Journal of Information Hiding and Multimedia Signal Processing ©2013 ISSN 2073-4212
Ubiquitous International Volume **4**, Number **1**, January **2013**

SASHIMI: Secure Aggregation via Successively Hierarchical Inspecting of Message Integrity on WSN

Chien-Ming Chen

Innovative Information Industry Research Center
Shenzhen Graduate School
Harbin Institute of Technology, Shenzhen, China
chienming.taiwan@gmail.com

Yue-Hsun Lin

CyLab
Carnegie Mellon University
Pittsburg, USA
tenma.lin@gmail.com

Yao-Hsin Chen

Information and Communications Research Laboratories
ITRI
Hisnchu, Taiwan, R.O.C.
yaohsin@itri.org.tw

Hung-Min Sun

Department of Computer Science
National Tsing Hua University
Hisnchu, Taiwan, R.O.C.
hmsun@cs.nthu.edu.tw

Received October 2012; revised November 2012

Abstract. *Aggregation schemes for reducing transmission cost have been proposed for wireless sensor networks for a long time. Aggregated results can be easily altered by adversaries since sensors are prone to being captured in a harsh environment. Hence, several secure data aggregation schemes have been proposed to solve this problem. Many schemes ensure data integrity during aggregation procedures, but most of them are post-active since integrity can only be confirmed after the data reaches the base station. Another limitation is that the network topology is assumed to be fixed. However, this assumption violates the characteristic of sensor networks. In this paper, we present a secure data aggregation scheme called SASHIMI. SASHIMI utilizes successively hierarchical inspecting of message integrity during aggregation. If attacks arise during aggregation, attacks can be detected within two levels of the hierarchal tree structure. In other words, penalty and overhead caused by attacks can be reduced. In average, SASHIMI incurs only $O(n)$ communication cost where n is the number of nodes. In the case of attacks, SASHIMI performs better than existing schemes. Moreover, SASHIMI supports dynamic network topology. Finally, a comprehensive analysis demonstrates that SASHIMI is more secure and efficient than other schemes.*

keyword: Wireless Sensor Network; Data aggregation; Data integrity

1. **Introduction.** Wireless sensor networks (WSN) are often put to use in hostile or outdoor environments, such as mountains, battlefields, or underwater environments. Typically, a WSN contains a large number of sensors which communicate with each others and a base station. After these sensors are deployed, they are responsible for collecting, processing and transmitting data. Once data arrives at the base station, it can be used for different purposes base on its different applications. Since sensors have limited energy, i.e., battery supply, it is inefficient for all sensors to transmit data to the base station individually. Therefore, data aggregation for WSN is a hot research topic.

Data aggregation is an efficient strategy to reduce the number of messages queried and returned by sensors. When the base station queries statistics, e.g., average of data values, each internal sensor sends an aggregated result instead of all readings. For instance, the administrator desires to know the sum (summation) of temperature values gathered by all deployed sensors. The sum function is performed by having each internal sensor forward a sum value presenting the sum of all received readings and its own data. Through sum aggregation, we can that guarantee the number of messages sent to the base station is minimal. Efficiency of data aggregation is significant for WSN, shown in previous research [1],[2].

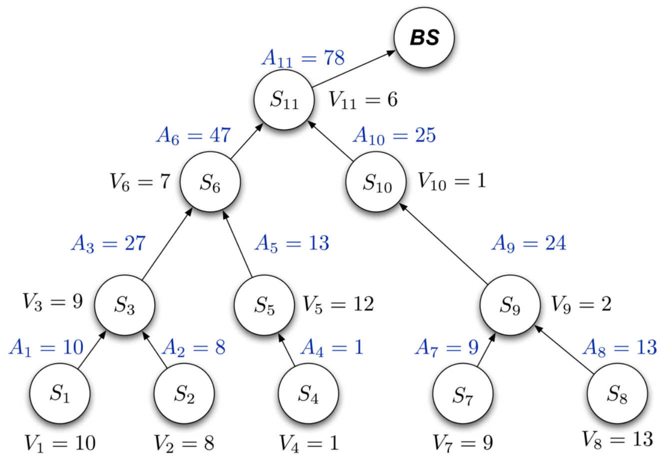


FIGURE 1. An aggregation scenario using sum function.

Fig. 1 is a simple example that explains how data aggregation works. Note that A_i denotes the aggregated result generated by sensor S_i , and V_i denotes the sensing reading of S_i. For example, S_3 performs aggregation on received values A_1, A_2, and its own reading V_3. The sum of these three values is 27. Instead of sending A_1, A_2 and V_3, S_3 calculates the sum and sends aggregated value A_3 to its parent.

Unfortunately, data aggregation in WSN faces a critical security threat. This is because sensors are usually deployed in unsafe environments that lack security mechanism due to cost consideration. An adversary can easily capture deployed sensor and take full control of the captured sensors to launch any types of attacks he wants via re-programming and compromising information in sensor storage. Therefore, we should design a secure aggregation scheme to identify malicious nodes while still considering the hardware constraint of sensors.

Several researcher have investigated secure aggregation schemes to ensure data integrity. Most of these schemes place emphasis on data integrity. More precisely, these schemes ensure the base station can obtain the correct aggregated result. In 2006, Chan *et al.* proposed a secure protocol for provably secure hierarchical in-network data aggregation [3].

Chan *et al.*'s scheme can provide integrity of aggregation against multiple malicious nodes. Unfortunately, Chan *et al.*'s solution has two drawbacks. First, their scheme only works under one assumption that all sensors must know the entire topology. If the topology is changed, Chan *et al.*'s scheme may malfunction. Second, their scheme has inefficient result-checking phase. In this phase, high overhead is required for all leaf and internal nodes. If data query occurs often, energy consumption is unaffordable for deployed sensors.

In this paper, we propose a data aggregation scheme with the necessary security properties for WSN. The proposed scheme is called SASHIMI, which is based on successively hierarchical inspecting of message integrity during aggregation. The basic idea of SASHIMI is that result checking and aggregation are performed concurrently. Result-checking is executed within two levels hierarchy, not only at the base station. Once attacks are detected, sensors would reply error reports to the base station via multiple pre-installed routing paths. Penaltis can thus be deduced in a better manner. Moreover, unlike Chan *et al.*'s scheme [3], SASHIMI does not require strong assumptions, e.g., fixed network topology, topology knowledge. While maintaining the same secure properties, SASHIMI is more efficient than prior schemes.

This paper is organized as follows. Section 3 describes the necessary preliminaries for understanding SASHIMI. Section 4 depicts the detail of SASHIMI for general WSN. In section 5, the performance and security of SASHIMI are given. To prove our concept, experiments are also conducted in section 6. Finally, we conclude SASHIMI in section 7.

2. **Related works.** In this section, we survey several data aggregation schemes proposed for different network topologies, i.e., cluster-based WSN, chain-based WSN, and tree-based WSN.

In cluster-based WSN, deployed sensors are divided into several clusters. Sensors within a cluster transmit their sensing reading to a specific node, the cluster head. Aggregation procedure is involved in the cluster head. Several schemes [4, 5, 6, 7, 8] have been proposed for cluster-based WSN. In view of chain-based WSN, the key idea is for each sensor to transmit only to its closest neighbor. The following schemes [9, 10] have been described. In tree-based WSN, sensors are organized as a tree. Data aggregation is performed at intermediate sensors. The final aggregation result is transmitted to the root sensor. Several schemes [3, 11] have been introduced.

Since an adversary may attempt to alter the aggregation result, researchers have placed emphasis on *secure* data aggregation schemes [12, 13, 14, 15, 16, 17]. In 2003, Hu and Evans proposed a secure aggregation scheme [12]. However, their scheme only provides protection on data aggregation against a single malicious node. Later, Jadia and Mathuria [13] proposed Efficient Secure Aggregation scheme (ESA) to enhance the security of [12]. But ESA becomes insecure when two consecutive sensors in the hierarchy are compromised. In 2006, Chan *et al.* [3] proposed a secure aggregation scheme for tree-based WSN. This scheme offers data integrity and authentication. Once the base station receives the final aggregation result, it broadcasts the result to all sensors. Each sensor is responsible for checking whether its sensing reading was correctly added to the final aggregation result. However, this approach still has several drawbacks. This scheme will be described in section 3.3 for further discussion. In 2008, Chan *et al.* proposed several enhancements [18] based on their previous scheme [3]. They support additional functions, authenticated broadcast and node-to-node message signatures for tree-based WSN based on their previous approach.

3. **Preliminaries.** In this section, we describe same background knowledge before introducing SASHIMI. We first introduce the network model of tree-based WSNs. Also, we define the attack models for WSN data aggregation schemes. Finally, we review a well-known scheme called SHIA (Secure Hierarchical in-network Aggregation) [3].

3.1. **Network Model.** The proposed scheme is designed for tree-based WSNs. Basically, a WSN is controlled by the base station (BS). Through wireless communication, the BS commands all deployed sensor nodes (SNs) to execute specific tasks. The BS is capable of high bandwidth, strong computing, sufficient memory and stable power supply. Expensive operations, such as cryptographic or routing procedures, are affordable for the BS. Compared with the BS, cheaper hardware limits a SN's computation, communication and storage capability.

After all SNs are deployed, the BS broadcasts query messages to all SNs. Once SNs receive messages, they begin to construct a query tree (also called an aggregation tree). With this query tree, each SN will have an unique path to the BS. Algorithms for constructing query trees have been described [2, 11, 19, 20]. For example, in *Tiny Aggregation Service* (TaG) [2], the BS broadcasts tree formation to all SNs. Each SN selects one SN which first send the tree formation to itself as its parent node. After all SN form constructing links, a tree-based WSN is constructed.

A typical query tree structure is depicted in Fig. 1. SNs are deployed as a tree network where the root node is the BS. For SN_1 and SN_2, SN_3 is the parent node of these two SNs. For SN_4, SN_6 is the grandfather node, SN_3 and SN_5 are child nodes of SN_6. And SN_3 and SN_5 are called siblings since they have the same parent, SN_6.

To reduce transmission overhead, sensing data should be aggregated before being sent to the BS. Each internal SN aggregates its sensing data with data from its child nodes. The BS would eventually receive the final aggregation result rather than the sensing data from each SN.

3.2. **Attack model.** Attacks occur when SNs transmit their data to the BS. Adversaries may cause data jamming or eavesdropping on wireless channels. Adversaries may also capture SNs to obtain all the secret stored in the SNs. For data aggregation schemes, we list the possible attacks as follows:

1. Maliciously altering aggregation results: If an adversary compromises an internal SN, she can maliciously alter the aggregation results maliciously.
2. Maliciously altering sensing data: The adversary can directly alter the sensing data of compromised SNs. Moreover, we assume that an adversary may compromise an arbitrary amount of SNs.

3.3. **SHIA Review.** In 2006, Chan *et al.* proposed a *Secure Hierarchical In-network Aggregation* scheme on WSN [3]. For short, we called it SHIA. SHIA [3] can perform several algebraic aggregation algorithms such as sum or average on a tree-based WSN. Since SHIA is the fundamental scheme we try to enhance, the scheme is described in detail in the following paragraphs. An example adopting SHIA aggregation is depicted in Fig. 2. SHIA has three phases.

Label generation phase: In Fig. 2(a), the base station R broadcasts a query request with attached nonce N through an authenticated broadcast channel. Note that N is used to prevent message replays. After receiving the query request, each SN generates its *Label* in a particular format. For example, node I generates $I_0 = \{1, a_I, r - a_I, I\}$. The first entry denotes the number of nodes in the subtree rooted at I. The second entry a_I is the sensing data of I. $r - a_I$ is the complement value of a_I where r is the maximum

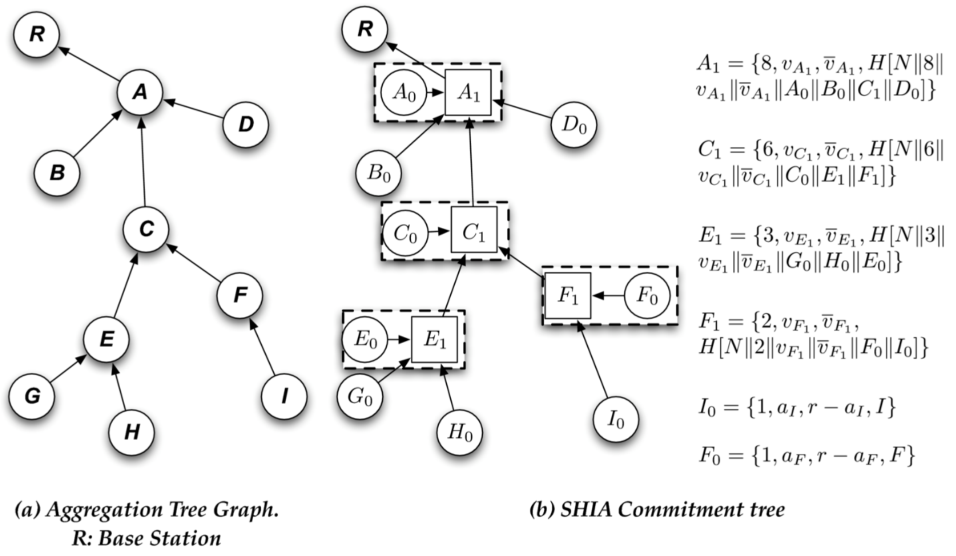


FIGURE 2. An example of SHIA Scheme

bound of the sensing data. The final term is the node identity, I. Similarly, F_0, which is the *Label* of node F, equals $\{1, a_F, r - a_F, F\}$.

Aggregation phase: This phase begins from bottom to top (leaf nodes to R). Each leaf SN transmits its *Label* to its parent SN. As shown in Fig. 2(b), I transmits I_0 to F. Then, F aggregates I_0 and F_0 as F_1. Note that F_1 equals $\{2, v_{F_1}, \overline{v}_{F_1}, H[N\|2\|v_{F_1}\|\overline{v}_{F_1}\|F_0\|I_0]\}$. The first term is 2 since it presents the total count of I_0 and F_0. v_{F_1} is the aggregated value of a_I and a_F, i.e., $v_{F_1} = a_I + a_F$. Similarly, $\overline{v}_{F_1}$ is the sum of complements, i.e., $\overline{v}_{F_1} = (r - a_I) + (r - a_F)$. The final term is the hashed of N, 2, v_{F_1}, $\overline{v}_{F_1}$, F_0 and I_0. F then sends F_1 to C. Similarly, E aggregates E_0, G_0 and H_0 as E_1 and sends E_1 to C. Thus, C can aggregate E_1, F_1 and C_0 as C_1. Eventually, R would receive the final aggregated result A_1.

Result-checking phase: After receiving A_1, R starts the result-checking phase. In this phase, each SN is responsible for verifying whether or not its sensing reading was actually added to the aggregation result. For example, node I would check the integrity of F_1, C_1 and A_1 since F, C and A are ancestors of I. Initially, R distributes A_1 to all SNs for verification using authenticated broadcast. Each SN would receive all its *off-path values*. Note that *off-path values* of node v are the set of *Label*s generated by all the siblings of each SN on the path from v to the root in the commitment tree. For example, the *off-path values* of node I, $Off(I)$, in Fig. 2(b) are $\{F_0, E_1, C_0, A_0, B_0, D_0\}$. After receiving them, I starts verification as follows:

1. Re-compute F_1' through buffered I_0 and F_0 where $F_0 \in Off(I)$
2. Re-compute C_1' through F_1' and E_1 where $E_1 \in Off(I)$
3. Re-compute A_1' through C_1', B_0, D_0 where $B_0, D_0 \in Off(I)$

After computing A_1', I compares A_1' and received A_1. If they are the same, I generates a successful commitment $MAC_{K_I^R}(N\|OK)$ where key K_I^R is the secret shared with I and the base station R. Otherwise, the aggregated data is assumed to have been altered or tampered with on the path from node I to the base station R. Hence, I generates a failure commitment $MAC_{K_I^R}(N \| FAIL)$ to R. Once R receives successful commitments from all SNs, the aggregation is identified as successful.

As described above, dispatching *off-path values* is necessary in SHIA for result-checking. Fig. 3 shows the dispatch process *off-path values*. Assume node T is the parent node of V and S, and T is the closet node to the BS. During commitment verification, T should distribute off-path sets V and S to V and S reversely. Once S receives the label of V, it will distribute it to U_1 and U_2. Then U_1 and U_2 will dispatch the label of V to their offsprings. Similarly, S should send the label of U_1 to U_2, the label of U_2 to U_1 for *off-path values* dispatch.

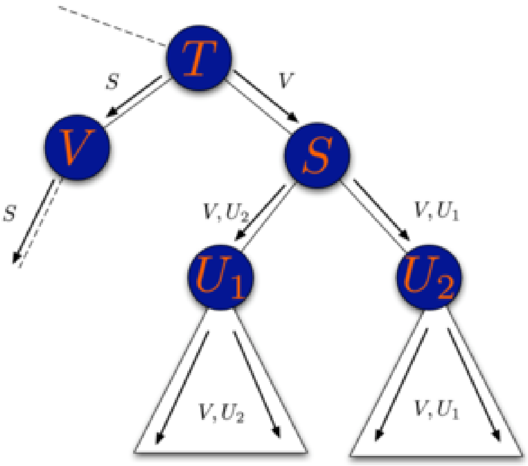


FIGURE 3. Off-path value dispatch

As we shown, SHIA [3] has the following drawbacks.

1. Each SN must realize the aggregation sequence when the result-checking phase begins. However, acquiring complete topology incurs additional overhead for deployed sensor nodes [21].
2. It does not support dynamic topology. Once the topology is changed, verification will fail.
3. Dispatching the *off path values* of each SN is a significant overhead.
4. Attacks, e.g., altering aggregation results, cannont be detected until aggregation is completed.

4. **The proposed scheme.** In this section, we propose SASHIMI to overcomes the drawbacks of SHIA. Notations used in this section are listed in Table 1.

The main idea of SASHIMI is to allow each SN to check the validity of the aggregation result generated by its parent node. After a SN confirms the validity of an aggregation result, it will report the checked result to its grandfather node. As a result, a grandfather node can verify whether or not the aggratation result from its child node is legal by checking all the results of its grandson nodes.

4.1. **Assumptions.**

1. The BS shares a unique key with each deployed SN. The key is kept secret.
2. Each SN shares a symmetric key with any SN that is wit two hops.

Note that assumption 2 can be achieved by random key distribution [22, 23, 24, 25]. Also, SNs are classified into three types in SASHIMI.

1. Leaf Node: SNs that are leaves of the query tree.

Table 1. Notations

Notation	*Description*
BS	Base station
Q	Query number
SN_i	Sensor node i
P_i	Parent node of SN_i
G_i	Grandfather node of SN_i
ID_i	Identity of SN_i
M_i	The message generated by SN_i
$K_{i,j}$	Pairwise key shared with SN_i and SN_j
Cl_i	Childlist of SN_i
H	Hash function
MAC_k	MAC function use key k
V_i	Sensing reading of SN_i
Agg_i	Aggregation result generated by SN_i
$aggregate$	Aggregation function, e.g., sum, min, max
$\hat{M}_i$	The broadcast messages from SN_i

2. L-Internal Node: SNs that have at least one child nodes and all children are Leaf Node. In other words, they are the last internal nodes.
3. O-Internal Node: SNs that belong to niether Leaf Node nor L-Internal Node. They are all the remain internal nodes.

In Fig. 1, SN_1, SN_2, SN_4, SN_7 and SN_8 belong to Leaf Node. SN_3, SN_5 and SN_9 belong to L-Internal Node, and SN_6, SN_{10}, and SN_{11} belong to O-Internal Node.

4.2. **Details of the Proposed Scheme.** In the beginning, the BS broadcasts authentic query messages to all deployed SNs. The query message contains the querying number Q_x. Aggregation procedure begins from bottom to top. Hence, the procedure begins at Leaf Node. We assume that SN_i belongs to Leaf Node and calculates and disseminates M_i to its parent SN_r as the following format.

$$M_i = < ID_i|ID_r|V_i|H(Q_x|V_i) >$$

After sending M_i, SN_i buffers M_i as Σ_i in its storage. Note that Σ_i will be used for checking the integrity of M_i.

Parent node SN_r has two possible roles, depedning on if it belong to L-Internal Node or O-Internal Node. We consider these two cases separately.

Belonging to L-Internal Node Assume that SN_r belongs to L-Internal Node and has k leaf child nodes, i.e., $SN_1, \cdots, SN_i, \cdots, SN_k$. The parent node of SN_r is SN_z. SN_r would execute the following actions.

1. Gather $\{M_1, \cdots, M_k\}$ from $SN_1, \cdots, SN_k$
2. Create aggregation message M_r such that:
   $M_r = < ID_r|ID_z|Agg_r|H(Q_x|Agg_r|M_1|\cdots|M_k) >$
   where $Agg_r = aggregate(V_r, V_1, \cdots, V_k)$
3. Create $\hat{M}_r = M_r|M_i \ \forall i \in \{1, \cdots, k\}|V_r$
4. Broadcast $\hat{M}_r$ to nearby SNs, i.e., $SN_z, SN_1, \cdots, SN_k$

When SN_r's child nodes $SN_1, \ldots, SN_k$ obtain $\hat{M}_r$, they can check to see if their sensing reading was actually added to $\hat{M}_r$. For example, SN_i executes the following steps.

1. Obtain $\hat{M}_r$ from its parent node SN_r
2. Verifie Agg_r by re-computing the aggregated result, i.e., recomputes $Agg_r = aggregate(V_r, V_1, \cdots, V_k)$, where $V_r \in M_r$, $V_j \in M_j$, $M_r, M_j \in \hat{M}_r$, $1 \leq j \leq k$
3. Compute $H' = H(Q_x|Agg_r|M_1|\cdots|\Sigma_i|\cdots|M_k)$ where M_i is replaced by Σ_i, $Q_x, Agg_r, M_j \in \hat{M}_r\ \forall j \neq i$
4. Verifie M_r by comparing H' with $H(Q_x|Agg_r|M_1|\cdots|M_k) \in \hat{M}_r$ If it succeeds, SN_i broadcasts the successful message SRM_i, where $SRM_i = ID_i|ID_z|MAC_{K_{i,z}}(Q_x|OK)$. Otherwise, SN_i broadcasts failure message FRM_i, where $FRM_i = ID_i|ID_{BS}|MAC_{K_{i,BS}}(Q_x|Fail)$

Belonging to O-Internal Node Another case is that SN_r belongs to O-Internal Node and the parent node of SN_r is SN_z. Without loss of generality, SN_r has k_1 child nodes belonging to Leaf Node, i.e., $SN_1, \cdots, SN_{k_1}$, and has k_2 child nodes belonging to L-Internal Node or O-Internal Node, i.e., $SN_{k_1+1}, \cdots, SN_{k_1+k_2}$. The messages sent from $SN_1, \cdots, SN_{k_1}$ are $M_1, \cdots, M_{k_1}$, and messages from $SN_{k_1+1}, \cdots, S_{k_1+k_2}$ are $\hat{M}_{k_1+1}, \cdots, \hat{M}_{k_1+k_2}$. SN_r would execute the following actions:

1. Wait for all SRMs from its grandson nodes in a time period
2. For each SRM_i, SN_r recomputes $MAC_{K_{z,i}}(Q_x|OK)$ and compares it with the MAC value in SRM_i.
3. If the verification fails or timesout, it will stop the aggregation procedure and report errors to the BS. Otherwise, it will pass with the next step.
4. Generate M_r such that M_r=$< ID_r|ID_z|Agg_r|H(Q_x|Agg_r|M_1|\ldots|M_{k_1}|\hat{M}_{k_1+1}$ $|\ldots|\hat{M}_{k_1+k_2}) >$ where $Agg_r = aggregate(V_r|V_1|\ldots|V_{k_1}|Agg_{k_1+1}|\ldots|Agg_{k_1+k_2})$.
5. Create $\hat{M}_r = M_r|M_1|\ldots|M_{k_1}|\hat{M}_{k_1+1}|\ldots|\hat{M}_{k_1+k_2}$.
6. SN_r broadcasts $\hat{M}_r$ to nearby sensor nodes, i.e., SN_z, $SN_1, \ldots, SN_{k_1+k_2}$ SN_r will buffer $\hat{M}_r$ for further result-checking

As described above, SN_r may belong to L-Internal Node or O-Internal Node. SN_z, the parent node of SN_r, should belong to O-Internal Node. Thus, it will wait for SRMs from all grandson nodes and further generate and broadcast $\hat{M}_z$.

In the aggregation flow, SNs belong to Leaf Node would send M_i messages to its parent node. When its parent node broadcasts $\hat{M}_i$ for verification, SNs will confirm the aggregated results and then broadcast decision messages. If the decision is successful, a SRM would be sent to its grandfather node to continue aggregation procedure. Otherwise, FRMs should be sent to the BS. Here we assume FRMs can be sent from different paths to the BS. Once the BS receives a FRM, it confirms the integrity of the FRM via the secret key between the sender and itself. Malicious nodes would be identified and revoked through BS broadcasting. In the end, aggregation would be done on the BS side. When the BS wants to raise the next aggregation, the BS would send authenticated request with Q_{x+1} to all deployed SNs for the next aggregation.

4.3. **Concrete Example.** Here we give an example to describe how SASHIMI works. In Fig. 4, SN_A, SN_B and SN_D belong to Leaf Node. In addition, SN_C and SN_E belong to L-Internal Node and O-Internal Node, respectively. The BS would broadcast a query number Q_x to all SNs to initial the aggregation procedure.

Fig. 4(a) depicts the initial step for aggregation. SN_A and SN_B create and disseminate M_A ($< A|C|V_A|H(Q_x|V_A) >$) and M_B ($< B|C|V_B|H(Q_x|V_B) >$) to SN_C individually. SN_A buffers M_A as $\sum_A$ and SN_B buffers M_B as $\sum_B$.

Once SN_C, which belongs to L-Internal Node, receives M_A and M_B, it will generate $\hat{M}_C$ and then broadcast it. Note that $\hat{M}_C$ equals $M_C|M_A|M_B$, where $M_C =< C|E|Agg_C|H(Q_x|Agg_C|A|B) >$ and $Agg_C = aggregate(V_C,V_A,V_B)$. As shown in Fig. 4(b), SN_A, SN_B and SN_E can obtain $\hat{M}_C$. Thus, SN_A and SN_B can verity the integrity of $\hat{M}_C$ respectively. For example, SN_A performs the following steps:

1. Recompute $aggregate(V_C, V_A, V_B)$ where V_C and V_B are derived from $\hat{M}_C$ and compare the results with the obtained Agg_C
2. Recompute $H' = H(Q_x|Agg_C|\sum_A|M_B)$ and compare the result with $H(Q_x|Agg_C|M_A|M_B)$ derived from $\hat{M}_C$

If the verification passes, SN_A broadcasts SRM_A. SN_B also broadcasts SRM_B.

In Fig. 4(c), if SN_E, which belongs to O-Internal Node, receives and verifies SRM_A and SRM_B from its grandson nodes, SN_E can confirm the integrity of $\hat{M}_C$. Then, SN_E will perform the following step.

1. Generate $M_E =< E|F|Agg_E|H(Q_x|Agg_E|M_D|\hat{M}_C) >$ where $Agg_E = aggregate(V_E|V_D|Agg_C)$. Note that Agg_C is obtained from $\hat{M}_C$
2. Create $\hat{M}_E = M_E|M_D|\hat{M}_C$

In Fig. 4(d), SN_E then broadcasts $\hat{M}_E$ to nearby SNs, i.e., SN_C, SN_D and SN_F. Similarly, SN_F, which belongs to O-Internal Node, would wait for SRM_C and SRM_D and then calculate $\hat{M}_F$. Obviously, the aggregation procedure is from bottom to top, util if receives the BS.

5. **Comparison.** In this section, we compare SASHIMI with SHIA in the following aspects.

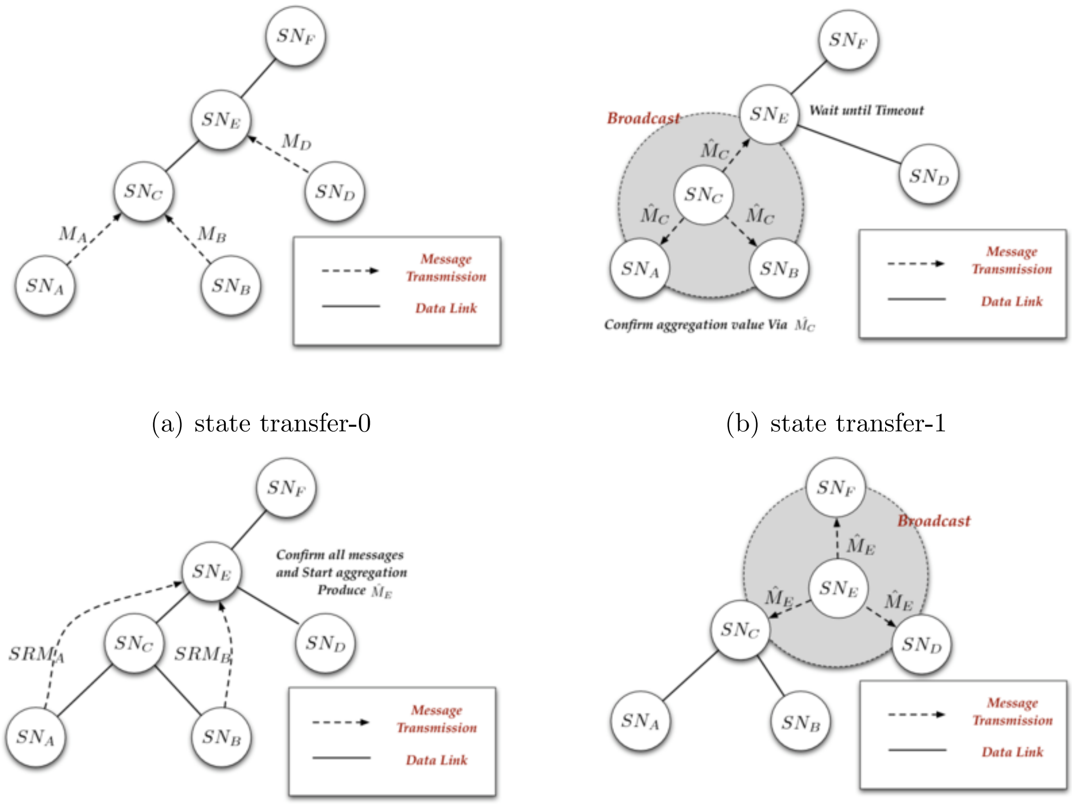
5.1. **The overhead of result checking.** The original purpose of data aggregation is to reduce communication complexity. However, an adversary may attempt to alter the final aggregated result. To guarantee that the BS obtains precise aggregation result, SASHIMI and SHIA allows each SN to verify its sensing data with aggregated result and corresponding MACs. The extra communication cost is actually incurred for result checking. Here we review the communication cost of SHIA and SASHIMI.

In SHIA, each SN must check all the aggregated results generated by its ancestors; consequently, each SN needs to receive its off-path value for verification. Fig. 5 shows an example of transmitting *off-path values.* In Fig. 5, base station R must transmit $Label_B$ and $Label_C$ to all nodes in a subtree rooted at node A. From the perspective of node D, it receives $Label_B$, $Label_C$, $Label_E$ and $Label_F$, and then performs the following actions:

1. transmit $Label_B$, $Label_C$, $Label_E$, $Label_F$, $Label_K$, and $Label_L$ to J
2. transmit $Label_B$, $Label_C$, $Label_E$, $Label_F$, $Label_J$, and $Label_L$ to K
3. transmit $Label_B$, $Label_C$, $Label_E$, $Label_F$, $Label_J$, and $Label_K$ to L

Relaying *off-path values* will obviously cause additional energy consumption for SNs in SHIA. In addition, each SN in SASHIMI only needs to verify the aggregated result of its parents; therefore, it only receives a message from its parent. For example, in Fig. 4, SN_A only receives $\hat{M}_C$ from SN_C.

As described above, SHIA requires additional overhead for transmitting *off-path values.* To realize the precise overhead of SHIA, we perform several experiments in the next section.



(a) state transfer-0 (b) state transfer-1

(c) state transfer-2 (d) state transfer-3

FIGURE 4. Concrete example of proposed scheme

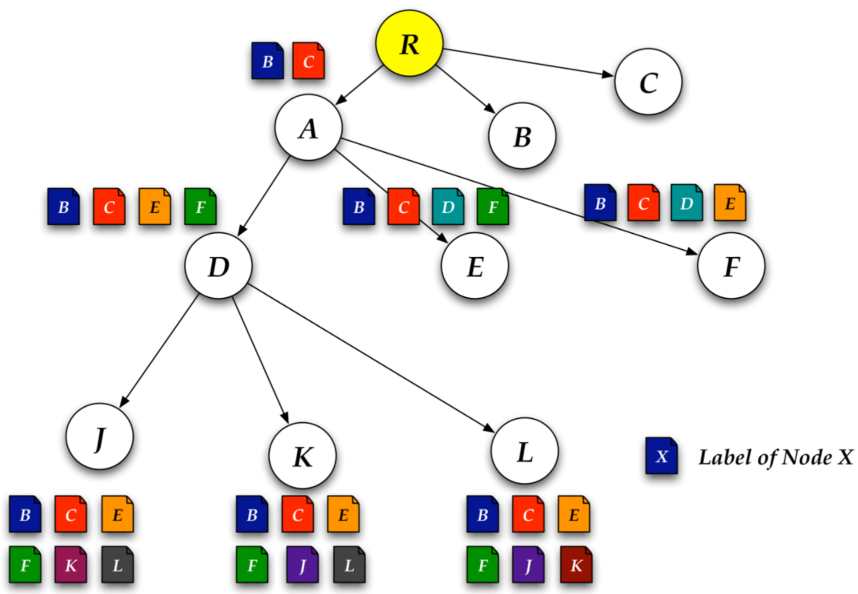


FIGURE 5. An example of transmitting off-path value

We also consider the storage overhead for result checking. Generally speaking, in SHIA, each SN needs to buffer the *Label*s of its child nodes. For example, in Fig. 5, D needs to buffer $Label_J$, $Label_K$ and $Label_L$ in the aggregation phase in SHIA. Then in the result checking phase, D transmits these buffered *Label*s to specific nodes. However, in SASHIMI, each SN only needs to buffer the message sent to its parent. For example, in Fig. 4, SN_B buffers M_B as Σ_B.

5.2. **Advantages over SHIA.** In section 3.3, we listed four drawbacks of SHIA. Here we demonstrate that SASHIMI overcomes these drawbacks.

1. In SHIA, each SN must realize the sequence of aggregation for commitment checking; thus, BS must broadcast the topology information to all SNs. Broadcasting topology inevitably incurs communication overhead for all deployed SNs. On the other hand, in SASHIMI, each SN only needs to know which SNs are within two hops.
2. If the network topology is changed, the BS needs to broadcast new topology information. This additional overhead would shorten the life-time of deployed SNs. Fortunately, in SASHIMI, each SN only maintains related information with SNs within two hops.
3. It is inefficient for SNs to dispatch *off-path values* from top to bottom in SHIA. SASHIMI utilizes a parallel methodology for result checking. Checking is followed by the aggregation procedure. This is why SASHIMI is efficient. Based on the experiments in chapter 6, we will show the efficiency of SASHIMI is indeed better than SHIA.
4. SASHIMI has low penalty cost. When attacks occur, SHIA continues aggregating results until the end. If the depth is high in WSN topology, the penalty becomes quite large since attacks can only be detected when the aggregation ends at the BS. In SASHIMI, attacks could be detected and reported during aggregation. Penalty estimation is given in chapter 6.

5.3. **Security Comparison.** Here we will prove the security of SASHIMI. The proposed scheme is secure against stealthy attacks, since the tampered results generated by parent nodes can be detected by its child nodes. Once attacks have been detected, the SN can notify the BS by broadcasting a FRM. Furthermore, if adversaries compromise two consecutive SNs on one path, the proposed scheme still remains secure. This is because FRMs are still broadcasted to the BS through different paths. When the BS receives FRMs, the tampered aggregation results can be identified and rejected. SHIA also maintains this security property. In conclusion, SASHIMI is as secure as SHIA.

An attack where a compromised node tampers with the aggregation results by modifying it sensing reading, e.g., inserting an extreme value, cannot be prevented. The only solution for this is by detecting the compromised sensors and revoking them. Hence, discussing sensing messages integrity of compromised sensors does not make sense. This requirement is not considered in our security requirements.

6. **Experiment.** To evaluate the performance of SASHIMI, we conducted some experiments and ran a few simulations.

6.1. **Assumptions & Design.** Our experiments places emphasis on SASHIMI and SHIA. The experiment starts by randomly deploying n sensors that form a query tree with maximum degree d. In addition, we define c to be proportion of compromised SNs in WSN. Through the following three experiments, we can realize the performance of SASHIMI under different n, d and c.

With a query tree, the communication cost of each sensor would vary since sensors are placed in different positions. We record the communication cost of all sensors in this query tree, and then calculate the following measurements.

1. The average total cost of each sensor.
2. The maximum total cost of each sensor.
3. The standard deviation of the total cost.

Each experiment runs 20 times with random topologies for each (n, d, c) tuple. Sensors used in experiment are MICAz sensor nodes. MICAz is capable of ATmega128L micro-controller. The architecture is 8-bit with 8MHz computation speed. The total programmable memory storage of each MICAz sensor is 128Kbytes. For the communication interface, MICAz uses ZigBee (802.15.4) to communicate with other MICAz sensors.For simple evaluation, energy consumption measurement is calculated based on the number of clock cycles [26].

6.2. **Evaluation on impact of d.** This experiment observes the impact of different values of d from 5 to 45. We set n to 3000 and c to 1%. Results are listed in Table 2. For example, in SASHIMI, if d is 5, the average cost of the entire WSN is 98.15 mJ and the maximum cost of all SNs is 205.9 mJ. From the average communication cost shown in Table 2, we observe:

1. The average cost of SASHIMI lies between 98.159 mJ and 94.589 mJ, and the cost of SHIA lies between 745.42 mJ and 2151.78 mJ. Obviously, the energy consumption of SASHIMI is better than SHIA.
2. Compared with SHIA, the cost of SASHIMI is more stable when d increases. Regardless of d changes, the average cost of SASHIMI remains at around 100 mJ. On the other hand, the communication cost of SHIA increases rapidly when d increases. This is because each SN needs to transmit enormous *off-path values* to its offspring in SHIA.

For the maximum cost among all SNs, we obverse the following facts from Table 2:

1. The maximum cost of SASHIMI is much lower than SHIA.
2. When d goes up, the gap between the maximum cost of SASHIMI and SHIA increases linearly. When d equals 5, the gap between the maximum cost of SASHIMI and SHIA is about 3939.2 mJ (4153.943-214.775). When d reaches 45, the gap grows to 107386.97 mJ (108664.08-1277.112).

Moreover, Table 2 shows the standard deviation δ of energy consumption with different value of d. Although the δ of SASHIMI and SHIA both increase with d. The variation of δ on SHIA is much higher than that of SASHIMI. δ of SASHIMI and SHIA are 49 and 791 , respectively, when d is 5 and the gap is 742. However, when d reaches to 45, δ of SASHIMI and SHIA becomes 124 and 2419, respectively, and the gap is 2195. Similar to average cost and maximum cost, the gap between δ on SASHIMI and SHIA also increases when d increases. In summary, energy consumption of SASHIMI is distributed to each sensor in a balanced manner. This property extends the lifetime of the SNs deployed in the WSN.

6.3. **Evaluation on impact of n.** The second experiment attempts to figure out the impact of different values for n ranging from 30 to 5000. We set d to 25 and c to 1%. Table 3 lists the results of Experiment 2.

We observe that when n lies between 30 and 5000, the cost of SASHIMI is less than SHIA. The gap between SHIA and SASHIMI increases with the number of SNs. For example, when n is 1000, the cost of SHIA and SASHIMI is about 1166 mJ and 96 mJ, respectively. The difference is 1070 mJ. However, when n reaches 5000, the cost of SHIA

TABLE 2. Results of the 1st experiment, unit is mJ

Degree	Avg(unit)		Max(unit)		Standard deviation	
	SASHIMI	SHIA	SASHIMI	SHIA	SASHIMI	SHIA
5	98.16	745.86	205.90	4780.08	49	791
7	98.28	757.65	255.60	7123.08	56	757
10	97.97	854.93	330.15	11556.14	66	957
12	97.90	919.51	379.85	14769.78	72	1004
15	97.51	1035.98	454.40	21157.11	79	1173
17	97.19	1132.54	504.10	25475.69	83	1209
20	96.86	1202.54	578.65	33508.45	90	1312
22	96.75	1295.83	628.35	38038.69	94	1420
25	96.48	1387.48	702.90	46010.22	99	1533
27	96.17	1352.92	751.36	53226.93	102	1587
30	96.04	1534.57	827.15	60234.63	106	1775
32	95.66	1610.13	875.61	69229.88	109	1862
35	95.31	1774.59	950.16	78072.04	113	1828
37	95.46	1700.50	993.65	86546.78	115	2028
40	94.88	1958.80	1070.68	102959.77	119	2164
45	94.59	2151.78	1186.23	129804.42	124	2419

and SASHIMI changes to about 1477.30 mJ and 96.44 mJ. The difference also increases to 1373.86 mJ. Therefore, We can deduce that performance of SASHIMI is better than SHIA if n is larger than 5000.

The maximum cost and standard deviation of SASHIMI and SHIA is also shown in Table 3. According to Table 3, we observe:

1. The standard deviation of SASHIMI fall between 68 and 100, and the standard deviation of SHIA falls between 1147 and 1666. This shows that the stability of SASHIMI is better than SHIA.
2. When the number of SNs increases, the standard deviation of SASHIMI also increases. However, when the number of SNs exceeds 200, the standard deviation of SASHIMI will remain at about 100 mJ. On the other hand, the standard deviation of SHIA is unpredictable.

6.4. **Evaluation of impact from c.** The final experiment focuses on the effect of varying c from 0% to 20%. We set n to 2000 and d to 30. As shown in Table 4, the average cost of SHIA is not affected by c. No matter how c changes, the average cost of SHIA still remains at about 1400 mJ. This overhead is affordable since the result-checking phase is performed after the BS receives the aggregation results. On the other hand, the energy consumption of SASHIMI decreases as c increases. This is because SASHIMI stops the aggregation procedure when it detects an attack. Therefore, SASHIMI saves more energy when there are many compromised node in the WSN.

Additionally, we observed that the maximum cost and standard deviation of SASHIMI decreases when c increases. In brief, the communication cost of SASHIMI distributes more evenly when c increases. For SHIA, these two values are constant (≈ 1400 mJ).

7. **Conclusion.** SASHIMI is an efficient and rapid-response aggregation algorithm for generic WSNs. SASHIMI provide aggregation integrity against multiple malicious sensor nodes form the bottom to the top. If an adversary tries to modify the aggregated result at intermediate nodes, tampering can be detected within two-levels of the hierarchy.

Table 3. Results of the 2nd experiment, unit is mJ

n	Avg(unit)		Max(unit)		Standard deviation	
	SASHIMI	SHIA	SASHIMI	SHIA	SASHIMI	SHIA
30	59.95	531.02	321.63	9741.64	68	1243
50	65.88	675.46	496.82	15394.13	88	1138
100	89.7	754.01	570.66	20074.81	82	1540
200	93.52	892.94	645.04	25647.42	90	1390
400	94.89	1048.10	668.99	31811.11	95	1562
600	95.71	1144.71	689.94	37724.52	96	1521
800	95.97	1155.44	696.16	38049.34	97	1065
1000	96.04	1166.87	699.17	39706.75	97	1213
1200	96.13	1230.13	701.66	40990.08	98	1447
1500	96.25	1278.54	702.90	41402.76	98	1346
1700	96.54	1293.54	702.90	41426.73	98	1521
2000	96.56	1329.33	702.90	43374.34	98	1415
2500	96.38	1320.79	702.90	43354.38	98	1569
3000	96.43	1392.73	702.90	46421.58	99	1561
3500	96.49	1383.04	702.90	47233.64	99	1640
4000	96.52	1402.74	702.90	50662.94	99	1584
4500	96.46	1473.25	702.90	52159.26	99	1671
5000	96.44	1477.30	702.90	50493.87	99	1666

Table 4. Results of the experiment 3, unit is mJ

c	Avg(unit)		Max(unit)		Standard deviation	
	SASHIMI	SHIA	SASHIMI	SHIA	SASHIMI	SHIA
0%	85.01	1449.43	809.40	58894.06	122	1753
1%	95.81	1482.02	824.67	61597.83	106	1625
2%	91.32	1492.61	823.42	61712.31	95	1740
3%	87.84	1481.17	824.67	59037.83	85	1654
4%	84.84	1498.49	813.48	58647.78	77	1574
5%	82.77	1483.57	817.92	60122.80	68	1597
6%	80.65	1540.66	798.57	58200.48	61	1721
8%	77.51	1491.99	761.30	56993.03	51	1674
10%	75.29	1509.46	703.61	58824.83	43	1663
12%	73.29	1475.79	646.99	59880.51	38	1737
14%	71.78	1468.19	600.48	58265.71	33	1668
16%	70.65	1497.92	512.27	58776.91	30	1687
18%	69.82	1495.10	491.85	57464.29	28	1681
20%	68.90	1489.99	389.79	56782.69	26	1671

Furthermore, aggregation is stopped and error reports are also transmitted to the base station. In SASHIMI, the cost of result checking is well distributed to all the deployed sensors.

Acknowledgment. The authors would like to thank anonymous reviewers for their valuable comments and suggestions, which certainly led to improvements of this paper. This work was supported by the National Science Council, Taiwan, R.O.C., under Contract NSC 100-2628-E-007-018-MY3. The corresponding author is Professor Hung-Min Sun.

REFERENCES

[1] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, Impact of network density on data aggregation in wireless sensor networks, *Proc. of the 22nd IEEE International Conference on Distributed Computing Systems*, pp. 457-458, 2002.

[2] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, Tag: a tiny aggregation service for ad-hoc sensor networks, *Proc. of the 5th symposium on Operating systems design and implementation*, vol. 36, pp. 131-146, 2002.

[3] H. Chan, A. Perrig, and D. Song, Secure hierarchical in-network aggregation in sensor networks, *Proc. of the 13th ACM Conference on Computer and Communications*, pp. 278-287, 2006.

[4] H. M. Sun, Y. H. Lin, Y. C. Hsiao, and C. M. Chen, An efficient and verifiable concealed data aggregation scheme in wireless sensor networks, *Proc. of IEEE International Conference on Embedded Software and Systems*, pp. 19-26, 2008.

[5] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, and C. MIT, Energy-efficient communication protocol for wireless microsensor networks, *Proc. of the 33rd International Conference on System Sciences*, vol. 8, pp. 3005-3014, 2000.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, *IEEE Trans. Wireless Communications*, vol. 1, no. 4, pp. 660-670, 2002.

[7] O. Younis and S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, *IEEE Trans. Mobile Computing*, vol. 3, no. 4, pp. 366-379, 2004.

[8] C. M. Chen, Y. H. Lin, Y. C. Lin, and H. M. Sun, RCDA: Recoverable concealed data aggregation for data integrity in wireless sensor networks, *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 4, pp. 727-734, 2012.

[9] S. Lindsey and C. Raghavendra, PEGASIS: Power-efficient gathering in sensor information systems, *Proc. of the IEEE Aerospace Conference*, vol. 3, pp. 1125-1130, 2002.

[10] K. Du, J. Wu, and D. Zhou, Chain-based protocols for data broadcasting and gathering in the sensor networks, *Proc. of the 17th International Symposium on Parallel and Distributed Processing*, 2003.

[11] M. Ding, X. Cheng, and G. Xue, Aggregation tree construction in sensor networks, *Proc. of the 58th International Conference on Vehicular Technology*, vol. 4, pp. 2168-2174, 2003.

[12] L. Hu and D. Evans, Secure aggregation for wireless networks, *Proc. of the Symposium on Applications and the Internet Workshops*, pp. 384-391, 2003.

[13] P. Jadia and A. Mathuria, Efficient secure aggregation in sensor networks, *Proc. of the 11th International Conference on High Performance Computing*, pp. 40-49, 2004.

[14] W. Du, J. Deng, Y. Han, and P. Varshney, A witness-based approach for data fusion assurance in wireless sensor networks, *Proc. of International Conference on Global Telecommunications*, vol. 3, pp. 1435-1439, 2003.

[15] B. Przydatek, D. Song, and A. Perrig, SIA: Secure information aggregation in sensor networks, *Proc. of the 1st International Conference on Embedded networked sensor systems*, pp. 255-265, 2003.

[16] Y. Yang, X. Wang, S. Zhu, and G. Cao, SDAP: A secure hop-by-hop data aggregation protocol for sensor networks, *Journal of ACM Transactions on Information and System Security*, vol. 11, no. 4, 2008.

[17] H. Sanli, S. Ozdemir, and H. Cam, SRDA: secure reference-based data aggregation protocol for wireless sensor networks, *Proc. of the 60th IEEE Conference on Vehicular Technology*, vol. 7, pp. 356-367, 2004.

[18] H. Chan and A. Perrig, Efficient security primitives derived from a secure aggregation algorithm, *Proc. of the 15th ACM Conference on Computer and Communications Security*, pp. 521-534. 2008.

[19] J. Zhao, R. Govindan, and D. Estrin, Computing aggregates for monitoring wireless sensor networks, *Proc. of the 1st IEEE Conference Workshop on Sensor Network Protocols and Applications*, pp. 139-148, 2003.

[20] B. Krishnamachari, D. Estrin, and S. Wicker, Modelling data-centric routing in wireless sensor networks, *Proc. of the IEEE International Conference on Computer Communications*, vol. 2, pp. 1-11, 2002.

[21] T. Melodia, D. Pompili, and I. Akyildiz, Optimal local topology knowledge for energy efficient geographical routing in sensor networks, *Proc. of the 23th IEEE AnnualJoint Conference on Computer and Communications Societies*, vol. 3, pp. 1705-1716, 2004.

[22] L. Eschenauer and V. Gligor, A key-management scheme for distributed sensor networks, *Proc. of the 9th ACM Conference on Computer and Communications Security*, pp. 41-47, 2002.

[23] H. Chan, A. Perrig, and D. Song, Random key predistribution schemes for sensor networks, *Proc. of the IEEE Symposium on Security and Privacy*, pp. 197-215, 2003.

[24] R. Di Pietro, L. Mancini, and A. Mei, Random key-assignment for secure wireless sensor networks, *Proc. of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 62-71, 2003.

[25] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili, A pairwise key predistribution scheme for wireless sensor networks, *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228-258, 2005.

[26] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, Energy analysis of public-key cryptography for wireless sensor networks, *Proc. of the IEEE International Conference on Pervasive Computing and Communications*, pp. 324-328, 2005.