Embedded Systems Security Challenges

Konstantinos Fysarakis[1], George Hatzivasilis[1], Konstantinos Rantos[2], Alexandros Papanikolaou[1] and Charalampos Manifavas[3]

[1]*Dept. of Electronic & Computer Engineering, Technical University of Crete, Chania, Greece*

[2]*Dept. of Computer & Informatics Engineering, Eastern Macedonia and Thrace Institute of Technology, Kavala, Greece*

[3]*Dept. of Informatics Engineering, Technological Educational Institute of Crete,*
Heraklion, Crete, Greece

kfysarakis@isc.tuc.gr, krantos@teikav.edu.gr, alxpapanikolaou@gmail.com, gchatzivasilis@isc.tuc.gr,
harryman@ie.teicrete.gr

Keywords: Embedded Devices, Security, Denial of Service, Lightweight Cryptography, Location Privacy, Secure Routing

Abstract: In a world of pervasive computing, embedded systems can be found in a wide range of products and are employed in various heterogeneous domains. The abovementioned devices often need to access, store, manipulate and/or communicate sensitive or even critical information, making the security of their resources and services an important concern in their design process. These issues are further exacerbated by the resource-constrained nature of the devices, in conjunction with the ever-present need for smaller size and lower production costs. This paper aims to provide an overview of the challenges in designing secure embedded systems, covering both node hardware and software issues, as well as relevant network protocols and cryptographic algorithms. Moreover, recent advances in the field are identified, highlighting opportunities for future research.

1 INTRODUCTION

Embedded systems (ESs) permeate our lives in various forms, ranging from avionics to e-textiles, automobiles, home automation and wireless sensor nodes. In terms of their physical size, they range from miniature wearable or sensor nodes (i.e. motes) to large industrial deployments of programmable logic controllers (PLCs).

The various intrinsic and application-specific characteristics of ESs complicate the task of guaranteeing the security, namely handling the confidentiality, integrity and availability aspects of their applications and the data they handle. Their characteristics habitually include resource constraints (namely computational capabilities, storage capacity, memory and power), dynamically formulated, remotely-managed networking and even unattended operation in hostile environment and time-critical applications. Therefore, while securing networked computer systems is not a novel concern, the techniques developed for personal and enterprise systems are often unsatisfactory or even inapplicable to embedded devices.

In addition to the above, ES applications often feature direct interaction with the physical world, being responsible for vital, time-critical applications, where a delay or a speed-up of even a fraction of a second in system's response or reaction could have dire consequences. This further differentiates ES security, as a security incident in a critical application may lead to asset damage or even personal injury and death. Researchers recently demonstrated that it is feasible to manipulate all critical sub-systems in modern automobiles by using a wireless-enabled MP3 player connected to the vehicle's embedded control network (Koscher et al., 2010). The presented attacks 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.

Moreover, next-generation ES services, like the ones pertaining to the Internet of Things (IoT), may require the integration of multiple administrative domains (e.g. one domain may host the devices and enable access to devices and information,whereas another domain may make use of the information for designing innovative services). Each domain will typically have its own security requirements and constraints, therefore ensuring interoperability of security is a challenging task (Alam et al., 2011).

The paper is organised as follows: Section 2 presents some physical security issues that are evident in embedded systems. The various access control mechanisms used for controlling access to resources are presented in Sec. 3. Indicative examples of cryptographic mechanisms specially-crafted for embedded systems are presented in Sec. 4. Various protocol and management issues are presented in Sec. 5, and the paper concludes in Sec. 6.

2 PHYSICAL SECURITY ISSUES

Regarding the physical layer and given the often unattended nature of deployed ESs, sometimes within hostile environments, the risk of device tampering should not be ignored. In the remainder of this section, some aspects of physical security in ESs are presented.

2.1 Side Channel Attacks

A malicious entity's physical access to a non-tamper-resistant device, apart from providing physical access to the system components, would also enable the launch of various attacks like micro-probing and reverse engineering or sophisticated side-channel attacks (SCA), timing attacks, simple power analysis (SPA), differential power analysis (DPA), as well as their electro-magnetic counterparts, SEMA and DEMA respectively, and also differential fault attacks (DFA). 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.

In the case of ESs utilising field-programmable gate arrays (FPGAs), the concept of run-time reconfiguration (Daněk et al., 2008) 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 the measurement of electromagnetic 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.

2.2 Trusted Platform Module

A Trusted Platform Module – TPM (Trusted Platform Module, 2009), is a microcontroller that can securely store a relatively small amount of information on it, which can then be used either for authenticating the platform (e.g. passwords, certificates, encryption keys) or for ensuring that the platform has not been breached (e.g. platform measurements, configuration data). It is worth clarifying that the microcontroller does not control the software that is running on the platform itself. Instead, through the tamper-resistant security functions it provides, the platform's operating system or any running applications access the necessary information, so as to determine and implement their security policies accordingly. The software embedded on such a TPM component is directly related to the component's physical size (the higher the memory requirements, the larger the module's surface) and consequently cost. Any optimisation of the module's software will have a direct impact on the overall execution speed, as well as to the power consumption (faster execution allows the module to return to a more power-sparring idle/sleep state).

2.3 Protection of Power Supply

Several types of embedded systems devices are mostly battery-powered, something that creates issues of energy constraints. Especially in cases where e.g. small sensor nodes are meant to be used in unattended environments, they are expected to operate for certain time intervals (sometimes spanning over a few months) until they have their power source replaced or recharged. An attacker could therefore launch a Denial-of-Service (DoS) attack, aiming at draining the battery power by forcing extensive use of the device's wireless connection or CPU.

In order for the electronic parts of the embedded device to function properly, continuous power is required, with both voltage and current levels lying within specific limits. The power source should also be able to monitor its own state and react accordingly in cases where an issue is detected that could affect the normal operation of the system. Moreover, suitable fail-safe mechanisms should exist, implemented in software and/or hardware, that would protect the device and prevent any potential damage from spreading across the rest of its components.

Most of the requirements mentioned above are satisfied in modern uninterruptible power supply (UPS) systems, nevertheless, they cannot easily be applied to the operating environments of some embedded systems, due to strict size and cost constraints. Instead, alternative solutions are implemented, such as energy scavenging, super-capacitors, micro-solar cells and remote/wireless power transferring schemes (Kurs et al., 2007; Karalis et al., 2008), to name a few. Still, such solutions must carefully be adopted to each specific scenario (e.g. a micro-solar cell is useless if the device cannot be reached by enough direct sunlight), also taking into consideration fail-safe options with respect to the criticality of the possible failures and the probability for them to occur, so as to protect the device effectively.

3 ACCESS CONTROL

Access control mechanisms are essential to prevent unauthorised/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 the application under consideration. Some common methods include:

1. Profile authentication: If a node has some specific characteristics (e.g. hardware specifications, operating system), it can join an existing network.
2. 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.
3. 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 of ES networks can demonstrate and the scalable remote manageability often required (Naedele, 2006). Moreover, even in wired embedded networks and in industries such as automotive and aviation, most control networks utilised (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 DoS attacks (Szilagyi and Koopman, 2009; Szilagyi and Koopman, 2010), leading to critical vulnerabilities like the ones already mentioned in (Koscher et al., 2010).

3.1 Policy-Based Access Control

ESs are often deployed in applications bound by strict security requirements (e.g. e-Health applications are a prime example), including secure transmission of sensitive data to remote entities, instructions that need to reach actuators in an unaltered form, robust entity authentication and access control mechanisms (Alhaqbani and Fidge, 2008).

Regarding the latter, among the proposed schemes that have gained popularity are those where decisions are made based on policy restrictions. Such a scheme is the standardised by OASIS eXtensible Access Control Markup Language (XACML), an XML-based general-purpose policy decision language. Besides being used for representing authorisation and entitlement policies for managing access to resources, it provides a processing model for evaluating requests and making decisions based on the defined set of policies (Rantos et al., 2012). Policy-based access control allows dynamic decision making on controlled nodes' resources based on policy restrictions set by the system owner.

An XACML architecture commonly consists of the following components:

- Policy Enforcement Point (PEP): The system entity that performs access control, by making decision requests and enforcing authorisation decisions.
- Policy Administration Point (PAP): The system entity that creates a policy or policy set.
- Policy Decision Point (PDP): The system entity that evaluates applicable policy and renders an authorisation decision.
- Policy Information Point (PIP): The system entity that acts as a source of attribute values.
- Context Handler: It orchestrates the communications among the stakeholders, converts, if necessary messages between their native forms and the XACML canonical form, and collects all necessary information for the PDP.

The aforementioned components are complemented by a well-defined set of policies which define the rules that should be taken into account when examining access requests.

In a typical data flow model, authorisation requests for accessing nodes' resources, i.e. PEPs are forwarded to PDPs together with the required attributes taken from the PIP. The PDP evaluates the request against the policy restrictions taken from PAP and issues an authorisation decision which the PEP has to enforce.

These components do not necessarily run on the same node, mainly because resource-restricted nodes do not have the capacity to accommodate them. In this case more distributed approaches have to be adopted to offload computationally expensive operations run by a PDP to more powerful devices.

Still, unprotected policy messages would expose the system's security, revealing private information to attackers who might also try to identify policy restrictions and do a mapping of the security measures taken for the specific environments, hence exploiting potential vulnerabilities. Moreover, in a more active approach, an attacker might masquerade as a legitimate entity or modify policy-related messages, such as authorisation requests and/or decisions, obligations or advices in an attempt to downgrade adopted measures and bypass access controls. To avoid the aforementioned problems, appropriate security measures, like the ones detailed in later sections of this paper, have to be deployed to safeguard message confidentiality, integrity and authentication.

3.2 Denial of Service (DoS)

The aim of a DoS or a Distributed DoS (DDoS) attack is to harm the availability of a specific node or a network of nodes, thus preventing or delaying legitimate entities from accessing the services or resources they wish to (Carl et al., 2006). DoS attacks on ES nodes can take multiple forms, such as exploiting the vulnerabilities in their software/firmware, or attacking the network they belong to by jamming, misrouting, flooding and so on. DoS attacks can be mounted more or less on every layer, by exploiting the particular characteristics or mechanisms found in them. Their aim is to cause the nodes to constantly process or send dummy data, thus draining their power source via unnecessary use of their wireless connection and prolonged demand for memory and CPU cycles. The effects of such attacks are particularly critical in the case of nano and micro/personal nodes, where the power reserve is usually rather limited. In addition, flooding types of DoS attacks consume part of the network's bandwidth, which may also indirectly affect the normal operation of the network or a significant part of it, depending on the overall network's capacity.

Another type of DoS attack involves the case where an attacker gains physical access to the device and modifies or destroys it as a physical entity. Depending on the role of this particular node for the rest of the network or cluster of nodes it belongs to, its unavailability could have a significant impact on these entities. For instance, it could lead to partial or total loss of the data sink, selection of non-optimal routes, or even the loss of a control node that is vital for the normal operation of the system.

Given that large-scale networks are most probably heterogeneous in their nature, they contain different capabilities and vulnerabilities, which need to be addressed independently for achieving effective protection against DoS attacks. What is more, in cases where there is provision for dynamic network size variation, shielding against DoS attacks can become a very challenging task (Raymond and Midkiff, 2008). In addition, the problem becomes even more complex and difficult to solve for cases where the available resources and capabilities exhibit strict limitations.

In cases where the ES network is of an unattended nature, the use of a remote management system is vital. Nevertheless, such systems offer additional attack surface and their compromise can allow an adversary to upload malicious firmware. In this way, the attacker is able to corrupt memory and data sent/received or even lead to permanent damage of hardware sub-components by intended misconfiguration (Permanent DoS – PDoS, or *bricking*). The success of such attacks is based on the fact that the various firmware upgrade mechanisms are usually insecure and do not employ complex security mechanisms for verifying authenticity and integrity, such as the ones used in digital certificates. This process of rendering a device unbootable or non-reflashable is also known as *phlashing* (Smith, 2008).

One of the main reasons for the DoS attacks being relatively easy to successfully launch is the use of old protocols that suffer from lack of security requirements. For instance, the IP protocol takes for granted various assumptions regarding the trust of network nodes and consequently does not dispute the related information found in the packet headers and/or payload. Any integrity-checking mechanisms are rather primitive and simple in nature (e.g. checksums), as their aim is to detect accidental data corruption and not deliberate modification of information. Therefore, continuing to use such protocols as the base for building custom network communication protocols on it makes it particularly hard to design (D)DoS-resilient systems and services. An additional obstacle is the fact that basic software methodologies do not take into consideration security requirements, able to deal with such kinds of attacks (Stefanidis and Serpanos, 2008).

The majority of the aforementioned attacks can be avoided by employing suitable authentication, access control and integrity-checking mechanisms. It is also equally important to provide secure mechanisms for node firmware deployment and software updates, able to verify both the authenticity and integrity of the firmware/software to be uploaded and reject it if it does not pass the required checks. Furthermore, more recent network protocols should be used, able to provide means and metrics for quantifying (D)DoS attack resilience (Aad et al., 2008). The use of intrinsically secure ES firmware offering various fail-safe mechanisms or even hardware redundancy could be employed in cases where dependability is highly critical (e.g. avionics and the military), as they are expected to increase the cost of the end-product.

4 CRYPTOGRAPHIC MECHANISMS

As has already been mentioned, embedded devices often have inherent limitations in terms of processing power, memory, storage and energy. Efficient algorithm designs and implementations that adhere to these constraints, while satisfying application demands, can significantly impact battery lifetime and allow the implementation of many applications.

Key management is an equally important issue, both from a security and a management point of view. The rather simple pre-shared key (PSK) scheme, where every embedded device has the necessary cryptographic keys pre-installed, is difficult to manage in distributed and dynamic environments (physical access to the device is required) or in cases where there is a large number of such devices. Moreover, the disclosure of the master key leads to the instant compromise of all the system/network. Having an appropriate scheme that triggers periodic re-keying limits the amount of ciphertext that has been encrypted with the same key, thus increasing the system's security level.

This section presents several lightweight cryptographic and key management schemes, suitable for resource-constrained devices.

4.1 Lightweight Cryptography

Lightweight Cryptography (LWC) 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 optimisation, aiming to maintain the security level that "traditional" algorithms and implementations offer while narrowing what is often referred to as "battery gap" (Doomun and Soyjaudah, 2009), i.e. the very high energy consumption overheads for supporting security on battery-constrained systems. A number of surveys (Preneel, 2009; Eisenbarth et al., 2007) provide an overview of this subject.

ISO/IEC 29192 includes ciphers and cryptographic mechanisms for LWC. The standards are PRESENT (Bogdanov et al., 2007) and CLEFIA (Akishita and Hiwatari, 2012) for block ciphers, and TRIVIUM (ECRYPT, 2008) and Enocoro (Watanabe et al., 2008) for stream ciphers.

Compact implementations of "traditional" ciphers, like AES (Feldhofer et al., 2005), are also applied to embedded devices. Newer lightweight block ciphers are Humminbird-2 (Engels et al., 2011), Piccolo (Shibutani et al., 2011), Simon (Beaulieu et al., 2013), SPECK (Beaulieu et al., 2013), ITUbee (Karakoç et al., 2013). For stream ciphers, the finalists of the eSTREAM project Grain, Salsa20, Rabbit and HC128 (ECRYPT, 2008) are suitable for LWC.

Hash functions design is another area where further research is required. For LWC, compact implementations of the standardised functions SHA-2 (Bogdanov et al., 2008), SHA-3 (Gaj et al., 2012) are examined. Newer functions that are suitable in this domain are Blake (Gaj et al., 2012), Photon (Guo et al., 2011), SPONGENT (Bogdanov et al., 2011) and other hash functions based on lightweight block ciphers, like DM-PRESENT (Bogdanov et al., 2008).

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 executed on powerful hardware. The performance gap is exacerbated on constrained devices, such as 8-bit microcontrollers. Even an optimised asymmetric algorithm (e.g. elliptic-curve cryptography – ECC), performs 100 to 1000 times more slowly than a standard symmetric algorithm (e.g. AES), which correlates to a proportionally higher power consumption.

In terms of practical relevance, two families of established public-key algorithms stand out: ECC and NTRU (Kamal and Youssef, 2009). ECC in particular is considered the most attractive option in ESs, due to its small operand length and its relatively low processing requirements. NTRU is the most popular lattice-based cryptosystem. Its security is based on the shortest vector problem and it can efficiently be deployed on embedded systems. Compared to ECC in hardware, NTRU is 1.5 times faster with only the 1/7 of the memory footprint. Compared to RSA in software, it is 200 times faster in key generation, almost 3 times faster in encryption and 30 times faster in decryption.

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 (Eisenbarth et al., 2007). 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 put into practice. Special care should be taken during the development of optimised implementations so that they do not introduce new leakage channels which could be exploited by Side-Channel Attacks (SCA).

A comparative analysis of lightweight ciphers on embedded systems was performed recently, where the authors evaluate the proposed schemes based on performance metrics and classify them for various types of embedded devices (Manifavas et al., 2013). Various cryptographic libraries exist that offer key establishment mechanisms and communication protocols, like for example ULCL (Hatzivasilis et al., 2014), CyaSSL and OpenSSL.

4.2 Key Distribution Mechanisms

Key distribution, either for initialisation (Kuo et al., 2007) 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 (Doyle et al., 2006) 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 (Huang et al., 2005a), 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 (Martin and Paterson, 2008).

This has led part of the research community to focus its attention on public-key schemes (e.g. ECC). This enables the distribution of 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.

Identity-based cryptography – IBC (Boneh and Franklin, 2003) provides an alternative solution to the very expensive, for many nodes, public-key cryptography. IBC allows publicly-available information that can uniquely identify participating nodes to be utilised for the secure exchange of keys. Thus, nodes do not have to depend on a public key infrastructure and digital certificates for the exchange of authenticated public keys. The advent of identity-based schemes and pairing-based cryptography has shown that such schemes offer a promising solution for managing keys in resource-constrained nodes (Oliveira et al., 2011).

Even with a public-key scheme, we still need to implement symmetric cryptography. Thus, alternative schemes that are based only in symmetric cryptography are also proposed (Chen and Chao, 2011; Simplicio et al., 2010). Some of them are location-aware and identity-based mechanisms and are energy-efficient. Their disadvantage is the pre-distribution phase that is required prior to first usage. The schemes are inefficient in dynamic environments but are proposed in applications where public-key schemes cannot be used.

5 NETWORK PROTOCOL AND MANAGEMENT ISSUES

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 for 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 mechanism (Yoshihama et al., 2007), which enables Trusted Platform Module (TPM) remote platform attestation using web services.

5.1 Secure Resource Management

Inspecting the problem from a higher level, middleware resources should be managed by monitoring their availability, enforcing a policy based on which of these resources are assigned, implementing a secure model for the identification and authorisation of requests, as well as an accounting system to track resource usage. Most of the above can be found in protocol Diameter (Calhoun et al., 2003), successor to RADIUS, which offers strong authentication, authorisation, accounting and resource management mechanisms. Diameter is already adopted by many IP systems like in the 3rd Generation Partnership Project (3GPP).

5.2 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 (Hatzivasilis and Manifavas, 2012). These systems are easy to implement, lightweight and can protect a MANET from a wide variety of attacks.

The basic concept is inspired from social behaviour 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 its peers about the individual in question. Consequently, much like in social networks, trustworthy behaviour is encouraged. The three main goals identified (Resnick and Zeckhauser, 2002) 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 are used in Intrusion Detection Systems (IDSs) and provide two main functionalities: Secure routing and resource management. Watchdog and Path-rater (Buchegger et al., 2004) are the building blocks of such systems. Watchdog is a monitoring component and based on its observations Path-rater ranks the available routing paths. The main steps in the reasoning process of a reputation-based scheme are as follows:

1. Gather information.
2. Score and rank.
3. Select entity.
4. Transaction.
5. Reward and punish.

Misbehaviour 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 detecting and isolating of malicious nodes. Known reputation-based systems for secure routing are IRmIDS (Madhavi and Kim, 2011), Reputated-ARAN (Abdalrazak and Sawant, 2012) and CSRAN (Zhang et al., 2008).

While secure routing provides P2P protection, reputation-based systems for secure resource management provide end-to-end protection. Such systems rank resources, providers and consumers. Based on these values, network entities are able to recognise legitimate providers and resources of good quality while keeping out selfish and malicious users. Several reputation-based schemes have been proposed for resource management in P2P and Grid networks (Damandeep and Jyotsna, 2012).

5.3 Anonymity and Location Privacy

Location-based applications constitute a rapidly expanding market, owing to the widespread use and advances in mobile devices and positioning systems alike. Enhanced reality applications and other similar services are starting to emerge and are expected to spread in the coming years. Other examples of such smart services include location-aware emergency response, enhanced entertainment and/or advertisement services, or even location monitoring of personnel and fleets of vehicles.

The location of individual users is necessary in order to enable the abovementioned services but, even though its disclosure may not pose a security risk for the embedded device itself, said information constitutes sensitive personal data of the user or users associated with each device and should be handled accordingly. 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). There is on-going research on the subject, including mechanisms for safeguarding location privacy (Gedik and Liu, 2008; Zhong and Hengartner, 2008) as well as reports on the weaknesses of current "sanitisation" mechanisms (Golle and Partridge, 2009; Gruteser and Grunwald, 2005).

Literature on this topic includes variations and enhancements of a few recurring methods. Anonymisation methods aiming to remove identifying information using generalisations and suppressions are the most popular in the literature, with k-anonymity being the basic mechanism used, as described in (Zhong and Hengartner, 2008). The principle of k-anonymity involves the use of a cloaking area, where there are at least k users in it, and blurs their identities in order to make each user's identity indistinguishable from the rest $k-1$ users. In general, there are two important and mostly unavoidable trade-offs when choosing this k-value: A trade-off between privacy and quality of service and a trade-off between privacy and personalisation (Liu, 2007). K-anonymity-based schemes have typically being deployed for privacy-preserving location monitoring and to allow mobile users to take advantage of location-based personalised services without compromising their privacy (Fysarakis et al., 2013).

Elementary anonymity schemes (e.g. a pure k-anonymity scheme) are inadequate if used independently and hence they are often combined with auxiliary mechanisms, in order to achieve better privacy safeguards (Golle and Partridge, 2009; Gruteser and Hoh, 2005; Liu, 2009). Attempting to further enhance these basic schemes, the use of personalised k values is proposed, for systems with context-sensitive privacy requirements (Gedik and Liu, 2008). Moreover, if combined with l-diversity (Machanavajjhala et al., 2007), another dimension can be added to k-anonymity, where l is a set of distinct locations. Other than the above suggested improvements, the use of dummy locations (Gupta et al., 2011) and semantic information (Byoungyoung et al., 2011) are also proposed in the literature, in order to address issues that are not solved by plain k-anonymity mechanisms.

Pseudonym-based methods are another common anonymisation theme, involving disposable pseudonyms for each node in the location service (Gruteser and Grunwald, 2005). As the pseudonyms change over time, they are being used as temporal identifiers, making it hard for potential attackers to track the users. Silence periods can be introduced to further enhance this concept, as detailed in (Huang et al., 2005b). Other literature schemes rely on path perturbation, i.e. trying to cross paths in areas where at least two users meet (Hoh and Gruteser, 2005).

5.4 Secure Service Discovery, Composition and Delivery Protocols

Services in distributed networks must be discovered, composed and delivered in a secure way. The Organisation for the Advancement of Structured Information Standards (OASIS) has released related standards including WS-Security, WS-Policy, WS-Trust and WS-Secure Conversation which have already been approved and the current trend is to bring web services into ESs and it is thus imperative to adopt the aforementioned specifications.

For this purpose, OASIS developed two standards: Devices Profile for Web Services (DPWS) and Web Services Dynamic Discovery (WS-Discovery), which specify the use of web-services-based communications in resource-constrained and ad hoc environments. The profile's architecture includes hosting and hosted services. A single hosting service is associated with each device while the same device may accommodate various hosted services. The latter represent the device's various functional elements and rely on the hosting service for discovery. Discovery services are included as well, enabling devices to "advertise" their presence on the network and search for other devices. Metadata exchange services provide dynamic access to services hosted on a device and their meta-data. Furthermore, publish/subscribe eventing services allow other devices to subscribe to messages provided by a certain service.

Additional research on this area has been conducted by the Service-Oriented Architecture for Devices (SOA4D) open-source initiative which facilitates the development of service-oriented software components adapted to the requirements of embedded devices. Web Services for Devices (WS4D) is another open source initiative, providing a number of toolkits aimed at developing DPWS-compliant applications for resource-constrained devices in ad-hoc networks, maintaining interoperability with regular W3C-specified Web Services. A detailed overview of the WS4D initiative can be found in (Zeeb et al., 2010).

5.5 Communications Security

Embedded nodes have quite a few choices regarding the protocols they adopt in their communication stack, depending on their computation capabilities and needs. One of the predominant solutions is the 6LoWPAN (Hui and Thubert, 2011) stack, i.e. IPv6 over 802.15.4 (IEEE Standard for Local and metropolitan area networks, 2011). Such an approach benefits from the adoption of well-known and standardised solutions for providing security at the network layer. This is IPsec which, however, has to utilise compressed header format (Raza et al., 2011; Rantos et al., 2013a; Rantos et al., 2013b) to fit into the limited message space provided by IEEE802.15.4, i.e. a for low-power, low data rate wireless communication standard for small devices.

The network layer, however, is not the only layer in the TCP/IP stack where messages can be protected. The others are application and data link layer. The corresponding security mechanisms for these two layers are the Datagram Transport Layer Security – DTLS protocol (Kothmayr et al., 2012), which is based on the well-known TLS (Transport Layer Security) but utilises datagram protocols and the inherent security mechanism of IEEE802.15.4 which is defined in the same standard. All these mechanisms were designed to provide at least confidentiality, integrity and message authentication, yet they have slightly different properties.

Security mechanisms found at the bottom layers of the communication stack relieve applications from deploying their own distinct security mechanisms. However, this comes at a cost. With regards to the IEEE 802.15.4 security protocol which protects messages at the data link layer, protection takes place on a node-by-node basis. This introduces significant computational overhead to the nodes which have to decrypt incoming messages, verify their integrity, and re-encrypt them, typically using a different set of keys, prior to forwarding them to the next node. This process consumes valuable node resources on routing nodes.

As opposed to 802.15.4, IPsec offers end to end protection of messages. Therefore, intermediate routing nodes' resources are only used for routing packets and not for message protection. In this sense, security provided at the network layer can be considered as a valuable mechanism for low power and lossy networks.

IPsec can be either used within such a network to secure communications among participating nodes in cases where these are deployed in a hostile environment to secure communications among participating nodes, or between a node and remote party. This second choice requires utilising a gateway, e.g. sink node, which can simply forward messages or set up a tunnel to further protect messages and also provide communicating node details and traffic flow confidentiality. As an example, consider the secure remote access to an aircraft's device to control it in case of an emergency. The aircraft's gateway can be used to further protect the message and hide the addresses of communicating entities, while padding can conceal communication patterns and characteristics.

Compared to IPsec, DTLS demonstrates similar characteristics, in terms of end-to-end message protection, but it suffers from an expensive handshake mechanism and the inability to cope with applications that utilise the TCP protocol.

One of the problems one should consider when deploying one of these three solutions, is the use of robust key management mechanisms designed for resource-constrained devices. Traditional public key cryptography solutions are considered inappropriate in some environments and alternatives, including those that utilise lightweight cryptography, elliptic curves and identity based cryptography should be considered.

6 CONCLUSIONS

This paper provided an overview of the various security issues in ESs design and implementation. The particular characteristics of such resource-constrained devices and the varied requirements of their applications not only introduce new vulnerabilities but also intensify existing ones. Moreover, mechanisms and techniques (e.g. for access control, cryptography, network routing etc.) that would typically be deployed to secure other types of computing devices are not always applicable or have limited efficacy in the context of embedded systems.

Further research is required for establishing secure mechanisms tailored for ESs, in order to address potential threats to their secure operation, including those exacerbated by the intrinsic characteristics of the devices and their application fields, especially in the case of critical systems' applications. What is more, given the widespread adoption of smart devices in our everyday lives (e.g. smart vehicles, smart houses, smart clothing), it is important to deal effectively with the inherent security concerns. The challenge lies with the researchers to put more effort in the above matters and come up with appropriate solutions, thus helping realise the promise of pervasive computing and the Internet of Things (IoT).

ACKNOWLEDGEMENTS

This work has been supported by the Greek General Secretariat for Research and Technology (GSRT), under the ARTEMIS JU research program nSHIELD (new embedded Systems arcHItecturE for multi-Layer Dependable solutions) project. Call: ARTEMIS-2010-1, Grant Agreement No.: 269317.

REFERENCES

Aad, I., Hubaux, J., and Knightly, E. W. (2008). Impact of denial of service attacks on ad hoc networks. *IEEE/ACM Transactions on Networking (TON)*, 16(4):791–802.

Abdalrazak, T. R. and Sawant, H. K. (2012). Collaborative trust-based secure routing based ad-hoc routing protocol. *International Journal of Modern Engineering Research (IJMER)*, 2(2):95–101.

Akishita, T. and Hiwatari, H. (2012). Very compact hardware implementations of the blockcipher CLEFIA. In Miri, A. and Vaudenay, S., editors, *18th international conference on Selected Areas in Cryptography (SAC '11)*, volume 7118 of *LNCS*, pages 278–292, Toronto, ON, Canada.

Alam, S., Chowdhury, M. M. R., and Noll, J. (2011). Interoperability of security-enabled Internet of Things. *Wireless Personal Communications*, 61(3):567–586.

Alhaqbani, B. and Fidge, C. (2008). Access control requirements for processing electronic health records. In Hofstede, A., Benatallah, B., and Paik, H.-Y., editors, *Business Process Management Workshops*, volume 4928 of *Lecture Notes in Computer Science*, pages 371–382. Springer Berlin Heidelberg.

Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. (2013). The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404.

Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., and Verbauwhede, I. (2011). SPONGENT: A lightweight hash function. In Preneel, B. and Takagi, T., editors, *13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '11)*, volume 6917 of *LNCS*, pages 312–325, Nara, Japan.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C. (2007). PRESENT: An ultra-lightweight block cipher. In Paillier, P. and Verbauwhede, I., editors, *9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2007)*, volume 4727 of *LNCS*, pages 450–466, Vienna, Austria.

Bogdanov, A., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., and Seurin, Y. (2008). Hash functions and RFID tags: Mind the gap. In Oswald, E. and Rohatgi, P., editors, *10th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2008)*, volume 5154 of *LNCS*, pages 283–299, Washington, D.C., USA.

Boneh, D. and Franklin, M. (2003). Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615. Also appeared in CRYPTO '01.

Buchegger, S., Tissieres, C., and Le Boudec, J.-Y. (2004). A test-bed for misbehavior detection in mobile ad-hoc networks, how much can watchdogs really do? In *6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '04)*, pages 102–111, Low Wood, Lake Windermere, UK.

Byoungyoung, L., Jinoh, O., Hwanjo, Y., and Jong, K. (2011). Protecting location privacy using location semantics. In *17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pages 1289–1297, San Diego, California, USA.

Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and Arkko, J. (2003). Diameter base protocol. RFC 3588, IETF.

Carl, G., Kesidis, G., Brooks, R. R., and Rai, S. (2006). Denial-of-service attack detection techniques. *IEEE Internet Computing*, 10(1):82–89.

Chen, C.-Y. and Chao, H.-C. (2011). A survey of key distribution in wireless sensor networks. *Security and Communication Networks*.

Damandeep, K. and Jyotsna, S. (2012). Proposed p2p trust and reputation based model to secure grid. *IJCA Proceedings on International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT 2012)*, iRAFIT(2):19–24.

Daněk, M., Kadlec, J., Bartosinski, R., and Kohout, L. (2008). Increasing the level of abstraction in FPGA-based designs. In Udo, K., editor, *International Conference on Field Programmable Logic and Applications (FPL 2008)*, pages 5–10, Heidelberg, Germany.

Doomun, M. R. and Soyjaudah, K. M. S. (2009). Analytical comparison of cryptographic techniques for resource-constrained wireless security. *International Journal of Network Security*, 9(1):82–94.

Doyle, B., Bell, S., Smeaton, A. F., McCusker, K., and O'Connor, N. (2006). Security considerations and key negotiation techniques for power constrained sensor networks. *The Computer Journal*, 49(4):443–453.

ECRYPT (2008). The eSTREAM project. Available online at: http://www.ecrypt.eu.org/stream/.

Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., and Uhsadel, L. (2007). A survey of lightweight-cryptography implementations. *IEEE Design & Test*, 24(6).

Engels, D., Saarinen, M.-J. O., Schweitzer, P., and Smith, E. M. (2011). The hummingbird-2 lightweight authenticated encryption algorithm. In Juels, A. and Paar, C., editors, *7th Workshop of RFID Security and Privacy (RFIDSec '11)*, volume 7055 of *LNCS*, pages 19–31, Amherst, Massachusetts, USA.

Feldhofer, M., Wolkerstorfer, J., and Rijmen, V. (2005). AES implementation on a grain of sand. *IEE Proceedings on Information Security*, 152(1):13–20.

Fysarakis, K., Manifavas, C., Papaefstathiou, I., and Adamopoulos, A. (2013). A lightweight anonymity & location privacy service. In *IEEE Int. Symposium on Signal Processing and Information Technology (ISSPIT 2013)*, Athens, Greece. To appear.

Gaj, K., Homsirikamol, E., Rogawski, M., Shahid, R., and Sharif, M. U. (2012). Comprehensive evaluation of high-speed and medium-speed implementations of five SHA-3 finalists using Xilinx and Altera FPGAs. Cryptology ePrint Archive, Report 2012/368.

Gedik, B. and Liu, L. (2008). Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18.

Golle, P. and Partridge, K. (2009). On the anonymity of home/work location pairs. In Tokuda, H., Beigl, M., Friday, A., Brush, A. J. B., and Tobe, Y., editors, *7th International Conference on Pervasive Computing (Pervasive 2009)*, volume 5538 of *LNCS*, pages 390–397, Nara, Japan.

Gruteser, M. and Grunwald, D. (2005). Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis. *Mobile Networks and Applications*, 10(3):315–325.

Gruteser, M. and Hoh, B. (2005). On the anonymity of periodic location samples. In Hutter, D. and Ullmann, M., editors, *2nd International Conference on Security* *in Pervasive Computing (SPC 2005)*, volume 3450 of *LNCS*, pages 179–192, Boppard, Germany.

Guo, J., Peyrin, T., and Poschmann, A. (2011). The PHOTON family of lightweight hash functions. In Rogaway, P., editor, *31st annual conference on Advances in cryptology (CRYPTO '11)*, volume 6841 of *LNCS*, Santa Barbara, CA, USA.

Gupta, K., Yadav, A. S., and Yadav, S. (2011). Location privacy using user anonymity and dummy locations. *International Journal of Innovative Technology & Creative Engineering*, 1(10):5–8.

Hatzivasilis, G. and Manifavas, C. (2012). Building trust in ad hoc distributed resource-sharing networks using reputation-based systems. In *16th Panhellenic Conference on Informatics*, pages 416–421, Piraeus, Greece.

Hatzivasilis, G., Theodoridis, A., Gasparis, H., Manifavas, C., and Papaefstathiou, I. (2014). ULCL: An ultra-lightweight cryptographic library for embedded systems. In *Measurable security for Embedded Computing and Communication Systems (MeSeCCS 2014)*, Lisbon, Portugal. To appear.

Hoh, B. and Gruteser, M. (2005). Protecting location privacy through path confusion. In *1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05)*, pages 194–205, Athens, Greece.

Huang, J., Buckingham, J., and Han, R. (2005a). A level key infrastructure for secure and efficient group communication in wireless sensor networks. In *1st IEEE/CreateNet Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm 2005)*, pages 249–260, Athens, Greece.

Huang, L., Matsuura, K., Yamane, H., and Sezaki, K. (2005b). Enhancing wireless location privacy using silent period. In *IEEE Wireless Communications and Networking Conference (WCNC '05)*, volume 2, New Orleans, LA, USA.

Hui, J. and Thubert, P. (2011). Compression format for IPv6 datagrams over IEEE 802.15.4-based networks. RFC 6282, IETF.

IEEE Standard for Local and metropolitan area networks (2011). IEEE standard for local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). Available online at: http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf.

Kamal, A. A. and Youssef, A. M. (2009). An FPGA implementation of the NTRUEncrypt cryptosystem. In *International Conference on Microelectronics (ICM)*, pages 209–212.

Karakoç, F., Demirci, H., and Harmanci, A. E. (2013). ITUbee: A software oriented lightweight block cipher. In *2nd International workshop on lightweight cryptography for security & privacy (LightSEC 2013)*.

Karalis, A., Joannopoulos, J. D., and Soljačić, M. (2008). Efficient wireless non-radiative mid-range energy transfer. *Annals of Physics*, 323(1):34–48.

Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., and Savage, S. (2010). Experimental security analysis of a modern automobile. In *IEEE Symposium on Security and Privacy (SP)*, pages 447–462, Oakland, CA, USA.

Kothmayr, T., Schmitt, C., Hu, W., Bruenig, M., and Carle, G. (2012). A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication. In *IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops)*, pages 956–963, Clearwater, FL, USA.

Kuo, C., Luk, M., Negi, R., and Perrig, A. (2007). Message-in-a-bottle: User-friendly and secure key deployment for sensor nodes. In *5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pages 233–246, Sydney, Australia.

Kurs, A., Karalis, A., Moffatt, R., Joannopoulos, J. D., Fisher, P., and Soljačić, M. (2007). Wireless power transfer via strongly coupled magnetic resonances. *Science*, 317(5834):83–86.

Liu, L. (2007). From data privacy to location privacy: Models and algorithms. In *33rd international conference on Very large data bases (VLDB '07)*, pages 1429–1430, Vienna, Austria. VLDB Endowment.

Liu, L. (2009). Privacy and location anonymization in location-based services. *SIGSPATIAL Special*, 1(2):15–22.

Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1). Article No. 3.

Madhavi, S. and Kim, T. H. (2011). An intelligent distributed reputation based mobile intrusion detection system. *International Journal of Computer Science and Telecommunications*, 2(7):53–58.

Manifavas, C., Hatzivasilis, G., Fysarakis, K., and Rantos, K. (2013). Lightweight cryptography for embedded systems – A comparative analysis. In *6th Internations Workshop on Autonomous and Spontaneous Security (SETOP 2013)*, volume 8247 of *Lecture Notes in Computer Science*, pages 371–382. Springer-Verlag Berlin Heidelberg, RHUL, Egham, U.K.

Martin, K. M. and Paterson, M. B. (2008). An application-oriented framework for wireless sensor network key establishment. *Electronic Notes in Theoretical Computer Science*, 192(2):31–41.

Naedele, M. (2006). An access control protocol for embedded devices. In *4th International IEEE Conference on Industrial Informatics (INDIN '06)*, pages 565–569, Singapore.

Oliveira, L. B., Aranha, D. F., Gouvêa, C. P. L., Scott, M., Câmara, D. F., López, J., and Dahab, R. (2011). TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. *Computer Communications*, 34(3):485–493.

Preneel, B. (2009). Research challenges in lightweight cryptography. Key Note Talk at the 2nd ACM conference on Wireless network security (WiSec '09).

Rantos, K., Papanikolaou, A., Fysarakis, K., and Manifavas, C. (2012). Secure policy-based management solutions in heterogeneous embedded systems networks.

In *IEEE International conference on Telecomunications and Multimedia (TEMU 2012)*, pages 227–232, Heraklion, Crete, Grece.

Rantos, K., Papanikolaou, A., and Manifavas, C. (2013a). IPsec over IEEE 802.15.4 for low power and lossy networks. In *11th ACM International Symposium on Mobility Management and Wireless Access (MobiWac 2013)*, pages 59–64, Barcelona, Spain.

Rantos, K., Papanikolaou, A., Manifavas, C., and Papaefstathiou, I. (2013b). IPv6 security for low power and lossy networks. In *Wireless Days 2013 (WD 2013)*, Valencia, Spain. To appear.

Raymond, D. R. and Midkiff, S. F. (2008). Denial-of-service in wireless sensor networks: Attacks and defenses. *IEEE Pervasive Computing*, 7(1):74–81.

Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T., and Roedig, U. (2011). Securing communication in 6LoWPAN with compressed IPsec. In *7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '11)*, pages 1–8, Barcelona, Spain.

Resnick, P. and Zeckhauser, R. (2002). Trust among strangers in internet transactions: Empirical analysis of eBay's reputation system. In Baye, M. R., editor, *The Economics of the Internet and E-commerce*, volume 11 of *Advances in Applied Microeconomics*, pages 127–157. Emerald Group Publishing Limited.

Shibutani, K., Isobe, T., Hiwarati, H., Mitsuda, A., Akishita, T., and Shirai, T. (2011). Piccolo: An ultra-lightweight blockcipher. In Preneel, B. and Takagi, T., editors, *13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '11)*, volume 6917 of *LNCS*, pages 342–357, Nara, Japan.

Simplicio, Jr., M. A., Barreto, P. S. L. M., Margi, C. B., and Carvalho, T. C. M. B. (2010). A survey on key management mechanisms for distributed wireless sensor networks. *Computer Networks*, 54(15):2591–2612.

Smith, R. (2008). Phlashdance, discovering permanent denial of service attacks against embedded systems. Talk at the EUSecWest Applied Security Conference. London.

Stefanidis, K. and Serpanos, D. N. (2008). Implementing filtering and traceback mechanism for packet-marking IP-based traceback schemes against DDoS attacks. In *4th International IEEE Conference on Intelligent Systems (IS '08)*, volume 2, pages 14–28, Varna, Bulgaria.

Szilagyi, C. and Koopman, P. (2009). Flexible multicast authentication for time-triggered embedded control network applications. In *IEEE/IFIP International Conference on Dependable Systems & Networks (DSN '09)*, pages 165–174, Lisbon, Portugal.

Szilagyi, C. and Koopman, P. (2010). Low cost multicast authentication via validity voting in time-triggered embedded control networks. In *5th Workshop on Embedded Systems Security (WESS '10)*, pages 10:1–10:10, Scottsdale, Arizona.

Trusted Platform Module (2009). Information technology – Trusted Platform Module – Part 1: Overview. ISO/IEC 11889-1:2009. Available online at: http://www.trustedcomputinggroup.org/resources/tpm_main_specification.

Watanabe, D., Ideguchi, K., Kitahara, J., Muto, K., and Furuichi, H. (2008). Enocoro-80: A hardware oriented stream cipher. In *3rd International Conference on Availability, Reliability and Security (ARES 08)*, pages 1294–1300.

Yoshihama, S., Ebringer, T., Nakamura, M., Munetoh, S., Mishina, T., and Maruyama, H. (2007). WS-attestation: Enabling trusted computing on web services. In Baresi, L. and Di Nitto, E., editors, *Test and Analysis of Web Services*, pages 441–469. Springer Berlin Heidelberg.

Zeeb, E., Moritz, G., Timmermann, D., and Golatowski, F. (2010). WS4D: Toolkits for networked embedded systems based on the devices profile for web services. In *39th International Conference on Parallel Processing Workshops (ICPPW '10)*, pages 1–8, San Diego, CA, USA.

Zhang, Y., Xu, L., and Wang, X. (2008). A cooperative secure routing protocol based on reputation system for ad hoc networks. *Journal of Communications*, 3(6):43–50.

Zhong, G. and Hengartner, U. (2008). Toward a distributed k-anonymity protocol for location privacy. In *7th ACM workshop on Privacy in the electronic society (WPES '08)*, pages 33–38, Alexandria, VA, USA.