On the Insecurity of Asymmetric Key-based Architecture in Wireless Sensor Networks

Abedelaziz Mohaisen[1], Jeong Woon Choi[2] and Dowon Hong[2]
[1] Computer Science Department, University of Minnesota
Minneapolis, MN 55455 - USA
[e-mail: mohaisen@cs.umn.edu]
[2] Electronics and Telecommunication Research Institute
Daejeon 305-700, South Korea
[e-mail: {jw_choi, dwhong}@etri.re.kr]
*Corresponding author: Dowon Hong

Received May 20, 2009; revised July 11, 2009; accepted July 17, 2009; published August 31, 2009

Abstract

In this article, we demonstrate that the asymmetric key-based architecture for securing wireless sensor networks recently introduced by Haque et al. is insecure under impersonation attack, since it does not provide authentication semantics. In addition, we show that, for the scheme to work correctly, the resulting key distribution construction should be symmetric and group-wise.

Keywords: Wireless sensor neworks, security architecture, key management, security analysis

This work was supported by the IT R&D program of MIC/IITA, the Korean government [2005-Y-001-05, Development of next generation security technology].

DOI: 10.3837/tiis.2009.04.003

1. Introduction

The security of wireless sensor networks (WSNs) is a challenging and exciting issue that has attracted a great deal of recent attention. This has resulted in many research contributions. These contributions considered both symmetric and asymmetric key based algorithms, as potential solutions. In particular, recent work questioned the long-standing claim that the asymmetric key based cryptography (AKC) is inefficient on resource-constrained sensor nodes and demonstrated relevant efficiency [1][2]. These results motivated the need for new designs of public key primitives suited to WSN settings [3] in addition to conventional public key primitives, such as key authentication [4], key revocation [5], and key distribution [2].

Recently, Haque et al. introduced *asymmetric key-based architecture* (AKA) to secure WSN [6]. AKA uses a linear construction based on the *pseudo-inverse* of a matrix to generate symmetric keys. These keys are ultimately used to secure communication between nodes and the base-station and to distribute keys required for node-to-node communication. AKA is asserted to be computationally more secure than the Diffie-Hellman key exchange protocol.

Our original contribution in this paper is twofold: first, we show that AKA is *insecure* by demonstrating that secret parameters at the side of one node, which are used to derive the secret key, can be derived by any malicious entity that impersonates the base-station. Second, we show that, in order for AKA to work correctly, the overall settings should be the group-wise symmetric key model. Finally, to avoid some of the criticisms directed at AKA, we introduce a proposal for a recovery mechanism.

The structure of this article is as follows. We first describe AKA and its basic assumptions in section 2 (more details are in [6]). In section 3, we analyze AKA security. In section 4, we introduce related work from the literature and our work in relation to them. This is followed by concluding remarks in section 5.

2. Related Works

The key distribution problem in WSNs has been thoroughly treated in the literature. Particularly, several constructions are introduced based on computationally hard problems. For instance, Liu et al. introduced a scheme that utilizes bivariate symmetric polynomials for key distribution [7] that exploits the difficulty of the polynomial factorization problem. Du et al. introduced a scheme that utilizes a symmetric matrix construction for key distribution [8] that exploits the linear independence merit of vectors to the solvability of linear systems (that is, difficulty of solving a system in n variables given $t<n$ equations). These original works have been extended, improved, and utilized for special scenarios, as in [9], [10], [11], [12], [13], and [14]. Other key assignment schemes *to improve* connectivity and resiliency are introduced. For instance, the early work of Eschenauer and Gligor [15] uses a random key assignment method. Blackburn et al. goes one step further, by utilizing Costas arrays to improve resiliency and reduce overhead [16].

3. Overview of AKA

AKA does not follow the conventional asymmetric architectures that utilize asymmetric algorithms to distribute symmetric keys (session keys) in an authentic manner. That is, AKA uses symmetric keys to distribute other symmetric keys to be used to decrypt communication

traffic between nodes in a symmetric manner that makes the naming of AKA inaccurate. Technically, AKA consists of two phases: key derivation and secure communication. In key derivation, AKA utilizes the pseudo-inverse of a matrix to establish a secret shared key between any node and the base station without revealing any of these entities' secret information. The non-unique pseudo-inverse of a matrix $\mathbf{A} \in \{0,1\}^{m \times n}$ is matrix $\mathbf{A}^{+} \in \{0,1\}^{n \times m}$ that satisfies the following properties:

$$\mathbf{A}\mathbf{A}^{+}\mathbf{A} = \mathbf{A} \quad (1a)$$

$$\mathbf{A}^{+}\mathbf{A}\mathbf{A}^{+} = \mathbf{A}^{+} \quad (1b)$$

Taking these properties into account, the key derivation phase between a node *Alice* and the base station, which is considered a trusted third party (*TTP*), is performed based on the description in **Fig 1**.

Participants: node (*Alice*) and base-station (*TTP*).

1. *Alice*: generate $\mathbf{X}_A \in \{0,1\}^{m \times n}$ where $m < n$, compute $\mathbf{X}_A^{+} \in \{0,1\}^{n \times m}$ and compute $\mathbf{X}_A^{+}\mathbf{X}_A \in \{0,1\}^{n \times n}$. Send $\mathbf{X}_A^{+}\mathbf{X}_A$ to *TTP*.
2. TTP: generate $\mathbf{Y}_A \in \{0,1\}^{n \times k}$ where $n < k$, compute $\mathbf{Y}_A^{+} \in \{0,1\}^{k \times n}$ and compute $\mathbf{Y}_A\mathbf{Y}_A^{+} \in \{0,1\}^{n \times n}$. Compute $a_1 = \mathbf{X}_A^{+}\mathbf{X}_A\mathbf{Y}_A$ and $a_2 = \mathbf{X}_A^{+}\mathbf{X}_A\mathbf{Y}_A\mathbf{Y}_A^{+}$ and send them both to *Alice*.
3. *Alice*: compute $\mathbf{X}_A(a_1) = \mathbf{X}_A\mathbf{Y}_A$ (secret key) and $b_1 = \mathbf{X}_A(a_2) = \mathbf{X}_A\mathbf{Y}_A\mathbf{Y}_A^{+}$. Send b_1 to *TTP*.
4. *TTP*: Compute $(b_1)\mathbf{Y}_A = \mathbf{X}_A\mathbf{Y}_A$ (secret key).

Fig. 1. The key derivation phases of AKA

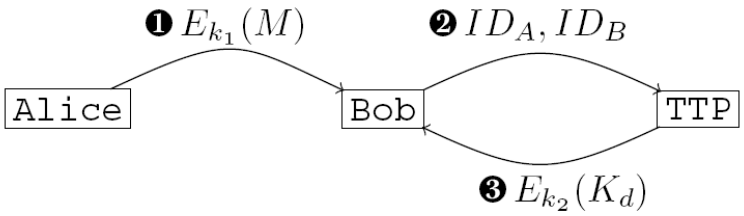

Fig 2. Key exchange for node-to-node communication: the order of the message is according to the numbers in the direction of the arrows. $k_2 = \mathbf{X}_B\mathbf{Y}_B$ and $k_1 = \mathbf{X}_A\mathbf{Y}_A$.

When two nodes, namely *Alice* and *Bob*, need to communicate securely using the assumption that they already both share secret keys with the base station (*TTP*), the procedure in **Fig. 2** is performed. A message intended for *Bob* from *Alice* is decrypted by a decryption key passed to *Bob* from the *TTP* in a secure manner.

Participants: node (*Alice*) and base-station (*Bob*) who share a group generator g and a prime p .

1. *Alice*: generate $a \in Z_p$, compute $A = g^a \bmod p$, and send A to *Bob*.
2. *Bob*: generate $b \in Z_p$, $B = g^b \bmod p$, and send B to *Alice*. *Bob* compute $K_{AB} = A^b \bmod p = g^{ab} \bmod p$ as the shared key.
3. *Alice*: receive B and compute $K_{AB} = B^a \bmod p = g^{ab} \bmod p$ as the shared key. The keys in both of step (2) and step (3) are equal.

Fig 3. The Diffie-Hellman Key Exchange

4. Security Analysis

4.1 Diffie-Hellman Key Exchange

We touch upon the security of Diffie-Hellman key exchange protocol (D-H) [17] that is used to demonstrate the security of AKA in [6], before detailing AKA security. The D-H protocol is a symmetric key establishment protocol that enables two parties to construct a key without prior knowledge of each other, as shown in **Fig. 3** [17]. D-H protocol security is based on the size of the group from which the prime p is generated that also determines the size of the resulting key. The difficulty of breaking D-H is equivalent to the difficulty of solving the discrete logarithm problem (DLP). Based on the *Pohlig-Hellman* algorithm, the time complexity to solve the general DLP has the complexity of $O(\sqrt{p})$. For instance, a 128 bit key provides a 64 bit security level. For special groups, from which p is generated, the general number field sieve (GNFS) can be applied to solve the DLP with complexity of $L_p[\frac{1}{3}, 1.923]$. This form of complexity can be represented in *O*-notation as [7]:

$$L_p[\frac{1}{3}, 1.923] = O(e^{(1.923+o(1))(\log n)^{1/3}(\log\log n)^{2/3}})$$

A demonstration of the complexity to solve DLP for several key size values based on different algorithms is shown in **Table 1**. Contrary to that shown in [6], and as we will show later in this article, AKA can be broken in a linear number of operations. Conversely, DLP is believed to be a hard problem and the best known algorithm to solve such a problem requires exhaustive search. **Table 1** illustrates the search space required for different key sizes (in bits) for different algorithms.

Table 1. Complexity of the discrete logarithm problem: note that p can be chosen carefully so that GNFS cannot be applied. PH denotes the *Pohlig-Hellman* algorithm

Key size (bit)	GNFS (bit)	BF (bit)	PH (bit)
100 bit	57.3423	99	50
200 bit	79.3255	199	100
300 bit	95.3806	299	150
400 bit	108.3934	399	200
500 bit	119.7412	499	250

4.2 Impersonation Attack on AKA

Eve is an attacker who would like to impersonate the base station. Since there is no semantics for authentication prior to the key establishment process, Eve can receive $\mathbf{X}_A^+\mathbf{X}_A$ from Alice, fabricate her own $\mathbf{Y}_{eve}$, such that at least *n* number of rows in $\mathbf{Y}_{eve}$ are linearly independent, compute $\mathbf{Y}_{eve}^+$, $\mathbf{Y}_{eve}\mathbf{Y}_{eve}^+$, $\mathbf{X}_A^+\mathbf{X}_A\mathbf{Y}_{eve}\mathbf{Y}_{eve}^+$, $\mathbf{X}_A^+\mathbf{X}_A\mathbf{Y}_{eve}^+$ and send the last two results to *Alice*. In response, *Alice* computes her key and sends $\mathbf{X}_A\mathbf{Y}_{eve}\mathbf{Y}_{eve}^+$. Given the aforementioned linear independence property, *Eve* can ensure that $\mathbf{Y}_{eve}\mathbf{Y}_{eve}^+$ is invertible. Eve then computes $(\mathbf{Y}_{eve}\mathbf{Y}_{eve}^+)^{-1}$ to obtain

$$\mathbf{X}_A\mathbf{Y}_{eve}\mathbf{Y}_{eve}^+(\mathbf{Y}_{eve}\mathbf{Y}_{eve}^+)^{-1} = \mathbf{X}_A \tag{2}$$

Since $\mathbf{X}_A^+$ is not unique, *Eve* can find a matrix $(\mathbf{X}_A^+)'$, such that $(\mathbf{X}_A^+)'\mathbf{X}_A\mathbf{Y}_A = \mathbf{X}_A^+\mathbf{X}_A\mathbf{Y}_A$. Finding the appropriate $(\mathbf{X}_A^+)'$ is not easy: for the general case, it can be as complex as the brute force search in a space of $m \times n$. However, since *Eve* also knows $\mathbf{X}_A$ and $\mathbf{X}_A^+\mathbf{X}_A$, she can use them both to reduce the complexity of finding $\mathbf{X}_A^+$ to a linear number of operations. Let $\mathbf{X}_A^+\mathbf{X}_A = \mathbf{R}_A \in \{0,1\}^{n\times n}$ be

$$\mathbf{R}_A = \begin{pmatrix} r_{00} & r_{01} & \cdots & r_{0n} \\ r_{10} & r_{11} & \cdots & r_{1n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n0} & r_{n1} & \cdots & r_{nn} \end{pmatrix} \tag{3}$$

Let $\mathbf{X}_A^+$ and $\mathbf{X}_A$ be represented as

$$\mathbf{X}_A^+ = \begin{pmatrix} x_{00}^+ & x_{01}^+ & \cdots & x_{0n}^+ \\ x_{10}^+ & x_{11}^+ & \cdots & x_{1n}^+ \\ \cdots & \cdots & \cdots & \cdots \\ x_{n0}^+ & x_{n0}^+ & \cdots & x_{nn}^+ \end{pmatrix} \tag{4}$$

$$\mathbf{X}_A = \begin{pmatrix} x_{00} & x_{01} & \cdots & x_{0n} \\ x_{10} & x_{11} & \cdots & x_{1n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n0} & x_{n1} & \cdots & x_{nn} \end{pmatrix} \tag{5}$$

Now, since $\mathbf{R}_A \in \{0,1\}^{n\times n}$ is known to any eavesdropper, including *Eve,* as it is communicated over an insecure channel, and because $\mathbf{X}_A$ is already known to Eve through the *invertibility* attack, Eve performs the following. Since $r_{ij} \in \mathbf{R}_A$ is computed as

$$r_{ij} = \sum_{t=1}^{m} x_{it}^{+} x_{tj} \bmod 2 \tag{6}$$

and, since the summation of modular 2 is the typical exclusive or operation, *Eve* can construct the linear system in **Fig. 4**. There are $m \times n$ unknowns in these n^2 number of equations known to Eve. However, since there are several zero values in each matrix, which leads to some of the unknowns (variables) being removed from these equations, the solvability of this system of equations is possible when $n = 2m$ for $P_r(x_{ij} = 1) = P_r(x_{ij} = 0) = \frac{1}{2}$. This condition is rational according to the settings of AKA.

This attack can also be applied by Alice to know the secret of the *TTP*. That is, the *TTP* first sends $\mathbf{X}_A^{+}\mathbf{X}_A\mathbf{Y}_A$ to *Alice*. However, since *Alice* already knows term $\mathbf{X}_A^{+}\mathbf{X}_A$, she can compute an arbitrary matrix $\mathbf{Y}_A^{'}$ that satisfies the above equality based on the method described above.

Alice can deviate, by selecting the appropriate parameters that make the term $\mathbf{X}_A^{+}\mathbf{X}_A$ invertible, to compute the exact $\mathbf{Y}_A$ generated by the *TTP*. Once she receives the term $\mathbf{X}_A^{+}\mathbf{X}_A\mathbf{Y}_A$ she computes $(\mathbf{X}_A^{+}\mathbf{X}_A)^{-1}\mathbf{X}_A^{+}\mathbf{X}_A\mathbf{Y}_A = \mathbf{Y}_A$.

$$r_{00} = x_{00}^{+} \bullet x_{00} \oplus x_{01}^{+} \bullet x_{10} \oplus x_{02}^{+} \bullet x_{20} \oplus \cdots \oplus x_{0(m-2)}^{+} \bullet x_{(m-2)0} \oplus x_{0(m-1)}^{+} \bullet x_{(m-1)0} \oplus x_{0(m)}^{+} \bullet x_{(m)0}$$
$$r_{01} = x_{00}^{+} \bullet x_{01} \oplus x_{01}^{+} \bullet x_{11} \oplus x_{02}^{+} \bullet x_{21} \oplus \cdots \oplus x_{0(m-2)}^{+} \bullet x_{(m-2)1} \oplus x_{0(m-1)}^{+} \bullet x_{(m-1)1} \oplus x_{0(m)}^{+} \bullet x_{(m)1}$$
$$r_{10} = x_{10}^{+} \bullet x_{00} \oplus x_{11}^{+} \bullet x_{10} \oplus x_{12}^{+} \bullet x_{20} \oplus \cdots \oplus x_{1(m-2)}^{+} \bullet x_{(m-2)0} \oplus x_{1(m-1)}^{+} \bullet x_{(m-1)0} \oplus x_{1(m)}^{+} \bullet x_{(m)0}$$
$$\vdots$$
$$r_{ij} = x_{i0}^{+} \bullet x_{0j} \oplus x_{i1}^{+} \bullet x_{j1} \oplus x_{i2}^{+} \bullet x_{2j} \oplus \cdots \oplus x_{i(m-2)}^{+} \bullet x_{(m-2)j} \oplus x_{i(m-1)}^{+} \bullet x_{(m-1)j} \oplus x_{i(m)}^{+} \bullet x_{(m)j}$$
$$r_{ji} = x_{j0}^{+} \bullet x_{0i} \oplus x_{j1}^{+} \bullet x_{i0} \oplus x_{j2}^{+} \bullet x_{2i} \oplus \cdots \oplus x_{j(m-2)}^{+} \bullet x_{(m-2)i} \oplus x_{j(m-1)}^{+} \bullet x_{(m-1)i} \oplus x_{j(m)}^{+} \bullet x_{(m)i}$$
$$\vdots$$
$$r_{nn} = x_{n0}^{+} \bullet x_{0n} \oplus x_{n1}^{+} \bullet x_{1n} \oplus x_{n2}^{+} \bullet x_{2n} \oplus \cdots \oplus x_{n(m-2)}^{+} \bullet x_{(m-2)n} \oplus x_{n(m-1)}^{+} \bullet x_{(m-1)n} \oplus x_{n(m)}^{+} \bullet x_{(m)}$$

Fig. 4. The gate-level linear system to compute $\mathbf{X}_A^{+}$ where $\oplus$ is a bitwise exclusive-or and $\bullet$ is a bitwise AND gate

4.3 Symmetric versus Asymmetric

Now recall the node-to-node communication in AKA shown in **Fig. 2**. After *Alice* sends an encrypted message to *Bob*, *Bob* requests the decryption key from the *TTP*, which in return sends it to *Bob* encrypted, using a secret key shared between the *TTP* and *Bob*. For this to work, the exchanged key should be the same key shared between the *TTP* and *Alice,* since the encryption scheme applied on *Alice's* side is symmetric; this makes the naming of AKA inaccurate. This limitation in AKA exposes two security problems:

1. Since the key used to encrypt messages from *Alice* to *Bob* is the same key used for secure communication between the *TTP* and Alice, once Bob knows the key passed to him via the *TTP* he will be able to intercept, decrypt and manipulate messages from

Alice that are not directed to him, unless the key is refreshed each time Alice wants to send a message at the expense of tremendous overhead.

2. Once Alice's key is revealed to Bob, Bob will be able to forge messages and send them to the TTP on behalf of Alice requesting the shared keys between the TTP and any arbitrary node. This will be affordable, since there is provision for authentication between the TTP and nodes.

After the execution of the instantiation of AKA in **Fig. 5**, both Alice and Bob will know each other's keys. This will be a critical issue, unless key refreshment is performed immediately each time. Utilizing the above scenarios will enable the participants to gain access to secret information used to generate the keys. In contrast, in the D-H key agreement settings, even if Eve knows g and b (as in the description of D-H), she cannot obtain the corresponding random value of Alice. That is, given a, g, g^{ab}, and g^{a}, there is no efficient method by which Eve can compute b (also known as the computational Diffie-Hellman problem). □

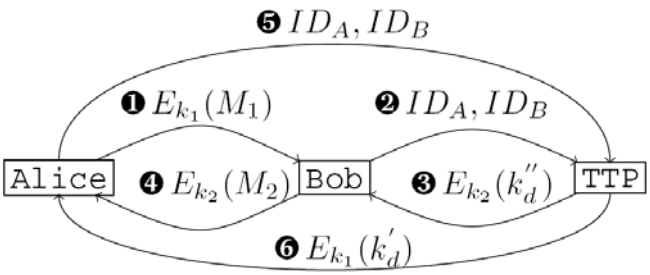

Fig. 5. Key exchange for node-to-node communication: the order of the message is according to the numbers and arrows. $k_2 = \mathbf{X}_B\mathbf{Y}_B$ and $k_1 = \mathbf{X}_A\mathbf{Y}_A$

4.4 Possible Countermeasure

The only possible countermeasure for the impersonation attack provided in section 3.2, is to select the parameters carefully, so that the *invertibility* of the linear construction that results from multiplying the matrix and its pseudo inverse at either side is impossible. That is, if we select $k < n$, we can always ensure that $\mathbf{Y}_{eve}\mathbf{Y}_{eve}^{+}$ has no inverse. If the deviating TTP cannot compute $(\mathbf{Y}_{eve}\mathbf{Y}_{eve}^{+})^{-1}$, then it also cannot compute Eq. (2) and cannot proceed with an impersonation attack. Both key authentication and key refresh processes are needed to mitigate the impact of the attacks in section 3.3. Strong authentication is indeed needed to overcome both attacks in section 3.2 and section 3.3.

4.5 On the Existence of a Trusted Third Party

AKA depends greatly in its operation on assuming that the base-station is a trusted third party. In the majority of WSN systems, this assumption is unrealistic for so many reasons. In the following, we mention two of these reasons.

1. One of these reasons is the cost of this assumption. While commercial WSN systems are desired to be reasonably priced, assuming the existence of a TTP in WSN system will be at high cost.

2. The other reason is the deployment scenario. The deployment scenarios of WSN limit the rationality of this assumption since many of these scenarios assume a hostile and adversarial environment as a basic natural assumption which contradicts with the existence of a TTP in the system. Particularly, assuming a hostile environment of deployment implies that any potential attacker can physically capture any entity in the network including the base-station, alter its contents, and act on behalf on it.

For both of the above reasons, among many other reasons, TTP is considered unrational assumption in the majority of WSN systems [7]. However, if the base-station in a WSN is considered a TTP, many traditional key distribution algorithms, that are proven to be secure against several attacks, can be brought to the WSN systems. This for instance includes key distribution centers (KDC) among others [7][11]. These algorithms, if brought to WSN systems, will minimize several security challenges and at low computational cost that is comparable to the cost of AKA.

5. Conclusion

In this article, we proved the asymmetric key-based architecture (AKA) for wireless sensor network to be insecure. Particularly, AKA is not based on a computationally hard problem. This makes breaking it in a linear number of operations possible. We suggest the existence of a strong authentication method prior to key establishment and careful selection for AKA's parameters to avoid this.

References

[1] A. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” In *PerCom*, pp. 324-328, 2005.
[2] D.J. Malan, M. Welsh, and M.D. Smith, “A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography,” In *First IEEE Int. Conf. on Sensor and Ad Hoc Comm. and Networks*, pp. 71-80, 2004.
[3] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz, “Comparing elliptic curve cryptography and rsa on 8-bit CPUs,” In *CHES*, pages 119-132, 2004.
[4] D. Nyang and A. Mohaisen, “Cooperative public key authentication protocol in wireless sensor network,” In *UIC*, pp. 864-873, 2006.
[5] A. Mohaisen, D. Nyang, Y. Maeng, and K. Lee, “Structures for communication efficient public key revocation in ubiquitous sensor network,” In *MSN*, pp. 822-833. Springer, 2007.
[6] Md. M. Haque, A.-S.K. Pathan, C.S. Hong, and E. Huh, “An asymmetric key-based security architecture for wireless sensor networks,” *TIIS*, vol. 2, no. 5, pp. 265-279, 2008.
[7] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” In *ACM CCS*, pp. 52-61, 2003.
[8] W. Du, J. Deng, Y.S. Han, P.K. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks,” *ACM Trans. Inf. Syst. Secur.*, vol. 8, no.2, pp. 228-258, 2005.
[9] A. Mohaisen, Y. Maeng, and D. Nyang, “On grid-based key pre-distribution: Toward a better connectivity in wireless sensor network,” In *PAKDD Workshops*, pp. 527-537, 2007.
[10] A. Mohaisen and D. Nyang, “Hierarchical grid-based pairwise key predistribution scheme for wireless sensor networks,” In *EWSN*, pp. 83-98, 2006.
[11] D. Liu, P. Ning, and R. Li, “Establishing pairwise keys in distributed sensor networks,” *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 1, pp. 41-77, 2005.
[12] T. Ito, H. Ohta, N. Matsuda, and T. Yoneda, “A key pre-distribution scheme for secure sensor networks using probability density function of node deployment,” In *SASN*, pp. 69-75. ACM, Oct.

2005.
[13] A. Mohaisen, D. Nyang, Y. Maeng, K. Lee, and D. Hong, "Grid-based key pre-distribution in wireless sensor networks," *TIIS*, vol. 3, no. 2, pp. 195-208, 2009.
[14] A. Mohaisen, D. Nyang, and T. AbuHmed, "Two-level key pool design-based random key pre-distribution in wireless sensor networks," *TIIS*, vol. 2, no. 5, pp. 222-238, 2008.
[15] L. Eschenauer and V.D. Gligor, "A key-management scheme for distributed sensor networks," In *ACM CCS*, pp. 41-47, 2002.
[16] S.R. Blackburn, T. Etzion, K.M. Martin, and M.B. Paterson, "Efficient key predistribution for grid-based wireless sensor networks," In *ICITS*, vol. 5155 of LNCS, pp. 54-69, 2008.
[17] W. Di_e and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

Abedelaziz Mohaisen is a Ph.D. student at the University of Minnesota Twin Cities. He was a member of the engineering staff at the Electronics and Telecommunication Research Institute (ETRI), in Korea, from 2007 to 2009. He received a B.E. degree in computer engineering from the University of Gaza, in Palestine, in 2005 and a M.E. degree in information and telecommunication engineering from Inha University, in Korea, in 2007. His research interests include networks security, data privacy, and cryptography. He is a member of ACM, IEEE, and KSII.

Jeong Woon Choi is a member of the engineering staff at the electronics and telecommunication research institute in Korea and part of the cryptography research team. He received his B.S., M.S., and Ph.D. in mathematical sciences from Seoul National University, Seoul, Korea, in 2002, 2004, and 2009, respectively. His research interests include quantum cryptography, applied cryptography and security in general.

Dowon Hong received his B.S., M.S. and Ph.D. degrees in mathematics from Korea University, Seoul, Korea in 1994, 1996, and 2000. He is currently a senior member of the engineering staff and team leader of the Cryptography Research team at the Electronics and Telecommunication Research Institute, Korea. His research interests are broadly in the areas of applied cryptography, network security, and digital forensics.