Applicability of Public Key Infrastructures in Wireless Sensor Networks

Rodrigo Roman and Cristina Alcaraz

Computer Science Department,
University of Malaga, Spain
{roman,alcaraz}@lcc.uma.es

Abstract. Wireless Sensor Networks (WSN) are becoming a key technology in the support of pervasive and ubiquitous services. The previous notion of "PKC is too expensive for WSN" has changed partially due to the existence of new hardware and software prototypes based on Elliptic Curve Cryptography and other PKC primitives. Then, it is necessary to analyze whether it is both feasible and convenient to have a Public Key Infrastructure for sensor networks that would allow the creation of PKC-based services like Digital Signatures.

Keywords - Wireless Sensor Networks, Public Key Cryptography, Public Key Infrastructure.

1 Introduction

Wireless Sensor Networks [1] can be considered as a key technology to support pervasive and ubiquitous services. They can be applied to a wide number of areas: such as farmland monitoring, smart office, detection of out-of-tolerance environmental conditions, emergency medical care, wearable smart uniforms, etc. However, these networks are quite difficult to protect, because every node becomes a potential point of logical and physical attack.

In this context, it would be extremely useful to have a cryptographic primitive such as Public Key Cryptography (PKC) in order to create services such as Digital Signatures. The use of PKC in sensor networks has been usually considered as "nearly impossible", but at present some studies [4] have started to consider the possibility of utilizing PKC in a highly-constrained networks. It is then the purpose of this paper to review the state of the art of PKC for sensor networks, and to analyze if it is both feasible and convenient to have a working Public Key Infrastructure in a sensor network environment.

The rest of this paper is organized as follows: In section 2 the architecture of a wireless sensor network is explained, alongside with how PKC could influence on solving some major security problems. In section 3, the major PKC primitives that could be applied to constrained environments such as sensor nodes are presented and studied. Finally, in section 4, there is a deep analysis of the applicability of Public Key Infrastructures to a sensor network environment, and in section 5, the conclusions are presented.

2 Wireless Sensor Networks

A Wireless Sensor Network, as a whole, can be seen as the "skin" of a computer system, since it is able to provide any physical information of a certain region or element to any external system. The ability of measuring their environment is not the only benefit of these networks: thanks to the wireless capabilities and the limited computational power of their elements, they are easy to set up, are capable of self-configuring themselves, and are relatively inexpensive. The main elements of a sensor network are the sensor nodes and the base station - the "cells" of the system and its "brain".

Sensor nodes are small and inexpensive computers that have limited computational and wireless capabilities: a typical sensor node uses a microcontroller of 8Mhz with 4KB of RAM and 128KB of ROM, and incorporates a transceiver compliant to low-power, low duty standards such as IEEE 802.15.4. On the other hand, the base station is a powerful, trusted device that acts as an interface between the user of the network and the nodes. Regarding their internal configuration, the nodes of the network can group themselves into clusters where all the organizational decisions inside a cluster are made by a single entity called "cluster head" (hierarchical configuration), or all the nodes can participate in both the decision-making processes and the internal protocols (flat configuration).

In a sensor network, amongst other issues, it is extremely important to provide certain basic security mechanisms and protocols in order to avoid attacks from malicious adversaries [3]. It was recently when Public Key Cryptography (PKC) started to be considered as a viable solution for this purpose. Since, in most cases, a node does not know in advance who will be on its neighborhood, PKC can be used for both authenticating such nodes and for allowing the secure exchange of pairwise keys. Any procedure that requires the participation of the base station can also take advantage of these primitives. For instance, it is possible to securely distribute new code to the nodes of the network if it has been previously signed by the base station. Lastly, there are many other services that can effectively use PKC: authenticated broadcast, data source authentication in data aggregation, privilege delegation, etc.

3 Public Key Cryptography Primitives for Sensor Networks

3.1 Existing PKC Primitives

The computational requirements of PKC primitives are quite expensive in comparison with other cryptographic primitives, such as Symmetric Key Encryption (SKE). For instance, the most popular algorithm for public key encryption, RSA [5], is quite inefficient when implemented in sensor nodes. However, there exists other PKC approaches based on various mathematical problems that can be specially useful in highly-constrained environments. The first example is the Rabin signature algorithm [6], proposed by Michael Rabin in 1979. It is very similar to

RSA, but its main advantage is the speed of its encryption and signature verification operations, which are based on a simple squaring operation. A disadvantage is the signature size, though: a single signature requires 512 bits.

One of the most suitable asymmetric cryptography primitives for WSN, the Elliptic Curve Cryptography cryptosystem (ECC) [7], was discovered in 1985. ECC is based on algebraic concepts related with elliptic curves over finite fields $\mathbb{F}_p$ or $\mathbb{F}_{2^m}$. The ECC's security is based in computing the equation $a^b = c$ given a and c, known as the discrete logarithm problem, and the main ECC primitive operation is the scalar point multiplication. The major benefit of ECC is the size of its keys (160 bit against 1024 bit in RSA [10]) and its speed while optimized.

The asymmetric algorithm NTRUEncrypt [8], and its associated signature scheme NtruSign, is based on arithmetic operations in a polynomial ring $R = \mathbb{Z}(x)/((x^N - 1), q)$. Its security is based on the hardness of resolving the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). NTRUEncrypt is much faster both for encrypting and for verification operations than RSA, since it uses simple polynomial multiplications. On the other hand, it also shares Rabin's scheme weakness: its signature requires 1169 bits.

The most recent asymmetric approach is the *multivariate public-key cryptosystem*, also known as $\mathcal{MQ}$-schemes [9]. Its security is centered in resolving $w = V^{-1}(z) = (\omega_1, ..., \omega_n) \in K^n$ given a quadratic polynomial map $V = (\gamma_1, ..., \gamma_m) : K^n \rightarrow K^m$. Its signature operations are extremely fast, but there is a significative storage cost for restricted environments due to the size of the keys in RAM. Concretely, it is necessary to book 879 bytes for the private key and 8680 bytes for the public key.

3.2 HW and SW prototypes

	ECC				NTRU	Rabin	$\mathcal{MQ}$
	Wolkerstorfer	Kumar & Paar	Gaubatz	Batina	Gaubatz	Gaubatz	Yang
Gates	23000	12000	18720	12000	3000	17000	17000
Frequency	68.5MHz	13.5Mhz	500khz	500kHz	500kHz	500kHz	100kHz
Point Mult.	9.98ms	18ms	$\sim$ 400ms	115ms	—	—	—
Encryption	—	—	—	—	58ms	2.88ms	—
Decryption	—	—	—	—	117ms	1.089s	—
Signing	—	—	—	—	234ms	1.089s	44ms
Verifying	—	—	—	—	58ms	2.88ms	—

Table 1. Summary of Hardware prototypes for PKC.

A summary of the different HW prototypes for PKC in sensor networks can be seen in table 3.2. In 2005, Gautbatz et. al. proposed in [13] several PKC hardware implementations of Rabin's scheme, NtruEncrypt and ECC primitives. All of theses implementations work with an operational frecuency of 500KHz, and they were designed having the node's hardware limitations in mind. Other

ECC prototypes for more powerful nodes were developed by Wolkerstorfer et. al. in [14] and Kumar and Paar in [15]. Finally, in 2006, Batina et. al. in [16] improved the previous ECC implementations, and Yang et. al. in [17] proposed an implementation for Multivariate public key cryptosystem.

	TinyECC		WMECC		TinyWMECC	
	Micaz	Telosb	Micaz	Telosb	Micaz	Telosb
Test - ROM size	28266	26048	57982	46156	29734	25774
Test - RAM size	2306	2327	1685	1657	1643	1599
ECC init	1.837s	-	1.809s	1.744s	1.809s	1.744s
ECDSA init	3.550s	5.225s	0s	0s	0s	0s
Pub. Key Gen.	1.788s	-	1.261s	1.425s	1.261s	1.425s
Signature	1.916s	4.361s	1.348s	1.498s	1.348s	1.498s
Verification	2.431s	5.448s	2.017s	2.207	2.019s	2.209s

Table 2. Software implementations of ECC

Regarding software implementations, as of 2007 the most important ECC libraries for sensor networks are TinyECC by Liu and Ning [12], and WMECC by Wang and Li [11]. These libraries work over the micaz and telosb motes in the "de-facto" standard Operative System for WSN, TinyOS, and their performance can be seen in table 3.2. Both libraries have different implementation approaches, although it is noteworthy that TinyECC has an improved SHA-1 function that allows it to have a reasonable code size compared with WMECC. Fortunately, taking advantage of the component capabilities of TinyOS and the optimized SHA-1 function in TinyECC, it was possible for us to improve the existing WMECC library by changing its SHA-1 function. This improvement, named TinyWMECC, completely solves the code size problem, and has been included recently into the main WMECC code branch.

4 Public Key Infrastructures in Sensor Networks

4.1 Adapting PKI for sensor networks

The use of PKC alone is not enough for protecting a WSN: it is necessary to have a Public Key Infrastructure that can be able to establish a trusted identity, amongst other things. The major components of a PKI, according to the PKIX model [2], are the following: the clients, which are the users of a PKI certificate; the Certification Authority (CA), which establishes identities and creates digital certificates; the Registration Authority (RA), which is responsible for the registration and initial authentication of the clients; and the Repository, which stores the certificates and the Certification Revocation Lists (CRLs). In order to provide the services of a PKI, such as initialization and certification, these components and their functionality must be mapped to the entities of a wireless sensor network.

It is not trivial to apply a PKI to a wireless sensor network, though. The architecture of these types of networks have several distinctive features regarding its initialization and maintenance. For example, all nodes have to be configured by the base station in a secure environment before their final deployment in the network. Also, the architecture of the network is highly decentralized, where the autonomous sensor nodes collaborate towards a common goal, but all the critical information produced by the network must be sent to the Base Station.

Although a sensor network is highly decentralized by nature, it is easy to notice that there is a central system, the base station, that takes the role of initializing the nodes of the network and interacting with the data provided by all these nodes. Therefore, it is clear that the base station can be considered as the Certification Authority. It is the base station, then, the entity responsible for creating the digital certificates that associate the identity of a node with its public/private key pair. Moreover, the base station can also take the role of Registration Authority, since it is in charge of assigning the identity of all the nodes of the network before the deployment of the network. As a side note, the base station can also create the public/private key pair of a node, as it is not efficient for a node to create its own key, and the base station is trustworthy.

Although the base station may also act as the Certificate Repository, this is not practical for sensor networks. Since most sensor nodes need to route their information through other nodes in order to send information to the base station, and the costs of doing so are quite high in terms of energy and time, it is better to adopt a decentralized solution for retrieving the certificates. As a result, every node will have its own certificate, and will provide it to any neighbor that requests it. This exchange can be done in the first steps of the lifetime of the network.

In order to deploy a PKI, it is also obligatory to select an appropriate hierarchy model. Fortunately, in most cases the architecture of a sensor network is extremely simple: one base station that serve as an interface to hundreds or thousands of sensor nodes, which only know and can communicate with the nodes belonging to the same network. Therefore, it is safe to consider that a sensor network will use a simple hierarchical PKI architecture, with only one root CA.

The basic functionality of a PKI, that is, registration, initialization, key generation, certification, and certification retrieval, are performed in the following way: The base station creates the public/private key pair of a sensor node, assigns an unique identification to it, and creates the digital certificate that links that unique identification with its public key. Later, it initializes the contents of the sensor node (such as configuration data and internal programming), including the certificate of the node and the certificate of the root CA (i.e. the base station itself). Later, when a node retrieves the certificate of one of its neighbors, it will be able to check its validity using the root CA certificate.

4.2 Other PKI functions in sensor networks

Thanks to the characteristics and peculiarities of the architecture of wireless sensor networks, it is possible to map the entities required by a PKI in the

elements of a sensor network, providing as a result some of the basic functions of a PKI. However, there are still other PKI functions whose applicability must be discussed, such as Key Pair Recovery, Key Update, Cross Certification, and Key Revocation. Some of these functions are not required for a sensor network context, whereas other functions could be important in certain scenarios.

For example, the issues of key archival and key pair recovery are simple to solve. Since the base station is considered as a fully trusted entity, all keys pairs can be stored inside it, or in another secure server that interacts with it for redundancy purposes. On the other hand, the issue of cross certification is a bit more complicated. For a typical sensor network, with only one base station that behaves as the root CA, it is not necessary to use cross-certificates. However, there are some scenarios where more than one base station can be able to control the network. Moreover, as seen in section 2, there are some hierarchical infrastructures where a set of "cluster heads" control a cluster of nodes.

The additional base stations can be static, also serving as an interface to the functionality of the network, or mobile, directly querying the nodes about their status. Mobile base stations can behave as any other node inside the network, except that they should have a short-lived certificate signed by the main base station, with enough privileges to influence over the nodes' operations. Regarding static base stations, there are usually only a few of them, thus it can be possible to simply preload their certificates, signed by the root CA, into all nodes. Finally, it is not necessary to consider a cluster head as a CA, since it has no need to either produce or sign any certificate of the other members of its cluster. As a conclusion, there is no need to use cross-certificates, even in these complex scenarios.

Regarding Key Revocation and Key Update, there may be some situations in which it is important to use these services. For example, if one node is subverted by an adversary but is discovered by the network, the base station may choose to revoke its certificate. Furthermore, the base station can introduce a new node into the network with a new certificate that replaces the malicious one. Updating the certificate of a certain node is an easy task, since the human administrator of the network has to physically obtain the node for putting inside the new certificate, alongside with the private key associated with it.

Alerting the nodes about the revocation of the previous certificate is not easy, though. It is prohibitive for the nodes to retrieve a Certificate Revocation List from the base station (pull model), since querying the base station is a time-consuming and energy-consuming process. A better solution would be to use a online revocation notification mechanism (push model), where the base station alerts the nodes of the network that a certain certificate has been revoked. Upon receiving this authenticated message, the nodes of the network can then request the public key of the node that had its certificate revoked. A malicious node will not be able to provide a valid certificate, whereas the certificate of a legitimate node will be accepted.

An aspect related to node revocation, and mentioned in mobile base stations, is the existence of a validity period inside all certificates. Nevertheless, for short-

lived networks, the context of the application ("deployment") is more important than the expiration date. For instance, a short-lived network may measure the level of ambient noise in a certain area for a week or more (*Deployment A*), but later the same nodes from that network can be reutilized in another area (*Deployment B*). It should be then more efficient to identify the deployment rather than the expiration date, and discard any certificate that belongs to a previous deployment (e.g. a node belonging to the *Deployment B* does not accept any certificate that was created during *Deployment A*). The root CA, then, has to assign new certificates to all the nodes before deploying them in a new area.

In long-lived networks, such as a network that monitors the overall conditions of a vineyard for an entire year, this notion of "deployments" may not be enough, since the nodes will be continuously monitoring an environment for a long period of time. Nevertheless, the expiration date of the certificates used in these networks should not allow a situation where the nodes are not able to use the PKI services. What expiration date should be chosen is unknown at present due to the lack of long-lived real-world deployments, but it is safe to assume that there is no danger in configuring the certificates of the network to have no expiration date. If there is no external influence, the network will function properly all its lifetime. And if there is any malicious influence, such as the destruction of the base station, the owner of the network can "reboot" the whole network, reconfiguring it and labelling it as a new "deployment".

5 Conclusions

From "Public-key cryptography is right out" to "Public-key is no big deal" [18], it is clear that there is a possibility to incorporate in the near future public key-based services such as Digital Signatures in wireless sensor networks. Therefore, as explained in this paper, the inclusion of a Public Key Infrastructure for sensor networks should be seriously considered. This is an immature area that is full of interesting research problems, like the coexistence of a PKI with other Public-key based schemes such as Homomorphic Encryption [19] and Identity-Based Cryptography [20].

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci (2002). *Wireless sensor networks: a survey.* Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 38, no. 4, pp. 393-422, March 2002.
2. Public-Key Infrastructure (X.509) (pkix) Charter. http://www.ietf.org/html.charters/pkix-charter.html
3. J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary. *Wireless Sensor Network Security: A Survey.* Security in Distributed, Grid, and Pervasive Computing, Editor: Yang Xiao, Auerbach Publications, CRC Press, ISBN 0-849-37921-0, 2006.
4. J. Lopez. *Unleashing Public-Key Cryptography in Wireless Sensor Networks.* Journal of Computer Security, vol 14, no. 5, pp 469-482, 2006.

5. R. Rivest, A. Shamir, L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Communications of the ACM, vol. 21, no. 2, pp. 120126, 1978.
6. M. O. Rabin. *Digitalized Signatures and Public Key Functions as Intractable as Factorization*. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology (1979).
7. I. Blake, G. Seroussi, N. P. Smart. *Elliptic Curves in Cryptography*. Cambridge University Press, ISBN 0-521-65374-6, 2000.
8. J. Hoffstein, J. Pipher, J. H. Silverman. *NTRU: a Ring based Public Key Cryptosystem*. In proceedings of the 3rd Algorithmic Number Theory Symposium (ANTS 1998), Portland (USA), June 1998.
9. C. Wolf, B. Preneel. *Taxonomy of Public Key Schemes Based on the Problem of Multivariate Quadratic Equations*. Cryptology ePrint Archive, Report 2005/077.
10. National Institute of Standards and Technology. *Recommended Elliptic Curves for Federal Government Use*. August 1999.
11. H. Wang, Q. Li. *Efficient Implementation of Public Key Cryptosystems on MICAz and TelosB Motes*. Technical Report WM-CS-2006-07, College of William & Mary, October 2006.
12. An Liu, Panos Kampanakis, Peng Ning, *TinyECC: Elliptic Curve Cryptography for Sensor Networks (Version 0.3)*. http://discovery.csc.ncsu.edu/software/TinyECC/, February 2007.
13. G. Gaubatz, J.-P. Kaps, E. Öztürk, B. Sunar. *State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks*. In Proceedings of the 2nd IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2005), Hawaii (USA), March 2005.
14. J. Wolkerstorfer. *Scaling ECC Hardware to a Minimum*. In ECRYPT workshop - Cryptographic Advances in Secure Hardware - CRASH 2005. Leuven (Belgium), September 2005. Invited Talk.
15. S. Kumar, C. Paar. *Are standards compliant elliptic curve cryptosystems feasible on RFID?*. In Proceedings of Workshop on RFID Security, Graz (Austria), July 2006.
16. L. Batina, N. Mentens, K. Sakiyama, B. Preneel, I. Verbauwhede. *Low-Cost Elliptic Curve Cryptography for Wireless Sensor Networks*. In Proceedings of the 3rd European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2006), Hamburg (Germany), September 2006.
17. B.-Y. Yang, C.-M. Cheng, B.-R. Chen, J.-M. Chen. *Implementing Minimized Multivariate Public-Key Cryptosystems on Low-Resource Embedded Systems*. In Proceedings of the 3rd International Conference on Security in Pervasive Computing (SPC 2006), York (UK), April 2006.
18. D. Wagner. *The Conventional Wisdom About Sensor Network Security ... Is Wrong*. IEEE Security & Privacy 2005, invited panelist, Security in Ad-hoc and Sensor Networks, May 2005.
19. E. Myketun, J. Girao, D. Westhoff. *Public Key Based Cryptoschemes for Data Concealment in Wireless Sensor Networks*. IEEE International Conference on Communications (ICC 2006), Istanbul (Turkey), May 2006.
20. L. B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, R. Dahab. *TinyTate: Identity-Based Encryption for Sensor Networks*. Cryptology ePrint Archive, paper 2007/020. http://eprint.iacr.org/2007/020.pdf