SELF-CERTIFIED PUBLIC KEY CRYPTOGRAPHY FOR RESOURCE-CONSTRAINED SENSOR NETWORKS

ITAMAR ELHANANY, BENJAMIN ARAZI, ORTAL ARAZI, DEREK ROSE, HAIRONG QI*

Abstract. As sensor networks continue to become one of the key technologies to realize ubiquitous computing, promising to revolutionize our ability to sense and control the physical environment, security remains a growing concern. The resource-constrained characteristics of sensor nodes, the ad-hoc nature of their deployment, and the vulnerability of wireless communications in general pose a need for unique solutions. A fundamental requisite for achieving security is the ability to encrypt and decrypt confidential data among arbitrary sensor nodes, necessitating the generation of joint private keys. Although the advantage of public key cryptographic key-generation is widely acknowledged, offering scalability and decentralized management, the scarce resources of sensor networks render the applicability of public key methodologies highly challenging. In this respect, Elliptic Curve Cryptography (ECC) has emerged as a suitable public key cryptographic foundation in constrained environments, providing high security for relatively small key sizes.

Recent results indicate that the execution of ECC operations in sensor nodes is feasible. In an effort to promote practical adoption of ECC-based key-generation in sensor networks, this paper presents a comprehensive security methodology that significantly reduces the overall communication and computation efforts involved. The technology developed has been implemented on Intel Mote2 platforms at the University of Tennessee. The encouraging performance results obtained accentuate the practicality and scalability properties of the proposed approach.

Key words. Security in wireless sensor networks, resource-constrained cryptography, self-certified key generation, Intel Mote 2

1. Introduction. The sensor network, as a network of embedded sensing systems, has been studied extensively since the late 90s. Considerable efforts have been directed towards making them trustworthy. This is particularly true in health and military applications, where critical information is frequently exchanged among sensor nodes through insecure wireless media. Traditionally, security is often viewed as a stand alone component of a system's architecture, for which a dedicated layer is employed. This separation is a flawed approach to network security, especially in resource-constrained, application-oriented sensor networks. Although the area of network security has been studied for decades, the many unique characteristics of sensor networks have made direct application of existing methodologies impractical. In particular, the following security considerations and requirements need to be discussed in the context of sensor networks.

First, the ad-hoc nature and the extreme dynamic environments in which sensor networks reside suggests that a prerequisite for achieving security is the ability to encrypt and decrypt confidential data among an *arbitrary* set of sensor nodes. For the same reason, the keys used for encryption and decryption should be established *at* the nodes instead of using keys generated off-line, prior to deployment. This is important in order to accommodate for the dynamics of the network, as well as the environment. If a communications channel is unavailable during a particular time frame, the protocol should be sufficiently adaptive. The reliability of the links, which is closely related to the issue of channel dynamics, must be reflected by any sensor network protocol such that erroneous links do not jeopardize the integrity of the key generation process. *Second*, due to high node density, scalability is an inherent concern. Ad-hoc formation

*B. Arazi is with the Computer Science and Computer Engineering Department at the University of Louisville, KY. I. Elhanany, D. Rose, H. Qi and O. Arazi are with the Electrical and Computer Engineering Department at the University of Tennessee (e-mails: {itamar, derek, hqi, oarazi}@utk.edu, respectively.

of node clusters, hosting collaborative processing, has been a solution in achieving both fault tolerance and scalability. Consequently, an ad-hoc cluster of nodes is required to establish a joint secret key, and any solid key generation scheme must scale with respect to the number of nodes in a cluster. The *third* aspect pertains to the scarce energy resource, along with low computation capability, which are always primary concerns in security solutions for sensor networks; there is a clear need for conserving energy on each node when adopting a security protocol. In addition to the efficient utilization of energy, its *balanced* consumption across the entire network should be viewed as a primary goal in an aim to prolong the network lifetime.

2. Related Work on Security for Sensor Networks. A simple solution for key establishment has been proposed in the literature in which a single network-wide shared key is used. However, a single node in the network being captured would easily reveal the network secret key. A current mainstream effort consists of random key pre-distribution, in which a different set of pre-established keys is issued to each node, thereby reducing the probability that capturing one node will jeopardize the entire network [1][2][3]. A trivial key pre-distribution scheme is to allow each node to hold $N-1$ secret pairwise keys, each of which is known only to the node and to one of the other $N-1$ nodes (assuming there are N nodes in the network). However, the constrained memory resources and the difficulty in adding new nodes to the network limit the effectiveness of this general scheme. Other researchers have extended the original notion of key pre-distribution to include a statistical element. In particular, methods such as those proposed in assume that each sensor node receives a random subset of keys drawn from a large key pool. To agree on a key for communication, two nodes find one common key within their subsets and use that key as their shared secret key. Additional information, such as data concerning the position and/or geographical distribution of the sensor nodes, can be used to further improve the key pre-distribution concept. Although straightforward in concept, these schemes offer partial solution with respect to scalability, cryptographic robustness and the ability to append and revoke security attributes.

The problems identified in the key pre-distribution approach triggered an in-depth study of public key cryptographic key-establishment for sensor networks. Two public such procedures are commonly recognized. A *fixed* key-establishment procedure pertains to the case where two specific nodes use the same secret value (private key) whenever they wish to establish a joint key. In *ephemeral* key-establishment, the two nodes generate a different key for each session established, based on a random component introduced by each node. Ephemeral key-establishment is more secure and is generally preferred in many applications. A major issue in public key cryptographic applications concerns *certification*, which ensures the safe exchange of public keys. A Certification Authority (CA) issues a certificate, attesting to the connection between a user's public key and his ID. Verifying a certificate needs only an explicit reference to the CA's public key. An authentication procedure which is based on certification therefore needs the following values as input: the user's public key, his ID, the certificate and the CA's public key. The latter is considered to be universal and known to all parties, while the first three values are unique to each user.

To further improve the computational efficiency of the key establishment procedure, *self-certified* public key cryptography was introduced, in which a user submits its ID along with its public key, but does not submit an explicit certificate, thereby reducing communication and management overheads. In identity-based systems [4], the user's public key is its actual ID, which avoids the need for any public value

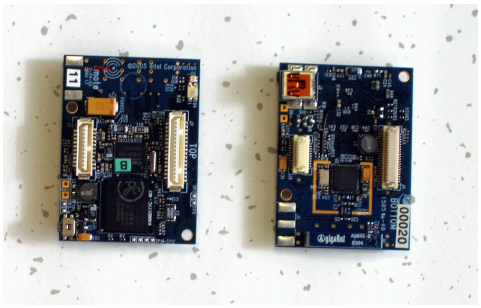

FIG. 3.1. *The Intel Mote 2 platform*

other than the user's ID. Nevertheless, an explicit reference to the CA's public key is still required. In the context of key establishment, self certification means that the authenticity of values submitted by the participating parties is *inherently embedded within the process of generating the session key.* This is in contrast to the case of explicit certification, whereby authenticity of the submitted values has to be verified prior to the actual generation of the joint session key. A well known self-certified key generation method is the MQV, adopted by the NSA.

3. Resource-Efficient Public Key Cryptography for Sensor Networks. Recently, there has been a growing effort in promoting public-key cryptography in sensor networks. Elliptic Curve Cryptography (ECC) [5] emerges as a suitable public key cryptographic foundation for sensor networks, providing high security for relatively small key sizes. Malan *et al.* [6] demonstrated an implementation of point by scalar multiplication over elliptic curves, which is the basic ECC operation in ECC, on MICA2 motes.

A need addressed by this paper and recent work by the authors [7] concerns an ECC self-certified [8] fixed key-generation, still executed using a single exponentiation. There are known ECC ephemeral-key-generation methods, in which the validity of a received ephemeral value is based on the validity of a received static value. In these cases, however, it is still necessary to provide for explicit certification of the received static value. Finally, in an effort to effectively distribute the computational load between the nodes, we propose to partition the self-certified key-generation process into secure and non-secure operations. The latter enables offloading the non-secure operations to available neighboring nodes, thereby distributing the power consumption. A novel algebraic approach for partitioning the key generation process was devised for both fixed and ephemeral key generation

The methodologies developed were implemented on the Intel Mote 2 [9] platform shown in Fig 1. The latter employs the Intel PXA271 XScale Processor running at a clock frequency ranging from 13 MHz to 416 MHz. The core frequency can be dynamically set in software, allowing the designer to carefully the adjust the timing/power trade-off so as to optimize performance of a particular application.

Figure 2 outlines the results obtained for establishing both ephemeral and fixed ECC 163-bit keys between two nodes. 163-bit keys in ECC are equivalent, from a cryptocomplexity perspective, to 1024-bit keys in RSA. The code was written in NesC targeting the TinyOS operating system. Nodes exchange messages using a 2.4 GHz embedded low-power radio transceiver. The entire process takes less than a second to complete at a clock rate of 104 MHz, with linear speed increase observed

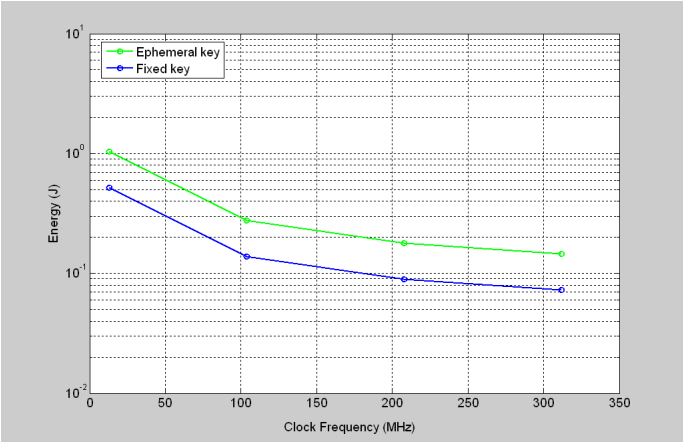

FIG. 3.2. *Energy consumption (J) for 163-bit ECC key generation on the Intel mote 2 platform*

with respect to the CPU clock frequency. As illustrated in figure 2, the methodology proposed is highly pragmatic, paving the way for broader development of resource-efficient security mechanisms for wireless sensor networks.

REFERENCES

[1] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, (Washington DC, USA), pp. 197–214, 2003.

[2] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proc. of IEEE INFOCOM 2004*, (Hong Kong, China), 2004.

[3] W. Zhang and G. Cao, "Group rekeying for filtering false data in sensor networks: A predistribution and local collaboration-based approach," in *Proceedings of the 2005 IEEE INFOCOM*, (Miami, FL, USA), 2005.

[4] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology - CRYPTO '86*, vol. 263, pp. 186–196, March 1987. Springer-Verlag.

[5] A. J. Menezes, *Elliptic Curve Public Key Cryptosystems*. Boston, MA: Kluwer Academic Publishers, 1993.

[6] D. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *Proc. of 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, (Santa Clara, CA), October 2004.

[7] B. Arazi, I. Elhanany, O. Arazi, and H. Qi, "Revisiting public-key cryptography for wireless sensor networks," *Computer*, vol. 38, no. 11, pp. 103–105, 2005.

[8] M. Girault, "Self-certified public keys," in *Advances in Cryptology–EUROCRYPT'91*, pp. 491–497, March 1991. LNCS - Springer-Verlag.

[9] R. Adler, M. Flanigan, J. Huang, R. Kling, N. Kushalnagar, L. Nachman, C.-Y. Wan, and M. Yarvis, "Intel mote 2: an advanced platform for demanding sensor network applications," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 298–298, ACM Press, 2005.