ISSN:0975-9646

Lightweight Implementation of Public Key Infrastructure for Wireless Sensor Network μPKI

Asst.Prof. Dipayan Kumar Ghosh
Assistant Professor in Computer Science & Engineering (CSE) Depratment
Calcutta Institute of Technology(CIT)
Uluberia , Howrah – 711316 , West Bengal , India .

Namita Ghosh
MCA, Haldia Institute of Technology(HIT)
Midnapore , West Bengal , India .

Abstract **Wireless sensor networks (WSN) grows and gains new in our lives ranging from military applications to civilian ones. However security in WSN was not carefully carried out, since only some symmetric encryption based protocols are proposed in literature, under the assumption that the nature of sensor nodes does not support public key encryption due to the limitation in battery and CPU power. However the new development of sensors technologies may allow more computational power and gives us the possibility to use public key encryption in WSN if the used algorithm is energy efficient such as ECC. Therefore in this paper we propose a lightweight implementation of Public Key Infrastructure (PKI). Our proposed protocol called μPKI uses public key encryption only for some specific tasks as session key setup between the base station and sensors giving the network an acceptable threshold of confidentiality and authentication.**

Key words: WSN, PKI, μPKI, key management, Public key encryption.

I. INTRODUCTION

Last dedicates have known the development of small, low cost, low power and multifunctional sensor nodes, having the possibility of sensing and collect application-specific data as temperature, pressure and movement to allow environment monitoring [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 [2].

The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance [3]. However, wireless sensor networks are now used in many civilian applications, including environment and habitat monitoring, healthcare applications, home automation, traffic control, environmental monitoring [3], or to detect and characterize Chemical, Biological, Radiological and Nuclear in some environments where the presence of human is not possible[4].

II. SECURITY IN WIRELESS SENSOR NETWORKS

Security is a very important issue when designing or deploying any network or protocol. However the recently developed networks as the wireless ones have not given the necessary attention to security when designing protocols by taking into account the specificity of these networks as the used medium and the devices constraints [5]. Thus, many security protocols were proposed trying to efficiently carry out the problem of security and the constraints of wireless networks [6]. However, in sensor network the problem of security is more challenging regarding the limitation of sensors and the area where the sensors are deployed such as battlefields [7].

The proposed schemes in literature are not secure since they use some simplified techniques to carry the limitations of sensors, given that the majority of these protocols makes use of symmetric encryption for ensuring all the security services instead of a combination of symmetric and asymmetric (public) encryption.

A. Public key cryptography

Public key cryptography was invented in seventies years, it uses two keys for both encryption and decryption. In the way that any message encrypted with one of the keys can only be decrypted with the other key. One of the keys is called private key which is kept secret by it holder, and the second one is publicly known by each entity in a given community, using these two keys, the public key cryptography can ensure both confidentiality, integrity and authentication. Often the management of generation, distribution, renewal and publication of these keys is achieved by a trust party called Certificate authority (CA) which composes what we call public key infrastructure (PKI) which is recognized as the most efficient and powerful tool to ensure key management in conventional networks. However PKI is omitted from the use in WSN, because of its great consumption of energy and bandwidth which are very crucial in sensor network, and all the most known solution given in literature use symmetric encryption which is more power saving.

However, last years known the development of new cryptographic algorithms more energy efficient and giving the same threshold of security as the conventional algorithms such as RSA [8,9]. Elliptic Curve Cryptography (ECC)[10], is one of these new algorithms and it is the most promise regarding the energy and time consumption, which makes it very attractive for data encryption in WSN. ECC offers the

equivalent security with much smaller key sizes which saves memory, computational and energy power for constrained wireless devices [9].

In the other hands, the new developed sensors will be more powerful concerning the CPU and memory capacities, making public key encryption possible for small sensors in WSN.

Thus, in this paper we are going to present a lightweight public key infrastructure for WSN called μPKI. Our proposed infrastructure does not offer all the services of a conventional PKI; however it gives the necessary threshold of security to manage the distribution of session keys in a WSN, in the way that the public encryption is only used for specific services over the network to ensure authentication; however confidentiality and integrity are achieved by symmetric encryption.

B. Security services

– ***Confidentiality:*** ensures that the exchanged data is kept secret from any unauthorized entities over the network. It is usually achieved using symmetric encryption which is more efficient concerning its consumption of devices resources. A mechanism ensuring confidentiality must also protect information using periodic key update from long term eavesdropping trying to learn from the encrypted data flow the used encrypting key.

– ***Integrity:*** implies that the message should be un-altered during its transmission from a source to destination by any intermediate sensor or malicious node. This is usually done in conventional network using MAC (Message Authentication Code) or digital signatures.

– ***Authentication:*** is the process of identification that a receiving entity is sure that the message it receives comes from a legitimate source, this is ensured using Public Key Infrastructure. However in WSN is usually done by pre-distributing some bootstrapping information used after to authenticate sensors by the base station.

III. STATE OF THE ART

In literature exist several key management schemes trying to solve the problem of security in WSN by taking into consideration the limitations of sensors (bandwidth and energy), the majority of them are based on symmetric key encryption and some others are based on asymmetric encryption:

A. Symmetric encryption based schemes

– *Shared key:* this solution is the simplest way for securing WSN, it uses a single shared key to encrypt traffic over the network, and this key may be periodically updated to ensure more security against eavesdropping. As any other scheme based on single key, this scheme is vulnerable against capture attack which is more possible in sensor network, since the capture of only one sensor can compromise the shared key and then the whole network.

– *Pre-distributed keys:* these solutions assume the existence of an off-line dealer which distributes a set of symmetric keys to sensors before their deployment, for example the authors in [10] proposed a random key pre-distribution scheme for WSN in which sensor obtains a subset of symmetric keys from a large key pool. After deployment, each sensor tries to find a shared key with each of its neighbours to secure the links with them. Other works have been proposed under on the same idea in [11, 12, 13] trying to solve the problem of scalability and the manner of obtaining the session key between sensors and the base station.

– *Tinysec:* is a link layer security protocol based on symmetric key encryption, TinySec [14] 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.

– *SPINS*: Perrig and al. proposed SPINS, a suite of security protocols optimized for sensor networks [15]. SPINS has two secure blocks, namely Secure Network Encryption Protocol (SNEP) and μTESLA, which can be run over the TinyOS operating system. SNEP is used to provide confidentiality through encryption and authentication; while μTESLA is used to provide authentication for broadcasted data.

– *Cluster based protocols:* these protocols are based on clustering, which mean that the whole network is divided into clusters [16,17], then 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.

B. Public key based schemes

– *Simplified SSL handshake:* In [9], the authors give the energy cost analysis of a simplified version SSL [18] applied to WSN, which reduces the amount of exchanged data between any pair of nodes to save energy and bandwidth.

The simplified handshake is used to setup a secure key between any two sensors in the network as the one in SSL [18].

As a brief analysis of this scheme, it seems that it is not energy saving since a handshake between each pair of sensors is too expensive concerning the amount of exchanged data. Therefore this scheme can not be applied to mobile sensor networks, since the mobility of sensors needs new handshake at each time a sensor changes its position and therefore its neighbour sensors, which consumes lot of energy.

– *TinyPK:* The TinyPK system described in [19] 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 [14], in order to deliver secret key to that underlying protocol. To do this, they implement the Diffie-Hellman key exchange algorithm.

As said above using a session key between each pair of sensors is not efficient and it consumes lot of energy and network bandwidth for the setup of the session key beyond of the energy consumed by the encryption algorithms. Using this scheme as an end-to-end security mechanism may be energy efficient however Diffie-Hellman key agreement is very

sensitive to man in the middle attacks which can be easily performed in such situation.

– *Simplified Kerberos protocol:* The authors in [20] proposed an adapted version of Kerberos [21] 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.

This scheme is very vulnerable against capture attacks to which sensor are very often exposed, and as the previous work the handshaking is not energy saving and it may consume lot of network resources if the trusted third party is far from the pair of nodes.

IV. ENCRYPTION ALGORITHMS

A. *Elliptic Curve Cryptography*

The ECC algorithm [22] can be classified as the one of the most efficient asymmetric algorithms regarding its energy cost as well as its encryption speed [9], making it the base of future key management and security protocol for WSN and any other wireless ad hoc network.

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 [23], 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.

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

Table 1 Energy cost of digital signature (mJ)

B. *Message authentication codes (MACs)*

Is the common solution to ensure integrity and authentication of messages in conventional networks [18]. A MAC can be viewed as hash function applied on data packets, resulting on a digest which is encrypted by the session key shared between the two entities, the encrypted digest is called MAC and it is sent with the original packet in the same message. A receiver sharing the same session key can verifies the integrity of the message by computing the MAC value and compares it with the received one if the verification fails; this means that an adversary has altered the packet during its transmission over the network.

C. *Symmetric cryptography*

Is a cryptographic method employing a single key for both encryption and decryption [18]. The use of a single key makes the decryption process a simple reversal of the encryption process. In literature, there exist lot of symmetric algorithms such as RC4, DES and AES. In our protocol we do not propose any algorithm to be used nor the method to implement it (hard or soft), which are let for the implementation and the specificity of the environment.

V. µPKI FOR WSN

In this section we are going to give an overview of µPKI (Micro Public Key Infrastructure). µPKI is a lightweight implementation of PKI for WSN since it only implements a subset of a conventional PKI services.

In µPKI, only the base station needs to be authenticated using a pair of keys. The public one is used to authenticate the base station by the sensors in the network, while the private key is used by the base station to decrypt some data sent by sensors which ensure its confidentiality.

A. *Network architecture*

We conceder a WSN composed of a set of sensor nodes wirelessly connected to each other, this sensors are used to forward the collected report to a centralized authority or base station Figure 1.

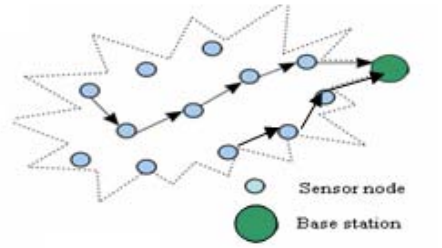

Figure 1 sensor network architecture

For the implementation of µPKI, we assume that:

- The base station have more computational and energy power compared to sensors.
- The base station has a pair of keys (private and public key).
- Each sensor is capable to use symmetric and asymmetric encryption, by implementing (hard or soft) each of these operations.
- Each sensor has the capacity to save at least the public key of the base station and a session key used for data encryption.
- Each sensor node gets the public key of the base station before deployment from an off-line dealer.

B. *µPKI System bootstrapping*

Before the deployment of the WSN, we suppose that an off-line dealer distributes the public key of the base station to each sensor in the network, which means that only legitimate sensors have the possibility to authenticate the base station trough its public key, this public key is used after in the handshake between the base station and sensors, since each link between any sensor node and the base station is secured using a symmetric session key which is periodically updated.

Two handshakes exist in µPKI, the first one between the base station and sensors intended to secure end to end transmission between them. However the second one is intended to secure sensor to sensor communication, this handshake is established

trough the cooperation of the base station which plays the role of authenticator between sensors during this phase.

C. *Base station to Sensor nodes Handshake*

This handshake is very simple and efficient, aims to setup a session key between the base station and any sensor over the network used for end to end traffic encryption between these two entities. We suppose that a sensor node needs to setup a secured link with the base station using µPKI in order to transmit some data to the base station, thus both the base station and the sensor node collaborate to execute the following steps:

1- *Generation of the session key,* As we have said links between the base station and sensors are secured using symmetric encryption, therefore any sensor willing to secure its transmission with the base station, generates a random key, encrypts it with the public key of the base station, already distributed to sensors by an off-line dealer. It embeds the encrypted key in a regular message Figure.2 and sends to the base station using the underlying protocol.

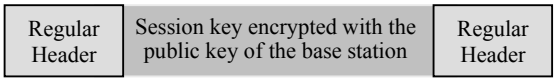

Figure.2 structure of session key's message

2- When the message containing the session key is received by the base station, it decrypts this message using its private key and saves the session key in a global table where are saved all the session keys corresponding to each sensor in the network. A global table is maintained by the base station and contains the pairs of sensors' identifier and the corresponding session key.

3- The base station encrypts an OK message using the established session key and sends it to the corresponding node; this Ok message is a challenging message ensuring the authenticity of the base station, since if this message is a successfully decrypted by the sensor using the key generated in step 1 means that the session key setup is successful Figure.3, otherwise an attack is assumed and therefore a new attempt is launched, by the sensor node to establish a new session key.

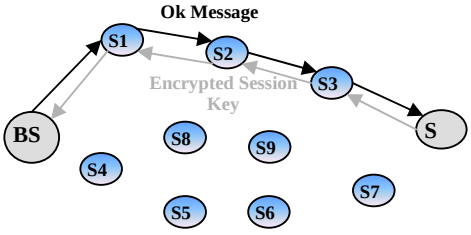

Figure 3 Session key establishment

The purpose of any handshake is the setup of a secure tunnel between two or more entities in a given community. As we can observe µPKI handshake ensures a great level of security since the session key sent to the base station over multi hops link can not be decrypted by any malicious sensors, because it is encrypted by the public key of the base station which means that only the base station can retrieves this key using the corresponding private key, as well as the Ok message which can only be decrypted using the true session key which guaranties an acceptable level of security due to the use of both symmetric and asymmetric encryption. After the establishment of this session key the sensor and the base station begin to use it for data encryption until the next key update.

D. *Sensor to Sensor handshake*

After the establishment of the session key between each sensor and the base station, we suppose that some sensors need to establish a secure channel between them for any purpose. To do so, both the base station and the sensors execute the following steps:

1- One of the two sensors sends a request to the base station in order to establish a secure tunnel with the other sensor. This request contains the identifier of the corresponding node.

2- When receiving this request the base station generates a random key for this purpose, it encrypts a copy for each sensor using the corresponding cryptographic key, and sends it embedded in a message using the underlying routing protocol to each sensor.

3- When receiving the new key by sensors they begin to use it to secure data transmission between themselves.

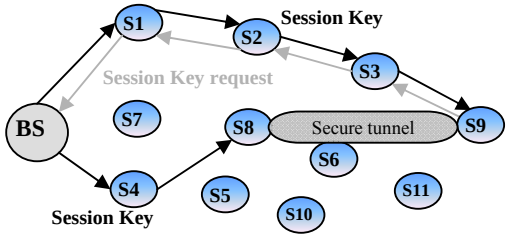

Figure 4 Sensor to sensor handshake.

E. *µPKI functioning*

After the achievement of the handshake, each two entities have a unique session key used to guaranty the confidentiality of the exchanged traffic using symmetric encryption.

In order to guaranty the integrity and the authenticity of the exchanged data between each communicating parties, we propose to apply on each sent packet a MAC function using the same session key. Hence, each communicating party verify the integrity and the authenticity of each packet by verifying the joined MAC, if the verification fails this means that an attacker has altered this packet, therefore a mechanism is launched as multi-path routing to avoid this attacker. Otherwise the base station launches any mechanism to detect and exclude this sensor from the network, if it exists.

As we can observe in figure 5, the original structure of the packet is kept unchangeable; we only join to the original packet the MAC applied on the data packet.

Figure 5 Data packet structure in µPKI

F. µPKI Key Update

A key update tries to prevent long term attack aiming to extract the encrypting keys by analysing the encrypted traffic over the network for long time, in a WSN an automatic key update must be defined, since a network can be deployed for many days or months. Therefore, in µPKI we propose to use a periodic key update for each established session key.

The key update is initiated by the sensor node by launching new handshake; the period of the key update is relative to the key length and the complexity of the used algorithm which means that this period is fixed by the administrator of the WSN.

G. Joining the Network

If a new node wants to join the network, the administrator of this network must load the public key of the base station into this node, after getting the public key of the base station the new sensor can automatically launch a handshake and join the network if there is any report to send.

VI. ANALYSIS

A. Security services

Scalability: this propriety deals with network widening is possible with µPKI, since µPKI manages the increasing number of sensor nodes by new handshakes and a new entry is created in the global table of the base station to manage this connection.

Confidentiality: this aspect is ensured by the use of symmetric encryption to encrypt the exchanged traffic between the base station and sensors. The confidentiality is enforced using periodic key update to prevent long term attacks.

Authentication: in µPKI we have tried to ensure authentication by using the public key cryptography at the level of the base station the authority which needs to be authenticated by sensors since all the WSN reports are sent to this base station. Consequently, we have ensured its authentication using a public key pre-installed in each deployed sensor.

Integrity: the integrity in µPKI is ensured using MAC (Message authentication codes) computed and joined to each sent packet between the base station and any sensor over the network as well as between sensor if there is any communication.

B. Energy cost analysis of µPKI

The energy cost of any key management scheme is determined by the energy required for the execution of cryptographic primitives and the energy needed for transmitting the encrypted data. According to [9], the transmission of a single byte of data requires 59,2µJ and 28,6µJ for reception.

The number of messages needed to be sent or received by a sensor for µPKI handshakes (Base station to Sensor or Sensor to Sensor handshake) are two messages, the size of each message is between 64 to 256 bits (according to session key length), added to 256 bits which is the size of the underlying protocols data checksum, node's IDs and protocol headers. Thus, the maximum size of each µPKI packet is 512 bits, the energy needed for transmitting such packet is 3,78mJ and 1,83mJ for receiving it.

As described in section 5 for Base station to Sensor handshake a sensor needs to send one message to the base station containing the session key (3,78mJ) and receive the Ok message sent by the base station (1,83mJ). Which is added to the cost of encrypting the session key using the public key of the base station which is 2,82mJ according to [9], as well as the energy needed to decrypt the Ok message sent by the base station which is 0,039mJ according to [9] if the used algorithm is AES and the using 128 bits key length. The total energy cost of µPKI handshake is 28,46mJ.

The Sensor to Sensor handshake is less energy consuming, since only one message needs to be send as a handshake request and one message for receiving the session key, added to the cost of decrypting the received message to retrieve the session key sent by the base station. Thus, the total consumed energy is 3,66mJ for received and sent data and 0,039mJ for decrypting it which results on 3,70mJ for the whole handshake.

Operations		Energy(mJ)
Base station to Sensor handshake	Encrypt session key	22,82
	Send session key	3,78
	Receive session key	1,83
	Decrypt Ok message	0,039
Sensor to Sensor handshake		3,70
Total energy cost		32,16

Table 2 Energy cost of µPKI (mJ)

Compared to the energy cost of the simplified Kerberos [20] and SSL [9] presented in section 3, which are respectively between 39,6mJ and 47,6mJ for simplified Kerberos[20] and 93,9mJ for simplified SSL[9] it seems that µPKI is more energy saving, which make it applicable for WSN. In addition to this it also guaranties a great threshold of security by using periodic key update and public key cryptography.

VII. CONCLUSION

In this paper we have presented a Public Key Infrastructure for wireless sensor network called µPKI. µPKI tries to solve the problem of security in WSN by the use of public key cryptography as a tool for ensuring the authenticity of the base station. µPKI is composed of two phases, the first is the µPKI sensor to base station handshake in which the base station and a given sensor node setup a session key to secure end to end link between them, this handshake is protected and authenticated using the public key of the base station. The second phase is the use of this session key for data encryption to ensure confidentiality and ensuring the integrity of the exchanged data using the MAC joined to each packet. We have also proposed sensor to sensor handshakes in order to establish secure tunnels between each two sensors; this

handshake is managed and supervised by the base station. For more security a periodic key update is defined for the session key. Compared to other PKI, µPKI is energy efficient and gives a considerable threshold of security.

REFERENCE

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", Computer Networks, Vol. 38, No. 4, pp 393–422.

[2] 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.

[3] David Culler, Deborah Estrin, and Mani Srivastava, "Overview of Sensor Networks", IEEE Computer society, Vol. 37, No. 8, pp. 41-49, 2004

[4] Carlos F.Garcia-hermandez and al, "Wireless sensor networks and applications International Journal of Computer Science and Network Security, Vol.7, No.3, pp. 264-273, March 2007..

[5] R. Ramanathan and J. Redi, "A Brief Overview of Ad Hoc Networks: Challenges and Directions", IEEE Commun. Mag., vol. 40, no. 5, pp. 20-22, 2002.

[6] Benamar KADRI, Abdallah M'HAMED, Mohammed FEHAM. "Secured Clustering Algorithm for Mobile Ad hoc Networks", International Journal of Computer Science and Network Security, Vol.7, No.3, pp 27-34; March 2007.

[7] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks", Communications of the ACM, vol. 47, no. 6, pp. 53--57, 2004

[8] N. Gura, A. Patel, A. Wander, H. Eberle, S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs", in Proceedings of the Sixth Workshop on Cryptographic Hardware and Embedded Systems (CHES'04), Cambridge, MA, USA, pp. 119-132, 2004.

[9] 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.

[10] 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.

[11] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," In IEEE Symposium on Security and Privacy, Berkeley, California, pp. 197–213, May 2003.

[12] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," In ACM CCS 2003, pp. 62–72, October 2003

[13] D. Liu and P. Ning, "Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks," in 2003 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), October 31, 2003 George W. Johnson Center at George Mason University, Fairfax, VA, USA.

[14] 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.

[15] Perrig and al. "SPINS: Security Protocols for Sensor Networks", Mobile Computing and Networking, pp.189-199, 2001 Rome, Italy.

[16] 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.

[17] S. Basagni, K. Herrin, E. Rosti and Danilo Bruschi. "Secure Pebblenets", Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, pp. 156 – 163, 2001.

[18] Bruce schneier, "cryptographie appliquée algorithms, protocoles », 2nd edition wiley, 2001.

[19] 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.

[20] 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).

[21] 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.

[22] D. Hankerson, A. Menezes, S. Vanstone, "Guide to Elliptic Curve Cryptography", Springer-Verlag New York, ISBN 0-387-95273-X, Inc. 2004.

[23] Crossbow Technology Inc., Processor/Radio Modules, http://www.xbow.com/