The Seven Properties of Highly Secure Devices

Galen Hunt, George Letey, and Edmund B. Nightingale
Microsoft Research NExT Operating Systems Technologies Group

ABSTRACT

Industry largely underestimates the critical societal need to embody the highest levels of security in every network-connected device—every child's toy, every household's appliances, and every industry's equipment. High development and maintenance costs have limited strong security to high-cost or high-margin devices.

Our group has begun a research agenda to bring high-value security to low-cost devices. We are especially concerned with the tens of billions of devices powered by microcontrollers. This class of devices is particularly ill-prepared for the security challenges of internet connectivity. Insufficient investments in the security needs of these and other price-sensitive devices have left consumers and society critically exposed to device security and privacy failures.

This paper makes two contributions to the field of device security. First, we identify seven properties we assert are required in all highly secure devices. Second, we describe our experiment working with a silicon partner to revise one of their microcontrollers to create a prototype, highly secure microcontroller.

Our experimental results suggest that in the near future even the most price-sensitive devices should be redesigned to achieve the high levels of device security critical to society's safety. While our first experimental results are promising, more ongoing research remains and we seek to enlist the broader security community in a dialog on device security.

1. INTRODUCTION

The next decade promises the universal democratization of connectivity to every device. Significant drops in the cost of connectivity mean that every form of electrical device—every child's toy, every household's appliances, and every industry's equipment—will connect to the Internet. This Internet of Things (IoT) will drive huge economic efficiencies; it will enable countless innovations as digital transformation reaches across fields from childcare to eldercare, from hospitality to mining, from education to transportation. Although no person can foresee the full impact of universal device connectivity, anticipation of this new frontier is widespread [1] [2].

Industry largely underestimates the critical need for the highest levels of security in every network-connected device. Even the most mundane device can become dangerous when compromised over the Internet: a toy can spy or deceive [3], an appliance can launch a denial of service [4] or self-destruct, a piece of equipment can maim or destroy [5]. With risks to life, limb, brand, and property so high, single-line-of-defense and second-best solutions are not enough.

We don't want to be alarmists. Although the state-of-the-art of security of internet-connected devices leaves much to be desired, we are quite optimistic for the future of device security. We believe it is within the realm of achievability for all devices, even the most price sensitive, to be engineered with

sufficient security to be trustworthy even in the face of aggressive assault from determined network attackers.

Our fears and our hopes for connected device security are grounded in decades of Microsoft experience as an active defender in the Internet security battle. Early attacks against network devices motivated Microsoft to pioneer automated remote update of devices in the field in Windows 95 [6]. Ongoing, evolving attacks motivated Microsoft to pioneer automated reporting and analysis of security attacks against Windows devices starting with Windows XP [7]. The desire to avoid in-field vulnerabilities continues to motivate Microsoft to create technologies and automated tools to detect and address vulnerabilities at design time [8] [9].

The goal of our research is to enable manufacturers, regardless of industry, to incorporate the highest levels of security in every network-connected device. We have identified seven necessary properties of highly secure, network-connected devices: a hardware-based root of trust, a small trusted computing base, defense in depth, compartmentalization, certificate-based authentication, security renewal, and failure reporting (in Section 2). For any network-connected device to be secure, we assert it must possess all seven of these properties. To implement these seven properties, the hardware and software (firmware) of the device must work together, with device security rooted in hardware, but guarded with secure, evolving software.

We find these security properties especially lacking in microcontroller-based devices. Some microcontroller families are beginning to evolve security features in hardware, such as cryptographic engines. However, just providing cryptographic acceleration or private key storage isn't enough to create a highly secure device if the microcontroller doesn't also provide defense in depth or compartmentalization. Overall, traditional microcontrollers lack sufficient security features to support implementation of devices with all seven properties of highly secure devices.

To address the security challenges facing network-connected devices that are powered by microcontrollers, we enlisted the help of MediaTek to revise one of their existing microcontrollers to create a Sopris, a proof-of-concept highly secure microcontroller (described in Section 4). Sopris is an experimental chip that allows us to explore the ability to create experimental microcontroller-powered systems that embody the seven properties of highly secure devices. The key hardware innovations in Sopris[1] are the addition of a security subsystem and the inclusion of a memory management unit (MMU) in the primary processor of the microcontroller. These innovations create a microcontroller architecture that we believe if combined with appropriate software would allow the creation of highly secure devices.

2. Properties of Highly Secure Devices

Building secure devices is challenging. From observation of existing best-in-class devices, we argue it is more of a science than an art. If one adheres rigorously to well-understood principles and practices, building secure devices is repeatable. We have identified seven properties we assert must be shared by all highly secure, network-connected devices: a hardware-based root of trust, a small trusted computing

[1] The name comes from the twin-summit Mount Sopris in the Elk Mountains of western Colorado.

base, defense in depth, compartmentalization, certificate-based authentication, security renewal, and failure reporting (summarized in Table 1).

Property	**Examples and *Questions to Prove the Property***
 Hardware-based Root of Trust	Unforgeable cryptographic keys generated and protected by hardware. Physical countermeasures resist side-channel attacks.
	Does the device have a unique, unforgeable identity that is inseparable from the hardware?
 Small Trusted Computing Base	Private keys stored in a hardware-protected vault, inaccessible to software. Division of software into self-protecting layers.
	Is most of the device's software outside the device's trusted computing base?
 Defense in Depth	Multiple mitigations applied against each threat. Countermeasures mitigate the consequences of a successful attack on any one vector.
	Is the device still protected if the security of one layer of device software is breached?
 Compartmentalization	Hardware-enforced barriers between software components prevent a breach in one from propagating to others.
	Does a failure in one component of the device require a reboot of the entire device to return to operation?
 Certificate-based Authentication	Signed certificate, proven by unforgeable cryptographic key, proves the device identity and authenticity.
	Does the device use certificates instead of passwords for authentication?
 Renewable Security	Renewal brings the device forward to a secure state and revokes compromised assets for known vulnerabilities or security breaches.
	Is the device's software updated automatically?
 Failure Reporting	A software failure, such as a buffer overrun induced by an attacker probing security, is reported to cloud-based failure analysis system.
	Does the device report failures to its manufacturer?

Table 1. Required Properties of Highly Secure Devices with Examples.

Highly secure devices have a *hardware-based root of trust*. Device secrets are protected by hardware and the hardware contains physical countermeasures against side-channel attacks. Unlike software, hardware has two important properties that may be used to establish device security. First, single-purpose hardware is immune to reuse by an attacker for unintended actions. Second, hardware can detect and mitigate against physical attacks; for example, pulse testing the reset pin to prevent glitching attacks is easily implemented in hardware. When used to protect secrets and device correctness, hardware provides a solid root of trust upon which rich software functionality can be implemented securely and safely.

Highly secure devices have a *small trusted computing base*. The trusted computing base (TCB) consists of all the software and hardware that are used to create a secure environment for an operation. The TCB should be kept as small as possible to minimize the surface that is exposed to attackers and to reduce the probability that a bug or feature can be used to circumvent security protections. On the contrary, in less secure systems, all security enforcement is implemented in a software stack that contains no protection boundaries.

Highly secure devices have *defense in depth*. In these devices, multiple mitigations are applied to each threat. In systems with only a single layer of defense, just a single error in design or implementation can lead to catastrophic compromise. Attackers are creative; threats are often not completely anticipated, so having multiple countermeasures often becomes the difference between a secure or compromised system.

Highly secure devices provide *compartmentalization*. Compartments are protected by hardware enforced boundaries to prevent a flaw or breach in one software compartment from propagating to other software compartments of the system. Compartmentalization introduces additional protection boundaries within the hardware and software stack to create additional layers of defense in depth. For example, a common technique is to use operating systems processes or independent virtual machines as compartments. On the contrary, many low-cost devices employed an RTOS design with no software separation.

Highly secure devices use *certificate-based authentication.* Certificates, instead of passwords, are used to prove identities for mutual authentication when communicating with other local devices and with servers in the cloud. A certificate is a statement of identity and authorization that is signed with a secret private key and validated with a known public key. Unlike passwords or other authentication mechanisms that are based on shared secrets, certificates can't be stolen, forged, or otherwise used to authenticate an impostor.

Highly secure devices have *renewable security*. A device with renewable security can update to a more secure state automatically even after the device has been compromised. Security threats evolve and attackers discover new attack vectors. To counter emerging threats, device security must be renewed regularly. In extreme cases, when compartments and layers of a device are compromised by zero-day exploits, lower layers must rebuild and renew the security of higher levels of the system. Remote attestation and rollback protections guarantee that once renewed, a device cannot be reverted to a known vulnerable state. A device without renewable security is a crisis waiting to happen.

Highly secure devices have *failure reporting*. When a failure occurs on these devices, a failure report is collected automatically and sent to a failure analysis system in a timely manner. In the best case, a failure is triggered by imperfect programming for an extremely rare sequence of events. In the worst case, a failure is triggered by attackers probing for new attack vectors. Whatever the case, a failure analysis system correlates failure reports that have similar root causes. With a sufficiently large reporting base, even extremely rare failure events can be diagnosed and corrected, and new attack vectors can be identified and isolated before they are widely exploited. Failure reporting creates a global 'immune system' for highly secure devices. Without failure reporting, device manufacturers are left in the dark as to the device failures experienced by their customers and may be caught off guard by emerging attacks.

3. EXPERIMENT: A HIGHLY SECURE, LOW-COST, MICROCONTROLLER-POWERED SYSTEM

The motivating hypothesis of our research is that even the most price-sensitive devices can be redesigned to become highly secure. Since a vast majority of the world's devices[2] are driven by microcontrollers, the clearest test of our hypothesis is to build a microcontroller-based device that can meet all seven properties required of highly secure devices.

3.1. CREATING A SECURE ROOT OF TRUST

Central to our work is the belief that practically any type of device can be converted into a highly secure device with a set of modest changes. We hypothesize that this can be done by including in the device an isolated hardware security module to provide a hardware-based secure root of trust and modifying the rest of the device hardware architecture to allow defense in depth and compartmentalization. We have created a hardware security module we call Pluton.[3] We believe that adding Pluton, or an equivalently-featured security module, to a device design is a significant necessary step towards creating highly-secure devices.

The Pluton security subsystem includes a Security Processor (SP) CPU, cryptographic engines, a hardware random number generator (RNG), a key store, and a cryptographic operation engine (COE). The cryptographic engines in Pluton include an AES [10] symmetric-key decryption and encryption engine, a SHA [11] hashing engine used for measuring code and checking certificates, and a public key engine for accelerating RSA [12] and ECC [13] public key operations. The hardware RNG is used to randomize the execution of the boot firmware so an adversary can't easily precisely time an attack, and for key generation and other cryptographic needs. The COE performs operations that require more than one cryptographic engine. An example of a COE command includes an attestation where a code measurement register is appended with a challenge from another device using the SHA engine, and the result is signed using the attestation private key in the key store using the public key engine.

3.2. EXPERIMENT

To create Sopris, we started from an existing state-of-the-art microcontroller, the Wi-Fi-enabled MT7687 [14], from our silicon partner MediaTek. The original MT7687 has a 192MHz CPU, 352KB RAM, 28 GPIO pins, a 12-bit ADC, a complete Wi-Fi and Bluetooth subsystem including an 802.11b/g/n radio, baseband, and MAC, and an interface for an in-package flash die (see Figure 1).

[2] According to [Databeans] over 9 billion new microcontroller-based devices were sold in 2016 alone [8].

[3] A pluton is geographic formation resulting from a mass of magma slowly cooling beneath the earth to form the heart of some mountains, such as Mt. Sopris.

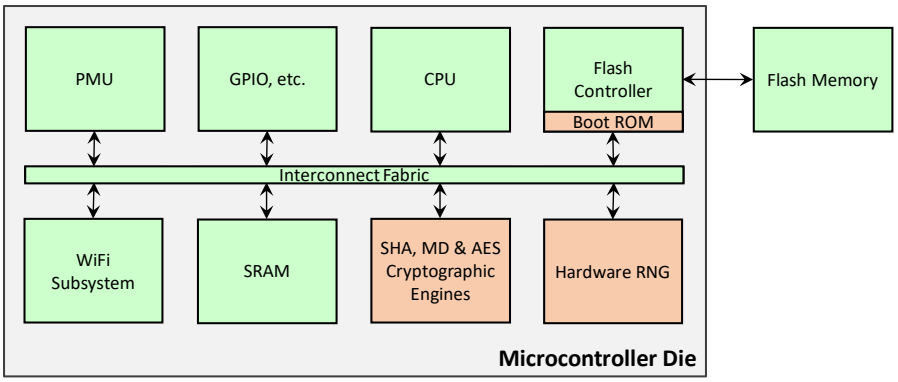

Figure 1. Architecture of the MT7687 Wi-Fi-enabled Microcontroller.

With MediaTek's assistance we modified and extended the MT7687. We made three changes to the MT7687 to convert it into Sopris (see Figure 2): we added a Pluton security subsystem, we upgraded the primary CPU to a CPU including a memory management unit (MMU), and we increased the amount of on-die SRAM. The Pluton security subsystem forms the hardware root of trust for Sopris. Unlike the much more primitive memory protection unit (MPU) found in most microcontrollers, the MMU on the Sopris processor supports multiple levels of isolation and multiple independent address spaces from which an OS can create process-isolation compartments. The addition of on-die SRAM allows easy experimentation with many OS configurations while maintaining the security of on-die memory.

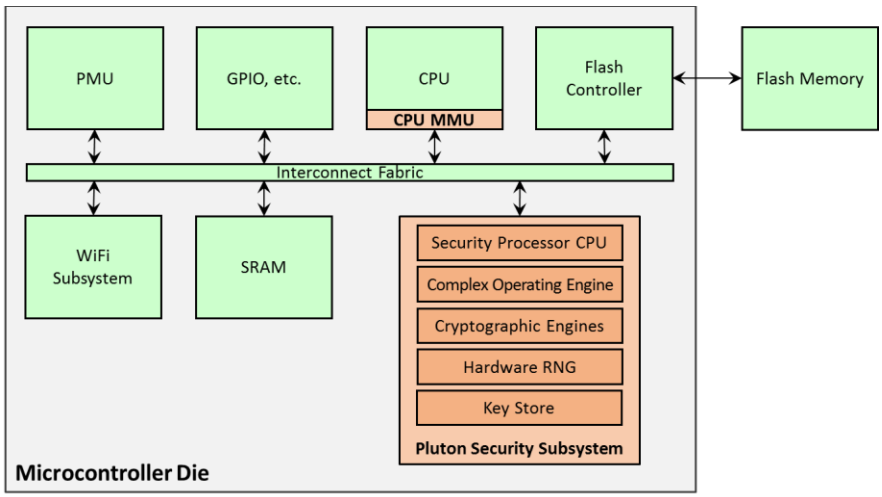

Figure 2. Architecture of the Experimental Sopris Highly Secure WiFi-enabled Microcontroller.

We have incorporated MediaTek's Sopris prototype microcontroller, with our Pluton security subsystem and MMU-enabled processor, into a small number of prototype devices. Figure 3 shows a prototype USB-powered developer board based on Sopris.

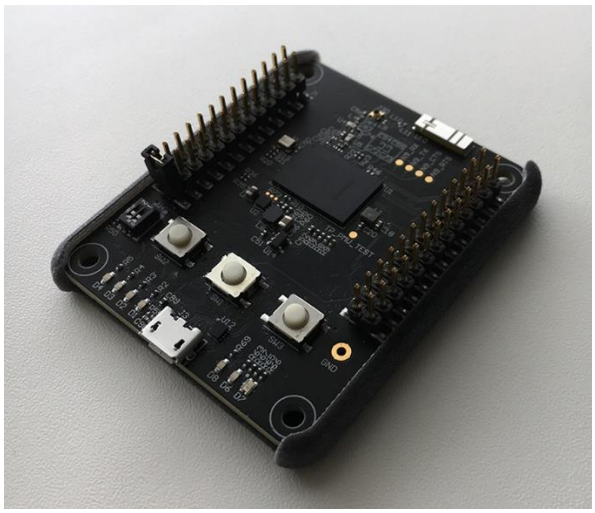

Figure 3. An Experimental Sopris-based Developer Board.

3.3. EVALUATION

Table 2 contrasts the highly-secure device properties found in Sopris with the properties found in traditional microcontrollers, such as the MT7687 from which it was derived. Sopris supports all the properties required to create a highly secure device as identified previously in Section 2.

Property	**Supported by traditional MCU**	**Supported by Sopris**
Hardware-based Root of Trust	No	Yes
Small Trusted Computing Base	No	Yes
Defense in Depth	No	Yes
Compartmentalization	No	Yes
Certificate-based Authentication	No	Yes
Renewable Security	No	Yes
Failure Reporting	No	Yes

Table 2. Properties of Highly Secure Devices supported by a state-of-the-art traditional microcontroller and our experimental Sopris chip.

The Sopris proof-of-concept provides a *hardware-based root of trust*; device secrets are protected in the Pluton security subsystem. Sopris provides a *small trusted computing base*; for most operations, the TCB for Sopris is isolated to the Pluton security subsystem. Sopris supports *defense in depth*; between the upgraded CPU and the Pluton security system up to seven layers of defense are supported in Sopris. Sopris supports *compartmentalization;* for example, separate compartments can be implemented using isolated address spaces enabled by the upgraded CPU. Sopris supports *certificate-based authentication*; for example, private keys stored in the Pluton security subsystem can form the basis of a secure per-device certificate chain. Sopris supports *renewable security*; for example, a software stack running on Sopris can use the multiple layers of hardware-protected defense in depth to implement renewable security. Finally, Sopris supports *failure reporting*; for example, failure handling code that runs on the Sopris can collect data about failures and relay that information to a failure analysis service through Wi-Fi.

In comparison to Sopris, existing microcontrollers do not support the properties required to create highly secure devices. For example, even though a microcontroller might include hardware crypto engines, keys directly usable by software in the microcontroller's primary processor are insecure if a flaw is found in that software. Lacking a small TCB, defense in depth, and compartmentalization, most microcontrollers use a single-compartment RTOS. In such systems, virtually any software vulnerability becomes a fatal flaw that allows the attacker to breach the single security layer and gain complete and permanent control of the device.

Whereas traditional microcontrollers are ill-fitted to the task of creating highly secure devices, the Sopris experiment proves that it is possible to construct a microcontroller that can readily provide the basis for highly secure devices.

4. Conclusion and Future Work

The coming decade will likely see the deployment of billions upon billions of network-connected devices. Although we applaud those in the industry who have begun to recognize the critical importance of security in these coming devices, we believe that many fail to appreciate the need to give each of these devices the highest levels of security available.

The desire to bring the highest level of security to even the most pedestrian devices has led us to think deeply about the properties of highly secure devices. We have identified seven properties we assert are critical in all highly secure, network-connected devices: hardware-based root of trust, small trusted computing base, defense in depth, compartmentalization, certificate-based authentication, security renewal, and failure reporting.

Grounded in the understanding of these seven properties of highly secure devices, we have set out to explore if it would be possible to bring these properties to experimental, low- cost applications. Our first research milestone has been a step in that direction: building a test device that utilized a modified proof-of-concept microcontroller with these properties. Based on an initial, property-based evaluation (Section 3.3), we believe that one could design systems that are highly secure using this design and appropriate software.

For the next phase of research evaluation, this approach is being packaged into a simple device board design with software that we hope to share with researchers and security experts across academia and industry. We look forward to learning together the strengths, weaknesses and potential of such an approach with the aid of the broader security community, and continue to share our learnings. We will share variations on the core architecture to further validate our hypothesis of the seven properties of highly secure devices.

Based on our preliminary experimental experience, we are hopeful that almost any device can be redesigned to achieve high levels of device security—levels that will be critical to society's safety in the near-future.

ACKNOWLEDGEMENTS

The Sopris experiment builds heavily on the work of others and we are grateful for their long hours and dedication from which we have benefited. We owe a special debt of gratitude to George Jen and team at MediaTek, who provided the original MT7687 design and worked with us to create Sopris.

REFERENCES

[1] C. MacGillivray and A. Wright, "Worldwide Internet of Things Connectivity Forecast, 2017–2021," IDC, 2017.

[2] D. W. Cearley, B. Burke and M. J. Walker, "Top 10 Strategic Technology Trends for 2016," Gartner, 2016.

[3] C. Wiking, "If Your Child Has This Doll You Should Get Rid of It Now," 17 Feb. 2017. [Online]. Available: https://mom.me/news/39826-if-your-child-has-doll-you-might-want-destroy-it/. [Accessed 17 Feb. 2017].

[4] N. Perlroth, "Hackers Used New Weapons to Disrupt Major Websites Across U.S.," *New York Times,* 21 Oct. 2016.

[5] E. Mills, "Internet-Connected Coffee Maker Has Security Holes," *CNET,* 17 Jun. 2008.

[6] C. Gkantsidis, T. Karagiannis, P. Rodriguez and M. Vojnovic, "Planet Scale Software Updates," in *ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Pisa, 2006.

[7] K. Kinshuman, K. Glerum, S. Greenberg, G. Aul, V. Orgovan, G. Nichols, D. Grant, G. Loihle and G. Hunt, "Debugging in the (Very) Large: Ten Years of Implementation and Experience," *Communications of the ACM,* vol. 54, no. 7, pp. 111-116, 2011.

[8] E. Bounimova, P. Godefroid and D. Molnar, "Billions and Billions of Constraints: Whitebox Fuzz Testing in Production," in *International Conference on Software Engineering*, San Francisco, 2013.

[9] P. Godefroid, P. de Halleux, M. Y. Levin, A. V. Nori, S. K. Rajamani, W. Schulte and N. Tillmann, "Automating Software Testing Using Program Analysis," *IEEE Software,* vol. 25, no. 5, pp. 30-37, 2008.

[10] National Institute of Standards and Technology, "197, Advanced Encryption Standard (AES)," *Federal Information Processing Standards (FIPS),* 2001.

[11] National Institute of Standards and Technology, "180-4, Secure Hash Standard," *Federal Information Processing Standard (FIPS),* 2012.

[12] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Strong Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM,* vol. 21, no. 2, pp. 120-126, 1977.

[13] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the Theory and Application of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 1985.

[14] MediaTek, Inc., "MT7687F Datasheet," Hsinchu, 2016.

[15] Databeans, Inc., "Q1 2017 Microcontroller Market Tracker," Reno, 2017.