2011 30th IEEE Symposium on Reliable Distributed Systems Workshops

Analysis and comparison of security protocols in wireless sensor networks

Valentina Casola, Alessandra De Benedictis, Annarita Drago and Nicola Mazzocca
Dipartimento di Informatica e Sistemistica
University of Naples Federico II, Napoli, Italia
Telephone : +39 0817683907
Fax : +39 0817683916
Email: {valentina.casola,alessandra.debenedictis,nicola.mazzocca}@unina.it, annarita.drago@fastwebnet.it

***Abstract*—Wireless sensor networks are widely used in several application domains thanks to their data acquisition and processing capabilities and their decentralized and self-organizing nature. A widely distributed monitoring system is typically characterized by different security requirements that should be addressed by means of specific security protocols and architectures. Indeed, security solutions should be properly designed as they could have a strong impact on the overall performances. In this paper, we focus our attention on security problems related to the data exchange between sensor nodes and evaluate the performances of two different cryptosystems used to guarantee confidentiality, integrity and authentication requirements.**

I. Introduction

Wireless Sensor Networks (WSNs) are widely adopted in critical scenarios and security issues are becoming a fundamental concern to be addressed by means of proper security policies and mechanisms. WSNs can be considered as belonging to wireless ad hoc networks, but they present a lot of proper features making the well-known security solutions not directly applicable. Unlike ad-hoc networks nodes, WSN nodes are typically provided with constrained processing and storage capabilities and limited energy resources; they are prone to failures due to harsh deployment environments and are easy to be compromised due to typically unattended operations. Furthermore, a WSN is often characterized by a dynamic topology due to node joining, mobility or failure, thus introducing further security and reliability issues. Security must be addressed at different architecture layers, namely at the physical node level, at the inter-node communication level, and at the application level. In this paper, we will focus our attention on security problems related to the data exchange between sensor nodes; we want to satisfy confidentiality, integrity and authentication requirements, but traditional network security protocols cannot be applied because of the limited resources and capabilities available on the sensor nodes. In particular, the key management problem and the needed resources to sign and encrypt messages are challenging open issues and some solutions have been proposed in the literature. Indeed, we are interested in understanding the impact of security solutions on the overall system performance and we are primary interested in understanding which are the parameters that can affect the provided security level against the feasibility of the sensor network.

At this aim, we will analyze and compare two different cryptosystems based respectively on ECC libraries and Identity based cryptographic techniques. In particular, ECC provides key agreement algorithms and digital signature that can be used to authenticate any packet in the sensor network, while the Identity based cryptography techniques provide an interesting solution to the key management problem. We will illustrate the implementation of key management and message signature protocols for both techiques and their integration within the SeNsIM-SEC framework [2]. Such cryptosystems have been implemented on TinyOS sensor middleware [15] by exploiting the WM-ECC and TinyPairing libraries [11], [13].

The introduction of security requirements opens new possible scenarios of adoption of sensor networks but can introduce a huge overhead to the whole architecture; at this aim, in the last part of the paper, we will present a performance analysis on the overhead introduced by the proposed security mechanisms and discuss possible design constraints that can arise.

The reminder of the paper is structured as follows: in Section 2 a brief overview of the security open issues and available security solutions is drawn and in Section 3 we will illustrate significant details on the implementation of the cryptosystems to compare. In Section 4 evaluation results will be presented and discussed. Finally in Section 5 some conclusions and future works will be drawn.

II. Security in Wireless Sensor Networks

A typical monitoring system is made of different sensor networks that can be heterogeneous in the technology aspects, in the data formats, in synchronization and localization standards and so on. Providing security services in wireless sensor networks is a technical challenge, due to hostile deployment environments and resource limitations.

A monitoring infrastructure can be considered as structured into two main layers, namely the *sensor network layer* and the *distributed application layer*.

978-0-7695-4451-9/11 $26.00 © 2011 IEEE
DOI 10.1109/SRDSW.2011.27

As discussed in [2], security issues arise at different levels; in this paper we will focus our attention on the transport layer and we will evaluate and compare two different cryptosystems.
Because of the low computational costs, Symmetric Key Cryptosystems are widely adopted in WSNs (TinySec [3], MiniSec [4], ZigBee [5]), but they require complex key distribution and management protocols. Moreover, symmetric cryptography only fulfills confidentiality requirements, while not considering other security issues such as authentication and integrity.

Asymmetric key cryptosystems can ensure a higher degree of security while guaranteeing a greater flexibility and manageability than symmetric ones: thanks to them, any two sensors can establish a secure channel between themselves; moreover, as nodes do not share the same common key for encrypting/decrypting messages, the tampering of some sensor devices or other network attacks will not affect the security of the whole system. Rivest-Shamir-Adelman (RSA) algorithm [6] and Elliptic Curve Cryptography (ECC) [7] are among the most well known public key algorithms used in security systems; the latter is based on the algebraic structure of elliptic curves over finite fields, and it is very promising in limited resource architectures.

ECC provides: (i) the Elliptic Curve Diffie-Hellman (ECDH) key agreement algorithm, to aid two communicating nodes with the possibility of achieving the same secret key without physically exchanging it across the network; (ii) the Elliptic Curve Digital Signature Algorithm (ECDSA) [7], a variant of the Digital Signature Algorithm (DSA), that operates on elliptic curve groups and can be used for signature generation and verification.

Although ECC techniques have improved the performances of the asymmetric cryptosystems, they do not address the problem of public keys authentication, thus making networks vulnerable to man-in-the-middle attacks (ECDH protocol). This is usually achieved by a Public Key Infrastructure through digital certificates, whose management is still too complex and expensive for sensor nodes. A possible solution, very suitable to the sensor networks, is given by Identity-based cryptography techniques (IBC), which rely upon unique node identifiers (for example the node ID assigned at the setup of the network) for both exchanging keys and encrypting data without the need for a PKI. The IBC techniques can be implemented thanks to the Pairing-Based Cryptography (PBC) [8][9][10], which has been recently adopted in WSNs [11][12], providing for a better solution for key management and a higher degree of security compared with the symmetric schemes and certificate-based public key schemes.

In the following section we will describe the implementation of two network systems with different security protocols to enable both key management and signatures; they will be integrated into a distributed monitoring platform named SeNsIM-SEC.

III. SeNsIM-SEC Architecture and implementation

In previous papers we proposed an architecture for interoperability of different sensor networks (SeNsIM-SEC) [2]; in Figure 1 the architectural model is represented, showing two different sensor networks to be integrated, having different features in terms of hardware platforms, operating systems, middlewares and security requirements. Each network is managed by a dedicated wrapper that is able to communicate with the specific underlying technology and acts as a connector for the mediator component; the mediator is responsible to properly format user requests and forward them to the different wrappers, that translate the incoming queries and inject them into the underlying networks, retrieve the results and pass them back to the mediator.

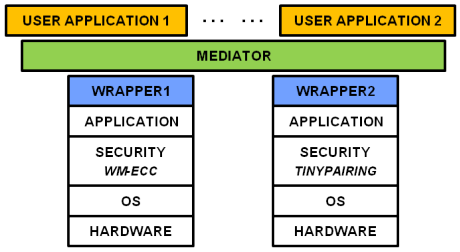

Fig. 1. The SeNsIM-SEC architecture

In the remainder of this section we will give some details about the two cryptosystems (WM-ECC and TinyPairing) adopted to secure different sensor networks.

A. The WM-ECC library

The WM-ECC library [13] is a publicly available open source implementation of a 160-bit ECC cryptosystem targeted to MICAz, TelosB and Tmote Sky platforms, based on recommended 160-bit SECG elliptic curve parameters (secp160r). Fundamental ECC operations are based on large integer arithmetic operations over finite fields as multiplication, division and modular reduction; in order to improve the performances of encrypting/decrypting operations, authors of WM-ECC library have exploited several optimizations by directly implementing many operations in assembly language. WM-ECC provides support for all the ECC operations and gives an optimized implementation of the ECDSA protocol for digital signature generation and verification, relying upon techniques such as sliding-window and Shamir trick, and has been proved to be more computationally efficient than its major counterparts.

In [2] we exploited the WM-ECC library in order to implement a hybrid cryptosystem that relies upon a public key function to ensure authentication of the base station (master node), and upon a key agreement protocol to establish a symmetric key to be used for

encryption/decryption between the base station and each of the motes (based on the Skipjack cipher).

In Figure 2 the execution of a query in a network, secured with the presented WM-ECC cryptosystem, is shown: at the system startup (red box in figure) the master node starts the ECDH protocol in order to achieve a common shared secret with each of the motes, then digitally signs every outgoing query packet and decrypts the incoming response packets before sending the results to the high level querying application. The mote in turn, is able to verify the digital signature of the incoming query packets, and to encrypt all outgoing response packets. Details on the implementation of this protocol are available in [2].

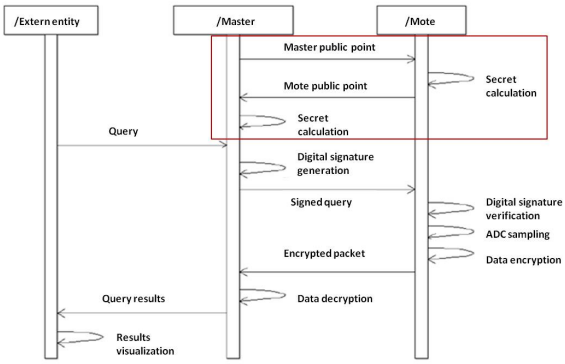

Fig. 2. Query execution in a network secured with the presented WM-ECC based cryptosystem

B. TinyPairing library

TinyPairing [11] is an open-source pairing-based cryptographic library for wireless sensors, designed to reduce memory occupancy (both ROM and RAM). It provides efficient and lightweight implementation of bilinear pairing, pairing-related functions and associated elliptic curve arithmetic operations such as scalar multiplication, point addition and more, and is the most efficient pairing based NesC implementation currently available. In their implementation, the authors include three well-known pairing-based cryptographic schemes which have been employed in some recent solutions to secure WSNs: Boneh-Franklin Identity-Based Encryption (BF IBE) basic scheme [8], Boneh, Lynn and Shacham's Short Signature (BLS SS) scheme [9], and Boneh and Boyen's Short Signature (BB SS) scheme [10]. The entire library is written in nesC for TinyOS v2.x without using any hardware-dependent code, so it is easy to port to most of sensor platforms.

Figure 3 shows the sequence of operations needed for executing a query in the TinyPairing-based cryptosystem that we set up.

- In the pre-deployment phase (red box in figure), carried out in a protected environment, the master node is assigned a unique 8 byte identifier based on manufacturer and serial number information and performs two operations: in the *key setup* operation it uses a randomly chosen large integer (secret master key) and generator point to obtain a *public point*; in the *key extract* operation, the master key and the unique 8 byte ID are used to achieve the *master private key* (dID) associated with its public ID, which will be used in decrypting operations.
  The master ID, its public point and the random generator point are sent to each of the motes, as this information is necessary in signature verifying and response encrypting operations.
- At the arrival of a query from the UART interface, the master builds a query packet with the received parameters, digitally signs it with its master key (according to the BLS SS scheme) and then broadcasts it to the motes via the radio channel;
- When receiving a query packet, a mote first verifies the digital signature using the master public point and generator point, achieved during the pre-deployment phase and, if the verification turns to be successful, starts to sample the required physical values according to the query parameters; each retrieved sample is collected and inserted into the payload of a response packet, which is encrypted using the master *ID* (according to the BF IBE scheme) before being sent back to the master;
- The master receives the response packet, extract its payload and decrypts it with its private key *dID*; then, the master returns the result values to the high level querying application through the UART interface.

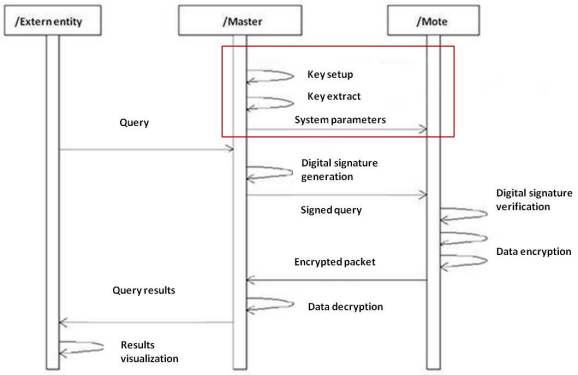

Fig. 3. Query execution in a network secured with the TinyPairing based cryptosystem

As illustrated, the query process is very similar to the WM-ECC based protocol, except for the key agreement phase that is more secure: only the master node calculates key information and sends the necessary parameters to the motes in order to let them perform cryptographic

operations.

IV. Security and Performance Analysis

The introduction of security mechanisms within a sensor network is a desirable feature but it may introduce a very heavy overhead that must be addressed by any sensor networks developer and deployer.

The analysis we conducted in this work aims at comparing the performances of the two implemented cryptosystems in terms of the latency introduced in the whole monitoring architecture, the overhead introduced by the protocol and the resource occupation to elaborate cryptographic functions on single nodes.This information, in fact, should be taken into account by any designer to meet his security and performance requirements.

Our testbed is shown in Figure 4: it is made of three sensor networks cooperating trough SeNsIM-SEC, they all are composed of Telosb motes with a 4.15 MHz MSP430 microcontroller and a CC2420 radio chip and having a 10 kB internal RAM and a 48 kB program Flash memory. Such networks have different security requirements and are managed with different techniques: two of them are configured to work with the two proposed cryptosystems while the third one works with the same application without any security mechanism. Each sensor network is composed of a master node directly connected to the wrapper component which is, in turn, connected to the mediator component via a TCP/IP network.

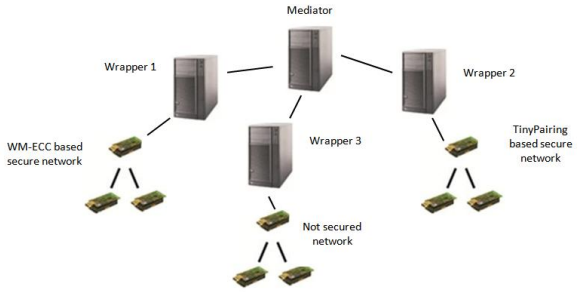

Fig. 4. The adopted testbed

To analyze the performance, we carried out several experiments by sending different queries to the underlying networks through the mediator interface. In the time overhead analysis, we did not consider the delays related to wrapper-mediator communications over the tcp/ip network (as they depend on how the application is distributed to monitor a particular environment) and we just considered a one level deep tree sensor network topology.

Indeed, these choices may heavily affect the total response time but, in this experiments, we just want to measure the overhead introduced by the security protocols and not the total response time.

In Table I the latency introduced by the security protocols is reported and compared. As illustrated, WM-ECC has overall better performance. On the contrary, the TinyPairing protocols introduce anomaly situations; in fact, it is easy to see that the signature and verification times as well as encryption and decryption times are not symmetric. This values are not surprising as they depend on the specific sequence of additional operations that are needed to implement the TinyPiring mechanism. As for the signature verification, in fact, it requires a double call of the same pairing function with different input parameters; as for the encryption, it requires a sequence of different operations to initialize the ECC-based algorithm (hash-to-point, point scalar multiplication, ...) as illustrated in [11]. This latency could be reduced by buffering some samples and encrypting them all together (at most 4 samples

Cryptosystem	**WM-ECC**	**TinyPairing**
Initialization	4,60104	17,08398
Sign	1,50702	8,69433
Verify	2,16589	30.163086
Data Encrypting	0,00046	28.74707
Data Decrypting	0,00046	13.02539

TABLE I
Latency in seconds

in this case, as the maximum data block that can be encrypted in a single step is 8 bytes and a single sample only occupies 2 bytes). The heavy difference with the WM-ECC based cryptosystem is due to the adoption of a hybrid approach relying on a fast symmetric cypher, while in the TinyPairing case an asymmetric scheme is adopted. As expected, increasing the security of the protocols is paid against a performance loss. From a network designer point of view, an encryption time of abut 29 seconds implies that we cannot choose a sample period shorter than this interval, otherwise we will loose samples.

In Figure 5 a) the packet overhead introduced by the two cryptosystems is compared with the not secure version. In order to make the security application feasible, we had to increase the payload length in both cases from the default 29 bytes. In particular, for the WM-ECC based network, we sized the packet to 80 bytes to enable the transmission of the public points and digital signatures; for the TinyPairing based network, we sized the packet to 60 bytes for the transmission of the encrypted packets. In general, packet size is a very crucial parameter as this also has a bad impact on battery consumption, usually very high during the transmission phase; it could be reduced by performing a little variation in the protocol consisting in splitting packets in two or three portions in order to fit smaller dimensions. We did not perform battery consumption evaluation yet but this parameter should be taken in consideration before deploying any wireless sensor network, especially if the environment is hostile and sensor not easily reachable.

As for resource constraints, encryption and authenti-

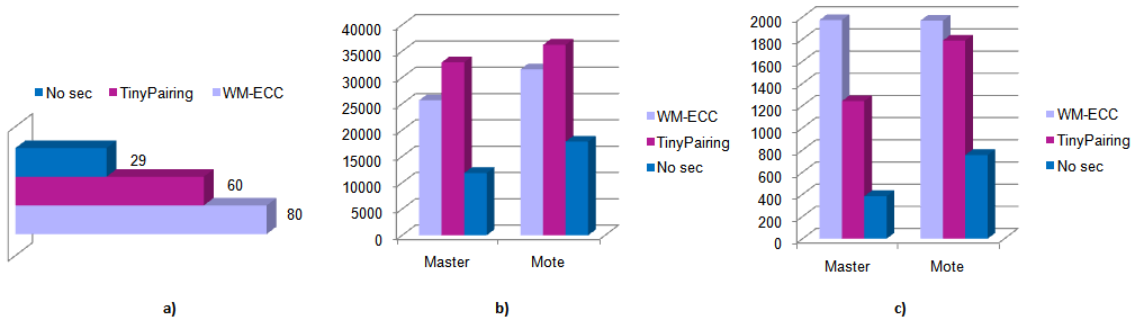

Fig. 5. a)Packet length , b) ROM and c) RAM occupancy in *bytes*

cation operations produce an additional cost in terms of memory and CPU. In Figures 5 and 5the memory overhead is reported in terms of RAM and ROM usage on master and end-nodes.

As illustrated, the secure query applications imply a higher RAM and ROM usage both on master and mote sides (even 5 times more), compared to the simple query application without security. Anyway, this is not a problem as Telosb motes have 48K bytes of memory and other common platforms, as MicaZ, have even more storage capabilities (128K).

As final remark, from a design point of view, the resulting values can be considered acceptable assuming to deploy sensor nodes that only run a monitoring application at a time, and depending on the available resources and on the security requirements, many solutions can be adopted based on different cryptographic schemes and protocols.

V. Conclusion and Future works

In this paper we addressed the security issues arising in a complex distributed monitoring infrastructure, focusing in particular on the transport level, responsible of the node-to-node communication in a sensor network. We compared two different cryptosystems relying upon the WM-ECC library and the TinyPairing library adopted to ensure confidentiality, integrity and authentication requirements. We integrated our secured networks into SeNsIM-SEC and carried out an evaluation of the overall system performances, in terms of resulting average response time, memory usage and packet length. Such evaluation highlighted the introduction of an overhead due to cryptographic operations and gave us the possibility to discuss constraints that should be taken in consideration before designing and deploying any WSN. As a future development, we plan to further analyze and compare more solutions to derive design criteria on the basis of the performance and security trade-off.

References

[1] F. Amato, V. Casola, A. Gaglione, A. Mazzeo,*A semantic enriched data model for sensor network interoperability*, In Simulation Modelling Practice and Theory, 2010, (in Press) Elsevier.

[2] V. Casola, A. De Benedictis, A. Mazzeo and N. Mazzocca. *SeNsIM-SEC: security in heterogeneous sensor networks*, in IEEE Proceedings of the 6th Conference on Network Architectures and Information System Security (SAR-SSI11), La Rochelle, May, 2011

[3] C. Karlof, N. Sastry, and D. Wagner, *TinySec: A Link Layer Security Architecture for Wireless Sensor Networks*, In Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD, November 2004.

[4] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, *MiniSec: A Secure Sensor Network Communication Architecture*, In Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN 2007), April 2007.

[5] ZigBee Alliance, *Zigbee specification*,Technical Report Document 053474r06, Version 1.0, ZigBee Alliance, June 2005.

[6] R.L. Rivest, A. Shamir, and L.A. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, In Communications of the ACM XXI (2), pp. 120-126, 1998.

[7] Certicom Research, *Standards for efficient cryptography*, SEC 1: Elliptic Curve Cryptography", Version 1.0, September 20, 2000. Certicom Research. Standards for efficient cryptography.

[8] D. Boneh and M. Franklin, *Identity-based encryption from the Weil pairing* in Advances in Cryptology (CRYPTO 2001), LNCS 2139, pp. 213-229, 2001.

[9] D. Boneh, H. Shacham, and B. Lynn, *Short signatures from the Weil pairing* in Advances in Cryptology (ASIACRYPT 2001), pp. 514-532,2001.

[10] D. Boneh and X. Boyen, *Short signatures without random oracles and the SDH assumption in bilinear groups*, J.Cryptology, 21(2), pp. 149-177, 2008.

[11] X. Xiong, D. S. Wong, X. Deng. *TinyPairing: A Fast and Lightweight Pairing-based Cryptographic Library for Wireless Sensor Networks*, in Proceedings of the IEEE Wireless Communications and Networking Conference (IEEE WCNC10), Sydney, Australia, April 2010

[12] L. B. Oliveira, D.o F. Aranha, C. Gouvêa, M. Scott, D. Camara, J. Lopez, and R. Dahab. *TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks*. In Computer Communications. 2010.

[13] H. Wang, B. Sheng, C.C. Tan and Qun Li, *WM-ECC: an Elliptic Curve Cryptography Suite on Sensor Motes*, Technical report, Oct. 30, 2007

[14] A. Liu, P. Kampanakis, and P. Ning. *TinyECC: Elliptic curve cryptography for sensor networks*, In Proc. 7th International Conference on Information Processing in Sensor Networks, IPSN 2008, St. Louis, Missouri, USA, April 22-24, 2008

[15] *TinyOS Project*, URL: http://www.tinyos.net.

[16] H. Wang and Q. Lin, *Efficient Implementation of Public Key Cryptosystems on Mote Sensors*, In Proceedings of the 8th International Conference on Information and Communications Security (ICICS 2006), Raleigh, North Carolina, USA. December 2006, pp. 519-528.