International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **10(A)**, October **2012**

 ISSN 1349-4198
pp. **6503–6514**

A LIGHTWEIGHT SECURE DATA AGGREGATION PROTOCOL FOR WIRELESS SENSOR NETWORKS

Hung-Min Sun, Chiung-Hsun Chen and Po-Chi Li

Department of Computer Science
National Tsing Hua University
No. 101, Section 2, Kuang-Fu Road, Hsinchu 30013, Taiwan
hmsun@cs.nthu.edu.tw

Received June 2011; revised November 2011

Abstract. *Data aggregation is a widely used technique in wireless sensor networks. There has been many related work proposed to address the data aggregation. However, very few of them focus on data integrity. An attacker can simply forge data to affect the decision of the base station via compromised nodes. Moreover, previous data aggregation protocols do not consider outlier data. Some emergent events cause outlier data will be ignored. We, therefore, propose a lightweight secure data aggregation protocol to resist attacks and handle outlier data. The detailed security analyses and simulation results will show that the proposed scheme is effective and efficient.*
Keywords: Data aggregation, Wireless sensor network, Data integrity

1. **Introduction.** Wireless sensor networks were originally deigned to monitor battlefields. Nowadays, they have a larger range of applications, e.g., monitoring fires, detecting mudflows and landslides, tracking wild lives [1]. A wireless sensor network is composed of one or several base stations and hundreds of sensor nodes. A base station (BS) may be a laptop or a powerful computer. On the other hand, the computation ability, memory, and energy of a sensor node are restricted. According to previous researches [2], communication cost is much more than computation cost in wireless sensor networks; sensor nodes must also reduce the energy consumption to extend their lives. Therefore, many kinds of data aggregation schemes [3-19] have been proposed. Although these schemes reduce communication costs, they do not consider security [20]. In these schemes, internal nodes in wireless sensor networks can easily forge, modify, and drop messages. When internal nodes launch forging, altering, or dropping attacks, the aggregated result from the sensing data will not represent the original sensing data. As a result, erroneous data are sent to the BS potentially causing it to adopt incorrect processes. Because previous schemes do not consider security, an attacker can learn the content of packets by eavesdropping messages.

Moreover, previous data aggregation protocols [3-19] do not deal with outlier values. When neglecting the outlier data, these protocols could induce serious consequences. For instance, an extremely high temperature could indicate a fire accident. However, the data aggregation process will hide the anomaly, and the base station will not be aware of the event given the aggregation values. In order to solve these problems, we propose a lightweight secure architecture to resist the attacks mentioned above and to process the outlier data. The proposed scheme transmits outlier sensing data without aggregation and utilizes an onion proof to detect misbehaving nodes. And the proposed scheme can resist eavesdropping attacks, forgery attacks, replay attacks, and dropping attacks. The analyses and simulations will show that the proposed scheme can resist attacks effectively.

The rest of this paper is organized as follows. Related work is discussed in Section 2. Section 3 describes the proposed lightweight secure data aggregation protocol. The analyses and simulations are in Section 4. Finally, we conclude the paper in Section 5.

2. **Related Work.** In this section, previous researches are described. In [4], Roy et al. mentioned that each sensor node computes a synopsis by its sensing value and identifier. Then, an aggregator combines these synopses from other nodes with its synopsis. After collecting all synopses, the base station can estimate the sum or count value of the network. Nevertheless, false results may occur the final synopsis is not located within a predefined range.

Castelluccia et al. [5] proposed that sensor nodes can aggregate encrypted data directly instead of decrypting them and performing aggregation. In this architecture, each node encrypts its sensing value by a homomorphic function and sends the ciphertext. After receiving these ciphers, an aggregator performs addition and forwards the results. Finally, the base station can compute the final result from these ciphertexts. This method can reduce the computation cost of decryption. An adversary, however, could arbitrarily modify or inject data into the ciphertext without being detected by the base station since the base station can not verity these ciphers.

In [7], Cam et al. mentioned that each sensor node computes a pattern code by its sensing value and a pattern seed issued by the group leader. Then, each node sends its identifier, pattern code, and a timestamp to the group leader. After collecting these pattern codes, the group leader will compare them and will request that the sensor node, which computes a specific pattern code, sends its real value to the base station.

Wagner [9] proposed the idea of resilient aggregation. In this scheme, the base station can utilize trimming and truncation to achieve resilient aggregation. Trimming means the base station will neglect fixed percentage highest and lowest values, e.g., five percents. Truncation means that the base station will set an upper and lower bound; any values located out of bounds, will be modified into the boundary. In this scheme, each sensor node must send its value to the base station and let the base station reduce arts of these values to compute the final result. Therefore, resilient aggregation does not involve real aggregation.

In [10], Hu et al. aimed to deal with the malicious nodes deployed in the networks and a node compromise. The method is divided into two parts: delayed aggregation and delayed authentication. Delayed aggregation is when node C receives a message from node B, and B is unable to aggregate the message immediately. Node B has to forward the message to B's parent A. When A receives this message, it deals with the message. Delayed aggregation increases communication overhead by one-hop but it can detect whether or not C or B as a compromised node if A is normal. However, the method is unable to detect modified data if there are two consecutive compromised nodes. Delayed authentication means that the node authenticates messages after a time delay, i.e., the authentication key is only revealed to the authenticator after time has expired.

Yang et al. [11] proposed a protocol which utilizes divide-and-conquer and commit-and-attest techniques. Divide-and-conquer means that the protocol will partition all sensor nodes into groups of similar sizes. Commit-and-attest means that each group leader will aggregate data along with a commitment. Whenever the base receives these messages, it can discover the suspicious group and ask the suspicious group to prove its correctness. The base station detects the attack by finding the inconsistency between the aggregated value and the reconstructed aggregate.

Because previous schemes do not consider security, an attacker could falsify or tamper with real sensing data when the attacker tries to cheat the BS in data aggregation protocols. In addition, data aggregation also leads to some side effects. For example, outlier values in wireless sensor networks often represent some special events, e.g., fire accidents. Previous data aggregation protocols ignore the meaning of outlier values; hence, we focus on handling outlier data. In other words, we proposed a scheme that can detect a forged emergency and guarantee that the base station can receive real emergency messages.

3. The Lightweight Secure Data Aggregation Protocol.

The Lightweight secure data aggregation protocol is divided into three parts, such as topology construction, data processing, and verification. The details are showed as follows.

3.1. Settings and notations.

Most data aggregation protocols of previous researches do not deal with outlier data. An outlier value could represent an emergency in wireless sensor networks; however, traditional data aggregation process would conceal abnormal values. Consequently, the base station will not be aware of unusual events. Hence, we propose a lightweight secure data aggregation protocol to solve this problem.

In this paper, all sensor nodes can be classified as base stations, leaf nodes, or aggregators. The leaf nodes are responsible for sensing data and reporting data to the base station. Aggregators are responsible for aggregating data and relaying data. The base station collects data from each sensor node and analyzes the received data to detect emergency events and attacks. In addition, sensing data are classified as either emergency data or usual data. If sensing data are out of the predefined thresholds, the sensing data is defined as emergency data. Otherwise, the sensing data is defined as normal data.

Some assumptions are defined in this paper. Firstly, the topology of the network is a tree rooted at the base station. The proposed protocol does not rely on any specific tree-construction protocol. In other words, the proposed protocol can be applied to all kinds of tree-construction routing protocol. Secondly, attackers can only compromise a small percentage of aggregators (at most half of all aggregators). Thirdly, sensor nodes can achieve loose time synchronization with the base station [21]. Fourthly, each sensor node shares pairwise keys with its neighbors. The pairwise keys are generated by sensor nodes after deployment [22,23]. Fifthly, the sensor network is secure for a period of time following initial deployment. Sixthly, each node has a unique pairwise key with the base station for secure communication. The unique pairwise key is preloaded in each node. Seventhly, each node can authenticate the broadcast packet from the base station by using μTesla scheme. Finally, the probability of an emergency event is pretty low. Table 1 shows the notations of this paper.

3.2. Topology construction.

After deploying sensor nodes, the base station *BS* sends a topology-construction packet to each sensor node in order to construct a topology tree [24]. In the topology-construction packet, the source and sender are set as S. The HopCount is set as zero. The sensing type can be temperature, humidity, etc. The default aggregation function is the average. Table 2 shows the format of the topology-construction packet.

Upon receiving the topology-construction packet, a sensor node A executes the following processes. Based on the source and *Seq* of the packet, it checks whether it has received the packet or not. If not, it increases the *HopCount* by one to generate *HopCount'*. Then it records the sender of the packet and the *HopCount'* as parent and Level, respectively, in its relation table. It sets A and *HopCount'* as the new sender and *HopCount* to assemble a new packet. Finally, it broadcasts the packet. If A receives the same request, it compares the *HopCount* with its Level. If the *HopCount* is less than its Level, it records the sender

Table 1. Notations

Notation	*Definition*
$K_{i,j}$	*The pairwise key shared between node i and j*
K_i	*The unique secret key shared by the base station and node i*
$E(K, m)$	*Using the unique secret key K to encrypt the message m*
$MAC(K, m)$	*Message authentication code generated by message m and key K*
$A, B, \ldots, Z$	*Identifiers of sensor nodes*
D_A	*Node A's data*
$Agg(D_A, D_B)$	*The aggregated value of A's and B's data by aggregation function*
T_A	*The timestamp of node A generated by A's local clock*
HopCount	*The hop count value of each node to the base station*
l	*The predefined lower bound of sensing threshold*
u	*The predefined upper bound of sensing threshold*

Table 2. The query message

Source	*Sender*	*Seq*	*HopCount*	*Sensing type*	*Aggregation function*

Table 3. The relation table of node D

Level	*Parent*	*Uncle*	*Sibling*	*Child*
1	A	B	C, E, F	I, J

as its uncle. If the *HopCount* is larger than its Level, it records the sender as its child. If they are equal, it records the sender as its sibling.

Via this process, each node can discover its parent, uncles, sibling, and children in the topology, and each node can also inform the *BS* about its relation to its neighbors. Hence, the *BS* can know the topology of the whole networks. Table 3 shows the construction of the relation table.

3.3. **Data processing.** In this phase, leaf nodes and aggregators perform different data processing according to the content of the messages. The contents of a message can be classified into an emergency event and a usual event. When a node senses an emergency event, the node transmits the emergency event via a single path or via multiple paths. On the contrary, a node transmits a usual event only by a single path. For a clearer description, Figure 1 is utilized to describe the data processing. All leaf nodes sense environment and forward sensing data to their parents. For example, in Figure 1, node N senses an emergency event and node O senses a normal event. O generates a timestamp T_O and encrypts its identifier O, sensing data D_O, and the timestamp T_O by the pairwise key shared with its parent G. Then O generates a data packet as (1) and sends the packet to G.

$$Data, O, T_O, E_{K_{OG}}[O, D_O, T_O] \tag{1}$$

Node N adopts the same processes. However, D_N is out of the predefined range. N must utilize the secret key K_N shared with the base station to generate a $proof_N$ and append the $proof_N$ to the packet. Therefore, N generates a data packet as (2) and sends the data packet to G. Finally, N evaluates a time period of receiving an acknowledgement time from S. It stores the next hop identity G, packet transmitting time, and the expected time period into its transmission table. The transmission table is shown in Table 4.

$$Data, N, T_N, E_{K_{NG}}[N, D_N, T_N, proof_N] \text{ where } proof_N = E_{K_A}[N, D_N, T_N] \tag{2}$$

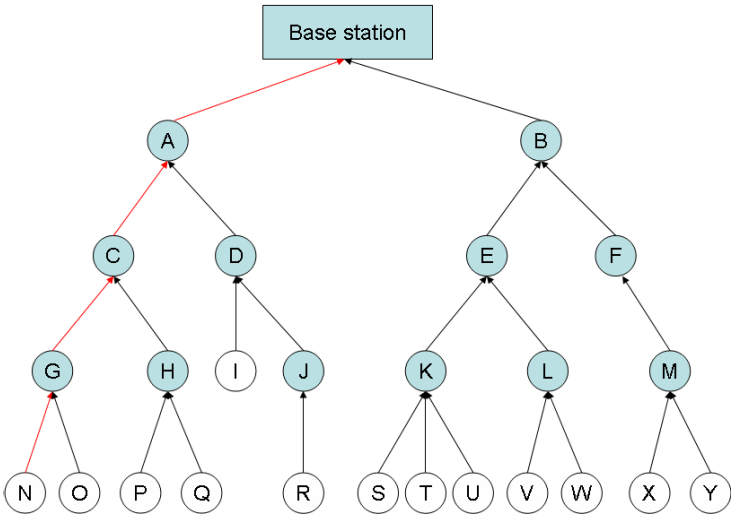

FIGURE 1. Data processing with single-path routing

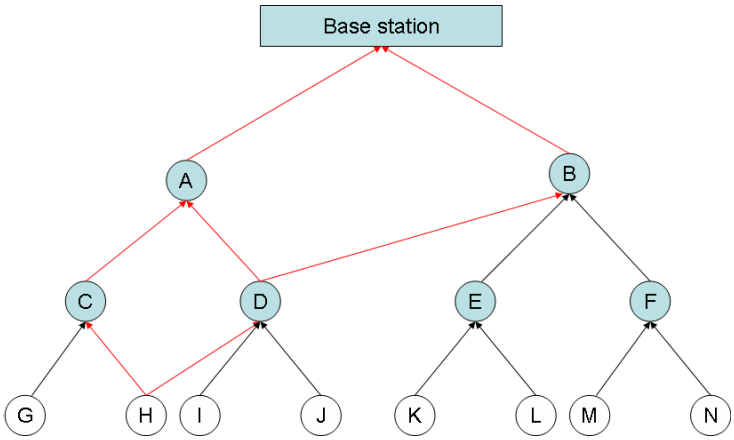

FIGURE 2. Data processing with multiple path routing

After sending an emergency data packet, the leaf node waits for an ACK from BS. If the leaf node does not receive an ACK after the expected time period, it adopts multiple paths to send the emergency data. It can send the data packet to its parent and its uncle at the same time. For example, in Figure 2, each node can send emergency data to its parent and its uncle. This way, the emergency data will have a better opportunity at arriving at the BS.

When an aggregator A receives a data packet from its children, it checks the timestamp of the packet. If the timestamp is not valid, A discards the packet directly. Otherwise, if all data packets contain normal events, it performs the aggregation function on the content of each data packet to generate D_A and sets A, T_A, and D_A as the sender, timestamp, and sensed data. Finally, it assembles a new data packet and sends the data packet to its parent. When some data packets contain emergency events, A transmits the emergency sensing data without aggregating and calls the aggregation function on other sensing data to obtaining D_A. Finally, it sets A, T_A, and D_A as the sender, timestamp, and sensing data and generates a new data packet with emergency data and $proof_G = E_{K_G}[G, proof_{children}]$. Then it sends the data packet to its parent. For example, after receiving data packets from N and O, G discovers that D_N is an emergency event. It generates a new data packet as (3). G appends the emergency data sent by N in (3). This way, the emergency

data will not be hidden within the aggregated value.

$$Data, G, T_G, E_{K_{GC}}[G, D_G, T_G], E_{K_{GC}}[N, G, D_N, T_N, proof_G] \\ \text{where } D_G = Agg(D_O),\ proof_G = E_{K_G}[G, proof_N] \tag{3}$$

When H receives data packets from P and Q, H generates another type of data packet as (4) to indicate no emergency event. It only applies aggregation function to generate D_H.

$$Data, H, T_H, E_{K_{HC}}[H, D_H, T_H] \text{ where } D_H = Agg(D_P, D_Q) \tag{4}$$

Table 4. Transmission table

Next Hop	Transmission Time	Expected Acknowledging Time

Each aggregator performs the same processes. Finally, BS can receive the data packet.

3.4. **Verification.** Whenever the base station *BS* receives a data packet from an aggregator, it can decrypt the data packet with a pairwise key shared with the aggregator. If *BS* cannot decrypt the data packet or the timestamp of the packet is not valid, *BS* discards the packet directly. Otherwise, it receives the aggregated sensing data. When the packet contains emergent data, *BS* checks whether the sequence of node identities recorded in the packet conforms to the network topology. If not, *BS* discards the packet directly. If yes, *BS* finds the corresponding pairwise keys shared with each node recorded in the packet and decrypts the onion *proof*. If all decryption are correct, *BS* accepts the emergency data and replies an acknowledgement (ACK) to the sender of the emergency data immediately. Otherwise, according to the error decryption, *BS* records the suspected nodes. For example, in Figure 1, when the decryptions are as Table 5, *BS* will record C and G as suspected nodes and increases their misbehavior count. When the misbehavior count of an aggregator is larger than a predefined threshold, *BS* defines the aggregator as an attacker. The base station maintains a misbehavior table to record the misbehavior count of each node. As shown in Table 6, the base station keeps the ID, behavior, and misbehavior count.

4. **Security Analyses and Comparisons.** Because wireless sensor networks are susceptible to several attacks, the proposed scheme must resist several attacks, such as eavesdropping attacks, replay Attacks, dropping attacks, and forgery attacks. The details are showed as follows.

Table 5. The example of decryption results

Node id	A	C	G	N
Decryption result	correct	correct	error	error

Table 6. The misbehavior table

Indemnifier	Behavior	misbehavior count
A	Normal	0
B	Attacker	Threshold + 1

4.1. **Eavesdropping attacks.** An adversary can overhear transmitted packets and try to learn information from the packets. Because each packet encrypts the sensing data with a pairwise key in proposed scheme, adversaries cannot decrypt the information of in the sensing data without the corresponding keys. For example, in Figure 3, node B is the parent of node A. When A sends its sensing data to B, the data packet is shown as (5).

$$Data, A, T_A, E_{K_{A,B}}(A, D_A, T_A) \tag{5}$$

When an adversary overhears the packet, he can only determine the sender and the timestamp of the packet. He cannot decrypt the packet without the pairwise key, $K_{A,B}$.

FIGURE 3. Altering attack

4.2. **Forgery attacks.** An adversary will alter a data packet in order to skew the aggregated value in wireless sensor networks. Firstly, a compromised node modifies an emergency message to a usual one. We utilize an example to show how the proposed protocol can resist this type attacks. In Figure 3, node A senses an emergent event and sends the message to its parent node B.

$$Data, A, T_A, E_{K_{A,B}}[A, D_A, T_A, proof_A] \tag{6}$$

When node B receives the packet from node A, it performs some simple processing and sends the data packet as (7) to node C.

$$Data, B, T_B, E_{K_{BC}}[B, D_B, T_B], E_{K_{BC}}[A, B, D_A, T_A, proof_B] \tag{7}$$

In this example, node C is an attacker. After receiving the packet, C alters the contents of the packet into another packet as (8) and forwards it.

$$Data, C, T_C, E_{K_{SC}}[C, D_C, T_C] \tag{8}$$

After receiving the packet (8), the base station can verify the value and discover the sensing data is in the predefined range. In such a case, the base station will not reply an acknowledgement to the sensor node A. Therefore, A cannot receive the acknowledgement from the base station within a reasonable time. It will retransmit the emergency sensing data via another route.

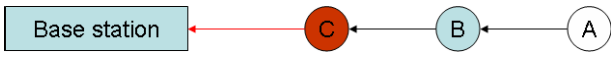

FIGURE 4. Altering attack

Secondly, we suppose a compromised node modifies a usual message to an emergency one. For example, in Figure 4, A sends a data packet (9) with usual sensing data to node B.

$$Data, A, T_A, E_{K_{A,B}}(A, D_A, T_A) \tag{9}$$

After receiving the message, B performs some processing and forwards another data packet as (10) to C. In this example, node C is an attacker. After receiving the packet, C alters the contents of the packet to cheat the base station. C generates a $proof_C$ and assembles a data packet as (10).

$$Data, C, T_C, E_{K_{SC}}[B, D_C, T_C], E_{K_{SC}}[A, B, C, D_A, T_A, proof_C] \tag{10}$$

After receiving the packet as (10), the base station decrypts the packet and discovers that node A senses an emergency. Then, the base station retrieves the corresponding keys

to decrypt $proof_C$. Because the $proof_C$ is faked, the base station will discover that B and C are suspicious nodes.

4.3. **Replay attacks.** In order to resist replay attacks launched by attackers, each sensor node that senses an emergency event must append a timestamp. After receiving a data packet, each aggregator and the BS will verify the timestamp. When the verification fails, the data packet will be directly dropped. In other words, the proposed scheme can resist reply attacks. For example, node A sends an emergency data packet to the base station with a timestamp T_A. When an adversary overhears the packet, the adversary can resend the packet after a period of time. When the base station receives the packet, it can decrypt the message and gain prior timestamp T_A. The base station will verify T_A. If the verification is correct, the BS accepts the data packet. Otherwise, it drops the data packet directly.

4.4. **Dropping attacks.** In this paper, we focus on the transmission of emergency data packets. Therefore, the proposed protocol will not resist dropping attacks on normal data packets.

Dropping attacks can be divided into selective dropping attacks and black hold attacks. In order to resist packet dropping, the base station has to reply an acknowledgment after receiving an emergency data packet. When a sensor node which senses an emergency cannot receive an acknowledgment from the base station, it retransmits the emergency data packet via other routes and requests another node to forward it. The message will eventually arrive at the base station unless the attacker can compromise all the neighbors of the source node.

The mechanism of retransmission causes delayed messages. Therefore, each node will transmit emergency data packets via multiple paths when the sensor network is focused on real time applications. Because attackers can only compromise a small amount of sensor nodes, the emergency data packets can still be transmitted to the *BS*. Figure 5 shows that the multiple path routing can indeed resist packet dropping attacks.

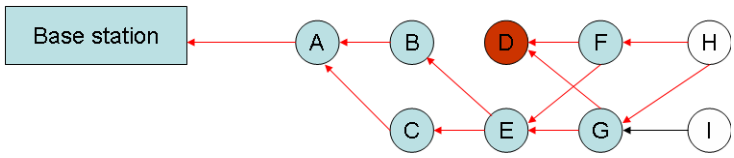

FIGURE 5. Resistance to dropping packets

4.5. **Comparisons.** In this part, the proposed protocol is compared with other methods based on security features, memory cost, computation cost, communication cost, and application environment.

4.5.1. *Comparisons of security features.* In Table 7, we display the comparisons of security features. In order to reduce the memory cost and computation cost for sensor nodes, some methods transmit messages in plaintext. Therefore, these methods cannot guarantee data confidentiality. In other words, an attacker can easily obtain the contents of messages by overhearing transmitted packets.

4.5.2. *Comparisons of memory cost.* In this part, we show the memory cost. Because the base station has virtually unlimited computation ability and storage, the memory cost of the base station is not considered here. The memory cost is the amount of stored keys in a sensor node. In all the compared protocols, each sensor node shares one unique key with the base station for secure communication with the *BS*.

TABLE 7. Comparisons of security features

Property / Scheme	data confidentiality	data integrity	replay attack resistance	dropping resistance	forging detection
Roy [4]	×	◯	×	×	◯
Castelluccia [5]	◯	×	×	×	×
Çam [7]	◯	◯	×	×	×
Hu [10]	×	◯	×	×	◯
Yang [11]	◯	◯	◯	×	◯
Chan [14]	×	◯	◯	×	◯
Proposed scheme	◯	◯	◯	◯	◯

◯ : The feature is satisfied.

× : The feature is not satisfied.

TABLE 8. Comparisons of memory cost

Memory cost / Schemes	# of stored keys
Roy [4]	2
Castelluccia [5]	1
Çam [7]	2
Hu [10]	2
Yang [11]	$n+2$
Chan [14]	2
Proposed scheme	$n+2$

n: the number of one-hop neighbors of each node (on average)

In some protocols, including ours, sensor nodes have to authenticate the broadcast from the base station and they keep the first key K_0 of the μTesla key chain. In addition, each node establishes one pairwise key with all its one-hop neighbors in some methods. We describe the number of neighbors as n. The comparisons of memory cost are shown in Table 8.

4.6. **Simulation.** In our simulations, we adopt Network Simulator (NS-2) [25,26] as MANETs simulation tool. Berkeley's Network Simulator (NS2) includes wireless extensions made by the CMU Monarch project. The simulation parameters are itemized in Table 9. The network consisted of 100 nodes in a 1000m × 1000m rectangular space. The total simulation time is 100 seconds. The transmission range of each node is 250m. The size of a data packet is 128 bytes. The percentage of malicious nodes is between 0% and 40%. Node mobility is 0 meter/second. The pause time is set to 100 seconds.

Because Roy [4], Hu [10], and Chan [14] do not adopt encryption functions, they cannot provide data confidentiality. For fair comparison, we only simulate similar schemes.

Firstly, we simulate that the influence of dropping attacks on the average emergency packet delivery ratio. Each attacker drops the emergency data packets. Figure 6 shows the average emergency packet delivery ratio. In Figure 6, the percentage of attacks is variable between 5% and 25%. Clearly, when the percentage of attacks is increased, the average emergency packet delivery ratio is decreased. Although all schemes have roughly the same average emergency packet delivery ratio when the percentage of attacks is 5%; only the proposed scheme can provide more than 80% of average emergency packet delivery ratio,

TABLE 9. Simulation parameters

Number of Nodes	100
Transmission Range (m)	10m
Simulation Area (m^2)	100m × 100m
Simulation Time (sec)	100
Data Packet Size (byte)	128
Frequency of sending Packet (times/s)	4
AES (256bit)	0.037ms
SHA-1	0.0015ms

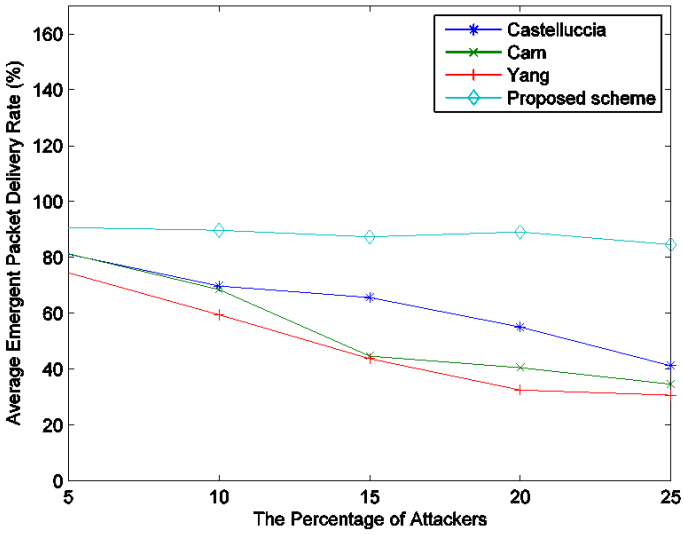

FIGURE 6. The influence of dropping attacks

when the percentage of attacks is raised to 25%,. In other words, the proposed scheme can resist dropping attacks more effectively.

Next, we simulate that the influence of emergency data packets on average delay of data transmission. Figure 7 shows the influence of emergency data packets. Because other schemes do not deal with emergency data, the average delay of data transmission of other schemes will not have any influence. On the contrary, the proposed scheme deals with the emergency data; thus the average delay of data transmission will increase when the amount of emergency data packets increases. Fortunately, when the percentage of emergent data packets is 50%, the average delay of data transmission is approximate 86ms and this delay is acceptable.

5. **Conclusions.** We proposed a lightweight secure data aggregation protocol for emergency detection in wireless sensor networks. Besides transmitting emergent data to the based station, the proposed scheme can also resist altering, forging, and dropping attacks. The proposed scheme can effectively detect the attacker. We also provide the security analysis and the simulation results to justify that the proposed scheme can resist attacks effectively and efficiently. In the future, we will focus on reducing the overhead of computation cost. In addition, we will aim to provide better security to resist other attacks.

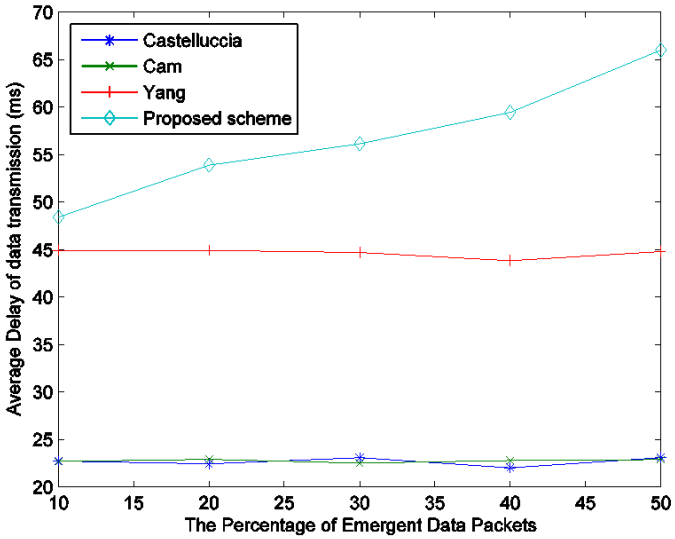

FIGURE 7. The influence of emergency data packets

Acknowledgment. This work was supported in part by the National Science Council, Taiwan, under Contract NSC 100-2218-E-007-006.

REFERENCES

[1] A. Perrig, J. Stankovic and D. Wagner, Security in wireless sensor networks, *Communications of the ACM*, vol.47, pp.53-57, 2004.

[2] D. Estrin, A. Sayeed and M. Srivastava, Wireless sensor networks, *Mobicom 2002 Tutorial*, http://nesl.ee.ucla.edu/tutorials/mobicom02, 2002.

[3] H. Li, Y. Zheng, K. Chen and M. Wen, A hash based secure aggregation protocol for sensor network, *Proc. of the IEEE International Conference on Mechatronics and Automation*, pp.1920-1924, 2006.

[4] S. Roy, S. Setia and S. Jajodia, Attack-resilient hierarchical data aggregation in sensor networks, *Proc. of the 4th ACM workshop on Security of Ad Hoc and Sensor Networks*, pp.71-82, 2006.

[5] C. Castelluccia, E. Mykletun and G. Tsudik, Efficient aggregation of encrypted data in wireless sensor network, *Proc. of the 2nd Annual International Conference on MobiQuitous*, pp.109-117, 2005.

[6] H. Çam, S. Özdemir, P. Nair and D. Muthuavinashiappan, ESPDA: Energy-efficient and secure pattern-based data aggregation for wireless sensor networks, *Proc. of IEEE on Sensors*, vol.2, pp.732-736, 2003.

[7] H. Çam, S. Özdemir, P. Nair, D. Muthuavinashiappan and H. O. Sanli, Energy-efficient secure pattern based data aggregation for wireless sensor networks, *Computer Communications*, vol.29, pp.446-455, 2005.

[8] L. Buttyán, P. Schaffer and I. Vajda, RANBAR: RANSEC-based resilient aggregation in sensor networks, *Proc. of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp.83-90, 2006.

[9] D. Wagner, Resilient aggregation in sensor networks, *Proc. of the ACM Workshop on Security in Ad Hoc and Sensor Networks*, pp.78-87, 2004.

[10] L. Hu and D. Evans, Secure aggregation in sensor network, *Proc. of Workshop on Security and Assurance in Ad Hoc Networks*, pp.384-391, 2003.

[11] Y. Yang, X. Wang, S. Zhu and G. Cao, SDAP: A secure hop-by-hop data aggregation protocol for sensor networks, *Proc. of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp.356-367, 2006.

[12] A. Mahimkar and T. S. Rappaport, SecureDAV: A secure data aggregation and protocol for sensor networks, *Proc. of the IEEE Global Telecommunications Conference*, vol.4, pp.2175-2179, 2004.

[13] K. Wu, D. Dreef, B. Sun and Y. Xia, Secure data aggregation without persistent cryptographic operations in wireless sensor network, *Proc. of the 25th IEEE Performance, Computing, and Communications Conference*, Phoenix, AZ, 2006.
[14] 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 Security*, pp.278-287, 2006.
[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] H. O. Sanli, S. Ozdemir and H. Cam, SRDA: Secure reference-based data aggregation protocol for wireless sensor network, *Proc. of Vehicular Technology Conference*, vol.7, pp.4650-4654, 2004.
[17] M. Raina, S. Ghosh, R. Patro, G. Viswanath and T. Chadrashekhar, Secure data aggregation using commitment schemes and quasi commutative functions, *Proc. of the 1st International Symposium on Wireless Pervasive Computing*, 2006.
[18] G. Horng, C. Wang and T. Chen, An efficient concealed data aggregation scheme for sensor networks based on secret sharing, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(A), pp.3085-3097, 2009.
[19] Y. Jin, J. Jo and Y. Kim, Energy-efficient multi-hop communication scheme in clustered sensor networks, *International Journal of Innovative Computing, Information and Control*, vol.4, no.7, pp.1741-1749, 2008.
[20] A. Perrig, R. Szewczyk, V. Wen, D. Culler and D. Tygar, SPINS: Security protocols for sensor networks, *Wireless Networks Journal*, vol.8, pp.521-534, 2002.
[21] S. Ganeriwal, R. Kumar and M. B. Srivastava, Timing-sync protocol for sensor networks, *Proc. of the 1st International Conference on Embedded Networked Sensor Systems*, pp.138-149, 2003.
[22] H. Chan, A. Perrig and D. Song, Random key predistribution schemes for sensor networks, *Proc. of the IEEE Symposium on Security and Privacy*, pp.197-213, 2003.
[23] S. Zhu, S. Setia and S. Jajodia, Leap: Efficient security mechanisms for large-scale distributed sensor networks, *Proc. of the 10th ACM Conference on Computer and Communications Security*, pp.62-72, 2003.
[24] M. Ding, X. Cheng and G. Xue, Aggregation tree construction in sensor network, *Proc. of the 58th IEEE Vehicular Technology Conference*, vol.4, pp.2168-2172, 2003.
[25] K. Fall and K. Varadhan, The Ns manual (formerly called "ns Notes and Documentation"), *The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC*, http://www.isi.edu/nsnam/ns/, 2007.
[26] CMU Monarch Group, *CMU Monarch Extensions to the NS-2 Simulator*, http://monarch.cs.cmu.edu/cmu-ns.html/, 1998.