Creating Cyber-Resilient Embedded Systems and Securing the Supply Chain

A Lattice Semiconductor White Paper.

March 2021

Learn more:

www.latticesemi.com

Contact us online:

www.latticesemi.com/contact
www.latticesemi.com/buy

TABLE OF CONTENTS

Introduction

When the first true general-purpose electronic computers were created — starting with the *Electronic Numerical Integrator And Computer (ENIAC)*, which was constructed at the University of Pennsylvania between 1943 and 1946 — they worked in splendid isolation and no one even considered the concept of cybersecurity.

When the first large-scale, general-purpose computer network to connect different kinds of computers together — the Advanced Research Projects Agency Network (ARPAnet) — was turned on in October 1969, only four independent nodes were involved, and nobody thought to worry about cybersecurity. The only fear then was physical attacks to take down the network with bombs.

As we now know, those halcyon days were too good to last. The 1970s saw the emergence of malicious hackers, network breeches, and malware; the 1980s was the era of computer worms; and the 1990s experienced the rise of computer viruses. Today, with the internet in general and the Internet of Things (IoT) in particular, computers are everywhere, everything is connected, and everyone — from individuals with their smart doorbells, speakers, and thermostats to multinational conglomerations to governments — is concerned about cybersecurity, or the lack thereof.

Unfortunately, even today's state-of-the-art cybersecurity systems struggle to address a constantly changing threat landscape. And now, a new threat vector has emerged that is particularly concerning because it can compromise a system even before it's assembled by attacking system components as they move through the global supply chain. That threat is an attack against component firmware. Firmware vulnerabilities are an increasingly common weakness that organizations need to be aware of and develop a strategy to remedy. The National Vulnerability Database reported that between 2016 and 2019, the number of firmware vulnerabilities grew over 700 percent[1].

To understand how compromised firmware can impact a system, consider this: in a whitepaper jointly authored with the Open Compute Project (OCP), the Cloud Security Industry Summit said, "Firmware represents a significant threat vector for computer systems, appliances, and associated infrastructure. If the first code that executes on a device when it powers on were to become compromised, then the entire system can and should no longer be trusted as secure. Firmware can be compromised through malicious attacks or unintentionally."[2]

As discussed in this paper, the solution to securing firmware is to implement cyber resilient systems that can protect themselves against attack, detect when they are being attacked, and can automatically recover from attacks. A key aspect of this is the ability to secure the supply chain from cradle (product creation) to grave (product decommissioning) in such a way that bad actors are severely limited in their ability to corrupt the firmware effectively at any point in the chain.

Industry 4.1

A prime example of the interconnectedness of things is presented by Industry 4.0 (the Fourth Industrial Revolution), which is defined as the ongoing automation of traditional manufacturing and industrial practices using smart technology. Large-scale machine-to-machine communication (M2M) and the industrial IoT (IIoT) are integrated for increased automation, improved communication and self-monitoring, and the deployment of smart machines that can analyze and diagnose issues without the need for human intervention.

[1] Source: National Vulnerability Database (2016 and 2019)
[2] https://www.opencompute.org/documents/csis-firmware-security-best-practices-position-paper-version-1-0-pdf

A big enabler of Industry 4.0 is OPC UA. The OPC acronym stands for Open Platform Communications, which refers to an interoperability standard for the secure and reliable exchange of data in the industrial automation space and in other industries. The OPC Foundation is responsible for the development and maintenance of this standard. The original standard, which was restricted to the Windows operating system (OS) is now known as OPC Classic. Released in 2008, the OPC Unified Architecture (UA) is a secure, platform-independent architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework, thereby ensuring the seamless flow of information among devices from multiple vendors.

OPC UA defines its own standardized communications protocols that are based on authorization and authentication, signatures and certificates, and encryption. These protocols manage all of the communications between the clients that are receiving and aggregating information and the servers that are analyzing, processing, and acting on this information. When a factory or production line is first brought online, any communications are initiated by a trusted server. Prior to this, requests from clients will not be serviced. It's only after a client has been authenticated by the trusted server that bidirectional communications across the network are allowed.

Information technology (IT) refers to the use of computers and networks to store, retrieve, transmit, and manipulate data or information. Operational technology (OT) refers to the hardware and software that monitors and controls physical devices, such as motors, generators, valves, and pumps. Historically, OT networks utilized proprietary protocols that were optimized for their required functions. More recently IT-standard network protocols such as TCP/IP are being implemented in OT devices and systems to reduce complexity and increase compatibility with more traditional IT hardware.

The use of OPC UA enables the convergence of IT and OT, removing "data islands" by connecting business systems to operational systems, thereby making it easier to sense, monitor, and react to anything that's happening anywhere in the factory in real-time, speeding "time to insight" and maximizing efficiency.

The fact that OPC UA facilitates communication is important because a key feature of Industry 4.0 is that more and more things are being connected into the system — which spans simple sensor and actuator devices at the very edge of the internet to sophisticated servers forming the cloud — and all of these elements want to communicate.

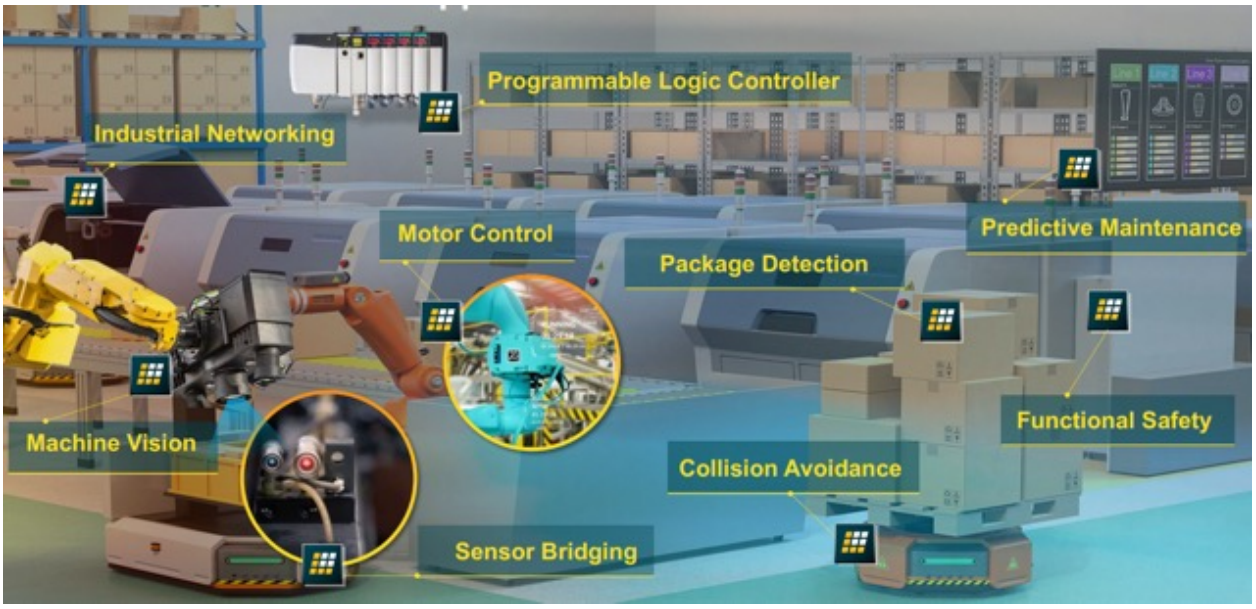

Figure 1. A few examples of industrial automation applications.

Consider predictive maintenance, for example. In the not-so-distant past, a dumb machine might have been equipped with nothing more than a temperature probe and a limit-based controller. If the machine's temperature exceeded a specified amount, its rudimentary controller would simply shut it down. Today, a smart controller may be equipped with an artificial intelligence (AI) application that employs a variety of sensors to monitor multiple aspects of the machine's condition and operation (e.g., temperature, power consumption, vibration) looking for trends and anomalies and communicating its conclusions (e.g. "This machine is exhibiting problems with its main bearing and will probably fail within 72 hours without maintenance") up the chain of command for further evaluation and action. But what happens if, once this controller has been installed, its firmware is attacked and compromised? Alternatively, how can the OT department guarantee that the device's firmware has not already been compromised when they bring it into the factory prior to it being connected into the network?

At the time of this writing, Industry 4.0 is already several years old. Unfortunately, like so many things, the Industry 4.0 concept is a double-edged sword. Prior to Industry 4.0 and OPC UA, nothing was connected, and attackers had to fight their way through multiple firewalls to get anywhere. And, even if they did eventually gain access to the heart of the factory, there was no way to reach certain devices that weren't connected to any network. One of the issues Industry 4.0 and OPC UA solves is that everything can be connected and disparate devices can talk to each other. Contrarily, one of the problems Industry 4.0 and OPC UA poses is that when everything is connected and disparate devices are talking to each other, it's possible for a single compromised device with malevolent intent to orchestrate the corruption and subjugation of the entire system.

Prior to the deployment of networks, the earliest form of computer security was perimeter security in which physical barriers were used to keep intruders out. The advent of networks resulted in the introduction of firewall security in which a network security device employs software and/or hardware to monitor incoming and outgoing network traffic and decide whether to allow or block specific traffic based on a defined set of rules. The current best practice is to employ defense-in-depth in which trusted sources and endpoints employ security at all levels, but even this is no longer sufficient to defeat some forms of attack.

Even if a system commences life in an uncompromised state, various parts of that system are going to require periodic firmware patches and firmware updates and power-cycling. This leads to the concepts of secure boot (ensuring that code launched by firmware is trusted) and secure update (ensuring that firmware patches and updates are trusted), but how can anyone guarantee that the lowest level firmware functions that control these actions have not themselves been compromised? Quis custodiet ipsos custodes? ("Who will guard the guards themselves?")

What is required now is Industry 4.1, where systems do not simply rely on cybersecurity, but also on cyber resiliency. As we know, cybersecurity does its best to keep invaders out, but can collapse if its defenses are breached. However, systems at every level — from Edge devices to cloud servers – that are also cyber resilient can protect themselves, and other less trusted devices, against attack, can detect when they are being attacked, and can automatically recover from attacks.

The Fall of Cybersecurity

The term cyberattack refers to an attempt to expose, alter, disable, destroy, steal, or acquire information through unauthorized access to a computer or network. Cyberattacks are usually aimed at accessing, modifying, or destroying sensitive information, extorting money from users, or interrupting normal business processes. They can range from installing spyware on a personal computer to attempting to destroy the infrastructure of a nation state.

The term cybersecurity refers to the technologies, processes, and practices that are employed to protect networks, devices, applications (programs), and data from cyberattack. In the case of the embedded world, there are various cybersecurity standards in play, such as ISO 27001, IEC 62443, and the National Institute of Standards and Technology (NIST) Cybersecurity Framework. The problem is that cybersecurity on its own is no longer sufficient in today's increasingly complex technological landscape.

A classic case of cybersecurity failure is exemplified by what happened to Danish shipping giant A.P. Moller-Maersk in 2017.

Operating out of 564 offices in 130 countries, Maersk is an integrated container logistics company that is responsible for 76 ports around the globe. Maersk also has responsibility for approximately 800 seafaring vessels, including container ships carrying millions of tons of cargo. In all, Maersk controls almost 20 percent of the world's shipping capacity.

In June 2017, Maersk was the victim of a massive cyberattack. In fact, Maersk wasn't the intended target, but was instead collateral damage of a worm called NotPetya that was unleashed by Russia to target Ukraine's banking systems. Unfortunately, a finance executive for Maersk's Ukraine operation asked his IT administrators to install an accounting package on a single computer. This infected package slipped past the company's cyber security defenses and — within a matter of hours — Maersk's worldwide operations were "dead in the water," as it were.

It took Maersk weeks to recover and cost the company at least $200M (some estimates place losses as high as $300M). The problem was that, although Maersk had a good cybersecurity posture, it hadn't made similar plans to ensure cyber resiliency.

It's important to note that the intention of this paper is not to denigrate cybersecurity, which forms the foundation for cyber resiliency. But it has to be understood that cybersecurity is not sufficient in and of itself.

The old paradigm was to think, "We won't get attacked because we have perfect security." The problem is that there is no such thing as perfect security. Anyone and everyone can be attacked, so the best approach to cybersecurity is to adopt an attitude of "when we get attacked," rather than "if we get attacked." Thus, the emerging mindset is changing from "Of course we can prevent an attack" to "How do we manage our way through an attack when it happens?" or "How do we become more resilient to an attack?" The answer is to create systems that are cyber resilient from the ground (firmware level) up.

The Rise of Cyber Resiliency

The term cyber resiliency refers to the ability to continuously deliver an intended outcome despite adverse cyber events such as cyberattacks. Cyber resiliency embraces information security, business continuity, and overall organizational resilience.

Operating systems and software applications have been practicing a high-level version of cyber resiliency for some time using techniques like segregated memory models and hypervisors (software, firmware, and/or hardware that creates and runs virtual machines). The idea underlying hypervisors is that if one of the virtual machines is corrupted, crashes, or is terminated, the others can continue to run in their own "containers."

This is all well and good, but current high-level cyber resiliency strategies are largely ineffective at the firmware level. For this reason, perhaps not surprisingly, it's at the firmware level that hackers are now predominantly focusing their attentions because there's no hope of addressing this type of attack at the OS or application level. If a company is attacked at the firmware level — in the BIOS or the boot code or the fundamentals of how the various components power-up — and if that company is not cyber resilient, then the system is going down and the company will no longer be able to function. In many cases, the resulting financial and reputational damage to the company can be devastating.

On the other hand, if the company's firmware can protect against attacks, detect attacks as they happen, and recover from those attacks and keep the system functional, this this would be classed as a cyber-resilient system.

Cyber resiliency guidelines and standards already exist or are poised to be made public. For example, NIST published its SP 800-193 Platform Resiliency Guidelines in 2018; the Trusted Computing Group (TCG) has a Cyber Resiliency (CyRes) Working Group that will publish a standard sometime in 2021; and SAE International has a Cyber Physical Security Working Group that covers elements of cyber resiliency and that will be publishing a standard in 2021.

Cyber resiliency has already been adopted at the server level and is now proliferating throughout embedded systems in commercial, industrial, medical, and military environments — anywhere where there is the potential for problems, which basically means everywhere. Furthermore, businesses that are adopting cyber resiliency practices are communicating this fact to their investors and shareholders, spreading the word that they are actively engaged in protecting their organizations, customers, and products.

Implementing Cyber Resiliency

Various cyber resiliency offerings are emerging in the market. Lattice Semiconductor's state-of-the-art cyber resiliency capabilities are founded on Lattice MachXO3D™ FPGAs, which offer a highly desirable combination of low power, substantial programmable logic resources, and large numbers of input/outputs (I/Os). The flash-based configuration of MachXO3D FPGAs provides "instant-on" capabilities that allow them to be the platform's first-on, last-off devices and to dominate the market for system control and power management functionality.

MachXO3D devices also boast hardware security features that bring NIST-level security to embedded systems. In fact, the MachXO3D is the only FPGA at <10K LUTs that is equipped with a NIST-certified Immutable Security Engine.

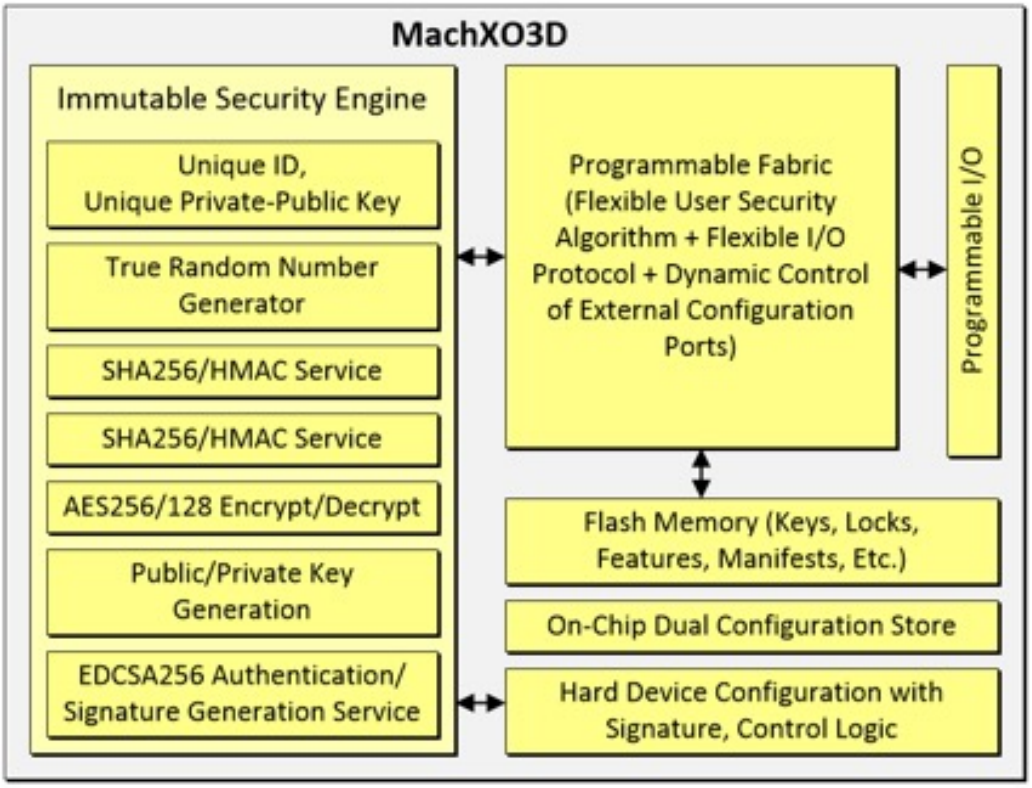

Figure 2. The MachXO3D secure control FPGA is the system's first-on, last-off, HRoT programmable logic device.

In addition to enabling hardware root-of-trust (HRoT) in the form of the system's first-on, last-off device, the MachXO3D's Immutable Security Engine also enables pre-verified cryptographic functions such as ECDSA, ECIES, AES, SHA, HMAC, TRNG, Unique Secure ID, and public/private key generation.

The Immutable Security Engine — along with Lattice's recently introduced software, hardware and services offering, the Lattice Sentry™ solution stack — supports security throughout the product lifecycle, including device manufacturing and transport, platform manufacturing, installation, operation, and even decommissioning. It also enables comprehensive protection against a variety of threats by providing data security, equipment security, data authentication, design security, and brand protection.

As defined by NIST SP 800 193, platform firmware resiliency (PFR) involves protection, detection, and recovery. Protection includes protecting the platform's firmware and critical data from corruption and ensuring the authenticity and integrity of any firmware updates. Detection includes cryptographically detecting corrupted platform firmware and critical data, both when the system is first powered on, while the system is running, and following any in-system updates. Recovery includes initiating a trusted recovery process and restoring any corrupted platform firmware and critical data to its previous value.

MachXO3D devices fully address cyber resiliency requirements by providing features such as a secure dual-boot capability. The combination of the MachXO3D's programmable logic, Immutable Security Engine, and secure dual-boot configuration block provides flexibility during design implementation and enables secure updates after the system has been deployed. In addition to providing HRoT by design, the use of on-chip logic dramatically minimizes the cyberattack surface area.

With today's distributed architectures, unsecure chip-to-chip, board-to-board, and system-to-system communications can be snooped or emulated by hackers. In systems with multiple firmware blocks, only one of these blocks needs to be hacked in order to take control of the entire system, and hackers will invariably attack the weakest link.

As was previously noted, the MachXO3D FPGA is the first device to turn on and the last device to turn off in an embedded system. Upon system power-up, the MachXO3D securely boots and self-checks to make sure only authenticated firmware is running on it. The MachXO3D can also check and verify the firmware associated with all of the other devices in the system, and it won't release those devices until it has verified the bona fides of the firmware associated with them.

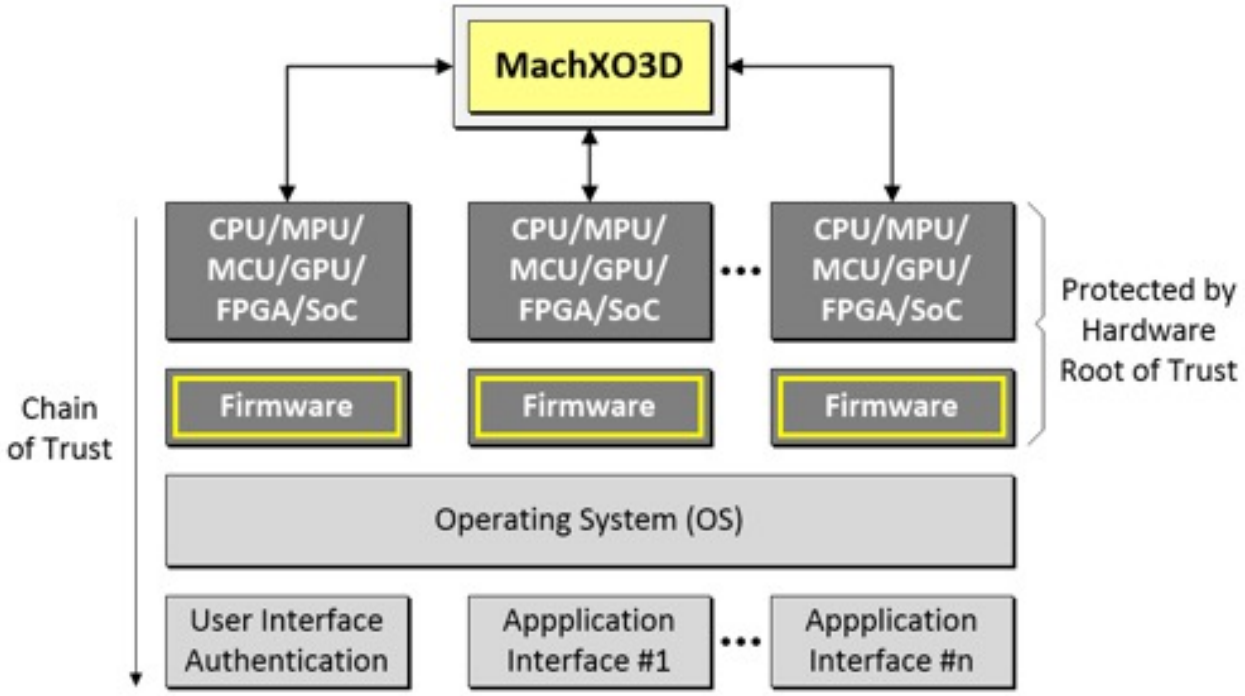

Figure 3. MachXO3D-based cyber resiliency solution.

Once the system is up and running — compliant with NIST SP 800 193 Platform Firmware Resiliency (PFR) guidelines — the MachXO3D continues to maintain cyber resiliency by protecting, detecting, and recovering itself from malicious attacks. Furthermore, the massively parallel processing capability of its programmable fabric gives the MachXO3D the ability to protect, detect, and recover multiple other platform firmware elements at the same time, especially those that are less trustworthy.

Generally speaking, a policy engine allows a system's designers and managers to create, monitor, and enforce rules regarding how computational and network resources and the organization's data can be accessed. In the context of cyber resiliency, the associated policy engine allows the system's designers and managers to specify what should happen when a cyberattack is detected.

By means of the Lattice Sentry solution stack, the MachXO3D can actively monitor in real-time multiple channels (e.g., I2C and SPI busses) looking for anomalous behavior. Designers have the choice of building a policy engine inside the MachXO3D, or of implementing the policy engine at a higher level, in which case the MachXO3D may be regarded as a primitive of that policy engine. With regard to recovery, any actions may be orchestrated by the higher-level policy engine, or they can be actioned as part of the MachXO3D's automated response ("I saw you trying to write to that device. You should never try to write to that device. I'm going to put you into reset, re-authenticate your firmware, and — if necessary — restore you to your previous 'known good' state").

As illustrated in Figure 3, the HRoT is the first link in chain of trust that protects the entire system, but how is it possible to guarantee that the HRoT is itself trustworthy? This leads to the final piece in the cyber resiliency jigsaw — securing the supply chain.

Securing the Supply Chain

The term supply chain refers to the system of organizations, people, activities, information, and resources involved in creating a product or service and supplying it to the end user. In the case of an embedded system, this may involve one or more contract manufacturers, OEMs, system integrators, and distributers, just to get the product into the end user's hands. As a result, today's global supply chains are extremely distributed and highly complex, and there are many points in the chain where bad things can, and do, happen.

Current supply chains make use of asymmetric (private/public key) cryptography in which intellectual property (IP) in the form of firmware is locked by its creator/owner using a public key. At some stage in the supply chain, this firmware has to be loaded into the part. This requires sharing the public key with a third party, such as the contract manufacturer, who will make use of a hardware security module (HSM) to load the encrypted firmware into the part.

The problem is knowing who to trust. Supply chain security holes have been exploited for years. As a result, fraudulent firmware can be loaded at any point in the supply chain into any type of programmable integrated circuit (IC), including central processing units (CPUs), microprocessor units (MPUs), microcontroller units (MCUs), graphics processing units (GPUs), and even field-programmable gate arrays (FPGAs).

A classic example is Zombie Zero, which many believe to be sponsored by a nation state that took a contract manufacturer located in that state and paired it with a hacker organization with the goal of breaking into the enterprise resource planning (ERP) systems of Fortune 100 companies around the world. Zombie Zero was extremely successful. The contract manufacturer loaded compromised firmware into handheld barcode scanners that were distributed to Fortune 100 companies. As soon as one of these scanners was installed on the wireless network inside a company's firewall, it mounted a sophisticated attack against that company's internal network. The end result was to exfiltrate every piece of scanned data (item type, source, destination address, and more) back to a botnet in China.

So, how can any company build a product that requires trusted partners when there is no guarantee that all (or any) of those partners can be trusted. Securing the MachXO3D is the critical element because this device forms the core of the HRoT. In order to address this conundrum, Lattice has come up with a game-changer solution that is based on the flash-based MachXO3D's dual-boot capability.

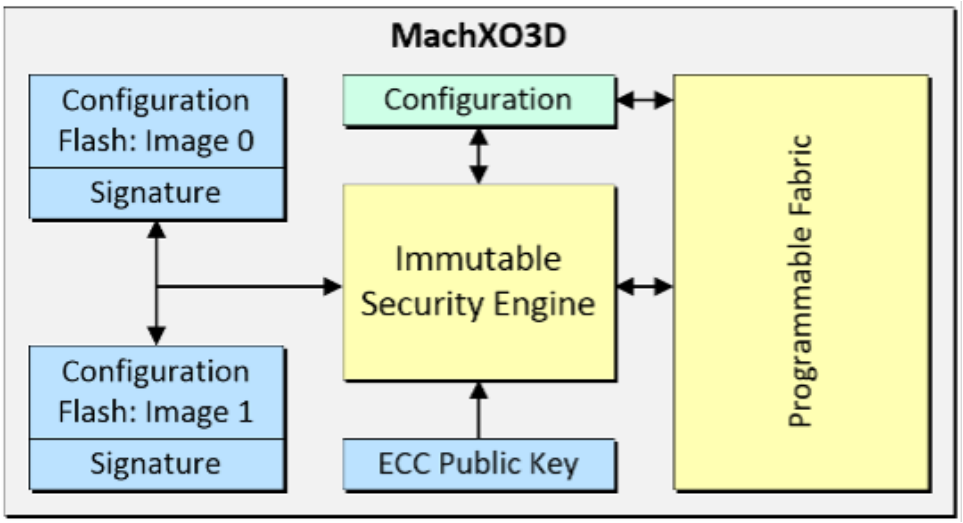

Figure 4. In addition to fully addressing the requirements of NIST SP 800 193 for cyber resiliency, the MachXO3D's dual-boot capability is key to securing the supply chain.

Like any other FPGA, a “raw” MachXO3D supports multiple programming mechanisms via a variety of ports (interfaces). In order to secure the supply chain, Lattice can protect the MachXO3D by loading it with a locking program that includes a cryptographic key based on asymmetric cryptography. This locked device can then be delivered to the manufacturer of choice.

One aspect of the locking program is that it disables all of the programming ports. The only way for the device to be reprogrammed is for it to be provided with firmware encrypted with the corresponding key. In a worse-case scenario, if sophisticated hackers were to physically deconstruct the MachXO3D layer-by-layer, all they could acquire would be the locking program itself, which would be of no use to them whatsoever. The customer’s IP is never in the part prior to first manufacture. Nor are any of the customer’s cryptographic keys or other assets.

Lattice also ships the unlock key directly and securely to the customer (the designer/owner of the system), independent of the locked parts. The customer uses this key when encrypting the IP assets that are going to be loaded into the MachXO3D by the manufacturer. In addition to the fact that this encrypted IP cannot be read by the manufacturer, it also cannot be loaded into a “raw” MachXO3D, thereby preventing the cloning and overbuilding of devices. This IP can be loaded only into a MachXO3D that already contains the locking program along with the corresponding lock key. Apart from anything else, there is no need for an expensive hardware security module (HSM) because the MachXO3D acts as its own HSM, protecting itself from modification.

This is where the MachXO3D’s dual-boot capability comes into play. The idea behind dual boot is that while one program is running, a new program can be loaded into the other flash segment. The program that’s currently running authenticates the new program against its unlock key. Once the new program has been authenticated, it will be loaded into the other flash segment, after which it will be reauthenticated to ensure it was loaded correctly and authorized.

If anyone if attempts to mount a cyberattack — such as cycling the power — while any of this is going on, the device will revert to its original locked program. It’s only after the device has been reprogrammed and has authenticated itself that the original locking program transfers ownership to the other boot segment and wipes itself and its keys out of the device permanently.

Now, this is the really clever part because — in addition to the customer’s IP — the new image also contains its own instantiation of the locking program, but this time containing the customer’s own cryptographic key(s). This means that this IP can only be overwritten by a new piece of IP that contains the corresponding unlock key, along with its own locking program and that program’s cryptographic key. This means that when the system leaves the manufacturer, it’s still locked, but this time with the customer’s own key, and — most importantly — at no stage did the customer have to share their IP or cryptographic keys with anyone else.

All of this sets the scene for what Lattice calls “Secure Ownership Transfer.” The initial customer IP may be a low-level test program. Once this has been successfully run, the customer may provide the manufacturer with a second firmware image to be loaded into the device. In this case, the new IP can only be loaded into a device that contains the previous IP. And, once again, this new IP will contain its own locking program along with its own cryptographic key(s).

The same process can follow the system as it makes its way through the system. Later, if the product is sent to a maintenance and repair facility, that facility may be provided with new IP firmware to be loaded into the device. Once again, this firmware can only be loaded into a device that already contains the unlock program with the appropriate cryptographic key(s). And, once again, this firmware will contain its own unlock program along with its own cryptographic key. And so it goes, all the way until the product is finally decommissioned.

This secure ownership transfer process ensures rigid compartmentalization that ensures that only authorized ‘owners’ have access to only their IP and cryptographic assets (i.e. keys) without contaminating or exposing a previous ‘owners.’ This dramatically increases the difficult to exploit the parts and their firmware throughout the supply chain.

Conclusion

Cyberattacks are growing in venality, velocity, and veracity. No system is immune from attack. All systems are in danger of being attacked. Traditional cybersecurity systems may prevent many attacks, but they are inadequate when systems are attacked at their lowest firmware levels.

The solution is to enhance existing cybersecurity with cyber resiliency that can protect against firmware attack, detect any ongoing firmware attacks as they happen in real-time, and restore the system to a known-good state.

A key feature to ensuring a truly cyber-resilient system is to secure that system's supply chain. And a key aspect of securing the supply chain is to ensure that no-one other than the owners of the encrypted firmware IP has access to any cryptographic keys.

The combination of Lattice MachXO3D FPGAs and the Lattice Sentry Stack secures the supply chain and provides system designers with a hardware root of trust that they can actually trust, and all this without them being obliged to trust any other entity in the supply chain.