# Embedded Systems Security, Aspects of secure embedded systems design and implementation

**Conference Paper** · September 2011

CITATIONS
0

READS
190

**3 authors:**

Konstantinos Fysarakis
Sphynx Technology Solutions AG
**52** PUBLICATIONS **373** CITATIONS
SEE PROFILE

Harry Manifavas
Qatar University
**57** PUBLICATIONS **891** CITATIONS
SEE PROFILE

Konstantinos Rantos
Eastern Macedonia and Thrace Institute of Technology
**39** PUBLICATIONS **257** CITATIONS
SEE PROFILE

# Embedded Systems Security

## Aspects of secure embedded systems design and implementation

Konstantinos Fysarakis, Charalampos Manifavas
Dept. of Applied Informatics & Multimedia
Technological Educational Institute of Crete
Estavromenos, Heraklion, Crete 71500, Greece
fisarakis@epp.teicrete.gr, harryman@epp.teicrete.gr

Konstantinos Rantos
Dept. of Industrial Informatics
Technological Educational Institute of Kavala,
Kavala, GR-65404, Greece
krantos@teikav.edu.gr

***Abstract*—Embedded Systems account for a wide range of products and are employed in various heterogeneous domains, including but not limited to: industrial systems (e.g. manufacturing plants), critical environments (e.g. military and avionics) nomadic environments (e.g. personal wearable nodes), private spaces (e.g. residences) and public infrastructures (e.g. airports). These devices often need to access, store, manipulate and/or communicate sensitive or even critical information, making the security of their resources and services an imperative concern in their design. The problem is exacerbated by their resource constraints, their diversified application settings, frequently requiring unattended operation in physically insecure environments and dynamic network formulation, in conjunction with the ever-present need for smaller size and lower production costs. This paper provides an overview of the challenges in Embedded Systems security, pertaining to node hardware and software as well as relevant network protocols and cryptographic algorithms, presents recent advances in the field and identifies opportunities for future research.**

***Keywords-embedded devices; security; denial of service; lightweight cryptography; location privacy; secure routing;***

## I. INTRODUCTION

Embedded computers systems permeate our lives in various forms, from avionics to e-textiles, automobiles, home automation and wireless sensor nodes. Physically, Embedded Systems (ESs) range from miniature wearable nodes to large industrial installations of Programmable Logic Controllers (PLCs).

The security (i.e. confidentiality, integrity and availability) of networked computer systems is not a novel concern but, in the context of ESs, their various intrinsic and often application specific characteristics render security techniques developed for personal and enterprise systems unsatisfactory or even inapplicable. Such characteristics habitually include resource constraints (namely computational capabilities, memory and power), dynamically formulated, remotely managed networking and even operation in hostile environment and time-critical applications.

An additional differentiating factor of ES security is that applications often include direct interaction with the physical world. Consequently, a security incident might lead to asset damage or even personal injury and death. In [1] researchers demonstrated that it is feasible to manipulate all critical sub-systems in modern automobiles using a wireless-enabled MP3 player connected to the vehicle's embedded control network. The attacks presented include accessing the brake controller, thus disabling or forcibly activating the brakes and consequently compromising the safety of the driver and passengers, as well as injecting malicious code to erase any evidence of tampering after a crash.

Furthermore, since ESs are often responsible for vital, time-critical applications where a delay or a speed-up of even a fraction of a second could have dire consequences. A recent and widely publicized example of such a case is the worm Stuxnet, a highly specialized malware which was designed to target the specific Siemens supervisory control and data acquisition (SCADA) systems installed in Iran's uranium enrichment infrastructure. The purpose of the worm was to take control of the PLCs causing periodic variations in the uranium enrichment centrifuges' rotor speed, thus destroying the devices. Indeed, because of Stuxnet, Iranian scientists were forced to replace approximately 1000 centrifuges over a few months when, prior to the attack, normal failure rates where in the region of 800 per year [2].

The abovementioned differentiating factors in embedded computing security must be taken under consideration during ESs design and implementation. Figure 1 ([3]) shows a

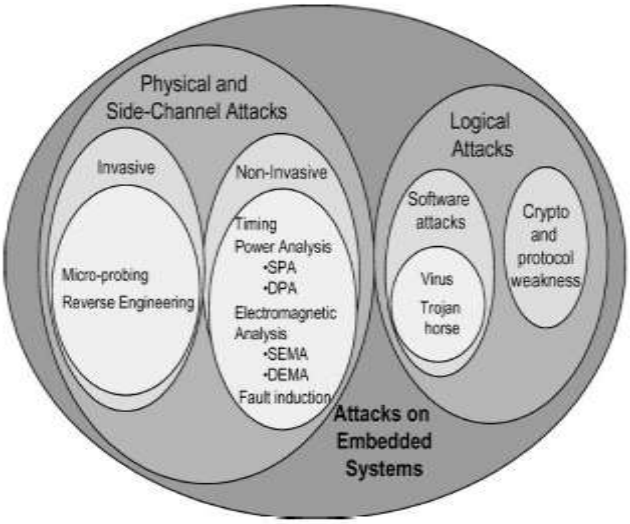


Figure 1. Overview of attacks on ESs.

graphical overview of the main types of attacks on ESs.

## II. Physical Security Issues

### A. Physical Attacks

With regard to the lower layers and given the often unattended nature of deployed ESs, the risk of device tampering should not be ignored. A malicious entity's physical access to a device would enable the launch of various attacks like micro-probing and reverse-engineering or sophisticated Side-Channel Attacks (SCA), like timing attacks, simple power analysis (SPA), differential power analysis (DPA), as well as their electro-magnetic counterparts SEMA and DEMA and differential fault attacks (DFA, [4]). The aforementioned methods can potentially expose critical information concerning the operation of the device (algorithms used, length of keys etc.) which could prove critical to the security both of the device itself and the network as a whole.

### B. Power Supply Protection

Many embedded systems have inherent energy constraints and are often battery powered. Some might get a daily battery charge but others may be expected to last months on a single charge. An attacker who fails to otherwise compromise the system could decide to instead launch a DoS attack by draining the battery power (e.g. by forcing the device to use its wireless connection or work at full CPU load). Therefore, the power source of an ES should satisfy three key requirements:

- Provide continuous power, without any unpredicted fluctuations of its output voltage or current levels, ensuring optimal operation of the powered device.
- Monitor its own state and prevent any power supply issues that might affect the system's operation.
- Feature fail-safe mechanisms to protect and prevent any further damage to the device in case of failure.

Most of the above are common in modern high-end Uninterruptible Power Supply (UPS) systems but are a challenge in some applications of ESs where even a backup battery is considered a luxury as it significantly increases size and cost. Research focuses on several technologies that could be considered like energy scavenging, super-capacitors, micro-solar cells and remote or even wireless power transferring schemes [5][6]. Still, these must be adapted to each specific scenario (e.g. there is no point in installing a vibration generator on a static device) and also consider fail-safe options like being able to power off non-critical systems, disconnect damaged sub-systems etc. in order to allow critical sub-systems of the device to operate or at least protect the device from permanent failure.

## III. Access Control

Access Control mechanisms are essential to prevent unauthorized/malicious entities to access the resources, physical or otherwise, available to the ESs as well as the hosting devices. The way Access Control is implemented varies depending on the hardware capabilities of the nodes, the type of network and, in general, the application considered. Some often-used methods include:

a) Profile Authentication: If a node has some specific characteristics (e.g. hardware specifications, O/S), it can join the network.

b) Access Code: Demonstrating knowledge of the code grants access to the network and its resources. This code can either be programmable or configurable. This category includes typical password access, based on memory data, switch configuration or any other procedure.

c) Predefined Topology: Only pre-established nodes can join the network (e.g. MAC filtering).

There is ongoing research on ES-specific Access Control protocols since the commonly used authentication schemes, typically password-based, can be impractical or even insecure when considering the heterogeneous nature ES networks can demonstrate and the scalable remote manageability often required [7]. Moreover, even in wired embedded networks and in industries like automotive and aviation, most control networks utilized (e.g. Controller Area Network, Time-Triggered Protocol, FlexRay) are designed with safety and reliability in mind and do not feature any built-in security mechanisms like node authentication, data encryption or prevention of Denial of Service (DoS) attacks [8], leading to critical vulnerabilities like the ones already mentioned ([1]).

### A. Denial Of Service

DoS and Distributed Denial of Service (DDoS) attacks aim to compromise the availability of a node or network of nodes, thus preventing authorized entities to access system resources or delaying system operations and functions [10]. In the case of ESs this can be realized by exploiting vulnerabilities on the nodes' software or firmware or by flooding the nodes with network traffic, thus consuming the nodes' CPU cycles, memory, network bandwidth and/or power. Large scale networks of embedded devices are often heterogeneous in nature, comprising of systems of various types and capabilities, like nano, micro and power nodes. Moreover these networks often need to be deployed in diversified environments with provisions for dynamic formulation. All of the above, make dealing with DoS and DDoS attacks a very important security concern, one that is particularly challenging to address [11]. The problem is only exacerbated in limited resources environments where we have to consider the constrained protection capabilities in conjunction with other inherited characteristics (e.g. limited power, unattended operation and management).

It is important to note that DoS attacks are not limited to the Node level itself in view of the fact that they can have many forms (e.g. jamming, resource exhaustion, misrouting, flooding and more) and can essentially be mount on all different layers [12].

On the physical layer, one should consider the case of a malicious individual gaining physical access to a device. This could mean permanent physical destruction of a node or a number of nodes, compromising the availability of the specific

node, cluster of nodes or even the system itself, in the case of destroyed control nodes.

Furthermore, the unattended nature of many ESs necessitates the implementation of remote management features which, as is most often the case, can be exploited to launch remote attacks. This is a major security concern since, in some cases, the outcome of a successful attack can be the permanent destruction of a node (PDoS or Bricking), thus requiring out-of-band hardware re-initialization or the installation of a new node in order to restore service. A typical example is the remote firmware upgrade which most network attached embedded devices support nowadays. Exploitation of this support/remote management feature is usually trivial since the mechanisms are turned "ON" by default, firmware binaries are freely available on the Internet and the protection mechanisms are typically elementary because the process is designed with error detection, not malicious attacks, in mind. This misuse of firmware update mechanisms to corrupt flash memory in a way that renders the device unbootable and non-reflashable is also referred to as Phlashing [13].

Other types of DoS attacks that should be mentioned are those that can be launched on higher layers and mainly consist of targeting and exhausting valuable limited resources (e.g. memory) and/or exhausting the power source of a device by unauthorized use of resources (e.g. wireless connection), which is particularly critical in the case of nano and micro/personal nodes. Furthermore, research and past experience indicate that poor design decisions in network protocols and operating systems can become a serious obstacle in designing DoS and DDoS resilient systems and services. The IP protocol, for example, is vulnerable to such attacks as a result of early assumptions concerning trust of network nodes and, additionally, basic software design methodologies don't take into account security requirements that would facilitate the deployment of DDoS resilient services [14].

The majority of the aforementioned attacks can be avoided with the implementation of sound authentication and access control mechanisms, ensuring availability, provision of undisrupted services and fair resources allocation to all legitimate participating entities. It is also critical to identify the design steps that will enable secure node firmware deployment and software updates as well as network protocols resilient to (D)DoS attacks, taking into account the flaws of existing protocols and facilitating the quantification of DoS resilience [15]. Moreover, an intrinsically secure ES firmware featuring fail safe mechanisms, even hardware redundancy if cost allows, is essential, especially in markets where high dependability is a prerequisite, like avionics and military. In the case of ESs utilizing Field-programmable gate arrays (FPGAs), the concept of runtime reconfiguration [16] can be explored to reduce component count and/or power consumption, increase fault tolerance etc. as needed. Self-reconfigurability can, for example, make a node more secure against side-channel attacks through measurement of EM radiation and also implement self-healing properties. Self-recovery mechanisms could reallocate functional blocks to mark and replace faulty resources, through device reprogramming in the case of self-reconfigurable nodes or through controlled degradation of service techniques in less "intelligent" devices.

## IV. CRYPTOGRAPHIC MECHANISMS

### *A. Lightweight Cryptography*

As already mentioned, embedded devices often have inherent limitations in terms of processing power, memory, storage and energy.

A Trusted Platform Module (TPM [17]), is an example of a component that can be integrated on ESs to provide tamper resistant hardware and software security functions. The software embedded on such a cryptographic component has direct impact on its:

- Size: Memory elements constitute a significant part of the module's surface.
- Costs: Directly linked to the surface of the component.
- Speed: Optimized code provides results faster.
- Power Consumption: The quicker a set of instructions is executed, the quicker the module can return to an idle state or be put in sleep mode where power consumption is minimal.

Lightweight Cryptography refers to algorithmic designs and implementations best suited for deployment in such devices (e.g. RFIDs, sensor nodes, contactless smartcards, mobile devices). There has already been significant effort on the subject of crypto optimization, aiming to maintain the level of security "traditional" algorithms and implementations offer while narrowing what is often referred to as "battery gap" [18], i.e. the very high energy consumption overheads of supporting security on battery constrained systems. A number of surveys ([19][20]) provide an overview of this subject.

Two symmetric ciphers developed for minimal resource requirements are DESL [21] and Present [22]. Moreover, a lot of researchers focus on developing lightweight hardware and/or software implementation of existing and well-established algorithms like AES, IDEA, TEA and DES. A characteristic example is the work of Feldhofer et al. [23] who presented a hardware implementation of the AES algorithm, supporting encryption, decryption and key setup and which occupies an area of 0.25mm2, the size of a small grain of sand, and draws only 3.0μA of current at 1.5V.

Even though several mature block ciphers are available and their characteristics (i.e. performance and security) are well understood and documented, stream cipher designs are lacking in comparison. The eSTREAM Project [24] led to the development of several efficient stream cipher designs but the level of security provided on resource-constrained environments is not adequate.

Hash functions design is another area where further research is required. The existing functions are not sufficiently lightweight [25] and, although the SHA-3 competition has helped our understanding of hash functions significantly, still hashes based on block ciphers may have an advantage.

Asymmetric algorithms and protocols must also be adapted to operate on devices with the aforementioned resource limitations. This is an elaborate task, since asymmetric ciphers are computationally far more demanding than their symmetric

counterparts and are usually used with powerful hardware. The performance gap is exacerbated on constrained devices such as 8-bit microcontrollers. Even an optimized asymmetric algorithm like elliptic-curve cryptography (ECC) performs 100 to 1000 times more slowly than a standard symmetric algorithm like AES which correlates to two or three orders-of-magnitude higher power consumption.

Since asymmetric ciphers come with such intrinsic performance issues they are mainly used for key-management facilities and non-repudiation, whereas integrity checks, entity authentication and the encryption are provided by symmetric primitives. Symmetric cryptosystems, on the other hand, must operate on a shared secret scheme, where the master key is shared among all authorized entities and is used for verifying the authenticity of those entities by their peers or by control nodes in order to be accepted in the network. If a component gets compromised and the master key is revealed, then the whole system is compromised. This scheme is not particularly suitable for the heterogeneous, distributed, unattended and dynamically formulated networks, often deployed at physically insecure environments. Conversely, the use of asymmetric cryptography ensures that, since authorized entities do not share a secret but each node has its own secret key, compromising a node does not imperil as significantly the overall ESs network but only that very one entity.

In terms of practical relevance, three families of established public-key algorithms stand out: ECC, RSA and Discrete Logarithms. ECC, in specific, is considered the most attractive option in ESs because of its small operand length and relatively low processing requirements. HyperElliptic Curve Cryptography (HECC, [26]) is another type of curve-based cryptography that has been recently revisited ([27]) by researchers. The advantage of this family of algorithms is that they offer RSA-level security while requiring much smaller parameter sizes. This correlates to smaller data-paths, less memory requirements and lower power consumption.

Figure 2 ([28]) shows the impact of cipher suite selection on energy consumption during the SSL handshake and record stages. The figure demonstrates that a careful choice of cryptographic algorithms, i.e. one that takes into consideration the size of data that will be typically processed and transferred, can greatly reduce the amount of energy consumed.

The need for lightweight cryptography introduces major multi-dimensional challenges in cryptographic algorithms design, from the ES Operating System (OS) to the hardware and software cryptographic provisions embedded on the device itself. Hardware and software co-design seems to offer the best results in terms of speed/size ratio for many ubiquitous computing applications [29]. Regarding primitives that cannot yet be effectively implemented (e.g. hashes in the case of crypto and public key crypto in the case of asymmetric), alternatives could be investigated so that the protocols which are based upon them can be researched further and, perhaps, employed. Special care should be taken during the development of optimized implementations so that they do not introduce new leakage channels which could be exploited by Side-Channel Attacks (SCA).

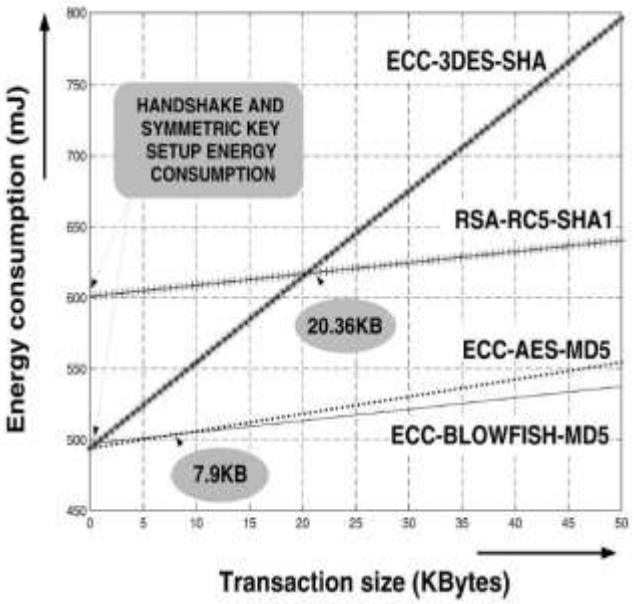


Figure 2. The impact of cipher suite selection on energy consumption during the SSL handshake and record stages.

### *B. Key Distribution Mechanisms*

Key distribution, either for initialization [30] or re-keying has been a challenging topic especially for dynamic, heterogeneous and resource-limited environments. The majority of these schemes is based on symmetric mechanisms, thus requiring pre-distribution of the shared secret with all the disadvantages already discussed. Other schemes [31] are being proposed as well, some of which feature location-aware and identity based mechanisms. Although some of the proposed schemes are indeed energy efficient [32], key management based on shared secrets has proven ineffective, especially in dynamically formulated infrastructures. There have been attempts to correlate key establishment techniques to applications but these were based solely on the use of symmetric keys and on a framework level [33].

This has led part of the research community to focus its attention on public key (e.g. ECC, [34]) schemes. This enables us to distribute authentic public keys via insecure channels as the verifying party does not need to have a copy of the secret key. Therefore, a mobile node's key database may, for example, be updated with all valid public keys once, according to a pre-defined schedule or ad hoc, and from that point onwards the device will be able to authenticate other entities in off-line mode.

## V. NETWORK PROTOCOL & MANAGEMENT ISSUES

### *A. Secure Resource Management*

Certain applications of embedded systems, like Wireless Sensor Networks, rely on the integrity of the platform for providing trustworthy services (e.g. measurements taken by a sensor). It is, therefore, essential to have a method of validating this integrity and assuring that system components have not been compromised. The integrity of the service requester

platform, i.e. control node, must also be validated before allowing it to allocate resources to the nodes it controls or receive the data these nodes have collected. In addition, it should be established that these secure resource management mechanisms will not act as a bottleneck in service performance. Examples of current research on the subject are the WS-Attestation [35] mechanism, developed by IBM, which enables TPM remote platform attestation using web services and the TECOM project [36].

Inspecting the problem from a higher level, middleware resources should be managed by monitoring their availability, enforcing a policy based on which these resources are assigned, implementing a secure model for the identification and authorization of requests as well as an account system to track resource usage. Most of the above can be found in protocol Diameter [37], successor to RADIUS, which offers strong authentication, authorization, accounting and resource management mechanisms. Diameter is already adopted by many IP systems like in the 3rd Generation Partnership Project (3GPP, [38]).

### *B. Reputation-based Schemes*

Reputation-based schemes are a novel paradigm for enhancing security in various applications, including secure routing and intrusion detection systems for Mobile Ad Hoc Networks (MANETS). These systems are easy to implement, lightweight and can protect a MANET from a wide variety of attacks, as CORE [39], CONFIDANT [40] and OCEAN [41], among other schemes, have demonstrated.

The basic concept is inspired from social behavior and relies on the cooperation of the nodes. Much like human interaction, each entity decides to trust or ignore a new, unknown entity based on the opinion of his/her peers about the individual in question. Consequently and much like social networks, trustworthy behavior is encouraged. The three main goals identified [42] for reputation systems are:

- To provide the required information in order to distinguish between a trustworthy principal and an untrustworthy one.
- To encourage principals to act in a trustworthy manner.
- To discourage untrustworthy principals from participating in the service.

Reputation-based mechanisms can also be used in Intrusion Detection Systems (IDSs). Watchdog and Path-rater [43] are commonly used components in such systems. Watchdog is a monitoring component and based on its observations Path-rater ranks the available routing paths. Misbehavior detection and intrusion detection can either be distributed, where information about entities' reputation changes are immediately broadcast to the whole network, or local, in which case each entity decides based solely on its own data about the reputation of other nodes. It should be noted, however, that the latter is not as effective in terms of speed in detection and isolation of malicious nodes [44].

### *C. Anonymity And Location Privacy*

Location-based applications are a relatively new and rapidly expanding market, owing to the widespread use and advances both in mobile devices and positioning systems. Enhanced Reality applications and services are starting to emerge and are expected to spread in the coming years while other examples include location-aware emergency response, entertainment and/or advertisement. The challenge lies in the fact that the location of an individual constitutes sensitive personal data as it can reveal information about his/her personal relationships, political affiliations, medical issues etc. Disclosure of such information can enable a malicious user to harass, blackmail or even enter the individual's residence (e.g. when he/she is away). Thus, such information needs to be handled accordingly and there is on-going research on the subject, including mechanisms for safeguarding location privacy [45][46] as well as reports on the weaknesses of current "sanitization" mechanisms [47][48].

### *D. Secure Service Discovery, Composition And Delivery Protocols*

Services in distributed networks must be discovered, composed and delivered in a secure way. The Organization for the Advancement of Structured Information Standards (OASIS, [49]) has released related standards including WS-Security, WS-Policy, WS-Trust and WS-Secure Conversation which have already been approved. The current trend is to bring web services into ESs and it is thus imperative to adopt the aforementioned specifications. More recently, OASIS developed two standards: Devices Profile for Web Services (DPWS, [50]) and Web Services Dynamic Discovery (WS-Discovery, [51]), which specify the use of web services based communications in resource-constrained and ad hoc environments. Further research has been done by the Service-Oriented Architecture for Devices (SOA4D, [52]) open source initiative which facilitates the development of service-oriented software components adapted to the requirements of embedded devices.

## VI. CONCLUSION

The aim of this paper was to provide an overview of the security concerns in ESs design and implementation. It is essential to consider the intrinsic and often application-specific characteristics of ESs and their particular requirements which not only introduce new vulnerabilities but also exacerbate existing ones and, moreover, limit the efficacy of established computer security techniques and mechanisms, including access control mechanisms, cryptographic primitives and network protocols.

Further research is required on various ES-friendly security mechanisms not merely because of the potentially dire consequences a successful security attack might have in the case of critical systems but also because these attacks are bound to become more common as ESs become an even more integral part of our lives, with the widespread adoption of smart devices in our homes, cars, clothes etc. The robustness and efficiency of these inherent security concerns become a challenging task to research and development efforts.

## REFERENCES


[1] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, "Experimental security analysis of a modern automobile," In Proc. of the IEEE Symposium on Security and Privacy, pp.447-462, 2010.

[2] "Did Stuxnet Take Out 1,000 Centrifuges at the Natanz Enrichment Plant?". Institute for Science and International Security. 22 December 2010.

[3] Srivaths Ravi, Anand Raghunathan, Paul Kocher, Sunil Hattangady, Security in Embedded Systems: Design Challenges, ACM Transactions on Embedded Computing Systems, Vol. 3, No. 3, August 2004, Pages 461–491.

[4] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults. Lecture Notes in Computer Science, 1233:37{51, 1997.

[5] André Kurs, Aristeidis Karalis, Robert Moffatt, J. D. Joannopoulos, Peter Fisher and Marin Soljačić. "Wireless power transfer via strongly coupled magnetic resonances," Science 6 July 2007, Vol. 317 no. 5834 pp. 83-86.

[6] Aristeidis Karalis, John D.Joannopoulos, and Marin Soljacic, "Efficient wireless non-radiative mid-range energy transfer," Annals of Physics Vol.323, 34, (2008).

[7] M. Naedele, "An access control protocol for embedded devices," 4th Int. IEEE Conf. on Industrial Informatics, 2006.

[8] C. Szilagyi and P. Koopman, "Flexible multicast authentication for time-triggered embedded control network applications", in Proc. DSN, 2009, pp.165-174.

[9] C. Szilagyi and P. Koopman, "Low cost multicast authentication via validity voting in time-triggered embedded control networks", in Proc. WESS, 2010, pp.10-10.

[10] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, "Denial-of-service attack detection techniques," IEEE Internet Computing, vol. 10, no. 1, 2006.

[11] D. R. Raymond and S. F. Midkiff. "Denial-of-service in wireless sensor networks: attacks and defenses," in IEEE Pervasive Computing, volume 7, pages 74–81, 2008.

[12] Wood A. D. and Stankovic J. A. "A taxonomy for denial-of-service attacks in wireless sensor networks", in Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, edited by Mohammad Ilyas and Imad Mahgoub, CRC Press LLC, 2005.

[13] Rich Smith - HP Labs, "PhlashDance, Discovering permanent denial of service attacks against embedded systems," EUSecWest 2008.

[14] K. Stefanidis and D. N. Serpanos, "Implementing filtering and traceback mechanism for packet-marking IP-based traceback schemes against DDoS attacks," in IEEE International Conference on Intelligent Systems, Varna, Bulgaria, 2008.

[15] Aad, I., Hubaux, J., and Knightly, E. W., "Impact of denial of service attacks on ad hoc networks," in IEEE/ACM Trans.Netw. 16, 4 (Aug. 2008), 791-802.

[16] Daněk Martin, Kadlec Jiří, Bartosinski Roman, Kohout Lukáš, "Increasing the level of abstraction in FPGA-based designs," International Conference on Field Programmable Logic and Applications , Eds: Kebschull Udo, International Conference on Field Programmable Logic and Applications, (Heidelberg, DE, 08.09.2008-10.09.2008).

[17] TPM Main Specification, ISO/IEC standard 11889, http://www.trustedcomputinggroup.org/resources/tpm_main_specification

[18] M. Razvi Doomun and KMS Soyjaudah, "Analytical comparison of cryptographic techniques for resource-constrained wireless security," International Journal of Network Security, Vol.9, No.1, PP.82–94, July 2009.

[19] Bart Preneel, "Research challenges in lightweight cryptography," WiSec 2009 invited talk.

[20] Thomas Eisenbarth, Sandeep Kumar, Christof Paar, Axel Poschmann, Leif Uhsadel, "A survey of lightweight-cryptography implementations," IEEE Design & Test, Volume 24, Issue 6, November 2007.

[21] G. Leander et al., "New lightweight DES variants," Proc. Fast Software Encryption (FSE 07).

[22] A. Bogdanov et al., "PRESENT: An ultra-lightweight block cipher," Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES 07).

[23] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES implementation on a grain of sand," IEE Proc, vol. 152, no. 1, Oct. 2005, pp. 13-20.

[24] The eSTREAM Project, http://www.ecrypt.eu.org/stream/

[25] Andrey Bogdanov, Gregor Leander, Christof Paar, Axel Poschmann, M.J.B. Robshaw, Yannick Seurin, "Hash functions and RFID tags: Mind The Gap," Workshop on Cryptographic Hardware and Embedded Systems (CHES 2008), LNCS, Springer-Verlag, 2008.

[26] N. Koblitz, "Hyperelliptic cryptosystems," Journal of Cryptology, 1(3):129–150,1989.

[27] Junfeng Fan, Lejla Batina, Ingrid Verbauwhede, "HECC goes embedded: An area-efficient implementation of HECC," Selected Areas in Cryptography 2008.

[28] N.R. Potlapally, S. Ravi, A. Raghunathan, and N.K. Jha, "Analyzing the energy consumption of security protocols", in Proc. ISLPED, 2003, pp.30-35.

[29] Thomas Eisenbarth, Sandeep Kumar, Christof Paar, Axel Poschmann, Leif Uhsadel, "A survey of lightweight-cryptography implementations," IEEE Design & Test, Volume 24, Issue 6, November 2007.

[30] Cynthia Kuo, Mark Luk, Rohit Negi, Adrian Perrig, "Message-In-a-Bottle: User-friendly and secure key deployment for sensor nodes," in Proceedings of the ACM Conference on Embedded Networked Sensor System (SenSys), 2007.

[31] Barry Doyle, Stuart Bell, Alan F. Smeaton, Kealan McCusker, and Noel O'Connor, "Security considerations and key negotiation techniques for power constrained sensor networks," The Computer Journal (Oxford University Press), 49(4):443–453, 2006.

[32] J. Huang, J. Buckingham, R. Han, "A level key infrastructure for secure and efficient group communication in wireless sensor networks," First IEEE/CreateNet Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm), 2005, pp. 249-260.

[33] K.M. Martin and M.B. Paterson, "An application-oriented framework for wireless sensor network key establishment," Proceedings of the Third Workshop on Cryptography for Ad-hoc Networks WCAN'07 (2007), Electronic Notes in Theoretical Computer Science 192(2), 31-41, 2008.

[34] Malan, David J., Matt Welsh, and Michael D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," IEEE SECON 2004: 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks: 4-7 October, 2004, Santa Clara, California, 71-80. Piscataway, N.J.: IEEE.

[35] Sachiko Yoshihama, Tim Ebringer, Megumi Nakamura, Seiji Munetoh, Takuya Mishina, Hiroshi Maruyama, "WS-Attestation: Enabling Trusted Computing on web services," October 2006.

[36] Trusted Embedded Computing, http://www.tecom-itea.org

[37] P. Calhoun et al., "Diameter base protocol," IETF, RFC 3588, September 2003.

[38] 3rd Generation Partnership Project (3GPP) , http://www.3gpp.org.

[39] P. Michiardi, R. Molva, "Core: A COllaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks," Institut Eurecom Research Report RR-02-062 - December (2001).

[40] Sonja Buchegger and Jean-Yves Le Boudec, "Performance analysis of the confidant protocol: Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks," Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne, June 2002.

[41] Sorav Bansal and Mary Baker, "Observation based cooperation enforcement in ad hoc networks," Technical Report, Stanford University, arXiv:cs.NI/0307012 v2 6 Jul 2003.

[42] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," in M. R. Baye, editor, The Economics of the Internet and E-Commerce, volume 11 of Advances in Applied Microeconomics. Amsterdam, Elsevier Science, 2002.

[43] Sonja Buchegger, Cedric Tissieres, Jean-Yves Le Boudec, “A test-bed for misbehavior detection in mobile Ad-hoc networks, How much can watchdogs really do?," Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04), pp. 102-111, 2004.

[44] S. Buchegger and J.-Y. Le Boudec, “The effect of rumor spreading in reputation systems for mobile ad-hoc networks," Proc. WiOpt'03(Modeling and Optimization in Mobile Ad Hoc and Wireless Networks), 2003.

[45] Gedik, B.; Ling Liu, “Protecting location privacy with personalized k-Anonymity: architecture and algorithms,” Mobile Computing, IEEE Transactions, Volume 7, Issue 1, pp. 1-18, Jan. 2008.

[46] Ge Zhong and Urs Hengartner, “Toward a distributed k-Anonymity protocol for location privacy,” http://www.cs.uwaterloo.ca/~uhengart/publications/wpes08.pdf

[47] Golle, P.; Partridge, K, “On the anonymity of home, work location pairs,” Proceedings of the 7th International Conference on Pervasive Computing; 2009 May 11-14; Nara, Japan. Berlin: Springer; 2009; LNCS 5538: 390-397.

[48] Marco Gruteser and Baik Hoh, “On the anonymity of periodic location samples,” http://www.winlab.rutgers.edu/~gruteser/papers/gruteser_anonymityperiodic.pdf

[49] OASIS: Advancing open standards for the global information society, http://www.oasis-open.org

[50] Driscoll and Mensch, "OASIS Devices Profile for Web Services (DPWS) Version 1.1," 2009, http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01

[51] OASIS Web Services Dynamic Discovery (WS-Discovery), http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01

[52] SOA4D Forge, https://forge.soa4d.org/