Mach-NX: The Root of Trusted Systems

By Bob Wheeler
Principal Analyst

December 2020

www.linleygroup.com

Mach-NX: The Root of Trusted Systems

By Bob Wheeler, Principal Analyst, The Linley Group

Robust system security requires a layered approach, and the root of trust must begin with a secure boot process. Building on its leadership position, Lattice advanced its secure-control platform by introducing the next-generation Mach-NX family. These new devices keep platform security one step ahead of emerging threats while easing customer designs. Lattice sponsored this white paper, but the opinions and analysis are those of the author.

Securing Systems Starts With Firmware

In today's threat landscape, complex systems require cradle-to-grave security. That starts by securing the supply chain to make sure systems can't be compromised during device programming, system manufacturing, distribution, or installation. Once in service, systems require secure over-the-air (OTA) updates to patch vulnerabilities or upgrade security protocols. Finally, systems must be securely decommissioned to prevent data loss.

Traditional markets that prioritize system security include data centers, service providers, and critical infrastructure. Multitenant and public-cloud data centers make perimeter security obsolete, as many threats can be launched from inside. As a result, data-center systems must protect against software attacks initiated within the system. Targets include compute servers, storage systems, and network switches and routers. In service-provider networks, base stations, broadband-access equipment, routers, and various gateways are potential targets. Even when user-plane data is encrypted, attacks can compromise the management plane, creating a back door into the system.

Viewed broadly, industrial-control systems include those deployed in critical infrastructure such as defense, public utilities, the energy grid, and transportation. Governments took an early interest in securing systems deployed in these sectors, particularly as sophisticated nation-state actors would seek to compromise them. Increasingly, however, cybercriminals target similar systems in other industries for financial gain. Imagine a ransomware attack bringing a factory floor to a standstill. Another new target is automobiles, which are increasingly connected and automated, with many now receiving OTA firmware updates.

Boot-time security begins with firmware, and complex systems have multistage boot processes that create a larger attack surface. System firmware requires boot-time authentication and OTA-update encryption and authentication. When the system detects an attack or fault, it must quickly recover to a stable and secure state.

Some of these systems employ a Trusted Platform Module (TPM) to create a root of trust (RoT) where cryptographic keys are securely stored. TPM implementations vary widely, however, from dedicated hardware modules to firmware- and software-based approaches. Researchers have discovered vulnerabilities in various implementations—even those that are independently certified—that enable discovery of private keys

through practical means. Thus, even a TPM can't fully protect all system firmware against compromise.

Protect, Detect, and Recover

One means of protecting system firmware is through monitoring the Serial Peripheral Interface (SPI) signals used to read and write associated Flash memories. Figure 1 shows a server example with Flash memories attached to both the South Bridge (or PCH) and the baseboard-management controller (BMC). Inserting switch devices into the SPI paths enable a programmable-logic device (PLD) to monitor the SPI signals for commands. The PLD can validate commands and addresses against authorized- and unauthorized-access tables. When it detects an unauthorized access, it uses the switch to block the command from reaching the Flash device. It can also log these events for management code running on the BMC.

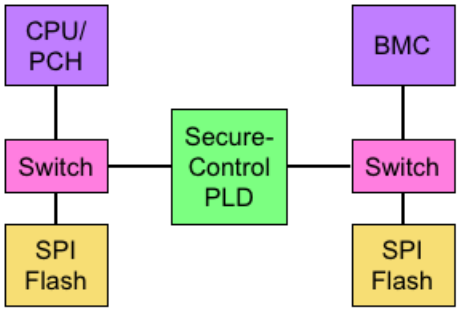

Figure 1. PFR system architecture. A secure-control PLD can implement PFR functions including SPI monitoring as well as handle control functions.

The PLD used to protect SPI access can also handle various system-control functions. These typically include power control such as sequencing, fan control, front-panel buttons and LEDs, as well as various sensing functions for power, thermal, and physical conditions. Because many of these functions use I^2C interfaces, the PLD can also buffer and multiplex signals for the BMC.

The U.S. National Institute of Standards and Technology (NIST) develops and specifies algorithms, protocols, and frameworks for network and system security. Its recent work includes guidelines for Platform Firmware Resiliency (PFR) as specified by NIST SP 800-193. The PFR core principles are *protection* of platform firmware from corruption, *detection* of corruption, and *recovery* from corruption into a state of integrity.

To protect BMC and CPU boot images, a secure-control PLD can authenticate the firmware before allowing the associated hosts to exit their reset state. Authentication involves reading the firmware data from Flash, generating a digest, reading the digital signature from Flash, and using the appropriate asymmetric cryptography to validate the result. The PFR guidelines specify cryptographic algorithms only by reference, but in practice, elliptic-curve cryptography (ECC) is now preferred, as legacy algorithms require lengthy keys. NIST's baseline mandate is 112-bit-equivalent security, which

requires 2,048-bit keys for RSA signatures. By contrast, Elliptic Curve DSA (ECDSA) requires a prime field of only 224 bits to achieve the same encryption strength. For 192-bit-equivalent strength, ECDSA uses 384 bits, whereas RSA requires 7,680 bits.

Firmware protection also includes an authenticated-update mechanism. Although not specified in the PFR guidelines, OTA updates may be encrypted for transport. Firmware-image encryption employs symmetric algorithms, and current implementations use the Advanced Encryption Standard (AES). Baseline security requires 128-bit keys (AES-128), but 256-bit keys are now common in sensitive applications. By using AES-256, the bulk encryption exceeds the strength of 384-bit ECC hashing (SHA-384) and message authentication (HMAC-384). Once the image is decrypted, its digital signature can be verified before the image is written to Flash.

The PFR guidelines include a function called the root of trust for detection (RTD). The idea behind this function is that a successful attack on system firmware or critical data should not compromise the RTD. By implementing SPI monitoring as described above, a PLD can not only serve as the RTD but can also prevent attacks that violate predefined Flash-access rules. At boot time, it can detect successful attacks by authenticating the active firmware image. When corruption is detected, the system needs to recover to an authorized state, preferably using a locally-stored firmware image. Note that the backup image can't be static, as older firmware versions may include known vulnerabilities.

After initially supplying PLDs used for system-control functions, Lattice advanced its silicon, intellectual property (IP), and software to incorporate a RoT and system-firmware protection as well as control functions in a single device. The new Mach-NX generation builds on its success with the proven MachXO3D along with specialized IP, software, and services.

Mach-NX Speeds Strong Encryption

To meet secure-control requirements, Mach-NX combines programmable and hardened logic along with copious I/Os. The chip's configuration (bitstream) is stored in on-chip Flash, which can manage two encrypted images. At boot time, the primary bitstream is authenticated while being downloaded into SRAM. If authentication fails, the device can automatically reboot from a secondary ("golden") bitstream. Mach-NX also includes User Flash Memory (UFM) for general-purpose use as well as for storing user cryptographic keys. The 1,064Kb UFM is encrypted, and it grows to 2,669Kb when dual boot is disabled.

A crucial block for system security, the Secure Enclave implements encryption protocols, a true random-number generator (TRNG), and an immutable ID unique to each device. It handles ECC protocols, including ECDSA and ECDH, with prime fields up to 384 bits. It also supports bulk encryption using AES with key lengths up to 256 bits. The block uses the TRNG in generating public and private key pairs. It provides a standard authentication interface through the Security Protocol and Data Model (SPDM) transported over the Management Component Transport Protocol (MCTP).

As Figure 2 shows, Mach-NX adds a hardened RISC-V core that runs management and control firmware. This compact 32-bit microcontroller executes the RV32I instruction set and integrates an interrupt controller, timers, and JTAG debugger. Lattice previously supplied this core as soft IP, but hardening it frees programmable logic for other functions; Mach-NX sports a total of 11K logic cells. The combination of available logic cells and UFM provide headroom for field upgrades that handle future security requirements.

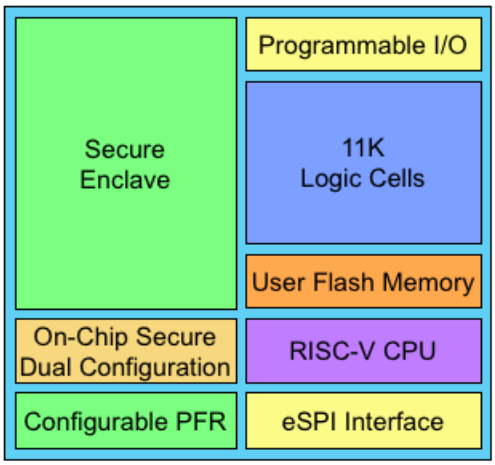

Figure 2. Mach-NX block diagram. The new chip includes a hard RISC-V CPU, a hard security block, Flash memory, programmable logic, and copious I/Os.

Another new Mach-NX feature is an Enhanced-SPI (eSPI) interface, which replaces the legacy Low Pin Count (LPC) bus as the BMC connection. The eSPI interface is backward compatible with SPI, as it layers a new protocol on top of SPI electrical and timing specifications. Like its predecessor, the new chip includes flexible I/Os for control functions, with up to 379 pins that offer LVCMOS, LVTTL, and LVDS from 1.2V to 3.3V. All package options use a 0.8mm pitch for PCB-layout ease.

Mach-NX is part of Lattice's new Nexus line, which is built in Samsung's 28nm fully depleted silicon-on-insulator (FD-SOI) process. Although FD-SOI is known for low leakage, power is a secondary concern for most secure-control applications. Another benefit of this process, however, is a large reduction in soft errors caused by radiation. Because the fabric configuration is stored in SRAM, a single-event upset (SEU) can cause unrecoverable failures. In a device of this logic density, FD-SOI practically eliminates SEUs.

Completing the Stack

To ease customers' designs, Lattice completes its secure-control platform with IP, software, and services that implement PFR and more. To configure Mach-NX with PFR IP, it supplies a drag-and-drop development tool called Propel Builder. The soft IP required in a PFR design includes a SPI master, a SPI monitor, an I²C monitor, and a register-based interface between the RISC-V CPU and programmable logic. As the name

suggests, the SPI monitor connects with external SPI-switch devices to monitor SPI-Flash accesses and block unauthorized commands. The soft PFR blocks consume 2.6K cells, leaving 8.4K cells available for user logic.

As a part of its Sentry PFR reference design, Lattice supplies source code for the firmware that runs on the embedded RISC-V core. The three major PFR-software components handle security management, log management, and out-of-band communications. Each component provides a set of APIs for application code, whereas low-level APIs provide access to the soft/hard-IP blocks. The company provides a sample application to demonstrate protection, detection, and recovery features. The Propel SDK allows customers to modify, compile, and debug the PFR firmware.

Because a production PFR design is only as secure as its supply chain, Lattice delivers a security service called SupplyGuard that works in conjunction with Mach-NX's immutable ID. The company assigns a customer-specific part number and factory-programs those devices with a cryptographic key. Using that key, the customer then programs the devices with a signed and encrypted bitstream. The device ID enables the customer's system to ensure its programming a device, which prevents scanning the bitstream. This method protects bitstream integrity as well as the customer's IP. Without the customer-specific key, the devices can't be reprogrammed. This "lock" allows the use of insecure manufacturing facilities without the risk of tampering.

Alternative Approaches

Mach-NX is a unique product, so it faces only indirect competition in PFR applications. To use an FPGA from another vendor, a customer would need to license an encryption block as soft IP from a third party. They would need to instantiate an MCU using soft IP and integrate it with the third-party block. Most FPGAs also require an external Flash configuration memory. As Table 1 shows, other FPGAs yield much lower performance for system-firmware authentication. Greater authentication time increases boot time, and boot time reduces uptime. Many cloud services include a service-level agreement (SLA) that specifies uptime, so boot time can directly impact SLA metrics. "Five nines" availability (99.999%) requires less than 5.3 minutes downtime per year, so system-boot time is important in cloud data centers.

	Lattice Mach-NX	Other FPGAs	BMC
F/W Authentication Time*	<5 seconds	>15 seconds	>10 seconds
ECC Support	Hardened	Third-party IP	None
Real-time SPI Monitoring?	Yes	Yes	No
F/W Recovery Time	<5 microseconds	>100 milliseconds	>100 milliseconds

Table 1. PFR-feature comparison for Mach-NX and alternative solutions. The Lattice chip delivers a unique combination of performance and robust security. *Time for 64MB firmware image and 33MHz SPI clock. (Source: Lattice)

Mach-NX also enables rapid recovery when system-firmware authentication fails. The device supports dual SPI memories so that one can store primary firmware while the

other keeps a golden version. If authentication fails, the Sentry firmware switches from the primary to the secondary SPI and allows the boot process to proceed. In the background, it then copies an authenticated firmware image to the primary SPI Flash. Alternative solutions must copy the firmware image before the boot process proceeds.

An alternative PFR implementation is to use the BMC, but this approach has several limitations. First, merchant BMCs lack support for elliptic-curve cryptography, so they offer weaker authentication. BMCs rely on external Flash memory, introducing supply-chain vulnerabilities that Mach-NX and SupplyGuard prevent. Finally, they lack support for inline SPI monitoring, so this feature must be added using an external PLD. Mach-NX can implement PFR as well as control functions, which require a PLD in any case.

A recent trend is the development of custom platform-security chips, but some of these are designed for smartphones and include biometric authentication. Google, however, developed a version of its Titan secure microcontroller for use in its cloud servers. Although other hyperscale cloud operators could develop similar chips, most OEMs are unwilling to expend the resources that custom-chip development requires. By providing a merchant but customizable solution, Lattice delivers similar capabilities without the burden of custom-silicon development.

Conclusion

Rather than simply delivering an FPGA, Lattice now offers a comprehensive secure-control platform that comprises optimized silicon, software, tools, and services. By examining customers' complete product cycles from system design to equipment decommissioning, it has addressed vulnerabilities at every stage. Mach-NX builds on the company's years of secure-control experience, improving on the field-proven MachXO3D by strengthening encryption, upgrading the BMC interface, and freeing programmable logic for custom features and field updates. Equally important, the Sentry reference design reduces customers' time and resource requirements to incorporate PFR in their systems.

Although servers have led adoption of robust platform security, we see an expanding market as always-on network connections expose a growing number of systems to cyberattacks. Networking and communications equipment, industrial-control systems, aerospace systems, and autonomous vehicles are all targets in the ever-increasing threat landscape. Customers across these sectors can rely on Lattice to assist them in securing their designs from manufacturing to field upgrades.

Bob Wheeler is a principal analyst at The Linley Group and a senior editor for Microprocessor Report. *The Linley Group offers the most-comprehensive analysis of microprocessors and SoC design. We analyze not only the business strategy but also the internal technology. Our in-depth articles cover topics including embedded processors, mobile processors, server processors, AI accelerators, IoT processors, processor-IP cores, and Ethernet chips. For more information, see our website at www.linleygroup.com.*