S-PkSec: An Asymmetric Key Based Security Management Scheme for Sensor Network Operation*

Md. Mokammel Haque, Al-Sakib Khan Pathan, and Choong Seon Hong

Department of Computer Engineering, Kyung Hee University
1, Seocheon, Giheung, Yongin, Gyeonggi 449-701, Korea

{malinhaque, spathan}@networking.khu.ac.kr and cshong@khu.ac.kr

Abstract

This paper proposes a public key based management scheme for secure sensor network operation namely S-PkSec (Public Key Based Security for Sensor Networks) and emphasizes detailed comparison with some similar type of schemes. Although there was a wide held belief of the incompatibility of public key cryptographic (PKC) schemes for wireless sensor networks (WSNs), some recent works have shown that, PKC or asymmetric key based schemes could be implemented for such networks in some ways. The major challenge of employing a PKC scheme in sensor network is posed by the limitations of resources of the tiny sensors. Considering this feature of the sensors, we enhance our previous work [1] with some effective comparisons and energy analysis with other two established asymmetric key based protocols. S-PkSec comprises basically of two parts; a key handshaking scheme based on simple linear operations and the derivation of decryption key by a receiver node. S-PkSec allows both base-station-to-node or node-to-base-station secure communications, and node-to-node secure communications. Analysis and simulation results show that, our proposed architecture ensures a good level of security for communications in the network and could effectively be implemented using the limited computation, memory and energy budgets of the current generation sensor nodes.

1. Introduction

Security is often viewed as a standalone component of many systems' architectures. But, in case of wireless sensor networks (WSNs), security is more than an important issue that must get commensurate emphasis. The types of services expected from wireless sensor networks often make security as the most important concern for deploying and using such types of networks. In many cases, the tiny sensors in a sensor network are used to collect specific data from particular target areas and the collected data are often considered secret and not to be disclosed in public. Hence, efficient and secure mechanisms are needed to transmit acquired data secretly to the appropriate recipients.

Sometimes wireless sensor networks carry confidential information which if exposed to the adverse units, might cause debacle for the friendly units. Especially in military applications of WSN, employing apropos security mechanisms for data transmissions is very crucial as these data could be used for taking tactical military decisions. If an adversary can thwart the work of the network by perturbing the information produced, stopping production, or pilfering information, then the usefulness of sensor networks is drastically curtailed. Likewise, for example in a disaster management related application, accurate and unmodified data are needed to predict upcoming disaster(s) and to warn the concerned people in advance about the occurrence of event(s).

Ensuring complete and a good level of security for such a type of networks however, is not a trivial task. As these sorts of networks use wireless communications, the threats and attacks against WSNs are more diverse and often large in scale. It is not possible to deal with all sorts of security threats with a single mechanism. Rather, a combination of different security schemes for a single network could be the solution. For example, an attack at the physical layer like, jamming [2] could not be handled by any key management scheme. Hence, several mechanisms at different layers could be employed at the same time to provide holistic security [3] for wireless sensor networks and side by side the level of security in the data transmission and communication phase could be increased

*This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-521-D00394). Dr. CS Hong is corresponding author.

using efficient key management schemes. Public key cryptography (PKC) could be the best choice for ensuring a satisfactory level of security for data transmissions within the network. However, the major challenge of employing a public key security scheme directly in wireless sensor network is the constrained energy, computation, and memory budgets of sensors participating in the network. Among several public key schemes, Elliptic Curve Cryptography (ECC) based algorithms have a proven and acceptable performance for low-powered sensor nodes [4], [5], [6]. Considering both the software and hardware configurations, elliptical curve based public key cryptography (PKC) has shown relatively better results on 8 bit mote platforms. However, the use of certificates in such a scheme consumes a huge amount of bandwidth and power.

Considering the special characteristics of wireless sensor networks, in this paper, we propose an efficient public key/asymmetric key based security scheme (S-PkSec) for WSN. Here S stands for Secure, Scalable Sensor network. In our scheme, we use pseudoinverse matrix for the first part while the second part is a simple method for transferring decryption key to the receiver node. Our analysis and simulation results show that our scheme demonstrates a considerable gain in the level of security and is suitable enough to be employed with the current generation sensor nodes.

The rest of the paper is organized as follows: Section 2 presents the literature review, Section 3 states the preliminaries and assumptions for our security architecture. Section 4 presents our proposed architecture and schemes, Section 5 deals with the analysis, comparison simulation results, and finally, Section 6 concludes the paper stating the achievements from this work with future research directions.

2. Related Work

Most of the works regarding public key cryptography in wireless sensor networks are conducted to fit the low-power characteristic of sensor nodes. Recently, several works like [5], [6], [7, [8] have addressed or successfully have implemented public key schemes with current generation sensors. Both from software and hardware perspectives, PKC schemes have shown reasonable performances. In this section, we discuss some of these exclusive research works and compare their contributions in this area.

[4] presents the first known implementation of elliptic curve cryptography over F_{2p} for sensor networks based on 8-bit, 7.3828 MHz MICA2 mote. The results show that, public key based scheme is viable for the modern-era sensors. In [5], the authors have conducted a comparative energy analysis upon RSA and ECC based public key algorithms for wireless sensor networks. They have used a simplified version of SSL for mutual authentication and key exchange. For their experiments, they have used Berkley/Crossbow motes platform, specifically the MICA2dots [9]. With the outcomes of their experiments, we see that, contrary to the widely held beliefs, authentication and key exchange protocols using optimized software implementations of public key cryptography are very viable on small wireless devices.

In [6], the authors have proposed C4W which is basically an identity based PKC infrastructure. They have shown that their identity based scheme consumes less energy as it is certificateless and thus it is efficient both in terms of computation and communication costs. The TinyPK system demonstrated in [10] shows that, a public-key based protocol is feasible even for an extremely lightweight sensor network. TinyPK is a software-based implementation of public key system tested on UC Berkeley MICA2 motes.

[7] has shown that special purpose ultra-low power hardware implementations of public key algorithms can be used on sensor nodes. The authors have shown that PKC tremendously simplifies the implementation of many typical security services and additionally reduces transmission power due to less protocol overhead. [7] also provides an in-depth comparison of three different PKC implementations (Rabin's scheme, NtruEncrypt and Elliptic curve) particularly targeted at wireless sensor networks.

A more recent work on hardware implementation of PKC is proposed for elliptic curve over binary extension fields in [8]. The authors have proposed a dedicated coprocessor for certain cryptographic operations. They have shown that a reasonable amount of power can be reserved in this case and thus improved performance could be achieved without degrading other performance parameters. Though the actual data path is 8 bits only, this specific purpose coprocessor can handle operands of even 163 bits.

Other than the above mentioned works, in [11], Du et al. have suggested sparing use of PKC due to its high power consuming characteristic and proposed the use of one-way hash function instead of certificate. Construction of Merkle tree forest from sensors' public keys and selection of height of the tree are the basics of their scheme. They have compared their scheme with other popular PK schemes for sensor networks and plotted the results which show significant performances.

A distributed and cooperative public key authentication is proposed in [12]. It is also a hash key based scheme. In this cooperative mechanism, each node stores a limited number of hashed keys for other nodes which help in the authentication procedure during public key operation. According to [12], this scheme is free from any cryptographic operation which is designed to be fit for the constrained resources of the sensors.

[13] has looked at several additive homomorphic public key encryption schemes and their applicability to WSNs when implemented on computationally limited sensor devices. The authors in this work, have provided recommendations for selecting the most suitable public

key schemes based on the topologies and the scenarios of wireless sensor networks.

Reviewing all these works, we are motivated to propose a public/asymmetric key based scheme for secure wireless sensor networks. We have considered the constrained resources of the sensors and have proposed an approach that shows considerable performance with the modern-era sensor node platform. The following section describes our proposed S-PkSec security scheme in detail.

3. Preliminaries

3.1. Network Model

We assume that, the base station (BS) has enough processing power and energy to do the calculations for the sensors in the network. The base station is a trusted entity and cannot be compromised in any way. The BS also has sufficient storage capacity to support the network. The sensors deployed in the network have the computational, memory, communication and energy resources like the current generation of sensor nodes (e.g., MICA2 motes [9]). Once the sensors are deployed over the target area, they remain relatively static in their respective positions.

3.2. Pseudoinverse matrix

The pseudoinverse matrix or generalised inverse matrix [14], [15] has a very nice property that could be used for cryptographic operations. It is well known that, a nonsingular matrix over any field has a unique inverse. For a general matrix of dimension $k \times n$, there might exist more than one generalized inverse. This is denoted by, $M(k,n) =$ {*A*: *A* is a $k \times n$ matrix}. Let, $A \in M(k,n)$. If there exists a matrix $B \in M(n,k)$ such that,

$$ABA = A \text{ and } BAB = B$$

then each of *A* and *B* is called a generalized inverse matrix (or pseudoinverse matrix) of the other. In this paper, we use the notation A_g to denote the generalized inverse matrix of *A*. We use pseudoinverse matrix for the key handshaking process in our security architecture.

It should be noted that, $(A_g)_g = A$ is not always true. The set of all possible pseudoinverse matrices of A is denoted by $\{A_g\}$, and $|\{A_g\}|$ is the cardinality of $\{A_g\}$. Then we have:

Lemma 1: Let A_g be a pseudoinverse matrix of A. Then,

$$rank(A_g) = rank(A)$$

Lemma 2: Let $A \in M(k,n)$ with $rank(A) = k$. If A can be written as $A = [A_1;0]$, where A_1 is a $k \times k$ nonsingular matrix then,

$$\{A_g\} = \{\begin{bmatrix} A_1^{-1} \\ Z \end{bmatrix} : Z \in M(n-k,k) \text{ is an arbitrary matrix}\}$$

Proof: Let $B = \begin{bmatrix} X \\ Z \end{bmatrix} \in M(n,k)$. It is then easy to verify that both $ABA = A$ and $BAB = B$ hold if and only if $X = A_1^{-1}$.

4. Our Proposed Scheme (S-PkSec)

In this section, we propose our scheme with two phases; key handshaking phase between a sensor node and the base station, and the encryption/decryption phase.

4.1. Key Handshaking Phase

Let n_i be a node in the network and S be the base station or sink. To derive a shared secret key between the node n_i and the base station, the following operations are performed:

1. Node n_i randomly generates a matrix X with dimension $m \times n$ and its psedoinverse matrix, X_g. These matrices are kept secret in the node.
2. n_i calculates $X_g X$ and sends it to the base station S.
3. In turn, S randomly generates another matrix Y with dimension $n \times k$, and finds out its pseudoinverse matrix Y_g. These matrices are also kept secret in the base station.
4. S calculates $X_g XY$ and $X_g XYY_g$. Then it sends the resultant matrices to n_i.
5. Upon receiving the products of matrices from S, n_i calculates, $XX_g XYY_g = XYY_g$ and sends it back to the base station.
6. Now, both the node n_i and the base station S can compute the common secret key. n_i gets it by calculating $X(X_g XY) = XY$ and the base station gets it by calculating $(XYY_g)Y = XY$. Both of these outcomes (XY) are the same matrix with dimension $m \times k$.

Basically, the key XY is locally computed by the node and the base station. Our mechanism ensures that, the individually calculated keys are same and this common key is used for encrypting the messages in the network. Thus, key handshaking process ensures a secure and efficient way of deriving distinct secret key (shared with

the base station) for each node taking part in the wireless sensor network. The derived common key could be used for node to base station or base station to node secure communications.

4.2. Encryption/Decryption Phase

The main module in secure node to node communications is a central key generator (CKG) which is located at the base station. The CKG helps any node in the network to decrypt the received encrypted messages from other nodes. If a node n_i wants to send message securely to another node n_j, it uses the key that it has derived using the key handshaking process. Say for example, the encrypted message sent from n_i to n_j is $E_{XY}(M)$. Here, *M* is the message sent from the sender to the receiver. E_{XY} means the message is encrypted with the key XY which is actually the shared secret key between the base station and the sender n_i. Upon receiving the encrypted message, n_j places its own identity and the identity of the sender to the CKG. In turn, CKG generates a decryption key and transmits it to n_j encrypting it with the secret shared key that it has with n_j. As the CKG in the base station has prior knowledge about the shared secret key between n_i and itself, it uses that knowledge to generate the decryption key. Now, n_j first decrypts the encrypted message (i.e., containing the corresponding decryption key) with its shared key, finds out the decryption key and uses that key to decrypt the message sent from node n_i.

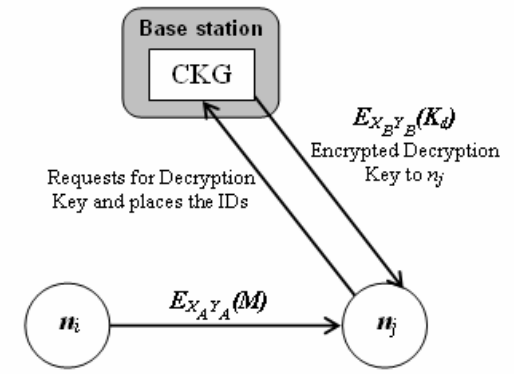

Figure 1. Encryption and decryption of message by two communicating nodes in the network in S-PkSec.

Figure 1 shows the secure communication method between two nodes in the network. In the figure, X_AY_A is the shared secret key between n_i and base station, X_BY_B is the shared secret key between n_j and the base station, and K_d is the decryption key provided by the base station.

5. Performance Evaluation

5.1 Initial Analysis

In our scheme, any node s_i sends the *SBS* an $n \times n$ matrix which is of n^2 bits. In turn, the *SBS* sends an $n \times k$ matrix and an $n \times n$ matrix. For this the total number of bits passed for the matrices is, $n^2 + nk = n(n+k)$ bits. Again, the node s_i sends the *SBS* an $m \times n$ bits. So, total number of bits for the matrices transmitted for deriving the shared key in the whole key handshaking process is,

$$\begin{aligned} & n^2 + n(n+k) + mn \\ &= n(n+n+k+m) \\ &= n(2n+k+m) \end{aligned}$$

All the calculations here are linear and can be performed very easily. Moreover, our scheme is adequately secure as capturing the messages like X_gX, X_gXY, X_gXYY_g, and XYY_g could not be in any way helpful to construct the locally computed secret shared key XY.

5.2 Simulation

We have analyzed our asymmetric key-based scheme S-PkSec in terms of energy cost, memory cost, security and scalability. In our simulation, we considered the specifications of Berkeley/Crossbow MICA2dot [9] motes. These motes are equipped with 8-bit ATmega128L microcontrollers with 4 MHz clock speed, 128 KB program memory and Chipcon CC1000 low-power wireless transceiver with 433-916 MHz frequency band. The major power consumers in this mote are the processor and the wireless transceiver. During the transmission and reception operations, the microcontroller is turned on alongside the wireless transceiver.

According to our calculations, the cost of transmission of one byte is 59.2 μJ while the reception operation takes about half of the transmission cost (28.6 μJ). The power to transmit 1 bit is equivalent to roughly 2090 clock cycles of execution of the microcontroller. In our case, we considered a packet size of 41 bytes (payload of 32 bytes, header 9 bytes). With an 8 byte preamble (source and destination address, packet length, packet ID, CRC and a control byte) for each packet we found that, to transmit one packet 49 × 59.2 = 2.9008 mJ ≈ 2.9 mJ energy is required. Accordingly, the energy cost for receiving the same packet is 49 × 28.6 = 1.4014 mJ ≈ 1.4 mJ. Considering the same packet size for all the network operations, to set up a shared secret key with the base station each node needs (two transmissions and one reception) ((2 × 2.9) +1.4) = 7.2 mJ of energy. This cost is one time cost as once the shared secret key is derived, it could be used for the entire lifetime of the network unless the key is exposed or the node quits the network.

For node to node communication, the sender needs one transmission (2.9 mJ) and the receiver needs two

receptions and one transmission (((2×1.4) +2.9) = 5.7 mJ). As a whole, the entire scheme could be well-afforded by the energy resources of the current generation sensor nodes.

5.3 Comparison

We compare our S-PkSec with C4W [6] and the one proposed in [5], which use simplified version of SSL handshake. Considering the energy consumption for communications, our scheme stays in the middle of other two schemes (shown in Table 1).

Table 1. Communication cost for our scheme and other PKC-based schemes

PKC-based Schemes	Communication Cost	
	Sender (n_i)	*Receiver (n_i)*
C4W	6.3 mJ	4.8 mJ
Our Scheme	10.1 mJ	5.7 mJ
SSSL	19.4 mJ	19.6 mJ

For supporting our scheme, for sender node (n_i), the numbers of transmissions and receptions are 3 and 1 respectively which take ((3×2.9) +1.4) = 10.1 mJ) of energy in total. In case of receiver node (n_B), it is 5.7 mJ for 1 transmission and 2 receptions. Figure 3 shows a comparative graph in terms of communication cost among these schemes.

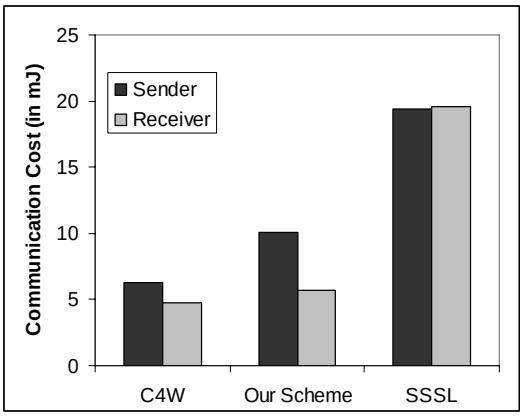

Figure 2. Communication cost for different PKC-based schemes

Though considering the communication cost, C4W exhibits better cost effectiveness than our scheme and SSSL, it requires pre-storing of all parameters before deployment. This in fact causes memory exploitation which is not present in our scheme. So, in terms of memory usage our asymmetric key-based scheme is better and its communication cost is satisfactory.

5.4 Further Comments

In encryption decryption phase two messages are transmitted over public channel. When the receiver node needs the decryption key to decrypt a message from a particular sender node, it requests the SBS for the corresponding decryption key. In return, CKG encrypts the decryption key with the shared secret key of the receiver node. As the shared secret key is not known to any other node in the network, the decryption key for that particular sender-receiver pair could not be exposed. Now, the problem arises if the shared secret key of a node in the network is somehow compromised. In such a case, the base station revokes the shared key and the key handshaking process is re-initiated for that particular node. If such a compromise happens, even in that case, only one node is affected in the network while all other nodes could properly operate with confidential message transmission.

As any node can get a corresponding decryption key from the CKG (base station) for any sender node in the network, any pair of nodes in the network could communicate between themselves maintaining the high level of security. As mentioned earlier, for base station to node communications or node to base station communications, the shared secret key derived from the handshaking process is used which takes very little computation and message transmission.

6. Conclusion and Future Works

On constructing complete security architecture for wireless sensor networks, in this paper we have presented an efficient approach which uses the asymmetry of public key cryptosystem for secure communications in the network. Though we think, the use of public key embedded sensor network is still far from reality, some good applications related to public key based scheme [16] make us positively convinced. So, we became enthusiastic to propose some mechanisms and protocols by means of asymmetric cryptography.

While comparing with other asymmetric key based scheme we only considered transmission cost. In future, we should consider the computation cost as well, because this is not trivial for regular and small size networks. Moreover, in order to obtain complete and accurate results we have to find a common unit of measurement among all those schemes. This will also allow us to set the optimum for fine comparison among them. In addition to this developing a complete mathematical model for our scheme is one of our future objectives. We should also consider the key management and key revocation technique of our scheme more details in the future.

In our scheme, we have used different keys for encryption and decryption of the messages for node-to-node communications. Our simulation results and analysis have shown a considerable level of security which is viable with the current generation sensor node platforms. Our PKC-based architecture does not require any central certificate authority and thus it is free from managing and verifying huge computations associated with certificates. In future, we will combine our work with other security mechanisms to construct large-scale security architecture for WSN.

7. References

[1] M. M. Haque, A.-S. K. Pathan, B. G. Choi, and C. S. Hong, "An Efficient PKC-Based Security Architecture for Wireless Sensor Networks", *Proceedings of the IEEE MILCOM'2007*, October 29-31, p.72 (CD), Orlando, Florida, USA.

[2] A. D. Wood, J. A. Stankovic, and S. H. Son, "JAM: A Jammed-Area Mapping Service for Sensor Networks," *Proceedings of the 24th IEEE International Real-Time Systems Symposium (RTSS'03)*, 2003, pp. 286-297.

[3] A.-S. K. Pathan, H.-W. Lee, and C. S. Hong, "Security in Wireless Sensor Networks: Issues and Challenges," *Proceedings of 8th IEEE ICACT 2006*, Volume II, 20-22 February, Phoenix Park, Korea, 2006, pp. 1043-1048.

[4] D.J. Malan, M. Welsh, and M.D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," *Proceedings of IEEE SECON 2004*, 4-7 October, 2004, pp. 71 – 80.

[5] A.S. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz, "Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks", *Proceedings of PerCom 2005*, pp. 324-328.

[6] Q. Jing, J. Hu, and Z. Chen, "C4W: An Energy Efficient Public Key Cryptosystem for Large-Scale Wireless Sensor Networks", *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Oct. 2006, pp. 827-832.

[7] G. Gaubatz, J.-P. Kaps, E. Ozturk, and B. Sunar, "State of the Art in Ultra-Low Power Public Key Crytography for Wireless Sensor Networks," *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, 2005*, pp. 146-150.

[8] G. Bertoni, L. Breveglieri, and M. Venturi, "Power Aware Design of an Elliptic Curve Coprocessor for 8 bit Platforms," *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, p. 337.

[9] Xbow Sensor Networks, Available at: http://www.xbow.com/

[10] R. Watro, D. Kong, S.-f. Cuti, C. Gardiner, C. Lynn and P. Kruus, "TinyPK: Securing Sensor Networks with Public Key Technology," *ACM SASN'04*, Washington, DC, USA, 2004, pp. 59-64.

[11] W. Du, R. Wang, and P. Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks," *Proceedings of ACM MobiHoc'05*, Illinois, USA, 2005, pp. 58-67.

[12] D. Nyang and A. Mohaisen, "Cooperative Public Key Authentication Protocol in Wireless Sensor Network," *UIC 2006, LNCS 4159*, Springer-Verlag, pp. 864-873, 2006.

[13] E. Mykletun, J. Girao, and D. Westhoff, "Public Key Based Cryptoschemes for Data Concealment in Wireless Sensor Networks," *Proceedings of IEEE International Conference on Communications*, 2006, pp. 2288-2295.

[14] A.B. Israel, and T.N.E. Greville, *Generalized inverses: theory and applications*, John Wiley & Sons, New York, 1974.

[15] T.L. Boullion, and P.L. Odell, *Generalized inverse matrices*, Wiley-Interscience, New York, 1971.

[16] V. Gupta, M. Wurm, Y. Zhu, M. Millard, S. Fung, N. Gura, H. Eberle and S. C. Shantz, *Sizzle: A Standards-based End-to-End Security Architecture for the Embedded Internet*, Technical Report, Sun Microsystems, June 2005.