# Data Security in Wireless Sensor Networks – the Task for Public Key Cryptography

*Michal VARCHOLA*

Dept. of Electronics and Multimedia Communications,
FEI TU of Košice, Slovak Republic

Miso.Varchola@tuke.sk

***Abstract*—The paper describes security algorithms and protocols provided by recent WSN stacks where symmetric-key schemes are commonly used. Using these schemes seems to be impractical in large scale networks; hence the paper intends their replacement by public-key schemes for mentioned low cost and low power MCU platforms. Moreover, the paper proposes implementation of Wireless Sensor Network (WSN) stacks embedded in common microcontroller (MCU) platforms – ARM7TDMI, x51, ColdFire, and HCS08 in order to compare it. Protocol stacks of proposed WSNs are based on the both – a proprietary and the ZigBee.**

***Keywords*—Elliptic Curve Cryptography, IEEE 802.15.4 Standard, Public Key Cryptography, Sensor Networks, ZigBee**

## I. Introduction

Wireless Sensor Networks (WSNs) have received considerable attention during last decade [1], [2], [3]. WSNs can be applied to a large number of areas, and its applications are continuously growing. They are expected to be used in a wide range of applications, from number military to various civil. Intentions of military are in target sensing or tracking in battlefields [4] or detection of biological or chemical weapons, or sensor nodes also could be deployed into enemy territory to observe it. WSNs penetrate into civil areas as well – biomedical, healthcare, building or home automation and environment from wildlife monitoring, early fire detection in forests or collecting microclimate data [5], [6], [7] to outdoor deployments of sensor networks to monitor storms, oceans, and weather events.

Paul Saffo from Institute for the Future says: [8] "Just as the personal computer was a symbol of the '80s, and the symbol of the '90s is the World Wide Web, the next nonlinear shift, is going to be the advent of cheap sensors." Let's add: secure sensors, because of besides the battlefield applications, security is critical in healthcare systems at hospitals or Home Automation (HA) too. WSNs are, in general, more vulnerable to attacks and unauthorized access than traditional (wired) networks. For example, an adversary can easily listen to the traffic and mislead communications between nodes.

WSNs are typically characterized by limited power supplies, low bandwidth, small memory sizes and limited energy. This leads to a very demanding environment to provide security. Because of that special characteristics and limitations of wireless sensor networks, desingers face an important challenge in security issue, particularly for the applications where WSNs are developed for use in a hostile environment or used for some crucial purposes. In order to establish a secure network, it is necessary to design secure protocols to deal with problems about key agreement and encryption in communications.

Many applications in the area of WSN would gain a lot from the availability of strong public-key cryptography (PKC). Recently a number of studies have been conducted to find out a practical way to use PKC in WSNs [9], [10], [11], [12].

The paper is organized as follows. Section 2 briefly presents the overview of recent platforms enabling to build WSNs. Next, Section 3 discusses security issues in WSNs. Section 4 deals with experimental result of implementation protocol stacks. Finally Section 5 presents conclusions and discusses future development.

## II. WSN Platform Overview

WSNs usually consist of a large number of ultra-small autonomous devices. Each device, called a node, is battery powered and equipped with integrated sensors, microcontroller (MCU) and Radio Frequency (RF) circuits. Nowadays, there are a lot of available MCUs and RF chips or their one-chip combination as well as various software (SW) protocol stacks. Selecting the most suitable platform is important decision in order to achieve better results. Aim of this section is to choose and to describe suitable platform for performing tests of cryptographic primitives and WSN stacks.

### A. Available hardware

Today's market offers various hardware (HW) platforms of different properties for WSN implementing. Manufacturers – Atmel [13], Ember [14], Freescale [15], Jennic [16], Microchip [17], Nordic [18] or Texas Instruments [19] produce wide variety of Radio Frequency (RF) transceivers for various frequency bands and standards from 433 MHz up to 2.4 GHz. Transceivers based on IEEE 802.15.4 standard are becoming most used in commercial area because of robust radio properties. This standard is a base of the ZigBee. Both standards are briefly described in the next sections. Manufacturers of RF chips offer various MCUs for running optimized network SW. Developers can choose between industry standard cores such as: 8051 clones (Texas Instruments - CC2431) and ARM7TDMI (Freescale - MC13225) or special vendor cores such as AVR (Atmel), HCS08 (Freescale), Coldfire (Freescale) or MPS430 (Texas Instruments). There is option to select 8-bit (AVR, HS08, 8051), 16-bit (MSP430) or 32-bit (Coldfire, ARM7TDMI) depending how powerful application has to be. Recent modern trend is to merge RF chip and MCU into one package for board area and silicon saving.

### B. Evaluation Hardware

Evaluation hardware was selected regarding to ambition of testing cryptographic protocols and interoperability between open and commercial WSN stacks. The industrial standard 8051 clone and ARM7TDMI cores was selected for testing open stacks and cryptographic protocols because of their general availability and simple porting assembly optimized SW form one clone to another. Analog Devices ADuC845 was chosen as 8051 clone. This MCU is based on modern single cycle x51 clone with 64 kB Flash and 2.3 kB Static RAM (SRAM). The most powerful peripheral in this MCU is a 24-bit sigma-delta Analog-to-Digital Converter (ADC) with programmable input gain amplifier in 1-128 gain range. The NXP (Phillips) LPC2138 was chosen as representative of ARM7TDMI architecture. This chip provides large 512 kB Flash and 32 kB SRAM memory. There is possibility to clock it at 60 MHz, what could ensure good performance for time-critical tasks. The MC13203 chip ensures RF connectivity to MCUs. Each chip is placed on its own evaluation board designed at Department of Electronics and Multimedia Communications (DEMC) with rich connectivity options.

Freescale products were chosen for running commercial WSN stacks because of availability wide range of products. It is possible to use cheaper 8-bit HCS08 core or faster 32-bit Coldfire with mutual compatibility (Flexis series [15]). There will be an option to use proclaimed powerful ARM7TDMI based MC13225 as well. Developer can choose between one-chip (MC13214) or more-chip solution (MC13203 + Flexis). Each solution is available with various memory size or with different stack usage privilege (from simple to ZigBee).

### C. Available Software Stacks

Almost each of chip vendor mentioned above provides the ZigBee or a proprietary protocol stacks. Proprietary solutions are usually available free of charge and in open ANSI (American National Standards Institute) C form such as – Simple Media Access Controller (SMAC) protocol by Freescale, JenNET by Jennic, or MiWi Wireless Networking Protocol Stack by Microchip. The ZigBee stack is available either for free of charge (e.g. MircoChip or Texas Instuments) or not (e.g. Freescale) and either in open ANSI C form or in binary form respectively, depending on chip vendor. Freescale provides powerful and easy to use Graphical User Interface (GUI) – BeeKit Development Environment, in which the users can create, modify, save and update wireless networking solution. It was chosen the Freescale platform because of supporting various MCU families, and supporting three levels of protocols – SMAC, IEEE 802.15.4 MAC and ZigBee by BeeKit sowtware. Latest two are described in next section. SMAC is set of functions for basic interfacing the IEEE 802.15.4 compliant radio. The SMAC is subset of the IEEE 802.15.4 compatible protocol and offers basic peer-to-peer connectivity only.

### D. IEEE 802.15.4 standard and ZigBee Stack

ZigBee is a low-cost, low-power, wireless mesh networking standard. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power-usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range.

The ZigBee Alliance [20] selected the IEEE 802.15.4 standard [21], released in May 2003, as the base of ZigBee networking and applications. IEEE 802.15.4 defines three frequency bands: 868 MHz, 915 MHz and 2.45 GHz. The latest is used the most frequently thanks to worldwide availability and 250 kbps bit-rate. Except frequency bands, modulation and spreading methods IEEE 802.15.4 also define relatively simple protocol, based on CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) access method to the medium.

The ZigBee specification identifies three kinds of devices that incorporate ZigBee radios, with all three found in a typical ZigBee network:

- *Coordinator (ZC):* organizes the network and maintains routing tables,
- *Routers (ZR):* can talk to the coordinator, to other routers and to reduced-function end devices,
- *End devices (ZED):* can talk to routers and the coordinator, but not to each other.

ZC and ZR are defined as Full-Function Devices (FFD), which are powered on all the time where mains power is recommended. ZED is defined as Reduced Function Device (RFD) where the protocol stack is shorter without ability of routing but this device could be battery powered. Sensors and actuators could be connected to each of these three ZigBee devices. Except common used mesh topology, it is possible to use tree or star topology, which take less HW and SW resources of the MCU.

Network devices, whether wired or wireless, are commonly described by the Open Systems Interconnection (OSI) reference model by International Organization for Standardization (ISO). The adaptation ISO-OSI network reference model for ZigBee purposes is illustrated in the Fig. 1. ZigBee network model does not use presentation, session or transport layer and user application is directly tied into Application layer (APL). This figure shows also IEEE 802.15.4, ZigBee Alliance, and ZigBee product end manufacturer particular responsibility for ZigBee certified product as well as HW and SW proportion in ZigBee.

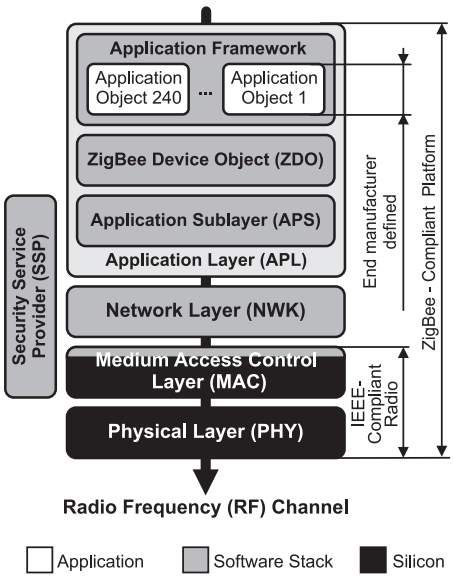


Fig. 1. Adaptation ISO-OSI to ZigBee standard where responsibility of IEEE 802.15.4, ZigBee Alliacne, and End Manufacturer is pointed out.

## III. Security Issues in WSNs

WSNs may have a few, hundreds or even thousand of nodes. As the networks grow, security and management begin to be complicated. Implementing security protocols in WSNs is not easy proposition and systems often, for reasons of complexity, limited resources or implementation fail to deliver required levels of security. The data security and network integrity of such systems are essentially based on the safe distribution of encryption keys and device authentication. Main aim of cryptographic protocols in WSNs are – establish a key between all sensor nodes that must exchange data securely, node addition/deletion should be supported, it should works in undefined deployment environment, and unauthorized nodes should not be allowed to establish communication with network nodes.

### *A. Security in the IEEE 802.15.4 standard*

While the IEEE 802.15.4 standard goes into great detail when describing the functionality of Physical Layer (PHY) and Medium Access Controller Layer (MAC), security related issues received much less attention. The standard outlines some basic security services at the MAC that can be combined with advanced techniques from upper layers to implement comprehensive security solution. The IEEE 802.15.4 device can choose to operate in unsecured mode, secured mode, and Access Control List (ACL) mode. In unsecured mode, none of the services mentioned are available. In secured mode, the device may use one of security suites supported by standard, all of which use the Data Encryption Service. A device operating in ACL mode can maintain a list of trusted devices from which it expects to receive packets. While these services are useful they are by no means sufficient. In particular, procedures for key management, device authentication, and freshness protection are not specified by the IEEE 802.15.4 standard.

### *B. Security Implementation in the ZigBee*

Security of ZigBee is provided by Advanced Encription Standard (AES) [22]. This symmetric algorithm means communicating nodes use the same key to encrypt and decrypt the messages, but the two communicating nodes must find a way to agree symmetric key. Currently, ZigBee uses symmetric-key key establishment (SKKE) to establish keys between communicating nodes. This protocol defines the mechanism by which a ZigBee device may deliver a shared key (link key) with another ZigBee device. Key establishment involves two entities, an initiator device and respond device, and should be prefaced by trust provisioning step in which trust information (a master key) provides a starting point for establishing a link key. The master key maybe preinstalled during manufacturing, it may be installed by a trust center, or it may be based on user entered data. As has been proposed in [23] SKKE scheme is not immune to malicious attacks completely.

### *C. Security Alternative (not only) for ZigBee*

Distributed systems are the ideal target to implement PKC where one key that only device knows binds the device to its identity on the network; and the second key, mathematically related to first is used by the network to verify that identity. This enables device identification to be performed rapidly, safety, and in cryptographically strong manner.

This property is useful for number of things – it greatly simplifies key exchange, as one example and it solves one critical problem secret-key cryptography (SKC) cannot solve – the problem of guaranteeing unique authentication. While personal computers have no computing limitation to implement well-known PKC algorithms such as RSA (Rivest-Shamir-Adleman), or Digital Signature Algorithm (DSA), WSNs nodes cannot use them due to constrains of used low cost and low power MCUs.

Elliptic Curve Cryptography (ECC) [24] offers secure and efficient alternative solutions for WSNs. ECC offers considerably greater security for given key size comparing to RSA. That smaller key size also makes possible much more compact implementations for a given level of security, which means faster cryptographic operations running on a smaller chips or more compact SW. This means less heat production, and less power consumption-all of which is of particular advantage in constrained WSN nodes.

An Elliptic Curve version of Menezes-Qu-Vanstone (ECMQV) protocol [24] was proposed as the key establishment mechanism and it may be suitable for ZigBee [25]. ECMQV is an efficient public-key agreement scheme that offers key authentication and key establishment. Like AES, ECMQV is fast, strong and can be inexpensively implemented in HW. In addition, by using elliptic curve methods, key sizes will be kept small even as security needs increase.

### *D. Suitable HW platform*

Implementation of ECC primitives in low performance MCU have been richly discussed in [26], [27], [28]. Computation of Elliptic Curve Digital Signature Algorithm (ECDSA) takes around 1.6 s in an 8-bit platform while the same computation takes around 100 ms in 32-bit platform that seems it can be usable in WSNs nodes. Moreover, Freescale announced the MC13225 chip, the composite of IEEE 802.15.4 radio and ARM7TDMI processor. This chip has only 20 mA current consumption in RX or TX mode, what ensures very long battery life for ZigBee end-device with appropriate active mode duty cycle. In addition this chip contains HW acceleration for both the IEEE 802.15.4 MAC and AES security and full set of MCU peripherals such as dual 12-bit analog-to-digital converter (ADC) or multiple serial channels. This chip do not need any passive matching parts to connect an antenna. The MC13225 chip, external crystal, onboard antenna, battery and optional sensor are all what application needs. In other words, using mentioned chip, PKC in WSN can became reality.

## IV. Experimental Results

Only the interoperability of SMAC based protocol between various MCUs was tried up to now. The SMAC is distributed in open (ANSI-C functions) form by Freescale for their HCS08 and Coldfire MCUs. This protocol was ported on ARM and x51 compatible microcontroller. A simple routing algorithm was written as extension for the SMAC. This routing algorithm allows star network building with 10,000 end devices.

A simple HA network (Fig. 2) was crated for interoperability demonstration. This network allows remote light switching, temperature regulation or detects door and window movements by accelerometers. There are Passive InfraRed (PIR) motion detector and Smoke Detector as well. Two options to visualize the network in computer are either Universal Serial Bus (USB)

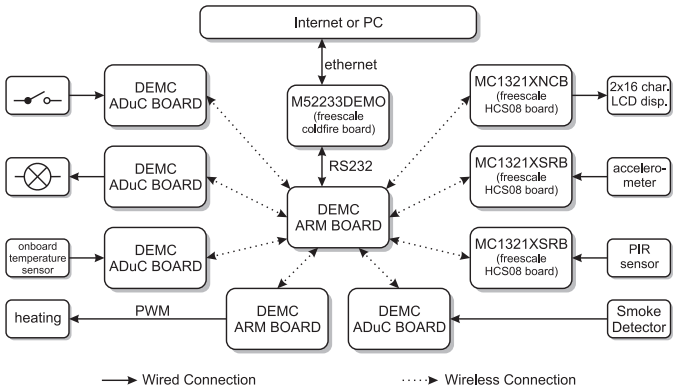


Fig. 2. Realized SMAC based Home Automation (HA) experimental network for testing hardware functionality of designed boards, HA sensors and actuators with proprietary routing algorithm and with Internet connection.

TABLE I
SMAC PROTOCOL AND ROUTING ALGORITHM MEMORY REQUIREMENTS OF VARIOUS MCUS

| MCU | HCS08 | ColdFire |
|---|---|---|
| SMAC ROM | 4491 B | 9512 B |
| Routing ROM | 462 B | 1608 B |
| SMAC RAM | 12 B | 197 B |
| Routing RAM | 413 B | 152 B |

connection of coordinator or Ethernet connection provided by Freescale M52233DEMO board with web-server SW. Coverage radius of this network was about 40 m, what is enough for flat or small house. A SMAC memory requirement for various used microcontrollers is shown in Table I.

## V. CONCLUSION

Paper describes implementation proprietary SMAC protocol by means of various MCU platforms. Simple routing protocol was created in order to a build simple HA network. It was compared resource requirements of implementation for both representative of Freescale family – HCS08 and ColdFire.

While ZigBee is a modern and powerful standard to creating WSN, its security lies upon AES and protocols such SKKE. However, while SKC has low requirements for processing power, it probably can not provide enough robustness and security in large scale networks. On the other hand, PKC provides availability of authentication and key exchange mechanisms that are more secure and reliable compared to SKC. Besides these advantages, the PKC has also one main disadvantage – it is computationally expensive. There are two strong limitations to implement PKC – first, to keep messages as short as possible because of each bit transmitted consumes about as much power as executing 800-1000 instructions [29], and as a consequence, any message expansion caused by security mechanisms comes at significant cost; and second, using low performance MCU is necessary in order to develop ever-cheaper sensor nodes. It is nowadays clear that it is possible to apply ECC based PKC, but the question that remains is how the application of strong public key cryptography affects the lifetime of the energy source and thus the lifetime of the sensor. This is why we would like to target next research to observe costs of PKC protocols in real low cost WSN platform (e.g. MC13225) and compare with the SKC ones in the terms of resource requirement, speed, network security, and global performance.

## ACKNOWLEDGMENT

This work has been done in the frame of the Slovak scientific projects VEGA 1/4088/07 of Slovak Ministry of Education and the grant – EU 6th Framework Programme, IST, MonAMI – Mainstreaming on Ambient Intelligence 035147. The MonAMI project is focused on Home Automation for elderly persons and persons with disabilities living at home (www.monami.info). Authors also thank to Freescale, Semiconductor, Inc. for providing software and hardware development tools used within this project.

## REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and CayirciE., "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
[2] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar, "Spins: Security protocols for sensor networks," in *Seventh Annual International Conference on Mobile Computing and Networks*, July 2001.
[3] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 41–77, February 2005.
[4] e. a. Burne, R., "Self-organizing cooperative sensor network for remote surveillance: improved target tracking results," in *Proceedings of the SPIE*, 2001.
[5] B. Fulford, "Sensors gone wild," Forbes Global, 2002, online. http://www.forbes.com/business/forbes/2002/1028/306.html.
[6] J. Polastre, "Design and implementation of wireless sensor networks for habitat monitoring," Master's thesis, University of California at Berkeley, 2003.
[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *First ACM Workshop on Wireless Sensor Networks and Applications*, 2002.
[8] P. Saffo, Homepage, online. www.saffo.com.
[9] G. Gaubatz, J. Kaps, and B. Sunar, "Public keys cryptography in sensor networks - revisited," in *The Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS)*, 2004.
[10] N. Gura, A. Patel, A. Wander, H. Eberle, and S. S. C., "Comparing elliptic curve cryptography and rsa on 8-bit cpus," in *Proceedings of the Workshop on Cryptography Hardware and Embedded Systems (CHES)*, August 2004.
[11] D. J. Malan, M. Welsh, and S. M. D., "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *The First IEEE International Conference on Sensor and Ad Hoc Communications and Networks*, October 2004, pp. 71–79.
[12] W. Du, R. Wang, and N. P., "An efficient scheme for authenticating public keys in sensor networks," in *MobiHoc'05*, 2005, pp. 58–67.
[13] Atmel, Offcial Site, online. www.atmel.com.
[14] Ember, Offcial Site, online. www.ember.com.
[15] Freescale, Offcial Site, online. www.freescale.com.
[16] Jennic, Offcial Site, online. www.jennic.com.
[17] Microchip, Offcial Site, online. www.microchip.com.
[18] Nordic, Offcial Site, online. www.nordicsemi.com.
[19] T. Instruments, Offcial Site, online. www.ti.com.
[20] ZigBee Alliance, Official Site, online. www.zigbee.com.
[21] IEEE, "Ieee std 802.15.4-2003," online. http://standards.ieee.org.
[22] NIST, *Federal Information Processing Standards Publication 197 – Advanced Encryption Standard*, October 2001, online. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
[23] M. J., "Zigbee: A long way to go?" online. http://www.sciencedirect.com/.
[24] D. Hankerson, J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. New York: Springer-Verlag, 2004.
[25] M. Blaser, "Industrial-strength security for zigbee: The case for public-key cryptography," *Embedded Computing Design*, May 2005.
[26] R. Roman and C. Alcatraz, "Applicability of public key infrastructures in wireless sensor networks," 2007.
[27] M. Drutarovsky and M. Varchola, "Cryptographic system on a chip based on actel arm7 soft-core with embedded true random number generator," in *Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, Bratislava, April 2008, accepted.
[28] M. Aydos, T. Yanik, and C. . K. Koc, "An high-speed ecc-basedwireless authentication protocol on an arm microprocessor," in *The 16th Annual Computer Security Applications Conference*, 2000, pp. 401–409.
[29] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proceedings of ACM ASPLOS IX*, November 2000.