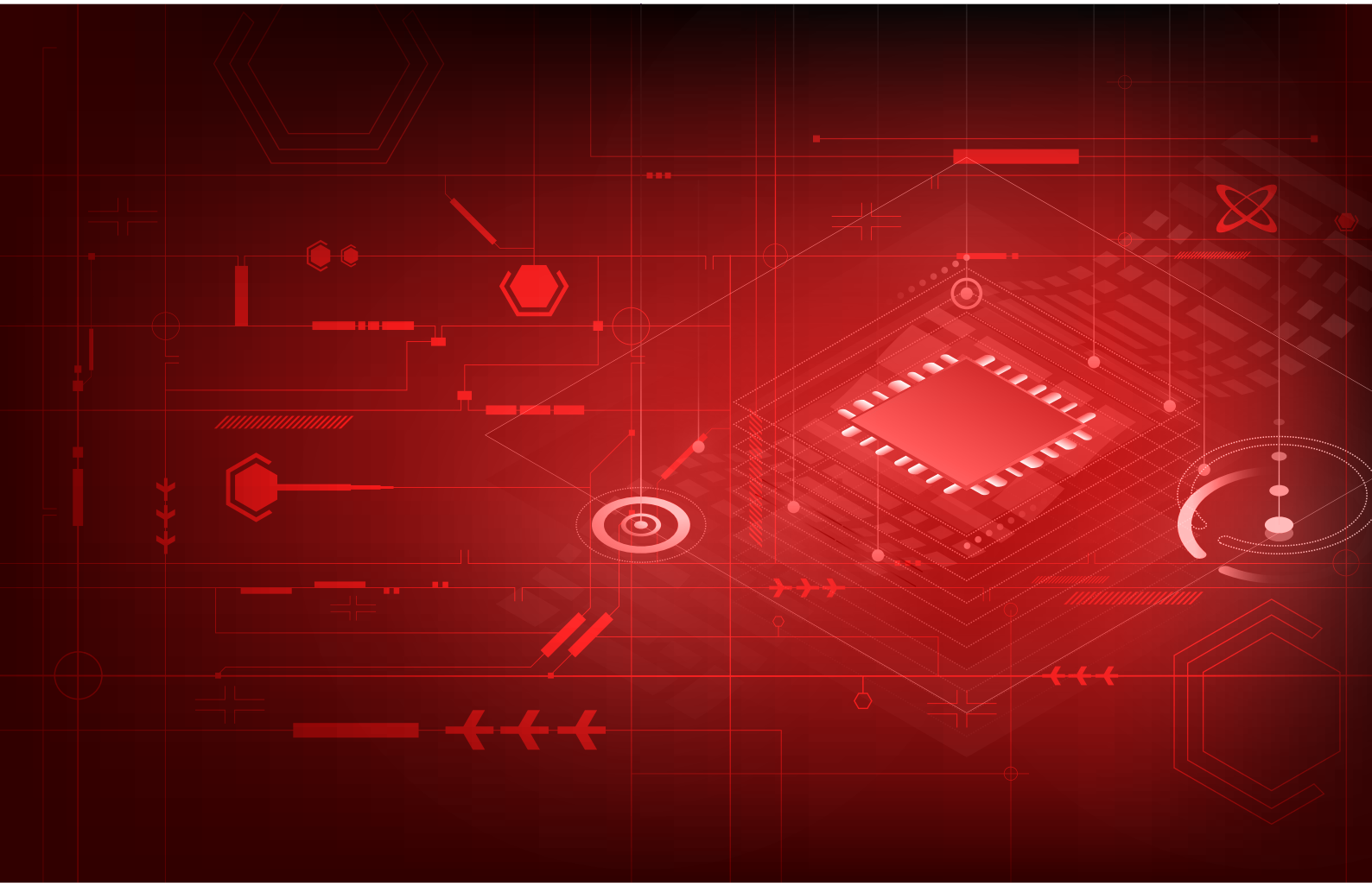
Aitech developed the Trusted Platform mechanism, combining Boot Guard functionality based on a combination of trusted hardware and firmware, providing a high level of embedded system confidence. It assures that there is a 100% fit between the SBC's software and hardware elements. Unlike other trusted platforms, Aitech's solution is based on a distributed architecture, with security keys and signatures embedded in various system elements: FPGA, FLASH, EEPROMs, SSD, etc. (See Figure 7).

FIGURE 7



3.1. Xilinx Zynq UltraScale+ MPSoC FPGA

One of the main C877 SBC security elements is the on-board Xilinx Zynq UltraScale+ MPSoC FPGA subsystem. A variety of important security building blocks were integrated by Xilinx into this sophisticated powerful device:



01 Boot image is both encrypted & authenticated for the highest level of security.

02 The programmable logic configuration bitstream (FPGA code) can be applied securely or non-securely.

03 The boot process is multi-stage and minimally includes the boot ROM and the first-stage boot loader (FSBL). Zynq UltraScale+ MPSoCs include a factory-programmed configuration security unit (CSU) ROM. The boot ROM determines whether the boot is secure or non-secure, performs some initialization of the system and clean-ups, reads the mode pins to determine the primary boot device, and executes the FSBL.

04 Anti-tamper to protect the FPGA IP from being copied, reverse-engineered, or modified.

The MPSoC is connected with the main SBC processor (Xeon) with eight lanes of high bandwith PCIe Gen. 3 bus. Independent from the main processor, the dual-core Arm Cortex-A53 operating at 1.5GHz is designed to perform secure OS boot from 64GB eMMC using fast 2GB DDR4 as system memory.

The native ARM interfaces consist of a dedicated RS-232 serial port and 1Gigabit Ethernet. In addition, the FPGA routes four general purpose SerDes interfaces capable of operating at 10 Gigahertz.

Aitech offers three assembly options for the C877 MPSoC FPGAs to satisfy customers' requirements for a combination between budget and performance:

01 ZU4CG with 192k System Logic Cells

02 ZU7CG with 504k System Logic Cells

This architecture is suitable for implementing various security techniques and allows implementing real secure IP cores utilizing high speed SerDes signals as a physical layer for either SATA III, USB3, 10Gbase-KR or other interfaces.

3.2. SSD Security

The C877's on-board SSD was specially designed by Aitech according to the most severe cybersecurity requirements. It supports secure erase, quick erase and AES256 Encryption.

Secure Erase

The C877 SSD supports a full set of secure erase protocols that can be triggered by either a software command or a hardware signal.

These algorithms include:

- NSA 9-12
- DOD 5220.22-M
- NSA 130-2
- Army AR 380-19
- Navy NAVSO P-5239-26
- Air Force AFSSI 5020 & AFSSI 8580
- IRIG 106-09

Once the secure erase process has initiated, it will continue to completion. If power is interrupted, the erase operation will pick up where it left off and complete when power is restored.

Quick Erase

The C877 SSD also supports quick erase, triggered by either a software command or a hardware signal. As for the secure erase, once the secure quick process has started, it will continue to completion. If power is interrupted, the erase operation will continue when power is restored.

SSD Encryption

The C877's on-board SSD supports a self-encrypted drive. All data stored in flash is encrypted with the AES256 algorithm.

The SSD supports two different encryption key management options:

- Every time 'power on' is applied on the C877 SBC, a password request prompt is sent to access the SSD. If the password is correct, the SSD will run well; if not, you will not be able to access the SSD.
- A data encryption key has been generated in the SSD controller hardware and is transparent to user. All data on the SSD will be encrypted, but the SSD itself will act as a regular mass storage device.

The data encryption key erase/renew can be triggered either by a software command or hardware signal. When this is applied on the SSD, which contains data, all data will be lost.

3.3. Anti-tamper

Aitech offers the C877 SBC, with a special battery-backed tamper detection mechanism recording unauthorized access to embedded system by opening the system case, for example. Even when the system power supply is switched off, the intruder's attempt to breach the system will be recorded in volatile memory.

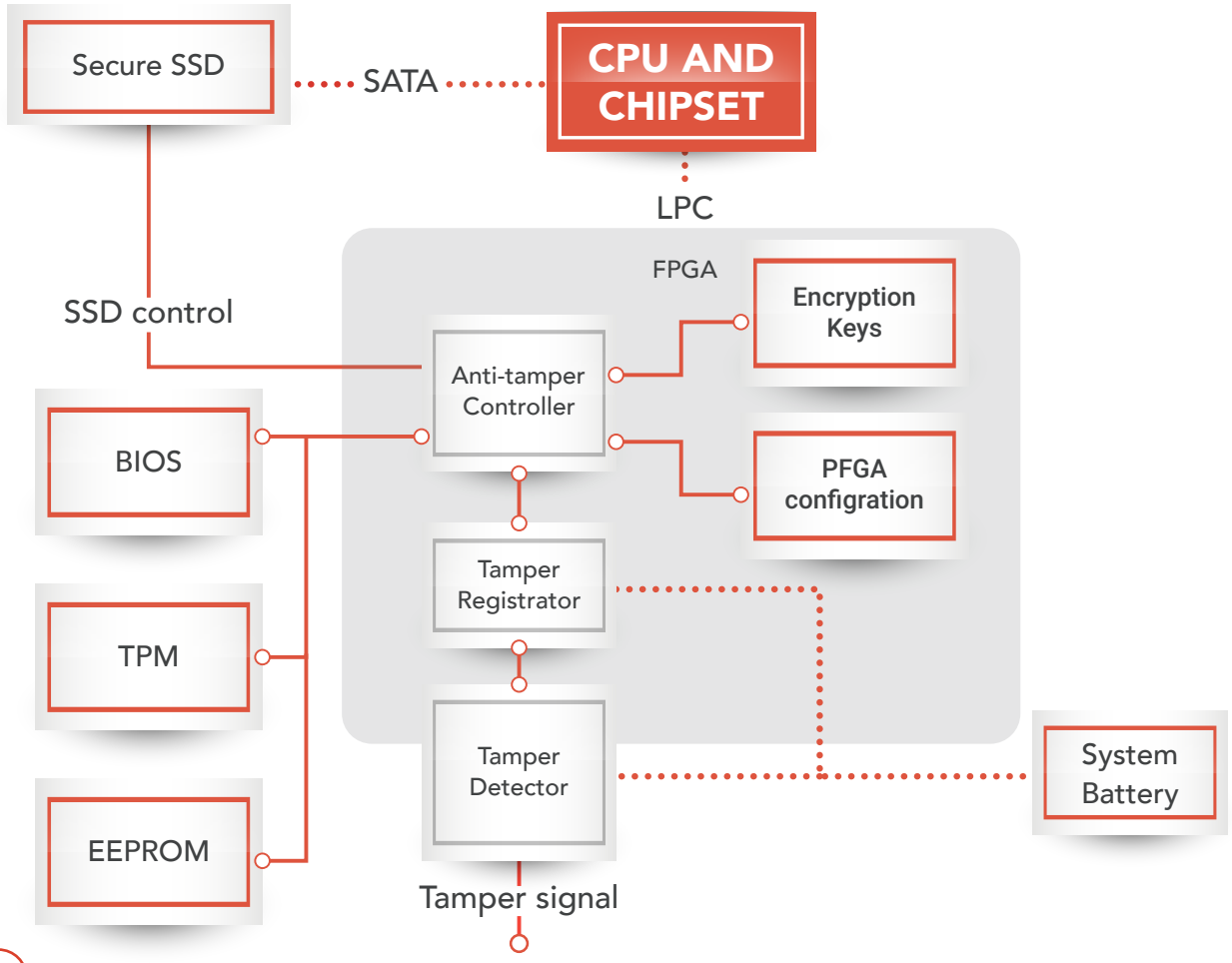
When, after a tamper attack, the embedded system is powered up, the Security Manager will initiate a complete system cleanup.

This includes deletion of all non-volatile memory devices:

- BIOS erase
- FPGAs configuration erase
- Encryption keys wipe from all storage elements: TPM, SSD, FPGA, EEPROM
- SSDs secure erase

Every time the embedded computer is powered on, it will focus on one mission only – wiping all information from its internal components. This scenario leaves the embedded system useless and difficult for full-operation recovery. (See Figure 8)

FIGURE 8



4. Conclusion

Today's embedded systems face multiple threats throughout all phases of their lifecycles, requiring designers and operators of the systems to build robust defenses that address these threats, with multiple-layers of protection, at all stages of the lifecycle – from development to deployment to end of life. In the case of autonomous systems, such as UAVs, protections should include contingencies for protection of data in systems that may fall into the wrong hands during deployment.

The security of systems that are used in critical industrial and military/aerospace applications can have real-world implications that can even impact public safety.

A comprehensive multi-layer protection strategy should be established from the early planning stages of a project and carefully implemented throughout the entire project lifecycle.

Aitech introduces AiSecure Cybersecurity Architecture on the C877 – a 3U VPX Xeon D SBC designed for cybersecurity. This platform integrates hardware and firmware design elements, providing great building blocks for implementing a cybersecurity protection suite for mil & aero applications. AiSecure offers a set of sophisticated tools that allow customers to implement various cyber protection techniques according to their needs and cybersecurity policies.

The following table summarizes cybersecurity threats of embedded systems and shows how the Aitech C877 deals with all these threats.

SYSTEM MODULE	CYBERSECURITY THREATS EXAMPLE	C877 CYBER COUNTERMEASURE
SOFTWARE	Unauthorized Program Installation, Viruses, Trojan Horses	Disk Write Protect, Antivirus, Secure Boot, Secure SATA, USB FPGA IP Cores
BIOS	Modification, Reprogramming	BIOS Password, Secure Boot, Trusted Platform, Boot Guard
FPGA	IP Theft, Reverse Engineering, Tampering	Trusted Platform, Device DNA, Bitstream Encryption, Bitstream Authentication, Readback/JTAG Disable, Readback CRC, Hashed Message Authentication (HMAC) Bitstream Authentication, Keyclear and IPROG
HARDWARE	Cloning, Tampering	Trusted Platform, Anti-tamper, Security Manager
NETWORK	DOS and DDOS Attacks, Rootkits, SQL Injection Attack, DNS Spoofing, IP Spoofing, ARP Spoofing, SSL Hijacking	Antivirus, Secure Network Stack, Secure Ethernet FPGA IP Core, vPRO Disable
DISK	Data/information Theft, IP Theft	Secure Erase, Fast Erase, Encryption, Anti-tamper

AUTHOR: **BORIS BAER**
R&D MANAGER

AITECH GROUP
JULY 2019
www.rugged.com