Secure Data Aggregation in Wireless Sensor Networks: A Survey

Yingpeng Sang
School of Information Science
Japan Advanced Institute of Science and Technology
Asahidai, Tatsunokuchi, Ishikawa, Japan, 923-1211
{yingpeng}@jaist.ac.jp

Hong Shen
School of Computer Science
The University of Adelaide
SA 5005, Australia

Yasushi Inoguchi, Yasuo Tan, Naixue Xiong
Japan Advanced Institute of Science and Technology
{inoguchi, ytan, naixue}@jaist.ac.jp

Abstract

Data aggregation is a widely used technique in wireless sensor networks. The security issues, data confidentiality and integrity, in data aggregation become vital when the sensor network is deployed in a hostile environment. There has been many related work proposed to address these security issues. In this paper we survey these work and classify them into two cases: hop-by-hop encrypted data aggregation and end-to-end encrypted data aggregation. We also propose two general frameworks for the two cases respectively. The framework for end-to-end encrypted data aggregation has higher computation cost on the sensor nodes, but achieves stronger security, in comparison with the framework for hop-by-hop encrypted data aggregation.

1. Introduction

Wireless sensor networks (WSN) consist of a great deal of sensor nodes with limited power, computation, storage, sensing and communication capabilities. Sensors are becoming more and more inexpensive due to the advancement of the relevant technologies, so WSN will have broad applications in either controlled environments (such as home, office, warehouse, etc) or uncontrolled environments (such as hostile or disaster areas, toxic regions, etc). WSN can be looked as an event-based system with one "sink" subscribing to specific data streams by expressing interest and queries. The remaining sensors act as "sources" to report environmental events to the subscriber sink. To save energy, *Data aggregation* is put forward as an in-network processing which is conducted on the *aggregator* nodes ([13]). An aggregator can compute the sum, average, minimum or maximum of the data from its children sensors, and send the aggregation results to a higher-level aggregator. WSN can choose its aggregators dynamically according to their power remnant to optimize the total power consumption of the aggregation, which is outside the scope of this paper.

In this paper, we will consider the security issues in the data aggregation of WSN. Specifically, the fundamental security issue is *data confidentiality* ([17]), which protects the sensitive transmitted data from passive attacks, such as eavesdropping. Data confidentiality is especially vital in a hostile environment, where the wireless channel is vulnerable to eavesdropping. Though there are plenty of methods provided by cryptography, the complicated encryption and decryption operations, such as modular multiplications of large numbers in public key based cryptosystems, can use up the sensor's power quickly ([20]).

The other security issue is *data integrity*, which prevents the compromised source nodes or aggregator nodes from significantly altering the final aggregation value ([12]). Sensor nodes are easy to be compromised because they lack expensive tampering-resistant hardware, and even those tampering-resistant hardware might not always be reliable (as pointed in [1]). A compromised node can modify, forge or discard messages.

Generally, two methods can be used for secure data aggregation in WSN: *hop-by-hop* encrypted data aggregation and *end-to-end* encrypted data aggregation. In the former, data is encrypted by the sensing nodes and decrypted by the aggregator nodes. The aggregator nodes then aggregate the data and encrypt the aggregation result again. At last the sink node gets the final encrypted aggregation result and decrypt it. In the latter, the intermediate aggregator nodes haven't decryption keys and can only do aggregations on the encrypted data.

Our Contributions: Our contributions in this paper include the following:

Proceedings of the Seventh International Conference on
Parallel and Distributed Computing,Applications and Technologies (PDCAT'06)
0-7695-2736-1/06 $20.00 © 2006 IEEE

1) We respectively survey the work for hop-by-hop and end-to-end encrypted data aggregation in WSN. There has been some survey work for key distribution schemes in WSN, e.g., [4], but our view on these schemes is their utilities for data aggregation. What's more, we also survey the integrity protection work for WSN.

2) We propose security frameworks respectively for hop-by-hop and end-to-end encrypted data aggregation in WSN. The previous work merely emphasized either protecting confidentiality or protecting integrity, but our frameworks systematically address both confidentiality and integrity issues.

The remainder of the paper is organized as follows: Section 2 models the network and attacks, defines the security goals and aggregation functions. Section 3 surveys the related work for hop-by-hop encrypted data aggregation in WSN. Section 4 surveys the related work for end-to-end encrypted data aggregation in WSN. Section 5 proposes and analyzes the security frameworks respectively for the two types of encrypted data aggregation. Section 6 concludes the whole paper.

2. Background

Network Model We consider a similar model with [19] in which the nodes in the WSN can be divided into four sets $\mathcal{S}, \mathcal{A}, \mathcal{F}$ and $\mathcal{R}$: 1) $\mathcal{S}$ is the set of sensing nodes, which sense their environment; 2) $\mathcal{A}$ is the set of aggregator nodes, which combine the sensing values from $\mathcal{S}$ by aggregation functions; 3) $\mathcal{F}$ is the set of forwarders, which transfer the aggregation results from $\mathcal{A}$ towards $\mathcal{R}$ hop-by-hop; 4) $\mathcal{R}$ is the set of readers of the WSN, which may be base stations, or merely the sinks which provide an access to the outside for the WSN. It should be pointed out that $\mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}$ may change over time and their intersections may not be ϕ.

Our network model can represent both the Hierarchical WSN (HWSN) and Distributed WSN (DWSN). In HWSN, nodes are deployed hierarchically according to their capabilities. The whole network is composed of base stations ($\in \mathcal{R}$), cluster heads ($\in \mathcal{A} \cup \mathcal{F}$) and sensor nodes ($\in \mathcal{S}$). In DWSN, nodes are deployed randomly in the environment. After nodes are deployed, a transmission structure should be constructed to aggregate data. For example, in [24] a minimum spanning tree (MST) is constructed to gather data with minimum energy cost in WSN. In the MST, the root node (sink) is in the reader set $\mathcal{R}$, every node in the WSN is in $\mathcal{S}$, every non-leaf node is in the aggregator set $\mathcal{A}$ and the forwarder set $\mathcal{F}$. The non-leaf nodes aggregate the data they received with their own sensing data.

Attack model We assume there is only one adversary in the WSN, it is a polynomial-time bounded probabilistic Turing machine, it can physically access the sensors and read their internal values. The adversary is also assumed to be restricted in one region, so it can only compromise a small number of sensors.

Security requirements In the data aggregation of WSN, two security requirements, *confidentiality* and *integrity*, should be fulfilled. An adversary can breach the data confidentiality by the following attacks: 1) eavesdropping the messages in the wireless channel; 2) compromising a node and obtaining all keys stored in it; 3) using the compromised node's keys to deduce the keys employed elsewhere in the network; 4) using the compromised node's keys to inject unauthorized malicious sensor nodes in the network.

The adversary can also spoil the data integrity by the following attacks: 1)injecting arbitrary chosen malicious data into the compromised sensing nodes in the set $\mathcal{S}$; 2)modifying, forging, or discarding messages in the compromised aggregator nodes in $\mathcal{A}$ and compromised forwarder nodes in $\mathcal{F}$.

Aggregation functions Given the sensing data s_i from the sensing node S_i in $\mathcal{S}$ for $i = 1, ..., n$, the following aggregation function $f(s_1, ..., s_n)$ can be calculated in the WSN: 1) the *Sum*: $f(s_1, ..., s_n) = \sum_{i=1}^{n} s_i$. 2) the *Average*:$f(s_1, ..., s_n) = \sum_{i=1}^{n} s_i/n$. 3) the *Median*: $f(s_1, ..., s_n) = s_{(r)}$, $r = (n+1)/2$, $s_{(1)}, ..., s_{(n)}$ is an sorted order of $s_1, ..., s_n$. 4) the *Minimum*: $f(s_1, ..., s_n) = min\{s_i | i = 1, ..., n\}$. 5) the *Maximum*: $f(s_1, ..., s_n) = max\{s_i | i = 1, ..., n\}$. 6) the *Count*: $f(s_1, ..., s_n) = |\{s_i | i = 1, ..., n\}|$.

3. Hop-by-hop Encrypted Data Aggregation in WSN

The general idea of hop-by-hop encrypted data aggregation in WSN is : 1) bootstrapping secure links among the nodes; 2) aggregating data inside the network; 3) authenticating the integrity of aggregation results.

3.1. Security Bootstrapping

The *bootstrapping* problem ([7]) is to establish a secure communication infrastructure from a collection of sensor nodes which may have been initialized with some secret information but have had no prior direct contact with each other. The bootstrapping of hop-by-hop encryption can be realized by two methods: 1) pair-wise key distribution among each pair of sensor nodes; 2) group-wise key distribution among a cluster of sensor nodes. In DWSN data confidentiality in aggregation can be protected by pair-wise key distribution schemes. In HWSN data confidentiality in aggregation can be protected by group-wise key distribution schemes.

Proceedings of the Seventh International Conference on
Parallel and Distributed Computing,Applications and Technologies (PDCAT'06)
0-7695-2736-1/06 $20.00 © 2006 IEEE

Pair − wise Key Distribution Schemes The common way for pair-wise key distribution is key pre-distribution, i.e., keys are stored in sensors before sensors are deployed. After the deployment, each sensor establishes a secret link with its neighbour using a common pair-wise key which has been stored in it. Key connectivity, the probability of one sensor node finds a common key with its neighbour, is an important factor to be considered in the pair-wise key distribution schemes.

Master key based solution: A simple solution is to store a *master* key in all the sensor nodes ([14]). After they are deployed, each pair of sensor nodes uses this master key to achieve a new pair-wise key. This scheme has low resilience because the compromising of one node will lead to the compromising of the whole network.

Pair-wise key pre-distribution solution: There is another straightforward solution in which each sensor node stores $N-1$ secret pair-wise keys, each of them is known only to this sensor node and one of the other $N-1$ sensor nodes. This solution has good resilience but is impractical because a sensor node has limited storage and the size of the network (N) could be very large. What's more, this solution isn't scalable to accept new nodes after the deployment of the network because the deployed nodes may haven't the keys of the new node.

Random key pre-distribution solutions: A basic random key pre-distribution scheme is proposed in [10]: in the *key-predistribution phase*, each sensor node receives a random subset of k keys from a large key pool of K keys. In the *shared-key discovery phase*, to agree on a key for communication, two nodes find one common key within their subsets and use this key as their shared secret key. The probability of key share among two sensor nodes is $\frac{((K-k)!)^2}{(K-2k)!K!}$. In the *path-key establishment phase*, any pair of nodes (i,j) can securely establish a pair-wise key $K_{i,j}$ through a path $i, v_1, ..., v_n, j$, ordinally by sending $E_{K_{i,v_1}}(K_{i,j}), E_{K_{v_1,v_2}}(K_{i,j})..., E_{K_{v_n,j}}(K_{i,j})$.

This scheme is improved in [7]: a random set of $(N-1)p$ $(0<p<1)$ pair-wise keys is stored in each sensor node. The key connectivity becomes p because with probability p two nodes can be connected. The memory required for storing keys is decreased and good resilience is kept.

Key pre-distribution schemes with deployment knowledge: A location-based scheme is proposed in [15] to improve the work in [10]: it assumes that each sensor node has an expected location that can be predicted. Then each sensor is preloaded with the pair-wise keys of its c closest neighbours. This solution has low memory usage but good connectivity.

Another work in [8] divides sensor nodes into $t \times n$ groups, and deploys sensors in each group by Gaussian distribution. Compared with [10], key connectivity is improved while keeping good resilience.

Other solutions: There are also a few key pre-distribution schemes based on other techniques. The scheme in [5] is based on block design in combinatorial design theory. In [9], each pair of nodes can calculate corresponding field of the key matrix and use it as the pair-wise key. The scheme in [16] uses the evaluation of symmetric polynomial $P(x,y)$ $(P(x,y)=P(y,x))$ at the ID of each nodes pair (i,j) to get a pair-wise key $K_{i,j}=P(i,j)$.

Group − wise Key Distribution Schemes Group-wise Key Distribution Schemes are mainly used for HWSN. There are two types of distributions:

Symmetric group-wise key distribution: In [2], a symmetric key can be generated among t nodes by evaluating a symmetric multivariate polynomial $P(x_1, ..., x_t)$ at each node.

Asymmetric group-wise key distribution: In [18], the memory of each sensor node is pre-loaded with the ECC (elliptic curve cryptography) domain parameters. After deployment, each sensor will compute its EC-public/private key pair and broadcast its public key to all nodes within the cluster. According to their comparisons, the computation complexity of ECC is lower than DSA/RSA cryptosystem, but higher than the symmetric cryptosystem.

3.2. Data Integrity

A few related work assumes that hop-by-hop data confidentiality has been protected by some key distribution schemes, and proposes independent schemes from those schemes for data confidentiality to protect data integrity.

In [12], the data integrity protection scheme assumes that each node (e.g., node A) is initialized before deployment with a symmetric pair-wise key, e.g., K_{AS}, shared with the base station S. A secure self-organizing protocol is also assumed to be used to form a routing hierarchy where each node has an immediate parent. In the i-th *data transmission* phase, a leaf node A computes a temporary key $K^i_{AS}(=E(K_{AS},i))$ based on K_{AS}, sends its data reading R_A, node id ID_A and message authentication code $MAC(K^i_{AS}, R_A)$ on R_A to its parent. The parent node B calculates the aggregation of its children nodes readings, sends the result $Aggr$, node id ID_B and message authentication code $MAC(K^i_{BS}, Aggr)$ on $Aggr$ to its parent. The final aggregation and its MAC is sent to the base station. In the *data validation* phase, the base station verifies the final aggregation, and broadcasts the temporary keys (K^i_{AS}, K^i_{BS}, ...). Using these pair-wise keys, the intermediate aggregation results can be verified by the intermediate aggregators. This scheme has low communication cost because the data readings of each sensor node isn't needed to be transmitted to the base station, but is vulnerable because the intermediate aggregation is easy to tamper if a parent and a child node in their hierarchy are compromised.

The vulnerability of [12] is improved in [18]. The integrity of sensor readings is ensured with the help of a Merkle Hash Tree. In the data transmission phase, the sensors transmit the encrypted value of their readings along with its hash to the cluster-head, and the cluster-head builds a Merkle Hash Tree based on the hash values of the readings. In the data validation phase, the base station queries to the cluster-head on the individual readings. The drawback of this scheme is its high communication cost on data validation.

The work in [21] proposes an efficient way to verify the data integrity. After the aggregator sends the aggregation results and the commitment of the sensor readings based on Merkel Hash Tree, the base station needn't to repetitively query the aggregator of the sensor readings. It can engage an interactive proof with the aggregator and checks whether the aggregation result is correct. The communication cost is lower than [18] because the interactive proof achieves sublinear communication complexity.

4. End-to-end Encrypted Data Aggregation in WSN

Hop-by-hop encrypted data aggregation leaves aggregator nodes vulnerable to attacks because the sensor readings will be decrypted on those aggregators. End-to-end encrypted data aggregation is an alternative to address this vulnerability issue. It provides end-to-end privacy between sensor nodes and the sink. The aggregators aggregate the encrypted sensor readings without decrypting them, so the end-to-end privacy should be realized by homomorphic cryptosystems.

4.1. Network-wise Key Distribution

End-to-end privacy needs to establish a network-wise key between the sink and all the sensor nodes. Network-wise key distribution schemes include the master key based and public key based solution.

Master key based solution: In [6] and [11], it's assumed that the sensor nodes share a common secret key K with the sink, but the aggregator nodes haven't this key. Modular addition and Domingo-Ferrer's scheme are used respectively by them to encrypt data, and both of them are additive homomorphic. Sensor nodes S_i $(1 \leq i \leq n)$ sends their encrypted readings $E_K(R_i)$ to the aggregator node. The latter calculates the encrypted aggregation $E_K(f(R_1, ..., R_n))$ based on $E_K(R_i)$ and sends it to the sink. The sink decrypts $E_K(f(R_1, ..., R_n))$ and gets the aggregation result. The limitation is that the whole network will be compromised if K on one sensor node is compromised.

Public key based solution: In [19], each sensor node uses the public key of the base station to encrypt its reading employing some homomorphic public key encryption schemes. The base station is assumed to have strong reliability so that it's not easy to be compromised. The public key encryption schemes are constructed on elliptic curves in [19], but computation requirement in encryption is still high for the sensor nodes.

4.2. Data Integrity

Compared with hop-by-hop encrypted data aggregation, there isn't any more efficient way proposed to protect data integrity in end-to-end encrypted data aggregation. In [23], it is assumed that there isn't any aggregator node inside the WSN. Each sensor node sends its reading to the sink using end-to-end encryption. The sink employs truncation and trimming on the readings to achieve robust aggregation result against spoofed sensor readings. But when the network size is very large, the communication cost will be very high for the transmission of all sensor readings to the sink.

5. Two Frameworks for Data Aggregation in WSN

As pointed out in Section 3, security requirements in data aggregation include data confidentiality and integrity, but by the survey in Section 3 and 4, related work focuses only on either one of the two requirements. In this section we proposes two frameworks respectively for hop-by-hop and end-to-end encrypted data aggregation in WSN, aiming at systematically tackling the attacks on both data confidentiality and integrity.

5.1. Framework 1: Hop-by-hop Encrypted Data Aggregation

1) *the bootstrapping phase:* For controlled environment, HWSN can be constructed and a group-wise key can be generated for all nodes within each cluster. For uncontrolled environment, DWSN can be constructed and pair-wise keys can be distributed among each pair of sensor node.

2) *the aggregator selection phase:* The sink or base station can select aggregators to construct a transmission structure with minimum energy cost (e.g., using the technique in [24]). The transmission structure is composed of $\mathcal{S}$, $\mathcal{A}$, $\mathcal{F}$, $\mathcal{R}$ as defined in the network model of section 2.

3) *the data aggregation phase:* Suppose n is the number of the children of an aggregator A, the children nodes S_i $(1 \leq i \leq n)$ encrypt their readings x_i as $E_{K_{S_i,A}}(x_i)$, and sends it to A. $K_{S_i,A}$ is the pair-wise

Proceedings of the Seventh International Conference on
Parallel and Distributed Computing,Applications and Technologies (PDCAT'06)
0-7695-2736-1/06 $20.00 © 2006 IEEE

key between A and S_i. A decrypts $E_{K_{S_i,A}}(x_i)$ and calculates $f(x_1, ..., x_n)$. f is the aggregation function.

4) *the data transmission phase:* Each aggregator encrypts its aggregation result and sends it to the upper level aggregator. The upper level aggregator decrypts the aggregation results and aggregate them as a new aggregation results. Finally, the sink gets the aggregation result of the whole network. The lowest level aggregators, i.e., the fathers of the sensing nodes, also commit the encrypted reading of their children $E_{K_{S_i,R}}(x_i)$ by the Merkle Hash Tree. $K_{S_i,R}$ is the pair-wise key between S_i and the sink.

5) *the data integrity verification phase:* The sink hashes all $E_{K_{S_i,R}}(x_i)$ again to check whether the commitment is right. If it's right, the sink decrypts all $E_{K_{S_i,R}}(x_i)$ and calculate the aggregation again to see whether the aggregation result from the aggregators is right.

5.2. Framework 2: End-to-end Encrypted Data Aggregation

1) *the bootstrapping phase* and *the aggregator selection phase* are the same with Framework 1, except that in the end of the bootstrapping phase the sink broadcasts a network-wise public key K.

2) *the data aggregation phase:* S_i sends the encrypted reading $E_K(x_i)$ to its aggregator A. A calculates the aggregation result $E_K(f(x_1, ..., x_n))$ based on $E_K(x_i)$ for $1 \leq i \leq n$. E is an asymmetric additive homomorphic encryption scheme such as ECC-ElGamal.

3) *the data transmission phase:* The readings are aggregated level by level on the aggregators until the sink gets the final aggregation result $E_K(f)$. The lowest level aggregators also commit all $E'_{K_{S_i,R}}(x_i)$ of its children by the Merkle Hash Tree, to the sink. E' is a symmetric encryption scheme such as AES. $K_{S_i,R}$ is the pair-wise key between S_i and the sink.

4) *the data integrity verification phase:* The sink checks whether the commitment is the hash of all $E'_{K_{S_i,R}}(K)$. If it's right, the sink decrypts all $E'_{K_{S_i,R}}(x_i)$ to check whether the final aggregation result is right.

5.3. Security Analysis

In both frameworks, data confidentiality can be protected by the keys established in the bootstrapping phase. For pair-wise keys and group-wise keys, the compromising of a small number of nodes won't lead to the compromising of the whole network because they are only partially used. The network-wise key in end-to-end encryption is safe because it's a public key.

With only some aggregators compromised, the modifying and forging on the aggregation result will be detected by the sink, because the sink can decrypt the sensor readings and aggregate them again. But when some sensing nodes are compromised, the aggregation result is easy to be tampered without being detected. Therefore, how to make the aggregation result more resilient in the two frameworks is still a challenging problem.

In Framework 2, public key encryption scheme is used, so it's less efficient than Framework 1. However, in Framework 1 sensor readings are decrypted before they are aggregated, so the compromising of an aggregator will make the adversary easy to read the sensor readings and aggregation result.

6. Concluding Remarks

We survey the related work for secure data aggregation in WSN and classify them into two general cases: hop-by-hop and end-to-end encrypted data aggregation. For every case, we summarize the proposed techniques for protecting data confidentiality and data integrity, and discuss their superiorities and limitations.

We also propose two systematic frameworks for the two case. The framework for hop-by-hop encrypted data aggregation is more efficient than the framework for the end-to-end one, but in the former one the sensor readings may be leaked to the adversary if the aggregator is compromised. In the future we will work for an efficient framework while keeping high resilience and security.

7. Acknowledgment

This research is conducted as a program for the "Fostering Talent in Emergent Research Fields" in Special Coordination Funds for Promoting Science and Technology by Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] R. Anderson, M. Kuhn, "Tamper Resistance - a Cautionary Note", *proceedings of the Second Usenix Workshop on Electronic Commerce*, pp. 1-11, November 1996.

[2] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences", in *Crypto 92*, 1992.

[3] H. Cam, S. Ozdemir, P. Nair, D. Muthuavinashiappan, H. O. Sanli, "Energy-Efficient Secure Pattern Based Data Aggregation for Wireless Sensor Networks", *Computer Communications*, Vol. 29, No. 1, Elsevier, Dec. 2005.

[4] S. A. Camtepe and B. Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: a Survey", Rensselaer Polytechnic Institute, technical report TR-05-07, March 2005.

[5] S. A. Camtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks", in *9th European Symposium on Research Computer Security*, 2004.

[6] C. Castelluccia, E. Mykletun and G. Tsudik, "Efficient Aggregation of Encrypted Data in Wireless Sensor Networks", *ACM/IEEE Mobiquitous Conference*, July 2005, San Diego, USA.

[7] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks", in *Proc. of the IEEE Security and Privacy Symposim 2003*, 2003.

[8] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge", in *IEEE Infocom04*, 2004.

[9] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks", in *Proceedings of the 10th ACM conference on Computer and Communications Security CCS03*, 2003.

[10] L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks", In *ACM CCS 2002*, Washington DC, 2002.

[11] J. Girao, D. Westhoff, and M. Schneider, "CDA: Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks", *40th International Conference on Communications, IEEE ICC 2005*, May 2005, Korea.

[12] L. Hu and D. Evans, "Secure aggregation for wireless networks", In *Workshop on Security and Assurance in Ad hoc Networks*, Jan 2003.

[13] B. Krishnamachari, D. Estrin, S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks", in *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS)*, Pages 575 - 578, 2002.

[14] B. Lai, S. Kim, and I. Verbauwhede, "Scalable session key construction protocol for wireless sensor networks", In *IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES)*, Austin, Texas, December 2002.

[15] D. Liu, and P. Ning, "Location-based pairwise key establishment for static sensor networks", in *1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2003.

[16] D. Liu, and P. Ning, "Establishing pairwise keys in distributed sensor networks", in *10th ACM conference on Computer and communications security CCS03*, 2003.

[17] W. Lou, W. Liu, Y. Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks", *IEEE INFOCOM 2004*, 2004.

[18] A. Mahimkar, T. S. Rappaport, "SecureDAV: A Secure Data Aggregation and Verification Protocol for Sensor Networks", *Proceedings of IEEE Global Telecommunications Conference (Globecom) 2004*, Nov, 2004, Dallas, TX, USA.

[19] E. Mykletun, J. Girao, and D. Westhoff, "Public key based cryptoschemes for data concealment in wireless sensor networks", in *IEEE International Conference on Communications (ICC2006)*, June 2006, Turkey.

[20] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar, "Spins: Security protocols for sensor networks", in *Proceedings of the 7th Annual ACM/IEEE Internation Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, July 2001, pp. 189C199.

[21] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks", In *Proc. of ACM SenSys 2003*, 2003.

[22] H. O. Sanli, S. Ozdemir, and H. Cam, "SRDA: Secure Reference-Based Data Aggregation Protocol for Wireless Sensor Networks", *Proc. of IEEE VTC Fall 2004 Conference*, Sept. 2004, Los Angeles, CA, USA.

[23] D. Wagner, "Resilient aggregation in sensor networks", in *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 78-87. ACM Press, 2004.

[24] K. Yuen, B. Li, B. Liang, "Distributed Minimum Energy Data Gathering and Aggregation in Sensor Networks", in *IEEE International Conference on Communications (ICC 2006)*, June 2006.