Security Challenges in Embedded Systems

DIMITRIOS N. SERPANOS, University of Patras, Greece and Industrial Systems Institute/RC 'Athena', Greece
ARTEMIOS G. VOYIATZIS, Industrial Systems Institute/RC 'Athena', Greece

Embedded systems security is a significant requirement in emerging environments, considering the increasing deployment of embedded systems in several application domains. The large number of deployed embedded systems, their limited resources and their increasing complexity render systems vulnerable to an increasing number of threats. Additionally, the involvement of sensitive, often private, information and the expectation for safe and dependable embedded platforms lead to strong security requirements, even legal ones, which require new technologies for their provision. In this article, we provide an overview of embedded security issues, used methods and technologies, identifying important challenges in this emerging field.

Categories and Subject Descriptors: C.3 [**Computer Systems Organization**]: Special-Purpose and Application-based Systems—*Real-time and embedded systems*

General Terms: Security, Design, Reliability

Additional Key Words and Phrases: Communication system security, computer security, embedded systems, security

ACM Reference Format:
Serpanos, D. N. and Voyiatzis, A. G. 2013. Security challenges in embedded systems. ACM Trans. Embedd. Comput. Syst. 12, 1s, Article 66 (March 2013), 10 pages.
DOI: http://dx.doi.org/10.1145/2435227.2435262

1. INTRODUCTION

Embedded systems constitute the majority of computational and communications base deployed worldwide. Their adoption in a wide range of products and processes in the consumer, enterprise, and industrial domains have led to embedded system presence in almost all technology products, from toys and telephones to cars and industrial process control.

Advances in embedded systems hardware and software technology enable the development of increasingly complex embedded systems. Additionally, the penetration of embedded systems to products and processes increases continuously, since it enables control of complex processes and leads to cost effective solutions at a wide front of uses. As embedded systems are being used and are continuously being employed in emerging applications in everyday life, they are increasingly involved in services that require security. However, security is a term that is being used differently, depending on the

This work is supported in part by a grant (PENED 2003), which is funded by the European Commission by 75%, from the Hellenic State-Greek Ministry of Development-General Secretariat of Research and Technology (GSRT) by 25% and by private industry through the Operational Programme "Competitiveness," Measure 8.3, 2000-2006.

Author's addresses: D. N. Serpanos, Department of Electrical and Computer Engineering, University of Patras, Patras, GR-26504, Greece; email: serpanos@ece.upatras.gr; A. G. Voyiatzis, Industrial Systems Institute/RC Athena, Patras Science Park building, Stadiou Str., Plarani GR-26504, Greece.

DOI: http://dx.doi.org/10.1145/2435227.2435262

user, system, and process involved. For example, security in an e-commerce transaction implies protection of financial data, while in a car it implies safety and dependability. Importantly, all these interpretations of security require development and adoption of new technologies, in order to provide the required properties.

Although security has received attention in the past in the context of conventional computational and communication environments, the dramatic evolution of embedded systems and related applications leads to need for new technologies to be developed, since embedded systems have brought significant differentiations to the models considered in the past. The large numbers of users, the limited resources of embedded systems and the complexity of developed embedded platforms leads to the need to reconsider the models for the capabilities of attackers, the necessary levels of protection of devices and systems, including physical protection, the performance limitations of systems for computationally intensive operations, etc.

The goal of this article is to provide a comprehensive overview of the directions and challenges of embedded systems security. Since security has different interpretations, as mentioned earlier, we specify the security issues addressed in the article and we distinguish security from privacy, safety and dependability. As these properties are important and considered as parts of security in many contexts, we also address their relationship and dependence on security.

The article is organized as follows. Section 2 presents the security requirements of applications and services on embedded systems, the differentiators of embedded systems from non-embedded ones and distinguishes security from privacy, safety and dependability. Section 3 describes the security issues and challenges in autonomous embedded systems, while Section 4 addresses issues and challenges in networked embedded systems and their applications and services. Finally, Section 5 addresses privacy protection, safety and dependability, presenting challenges in these technical areas, which demonstrate the need to differentiate them from security, as they require advances in different direction.

2. SECURITY AND EMBEDDED SYSTEMS

Security is considered a significant requirement in emerging embedded systems, especially considering the connectivity that most of them provide to system networks, private networks, or the Internet. However, the term security is being used indicating different properties or services, depending on the context of system use and considering the attacks that have been and are being developed for systems overall.

In order to specify the requirements placed on systems for security, one needs to identify the areas of applications and services where embedded systems are used, since applications designate the security requirements, which, in turn, need to be met through appropriate security mechanisms and policies. A useful classification of the driving applications for embedded systems appears in the ARTEMIS Strategic Research Agenda [ARTEMIS 2006], where applications and services are classified in 4 major categories: (i) industrial systems, including automotive and aerospace transport, (ii) nomadic environments, targeting on mobile communications and information services, (iii) private spaces, focusing on home services, and (iv) public infrastructure, aiming to secure and dependable infrastructures. Although this classification is not unique for embedded systems applications and services, it is definitely a comprehensive one, indicating the areas of industrial interest driving the development of embedded systems technology.

Analyzing the application/service areas and considering their requirements for security from all perspectives, vendors, providers and customers, it becomes clear that the range of applications places several security requirements.

(1) *Confidentiality*: protection of stored or transmitted data from disclosure
(2) *Integrity*: verification of the correctness of stored or transmitted data
(3) *Authentication*: identification of the producer or sender of data
(4) *Access control*: availability of services only to authorized users
(5) *Non-repudiation*: inability of transaction participants to deny actions
(6) *Dependability*: provision of functionality in a dependable fashion, i.e. continuous service, expected/promised quality of service (such as real-time requirements), etc.
(7) *Safety*: provision of services without possibility of hazard to the users
(8) *Privacy*: protection of personal information and resources from unauthorized parties

In order to provide security in systems, one does not need to consider only the functional requirements of applications and services, but other factors as well, such as the profile of attackers and the resources that are available for the provision of secure applications. The last two factors, attacker profile and available resources, are the main differentiators of security provision in embedded systems and services relatively to non-embedded ones. Specifically, the main differentiators that need to be considered are:

(1) deployment in significantly larger numbers than non-embedded systems (servers, desktops, etc.);
(2) simpler and limited resources, including power, computational and communications;
(3) low cost requirements, considering their adoption in consumer devices of various levels;
(4) deployment in hostile environments of various types and ability of malicious users to capture systems;
(5) strict safety requirements, especially in some application domains, such as industrial, automotive and aeronautics.

These differentiators of embedded systems actually make security requirements on embedded systems significantly stricter than in traditional computational systems. Specifically, embedded systems are deployed in more hostile environments, in much larger numbers than traditional computational systems, and they are available to a wider population, which includes larger numbers of potential attackers; thus, one needs to provide hard—and thus demanding—security mechanisms in order to protect against a significantly larger number of potential attacks. Furthermore, the safety requirements in a wide range of applications, such as transport systems, surveillance, etc., as well as dependability and protection of privacy, lead to additional requirements, considering the criticality of safety and continuous operation in several application domains of embedded systems. Importantly, all these stricter requirements have to be provided in systems with limited resources and at low cost, in order to meet consumer and client expectations.

Safety, privacy and dependability are considered security requirements in many application domains. However, they differ from the typical security considerations in several ways. Privacy protection and safety, for example, differ in that they are usually requirements for processes, applications and services, rather than system-level requirements, which are met with the typical security mechanisms (encryption, authentication, etc.). Dependability is typically considered as an application and/or system requirement, providing reliability, availability, etc., in the context of accidental and nonmalicious failures, in contrast to security that is associated to malicious interventions. Importantly, in our view, privacy and safety are dependent on security, since they need to employ security mechanisms for their implementation; furthermore, safety and privacy protection can be considered to overlap, since privacy protection is considered a

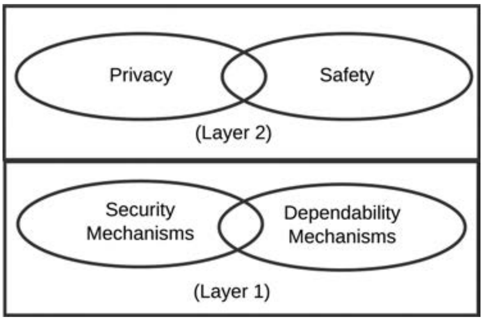

Fig. 1. Security property layers.

safety issue in several contexts, as in the case of financial transactions. In contrast, dependability is mainly a system issue, which is complementary to security, as indicated in Figure 1. The layering shown in the figure indicates that security and dependability are a requirement for privacy and safety. One can easily derive this, considering that, if security mechanisms are not provided, an attacker can (i) easily collect any data, attacking privacy, and (ii) alter processes, leading to unsafe conditions. Dependability is complementary, since an attacker can insert faults and failures that the dependability mechanisms cannot recover from, in analogy to attacks on security mechanisms.

3. AUTONOMOUS EMBEDDED SYSTEM SECURITY

Embedded computing systems employ structures similar to the ones for general-purpose computing systems. The typical structure of an embedded computing system is shown in Figure 2, where the system contains 4 main subsystems: (i) processing, (ii) memory, (iii) communication, and (iv) power. In a secure system, all components need to be protected individually as well as the complete system, depending on the specific requirements and the operational environment. For example, in a sensor system, where sensors are distributed in a potentially hostile environment, protection of the complete system may be infeasible, and thus, subsystems have to be protected individually and the complete system needs to be adaptive to operating with a dynamic sensor population.

There are several levels of protection used for security of stand-alone embedded systems, ranging from physical and hardware security to trusted computing platforms.

Physical security of complete systems or subsystems is addressed through antitampering techniques, and specifically techniques for tamperevidence, tamperresponse and tamper-resistance. Tamper-evidence techniques enable the identification of tampering of devices, while tamper-response techniques include tamper-detection and reaction with actions that protect sensitive information appropriately. Tamperresistance techniques, finally, prevent tampering attacks and protect sensitive information from non-intrusive attacks that have been developed. Anti-tamper technologies are especially important in sensor systems, which are often deployed for critical applications, like surveillance, in highly hostile environments.

Anti-tamper technologies are increasingly important to deployed embedded systems, because system protection requires physical as well as algorithmic mechanisms. For example, encryption had been long considered a sufficient solution for many applications and services, but the advances of attacks for more than a decade have demonstrated that it is not sufficient, especially in low-cost embedded systems with limited resources, where encryption mechanisms can be broken in simple ways. Special attention has to be paid to side-channel attacks [Zhou and Feng 2005], which have changed

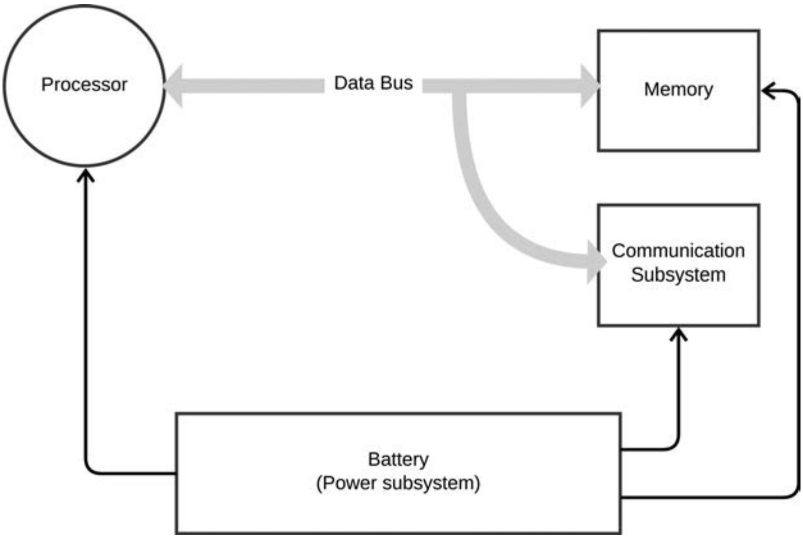

Fig. 2. Structure of a typical embedded computing system.

the model of attacks, breaking cryptosystems using methods that rather than attacking the cryptographic algorithms themselves exploit physical characteristics (timing, power, etc.) [Kocher 1996; Kocher et al. 1999; Quisquater and Samyde 2001] or introduce faults during the cryptographic computations [Bar-El et al. 2006; Joye 2009].

Anti-tampering techniques protect against attacks after system deployment. New business environments can drive embedded systems insecure by planting *hardware* Trojans during the design and manufacture phase [Jin and Makris 2010].

Importantly, physical and hardware attacks have not been proven successful only against specialized circuits, but against more complex systems as well, such as microcontrollers and processors [Anderson and Kuhn 1996; Blythe et al. 1993]. Several hardware and architectural techniques have been developed for protection against physical and hardware attacks. Dedicated hardware is necessary, for example, to protect sensitive parts of the system memory. One such solution is a hardware implementation of a type of execute-only memory, which allows instructions stored in memory to be executed but not manipulated otherwise [Lie et al. 2000]. Side channel attacks can be prevented by specialized design techniques in ASICs, or using architectural concepts, such as decay caches [Keramidas et al. 2008], for protection from cache information leakage, and bus encryption [Best 1981; Kuhn 1997] to protect data exchanged between the processor and the main memory.

The diversity of the attacks [Ravi et al. 2004] against embedded systems creates the need for software as well as hardware protection technologies, in order to address security problems in embedded systems. Hardware modifications are necessary and sufficient to defend against some of those attacks, as mentioned above. However, for more complex attacks on highly programmable systems, operating systems (OS) enhancements or other software techniques must be used; importantly, software solutions to attacks are cost effective, considering the high cost of several hardware solutions. The availability of such technologies enables the development of trusted computing platforms for applications and services.

Security in more complex, highly-programmable systems requires development of methodologies for a wide range of issues, from methodologies for secure bootstrapping, in order to verify system integrity in every step of the booting procedure [Arbaugh et al. 1997] to the use of process isolation and process level attestation techniques [Microsoft 2011] to protect running processes. Additional OS enhancements include techniques for context switching, exception handling, inter-process communication and memory management [Lie et al. 2003; Garfinkel et al. 2003]. Importantly, software authentication and validation is necessary and feasible, since it can be achieved by using techniques like oblivious hashing [Chen et al. 2003] or program shepherding [Kiriansky et al. 2002].

4. NETWORKED EMBEDDED SYSTEMS AND APPLICATION SECURITY

Secure communication is based on the use of encryption and authorization mechanisms together with a secure routing method across a network. The decision of the encryption scheme to be used is critical since the encryption complexity defines the level of security offered by the communication. As increasing computational resources become available to embedded systems, traditional public key cryptography is becoming a viable option for some applications, although it is still too demanding computationally for most embedded applications and services. Elliptic curve cryptography provides a promising solution to embedded systems, due to its lower computational resources than algebraic public key cryptography, while providing a high level of security [Miller 1986].

Key management is an important process in the establishment of secure communication as keys are the base of the encryption and decryption mechanisms. Often, key management constitutes the weak point of a security system, since disclosure or leakage of keys renders even the strongest cryptosystem ineffective. Global communication keys usually cannot be pre-defined in the networked systems, since the security of the network is easily compromised. Thus, it is necessary to establish an effective mechanism of generating and distributing keys. Two effective ways of achieving this is the use of temporary global keys and random key distribution. With temporary global keys, a global permanent key is used to establish a main key, and then, the global key is destroyed in order to avoid key leakage, the main risk with using global keys [Perrig et al. 2004]. In random key distribution, a large number of keys exist and communication is accomplished through the use of choosing random subsets of keys. With appropriate set size choices, communication between all ends of a network can be accomplished [Chan et al. 2003].

Networked embedded systems constitute distributed systems, since they coordinate to provide applications and services. As such, they need to defend against known distributed system attacks at the application layer in addition to the communication layers they implement. Such attacks include security flaws in the management of application complexity, distributed denial-of-service, secure upgrading, etc.

Upgrading systems in a networked embedded environment is a necessity for fixing software issues as well as for implementing new features and services. The increasing software complexity of systems and services leads to an increased number of software bugs and therefore, systems need to be updated during their operation. Furthermore, new features and new services can be implemented by software upgrade on existing hardware. However, software upgrading leads to security risks, because it offers possibility for malicious software to replace legitimate software. To face these security risks, it is important to implement defences at both the communication protocols and the internal organization of the system. The transmission of mobile code, i.e., system software transmitted over a communication link, must be done with higher security requirements than normal communication. Moreover, upgradeability can be limited, or not allowed at all, for the software components that access crucial system resources.

Remote management of networked systems is another example of a service which provides functional advantages but poses security risks, due to computation and transmission of critical information. Remote management implementations need to be built in combination with an effective intrusion detection system to prevent attacks. Remote management communication requires implementation with high encryption requirements as well as limitation of the access rights of the remote users.

All networked embedded systems, especially those connected to the Internet, need to control the communicated data, allowing only appropriate (e.g., authorized) users and processes to send/receive “legal” data. Firewalls can usually be implemented at the network and application layers, either in end systems or in the network infrastructure [Bolding 1995]. Since networked embedded systems often have very specific and well-defined communication needs, firewalls can be strictly configured to allow the limited type of legitimate communication. The point where the firewall can be implemented, at the network or application layer and in the node or in the network, depends on the nature of the networked systems, its available resources and the network topology. For instance, ad-hoc networks need to have a level of protection at the node level, while more centralized systems can more successfully implement network layer and network level protection [Slijepcevic et al. 2002].

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks are a prevalent threat against networked embedded systems. Such attacks prevent the system from either working or serving its intended users by overloading one or more resources of the networked system. The targeted resources, usually, are CPU, memory, and network bandwidth. There are two basic types of DoS attacks [Hussain et al. 2003]. The first type targets specific software and/or hardware vulnerabilities that can be exploited by sending one or more carefully crafted packets in order to crash the networked system. The attacks of the second type take the form of massive amounts of network traffic originating from thousands of compromised systems that mix with the legitimate traffic and prevent the victim system from serving its legitimate users. Such attacks are called DDoS attacks.

The first type of attack is yet another manifestation of the ongoing race between system vulnerabilities and software patches. The second type is the one that cannot be easily stopped, mainly due to the fact that shared network services can usually be accessed by all systems that are connected to the same network. The current architecture of IPv4 makes things worse by allowing IP packets to be sent with arbitrary values on their source IP address field [Wang et al. 2007]. Systems and mechanisms that deal with DDoS attacks combine intrusion detection and traceback schemes for early identification, filtering and tracing of a DDoS attack [Peng et al. 2007]. Such mechanisms include ingress/egress filtering for attack prevention [Ferguson and Senie 1998], signature and anomaly-based detection [Cabrera et al. 2001; Wang et al. 2002], packet marking [Belenky and Ansari 2003; Savage et al. 2001], and packet logging [Snoeren et al. 2002] for attack traceback.

Finally, an important category of networked embedded systems that requires special attention is that of sensor networks. Sensor networks differ from other embedded system networks in that they usually form ad-hoc networks of a large number of nodes, often with very limited computational resources. As a consequence, protocols employed for communication between sensor nodes must meet stricter performance requirements. The limitation lead sensor networks to implement encryption mechanisms at the link layer, typically. A good encryption strategy for sensor networks is to use encryption mechanisms with different complexity, depending on the value of information exchanged [Zhu et al. 2003].

5. PRIVACY, SAFETY AND DEPENDABILITY

Privacy protection, safety and dependability are system properties often associated with security, as mentioned in Section 2. Their importance is growing significantly in modern environments, due to the high penetration of embedded systems in consumer, enterprise, and industrial environments. Since their interdependence with security is strong, as shown in Figure 1, we need to address them, in order to demonstrate their relationship with security as well as the important technologies they require for the development of reliable and secure platforms for all emerging services.

5.1. Privacy Protection

Privacy protection is a significant requirement in embedded systems, considering several of the applications areas such as mobile phones, home networks, health systems, e/m-banking systems, etc. The requirement is to protect sensitive personal information involved in applications, according to legal directives. Specifically, privacy protection requires the conditional collection and storage of personal information. Importantly, privacy requires often the time-limited storage of relative information as well as its access only by appropriately authorized personnel, when necessary. Such requirements constitute a significant burden on systems, which do not have to provide simple confidentiality mechanisms (encryption/decryption), but complex access control mechanisms and periodic auditing of systems as well, in order to ensure compliance with the required policies and laws. Furthermore, the increasing amount of information considered as personal or private leads to a need for development of adaptive and scalable solutions that can accommodate new requirements as they emerge [Müller 2006].

5.2. Safety

Safety is a generic term used to describe requirements for information and process protection against failures that lead to significant risk to life or stability. The generality of the term originates from the fact that safety is a property of the service (or process) that is implemented using a computational and communications platform (embedded in this case). Although the concept of safety is relatively simple and clear in some cases, e.g., automobile and avionics systems [eSafety Forum 2011; Tadlock 2002], it can be extended to a wide range of process properties. For example, privacy protection in several environments can be considered a safety property as well for the persons involved (as in the case of an electronic transaction, for example, where leakage of private information can lead to risks of digital fraud and to significant loss of funds, affecting stability in one's life).

There are two main problems posed by safety: (i) the development of safe systems based on limited-resource subsystems, such as embedded systems, and (ii) the verification that the developed system meets the safety requirements placed. Clearly, the first problem includes addressing several of the security issues addressed above. Importantly, verification is a critical issue here, especially because it is a computationally hard problem. Although there have been technologies developed for system development, which lead to computationally feasible system verification, such as straight-line code adoption, etc., there is still significant requirement for new techniques which will enable safety property verification, especially considering the increasing complexity and capabilities of deployed embedded systems.

5.3. Dependability

Dependability of computational and communication systems has been a desirable property for many application domains. As a result, development of dependable systems has experienced significant advances and has been considered as a mature field for a long time, using well established and deeply analyzed techniques [Siewiorek and Swarz 1982]. However, most of the methods developed up to date are based on the assumption of accidental faults and errors, using appropriate fault models. In contrast, conventional systems, and especially embedded systems, can be attacked by malicious attackers inserting faults and causing errors and failures that differ significantly from the models used up to date. Furthermore, the increasing complexity of highly integrated embedded systems and networks of embedded systems render many of the existing techniques inappropriate or with limited effectiveness in the emerging environments. Thus, it is a clear challenge to develop new methods and technologies that combine security and dependability appropriately, in order to develop the desired embedded platforms [Serpanos and Henkel 2008].

6. CONCLUSION

Embedded systems security is an emerging field in embedded systems technology, relevant to all application domains of these systems. In this article, we addressed embedded systems security and identified its relationship to privacy protection, safety and dependability, which are often considered security properties. We presented security requirements based on the application domains and we described technologies and methods for autonomous and networked embedded systems. Finally, we presented the challenges placed by privacy, safety and dependability requirements, which lead to complementary technical problems than the conventional security ones.

REFERENCES

Anderson, R. and Kuhn, M. 1996. Tamper resistance: A cautionary note. In *Proceedings of the 2nd Workshop on Electronic Commerce*. USENIX Association, Berkeley, CA, 1–11.

Arbaugh, W., Farber, D., and Smith, J. 1997. A secure and reliable bootstrap architecture. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, Los Alamitos, CA, 65–71.

ARTEMIS 2006. ARTEMIS strategic research agenda 1st Ed. http://www.artemis-office.org.

Bar-El, H., Choukri, H., Naccache, D., Tunstall, M., and Whelan, C. 2006. The sorcerer's apprentice guide to fault attacks. *Proce. IEEE 94,* 2, 370–382.

Belenky, A. and Ansari, N. 2003. IP traceback with deterministic packet marking. *IEEE Comm. Letters, 7,* 4, 162–164.

Best, R. 1981. Crypto microprocessor for executing enciphered programs.

Blythe, S., Fraboni, B., Lall, S., Ahmed, H., and de Riu, U. 1993. Layout reconstruction of complex silicon chips. *IEEE J. Solid-State Circuits, 28,* 2, 138–145.

Bolding, D. 1995. Network security, filters and firewalls. *Crossroads 2,* 1, 8–10.

Cabrera, J., Lewis, L., Qin, X., Lee, W., Prasanth, R., Ravichandran, B., and Mehra, R. 2001. Proactive detection of distributed denial of service attacks using MIB traffic variables—a feasibility study. In *Proceedings of the IEEE/IFIP International Symposium on Integrated Network Management*. IEEE, 609–622.

Chan, H., Perrig, A., and Song, D. 2003. Random key predistribution schemes for sensor networks. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, Los Alamitos, CA, 197–213.

Chen, Y., Venkatesan, R., Cary, M., Pang, R., Sinha, S., and Jakubowski, M. 2003. Oblivious hashing: A stealthy software integrity verification primitive. In *Proceedings of the 5th International Workshop on Information Hiding* (Revised Papers). Springer-Verlag, Berlin, 400–414.

eSafety Forum 2011. http://ec.europa.eu/information_society/activities/esafety/forum/index_en.htm.

Ferguson, P. and Senie, D. 1998. RFC 2267: Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing.

Garfinkel, T., Rosenblum, M., and Boneh, D. 2003. Flexible OS support and applications for trusted computing. In *Proceedings of the 9th Conference on Hot Topics in Operating Systems*. Vol. 9, USENIX Association, Berkeley, CA, USA, 25–25.

Hussain, A., Heidemann, J., and Papadopoulos, C. 2003. A framework for classifying denial of service attacks. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM, New York, NY, 99–110.

Jin, Y. and Makris, Y. 2010. Hardware trojans in wireless cryptographic ics, *IEEE Test Computers, 27,* 1, 26–35.

Joye, M. 2009. Protecting RSA against fault attacks: The embedding method. In *Proceedings of the Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE Computer Society, Los Alamitos, CA, 41–45.

KERAMIDAS, G., ANTONOPOULOS, A., SERPANOS, D., AND KAXIRAS, S. 2008. Non deterministic caches: A simple and effective defense against side channel attacks. *Design Autom. Embed. Syst. 12,* 3, 221–230.

KIRIANSKY, V., BRUENING, D., AND AMARASINGHE, S. 2002. Secure execution via program shepherding. In *Proceedings of the 11th USENIX Security Symposium*. USENIX Association, Berkeley, CA, 191–206.

KOCHER, P. 1996. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology - CRYPTO'96*. Springer-Verlag, Berlin, 104–113.

KOCHER, P., JAFFE, J., AND JUN, B. 1999. Differential power analysis. In *Advances in Cryptology-CRYPTO'99*. Springer-Verlag, Berlin, 789–789.

KUHN, M. 1997. The TrustNo1 cryptoprocessor concept. http://www.cl.cam.ac.uk/ mgk25/.

LIE, D., THEKKATH, C., AND HOROWITZ, M. 2003. Implementing an untrusted operating system on trusted hardware. *ACM SIGOPS Operat. Syst. Revi. 37,* 5, 178–192.

LIE, D., THEKKATH, C., MITCHELL, M., LINCOLN, P., BONEH, D., MITCHELL, J., AND HOROWITZ, M. 2000. Architectural support for copy and tamper resistant software. *ACM SIGPLAN Not. 35,* 11, 168–177.

MICROSOFT. 2011. Shared source initiative. http://www.microsoft.com/resources/ngscb/default.mspx.

MILLER, V. 1986. Use of elliptic curves in cryptography. In *Advances in Cryptology—CRYPTO85*. Lecture Notes in Computer Sciences, vol. 218, Springer-Verlag, Berlin, 417–426.

MÜLLER, G. 2006. Special issue: Privacy and security in highly dynamic systems-introduction. *Comm. ACM 49,* 9, 28–31.

PENG, T., LECKIE, C., AND RAMAMOHANARAO, K. 2007. Survey of network-based defense mechanisms countering the dos and ddos problems. *ACM Comput. Surv. 39,* 1, 3–es.

PERRIG, A., STANKOVIC, J., AND WAGNER, D. 2004. Security in wireless sensor networks. *Comm. ACM 47,* 6, 53–57.

QUISQUATER, J. AND SAMYDE, D. 2001. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *Proceedings of the International Conference on Research in Smart Cards: Smart Card Programming and Security*. Springer-Verlag, Berlin, 200–210.

RAVI, S., RAGHUNATHAN, A., KOCHER, P., AND HATTANGADY, S. 2004. Security in embedded systems: Design challenges. *ACM Trans. Embed. Comput. Syst. 3,* 3, 461–491.

SAVAGE, S., WETHERALL, D., KARLIN, A., AND ANDERSON, T. 2001. Network support for IP traceback. *IEEE/ACM Trans. Network. 9,* 3, 226–237.

SERPANOS, D. AND HENKEL, J. 2008. Dependability and security will change embedded computing. *Computer 41,* 1, 103–105.

SIEWIOREK, D. AND SWARZ, R. 1982. *The Theory and Practice of Reliable System Design*. Digital Press, Bedford, MA.

SLIJEPCEVIC, S., POTKONJAK, M., TSIATSIS, V., ZIMBECK, S., AND SRIVASTAVA, M. 2002. On communication security in wireless ad-hoc sensor networks. In *Proceedings of the 11th IEEE International Workshop on Enabling Technologies*. IEEE Computer Society, Los Alamitos, CA, 139–144.

SNOEREN, A., PARTRIDGE, C., SANCHEZ, L., JONES, C., TCHAKOUNTIO, F., SCHWARTZ, B., KENT, S., AND STRAYER, W. 2002. Single-packet IP traceback. *IEEE/ACM Trans. Network. 10,* 6, 721–734.

TADLOCK, D. E. 2002. Avionics Safety. In *Proceedings of the Joint ESA-NASA Space-Flight Safety Conference*. B. Battrick and C. Preyssi, Eds., Vol. ESA SP-486, European Space Agency, Noordwijk, The Netherlands, 75–80.

WANG, H., JIN, C., AND SHIN, K. 2007. Defense against spoofed IP traffic using hop-count filtering. *IEEE/ACM Trans. Network. 15,* 1, 40–53.

WANG, H., ZHANG, D., AND SHIN, K. 2002. Detecting SYN flooding attacks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*. Vol. 3. IEEE, Los Alamitos, CA, 1530–1539.

ZHOU, Y. AND FENG, D. 2005. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. http://eprint.iacr.org/2005/388.

ZHU, S., SETIA, S., AND JAJODIA, S. 2003. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*. ACM, New York, NY, 62–72.

Received June 2011; accepted November 2011