Why Now is a Good Time to Secure Your Embedded Systems with SHA-3

By Scott Jones, Managing Director, and Nathan Sharp, Sr. Business Manager, Embedded Security

November 2018

maxim integrated™

Abstract

Published by the National Institute of Standards and Technology (NIST), Secure Hash Algorithms continue to evolve to provide increasingly stronger levels of cryptography-based security. The latest iteration, SHA-3, has a new internal computational structure as compared to previous NIST-specified hash algorithms to address known vulnerabilities of its predecessors. As embedded systems become smarter and more connected, it's more critical than ever to safeguard them from attack. SHA-3 can help; however, implementing cryptographic hash functions can be challenging without a background in cryptography. This white paper discusses the merits of SHA-3 and highlights how secure authenticators designed with SHA-3 algorithms and physically unclonable function (PUF) technology can provide strong embedded security without requiring cryptography expertise.

Introduction

Safeguarding Embedded Systems

Embedded security ICs are evolving to provide more robust protection

Earlier this spring, a report was published describing how hackers used a network-connected, but unsecured, fish-tank thermometer in a casino lobby to break into the network and steal data1. The incident raises yet another spotlight on how vulnerable embedded systems can be without proper protection.

Hackers continue to get more sophisticated in their techniques to attack ICs that implement security in an embedded system. Microprobing, focused ion beam (FIB), and reverse-engineering are just a few examples of invasive attack techniques in their arsenal. Because of this, the risk is high that security implemented in software on a general-purpose microcontroller will be broken and circumvented. Encryption, for instance, might be relatively easy and cost-effective to implement in software, but for a nominal fee a hacker will extract the firmware to obtain the keys.

Hardware-based embedded security ICs provide a stronger level of protection—but even these products must continue to evolve to stay ahead of cybercriminals. An example of this evolution combines the SHA-3 cryptographic hash function, a latest generation cryptographic algorithm, with the protection provided by the technology of a PUF. PUF, together with the SHA-3 cryptographic hash function, provides a powerful combination to prevent counterfeiting, securely manage the lifecycle of an end product, store and ensure the integrity operating parameters of a sensor or tool, enable or disable subsystem features, and also to safeguard these embedded systems from invasive attacks. Let's examine both technologies in the next sections of this paper.

SHA-3: Robust Challenge-and-Response Authentication

Cryptographic hash algorithms turn an input digital message into a short message digest that can then be used in digital signatures and other security applications. A change in the original message—even a single bit—results in a significant change in value to the digest; this is called the avalanche effect. Because of this, it's fairly easy to detect either accidental or intentional changes made to the original message. Additional properties of cryptographic hash algorithms include: 1) they are one-way functions, so you cannot obtain the input from the output value; 2) the probability is near zero that more than one input message will create the same digest output (an occurrence that cryptographers call a "collision")[2,3]. Ultimately, cryptographic researchers found vulnerabilities with the first iteration of SHA, SHA-1, in terms of finding a collision. By that time, NIST had approved SHA-2 and, while SHA-2 shares a similar mathematical implementation as SHA-1, it is still an approved NIST algorithm providing better protection compared to SHA-1.[4]

Released by NIST on August 5, 2015, SHA-3 is based on the KECCAK cryptographic function, which consists of a structure that utilizes sponge construction[5]. Sponge construction represents a class of algorithms that take (absorb) an input bit stream of any length to produce (squeeze) an output bit stream of any desired length. Sponge functions can be used to model or implement cryptographic hashes, message authentication codes, and other cryptographic primitives. The KECCAK function is considered to be strong due to its intricate, multi-round permutation *f*, the function that transforms the state memory[6] of the hashing algorithm.

SHA-3 is the first cryptographic hash algorithm that NIST has adopted using a public competition and vetting process. NIST selected the KECCAK algorithm as the foundation of the SHA-3 standard after a competition that assessed candidates on:

- Performance level, regardless of implementation
- Ability to withstand known attacks, while maintaining a large safety factor
- Ability to be subjected to cryptanalysis
- Code diversity[7]

An additional advantage of SHA-3 is its silicon implementation efficiency. This makes it cost-effective compared to other algorithms and optimal for securing embedded sub-systems, sensors, consumer electronics, etc[8].

PUF Technology Protects Against Invasive Attacks

The security advantages of PUF technology stems from the fact that it is derived from the complex and variable physical and electrical properties of ICs. Since PUF depends on unpredictable, uncontrollable, random physical factors that get introduced in the IC manufacturing process, it is virtually impossible to duplicate or clone. PUF technology natively generates a digital fingerprint for its associated IC; this fingerprint can be used as a unique key, or secret, to support algorithms for authentication, identification, anti-counterfeiting, hardware-software binding, and encryption/decryption.

The way that PUF is implemented differs from vendor to vendor. Maxim's approach ensures that the unique binary value generated by each PUF circuit is guaranteed to be repeatable over temperature and voltage and as the device ages. Called ChipDNA™ technology, Maxim's PUF circuit relies on the naturally occurring random analog characteristics of fundamental MOSFET devices to produce the cryptographic keys. This implementation of PUF provides a high level of security because the unique binary value is generated only when needed by the PUF circuit and is not stored anywhere on the chip. As such, any attempts to invasively break into the IC to discover the secret key are useless. In addition, if a device with ChipDNA technology does face an attack, the attack itself can cause the electrical characteristics of the PUF circuit to change, further impeding the intrusion.

Robust Security Without Cryptography Expertise

For someone without a background in cryptography (which is not uncommon in the world of embedded systems design), implementing hash functions and symmetric key-based authentication comes with a level of complexity. Using an embedded security IC with SHA-3 functions and PUF technology built in alleviates the challenges, providing robust embedded security without requiring cryptography expertise.

Maxim's newest secure authenticator is the first such device on the market with a SHA3-256 cryptographic engine.

The DS28E50 DeepCover® secure authenticator also features ChipDNA PUF technology. Equipped with this combination of security functions, the device can be integrated into an embedded system to prevent counterfeiting, aftermarket cloning, unauthorized usage, and invasive attacks. A single-contact 1-Wire® bus simplifies communication with the end application. The DS28E50 can be used with a coprocessor, which would offload the design's host processor from running the SHA-3 algorithm and securely storing the system key. For implementations without the coprocessor, Maxim offers software that can be integrated into the design to handle these functions. The device is available in a 3mm x 3mm TDFN package. See Figure 1 for a functional diagram.

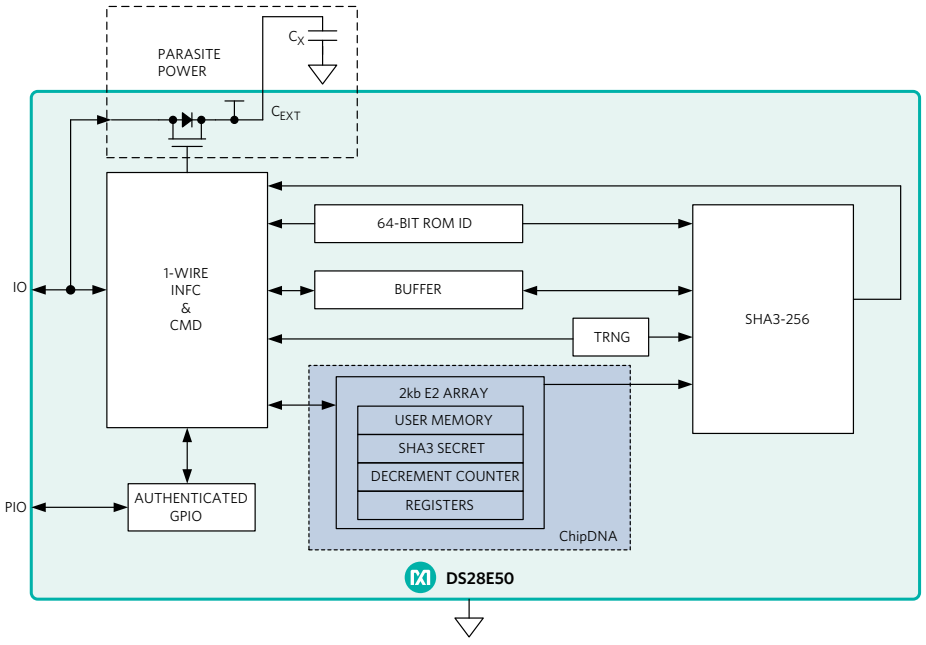

Figure 1. DS28E50 functional diagram

Where Can Secure Authenticators Be Used?

In addition to anti-counterfeiting, anti-cloning, and usage control functions, secure authenticators provide many other applications. For example, they can be used to secure end-customer feature upgrades, to manage third-party vendors, and for secure boot/software updates. These devices do so via features such as bi-directional authentication, secure memory, encrypted system data storage, secure use counting, system session key generation, secure general-purpose IOs, NIST-compliant random numbers, and the integration of public or secret key algorithms.

The use cases for secure authenticators are wide-ranging; here are a few that illustrate their value:

In an electrosurgical application, depicted by the diagram in Figure 2, it's critical to ensure that the medical device is genuine, has not been used beyond its defined limits, and has not been used in any unauthorized manner. A secure authenticator enables device manufacturers to cryptographically prove that the sensor in their device is genuine, to enforce usage control limits, and to ensure that their device is used as intended.

Secure authenticators can be integrated into an array of applications

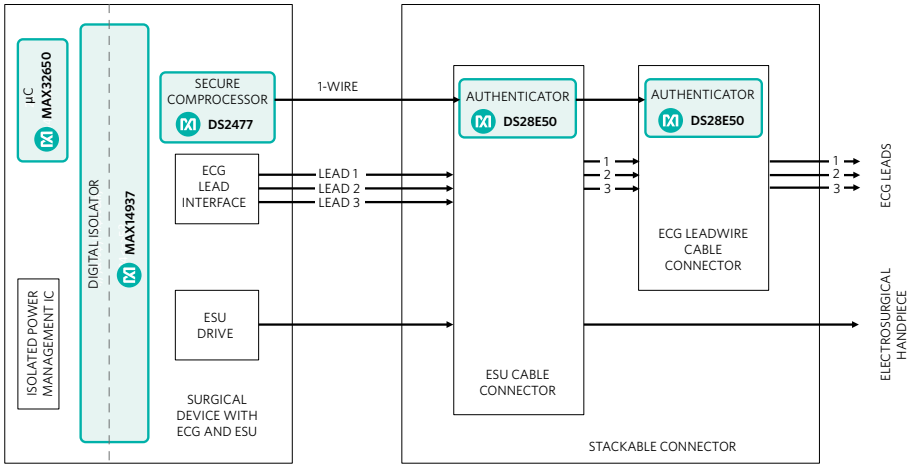

Figure 2. Electrosurgical application

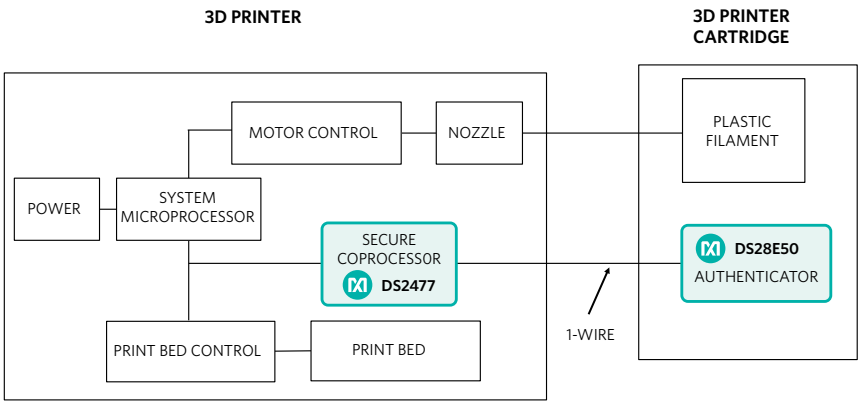

Figure 3. 3D printer cartridge authentication

Bi-directional authentication, secure memory, and encrypted system data storage are just a few key functions of secure authenticators

Cloning of printer cartridges can be costly to the original manufacturers and, if quality is compromised, to consumers. As depicted in Figure 3, secure authenticators embedded in the 3D printer and its accompanying cartridges can ensure authenticity and protect the IP from counterfeiting and illegal copying via a SHA-3 challenge-and-response approach.

As Figure 4 depicts, a secure authenticator can be used for secure boot or secure download of a data file, verifying signatures to ensure that the data shared between the host and the remote device is valid before triggering the execution (or a reset or shutdown should the signature fail).

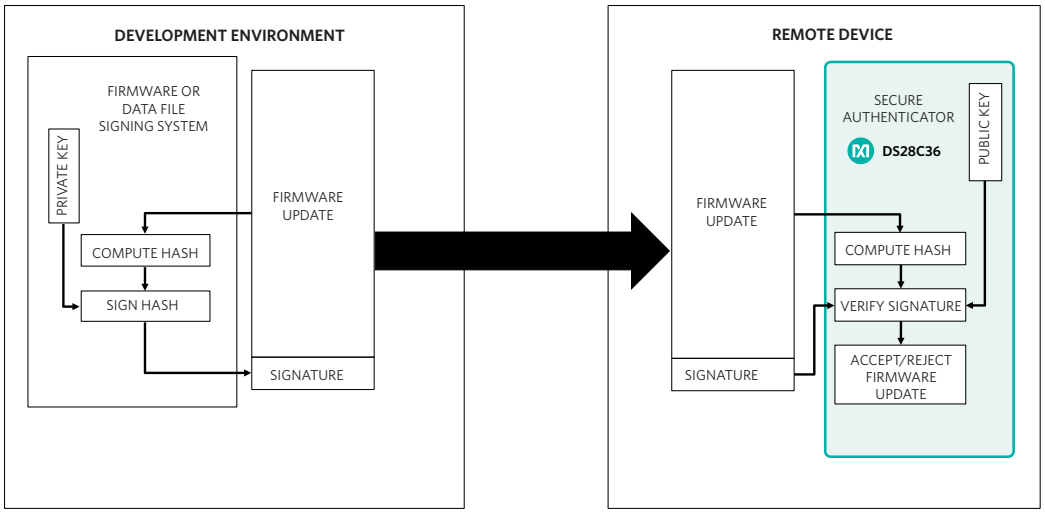

Figure 4. Secure boot/download

Summary

Whether in medical consumables, industrial, consumer, or an array of other applications, embedded systems continue to be vulnerable to the prying reach of increasingly sophisticated cybercriminals. Embedded security ICs can protect these products from counterfeiting, cloning, unauthorized usage, invasive attacks, and other security threats. Many of these devices can be integrated without requiring cryptography expertise, allowing you to focus on your core competencies. By preventing security attacks from the ground up, you can build the consumer trust that is critical to any product success.

For More Information

Learn more about ChipDNA PUF technology from use cases, a video, a webinar, and white papers.

Sources

1. Hackers Stole a Casino's High-Roller Database Through a Thermometer in the Lobby Fish Tank
2. How Does a Hashing Algorithm Work?
3. Cryptographic Has Function
4. Why Aren't We Using SHA-3?
5. The Sponge and Duplex Constructions
6. Sponge Function
7. Keccak: The New SHA-3 Encryption Standard
8. NIST Released SHA-3 Cryptographic Hash Standard

Learn more

For more information, visit:
www.maximintegrated.com