Copyright ⓒ 2008 KSII

An Asymmetric Key-Based Security Architecture for Wireless Sensor Networks

Md. Mokammel Haque, Al-Sakib Khan Pathan, Choong Seon Hong, *member, KSII*, and Eui-Nam Huh, *member, KSII*
Department of Computer Engineering, Kyung Hee University
1 Seocheon, Giheung, Yongin, 446-701 Gyeonggi, South Korea
[email: malinhaque, spathan@networking.khu.ac.kr, cshong, johnhuh@khu.ac.kr]
*Corresponding author: Choong Seon Hong

Received September 3, 2008; accepted September 20, 2008; published October 25, 2008

Abstract

In spite of previous common assumptions about the incompatibility of public key cryptography (PKC) schemes with wireless sensor networks (WSNs), recent works have shown that they can be utilized for such networks in some manner. The major challenge of employing a PKC-based scheme in a wireless sensor network is posed by the resource limitations of the tiny sensors. Considering this sensor feature, in this paper we propose an efficient PKC-based security architecture with relatively lower resource requirements than those of previously proposed PKC schemes for WSN. In addition, our scheme aims to provide robust security in the network. Our security architecture comprises two basic components; a key handshaking scheme based on simple, linear operations and the derivation of a decryption key by a receiver node. Our architecture enables node-to-base-station and node-to-node secure communications. Analysis and simulation results show that our proposed architecture ensures a good level of security for network communications, and can be effectively implemented with the limited computational, memory, and energy budgets of current-generation sensor nodes.

Keywords: Architecture, pseudoinverse, public key, sensor

A preliminary version of this paper appeared in the Proceedings of IEEE MILCOM 2007, October 29-31, Orlando, Florida, USA. This is the extended and revised version. This research was supported by the MKE, the Korean government, under the ITRC support program supervised by the IITA (IITA -2008 – (C1090 – 0801 -0002)).

DOI: 10.3837/tiis.2008.05.004

1. Introduction

Security is an indispensable requirement for most sensor network applications. Though security is regarded as a standalone component of the architectures of many systems, in case of wireless sensor networks (WSNs), it must get adequate attention. The types of services expected in wireless sensor networks often require the network security architecture. In most application domains, the sensors are used to collect a specific type of data from particular target areas, and the collected data are often considered secret and are not intended for public disclosure. Hence, efficient and secure mechanisms are needed to transmit acquired data securely to the appropriate recipients.

Wireless sensor networks occasionally carry confidential information, which, if exposed to enemy parties, might cause a debacle for friendly parties. Especially in military, medical, and emergency applications of WSN, employing apropos security mechanisms for data transmissions is crucial, as this data can be used to make tactical decisions. If an adversary thwarts the operation of the network by modifying the transmitted information, stopping transmission, or by eavesdropping on information, then the usefulness of sensor networks is drastically curtailed. Likewise, for example in a disaster management related application, accurate and unmodified data is needed to predict upcoming disaster(s) and provide advanced warning to concerned parties about event(s).

However, ensuring a good level of security for such types of networks is not a trivial task. As WSNs use wireless communications, the threats and attacks against them are more diverse and often large-scale. It is not possible to deal with all types 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 such as jamming [1] cannot be handled by any key management scheme. Hence, several mechanisms at different layers can be employed simultaneously, to provide holistic security [2] for wireless sensor networks and in addition the level of security in the data transmission and communication phase can be increased via efficient key management schemes. Public key cryptography (PKC) can be the best choice for ensuring a satisfactory level of security for data transmissions in the network. However, the major challenge of employing any public key-based security scheme in WSN is the constrained energy, computational, and memory budgets of sensors participating in the network. Because of this fact, PKC-based schemes were not utilized in these networks for quite some time.

To counter the bias against PKC for WSNs, several public key schemes have recently demonstrated an acceptable performance for low-power sensor nodes [3] [4] [5]. Considering both the software and hardware configurations, elliptical curve-based public key cryptography (ECPKC) has shown relatively better results for eight-bit mote platforms. However, the use of certificates in such a scheme consumes substantial bandwidth and power.

Considering the special characteristics of wireless sensor networks, in this paper we propose an efficient public key based security architecture for WSNs. Before presenting our proposal, we analyze related studies and their benefits and drawbacks, to provide readers with a clear picture of the current status of PKC-related research works in wireless sensor networks. Many recent works have suggested the feasibility of PKC in WSNs, with little energy cost. This could be especially helpful in applications that require a high level of security. Inspired by the encouraging results of practical implementations, we propose a security architecture that can exploit the benefits of PKC in WSNs. In our scheme, we use a pseudo-inverse matrix for the first component, while the second component is a simple method for passing the decryption key to the receiver node [6]. Our analysis and simulation results show that our scheme demonstrates a considerable gain in the level of security, and is suitable for current generation sensor nodes.

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

2. Literature Review and Motivation

Most works about public key cryptography in wireless sensor networks are based on the low-power characteristics of sensor nodes. Several recent works have successfully implemented public key schemes with current-generation sensors. In terms of both software and hardware perspectives, PKC schemes have shown reasonable performance. In this section, we discuss some of these exclusive research works and their comparative features.

[3] presented the first known implementation of elliptic curve cryptography over F2p for sensor networks based on eight-bit, 7.3828 MHz MICA2 motes [7]. The results show that a public key based scheme is feasible for current-generation sensors. The TinyPK system demonstrated in [8] shows that a public-key based protocol is feasible, even for an extremely lightweight sensor network. TinyPK is a software-based implementation of a public key system tested on UC Berkeley MICA2 motes.

Gaubatz et al. [9] proposed a custom hardware-assisted approach which makes public key cryptography feasible in WSN environments, provided that suitable algorithms and associated parameters, careful optimization, and low-power design techniques are selected. In order to validate their claim, they present proof of concept implementations of two different algorithms; Rabin's Scheme and NtruEncrypt. Their work demonstrates that in spite of common assumptions about public key cryptography, it is possible to achieve a level of power consumption that is sufficiently low to enable its use even for battery-attached sensor nodes.

In [10] it was shown that special purpose, ultra-low power hardware implementations of public key algorithms can be used for sensor nodes. The authors show that PKC tremendously simplifies the implementation of many typical security services and reduces transmission power, because of a lower protocol overhead. Also, [10] provides an in-depth comparison of three different PKC implementations (Rabin's scheme, NtruEncrypt, and Elliptic Curve) particularly aimed at wireless sensor networks.

Blaß and Zitterbart [11] presented an efficient and lightweight implementation of public-key cryptography algorithms relying on elliptic curves. They checked their codes by implementing them on popular eight-bit ATMEGA128 microcontroller, the heart of the MICA2 platform. Their work concluded that public-key cryptography is feasible for sensor networks.

Gupta et al. [12] showed that elliptic curve cryptography not only makes public-key cryptography feasible for these devices, but also enables the creation of a complete, secure web server stack that runs efficiently with very stringent resource constraints. Their work, viz., Sizzle, is the first time the Internet's dominant security protocol (SSL) has been applied to devices with significant computational, memory, and energy constraints. It uses highly optimized implementations of public-key cryptography to support scalable key management and end-to-end security, without sacrificing efficiency.

In [4], the authors conducted a comparative energy analysis of RSA and ECC based public key algorithms for wireless sensor networks. They use a simplified version of SSL for mutual authentication and key exchange. For their experiments, they use the Berkley/Crossbow motes platform, specifically MICA2DOT. Based on the results of their experiments, it is clear that contrary to common assumptions, authentication and key exchange protocols using optimized software implementations of public key cryptography are feasible for small, wireless devices.

Murphy et al. [13] showed that it is possible to implement public key algorithms for resource constrained sensor node platforms. Via a hardware/software co-design approach, they successfully map a public key cryptosystem based on Rabin's scheme to the motes developed by Tyndall National Institute. Their implementation focuses on efficient architectures that execute public key algorithms using minimal resources. Their finding is that the hardware implementation of the encryption algorithm is much faster than the software implementation. Software implementations of the algorithm are also realizable, and provide the benefits of low cost and high flexibility. However the time necessary to perform encryption/decryption is significantly increased if a software-only approach is used.

Piotrowski et al. [14] investigated four types of nodes; MICA2DOT, MICA2, MICAz, and TelosB, and estimated the power consumption for most common RSA and ECC operations. Their work gives an indication of how public key cryptography influences a wireless node’s lifetime. In [5], the authors propose C4W, which is basically an identity-based PKC infrastructure. They show that their identity-based scheme consumes less energy, as it is certificate-less, and thus it is efficient, both in terms of computational and communication costs.

A hardware implementation of PKC for elliptic curves over binary extension fields was proposed in [15]. In this paper, the authors proposed a dedicated coprocessor for certain cryptographic operations. They showed that a reasonable amount of power savings can be achieved in this case, and thus improved performance can be achieved, without degrading other performance parameters. Though the actual data path is only eight bits, this special purpose coprocessor can handle operands of even 163 bits.

A distributed, cooperative public key authentication scheme is proposed in [16]. 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 authentication during public key operation. According to [16], this scheme avoids cryptographic operations, and is designed for the constrained resources of the sensors. However the major drawback of the proposed method is that it is only designed for one-hop authentication, which makes it impractical and inefficient for conventional multi-hop WSNs.

In [17] there was an examination of several additive homomorphic public key encryption schemes and their applicability to WSNs, when implemented on computationally limited sensor devices. The authors in this work provide recommendations for selecting the most suitable public key schemes based on topologies and scenarios for wireless sensor networks.

In their works, Roman and Alcaraz [18] discussed the applicability of public key infrastructures to wireless sensor networks and Ugus et al. [19] implement elliptic curve and finite field arithmetic operations on a MICAz mote, which is a typical device employed in wireless sensor networks.

Other than the aforementioned works, in [20], Du et al. suggest limited use of PKC, due to its high power consumption characteristics, and propose the use of a one-way hash function instead of a certificate. Construction of a Merkle tree forest from sensors’ public keys and selection of the height of the tree are the basics of their scheme. They compare their scheme with other popular PK (public key) schemes for sensor networks, and plot the results, which show significant gains. However, the drawback of this scheme is that it requires storing some information in the sensors' memory, prior to their deployment.

After reviewing these works, we were motivated to propose a public key based architecture for secure wireless sensor networks. We consider the resource limitations of the sensors and propose an approach that demonstrates good performance with current generation sensor node platforms. We first derive the shared secret key for node-to-base station communications and then exploit the asymmetric nature of PK schemes for node-to-node secure communications. The following section describes our security architecture in detail.

3. Network Assumptions and Preliminaries

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

3.1 Pseudo-inverse Matrix

The pseudoinverse matrix or generalised inverse matrix [21] [22] has an elegant property that can be exploited 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;

$ABA = A$ and $BAB = B$

then A and B are both called a generalized inverse matrix (or pseudo-inverse matrix) of the other. In this paper, we use the notation A_g to denote the generalized inverse matrix of A. We use pseudo-inverse matrix for the key handshaking in our security architecture.

It should be noted that, $(A_g)_g = A$ is not always true. The set of all possible pseudo-inverse 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 pseudo-inverse 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, $\{A_g\} = \{\begin{bmatrix} A_1^{-1} \\ Z \end{bmatrix} : Z \in M(n-k,k)$ is an arbitrary matrix$\}$

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

3.2 Our PKC-Based Security Architecture

In this section, we present our public key based security architecture with two separate but interrelated components. The first component is key handshaking, in which each network node can derive a shared secret key with the base station, and the second component is used for confidential and authenticated data transmissions between two participating network nodes.

3.3 Key Handshaking between a Node and Base Station

Let n_i be a network node 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 psedo-inverse matrix, X_g. These matrices are kept secret in the node.
2. 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 determines its pseudo-inverse matrix Y_g. These matrices are also kept secret in the base station.
4. 4. S calculates $X_g XY$ and $X_g XYY_g$. Then, it sends the resultant matrices to n_i.
5. After 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. Here, both the node n_i and base station S can compute the secret shared key; n_i calculates $X(X_g XY) = XY$ and the base station calculates $(XYY_g)Y = XY$. Both these outcomes (XY) are the same matrix with dimension $m \times k$.

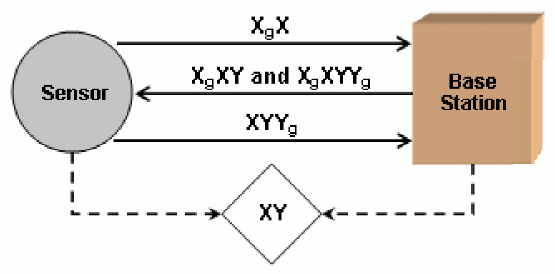

Fig. 1. Key handshaking between a sensor and base station, to derive a shared secret key

Fig. 1 shows the communications between a sensor and the base station in key handshaking. Basically, the key is locally computed by the node and the base station. Our mechanism ensures that the individually calculated keys are the same, and this common key is used for encrypting the network messages. Thus, key handshaking ensures a secure and efficient means of deriving unique secret keys (shared with the base station) for each node participating in the wireless sensor network. The derived shared key can be used for node-to-base station or base station-to-node secure communications.

3.4 Encryption/Decryption of Data for Node-to-Node Communications

The main module in secure node-to-node communications is the central key generator (CKG), which is located at the base station. CKG helps any network node to decrypt the received encrypted messages from other nodes. If a node n_i wants to send a secure message to another node n_j, it uses the key derived via key handshaking. For example, assume that the encrypted message sent from n_i to n_j is $E_{XY}(M)$. Here, M is the message sent from the sender to receiver. E_{XY} means the message is encrypted with the key XY, which is actually the shared secret key between the base station and sender n_i. After receiving the encrypted message, n_j sends 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 keys of both nodes, it uses that knowledge to generate the decryption key. Here, n_j first decrypts the encrypted message (i.e., containing the corresponding decryption key) with its shared key, determines the decryption key and uses that key to decrypt the message sent from node n_i.

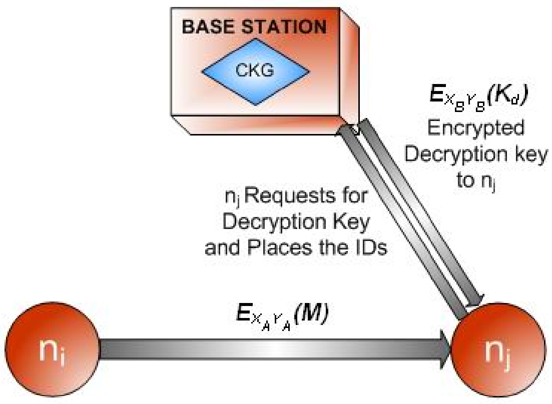

Fig. 2. Encryption/decryption of message by two communicating network nodes

Fig. 2 shows the secure communication method between two network nodes. In the figure, X_AY_A is the shared secret key between n_i and the base station, X_BY_B is the shared secret key between n_j and the base station, M is the message that is encrypted by n_i, and K_d is the decryption key provided by the base station.

4. Performance Evaluation and Comparison

To analyze the performance of our security scheme, we considered the specifications of the MICA2DOT [7] mote platform. We analyzed the PKC-based architecture in terms of the energy cost, memory cost, computational cost, security, and scalability. In this section, we discuss the detailed analysis and simulation results for our approach.

In key handshaking, we used linear matrix operations, matrix multiplication to be precise. The complexity of matrix multiplication is very low; hence it can be performed very quickly. In our shared secret key derivation scheme, node n_i sends the base station an $n \times n$ matrix, which consists of n^2 bits. In turn, the base station sends an $n \times k$ matrix and an $n \times n$ matrix.

For this, the total number of bits used for the matrices is: $n^2 + nk = n(n+k)$. Again, the node n_i sends the base station $m \times n$ bits. So, the total number of bits used for the matrices transmitted, to derive the shared key in overall key handshaking is;

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

All these computations are linear and can be performed very easily.
In the first component, for key handshaking we use the public channel for the message transmission. However, capturing the messages such as $X_g X$, $X_g XY$, $X_g XYY_g$ and XYY_g may not be helpful for constructing the locally computed secret shared key XY. It might seem that a prospective attack can gain some information about matrix Y from knowledge about $X_g X$ and $X_g XY$. Let us consider, $rank(X) = r$. Let us assume that;

$$X_g X = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$$

Here, I_r is an identity matrix of order $r \times r$. Then, only the first r rows of Y can be determined from $X_g XY$. As Y is chosen randomly, there are $2^{(n-r)\cdot k}$ ways to choose the last $(n-r)$ rows of Y. The knowledge about $X_g XYY_g$ might be helpful for determining Y but according to Lemma 2 it is of little use to the adversary. Without knowledge about the last $(n-r)$ rows of Y, even if X is completely known, the probability of determining the correct value of each element of XY would be $\frac{1}{2}$, considering that any row vector of X has a nonzero element in any of the last $(n-r)$ positions, which is likely when $(n-r)$ is very large and X is chosen at random. However, the probability of determining X from knowledge about $X_g XYY_g$ and XYY_g is even smaller where $rank(A_g ABB_g) \leq rank(A_g A)$. So, based on this analysis of the key handshaking phase, it can be assumed that, the probability of successful cracking of this scheme is, $2^{-(n-r)\cdot k}$. So, the security of the handshaking scheme is reasonably high for carefully chosen parameters. To ensure that $2^{(n-r)\cdot k}$ is a large number, n must be considerably larger than r. And this can be guaranteed by ensuring that $m < n$.

There are vulnerabilities in key handshaking between a node and the base station, if there is sort of identification problem with the participating entities during communications. In our scheme however, this threat has been completely eliminated because; (a) The base station is a trusted entity and cannot be compromised by any means and (b) The IDs of the communicating nodes are checked by the base station before further communications. In the second component of node-to-node communication, when the receiver node requests the corresponding decryption key, the key is not sent as a plaintext, rather it is encrypted with the shared secret key of the receiver. So, there is no way any adversary can determine the decryption keys for a particular sender-receiver pair.

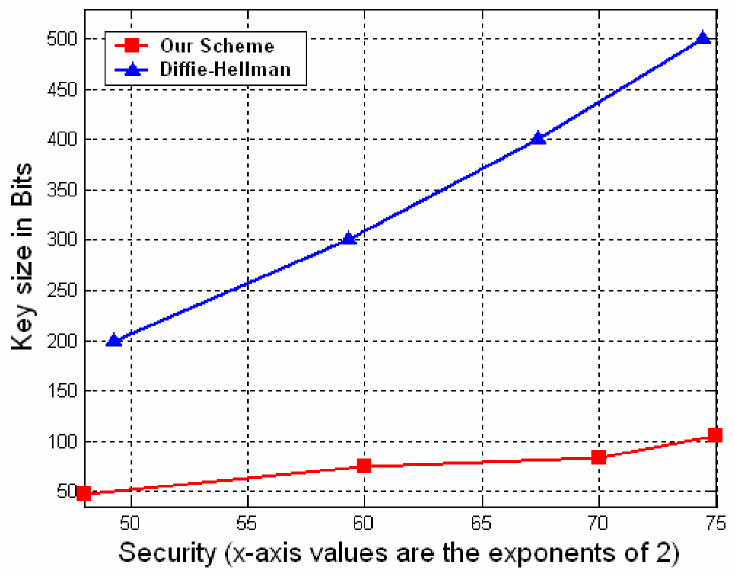

Fig. 3. Key sizes, required to ensure similar levels of security in our key handshaking scheme and the Diffie-Hellman key exchange scheme. It is evident that our handshaking scheme requires a much smaller size key for each of the cases plotted in this figure

Here, let us compare our shared key derivation scheme with Diffie-Hellman's scheme [23]. In the Diffie-Hellman key exchange scheme, a primitive element α over a finite field $GF(p)$ is used. The first user randomly chooses an integer x from $GF(p)$ and sends $\alpha^x \bmod p$ to the second user. In turn, the second user chooses an integer y and sends $\alpha^y \bmod p$ to the first user. Both users can derive a shared secret key $\alpha^{xy} \bmod p$ which is $\log_2 p$ bits in length. And $2\log_2 p$ bits of data are transmitted via the public channel in this key exchange.

The security of the Diffie-Hellman (D-H) scheme is based on the difficulty of a discrete logarithm problem, such that given α^x and α^y over the finite field $GF(p)$, the problem is how to determine α^{xy}. If x can be determined from α^x, which is actually a discrete logarithm problem, then $\alpha^{xy} = (\alpha^y)^x$ can be determined. The average computational complexity of this discrete logarithm problem using the best method known to date [24] is $O(\exp((1.923+o(1))(\log_2 p)^{\frac{1}{3}}(\log_2 \log_2 p)^{\frac{2}{3}}))$ bit operations.

Based on the aforementioned analysis, we found that to achieve a security level (complexity) of $2^{49.3}$ in the D-H scheme, a key size of 200 bits is needed. On the other hand, to achieve a similar level of security (1/probability) of 2^{48}, a 48-bit key is required in our scheme. Likewise, to achieve security of $2^{59.3}$, $2^{67.4}$ and $2^{74.4}$ in the D-H scheme, 300, 400 and 500 bit keys are required, respectively. On the other hand, to achieve security of 2^{60}, 2^{70} and 2^{75} in our approach, 75, 84 and 105 bit keys are required, respectively. In this analysis, the sizes of p in bits for the D-H scheme are 200, 300, 400 and 500, respectively. For our approach, (m,n,k) is (4, 8, 12), (5, 9, 15), (6, 11, 14) and (7, 12, 15) respectively. **Fig. 3** shows the level of security achieved with the key sizes in our approach and the D-H scheme. It is clear from the figure that in all cases, our approach needs keys with a much smaller size than that of the D-H scheme. In the figure, the x-axis shows the values of the exponent of 2 to plot the security level for various key sizes. For example, a value of 48 along the x-axis corresponds to 2^{48}. This is in order to fit the values in the graph for visual representation.

The volume of traffic carrying the key-related information is also dependent on the size of the key. **Fig. 4** plots the key sizes in bits versus the volume of traffic (considering only the key-related information) that pass through the open public channel in our overall security architecture and the Diffie-Hellman scheme. The data in this figure is based on the fact that the same level (or a similar level) of security is required for both schemes. From the figure, note that in our approach the key sizes and volume of key-related traffic are relatively smaller than those of the Diffie-Hellman scheme. For example, whereas the size of the key in bits is only 48 in our scheme, the Diffie-Hellman scheme requires a 200-bit key to provide the same level of security. So, the volume of key-related traffic is much higher in their scheme than that of our scheme. Considering this case, the volume of traffic in our scheme is 304 bits, while in the case of the Diffie-Hellman scheme, it is 400 bits. As the key size increases, the difference in the volume of key-related traffic between these schemes also increases.

Here, as Diffie-Hellman scheme is based on discrete logarithm problem, the eavesdropper's computational capability is considered when the security is analyzed. However, our first component is not a discrete logarithm problem rather it is a probabilistic problem, to crack the scheme. Hence, the increase of the computational capability of the eavesdropper or the man-in-the-middle adversary does not affect the probability of cracking the scheme.

In the second phase of our architecture, only two messages are transmitted over the public channel. When the receiver node needs the decryption key to decrypt a message from a particular sender node, it sends a request to the base station for the corresponding decryption key. In return, the base station encrypts the decryption key with the shared secret key of the receiver node. As the shared secret key is not known to any other network node, the decryption key for that particular sender-receiver pair cannot be exposed. Here, there is a problem if the shared secret key of a network node is somehow compromised. In such a case, the base station revokes the shared key and key handshaking is re-initiated for that particular node. If such a compromise occurs, even in that case, only one network node is affected, while all other nodes can continue to operate and transmit their messages.

As any node can obtain a corresponding decryption key from the base station for any sender network node, any pair of network nodes can 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 handshaking is used, which requires very little computation and message transmission.

In our simulation, we considered the specifications of Berkeley/Crossbow MICA2DOT motes [7]. These motes are equipped with eight-bit ATmega128L microcontrollers with a four MHz clock speed, 128 kB program memory and Chipcon CC1000 low-power wireless transceiver with a 433-916 MHz frequency band. The major power consumers in this mote are the processor and wireless transceiver. During the transmission and reception operations, the microcontroller is turned on along with the wireless transceiver. According to our calculations, the cost of transmission of one byte is 59.2 μJ, while the reception operation has about half the transmission cost (28.6 μJ). The power to transmit one bit is equivalent to roughly 2,090 clock cycles of execution of the microcontroller. In our simulation, we considered a packet size of 41 bytes (payload of 32 bytes, header 9 bytes). With an eight 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 $49 \times 59.2 = 2.9008$ mJ ≈ 2.9 mJ energy is required.

Accordingly, the energy cost for receiving the same packet is $49 \times 28.6 = 1.4014$ mJ ≈ 1.4 mJ. Considering the same packet size for all network operations, to set up a shared secret key with the base station, each node needs (two transmissions and one reception) $((2 \times 2.9)+1.4) = 7.2$ mJ of energy. This cost is a one-time cost, as once the shared secret key is derived it can be used for the entire lifetime of the network, unless the key is exposed. For node-to-node communication, the sender needs one transmission (2.9 mJ), and the receiver needs two receptions and one transmission $(((2 \times 1.4)+2.9) = 5.7$ mJ). Overall, the entire scheme is within the energy resource budget of current-generation sensor nodes.

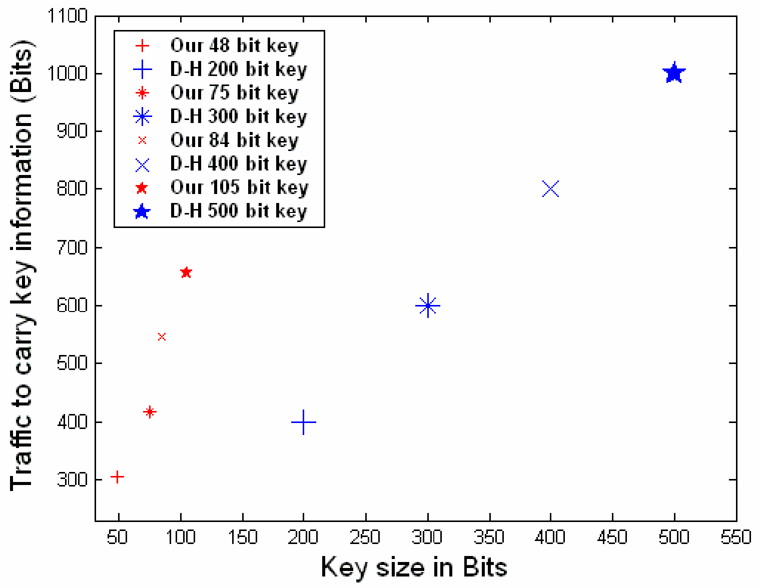

Fig. 4. Key size versus volume of traffic to carry the key-related information through the public channel in our approach (overall) and in the Diffie-Hellman (D-H) scheme, to ensure similar levels of security

Table 1. Communication Costs for Our Scheme and Other PKC-Based Schemes

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

We compare our scheme with C4W [5] and the one proposed in [4], which uses a simplified version of the SSL handshake. Considering the energy consumption for communications, our scheme lies mid-way between the other two schemes (shown in **Table 1**). To support our security architecture, for sender node (n_i), the number of transmissions and receptions is three and one, respectively, which consumes $((3 \times 2.9) + 1.4) = 10.1$ mJ) of energy in total. In the case of the receiver node (n_B), it is 5.7 mJ for one transmission and two receptions. **Fig. 5** shows a comparative graph of the communication costs for these schemes.

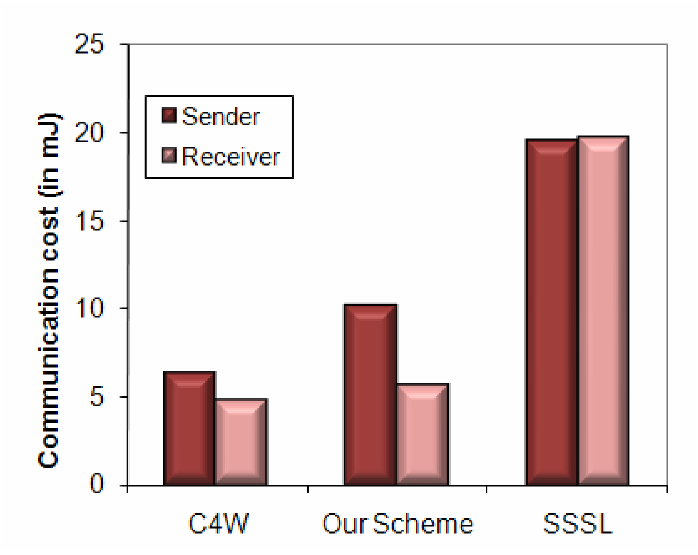

Fig. 5. Communication costs comparing PKC-based schemes

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

Finally, it should be mentioned that our approach is highly scalable, as any number of new sensors can be added to an existing wireless sensor network, based on the requirements. Any newly added node can exchange a shared secret key with the base station via key handshaking, and then it can be used for node-to-node secure communications.

6. Conclusions and Future Works

Use of PKC or PKC-based schemes in WSNs is no longer a fiction. We have shown that many researchers are aiming to find the means to optimize PKC-based schemes. To facilitate the development of holistic security architecture for wireless sensor networks, in this paper we have presented an efficient approach, which exploits the asymmetry of a public key cryptosystem for secure network communications. We have used different keys for encryption/decryption of messages for node-to-node communications. Our simulation results and analysis have shown a considerable level of security that is feasible with current-generation sensor node platforms. As the keys are generated after deployment of the sensors, no pre-assignment of keys is required, and in our approach there is no need to store a look-up key table. Our PKC-based architecture does not require any centralized certificate authority and thus it avoids the need of managing and verifying huge number of communications associated with certificates. In the future, we will combine our work with other security mechanisms associated with different operation layers, to construct a holistic security architecture for wireless sensor networks.

References

[1] 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*), pp. 286-297, 2003.

[2] 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, pp. 1043-1048, 2006.

[3] 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*, pp. 71-80, Oct. 2004.

[4] 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'05*, pp. 324-328, 2005.

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

[6] 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 Military Communications Conference* (*IEEE MILCOM 2007*), Orlando, Florida, USA, Oct. 2007.

[7] Xbow Sensor Networks, available at: http://www.xbow.com/

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

[9] G. Gaubatz, J.-P. Kaps, and B. Sunar, "Public Key Cryptography in Sensor Networks-Revisited," *ESAS 2004*, *LNCS 3313*, Springer-Verlag, pp. 2-18, 2005.

[10] 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*, pp. 146-150, 2005.

[11] E.O. Blaß, and M. Zitterbart, "Towards Acceptable Public-Key Encryption in Sensor Networks," *Proceedings of ACM 2nd International Workshop on Ubiquitous Computing*, Miami, USA, pp. 88-93, May 2005.

[12] 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," *SMLI TR-2005-145*, June 2005.

[13] G. Murphy, A. Keeshan, R. Agarwal, and E. Popovici, "Hardware - Software Implementation of Public-Key Cryptography for Wireless Sensor Networks," *Proceedings of Irish Signals and Systems Conference*, pp. 463-468, 28-30 June 2006.

[14] K. Piotrowski, P. Langendoerfer, and S. Peter, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime," *Proceedings of ACM SASN 2006*, Virginia, USA, pp. 169-176, 2006.

[15] 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*), pp. 337, 2006.

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

[17] 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*, pp. 2288-2295, 2006.

[18] R. Roman and C. Alcaraz, "Applicability of Public Key Infrastructures in Wireless Sensor Networks," *EuroPKI 2007*, *LNCS 4582*, Springer-Verlag, pp. 313-320, 2007.

[19] O. Ugus, A. Hessler, and D. Westhoff, "Performance of Additive Homomorphic EC-ElGamal Encryption for TinyPEDS," Technical Report 6, July 2007, available at: http://www.ist-ubisecsens.org/publications/EcElgamal-UgHesWest.pdf
[20] W. Du, R. Wang, and P. Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks," *Proceedings of ACM MobiHoc'05*, Illinois, USA, pp. 58-67, 2005.
[21] A.B. Israel and T.N.E. Greville, Generalized inverses: theory and applications, John Wiley & Sons, New York, 1974.
[22] T.L. Boullion and P.L. Odell, Generalized inverse matrices, Wiley-Interscience, New York, 1971.
[23] M.Y. Rhee, Internet Security: Cryptographic Principles, Algorithms and Protocols, WILEY, 2003.
[24] A.J. Menezes, P.C.V. Oorschot, and S.A. Vanstone, Handbook of applied Cryptography, CRC Press, 1997.

Md. Mokammel Haque received his MS degree in Computer Engineering in 2008 from the Department of Computer Engineering, Kyung Hee University, South Korea. He received his B.Sc. degree in Computer Science and Engineering in 2004 from Chittagong University of Engineering and Technology (CUET), Bangladesh. His research interests include mobile ad hoc networks, security, and applications in sensor networks. He is currently serving as a lecturer in the Department of Computer Engineering, CUET. He received the IEEE student travel grant award in IEEE MILCOM 2007, USA. He is a student member of IEEE and a member of IEB (Institution of Engineers, Bangladesh).

Al-Sakib Khan Pathan is working towards his PhD at Networking Lab, Department of Computer Engineering in Kyung Hee University, South Korea. He received his B.Sc. degree in Computer Science and Information Technology from Islamic University of Technology, Bangladesh in 2003. Before joining Networking Lab, he was with CSE Laboratories, North South University, Bangladesh. His research interests include Wireless Networking, Wireless Sensor Networks, Network security and E-Services Technologies. He was awarded a 'Scholar Student of the Year' (for foreign MS/PhD students) twice in 2006 and 2007 in South Korea. He is also the recipient of the IEEE ComSoc Seoul Chapter best paper award in JCCI 2007 in South Korea, and the IEEE student travel grant award in IEEE MILCOM 2006, USA. He is a student member of IEEE, associate member of Korea Information Processing Society (KIPS) and member of Korean Institute of Information Scientists and Engineers (KIISE).

Choong Seon Hong received his B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, Korea, in 1983, 1985, respectively. In 1988 he joined KT, where he worked on Broadband Networks as a member of the technical staff. In September 1993, he joined Keio University, Japan. He received a Ph.D. degree at Keio University in March 1997. He had previously worked for Telecommunications Network Lab, KT, as a senior member of the technical staff and a director of the networking research team until August 1999. Since September 1999, he has been working as a professor of School of Electronics and Information, Kyung Hee University. He has served as a Program Committee Member and an Organizing Committee Member for international conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN and TINA. His research interests include ad hoc networks, network security and network management. He is a member of IEEE, IPSJ, KIPS, KICS and KISS.

Eui-Nam Huh has earned a BS degree from Busan National University in Korea, a Master's degree in Computer Science from University of Texas, USA in 1995 and a Ph. D degree from Ohio University, USA in 2002. He was a director of Computer Information Center and Assistant Professor in Sahmyook University, South Korea during the academic year 2001 and 2002. He has also served in the WPDRTS/IPDPS community as program chair in 2003. He has been an editor of Journal of Korean Society for Internet Information and Korea Grid Standard group chair since 2002. He was also an Assistant Professor at Seoul Women's University, South Korea. Currently, he is at Kyung Hee University, South Korea, as Professor in Dept. of Computer Engineering. His research areas of interest are: High Performance Networks, Sensor Networks, Distributed Real-Time Systems, Grid Middleware Monitoring, and Network Security.