Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 509313, 7 pages
http://dx.doi.org/10.1155/2014/509313

Research Article

Flight Protection Data via Dynamic Sensor Networks

Deok Gyu Lee[1] and Jong Wook Han[2]

[1] *Department of Information Security, Seowon University, 377-3 Musimseoro, Heungdeok-gu, Cheongju, Chungbuk 361-742, Republic of Korea*
[2] *Electronic and Telecommunications Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon 305-700, Republic of Korea*

Correspondence should be addressed to Deok Gyu Lee; deokgyulee@gmail.com

Received 1 September 2013; Accepted 24 December 2013; Published 10 February 2014

Academic Editor: Hwa-Young Jeong

In wireless sensor networks, adversaries can compromise sensor nodes and use them to inject forged reports, which can lead to false alarms and energy depletion. Recently, several research solutions have been proposed to detect and drop such forged reports during the forwarding process. Since each of them has its own energy consumption characteristics, employing only a single filtering solution for a network is not a recommended strategy, in terms of energy savings. While a technique for the adaptive selection of filtering solutions has been proposed, it considers only static networks. This paper relates to a system and method for automatically protecting flight data in response to a variety of kinds of cyberterror that paralyze control service in a flight data system by enhancing the availability, reliability, and integrity of the flight data system when damage due to external or internal viruses or hacking, such as the alteration or modification of flight data, occurs. The flight data protection system has an advantage in that it can manage a system safely by providing an embedded system, using an Enhanced Write Filter (EWF), and protecting an operating system.

1. Introduction

The amount of data processed by a flight data processing system, which is the essence of flight control, is huge because a multitude of countries and a variety of kinds of flight data related to the flight data processing system have to be handled. Furthermore, the flight data processing system is used by a number of specific persons, whereby external and internal attacks against the flight data processing system are diverse and increasingly sophisticated, thereby increasing the threat to an application layer, that is, the entire data processing system.

Recently, wireless sensor networks (WSNs) have been applied in many different areas, for instance, the voltage variation monitoring in electric power companies, temperature and humidity remote controlling in museums, and human health tracking systems. Normally, the client device needs to obtain authentication from the system which it wants to access [1]. A sensor network consists of a large number of small, inexpensive, and self-powered devices that can sense, compute, and communicate with other devices. Nodes act as information sources and sense and collect data samples from their environment. Wireless sensor networks (WSNs) have a wide range of civil and military applications. One of the important applications of a WSN is area monitoring, where nodes are deployed over a region to monitor an event or phenomenon. For example, in military applications, to detect intrusion in a battlefield, large quantities of sensor nodes are required along with high security [2].

Each of filtering solutions has its own energy consumption characteristics [3]. For example, the key inheritance-based filtering (KIF) [4] can conserve energy resources for heavy false traffic but consumes too much energy for legitimate traffic. On the other hand, the overhead of the statistical en-route filtering (SEF) [5] is relatively small. However, it does not guarantee that a forged report can always be detected during forwarding. In [3], Lee and Cho pointed out that employing only a single filtering solution for a network is not a recommended strategy, in terms of energy savings, and proposed a method for the adaptive selection of filtering solutions in which a fuzzy rule-based system adaptively chooses among three filtering solutions by considering

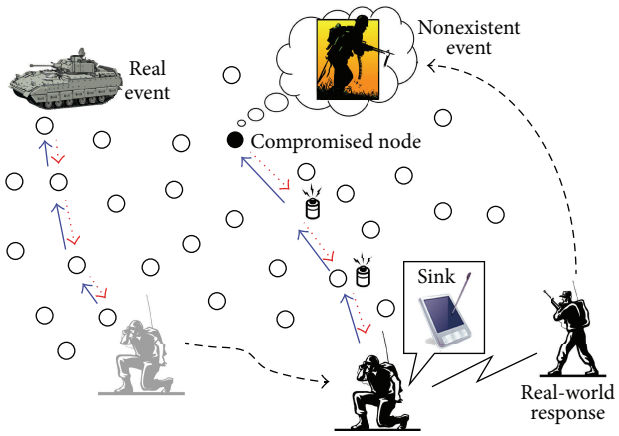

Figure 1: Dynamic WSNs.

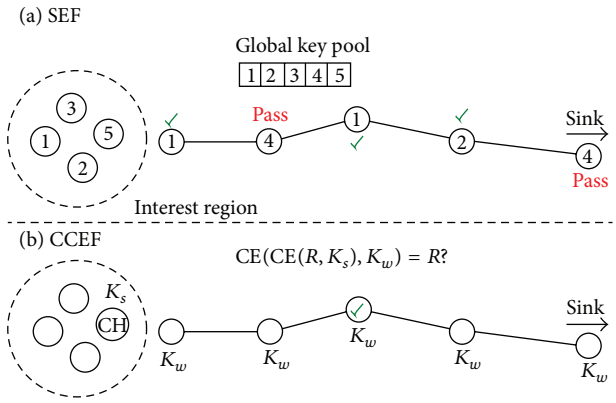

Figure 2: En-route verifications in SEF and CCEF.

network status. Their method assumes that the network is static (i.e., nodes and sinks are fixed). However, in many real-world applications, networks may be dynamic. That is, nodes and sinks may be mobile. For example, soldiers with mobile devices (e.g., PDAs) may query battlefield situations through WSN (Figure 1) [6].

This paper provides flight data protection technology in which a flight data processing system installed in a network can monitor a variety of kinds of hacking and cyberterror in real time and can provide seamless control service by automatically recovering from a disaster when the flight data processing system is forged or modified.

In accordance with an aspect of the present paper, there is an apparatus provided for protecting flight data, including a flight data verification module for classifying original flight data for each field and verifying the classified data in order to protect the flight data against an external attack, a flight data database for storing the verified flight data for each field, a flight data monitoring module for hooking messages for the original flight data input to and output from the flight data verification module and monitoring the hooked messages, a host message monitoring module for generating flight data for recovery in response to a monitoring result message provided by the flight data monitoring module, and a flight data restoration management module for restoring the original flight data by using the generated flight data for recovery.

2. Related Work

2.1. Statistical En-Route Filtering (SEF). SEF [5] is the first paper that addresses false data injection attacks in the presence of compromised nodes [7]. It also presents the general en-route filtering framework, which serves as the base of other filtering solutions [8–14]. SEF can probabilistically detect forged reports. In SEF, sinks maintain a global key pool which is separated into multiple partitions. Every node loads a small number of secret keys from a randomly selected partition in the global key pool, before it is deployed. When an event occurs, one of the detecting nodes collects MACs of the event from other detecting nodes. Then, it produces a sensing report and forwards the report to a sink. A report is forwarded if and only if it has multiple MACs generated by multiple nodes, using secret keys from different partitions in the global key pool. Figure 2(a) shows the essential process of the en-route verification in SEF. The overhead of SEF is relatively small. However, it does not guarantee that a forged report can always be detected during forwarding. Thus, it is not suitable for massive false data injection attacks [3].

2.2. Commutative Cipher-Based En-Route Filtering (CCEF). CCEF [15] was proposed to defend against false data injection attacks without symmetric key sharing among sensor nodes. To endorse and verify sensing reports, CCEF uses a commutative cipher CE, which satisfies the following property: for any message M and any two keys K_1 and K_2,

$$\mathrm{CE}\left(\mathrm{CE}\left(M, K_1\right), K_2\right) = \mathrm{CE}\left(\mathrm{CE}\left(M, K_2\right), K_1\right). \quad (1)$$

In CCEF, every node is preloaded with an authentication key, which is shared only with sinks. For each session, a sink prepares two keys K_s (a session key) and K_w (a witness key) that satisfy.

$$\mathrm{CE}\left(\mathrm{CE}\left(M, K_S\right), K_W\right) = M. \quad (2)$$

Then, it sends a query to a cluster head (CH) at the location of interest. K_S encrypted by the CH's authentication key and K_W as plaintext are included in the query. Each intermediate node stores K_W for the purpose of future verification. A sensing report is generated by CH. The report is endorsed with a MAC produced by CH using K_s (i.e., $\mathrm{CE}(M, K_S)$) and multiple MACs produced by its neighbor nodes using their authentication keys. Then the report is forwarded to the sink along the reversed path as the query traverses. Each forwarding node verifies the report by (2) with a certain probability CVP. Figure 2(b) shows the essential process of the en-route verification in CCEF. The detection power of CCEF can be controlled by adjusting CVP. For example, a large CVP (e.g., 1.0) can provide the early detection of forged reports. On the other hand, the computation overhead can be reduced with a small CVP. However, usually, the computation overhead of CCEF is even heavier than that of SEF.

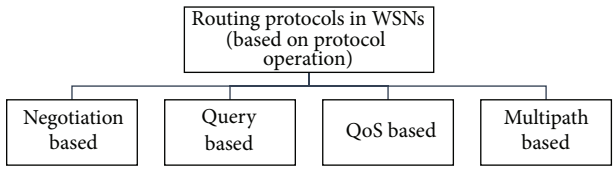

Figure 3: Classification of routing protocols based on protocol operation in the WSN.

2.3. Low-Energy Adaptive Clustering Hierarchy (LEACH). LEACH is a hierarchical routing algorithm for sensor networks. It is aimed at making energy consumption in each node uniform by selecting CH (cluster heads) for the next round based on the function $P_t(t)$, which indicates the possibility of becoming the next CH. The function $P_t(t)$ is calculated in each node of a cluster at the start of each round. This function is also selected in such a way that the expected number of CH nodes for a round remains a constant k [16]

$$E\,[\#\mathrm{CH}] = \sum_{i=1}^{N} P_i\,(t) \times 1 = k. \tag{3}$$

2.4. The Taxonomy of Routing Protocols. Many routing solutions that have been specifically designed for WSNs have been proposed [17]. In these possible, the unique properties of the WSNs have been taken into account. These routing techniques can be classified according to the protocol operation as *negotiation based, query based, QoS based,* and *multipath based*, as depicted in Figure 3.

The negotiation based protocols have the objective to eliminate the redundant data by including high level data descriptors in the message exchange.

In query based protocols, the sink node initiates the communication by broadcasting a query for data over the network.

The multipath based protocols were initiated with objectives to provide reliability and to balance the traffic load in the network. These protocols use multipath in order to achieve better energy efficiency and network robustness in case of node failures. Multipath routing protocols have been discussed in WSN literature for several years now.

QoS based protocols allow sensor nodes to balance between the energy consumption and certain predetermined QoS metrics, such as delay, energy, reliability, bandwidth, and so forth, before they deliver the data to the sink node.

3. Proposed Scheme

Figure 4 is a block diagram of a flight data protection system including an apparatus for protecting flight data in accordance with an embodiment of the present paper. The flight data protection system can include a flight data verification module, an embedded system module, a sensor module, a flight data dictionary module, a flight data restoration management module, a host message monitoring module, a flight data database (DB), and a flight data monitoring module.

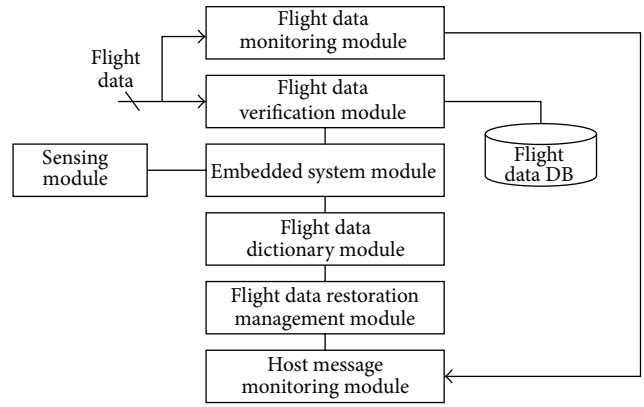

Figure 4: Block diagram of flight data protection system on sensor networks.

The flight data verification module can function to classify the original flight data into fields and verify the classified flight data in order to protect the original flight data against malicious attacks. For example, the flight data verification module can divide received flight data into data that must be secured and data that do not need to be secured, subdivide the data that must be secured for each field, and verify the subdivided data. The flight data verified by the flight data verification module can be classified into fields and stored in the flight data DB.

The embedded system module stores a proper embedded system so that the flight data can be classified into fields and combined by the flight data verification module. The flight data dictionary module can store the original flight data and reproduce the stored original flight data. The flight data restoration management module can function to restore the original flight data depending on the situation and can store an event log. The host message monitoring module can provide the flight data restoration management module with flight data for recovery in response to a monitoring result message provided by the flight data monitoring module. The host message monitoring module can become a mediator for the transmission of the flight data for recovery depending on the situation and for connection with the flight data monitoring module. The flight data DB can store the flight data, which has been classified into fields and provided by the flight data verification module.

The flight data monitoring module can monitor the original flight data using a filter driver. The flight data protection apparatus sends a monitoring result message over an external network and becomes a mediator for the transmission of flight data for recovery depending on the situation and for connection with the daemon and agent of the system (alternatively called a host).

From among the elements of the flight data protection apparatus, the embedded system module, the flight data DB, an OS depository module (not shown), and the flight data dictionary module protect the system by means of write prevention using an Enhanced Write Filter (EWF), thereby being capable of increasing the availability, reliability, and integrity of the system.

The sensor module can provide the flight data collection management module with flight data for recovery in response to a monitoring result message provided by the embedded system module. For each session, a sink randomly selects one sensor node at the location of interest as CH. It then evaluates the fitness of the filtering solutions and determines CVP with the fuzzy rule-based system. The sink constructs a query that includes not only the application-specific interests, but also some additional fields. If SEF is chosen, the query additionally includes (1) a query ID; (2) the ID of CH; (3) the ID of SEF. If CCEF is chosen, the query additionally includes (1) a query ID; (2) the ID of CH; (3) the ID of CCEF; (4) CVP; (5) a session key encrypted by CH's authentication key; (6) a witness key as plaintext. The query is authenticated by an authentication technique such as μTESLA [18]. The sink sends the query, which is forwarded by hop-by-hop to CH. Every intermediate node stores the query ID and the ID of chosen filtering solution. If the ID of solution is equal to that of CCEF, it additionally stores CVP and the witness key for the purpose of further verification.

CH responds to the query by collaborative generation of a sensing report. The report should contain multiple MACs generated by multiple nodes, using secret keys from different partitions in the global key pool if SEF is the current filtering solution. If CCEF is the current, the report should be endorsed with a MAC produced by CH using the session key and multiple MACs produced by its neighbor nodes using their authentication keys should be attached to the report. CH forwards the report to the sink. Every forwarding node verifies the report based on the current filtering solution. In SEF, the node verifies the report if it has any of the secret keys used to generate the MACs. In CCEF, the node verifies the report using the witness key with a probability CVP. The report is dropped if verification of a report fails. The report is finally verified by the sink.

This paper presents a fuzzy-based filtering solution selection method (FSS) for dynamic WSNs. A fuzzy rule-based system is exploited to choose the most energy-efficient solution, between SEF and the commutative cipher-based filtering (CCEF). The ratio of false traffic in the network (FTR), the distance between a sink and the source node as a hop count (DTC), and the detection power of SEF (SDP) are used to evaluate the two filtering solutions. The fuzzy system simultaneously controls the detection power of CCEF by determining the verification probability of CCEF (CVP), to achieve further energy savings.

3.1. Flight Data Network Model and Assumptions. We consider a WSN composed of a large number of small sensor nodes. The network users query the network through mobile sinks such as aviation flights. We assume that the topology of the network does not change during each query-response session. That is, during the session, sinks and nodes do not move or move within a limited range. However, in real-world WSNs, the topology may change during a session. Some mechanism to recover corrupt sessions may be required. Such recovering behavior usually needs the redistribution of keys. Thus, in real-world applications, SEF and CCEF can be more energy efficient than IHA and KIF due to their small overhead for key redistribution (refer to Table 1).

Table 1: Key redistribution overhead in filtering solutions.

Solution	Key redistribution needed	Redistribution overhead/node
SEF	Never	—
CCEF	Always (K_w)	O(1)
IHA	When topology changes	O(t)
KIF	When topology changes	O(t^2)

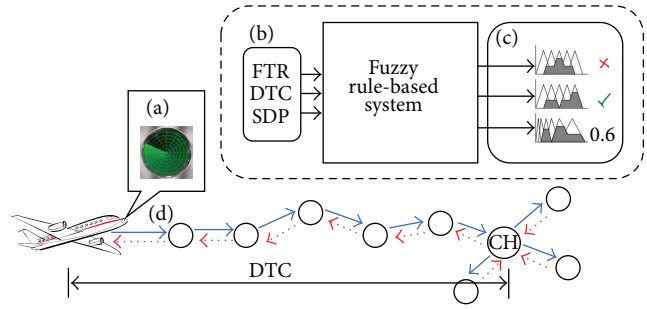

Figure 5: FSS overview.

Sensor nodes are similar to the current generation of sensor nodes, such as MICAz [19], in their computational and communication capability and power resources. We assume that every node has several hundred bytes of memory for keying materials used in SEF and CCEF. Before node deployment, each node may load 25~100 secret keys and 1 authentication key for SEF and CCEF, respectively. Sensor nodes may be compromised or physically captured. Once compromised, a node can be used to inject forged reports into the network. However, we assume that sinks cannot be compromised.

We assume that sinks can know all authentication keys (used in CCEF) of the nodes within the region of interest. Note that mobile sinks may have no need to store all authentication keys of the entire network. They can obtain authentication keys of the nodes in the interest area from some fixed and powerful base stations via direct communication. We further assume that sinks can know or estimate FTR. To achieve this, we can deploy tamper-resistant nodes that only record the number of dropped reports and report this number to sink nodes [3]. Sinks also know or can estimate DTC and SDP. We further assume that sinks have a mechanism to authenticate broadcast messages (e.g., based on μTESLA [18]), and every node can verify the broadcast messages.

3.2. Overview. Mobile sinks initiate query-response sessions. For each session, a sink randomly selects one sensor node at the location of interest as CH (Figure 5(a)). It then evaluates the fitness of the two filtering solutions—SEF and CCEF—with a fuzzy rule-based system. The fitness results lead to a choice of filtering solutions for the session. The fuzzy system simultaneously determines CVP (Figure 5(c)). For the fuzzy inference, FTR, SDP, and DTC are used (Figure 5(b)). The sink constructs a query and sends it to CH (Figure 5(d)). CVP

is included in the query. If CCEF is chosen for the session, it is stored by intermediate nodes. CH responds to the query by generating and endorsing a sensing report. The report is forwarded along the reversed path, as the query traverses. The report is verified by forwarding nodes, based on the chosen filtering solution. If SEF is chosen, a node only verifies the report if it has any of the secret keys that were used to generate the attached MACs. If CCEF is chosen, a forwarding node verifies the report with CVP.

3.3. Factors That Affect the Solution Selection. FTR has the greatest effect on the energy consumption characteristics of filtering solutions [3]. If false traffic utilizes a very small proportion of the total, most reports can be delivered to sinks. Thus, between SEF and CCEF, SEF is more efficient, in terms of energy saving, since SEF consumes less energy than CCEF, during verification and forwarding. On the other hand, CCEF can be very energy efficient for false traffic, if CVP is large. For example, if CVP is set to 1.0, a forged report can be detected and dropped at the very next hop node of the compromised node that injected the forged report, before it consumes a significant amount of energy. On the other hand, in SEF, a forged report can be forwarded a significant number of hops, or be delivered to a sink, before it is detected. Therefore, in order to save energy resources, we have to choose one of the filtering solutions based on FTR.

DTC also affects the energy consumption characteristics of the solutions. In SEF, a long-range delivery may cause a duplicate verification problem. That is, a MAC attached to a report may be verified by two or more forwarding nodes. Thus, an additional computation overhead may arise. On the other hand, in CCEF, we can prevent the duplicate verification problem by adjusting CVP. For example, theoretically, a report may be verified by only one node if CVP is set to 1/DTC. Therefore, we have to choose one of the filtering solutions based on DTC.

We can control the detection power of CCEF by adjusting CVP. A large CVP increases the detection power. On the other hand, SDP is predominantly determined before node deployment and gradually decreases with the compromising of nodes. If SDP is enough to detect forged reports at an early stage, SEF may be a good choice in terms of energy savings. If several partitions are compromised, choosing CCEF may result in energy efficiency. Therefore, we have to choose one of the filtering solutions based on SDP. SDP can be determined by

$$\text{sdp} = \frac{k}{m} \cdot \frac{(t-c)}{n}, \tag{4}$$

where m is the number of secret keys in each partition of a global key pool, k is the number of secret keys assigned to each node, n is the number of partitions in the key pool, t is the security threshold value (which determines the number of MACs carried in each report), and c is the number of compromised partitions.

In dynamic WSNs, sinks can move. Nevertheless, the mobility factors of sinks (e.g., speed or direction) are not considered in FSS since we assume that the topology of the network does not change during each query-response session. That is, the network is static during the session. Therefore, such mobility factors have less impact than the above three factors.

3.4. Fuzzy-Based Filtering Solution Selection. In FSS, a fuzzy rule-based system is exploited to evaluate the fitness of each filtering solution and to determine CVP. One aspect of the appeal of fuzzy rule-based systems is that they can be used for approximate reasoning, which is particularly important when there is uncertainty in reasoning, in addition to imprecision in data [20]. In the fuzzy reasoning, FTR, which has the greatest effect on the energy consumption characteristics of the filtering solution, is the primary source of uncertainty. In real-world WSNs, it may be impractical to measure FTR precisely since sensor nodes may malfunction [21]. For example, a legitimate report may be regarded as false traffic if any forwarding node fails to forward the report. Moreover, SDP can be a source of uncertainty since it is not easy to obtain c in (3). Therefore, approximate reasoning is needed, to handle such fuzzy information [22].

3.5. Query and Response. For each session, a sink randomly selects one sensor node at the location of interest as CH. It then evaluates the fitness of the filtering solutions and determines CVP with the fuzzy rule-based system. The sink constructs a query that includes not only the application-specific interests, but also some additional fields. If SEF is chosen, the query additionally includes (1) a query ID; (2) the ID of CH; (3) the ID of SEF. If CCEF is chosen, the query additionally includes (1) a query ID; (2) the ID of CH; (3) the ID of CCEF; (4) CVP; (5) a session key encrypted by CH's authentication key; (6) a witness key as plaintext. The query is authenticated by an authentication technique such as μTESLA [18]. The sink sends the query, which is forwarded by hop-by-hop to CH. Every intermediate node stores the query ID and the ID of chosen filtering solution. If the ID of solution is equal to that of CCEF, it additionally stores CVP and the witness key for the purpose of further verification.

CH responds to the query by collaborative generation of a sensing report. The report should contain multiple MACs generated by multiple nodes, using secret keys from different partitions in the global key pool if SEF is the current filtering solution. If CCEF is the current, the report should be endorsed with a MAC produced by CH using the session key and multiple MACs produced by its neighbor nodes using their authentication keys should be attached to the report. CH forwards the report to the sink. Every forwarding node verifies the report based on the current filtering solution. In SEF, the node verifies the report if it has any of the secret keys used to generate the MACs. In CCEF, the node verifies the report using the witness key with a probability CVP. The report is dropped if verification of a report fails. The report is finally verified by the sink.

4. Simulation Results

To show the effectiveness of FSS, we compare FSS with SEF, CCEF, and ASFS through the simulation. We use a field size

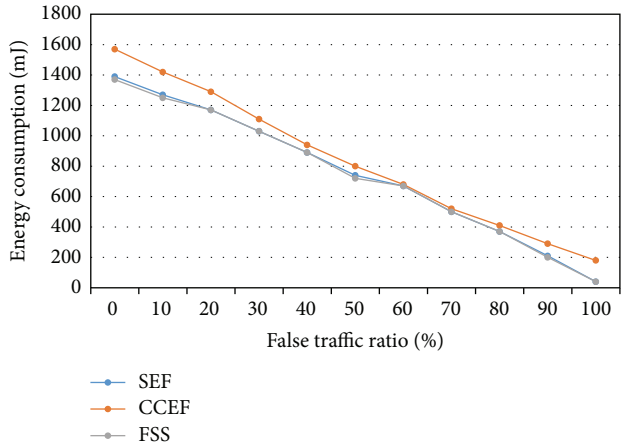

Figure 6: Average energy consumption per report (SDP = 0.025).

of $500 \times 30\ m^2$, where 1,500 nodes are uniformly distributed. Sink nodes randomly move within the field. The motion speed of sinks is set to 10 meters per second with a random way point mobile model [22]. Each node takes 16.25, 12.5 μJ to transmit/receive a byte [7]. Each MAC generation consumes 15 μJ and one commutative cipher computation consumes 9 mJ [19]. The size of an original report is 24 bytes. The size of a MAC is 1 byte. We use a global key pool of 1,000 keys for SEF.

Figure 6 shows the average energy consumption per report delivery when $0 \leq \text{FTR} \leq 100$ and SDP = 0.025. As shown in the figure, SEF (empty rectangles) consumes relatively less energy up to about 60 percent false traffic. On the other hand, CCEF with CVP = 1.0 (empty circles) is energy efficient if FTR exceeds about 60 percent of the total. Since FSS (filled triangles) adaptively chooses the more energy-efficient solution (SEF or CCEF) according to FTR, it can conserve energy.

5. Conclusion

In this paper, we proposed FSS for dynamic WSNs. A fuzzy rule-based system is exploited to choose the most energy-efficient solution, between SEF and CCEF, and to determine CVP that can conserve energy. This paper relates to flight data protection technology and, more particularly, to a flight data protecting apparatus and method for detecting the forgery and alternation of flight data and automatically providing disaster recovery in response to a variety of kinds of cyberterror that interfere with the operation of a flight data processing system by increasing the availability, reliability, and integrity of a system in providing control service, such as the transmission of the flight data internally or externally.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by a Grant (code no. 07aviation-navigation-03) from Aviation Improvement Program funded by Ministry of Construction & Transportation of Korean government.

References

[1] P. Kumar, A. Gurtov, M. Ylianttila, S. -G. Lee, and H. Lee, "A strong authentication scheme with user privacy for wireless sensor networks," *ETRI Journal*, vol. 35, no. 5, pp. 889–899, 2013.

[2] K. Rajendiran, R. Sankararajan, and R. Palaniappan, "A secure key predistribution scheme for WSN using elliptic curve cryptography," *ETRI Journal*, vol. 33, no. 5, pp. 791–801, 2011.

[3] H. Y. Lee and T. H. Cho, "Fuzzy adaptive selection of filtering schemes for energy saving in sensor networks," *IEICE Transactions on Communications*, vol. 90, no. 12, pp. 3346–3353, 2007.

[4] H. Y. Lee and T. H. Cho, "Key inheritance-based false data filtering scheme in wireless sensor networks," *Lecture Notes in Computer Science*, vol. 4317, pp. 116–127, 2006.

[5] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 839–850, 2005.

[6] H. S. Seo, H. Y. Lee, S. J. Lee, and D. G. Lee, "Fuzzy-based filtering solution selection method for dynamic sensor networks," *Intelligent Automation and Soft Computing*, vol. 16, no. 4, pp. 579–592, 2010.

[7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[8] S. Li and D. Zhang, "A novel manifold learning algorithm for localization estimation in wireless sensor networks," *IEICE Transactions on Communications*, vol. 90, no. 12, pp. 3496–3500, 2007.

[9] B. Przydatek, D. Song, and A. Perrig, "SIA: secure information aggregation in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 255–265, November 2003.

[10] Q. Zhang, T. Yu, and P. Ning, "A framework for identifying compromised nodes in sensor networks," in *Proceedings of the Securecomm and Workshops*, pp. 1–10, September 2006.

[11] W. Du, L. Fang, and N. Peng, "LAD: localization anomaly detection for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 66, no. 7, pp. 874–886, 2006.

[12] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 259–271, May 2004.

[13] F. Li and J. Wu, "A probabilistic voting-based filtering scheme in wireless sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '06)*, pp. 27–32, July 2006.

[14] H. Yang and S. Lu, "Commutative cipher based En-route filtering in wireless sensor networks," in *Proceedings of the IEEE 60th Vehicular Technology Conference (VTC-Fall '04)*, pp. 1223–1227, September 2004.

[15] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward resilient security in wireless sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 34–45, May 2005.

[16] H.-R. Lee, K.-Y. Chung, and K.-S. Jhang, "A study of wireless sensor network routing protocols for maintenance access hatch condition surveillance," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 237–246, 2013.

[17] R. Sumathi and M. G. Srinivas, "A survey of QoS based routing protocols for wireless sensor networks," *Journal of Information Processing Systems*, vol. 8, no. 4, pp. 589–602, 2012.

[18] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[19] http://www.xbow.com/.

[20] N. Serrano and H. Seraji, "Landing site selection using fuzzy rule-based reasoning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 4899–4904, April 2007.

[21] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 255–265, August 2000.

[22] Y.-K. Kwok and L.-S. Cheung, "A new fuzzy-decision based load balancing system for distributed object computing," *Journal of Parallel and Distributed Computing*, vol. 64, no. 2, pp. 238–253, 2004.