On the Inefficiency of the Resources Optimal Key Pre-Distribution Scheme for Wireless Sensor Network

Abedelaziz Mohaisen[1] and DaeHun Nyang[2]
[1]Computer Science and Engineering Department, University of Minnesota, Minneapolis, MN 55455, USA
Email: mohaisen@cs.umn.edu
[2]Graduate School of Information Technology and Telecommunication, Inha University, Incheon, Korea
Email: nyang@inha.ac.kr

Abstract— **In this paper we re-evaluate the resources requirements of the "resources optimal key pre-distribution (RKPD) scheme in wireless sensor networks". Our evaluation shows that RKPD has excessive requirements in terms of memory, computation, and communication. Thees requirements are problematic for that they make RKPD less beneficial by violating the purpose that it was designed for. Furthermore, because RKPD is a hybrid scheme that uses two well-known schemes in literature, we show that RKPD inherently have two security flows that challenge its chances of being deployment in real wireless sensor network. Detailed analysis, comparisons and examples are provided to evidence our arguments and conclusions.**
Key words: Key pre-distribution, sensor networks, resources re-evaluation, security analysis.

I. INTRODUCTION

The security of wireless sensor networks (WSNs) is a challenging issue that has attracted a great research interest where several security aspects have been thoroughly researched and solutions have been introduced. One of the fundamental issues researched in the context of WSN is the key pre-distribution (KPD), an essential step toward deploying symmetric key algorithms which are shown empirically to be computationally light for securing typical WSNs [1]–[4]. In typical KPD schemes, sets of keys or keying materials are assigned to each sensor node in a pre-deployment phase and then used for securing communication traffic between sensor nodes in an online phase. The need for KPD is motivated and necessitated by the fact that WSN lacks infrastructure which makes the existence of centralized key distribution center to provide online key distribution almost impossible [1], [5]. Another equally important reason is that sensor nodes have limited resources featured by short range communication, low computation power, and limited amount of memory which signify the need for efficient key pre-distribution schemes [2]–[4]. An efficient KPD scheme is characterized by its strong resiliency which implies that a typical WSN that uses such scheme will still recover from attacks and minimize the impact of compromised nodes on the security of other uncompromised nodes.

This work was supported by a research grant from Inha University. DaeHun Nyang is the corresponding author.

To meet both resources and security requirements in WSN, Dai et al. [6] have recently introduced a resource-optimal KPD (RKPD) scheme that utilizes two well-known schemes from literature. RKPD is claimed to have comparable security to that of the original works from which it is designed as well as having minimal memory, computation, and communication requirements. In spite of that, we show that RKPD requires an extensive amount of resources as well as being insecure. Particularly, we re-evaluate the resources requirements and show that RKPD requires the sum of resources required for both schemes used in building it. Since RKPD uses two different scheme for building halves of keys using each, we point out a critical misconception on the claimed security of RKPD and demonstrate its insecurity. We argue that RKPD does not provide any substantial benefit over the two schemes used in building it when considered apart.

The rest of this article is organized as follows. In section II we review RKPD scheme in relation with the two schemes used in building it. In section III we re-evaluate the memory, communication and computation required for each of the three different schemes and compare them according to their resource requirements. In section IV, we analyze the security of the RKPD and compare it to the two schemes which is based on. In section V summarize some of the related works followed by some concluding remarks in section VI.

II. PRELIMINARIES

To address the key distribution challenge in WSN, Park et al. introduced a new secure KPD scheme based on the concept of lower-upper (LU) decomposition of symmetric matrices used for storing the whole set of network keys [7]. With the same goal in mind, Du et al. introduced another scheme based on the symmetry property of matrices [3]. With both security and resource requirements as two objectives in mind, Dai et al. introduced a hybrid scheme that uses both of these schemes [6]. In this section, we provide the underlaying technical details of each of the three different schemes and motivate for the evaluation of their security. Because the LU decomposition scheme is used for a single space key distribution model, we explain

doi:10.4304/jcm.5.2.164-168

the details of the single space version of the DDHV for fair comparison noting that multi-space key distribution model is a straightforward extension.

A. LU Decomposition Scheme

LU decomposition of matrix A is a matrix factorization which writes a matrix A as the product of a lower and an upper triangular matrices according to the form $A = LU$. In the context of KPD in WSN, the matrix A is basically a symmetric with the following particular properties:

- The matrix A is symmetric which means that two elements a_{ij} and a_{ji} in A are equal for $0 < i \leq n$ and $0 < j \leq n$. Note tat symmetry property implies that A is a square matrix and has a full rank.
- All elements of A are randomly generated with non-zero values over a finite field $\mathbb{Z}_q$. That is, each element in the matrix A is represented as binary sequence of q bits.
- All rows and columns of A are linearly independent. This important property is definite because A has a full row rank and a full column rank.

The LU decomposition scheme for KPD consists of two phases: offline phase and online phase [7]. The offline phase is performed on a key distribution server for generating keying materials and assigning them to the nodes while the online phase is performed at the operation time of the network for establishing pairwise keys between pairs of nodes. Technical, the offline phase of the LU decomposition scheme is performed as follows:

1) A symmetric matrix A of size $n \times n$ is constructed with random elements in a finite field $\mathbb{Z}_q$.
2) The matrix A is decomposed into L and U using the proper LU decomposition algorithm (e.g., Doolittle algorithm). The cost of computation is determined by the size of A since all of its elements are non-zero valued. Though, in the analysis part we discard this overhead because the procedure is performed at the server side where overhead is not a concern.
3) The i-th row $L_r^{(i)}$ of of the matrix L and the i-th column $U_c^{(i)}$ of matrix U are to node s_i. $L_r^{(i)}$ is kept private and $U_c^{(i)}$ is declared public during the operation time of the network.

The offline phase is performed at the operation time of the network. This phase is initiated when two nodes, s_i and s_j, need to establish a pairwise key between each other. In this phase, the two nodes first who want to establish a pairwise key first exchange the public columns and compute a product as follows:

$$s_i : k_{ij} = L_r^{(i)} U_c^{(j)}$$
$$s_j : k_{ji} = L_r^{(j)} U_c^{(i)}$$

The resulting inner product of the two vectors is the secret used as a pairwise key between the two nodes. The security of the LU decomposition scheme is established upon the hardness of recovering the matrix A. This hardness is equivalent to the computational effort required to recover n independent rows in the matrix L given matrix U. It was believed that the LU decomposition scheme is secure for up to n nodes compromise (where n is also the order of the matrix A) [7], however recent results have shown that the LU decomposition scheme is entirely insecure [8][1]. Given the matrix U an attacker can compute the matrix L without any further information about A beside knowing that A is symmetric. Particularly, because the matrix A is symmetric and the LU decomposition is an elementary row process that inherits the symmetry property of the matrix A, the ratio between the column elements in U and the row elements in L is identical [8].

B. DDHV Scheme

Another matrix-based scheme that utilizes the symmetry property of a matrix for assigning pairwise symmetric keys for sensor nodes in a WSN of n nodes is introduced by Du et al. [3]. This work, which will be referred as DDHV for brevity, extends results that were introduced earlier by Blom [9]. For its basic form, the following matrices are defined: a public matrix G of size $(\lambda+1) \times n$ and a private symmetric matrix D of size $(\lambda+1) \times (\lambda+1)$ where elements of G and D are randomly generated over the finite field $\mathbb{Z}_q$. A matrix A is computed as $A = (DG)^T$. The size of A is $n \times (\lambda+1)$. For any node, s_i, a row $A_r^{(i)}$ from A and a column $G_c^{(i)}$ from G are assigned. When s_i and s_j need to establish a key between each other, they first exchange their public information $G_c^{(i)}$ and $G_c^{(j)}$ respectively. Then, $k_{ij} = A_r^{(i)} \times G_c^{(i)}$ is computed by s_i and $k_{ji} = A_r^{(j)} \times G_c^{(i)}$ is computed by s_j and used as a pairwise key. Note that both keys are equal because the matrix from which they are derived is symmetric. That is,

$$\begin{aligned}(AG)^T &= ((DG)^T G)^T = G^T (DG) = (G^T D) G \\ &= (G^T D^T) G = (DG)^T G = AG\end{aligned}$$

To reduce the communication overhead, the DDHV scheme suggested the matrix G to be constructed according to the *Vandermonde matrix* in which column i is represented as $[(s^i)^0, (s^i)^1, \ldots, (s^i)^\lambda]$ where s is an initial seed. With this construction, node s_i stores only the field element s^i and generates the whole column from that element over modular operations. To construct the corresponding column from s^i, λ number of multiplications over $\mathbb{Z}_q$ are required. Similarly, to generate the key from A_r and G_c, another λ number of multiplications over $\mathbb{Z}_q$ are required. The DDHV scheme is secure for up to λ nodes' compromise. In other words, an attacker needs to know λ different and linearly independent vectors in D in order be able to reconstruct the matrix A which explicitly includes the private matrix D [3].

[1]This work was introduced at the time of writing this article. Though this attack can be utilized for the benefit of our work, our initial results have shown that, even while assuming a theoretical security of LU decomposition, the entire design of RKPD is disadvantaged with high resources requirements and security limitations (see section IV).

C. RKPD Scheme

Resource-optimal key pre-distribution scheme (RKPD) [6] combines both of the above schemes where the shared key between two nodes is the result of two halves. The first half is generated by calling the the LU decomposition scheme and the second half by calling the DDHV scheme. Technically, RKPD consists of the following two phases:

1) Offline Phase: for assigning the proper keying material to each node, these steps are performed:
   a) The i-th row $L_r^{(i)}$ in L and the i-th column $U_c^{(i)}$ in U are assigned to node s_i.
   b) The i-th column of matrix $G_c^{(i)}$ and i-th row of matrix $A_r^{(i)}$ are assigned to node s_i.
2) Online Phase: these steps are performed to establish the shared key between two nodes s_i and s_j:
   a) s_i and s_j exchange $U_c^{(i)}, U_c^{(j)}$ from matrix U and field seeds to generate $G_c^{(i)}$, $G_c^{(j)}$ from matrix G.
   b) Node s_i computes k_{ij}^1 and node s_j computes k_{ji}^1 as partial keys as follows:

$$k_{ij}^1 = L_r^{(i)} U_c^{(j)} = (\sum_{d=1}^{n} l_{jd} u_{id}) \mod 2^{\frac{|k|}{2}}$$

$$k_{ji}^1 = L_r^{(j)} U_c^{(i)} = (\sum_{d=1}^{n} l_{jd} u_{id}) \mod 2^{\frac{|k|}{2}}$$

   c) Node s_i computes k_{ij}^2 and node s_j computes k_{ji}^2 as partial keys as follows:

$$k_{ij}^2 = A_r^{(i)} G_c^{(j)} = (\sum_{d=1}^{\lambda} a_{id} g_{jd}) \mod 2^{\frac{|k|}{2}}$$

$$k_{ji}^2 = A_r^{(j)} G_c^{(i)} = (\sum_{d=1}^{\lambda} a_{jd} g_{id}) \mod 2^{\frac{|k|}{2}}$$

The shared key between node s_i and node s_j is made of concatenating the corresponding first half with the second half as $k_{ij} = k_{ij}^1 || k_{ij}^2 = k_{ji} = k_{ji}^1 || k_{ji}^2$. It can also be derived by using a one-way hash function $h(m) : \{0,1\}^{|m|} \rightarrow \{0,1\}^{|k|}$ as $k_{ij} = h(k_{ij}^1 || k_{ij}^2) = k_{ji} = h(k_{ji}^1 || k_{ji}^2)$. An illustration of the RKPD protocol is shown in Figure 1.

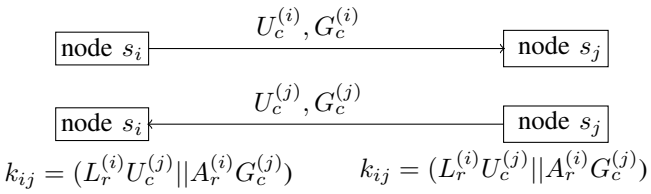

Figure 1. An illustration of the RKPD protocol.

III. Resources Requirements of RKPD

In this section we re-evaluate the required resources of the RKPD in terms of memory, computation, and communication. We assume that all operations in the DDHV and LU decomposition schemes are performed in modular q.

Theorem 1: RKPD consumes more resources than that of the LU decomposition and the DDHV schemes apart.

Proof: the proof follows from section III-A, section III-B, section III-C and Table I. ■

A. Memory Requirements

The RKPD scheme requires a space of memory for storing the public and private keying materials used for generating the symmetric pairwise keys. Since the RKPD uses both of the DDHV and LU decomposition schemes, it requires the sum of memory required of both schemes. In the LU decomposition scheme each node s_i must store a public column $U_c^{(i)}$ and a private row $L_r^{(i)}$. Each element in $U_c^{(i)}$ and $L_r^{(i)}$ requires q bit of storage. However, because the resulting partial key from the LU decomposition scheme has the length of $\frac{|k|}{2}$ bits when used for the RKPD, we can use a smaller space for generating these elements and storing them[2]. The number of elements in each row and column in the upper and lower triangular matrices is n elements. However, because the number of non-zero elements in any upper or lower triangular matrix is $n(\frac{n+1}{2})$, each node needs to store only $\frac{n+1}{2}$ non-zero elements *at average*. To store both $U_c^{(i)}$ and $L_r^{(i)}$, a node s_i needs $(n+1)q$ bits *at average*. Similarly, for the DDHV part, each node needs to store a column from the public matrix G and a row from the private matrix A at the expense of 2λ elements of storage. However, since any column in the matrix G can be represented by only one field element, the required memory is reduced to $(\lambda+1)q$. The overall memory for both scheme utilized for the RKPD is $(n+1)q + (\lambda+1)q = (n+\lambda+2)q$ bits.

B. Communication

According to Figure 1, two messages are exchanged between node s_i and node s_j for establishing a pairwise secret key. These messages exchange the public information between both nodes. The required communication overhead for that is $(\frac{n+1}{2}+1)q$ bits, where $(\frac{n+1}{2})q$ bits are required for representing the non-zero elements in the public column as in the LU decomposition and q bits are required for representing the field element of the public column in the DDHV scheme.

C. Computation

The multiplication of two vectors each of n elements requires n multiplications and $n-1$ additions over the field q. Because of the zero elements in the LU decomposition setting, only $\frac{n+1}{2}$ multiplications and $\frac{n+1}{2}-1$ additions are required. For constructing a public column from its field element in the DDHV scheme, $\lambda - 1$ multiplications are required. To construct the partial key in the DDHV scheme, λ multiplications and $\lambda - 1$ additions are required over the field q. The addition operations are negligible compared to computation overhead required for

[2]Note that q does not directly depend on the length of key.

multiplications making the overall required computation $\frac{n+1}{2} + \lambda$ multiplications.

A comparison between the three different schemes in terms of used resources is shown in Table I.

TABLE I.
A COMPARISON BETWEEN RKPD, DDHV, AND LU DECOMPOSITION IN TERMS OF MEMORY (IN BIT), COMMUNICATION (IN BIT) AND COMPUTATION (IN MULTIPLICATIONS OVER q).

Scheme	Memory	Communication	Computation
DDHV	$(\lambda+1)q$	q	(λ)
LU	$(n+1)q$	$(\frac{n+1}{2})q$	$\frac{n+1}{2}$
RKPD	$(\lambda+n+2)q$	$(\frac{n+1}{2}+1)q$	$\frac{n+1}{2}+\lambda$

IV. SECURITY ANALYSIS OF RKPD

In this section, we provide our insight on the security of the RKPD scheme. We show that combining the DDHV and the LU decomposition schemes does not improve the security neither reduce the resources consumption, contrary to what was claimed in [6]. Before detailing the security of the RKPD in relation with the DDHV and LU decomposition, we provide two definitions used as the metrics of security for KPD schemes.

Definition 1 (λ-security): A wireless sensor network of n nodes is said to be λ-secure if the compromise of any $K < \lambda$ will not affect the security of other than the compromised nodes.

Definition 2 ($(n/n$-security): A wireless sensor network of n nodes is said to be perfectly secure (or n/n-secure) if $\lambda = n$ in the Definition 1.

In this paper, the attack model is the standard "node capture" model. The adversary can observe all communications between nodes in the network and can capture a number of nodes to extract the keys stored in them [10].

Because the required resources for the RKPD are always higher than the resources required for both of DDHV and LU decomposition, fair comparison can not be performed on the ground of same resources consumption. We here also show that the comparison can not be held on the ground of same-security.

Suppose that the LU decomposition scheme is secure, an attacker can apply "node capture" attack on the network to compromise up to $\lambda + 1$ nodes from which the attacker can obtain sufficient information for breaking DDHV scheme. Once the DDHV scheme is broken, all partial keys constructed by the DDHV at each node will be reconstructed by the attacker and the key-related security determined by the length of secret key will merely dependent on the remaining half. This is, if the key length size is $|k|$ bits, the security of the RKPD scheme after compromising λ nodes is $\frac{|k|}{2}$ bits. This is particularly critical since λ is chosen much less than n for resources feasibility [3]. To understand this security problem, see the example below.

Example 1. for a key of size $|k| = 64$ bit, the attacker needs to try 2^{32} keys only under the above attack to obtain the partial key generated by the LU decomposition. For instance, the brute force attack on the full key of 64 bits that utilizes the encryption and verification of a known cipher-text/plan-text over a packet of 1024 bits on a 2.5 Gbps high speed encryption core [11] takes 7432.8 years. On the other hand, it takes only 27.3067 minutes to perform the same attack on half of the key using the same computational machine for same attack settings.

TABLE II.
COMPARISON BETWEEN THE SECURITY IN THE THREE DIFFERENT SCHEMES.

Scheme	Security
DDHV	$(\lambda)-$security
LU	insecure
RKPD	$\lambda-$security

Because the LU decomposition scheme is insecure under a combination of "node capture" and eavesdropping attacks, no further computational effort is required to reveal the entire key. The LU decomposition scheme was analyzed in [8] and shown to be insecure. Particularly, an eavesdropping attacker can listen to the communication taking place between the different sensor nodes in the network and collect sufficient information about the matrix U. Because the LU decomposition is performed using elementary operations that maintain the symmetry property, the ratio between the elements of rows in L and columns in U are identical. If the attacker compromise one node using the "node capture" attack and extract the secret information from it, including the private row, she can scale the ratio between elements in the different columns in U and obtain L. Now, we sum up with remarks featuring the security of the three schemes

1) The DDHV scheme provides λ-security. To achieve the highest possible security, we need to set $\lambda = n$ at the expense of high memory and computation (as shown in Table I.
2) The LU decomposition scheme is insecure regardless to the amount of resources consumed.
3) Since the RKPD combines both schemes, it requires an overhead in terms of memory, computation, and communication equivalent to the summation of both schemes' overhead. On the other hand, the security of the LU decomposition is computationally equivalent to the security of the DDHV scheme. Similar to DDHV scheme, the RKPD scheme can achieve the highest possible security when we set $\lambda = n$ at the expense of memory and computation that are double

V. RELATED WORKS

Several constructions are introduced literature to solve the problem of key distribution in WSN. For instance, Liu et al. introduced a scheme that utilizes bivariate symmetric polynomials for key distribution [12] and exploits the hardness of polynomial factorization problem. Du et al. introduced a scheme that utilizes a symmetric matrix construction for key distribution [3] which uses

the linear independence merit of vectors to the solvability of linear systems (I.e., hardness of solving a system in n variables given $t < n$ equations). These original works have been extended, improved, and utilized for special scenarios in [4], [13]–[15]. Other key assignment schemes *to improve* the connectivity and resiliency are introduced. For instance, the early work of Eschenauer and Gligor [1] uses a random key assignment method. Blackburn et al. go one step further by utilizing Costas arrays to improve resiliency and reduce the overhead [16].

In addition to these works on key distribution, some works are introduced on security analysis. For instance, Zhu et al. [8] analyzed the security of the LU decomposition scheme alone and showed its insecurity as explained in section IV. Paterson and Stinson in [10] introduced two attacks on Cheng-Agrawal scheme in [17]. Other instances of works where KPD is studied from a cryptographic prospect can be found in [18]–[20].

VI. Conclusion

In this article we evaluate the resources' requirements and security in the resource-optimal key pre-distribution (RKPD) scheme which uses both the DDHV and the LU decomposition schemes. We argued that RKPD does not provide any benefit over the DDHV scheme and that it consumes memory, computation, and communication equivalent to the sum of overheard consumed by DDH and LU decompsition scheme. Our result particularly shed the light on that not every combination of two schemes would necessarily provide a merit.

References

[1] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks." in *ACM CCS*, 2002, pp. 41–47.

[2] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks." in *ACM CCS*, 2003, pp. 52–61.

[3] 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.

[4] A. Mohaisen and D. Nyang, "Hierarchical grid-based pairwise key pre-distribution scheme for wireless sensor networks." in *EWSN*, 2006, pp. 83–98.

[5] H. Chan, A. Perrig, and D. X. Song, "Random key predistribution schemes for sensor networks." in *IEEE Symposium on Security and Privacy*, 2003, pp. 197–.

[6] T. T. Dai, A.-S. K. Pathan, and C. S. Hong, "A resource-optimal key pre-distribution scheme with enhanced security for wireless sensor networks." in *APNOMS*, 2006, pp. 546–549.

[7] S. J. Choi and H. Y. Youn, "An efficient key pre-distribution scheme for secure distributed sensor networks." in *EUC Workshops*, 2005, pp. 1088–1097.

[8] Y. Z. Bo Zhu, Yanfei Zheng and K. Chen, "Cryptanalysis of lu decomposition-based key pre-distribution scheme for wireless sensor networks," Cryptology ePrint Archive, Report 2008/411, 2008, http://eprint.iacr.org/.

[9] R. Blom, "An optimal class of symmetric key generation systems," in *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 335–338.

[10] M. B. Paterson and D. R. Stinson, "Two attacks on a sensor network key distribution scheme of cheng and agrawal," Cryptology ePrint Archive, Report 2008/326, 2008, http://eprint.iacr.org/.

[11] ALTERA Tech, "APEX EP20K400E Proccessor," http://www.altera.com/products/devices/apex/features/apx-true_lvds.html.

[12] D. Liu, P. Ning, and K. Sun, "Efficient self-healing group key distribution with revocation capability." in *ACM Conference on Computer and Communications Security*, 2003, pp. 231–240.

[13] A. Mohaisen, Y. Maeng, and D. Nyang, "On grid-based key pre-distribution: Toward a better connectivity in wireless sensor network," in *PAKDD Workshops*, 2007, pp. 527–537.

[14] 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.

[15] 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*, 2005, pp. 69–75.

[16] S. R. Blackburn, T. Etzion, K. M. Martin, and M. B. Paterson, "Efficient key predistribution for grid-based wireless sensor networks," in *ICITS*, ser. LNCS, R. Safavi-Naini, Ed., vol. 5155. Springer, 2008, pp. 54–69.

[17] Y. Cheng and D. P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 35–48, 2007.

[18] M. Burmester, "Cryptoanalysis of the chang-wu-chen key distribution system," in *EUROCRYPT*. Springer, June 1993, pp. 440–442.

[19] S.-M. Yen, "Cryptanalysis of an authentication and key distribution protocol," *IEEE Communication Letters*, vol. 3, no. 1, pp. 7–8, 1998.

[20] Q. Tang, "On the security of a group key agreement protocol," *Comput. J.*, vol. 50, no. 5, pp. 589–590, 2007.

Abedelaziz Mohaisen is Ph.D. student at the University of Minnesota Twin Cities. He was a member of 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 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.

DaeHun Nyang received the B.Eng. degree in electronic engineering from Korea Advanced Institute of Science and Technology, M.S. and Ph.D. degrees in computer science from Yonsei University, Korea on 1994, 1996, and 2000 respectively. He has been a senior member of engineering staff of Electronics and Telecommunications Research Institute, Korea from 2000 to 2003. Since 2003, he has been with the graduate school of Information and Telecommunication Engineering at Inha University, Incheon, Korea where he is currently an associate professor and the founding director of the Information Security Research Laboratory. He is also a consultant for Korean Information Security Agency, member of board of directors and editorial board of Korean Institute of Information Security and Cryptology. Dr. Nyang's research interests include cryptography and information security, privacy, biometrics and their applications to authentication, public key cryptography. Also, he is interested in the security of WLAN, RFID, WSN, and MANET.