Kalpa Publications in Computing
Volume 2, 2017, Pages 196–204
ICRISET2017. International Conference on Research and Innovations in Science, Engineering &Technology. Selected Papers in Computing

Modeling and Simulation of Vote Length Analysis for Probabilistic Voting-based Filtering in Wireless Sensor Networks: Against False report and vote injection attacks

Su Man Nam[1*] and Tae Ho Cho[2†]
[1]College of Information and Communication Engineering
Sungkyunkwan University, Suwon 16419, Republic of Korea
[2]College of Software Sungkyunkwan University Suwon 16419, Republic of Korea
sm38good@skku.edu, thcho@skku.edu

Abstract

In large-scale wireless sensor networks, sensors are vulnerable to false report and false vote injection attacks since they are deployed in hostile environments. These attacks drain their limited energy resources of forwarding nodes and drops important data. Probabilistic voting-based filtering scheme simultaneously detects both the attacks through vote verification. To effectively detect them, it is important to define the vote length of the reports since the vote length is fixed at the initial phase. We find the effective vote length using a simulation model since it is nearly impossible to evaluate the security protocol performance on the real nodes. We demonstrate that the security protocol, in which the vote length is five, achieves better detection ratio against the two attacks.

1 Introduction

Wireless sensor networks (WSNs) are lately feasible machineries for a variety of applications without infrastructures (K.Younis & M.Younis, 2005). A sensor network comprises a number of sensor nodes (SNs) and a base station (BS) (K.Younis & M.Younis, 2005). The SNs function sensing, computing, and wireless communication. The BS collects the event information. However, since the WSN uses wireless communication, malicious attackers can easily inject various attacks using compromised nodes (CNs) to acquire sensitive data and destroy the network.

[*] First author
[†] Corresponding author

R. Buyya, R. Ranjan, S. Dutta Roy, M. Raval, M. Zaveri, H. Patel, A. Ganatra, D.G. Thakore, T.A. Desai, Z.H. Shah, N.M. Patel, M.E. Shimpi, R.B. Gandhi, J.M. Rathod, B.C. Goradiya, M.S. Holia and D.K. Patel (eds.), ICRISET2017 (Kalpa Publications in Computing, vol. 2), pp. 196–204

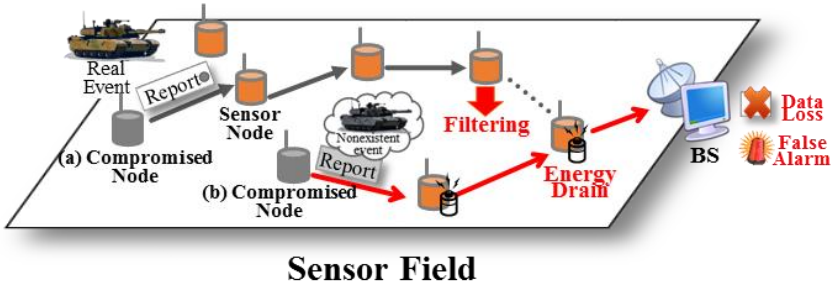

Figure 1: False vote and false report injection attacks

Figure 1 shows two attacks of application layers of the sensor network. In a false vote injection attack (FVIA) (F.Li, A.Srinivasan, J.Wu, 2008), a CN injects false votes on a legitimate report after sensing a real event. The legitimate report is intentionally dropped in an en-route node by the false vote, even though the report has correct data. In a false report injection attack (FRIA) (F.Li, A.Srinivasan, J.Wu, 2008) (F.Ye, H.Luo., S.Lu, L.Zhang,, 2005), a CN injects a false report about a non-existent event and forwards it to the BS. The false report drains en-route nodes' energy and generates a false alarm in the BS.

F. Li *et al*. proposed a probabilistic voting-based filtering scheme (PVFS) (F.Li, A.Srinivasan, J.Wu, 2008) to simultaneously detect the attacks through vote verification of reports in verification nodes. The PVFS makes a division of the two attacks through the threshold (TH) verified false votes in a report. It is important to define the vote length in a report to effectively detect the attacks. In (F.Li, A.Srinivasan, J.Wu, 2008) (F.Ye, H.Luo., S.Lu, L.Zhang,, 2005), the vote length is fixed in a report at the initiatory phase.

In this paper, we analyze the performance of the sensor network for finding the effective vote length through a simulation model proposed in (NamS.M. & ChoT.H., 2016). We evaluate the network efficiency because the number of votes affects the network performances. Experimental results show the energy consumption and the ratio of the detected attacks according to the length.

The remainder of the paper is organized as follows: Section 2 explains a security protocol, DEVS, and motivation. Section 3 shows a simulation model and attacks analysis. Section 4 presents a performance evaluation. We conclude this paper in Section 5.

2 Background

2.1 PVFS

In (F.Li, A.Srinivasan, J.Wu, 2008), a BS generates a key pool and distributes keys to all SNs from the key pool. After deploying the nodes in a sensor field, they form clusters, each of which comprises a cluster head (CH) and multiple SNs. Each CH calculates its hop count between itself and the BS. The CH probabilistically chooses its verification nodes among its en-route nodes based on distance. The CH forwards keys of the source area to the verification nodes so that the nodes verify reports' votes. When an event is generated, the CH forwards an event data to its SNs. As the data is normal, the SN produces a vote by the data and transmits it to the CH; while when the data is an error, the SN discards it. After collecting all the votes from its SNs, it chooses a fixed number of votes and attaches them to a report. The source CH then forwards the report to its next forwarding CH. In forwarding a report toward the BS, each verification nodes confirm the votes of the report. The report

is forwarded to the forwarding node as the verification is normal. As the BS receives the report, it verifies all of the votes using the key pool.

While generating a report, if a CH fabricates a report about a non-existent event, it generates multiple false votes using some of the captured keys and attaches them to the report. A verification node receives the report and authenticates that the vote matches an index among its verification keys. If the verification node detects a fabricated vote, it increases the quantity of detected false votes by one and forwards the report to the next hop node. Should the next verification node detect another false vote in the report, it also increases the number by one. If the quantity of detected false votes reaches a predefined TH, the report is immediately dropped to protect against a FRIA (in this case we assume that TH is 2). In contrast, when a SN is compromised in a source cluster, a legitimate report can be forwarded with a fabricated vote. Because the number of verified false votes does not reach the TH, the report continues to be forwarded to protect against a FVIA. Thus, the PVFS achieves simultaneous detection of injected false data against FVIA and FRIA in a sensor network.

2.2 DEVS formalism

Zeigler developed the DEVS (Discrete EVent System specification) formalism (B.P.Zeigler, DEVS theory of quantized systems, 1998) (B.P.Zeigler, Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems, 1990), which is a theoretical, well-grounded means of expressing hierarchical and modular models to evaluate real world systems. Moreover, the formalism has three merits: model reusability, expandability, and availability. The DEVS formalism defines two models: atomic models and coupled models. An atomic model has inputs, states, outputs, functions, and a time base. The functions of the system decide the next states and outputs through the input and current states. An atomic model is defined as follows:

$$\mathrm{M} = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle.$$

- X: an external input set
- S: a sequential state set
- Y: an external output set
- δ_{int}: an internal transition function
- δ_{ext}: an external transition function
- λ: an output function
- t_a: a time advance function

A coupled model contains several atomic models and the other coupled models to build a larger coupled model. A coupled model is defined as follows:

$$\mathrm{DN} = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select \rangle,$$

- D: a set of component names
- M_i: a component of the basic model
- I_i: a set of influences of I
- $Z_{i,j}$: an output translation

- select: a tie-breaking function

2.3 Motivation

In a sensor network, it is nearly impossible to evaluate the countermeasure performance after deploying sensors because the nodes have limited resources. We evaluate the performance of the PVFS-based virtual WSN using a simulation model proposed in (NamS.M. & ChoT.H., 2016) to measure the network performance according to vote length.

3 Simulation

In this paper, to compare the network performance according to the vote length in the PVFS, we use the simulation model proposed in as shown in Figure 2.

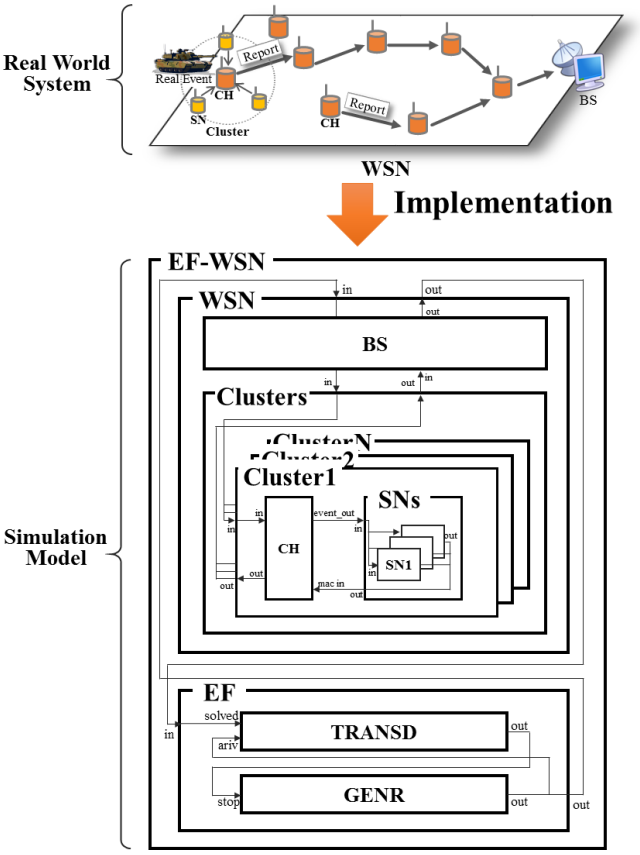

Figure 2: Simulation model implementation through a real world system

In Figure 2, the upper case letters are DEVS models, and lower case letters are ports of the models. For the simulation evaluation, the proposed model EF-WSN includes a WSN and an experimental frame EF. The WSN consists of BS and CLUSTERS; EF consists of a generator GENR and a

transducer TRANSD. In the WSN model, as an event occurs, CH generates a report and forwards it toward BS. In EF, GENR randomly generates events, and TRANSD measures ARCH processing results.

3.1 Model analysis

In (NamS.M. & ChoT.H., 2016), the WSN model has three atomic models for representing the system behavior: CH, SN, and BS.

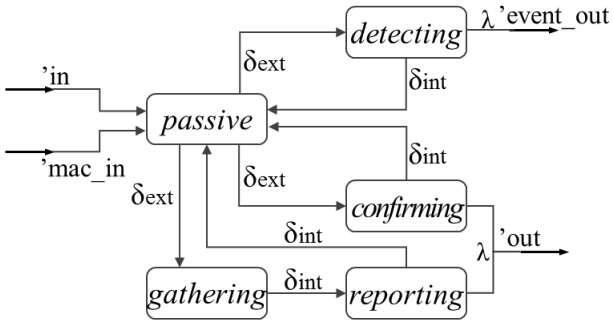

Figure 3: CH's state transition diagram

Figure 3 presents a CH model's state transition diagram. The CH model owns five essential states. This model executes three types of the state transitions: 1) *passive* → *detecting*, 2) *passive* → *gathering* → *reporting*, and 3) *passive* → *confirming*. Initially, the model is at *passive* state. *passive* state represents that model is ready to process a data (e.g., event, MAC, report). When an event comes into 'in port, the model goes into the *detecting* state. This state transition is done by the external transition function. At end of the detecting state, an event data is generated and forwarded to SN models through 'event_out port. When a MAC comes into 'mac_in port, the CH model goes into *gathering* state and collects MACs from the SN models within a time of this state. After the time, transition from *gathering* to *reporting* state occurs. *reporting* state represents the generation of a report including *s* collected MACs. The CH model then transmits the report through 'out port. As a report comes into 'in port, the CH model goes into *confirming* state and verifies the report at this state. The model forwards the report to another CH through 'out port and transfers the initial state *passive* by the internal transition function.

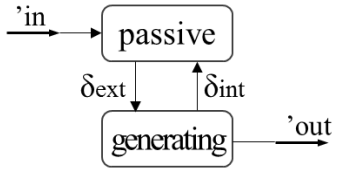

Figure 4: SN's state transition diagram

Figure 4 illustrates a SN model's state transition diagram. This model has two essential states. At first, the model is at *passive* state. When an event data comes into 'in port, the model goes into

generating state. At this state, the model generates a MAC through the event data. It then transmits the MAC to its CH model through 'out port and transfers the initial state.

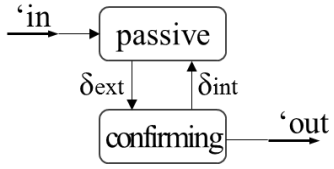

Figure 5: BS's state transition diagram

Figure 5 demonstrations a BS's state transition diagram. This model includes two essential states. Initially, the model is at *passive* state. When an event data comes into 'in port, the model goes into *confirming* state. This *confirming* state executes the verification of all reports. Then, a result of the verification is transmitted through 'out port and the initial state.

3.2 Attack analysis

In (F.Li, A.Srinivasan, J.Wu, 2008), to effectively detect FVIA and FRIA, no performance analysis was showed according to the vote length.

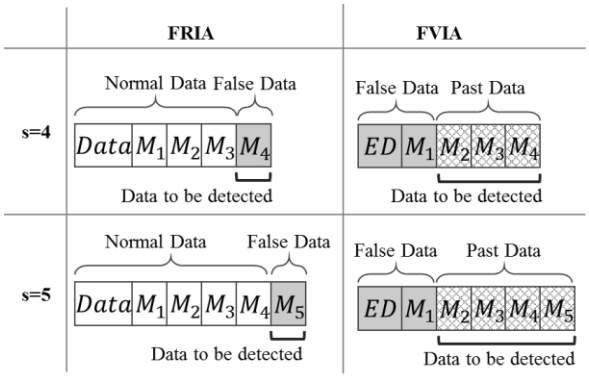

Figure 6: Attack format for FRIA and FVIA

Figure 6 shows attack format for the FVIA and the FRIA according to the vote length (s is a required number of votes for a legitimate report). In FVIA, a normal source node generates a true report including a false vote, which is collected from a CN during report generation phase of the PVFS. The report is continually forwarded because the verified quantity of the false votes reaches the TH yet. In the FRIA, a CN injects a false report using a false data, false votes generated by captured keys, and previous votes received in the past. When the fabricated report is verified, the false vote is normal through the false event data; while the past vote is an error. When the verified number approaches two, the false report is dropped.

The detection ratio of these attacks is a distinction according to the vote length. Even so, as a report has long vote length, en-route nodes consume many amounts of energy resource for vote verification. Therefore, the PVFS needs suitable vote length of a report according to the environment of the sensor network.

4 Simulation Results

A simulation was executed to analyze the security protocol PVFS with vote length 4 and 5 using DEVS. A sensor field was 1,000 × 1,000 m^2, included 1,000 SNs (100 CHs and 900 SNs), and organized 100 clusters that a cluster consists of a CH and nine nodes. The initial energies of CH and SN were 2 J and 1 J, respectively. Each node consumes as follows (F.Ye, H.Luo., S.Lu, L.Zhang,, 2005):

- Transmission: 16.25 µJ per byte
- Reception: 12.5 µJ per byte
- Vote generation: 15 µJ per byte
- Verification: 75 µJ per a vote

In this simulation, 500 events were randomly produced in the field. We set two clusters for generating the FRIA and FVIA.

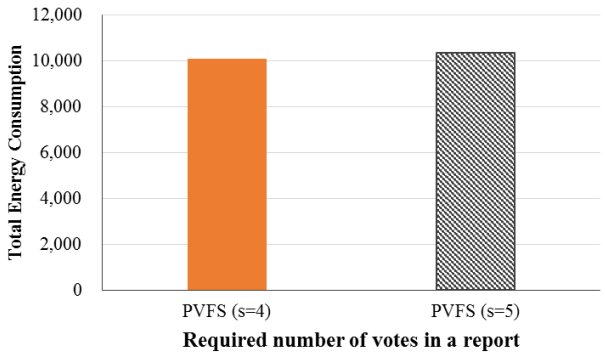

Figure 7: Required quantity of votes in a report versus total energy consumption

Figure 7 shows a required quantity of votes in a report versus total energy consumption. As shown in Figure 7, that do not affect the energy consumption of the whole network according to the vote length.

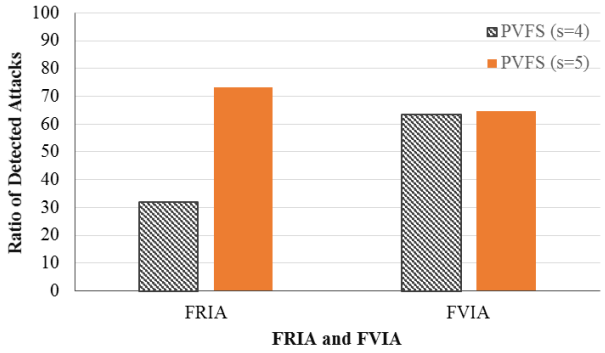

Figure 8: FRIA and FVIA versus ratio of detection attacks

Figure 8 represents the FRIA and FVIA versus the ratio of detection attacks. As shown in Figure 8, there is a big difference between the four and five of the vote length because the five vote length increases the attack detection. On the other hand, there is no difference for the FVIA. Therefore, the vote length affects the FRIA detection as shown in the simulation result.

5 Conclusions

The simulation model for the PVFS in the sensor network analyzes the network performance according to vote length of a report against the FRIA and the FVIA. The PVFS, which the vote length is five, improves the detection of the two attacks.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. NRF-2015R1D1A1A01059484).

References

B.P., Z. (1990). *Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems.* Academic press.

B.P., Z. (1998). *DEVS theory of quantized systems.* Advanced Simulation Technology Thrust DARPA Contract.

F., L., A., S., & J., W. (2008). PVFS: A Probabilistic Voting-based Filtering Scheme in Wireless Sensor Networks. *International Journal of Security and Network*, 173-182.

F., Y., H., L., S., L., & L., Z. (2005). Statistical en-route filtering of injected false data in sensor networks. *Selected Areas in Communications, IEEE Journal On*, 839-850.

K., Y., & M., Y. (2005). A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 325-349.

Nam, S., & Cho, T. (2016). Modeling and Simulation of Threshold Analysis for PVFS in Wireless Sensor Networks. *International Journal of Research -GRANTHAALAYAH (IJRG)*, 1-10.