Why is IPSec a viable option for Wireless Sensor Networks

Jorge Granjal Ricardo Silva Edmundo Monteiro Jorge Sá Silva Fernando Boavida
jgranjal@dei.uc.pt rnsilva@dei.uc.pt edmundo@dei.uc.pt sasilva@dei.uc.pt boavida@dei.uc.pt
University of Coimbra, Portugal

Abstract

Many issues still remain to be addressed in order to achieve acceptable security in Wireless Sensor Networks (WSNs). This necessity, together with the adoption of IPv6 on WSNs being defined at the 6lowpan working group of the IETF, motivates our investigation on the feasibility of the application of IPSec on sensor nodes. IPSec is already part of IPv6, which makes it a natural candidate to be directly employed or adapted for WSNs. We discuss results obtained from practical experiments on the usage of cryptographic algorithms typically employed within IPSec in real sensor nodes. We analyze the security and performance tradeoffs involved when employing cryptography measures in WSNs, also in the context of their usage side-by-side with IPv6. The results obtained show that the adoption of a security architecture such as IPSec is viable, and also point towards the successful design and deployment of a security architecture for WSNs.

Keywords – wireless sensor networks, security, IPSec, 6lowpan.

1. Introduction

WSNs may interact with sensitive data and operate in hostile and unattended environments [1], and present themselves as one of the most promising technologies for the near future. Although many of the application scenarios planned for WSNs are critical in respect to security (i.e. in the areas of healthcare, military, surveillance, and physical infrastructure monitoring), many open issues still remain to be addressed.

We start by addressing the current status of security on WSNs. This discussion allows us to contextualize the purpose of our practical tests and the remaining discussion throughout the paper. We then proceed to discuss the results obtained from practical experiments conducted on the usage of cryptographic algorithms on WSNs. The results allow us to rationalize on the tradeoffs between security and the required energy and computational capability, and also on the feasibility of applying IPSec on WSNs with IPv6 support.

2. Current status of security on WSNs

Although many security protocols and algorithms have been proposed for traditional wireless and ad hoc networks, many of them are not well suited to WSNs. It is therefore important to analyze the feasibility of applying security mechanisms or architectures designed for other contexts to WSNs. New security models and mechanisms that are tailored to WSNs may also need to be designed.

Several threats can be targeted against WSNs, as we proceed to discuss. At the physical layer, WSNs are vulnerable to jamming, tampering and traffic analysis. At the link layer attacks usually exploit vulnerabilities of the MAC protocol, by allowing the insertion of collisions, energy exhaustion, and unfairness on the usage of the communications medium [2]. IEEE 802.15.4 [3] has been designed as a link layer security architecture, which provides security mechanisms that guarantee confidentiality, integrity, access control, and replay protection. At the routing layer, we verify that many of the routing protocols already proposed for WSNs weren't designed with security in mind. Several attacks on routing protocols for WSNs are described in [5], such as black holes, wormholes, spoofing, selective forwarding, sinkholes, hello floods, and acknowledgment attacks. At the transport layer we find that the main attacks targeted against transport protocols are the flooding and the desynchronization attacks. Several solutions have been proposed against these threats. Possible approaches to address security at the physical layer include mapping and isolation of jammed regions [6], software and hardware attestation techniques, algorithmic solutions implementing resilience by redundancy, replication of state among sensor nodes, majority voting techniques, probabilistic routing schemes, and fake messages. To address security at the link layer proposed solutions include error correcting codes [2], time division multiplexing, limit rate request, and small frames [2]. At the network and routing layer, proposed solutions include packet

1-4244-2575-4/08/$20.00 © 2008 IEEE

leashes [9], multi-path routing techniques, and verification of the bi-directionality of the link. Possible approaches to achieve security at the transport layer include client puzzles [2] and packet authentication.

In general we realize that most of the existing security proposals for WSNs focus on copying or adapting security techniques already used on the Internet. For many of the proposals it is also currently unclear if they can be effectively implemented in WSNs, and for many of the described attacks there is no complete solution currently available. While several research proposals do present new ideas, they generally lack an analysis on the feasibility of their application to real WSNs scenarios. It is also important to note that the appropriate security mechanisms will frequently be determined by factors such as energy constraints, or the level of control over the topology of the network. The level of energy available dictates the type of cryptographic solutions that can be employed, together with their usage methodology (i.e. types of cryptographic algorithms, together with their key and block sizes). The level of control over the topology during deployment of sensor nodes significantly determines and facilitates security boot-up procedures and operations. If the topology can be planned beforehand, we are able to securely employ pre-distribution of cryptographic keys. Also, the detection of attacks on nodes or on data aggregation and routing mechanisms is also more facilitated in this situation, since the specific location of each sensor node is known in advance.

One fundamental problem is the inexistence of a security architecture for WSNs. Such architecture should be designed taking into consideration knowledge obtained from research on the application of specific security mechanisms to real sensor nodes, as this allows the selection and design of the most appropriate cryptographic algorithms and security solutions to specific situations and usage requirements.

3. Experimental analysis of cryptographic algorithms on WSNs

3.1. Related work

The majority of the existing studies concerning the feasibility of application of cryptographic algorithms on sensor nodes obtain their results from simulation works. Many of them use PowerTOSSIM [11] to simulate the energy consumption of the cryptographic algorithms. For example, in [12] the creation of energy maps using statistical models is discussed, while in [13] the authors discuss reduction of energy consumption due to the usage of more efficient algorithms. We present results from experimental measurements conducted with the dual purpose of investigating the complete resource consumption due to the computational requirements of several cryptographic algorithms when applied to sensor nodes, and of analyzing their feasibility of application in the context of the usage of IPSec and IPv6 on WSNs. As far as we know, this is the first experimental study with these goals in mind. Other experimental studies do exist, but they focus on measuring the overhead introduced by cryptography on the radio transmission of data, or on testing cryptographic algorithms on sensor nodes only as a proof of concept study [14][15][16][17]. These studies completely lack practical applicability criteria when choosing the algorithms to test and the metrics used.

3.2. Experimental scenario

Our tests were conducted employing two MicaZ [18] motes. The MicaZ is powered by an ATMEL ATmega 128L 8-bit microprocessor running at 16 MHz. It has 128k bytes of EEPROM memory to store executable code and 4k bytes of internal SRAM memory for temporary storage during program execution. Communications run at 2.4 GHz and data can be transmitted at the rate of 250kbps. We consider the MicaZ to be representative of the computational power currently available on commercial sensor nodes, as the ATmega 128L processor is close to the class of microcontrollers considered as supporting heavy-duty applications [19]. The algorithms tested are typically used as security primitives within IPSec, and are described in Table 1.

Table 1. Algorithms tested

Algorithm	Type	Key / Hash	Block Size
AES	Block	192 bits	128 bits
3DES	Block	192 bits	64 bits
SHA1	Hash	160 bits	512 bits
SHA2	Hash	512 bits	512 bits

Each algorithm was implemented in a simple NesC application, used for running the algorithm with different plaintext sizes. An adaptation layer for the transmission of IPv6 packets over IEEE 802.15.4 has already been specified by the 6lowpan working group of the IETF. Each application was compiled with and without IPv6 support, using the 6lowpan implementation already available with TinyOS 2.0. Resource consumption was measured in two phases. First, the memory requirements for each algorithm in terms of EEPROM (program memory) and SRAM (temporary execution memory) with and without IPv6

support was measured. These values are given at the end of the compilation phase for each version of the test program by the *avr-gcc* compiler. Next the execution time and energy consumption for each algorithm was measured.

Execution time was measured by having the program send to the base station a packet with the current system time before and after the execution of each algorithm. The subtraction of the two values allowed us to compute (in milliseconds) the processing time for each execution of the algorithm. The energy consumption was measured by having the program executing the algorithm and sending to the base station the current energy level of the batteries, at each 5 seconds. This allows us to measure energy consumption in a heavy-duty scenario, therefore obtaining values that can be considered as majoring the energy consumption values of real scenarios employing IPSec. The same measurement procedure was also employed without executing any cryptographic code at all, in order to obtain a baseline in terms of energy consumption. In order to circumvent problems already identified while measuring battery consumption on MicaZ motes [10], each algorithm was executed during one hour, and measurements were only considered valid after battery consumption stabilization has been clearly identified. The energy consumption values have been converted to mV/h from the values in hexadecimal returned by the ADC on the MicaZ.

3.3. Tests results

In Table 2 we present the memory usage values obtained for the tested algorithms. For each algorithm we measure memory usage with and without IPv6 support. We also present the memory usage of a program without any cryptographic code. This allows us to draw a baseline corresponding to the memory usage due to the TinyOS code alone, also with and without IPv6 support.

Table 2. Memory usage (bytes) of the tested algorithms

Algorithm	EEPROM	SRAM	EEPROM (6lowpan)	SRAM (6lowpan)
AES	14138	1661	23404	3743
3DES	41938	2491	33930	4552
SHA1	13540	357	22726	2471
SHA2	26868	1053	35976	3135
Baseline	11708	285	26996	2677

From our attempts to run the cryptographic algorithms with 6lowpan support, we were only able to normally execute the test program for SHA1 (the lighter of the two signature algorithms). We were unable to execute neither SHA2 nor any of the two encryption algorithms. Although all the algorithms fit in the available 128k bytes of EEPROM memory, the 3DES, AES and SHA2 algorithms require higher values of SRAM. 3DES is above the 4k bytes limit, causing the mote not to boot at all. As for AES and SHA2, although they stay below that limit, the small value of free memory during execution causes the algorithm to stop after a few operations. The SRAM memory available for SHA2 (the less demanding of the three in respect to SRAM usage) represents 77% (3135 bytes) of the total available memory, and an increase of 17% from the baseline code with 6lowpan support. This leaves not enough room for normal processing, as the code of 6lowpan is using a very significant amount of space from the available SRAM. On the other hand, SHA1 uses only around 60% (2471 bytes). We are therefore able to conclude that the amount of SRAM memory available for program execution is the critical factor for the successful implementation of extra security mechanisms above IPv6 (including the TinyOS code) on the MicaZ. The current implementation of 6lowpan, plus the code for TinyOS, leaves not enough SRAM for extra cryptographic mechanisms (SHA1 being the only exception). The same limitation will certainly remain if we add extra code for the support of other fundamental IPSec operations (i.e. setup and maintenance of security associations, and key negotiation). To add to this conclusion, we note that the currently available implementation of 6lowpan for TinyOS is not yet complete.

All the algorithms executed without problems without 6lowpan support. The most demanding of the two encryption algorithms in respect to memory consumption is 3DES. When comparing to the baseline code, 3DES uses an additional 53% of the available SRAM comparing to an additional 34% by AES, and an additional 23% of the available EEPROM comparing to an additional of only 1.8% by AES. The higher memory requirement of 3DES is due to the large data tables (S-boxes) used by the algorithm. In respect to memory usage requirements, AES is clearly the best choice for an encryption algorithm to be used with IPSec. SHA2 is clearly the most demanding of the two signature algorithms in respect to memory consumption. When comparing to the baseline code, SHA2 uses an additional 19% of the available SRAM comparing to an additional of only 1.7% for SHA1, and an additional 12% of the available EEPROM space comparing to an additional of only 1.4% for SHA1. In respect to memory usage requirements, SHA1 is clearly the best choice for a signature algorithm to be used with IPSec. In Figures 1 and 2 we present the

results obtained from the measurements on the processing times and energy consumption.

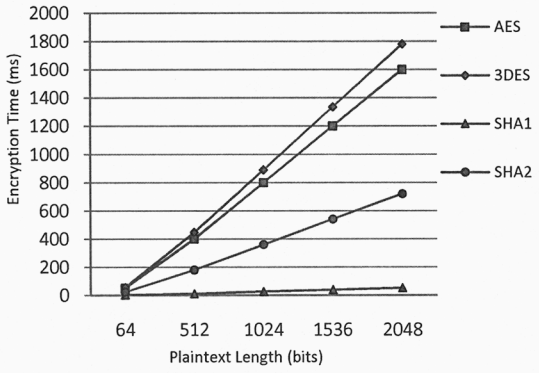

Figure 1. Encryption times (ms)

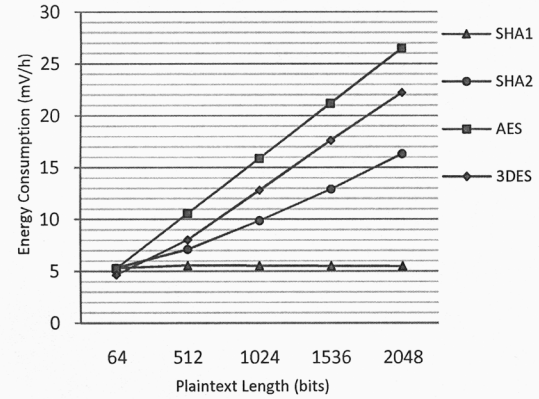

Figure 2. Energy consumption (mV/h)

As expected, encryption time and energy consumption increase linearly with plaintext length. The values measured for the energy are consistent with the baseline value. This value was obtained by measuring the energy level of an idle sensor node that only reports its energy level at every 5 seconds. The measurements were obtained during a period of 4 hours, and the medium value was calculated to be very close to 5mV/h. SHA1 is clearly the most efficient of the two signature algorithms, with only 52 ms to encrypt 2048 bits comparing to 720 ms by SHA2. SHA1 also spends a low and almost constant value of energy for different plaintext sizes, between 5.29 mV/h for 128 bits and 5.47 mV/h for 2048 bits plaintext (5.8% and 9.4% of the energy baseline value respectively). SHA2 consumes 16.3 mV/h to encrypt 2048 bits of plaintext (3.26 times the energy baseline value) although only 5.3 mV/h for 128 bits. The later value is very similar to the obtained for SHA1.

Concerning the encryption algorithms, we verified that 3DES needs 1780ms to encrypt 2048 bits of plaintext comparing to 1600 for AES. When considering plaintexts with lower lengths the two algorithms behave similarly. AES needs 119 ms to encrypt 128 bits of plaintext comparing to 121 bits by 3DES. Similar observations can be derived by observing their energy consumption. AES spends 26.4 mV/h to encrypt 2048 bits of plaintext (5.28 times the energy baseline value), comparing to 22.2 by 3DES (4.44 times the energy baseline value). Considering smaller plaintext data, AES consumes 5.42 mV/h to encrypt 128 bits of plaintext (8% of the energy baseline value), and a very similar value is obtained for 3DES.

4. IPSec and IPv6 on WSNs

The adoption of IPv6 on WSNs is currently being defined in the 6lowpan working group of the internet area of IETF. This group works on adapting IPV6 to low power WPANs employing IEEE 802.15.4. RFC 4919 [8] discusses assumptions, problem statement and goals for 6lowpan, while the adaptation layer for the transmission of IPv6 packets over IEEE 802.15.4 is described in RFC 4944 [7]. The usage of security mechanisms in the context of 6lowpan is currently mostly undefined [4]. Therefore, the feasibility of employing IPSec (a mandatory part of IPv6) with IPv6 on WSNs is currently an open issue. With this problem in mind, we focus our experiments on measuring the resource consumption due to the processing requirements of some of the cryptographic algorithms typically employed within IPSec.

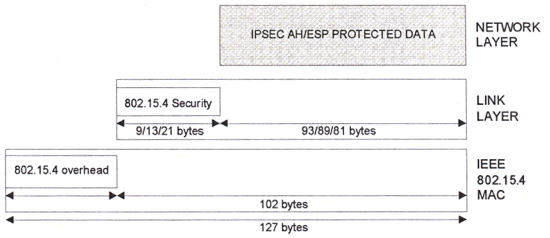

Figure 3. IPsec in the context of 6lowpan

Figure 3 illustrates the available transmission payload for IPSec at the network layer. The packet format and size at the MAC level is defined by IEEE 802.15.4. Four link-layer security modes are available in IEEE 802.15.4. The payload space at the network layer is 81 bytes for the AES-CCM-128 mode (which uses 21 bytes for security), 89 bytes for AES-CCM-64 (which uses 13 bytes), and 93 bytes for AES-CCM-32 (which uses 9 bytes). Without link-layer security the available payload space is 102 bytes. The ability of

efficiently applying cryptographic algorithms to a payload with size between 81 and 102 bytes can therefore be considered a basic requirement for the successful implementation of IPSec on real sensor hardware. Considering this requirement, we present in Table 3 the values of energy and processing time required to encrypt 128-bytes of plaintext. In the same table we also present an estimated value for the energy life time of the MicaZ when running each algorithm.

Table 3. Processing time and estimated life time values

Algorithm	Measured energy (mV/h)	Estimated node life time (hours)	Measured processing time (ms)
AES	15,87	189,04	800
3DES	12,8	234,37	890
SHA1	5,51	544,46	26
SHA2	9,86	304,26	360
Baseline	5	600	-

The calculus of the estimated values takes into consideration the baseline energy value (measured experimentally as 5mV/h), and also that two new batteries (the MicaZ uses two 1.5v AA-type batteries) provide approximately 3000mV of available energy. This is a theoretical value, as the real value varies for each pair of new batteries, but nevertheless we consider it to be a good base value for our purpose of comparing the estimated life time values.

We observe that both SHA1 and SHA2 require relatively low processing times to encrypt 128 bytes of data, and as such we consider both algorithms to represent viable choices regarding their application in IPSec with IPv6 at the network layer. SHA1 represents clearly the best choice, as it needs much less memory and is significantly faster. Considering the tested encryption algorithms, although they are demanding in terms of energy and processing time we consider that they constitute viable options regarding their usage within IPSec. AES needs less memory and is faster, while 3DES in better in respect to energy consumption. The estimated values for the life time of a sensor node shows us that the employment of the tested algorithms on the MicaZ is viable, although we must always seek an acceptable compromise between security and the life time (energy consumption) of the sensor node. This observation also adds to our conclusion concerning the feasibility of applying IPSec on WSNs.

The possibility of applying IPSec to WSNs brings the advantage of introducing security at the network layer, thus bringing authentication and encryption transparently to higher layers protocols and solving many of the security issues previously discussed. We are able to foresee two scenarios for the usage of IPSec on WSNs: end-to-end security with IPSec in transport mode, or employing a security gateway between the WSN and the Internet using IPSec in tunnel mode, as we illustrate in Figure 4.

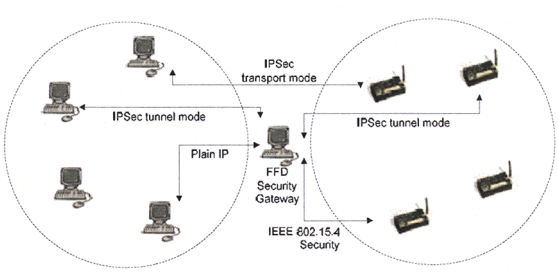

Figure 4. Usage scenarios of IPSec on WSNs

Security at the network layer can be achieved directly between a node on the Internet and a node on the WSN, employing IPSec in transport mode. A security gateway can also be used, allowing a node on the Internet to connect indirectly to nodes on the WSN. A node on the internet can connect to the security gateway using IPSec in tunnel mode (or without security), while security on the WSN may be achieved using IPSec in tunnel mode to connect to a sensor node or simply by using the security mechanisms available on IEEE802.15.4. It is currently undefined what security mechanisms can be employed above the network layer on WSNs using IPv6. Therefore, IPSec would bring the advantage of introducing security mechanisms that transport and application layers can beneficiate from. The security gateway can handle IPSec, TLS/SSL, or other IP traffic from the Internet side and enforce the usage of IPSec on the WSN. IPSec also presents the added benefit of being easy to integrate with new automatic key management mechanisms designed for WSNs.

5. Conclusions

As many issues still remain to be addressed in respect to achieving security on WSNs, there exists a fundamental necessity of obtaining more knowledge from the application of security mechanisms on real sensor hardware. The definition of a security architecture for WSNs is also a necessity. With this in mind, we conducted tests on the application of several well known cryptographic algorithms to real sensor nodes, with the purpose of obtaining knowledge and results towards the identification of security mechanisms that can be applied to WSNs and that are good candidates to be part of a security architecture designed for WSNs. Our results showed that currently the main limitation of the tested sensor nodes is on the

available memory (particularly temporary storage). The computational and energetic demands introduced by cryptography, although significant, do not compromise the applicability of security solutions such as IPSec on sensor nodes. We thus consider the future implementation of IPSec with IPv6 on WSNs to be viable, particularly as new sensor nodes become available with more storage space and computational capabilities. On future work we plan to conduct tests using other cryptographic algorithms, such as public-key algorithms (RSA or elliptic curve variants) and session key negotiation algorithms (i.e. Diffie-Hellman). This will allow us to identify further security mechanisms that can be successfully employed on WSNs, and also gain more knowledge towards the definition of a complete security architecture for WSNs.

6. References

[1] J.P. Walters et al., "Wireless Sensor Network Security: A Survey," Security in Distributed, Grid, and Pervasive Computing, Auerbach Publications, CRC Press, pp. 1-49; http://www.cs.wayne.edu/~weisong/papers/walters05-wsn-security-survey.pdf.

[2] A.D. Wood and J.A. Stankovic, "Denial of Service in Sensor Networks," *Computer*, vol. 35, 2002, pp. 54-62.

[3] Wireless medium access control and physical layer specifications for low-rate wireless personal networks, Standard 802.15.4, IEEE, 2003.

[4] S. Daniel Park et al., "draft-daniel-6lowpan-security-analysis-02.txt - IPv6 over Low Power WPAN Security Analysis," Feb. 2008; http://tools.ietf.org/id/draft-daniel-6lowpan-security-analysis-02.txt.

[5] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, vol. 1, Sep. 2003, p. 293—315.

[6] A. Wood, J. Stankovic, and S. Son, "JAM: a jammed-area mapping service for sensor networks," *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*, 2003, pp. 286-297.

[7] G. Montenegro et al., "RFC 4944 - Transmission of IPv6 Packets over IEEE 802.15.4 Networks," Sep. 2007; http://tools.ietf.org/html/rfc4944.

[8] N. Kushalnagar, G. Montenegro, and C. Montenegro, "RFC 4919 - IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," Aug. 2007; http://tools.ietf.org/html/rfc4919.

[9] Yih-Chun Hu, A. Perrig, and D. Johnson, "Wormhole attacks in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, 2006, pp. 370-380.

[10] M. Kramer and A. Gerldy, "Energy Measurements for MicaZ Node," Fachgesprach " 2006.

[11] V. Shnayder et al., "Simulating the power consumption of large-scale sensor network applications," *Proceedings of the 2nd international conference on Embedded networked sensor systems*, Baltimore, MD, USA: ACM, 2004, pp. 188-200; http://portal.acm.org/citation.cfm?id=1031495.1031518.

[12] L. Zhong, J. Rabaey, and A. Wolisz, "An integrated data-link energy model for wireless sensor networks," *Communications, 2004 IEEE International Conference on*, 2004, pp. 3777-3783 Vol.7.

[13] R. Mini, B. Nath, and A. Loureiro, *Prediction-based Approaches to Construct the Energy Map for Wireless Sensor Networks*, 2003; http://citeseer.ist.psu.edu/mini03predictionbased.html.

[14] G. Guimaraes et al., "Evaluation of Security Mechanisms in Wireless Sensor Networks," *Proceedings of the 2005 Systems Communications*, IEEE Computer Society, 2005, pp. 428-433; http://portal.acm.org/citation.cfm?id=1090953.1091991&coll=GUIDE&dl=GUIDE&CFID=9464521&CFTOKEN=23122276.

[15] M. Passing and F. Dressler, "Experimental Performance Evaluation of Cryptographic Algorithms on Sensor Nodes," *3rd IEEE International Conference on Mobile Ad Hoc and Sensor Systems (IEEE MASS 2006): 2nd IEEE International Workshop on Wireless and Sensor Networks Security (WSNS'06)*, Vancouver, Canada: 2006, pp. 882-887; http://http://www7.informatik.uni-erlangen.de/dressler/publications/wsns2006.pdf.

[16] "BTnodes - A Distributed Environment for Prototyping Ad Hoc Networks : Main - Overview browse"; http://www.btnode.ethz.ch/.

[17] Kyung Jun Choi and Jong-In Song, "Investigation of feasible cryptographic algorithms for wireless sensor network," *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, 2006, pp. 3.

[18] "MICAz Wireless Measurement System"; http://www.xbow.com/products/product_pdf_files/wireless_pdf/6020-0060-01_a_micaz.pdf.

[19] R. Roman, C. Alcaraz, and J. Lopez, "A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes," *Mob. Netw. Appl.*, vol. 12, 2007, pp. 231-244.