INVITED: Can IoT be Secured: Emerging Challenges in Connecting the Unconnected

Nancy Cam-Winget[1], Ahmad-Reza Sadeghi[2], Yier Jin[3]
[1]Cisco Systems, Inc
[2]Technische Universität Darmstadt, Germany
[3]University of Central Florida
ncamwing@cisco.com, ahmad.sadeghi@trust.cased.de, yier.jin@eecs.ucf.edu

ABSTRACT

Embedded, mobile, and cyberphysical systems are becoming ubiquitous and are used in many applications, from consumer electronics, industrial control systems, modern vehicles, to critical infrastructures. Current trends and initiatives, such as Internet of Things (IoT) and smart cities, promise innovative business models and novel user experiences through strong connectivity and effective use of next generation embedded devices. These systems generate, process, and exchange vast amount of security-critical and privacy-sensitive data, which makes them attractive targets of attacks. Cyberattacks on IoT systems are highly critical since they may cause physical damage and threaten human lives. The complexity of these systems, the lack of security and privacy by design for current IoT devices, and potential impact of cyberattacks will bring about new threats. This paper gives an overview on the related security and privacy challenges, and an outlook on possible solutions towards a holistic security framework for IoT systems.

1. INTRODUCTION

Current commercial and industrial trends and initiatives aim to "connect the unconnected." Today, millions of embedded devices are used in safety and security critical applications such as industrial control systems, modern vehicles, and critical infrastructures. This network of ubiquitous smart objects is known as the Internet of Things (IoT) and enables novel applications and services, in both commercial and industrial sectors [36, 59, 37]. The number of computation components integrated into industrial control systems, production systems, and factories is steadily increasing. Programmable logic controllers are replaced by the more advanced cyberphysical systems (CPS), which are programmable embedded devices that control physical processes. CPS typically communicate over closed industrial communication networks but are increasingly often connected to the Internet.

With the evolution of IoT leveraging classical computing and enabling production systems to communicate over the Internet, emerging megatrends such as mobile computing, cloud computing, and Big Data, are becoming important drivers of innovation in various sectors of our society. Cloud-based services are used to monitor and optimize complex supply chains; Big Data algorithms predict machine failures, which reduces downtime and maintenance costs; interconnected production systems enable tight integration and optimization of production and business processes as well as outsourcing production steps to other locations, companies, and freelancers. In the near future, cloud-based services will allow considering more customer requirements in the production process and planning, enabling a new level of product individualization at a minimal cost [24].

IoT devices generate, process and exchange vast amounts of control and safety-critical data that affects both the overall security and privacy of the underlying systems and the humans interacting with those systems. The connectivity to Internet makes them appealing targets of various attacks [43, 42, 60, 23, 27, 22, 21]. To ensure the correct and safe operation of IoT systems, it is crucial to assure the integrity of the underlying devices, in particular their code and data, against malicious modifications [64]. Recent studies have revealed many security vulnerabilities in embedded devices [12, 16, 27, 11, 22] that are core components of the IoT. This poses new challenges on the design and implementation of secure and privacy-enhancing embedded systems that typically must provide multiple functions, security features, and real-time guarantees at a minimal cost.

In this paper, we give an overview of the security and privacy issues associated with the development of IoT systems in Section 2, and discuss potential solutions and recent research directions for securing IoT devices in Section 3. Finally, conclusions are drawn in Section 4.

2. SECURITY & PRIVACY CHALLENGES

Through connecting the unconnected, IoT promises to enable real-time remote controls and monitoring of production systems such as conditioning monitoring, structural health monitoring, remote diagnosis and productivity controls. IoT also becomes the basis of smart factories that dynamically organize and optimize production processes with regard to resource-utilization (i.e., costs, availability, material, and labor) based on data generated and collected by the underlying cyberpyhsical systems (CPS), even across company boundaries [65]. In smart factories, smart products know their own identity, history, specification, documentation, and even control their own production process (cf. Figure 1).

DAC '16, June 05 - 09, 2016, Austin, TX, USA

http://dx.doi.org/10.1145/2744769.2747942.

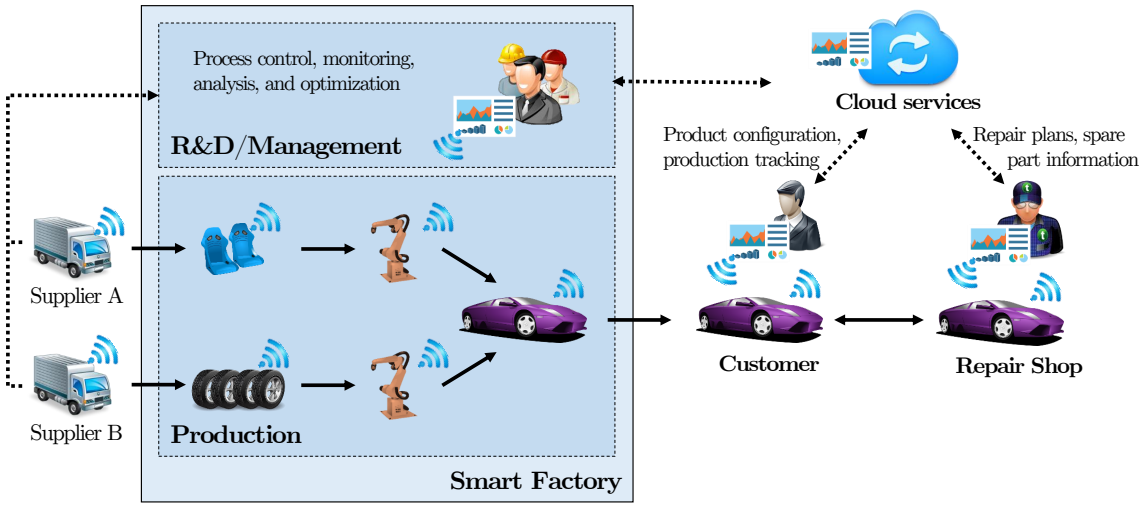

Figure 1: Industrial Internet of Things (IoT)

However, the IoT trend also brings about many new challenges with regard to different aspects including security, privacy, standardization, legal, and social aspects. In particular the increased diversity and large numbers of devices in IoT systems require highly scalable solutions, for instance, naming and addressing, data communication, knowledge management, and service provisioning. Furthermore, most IoT devices have only limited resources and must be used in architectures supporting low power, low cost, fully networked integrated devices that are compatible with standard communication protocols. Among all these emerging challenges, security and privacy threats are the main concerns which, if not properly addressed, can inhibit the overall benefits of such IoT systems. For example, an IoT-enhanced production system can be divided into multiple layers and, therefore, is vulnerable to attacks from these layers. Smart factories consist of several cyberphysical production systems (CPPS) consisting of electronics (e.g., processor and memory) and monitors that control physical processes through sensors and actuators (cf. Figure 2) [49]. The electronics are driven by software (e.g., embedded operating systems and applications) and interact with humans and other CPPS through various network connections (e.g., Ethernet or WiFi). Attack surfaces exist on all these abstraction layers as well as across them (cf. Figure 2) [54, 63, 3, 30]. Electronics are subject to physical attacks, including invasive hardware attacks, side-channel attacks, and reverse-engineering attacks [45]. Software can be compromised by malicious code, such as Trojans, viruses, and runtime attacks [55]. Communication protocols are subject to protocol attacks, including man-in-the-middle and denial-of-service attacks [28]. Also humans operating CPPS can be subject to social attacks, such as phishing and social engineering. The majority of these issues have been investigated by researchers for many years and there are a number of practical solutions that are particially deployed to reduce the affect of various attacks. However, in the promised IoT landscape we are faced with thousands and potentially millions of connected devices all over the place facing us with the challenging manageability problem of security and privacy. As a result, any effective solution should also be developed from a cross-layer perspective and take not only all system levels but also the scale into consideration.

Unfortunately, current IoT devices lack proper security and privacy-enhancing design suffering even from basic and known issues that can be solved with standard solutions. In the following we will give some examples of these problems at different system abstraction layers.

Boot Process Vulnerabilities. The boot sequence is one of the main attack targets. Compromising this component allows the adversary to compromise other high-level protection mechanisms and subsequently attempt to take over the control over the whole system. One prominent example of this type of attack is the compromise of the Google Nest Thermostat [21, 4]. Through the boot process vulnerability in the Nest Thermostat, attackers can send a modified initial boot-loader (`x-loader`) to the device, coupled with a custom full boot-loader (`u-boot`) crafted with an argument list to be passed to the on-board kernel. Arbitrary payloads can then be inserted into the device through the custom `u-boot` image [4]. Mitigation methods to this type of vulnerability were discussed in [41, 15].

Hardware Exploitation. Hardware level exploitation is a critical point for security as most security protection implementations are located at the software or firmware levels. These attacks target the hardware implementations themselves, which involve looking for debugging ports left open by manufacturers, reflashing external memory, timing attacks, etc. For example, the exploits on Xbox 360 allows systems to downgrade to a vulnerable kernel version through a timing attack [1]. Another example of this type of attack is the ID manipulation on the Itron Centron smart meter [62]. The meter stores its identity on an external EEPROM, which lacks read or write protection. By looking at the identity of the meter and cross-referencing it with the data from the EEPROM dump, the identity can be located and modified [62]. Given this information and access to the EEPROM, attackers can easily re-flash the EEPROM. As a result, the meter can be made to masquerade as any other smart meter. In order to prevent this type of attack, various countermeasures have been developed, e.g., protection methods to prevent timing attacks [9].

Chip-Level Exploitation. Chip-level exploitation of

integrated circuits, including semi-invasive and invasive intrusions are a serious threat to smart devices, as trusted boot sequences rely on trusted on-chip assets. For a long time, encryption/decryption keys, and other sensitive information was stored on-chip as it was considered a secure means of storage. Newly developed invasive methods can reveal valuable assets stored in the chip, and may compromise any protocols utilizing the secret information. For example, by "bumping" the internal memory on an Actel ProASIC3 FPGA, researchers were able to extract the stored AES key [50].

Encryption, Hash Function and Authentication Implementations. Encryption and hash functions are used in smart devices to secure passwords and other sensitive information, in addition to playing a key role in device communication and authentication. These functions are considered to provide reasonable protection level based on the modern cryptographic design principles and security models. However, improper implementations of these functions, and the utilization of cryptographically weak encryption algorithms threaten the security of the underlying devices. Similarly, many attacks today result from weak authentication mechanisms. While system designs will impose strong authentication mechanisms, e.g. x.509 certificate based TLS [58], unless the credentials (e.g., keys) are securely stored they can be subject to attack. As IoT devices are now exposed in open and public spaces, the ability for any attacker to recover such credentials becomes a trivial attack; once the keys are recovered, those identities are then compromised obviating the security properties afforded by any encryption mechanism. For example, the Sony PlayStation 3 firmware was downgraded due to a series of vulnerabilities in weak cryptographic applications [10, 31]. Another example of a weak encryption implementation causing device-level security vulnerabilities is the Haier SmartCare home automation system [62]. It uses DES encryption on the password without adding a salt. Consequently, the total keyspace size is drastically reduced and the DES password can be cracked in several hours. Interestingly, while these problems have been repeatedly found in modern smart devices, mitigation methods had already been proposed decades ago [46].

Backdoors in Remote Access Channels. Smart devices are often equipped with channels that allow for remote communication and debugging after manufacturing. These channels are also used for over-the-air (OTA) firmware upgrades. Although very useful, their implementations are not always secure. During development, manufacturers may leave in APIs which allow arbitrary command execution. Additionally developers may not properly secure the communication channel. This attack vector can be exploited to remotely obtain the status of the device, or even control the device. A modern example of a backdoor in a remote channel is the Summer Baby Zoom WiFi camera, which has hardcoded credentials for administrator access [19]. Efforts to mitigate these vulnerabilities include requiring users to change default credentials before usage, sanitizing string input to avoid remote command execution, etc.

Software Exploitation. Software-level vulnerabilities in smart devices are mostly similar to those in traditional general purpose computing systems. Because smart device software stacks are often derived from the general computing domain, any software vulnerabilities found in the general computing area will also affect these devices. Therefore, software patches are required to update smart devices against known software-level attacks. Recent examples include a stack-based buffer overflow attack in glibc [2] as well as multiple smart house devices [51]. Methods to mitigate software exploitation attacks often follow those developed in the traditional general computing areas [14, 13]. However, as discussed in [4] these solutions may not be suitable for smart devices due to their resource constraints.

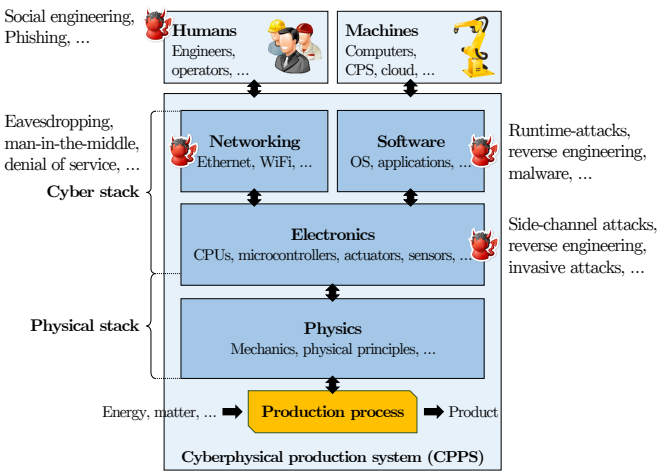

Figure 2: Cyberphysical production system (CPPS) architecture and attack surfaces

Attacks on Industrial IoT Systems. The above mentioned security vulnerabilities would not just affect the system itself but also affect the physical system controlled by IoT/CPS. One of the first successful attacks against industrial control systems was the Slammer worm, which infected two critical monitoring systems of a nuclear power plant in US in 2003 [42]. In the same year, a computer virus infected the signal and dispatching control system of a major transportation network in US leading to complete stop of passenger and freight trains [43]. In the following years, many security incidents affecting industrial control systems and critical infrastructure have been reported in literature [23, 35]. While these attacks seem not to have specifically targeted industrial control systems, Stuxnet [60, 23, 35] indicates a new trend towards highly targeted attacks and sabotage by powerful adversaries, e.g., nation states. Stuxnet exploited multiple zero-day vulnerabilities[1] and caused centrifuges at an Iranian nuclear facility to fail.

3. SECURING THE IoT

Adapting existing information security concepts to IoT systems, e.g., commercial IoT systems and cyberphysical production systems (CPPS), is not straightforward. There are many differences between classical IT systems and IoT systems. Integrity and confidentiality are primary protection goals of classical enterprise IT systems and hence, protection against cyberattacks is often a tradeoff between security and availability. For instance, in case of successful cyberattack, the affected IT systems are typically temporarily disabled and then restored after the attack. However, this

[1]Zero-day vulnerabilities are those vulnerabilities which are unknown before they are exploited, i.e., no security patches are available to fix them.

approach cannot be applied to most of the IoT systems (e.g., CPPS), where availability is a fundamental requirement.

Other differences are due to the strict real-time requirements of IoT systems, their constrained computational, memory, and energy resources, and the long lifetime of industrial production systems. Other facets of IoT design include protection of design and configuration data (intellectual property) and detection of counterfeit components (product piracy). Many industrial areas have legal requirements to have auditable logs of production steps (provenance and accountability). With the increasing number of interconnected IoT systems and the possibility to use Big Data techniques to analyze data collected by IoT systems, privacy becomes a fundamental aspect [36, 30]. For example, Big Data analysis allows enterprises, governments and malicious adversaries to learn even more about personal and sensitive information of individuals.

To counter these security and privacy risks, a holistic cybersecurity concept for IoT systems is required that addresses the various security and privacy risks at all abstraction levels. In particular, security and privacy aspects must be preserved during the lifetime of smart production systems and smart products. In the following, we will focus briefly on the recent security solutions for embedded devices which are at core of IoT. This concerns both security architectures for individual platforms as well as their collective operation.

3.1 Security Architectures for IoT System

There is a rich body of literature on security architectures for embedded IoT systems, mainly due to the broad range of devices considered as embedded systems [16, 12]. On the upper end are Intel and ARM architectures, which are widely used in mobile devices (e.g., smartphones and tablets). For these systems, a variety of security architectures have been proposed: software-based isolation and virtualization [32]; Trusted Computing based on secure hardware such as the Trusted Platform Module [57]), and processor architectures providing trusted execution environment (e.g., ARM TrustZone [61], AEGIS [53], OASIS [39], and Intel Software Guard Extensions (SGX) [33]. However, all these approaches are too complex for low-end embedded systems, which are typically designed for specific tasks and optimized for low power consumption and minimal costs. Often they must provide multiple features and meet strict real-time requirements. Security solutions for these devices require a minimal trusted computing base, for instance lightweight hardware-enforced isolation of security-critical code and data from other software on the same platform. Prominent examples of research solutions are SMART [17], SPM [52], SANCUS [38], TrustLite [25], and TyTAN [8]. SMART protects the integrity of only one specific embedded application (task) with read-only memory, which does not allow code changes after deployment. SPM provides hardware-enforced isolation of tasks by granting access to a task's data region only to the task itself. However, these tasks have a fixed memory layout and cannot be interrupted. Further, the task measurement of SPM is performed in hardware, i.e., it is non-interruptible and at the same time dependent on the memory size of the measured task, which violates real-time requirements. SANCUS extends SPM with a mechanism to generate and manage cryptographic secrets of tasks but inherits SPM's limitations. TrustLite generalizes the concept of SPM [52] and SMART [17] and supports interrupting tasks. However, TrustLite requires all software components to be loaded and their isolation to be configured at boot time. In contrast TyTAN [8] provides dynamic loading and unloading of multiple tasks at runtime, secure inter-process communication (IPC) with sender and receiver authentication, and real-time scheduling.

3.2 Integrity Verification of IoT Systems

A key mechanism to verify integrity of a system's software configuration is *attestation*, which enables the detection of unintended and malicious software modifications. Various approaches to remote attestation have been proposed to-date. Common to all of them is that the device to be attested, called *prover*, sends a status report of its current software configuration to another device, called *verifier*, to demonstrate that it is in a known and, thus trustworthy, state. Since malicious software on the prover's platform could forge this report, its authenticity is typically assured by secure hardware [57, 18, 29, 26] and/or trusted software [48, 47, 29] as *trust anchor*. Attestation based on secure hardware components is most suitable for advanced computing platforms, such as smartphones, tablets, laptops, personal computers, and servers. However, the underlying security hardware is often too complex and/or expensive for low-end embedded systems. In contrast, software-based attestation [48, 47], does not require secure hardware or cryptographic secrets. However, security guarantees of software-based attestation are relying on strong assumptions, such as (1) the adversary being passive while the attestation protocol is executed, and (2) optimality of the attestation algorithm and its implementation (timing aspect) so that no malware can perform faster. Moreover, software-based attestation assumes that the device to be attested has been already authenticated (e.g., by optical means). Such assumptions are hard to achieve in practice [5]. Consequently, software-based attestation has very limited applications in practice and is not suitable for *remote* attestation. Hence, a secure and practical attestation scheme requires at least some basic security features in hardware but these should be kept as small as possible [18, 26].

The next generation of IoT systems will constitute *device swarms*, i.e., large self-organizing and heterogeneous networks of collaborative embedded devices. Verifying correct and safe operation of these systems requires an efficient and scalable *swarm attestation* mechanism to collectively verify the software integrity of all devices in order to detect unintended and malicious software modifications. However, naïve applications of remote attestation do not scale to these systems. In particular, device swarms with dynamic topologies, such as vehicular ad-hoc networks, robots, drones and sensors in fluid environments, require novel and flexible solutions. As the first step towards tackling this challenge some solutions have been proposed recently in [40] to attest multiple provers running the *same* software at once, or in [6] to collectively attest a very large number of connected devices. The design of an efficient attestation scheme for large dynamic and heterogeneous networks of embedded systems is a challenging open research problem.

3.3 Secure IoT Device Management

Many IoT devices (such as sensors) do not have appropriate user interfaces or suitable communication interfaces for performing pairing using legacy solutions, e.g., PIN codes

as used in Bluetooth. Also, as the number of IoT devices grows, for example, in smart home scenarios, it becomes increasingly burdensome for the user to introduce new devices, if it involves manually pairing the new device with each existing device. This becomes even more challenging with transient pairing. Therefore, pairing of devices should be achieved with *minimal or zero user interaction.* Once a device joins a group of devices, it can collaborate with all devices in this group and access the user's and the other devices' data (device-centric authentication [20]).

New ways of establishing trust among IoT devices have been presented with the premise of strongly improving user-experience by eliminating the need for the user to explicitly specify or point out the devices to be paired with each other [44, 34, 56]. This can be achieved by utilizing the fact that devices that are located in the same place also consistently observe similar ambient context information. For example, IoT devices in the living room of a user's smart home will, for most of the time, observe similar changes in ambient contextual parameters like noise or light.

The management of IoT devices in future smart spaces will be extremely challenging due to their heterogeneity. Additionally, these devices will produce a large volume of nonuniform data that needs to be processed in real-time. In the context of secure pairing based on ambient data, local IoT systems need to process and analyze heterogenous data inputs with low latency to make appropriate decisions. Existing approaches rely on cloud-based services to perform these operations remotely. Unfortunately, critical privacy issues are raised when exporting substantial amounts of personal data to external services. Furthermore, the increasing number of devices connected to IoT will require highly scalable solutions with respect to data storage, latency of services, and management of data and devices.

Local data management and local distributed analytics are expected to improve latency of local services because only minimal information will be exchanged outside local and low-latency network. For the same reason, local analytics and data management will improve user data privacy. These features will maximize usage of resources available in IoT systems and provide building blocks for developers to create innovative services.

Performing local data management and analytics, however, raises several challenges due to diversity of devices and the need for scalable solutions. For instance, computation capacity of devices varies considerably, and thus analytical tasks cannot be distributed uniformly among IoT devices. Moreover, devices have several non-negligible constraints such as power management, constrained resources (e.g., limited computation power, storage, communication means, and energy), and permeability to attacks. Finally, interoperability between devices requires a data abstraction model supported by an extensible but lightweight API, e.g., the Representational State Transfer (REST) architecture [7].

4. CONCLUSION

Internet of Things (IoT) is an emerging technology. Today's IoT systems are not sufficiently enhanced to fulfill the desired functional requirements and bear security and privacy risks. Particularly, attacks on cyberphysical systems may cause physical damage and threaten human life.

Protecting IoT requires a holistic cybersecurity framework covering all abstraction layers of heterogeneous IoT systems and across platform boundaries. However, existing security solutions are inappropriate since they do not scale to large networks of heterogeneous devices and cyberphysical systems with constrained resources and/or real-time requirements. Further research is required to develop and design appropriate IoT security mechanisms, including novel isolation primitives that are resilient to run-time attacks, minimal trust anchors for cyberphysical systems, and scalable security protocols.

Acknowledgments

This work was supported in part by the Florida Cybersecurity Center (FC2) Collaborative Seed Grant Program, the Southeastern Center for Electrical Engineering Education (SCEEE 15-001), the German Science Foundation (project S2, CRC 1119 CROSSING) and European Union's Seventh Framework Programme (609611).

5. REFERENCES

[1] Xbox 360 timing attack. 2007. [Online]. http://beta.ivc.no/wiki/index.php/Xbox_360_Timing_Attack.

[2] Critical security flaw: glibc stack-based buffer overflow in getaddrinfo() (cve-2015-7547). 2015. [Online]. https://access.redhat.com/articles/2161461.

[3] C. Alcaraz, R. Roman, P. Najera, and J. Lopez. Security of industrial sensor network-based remote substations in the context of the internet of things. *Ad Hoc Netw.*, 11(3), 2013.

[4] O. Arias, J. Wurm, K. Hoang, and Y. Jin. Privacy and security in internet of things and wearable devices. *IEEE Transactions on Multi-Scale Computing Systems*, 1(2):99–109, 2015.

[5] F. Armknecht, A.-R. Sadeghi, S. Schulz, and C. Wachsmann. A security framework for the analysis and design of software attestation. In *ACM Conference on Computer & Communications Security (CCS)*. ACM, 2013.

[6] N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann. Seda: Scalable embedded device attestation. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[7] M. Blackstock and R. Lea. Toward interoperability in a web of things. In *ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp)*. ACM, 2013.

[8] F. Brasser, P. Koeberl, B. E. Mahjoub, A.-R. Sadeghi, and C. Wachsmann. TyTAN: Tiny trust anchor for tiny devices. In *Design Automation Conference (DAC)*. ACM, 2015.

[9] D. Brumley and D. Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.

[10] bushing, marcan, segher, and sven. Console hacking 2010: Ps3 epic fail. In *27th Chaos Communication Congress*, 2010.

[11] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Conference on Security*. USENIX Association, 2011.

[12] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti. A large-scale analysis of the security of embedded firmwares. In *USENIX Conference on Security Symposium*. USENIX Association, 2014.

[13] C. Cowan, S. Beattie, J. Johansen, and P. Wagle. Pointguard tm: protecting pointers from buffer overflow vulnerabilities. In *Proceedings of the 12th conference on USENIX Security Symposium*, 2003.

[14] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, Q. Zhang, and H. Hinton. Stackguard: Automatic adaptive detection and prevention of buffer-overflow attacks. In *Usenix Security*, 1998.

[15] A. Cui, J. Kataria, and S. J. Stofo. From prey to hunter: Transforming legacy embedded devices into exploitation sensor grids. In *Proceedings of the 27th Annual Computer Security Applications Conference*, 2011.

[16] A. Cui and S. J. Stolfo. A quantitative analysis of the insecurity of embedded network devices: Results of a wide-area scan. In *Annual Computer Security Applications Conference (ACSAC)*. ACM, 2010.

[17] K. Eldefrawy, A. Francillon, D. Perito, and G. Tsudik. SMART: Secure and minimal architecture for (establishing a dynamic) root of trust. In *Network and Distributed System Security Symposium (NDSS)*, 2012.

[18] K. Eldefrawy, G. Tsudik, A. Francillon, and D. Perito. SMART: Secure and minimal architecture for (establishing a dynamic) root of trust. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2012.

[19] B. Fowler. Some top baby monitors lack basic security features, report finds. 2015. [Online]. http://www.nbcnewyork.com/news/local/Baby-Monitor-Security-Research-324169831.html.

[20] E. Grosse and M. Upadhyay. Authentication at scale. *IEEE Security Privacy*, 11(1), 2013.

[21] G. Hernandez, O. Arias, D. Buentello, and Y. Jin. Smart nest thermostat: A smart spy in your home. In *Black Hat USA*, 2014.

[22] A. G. Illera and J. V. Vidal. Lights off! The darkness of the smart meters. In *BlackHat Europe*, 2014.

[23] M. Kabay. Attacks on power systems: Hackers, malware, 2010.

[24] H. Kagermann, W. Wahlster, and J. Helbig. Securing the future of German manufacturing industry — Recommendations for implementing the strategic initiative Industrie 4.0, 2013.

[25] P. Koeberl, S. Schulz, A.-R. Sadeghi, and V. Varadharajan. TrustLite: A security architecture for tiny embedded devices. In *European Conference on Computer Systems (EuroSys)*. ACM, 2014.

[26] J. Kong, F. Koushanfar, P. K. Pendyala, A.-R. Sadeghi, and C. Wachsmann. PUFatt: Embedded platform attestation based on novel processor-based PUFs. In *Design Automation Conference (DAC)*. ACM, 2014.

[27] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *IEEE Symposium on Security and Privacy (S&P)*, 2010.

[28] F. Koushanfar, A.-R. Sadeghi, and H. Seudie. Eda for secure and dependable cybercars: Challenges and opportunities. In *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012.

[29] X. Kovah, C. Kallenberg, C. Weathers, A. Herzog, M. Albin, and J. Butterworth. New results for timing-based attestation. In *IEEE Symposium on Security and Privacy (S&P)*, 2012.

[30] J. S. Kumar and D. R. Patel. A survey on internet of things: Security and privacy issues. *International Journal of Computer Applications*, 90(11), 2014.

[31] R. Lemos. Sony left passwords, code-signing keys virtually unprotected. *eWeek*, 2014. [Online]. http://www.eweek.com/security/sony-left-passwords-code-signing-keys-virtually-unprotected.html.

[32] J. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V. Gligor, and A. Perrig. TrustVisor: Efficient TCB reduction and attestation. In *IEEE Symposium on Security and Privacy (S&P)*, 2010.

[33] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar. Innovative instructions and software model for isolated execution. In *Hardware and Architectural Support for Security and Privacy (HASP)*. ACM, 2013.

[34] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *Conference on Computer and Communications Security (CCS)*. ACM, 2014.

[35] B. Miller and D. Rowe. A survey SCADA of and critical infrastructure incidents. In *Research in Information Technology (RIIT)*. ACM, 2012.

[36] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Survey internet of things: Vision, applications and research challenges. *Ad Hoc Netw.*, 10(7), 2012.

[37] Nest Labs. Open source compliance. [online]. https://nest.com/legal/compliance.

[38] J. Noorman, P. Agten, W. Daniels, R. Strackx, A. Van Herrewege, C. Huygens, B. Preneel, I. Verbauwhede, and F. Piessens. Sancus: Low-cost trustworthy extensible networked devices with a zero-software trusted computing base. In *USENIX Conference on Security*. USENIX Association, 2013.

[39] E. Owusu, J. Guajardo, J. McCune, J. Newsome, A. Perrig, and A. Vasudevan. OASIS: On achieving a sanctuary for integrity and secrecy on untrusted platforms. In *ACM Conference on Computer & Communications Security (CCS)*. ACM, 2013.

[40] H. Park, D. Seo, H. Lee, and A. Perrig. SMATT: Smart meter attestation using multiple target selection and copy-proof memory. In *Computer Science and its Applications*. Springer, 2012.

[41] B. Parno, J. M. McCune, and A. Perrig. Bootstrapping trust in commodity computers. In *Security and privacy (SP), 2010 IEEE symposium on*, 2010.

[42] K. Poulsen. Slammer worm crashed Ohio nuke plant network, 2003.

[43] PR Newswire. Computer virus strikes CSX transportation computers, 2003.

[44] M. Rostami, A. Juels, and F. Koushanfar. Heart-to-heart (h2h): authentication for implanted medical devices. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013.

[45] M. Rostami, F. Koushanfar, and R. Karri. A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, 2014.

[46] B. Schneier. Cryptographic design vulnerabilities. *Computer*, 31(9):29–33, 1998.

[47] A. Seshadri, M. Luk, A. Perrig, L. van Doorn, and P. Khosla. SCUBA: Secure code update by attestation in sensor networks. In *ACM Workshop on Wireless Security (WiSe)*. ACM, 2006.

[48] A. Seshadri, M. Luk, E. Shi, A. Perrig, L. van Doorn, and P. Khosla. Pioneer: Verifying code integrity and enforcing untampered code execution on legacy systems. In *ACM Symposium on Operating Systems Principles (SOSP)*. ACM, 2005.

[49] D. Shahrjerdi, J. Rajendran, S. Garg, F. Koushanfar, and R. Karri. Shielding and securing integrated circuits with sensors. In *Computer-Aided Design (ICCAD), 2014 IEEE/ACM International Conference on*. IEEE, 2014.

[50] S. Skorobogatov. Fault attacks on secure chips: from glitch to flash. In *Design and Security of Cryptographic Algorithms and Devices (ECRYPT II)*, 2011.

[51] M. Smith. Security holes in the 3 most popular smart home hubs and honeywell tuxedo touch. 2015. [Online]. http://www.networkworld.com/article/2952718/microsoft-subnet/security-holes-in-the-3-most-popular-smart-home-hubs-and-honeywell-tuxedo-touch.html.

[52] R. Strackx, F. Piessens, and B. Preneel. Efficient isolation of trusted subsystems in embedded systems. In *Security and Privacy in Communication Networks*. Springer, 2010.

[53] G. E. Suh, D. Clarke, B. Gassend, M. van Dijk, and S. Devadas. AEGIS: Architecture for tamper-evident and tamper-resistant processing. In *Annual International Conference on Supercomputing (CIS)*. ACM, 2003.

[54] H. Suo, J. Wan, C. Zou, and J. Liu. Security in the internet of things: A review. In *International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 2012.

[55] L. Szekeres, M. Payer, T. Wei, and D. Song. Sok: Eternal war in memory. In *2013 IEEE Symposium on Security and Privacy (SP)*, 2013.

[56] H. T. T. Truong, X. Gao, B. Shresthab, N. Saxena, N. Asokan, and P. Nurmi. Using contextual co-presence to strengthen zero-interaction authentication: Design, integration and usability. *Pervasive and Mobile Computing*, 2014.

[57] Trusted Computing Group (TCG). Website, 2011.

[58] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet x. 509 public key infrastructure (pki) proxy certificate profile. Technical report, 2004.

[59] O. Vermesan and P. Friess. *Internet of Things — From Research and Innovation to Market Deployment*. River Publishers, 2014.

[60] J. Vijayan. Stuxnet renews power grid security concerns, 2010.

[61] J. Winter. Trusted computing building blocks for embedded linux-based ARM Trustzone platforms. In *ACM Workshop on Scalable Trusted Computing (STC)*. ACM, 2008.

[62] J. Wurm, O. Arias, K. Hoang, A.-R. Sadeght, and Y. Jin. Security analysis on consumer and industrial iot devices. In *21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016.

[63] K. Zhao and L. Ge. A survey on the internet of things security. In *Computational Intelligence and Security (CIS)*, 2013.

[64] S. Zonouz, J. Rrushi, and S. McLaughlin. Detecting industrial control malware using automated PLC code analytics. *IEEE Security and Privacy*, 12(6), 2014.

[65] D. Zuehlke. Smartfactory — towards a factory of things. *Annual Reviews in Control*, 34(1), 2010.