International Journal of Sciences and Techniques of Automatic control & computer engineering
IJ-STA, Volume 6, N° 1, June 2012, pp. 1782–1791.

Key Management of Wireless Sensor Networks - Design and Implementation on FPGA

Mohamed Wassim JMAL[1], Haythem GUIDARA[1] and Mohamed ABID[1]

[1] CES research unit, University of Sfax, Tunisia
mohamed.wassim-jmal@ceslab.org,
haythem.guidara@gmail.com, mohamed.abid@enis.rnu.tn

Abstract. *The security in Wireless Sensor Network is becoming more and more important due to the expansion of its application domains. The main question is the capability of existent sensor nodes to handle the implementation of the new complex security techniques. In this paper, different techniques used in key management are presented showing incapability of the existent node to deal with them especially with the asymmetric cryptography. An alternative architecture is proposed to solve this problem. The proposed solution is based entirely on open sources and will be validated with an FPGA implementation. A physical realization and performance measurement was also presented in this paper.*

Keywords. *Wireless Sensor Networks, Key Management, Elliptic Curve Cryptography, FPGA implementation, Open Sources.*

1. Introduction

The wireless network sensor is a result of combination between major domains: Micro Electro-Mechanical Systems (MEMS) technology, wireless communications and digital electronics. The recent advance in these domains has made the WSN more and more important with increasing the multi-functionality and capabilities of sensor nodes [1]. A wireless sensor network WSN is a collection of hundreds to thousands of sensor nodes connected to each other through short range wireless links, used as an infrastructure to forward the collected report to the centralized authority over a base station. Sensor nodes are self powered and equipped with low computational power CPU allowing the sensor to execute some specific treatment before sending a report to the centralized authority.

The application range integrating WSN is spreading taking advantage of the flexibility and deployment's easiness. Environmental monitoring, biomedical research,

This paper was recommended for publication in revised form by the editor Staff.
Edition: CPU of Tunis, Tunisia, ISSN: 1737-7749

human imaging, tracking, and military applications are the most important WSN applications [2]…

By its nature, Wireless sensor network have very limited circuit area and power supply making additional critical issues beside real time performance, cost and robustness. Taking consideration of these constraints, the problem of security represents a real challenge.

For the purpose of securing the WSN, several key management schemes was established trying to solve the problem of security in WSN by taking into consideration the limitations of sensors, the majority of them are based on symmetric key encryption. But to achieve a high level of security the use of asymmetric encryption is crucial.

Based on microcontroller architectures with severe limited computing abilities and optimized operations, strong asymmetric cryptography is commonly seen as infeasible on sensor devices. Thus, a new sensor architecture based on microprocessor is needed. The use of open sources in the development of the architecture is a solution that can be considered.

The remainder of this paper is organized as follows. First, key management schemes used in WSN and the evaluation of their needs to advanced and complex algorithms were presented. Sections 3 deals with elliptic curve cryptography and its advantages compared to other asymmetric encryption systems. The possibility of its implementation on existent WSN platforms with a software design was presented in Section 4. Section 5 shows an overview of the existent sensor nodes and their characteristic. In section 6, the design of processor architecture for WSN is proposed. The design validation on FPGA is described on Section 7. Finally, we conclude in Section 8.

2. Key Management Schemes in WSN

The key management is an important service for the security of any system based on communication, since it provides effective, secure and reliable mechanisms of key management used in cryptographic operations.

Under the constraints of WSN, the design of a management key system is a challenge. Select a cryptographic solution suitable for WSN is another challenge.

In literature, there are several key management schemes to resolve security issues in WSN, taking into consideration the constraints and limitations of sensors (bandwidth and energy consumption). The majority of schemes are based on symmetric encryption and there is other based on asymmetric encryption.

The schemes based on symmetric encryption are:

- Shared key: this is the simplest solution for securing the wireless sensor network. It consists of using one key for securing communication over the network. This scheme is vulnerable against capture attack and the capture of one sensor will compromise the whole network.

– Pre-distributed keys: It consists in distributing a set of symmetric to each sensor before deployment. Before communicating, each sensor tries to find a shared key with each neighbor sensor to secure the links between them. This scheme cannot offer possibility for network reinforcement and it requires much memory for key storage [3].

– Tinysec: is a link layer security protocol based on symmetric key encryption, TinySec supports two different security options: authenticated encryption (TinySec-AE) and authentication only (TinySec-Auth). The use of MAC layer security instead of end to end security may avoid denial of service attacks, however this scheme still vulnerable to lot of attacks as capture attacks. In other hands, this protocol can be used by any other key management scheme as an underlying tool for encryption [4].

– Cluster based protocols: It is based on dividing the whole network into clusters. A set of symmetric keys are used to ensure intra and inter cluster communication as well as integrity, confidentiality and authentication over each cluster and therefore over the whole network[5].

The schemes based on asymmetric encryption are:

– Simplified SSL handshake: it consists in setting up a secure key between any sensor pairs by a simplified version of SSL handshake. The major problem of this scheme is that it cannot be applied in a mobile network since the mobility of nodes sensor need a new handshake with every position change and that consumes a lot of energy [6].

– TinyPK: The TinyPK system described in [7] is designed specifically to allow authentication and key agreement between resource constrained sensors. The protocol is designed to be used in conjunction with other symmetric encryption based protocols as TinySec, in order to deliver secret key to that underlying protocol. To do this, they implement the Diffie-Hellman key exchange algorithm.

– Simplified Kerberos protocol: The authors in [8] proposed an adapted version of Kerberos [9] for WSN in order to setup a session key between each communicating pair of sensors by contacting a trusted third party which may be the base station or a cluster head in a hierarchical network. They assume that a long term key is shared between each node and the trusted authority which is responsible of the generation of the secret key for each pair of sensors.

Any system that requires high security level like in military domain has to either use only symmetric encryption with pre distributed key and it will occupy huge memory space especially with large network, or use hybrid encryption combining symmetric and asymmetric encryption which is a compromise between performance and security level.

3. Elliptic Curve Cryptography

As said above, any cryptosystem with high security level must be based totally or partially on the use of asymmetric encryption.

The ECC algorithm [10] can be classified as the one of the most efficient asymmetric algorithms regarding its energy cost as well as its encryption speed, making it the base of future key management and security protocol for WSN and any other wireless ad hoc network [11].

In table 1 we give the energy cost of the RSA and ECC algorithms for signature and verification applied to Berkeley/Crossbow motes platform, specifically on the Mica2dots, as we can observe the ECC is always more efficient compared to RSA for the two used key length, given that the length of keys used by ECC are much smaller than RSA's keys which may save lot of memory space for sensors. Also, ECC's encrypted blocks are more small than the RSA's ones which saves network bandwidth during transmission.

Table 1. Energy cost of digital signature (mJ)[12]

Algorithm	Sign (mJ)
RSA-1024	304
ECC-160	22,82
RSA-2048	2302,7
ECC-224	61,54

4. Software Design

For the implementation of elliptic curve cryptography, many libraries provide the source code and files needed for the development of ECC. Among these libraries we quote: borZoi, Crypto++, LibTomCrypt, LiDIA, MIRACL, OpenSSL, Bouncy Castle, FlexiProvider, IAIK and Benchmarking. Considering both performance and the development language, our choice was to use open source libraries that offer basic cryptography functions. In particular, we used MIRACL C library [13]. MIRACL is a Big Number Library which implements all of the primitives necessary to design Big Number Cryptography into real-world application. It is primarily a tool for cryptographic system implementers. The library does not provide fixed and rigid cryptographic interfaces but rather offers the possibility of change and improvement. In addition the library is easy to use for creating new applications as needed.

To develop testing cryptographic program, we had to develop a static cryptographic library based on some MIRACL files including all the necessary primitives. The test program will only generate the two pair of key: public key and private key and this from curve parameters. These parameters were chosen according to National Institute of Standards and Technologies recommendations.

In The following table we give the needed files for building the library and the size of the main program.

Table 2. File needed for application development

Library	Library files	Library size	Code size
ECCcrypt.a	Mirdef.h, Miracl.h, Mrcore, Mrarth0, Mrarth1, Mrarth2, Mralloc, Mrio1, Mrio2, Mrxgcd, Mrarth3, Mrrand, Mrmonty, Mrcurve, Mrcrt, Mrecgf2m, Mrgcd, Mrmonty, Mrpower, Mrprime, Mrand, Mrshs, Mrstrong, Mrxgcd	293 Ko	16 Ko

The main question is: Does the existent nodes platform can manage the complexity of algorithm from a performance perspective?

5. Wireless Sensor Nodes

The sensor nodes are available in a multitude of models in relation to the application for which they are intended. Among the most common models we find the MICA sensors developed by UC Berkeley and marketed by Crossbow, Imote sensors marketed by Crossbow and TinyNode sensors developed for real applications related to the industry by Shockfish.

5.1. MICA 2

MICA2 [14] is a third generation sensor used for the wireless sensor network with low energy consumption. This sensor is developed by Berkeley University and used in:

- Environmental Controls.
- Surveillance and security.
- Sensor networks with large capacity (more than1000 nodes).

5.2. TinyNode

This sensor node is developed by Shockfish SA. It is optimized to support TinyOS. We find this type of node in the following applications [15]:

- Environment monitoring.
- Precision agriculture.
- Parking Management.

5.3. TelosB

The platform TelosB [14] was developed and published in the scientific community by the University of Berkeley. This platform provides low power consumption for long battery life and a rapid awakening of the waking state. The microcontroller TPR2420 used in TelosB, is compatible with the distribution of open-source TinyOS. This node type can be used in the following applications:

– Low power Platform for research and development.

– Wireless Network Sensor experimentation.

These platforms mentioned above are built around microcontrollers like ATMega 128L and MSP430 functioning at 8 MHz. The results of implementation of different asymmetric encryption on MICA2 are given in Table 2.

Table3. Average times for different operations [12]

Operation	Execution without optimization [s]	Execution time with memory optimization [s]
Point multiplication (fixed)	≈34	6.74
Point multiplication (random)	≈34	17.28
Key generation	≈34	6.74
Complete Diffie-Hellman key exchange	≈68	24.02
El-Gamal encryption	≈68	24.07
El-Gamal decryption	≈34	17.87
ECDSA signature	≈34	6.88
ECDSA verification	≈68	24.17

Due to its complexity, ECC algorithm cannot be supported by the platform mentioned above. Any implementation on these targets will cause important delays in WSN and even dysfunction

5.4. Imote2

Integrating a high performance, low power, PXA271 Intel XScale[R] processor and a 802.15.4 radio with a built in 2.4 GHz antenna, the Imote2 [14] provides a high performance platform for advanced, compute intensive wireless sensor network applications such as digital imaging and industrial vibration monitoring.

The PXA271 processor on the Imote2 can be configured to support a low voltage (0.85V), low frequency (13 MHz) mode as well as other scalable processing modes of up to 416 MHz. The 256KB of on-chip SRAM, 32 MB of SDRAM and 32MB of FLASH memory provide several orders of magnitude more resources for memory intensive applications than previous WSN hardware platforms.

Power consumption is extremely low making this platform ideally suited for demanding but battery powered applications.

This platform is an excellent choice for ECC implementation thanks to its preferment processor and memory capacity.

The major inconvenient in Imote2 is that it is costly especially with large deployment. The high cost of this platform is due to the use PXA271 processor. This processor came with a DSP coprocessor that is not necessary in all application.

The idea is to build a platform that can handle new cryptosystem algorithms with the possibility of processor adaptation according the needed application.

6. Proposed solution

In this part we present an architecture development to be used in wireless sensor network. The architecture is dedicated to be the core of the node platform based in the use of the open hard and soft sources. The hardware development of the architecture is based on the use of the LEON3 soft core processor. To make Leon suitable for wireless sensor network changes are made in the open VHDL packages of the LEON3 processor.

The LEON processor [16] is an open source, soft core processor developed and supported by Gaisler Research. It is implemented as a high level VHDL model, fully synthesizable with common synthesis tools. The model is extensively configurable through a (graphical) configuration package, allowing options such as cache size and organization, arithmetic operation implementation, I/O modules and other IP cores to be selected.

LEON3 came with the following features:

- Separate instruction and data caches (Harvard architecture)
- Interrupt controller
- On chip debugging support
- 2 24 bit timers
- Watchdog timer
- 2 UARTS
- Ethernet MAC and PCI interface
- Co-processor support

LEON3 is distributed with a large GPLed library with, among others, IP cores for a CAN controller, CCSDS telemetry and telecommand functions.

To minimize energy consumption and to meet the needs of the system we have tailored the LEON3 processor following this architecture (fig.1):

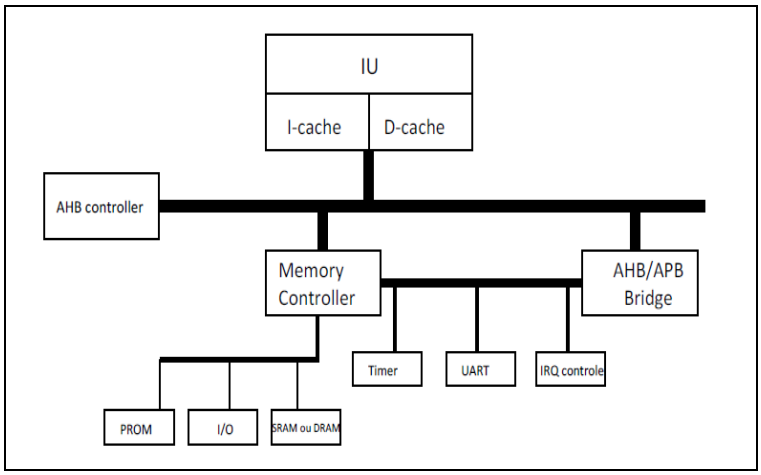

Fig. 1. The enhanced architecture for WSN based on LEON3.

The first step in the design process is the configuration of the LEON3 by using a graphical configuration tool provided with the LEON3 model from Gaisler Research.

This tool is built upon the TCL/TK software, providing some menus that allow the designer to configure the system on chip.

Second, we simulate the model by running a test bench provided with LEON VHDL model. The simulation is done by a simulation tool like Modelsim from Mentor, NCSIM from Cadence2, VSS from Synopsys or GHDL from GNU.

The test program previously mentioned has to be compiled with BCC: Bare-C Cross Compiler, is a C/C++cross compiler for the LEON3 processor based on the GNU tool chain, the binutils and the standard Newlib library, with full math support and simple I/O operations (non-files).

The result of compilation is the executable file by LEON3. To ensure that the program is working before hardware implementation we used TSIM2 a complete LEON3 instruction-level simulator. TSIM can run in standalone mode or connected through a network socket to the GDB debugger, acting like a remote target using a common debugging protocol. TSIM is a commercial application but it's also available as an evaluation version for non-commercial uses.

7. Design validation on FPGA

To validate the designed architecture on a real hardware target, we have chosen the cycloneIII starter board from Altera for the implementation. This board embed an EP3C25F324 FPGA and a USB Blaster. It also offers 1Mbyte of SSRAM and 50 MHz oscillator.

The first step is the synthesis. From the HDL files previously configured and with a synthesis tool like simplify from synopsis or Quartus provided by Altera.

The result of synthesis was the use of 9167 logic element.

The FPGA programming file is generated by Quartus program after pins assignment. This file is downloaded to the FPGA via USB cable provided with the development board.

To get the test program running on LEON3 we used a tool provided by Gaisler research named GRMON. GRMON is a debug monitor for LEON processors. It communicates with the LEON debug support unit (DSU) and allows non-intrusive debugging of the complete target system. To be able to use this tool we had to include the debug support unit in the LEON3 architecture to provide a link for the communication with GRMON.

The performance test was realised using sect163r1 which is elliptic curve domain parameter over F2m [17] recommended by Standards for Efficient Cryptography Group (SECG).

The result of the test shows that the program time execution on the architecture is about 760 ms . This execution time is very promising since we didn't try to use any optimization that may reduce this time like in [18].

The energy consumption is about 100mJ but because it's measured on FPGA it is not accurate. However it can give us an overview on system consumption.

The authors in [19] and [18] mentions many ways to ameliorate time execution and energy consumption like the use of new signed binary representation and also the use of tinyECC library that offers many optimization techniques.

8. Conclusion

The complex security techniques used in specific WSN application like in military domain need important resources of calculation, memory and energy. The majority of existent WSN platform does not satisfy these needs because they are based on microcontrollers. Therefore the necessity to new platform built around microprocessor able to handle the complexity of the algorithms. The development of such platform must take consideration of system cost, the time to market and easiness of adaptation in the case of application changing.

The proposed solution based on open core processor had solved the problems mentioned above and can be ameliorated with hardware accelerator if needed.

The proposed architecture has been validated with an implementation on FPGA letting the test of functionality and performance. The results of tests are very promising for an ASIC implementation.

References

1. O.Moussaoui and al, "Efficient saving in wireless sensor networks through hierarchical-based clustering", In proceeding of the international IEEE Global Information Infrastructure Symposium, Marrakeche, Morocco, pp. 226-229, July, 2007.
2. Akyildiz I. F. Su W., Sankarasubramaniam Y. et Cayirci E. "Wireless sensor networks: a survey", Computer Networks. Vol. 38(4). - pp. 393-422, 2002

3. L. Eschenauer, V.D. Gligor, “A key-management scheme for distributed sensor networks”, in Proceedings of the 9th ACM conference on Computer and Communication Security, pp. 41-47, November 2002.
4. C. Karlof, N. Sastry, and D. Wagner. Tinysec “A link layer security architecture for wireless sensor networks”, In Second ACM Conference on Embedded Networked Sensor Systems (SensSys 2004), pages 162–175, November 2004.
5. Benamar KADRI, Mohammed FEHAM, Abdallah M’HAMED. “A new management scheme of cluster based PKI for ad hoc networks using multi-signature”, In proceeding of the international IEEE Global Information Infrastructure Symposium, Marrakeche, Morocco, pp 167-172, 2007.
6. Wander, A.S., Gura, N., Eberle, H., Gupta, V., and Shantz, S.C., “Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks”, In proceedings of PerCom pp. 324-328, 2005.
7. R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus. Tinypk, “securing sensor networks with public key technology” In Proceedings of the 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN ’04), pages 59–64, 2004.
8. Johann.G, Alexander.S, Stefan.T. “The Energy Cost of Cryptographic Key Establishment”, in Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, pp. 380–382, (ASIACCS 2007).
9. J. T. Kohl and B. C. Neuman. “The Kerberos Network Authentication Service” (Version 5). Internet Engineering Task Force (IETF), Internet Draft RFC 1510, Sept. 1993.
10. Koblitz, N. “Elliptic curve cryptosystems”. Mathematics of Computation,Vol.48, 1987.
11. Lenstra, A. K., Verheul, E. R. “Selecting Cryptographic Key Sizes”. Journal of Cryptology: the journal of the International Association for Cryptologic Research, 2001.
12. Erik-Oliver Blaß and Martina Zitterbart “Towards Acceptable Public-Key Encryption in Sensor Networks”, 2005.
13. MIRACL, “Multiprecision integer and rational arithmetic C library”, [Online] [Accessed Sept. 3, 2010]. Available: www.shamus.ie
14. http://www.xbow.com/ Internet site of Crossbow Technology, Inc. [accessed Mars 2010].
15. http://www.shockfish.com/ Internet site of Shockfish SA [accessed Mars 2010].
16. Johannes van der Horst “Literature Study:Radiation tolerant implementation of a LEON processor for space applications” June 6, 2005.
17. Certicom Research. Standards for efficient cryptography - SEC2: Recommended elliptic curve domain parameters. http://www.secg.org/download/aid-386/sec2_final.pdf, September 2000.
18. Bangju Wang; Huanguo Zhang; Yuhua Wang. “An Efficient Elliptic Curves Scalar Multiplication for Wireless Network” Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on Digital Object Identifier: 10.1109/NPC.2007.43 Publication Year: 2007 , Page(s): 131 – 134.
19. An Liu, Peng Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," in Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), SPOTS Track, pages 245--256, April 2008.