Vol. 1, No. 1 | Jan – June 2017

Enhancing the Statistical Filtering Scheme to Detect False Negative Attacks in Sensor Networks

Muhammad Akram, Muhammad Ashraf,
College of Information and Communication Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea
akram.khan@skku.edu, ashraf84@skku.edu

Tae Ho Cho
College of Software, Sungkyunkwan University, Suwon 16419, Republic of Korea
thcho@skku.edu

Abstract

In this paper, we present a technique that detects both false positive and false negative attacks in statistical filtering-based wireless sensor networks. In statistical filtering scheme, legitimate reports are repeatedly verified en route before they reach the base station, which causes heavy energy consumption. While the original statistical filtering scheme detects only false reports, our proposed method promises to detect both attacks.

Keywords: WSNs; SEF; en route filtering; false positive attack; false negative attack; energy efficiency.

1. Introduction

Wireless Sensor Networks (WSNs) comprise tiny nodes equipped with restricted computational resources and limited energy supply. WSNs are usually deployed in an exposed environment which increases their proneness to security compromises such as cryptographic information capture [1]. Compromised nodes are exploited by attackers to initiate numerous attacks, such as denial of service, sinkhole attack, and eavesdropping [2]. Usually, attackers use compromised nodes to create bogus event reports, and inject them into the network to drain the energy of the network [1, 2]. Various filtering schemes have been proposed to detect and filter these bogus reports en route [1-5].

Compromised sensor nodes can also be exploited to block authentic data from being delivered to the base station (BS), by attaching false Message authentication codes (MACs) to legitimate reports [1, 2, 6]. These true reports with false MACs attached to them are dropped en route at the intermediate verification nodes. PVFS counters these two attacks simultaneously, whereas other filtering schemes only focus on countering the false report injection [FRI] attack, which is also known as the false positive attack [1-8]. All of these filtering schemes use either static or dynamic authentication key sharing [1-5, 7, 8].

We propose to enhance the filtering capacity of the SEF scheme so that it not only filters false reports, but also allows legitimate reports with false MACs to reach the BS station without failure. The probabilistic

voting-based filtering scheme (PVFS) [2] is a static scheme that deals with both the attacks, and filters false reports at the probabilistically chosen verification nodes. In statistical en route filtering (SEF), each intermediate node verifies the report probabilistically, and if it detects an invalid MAC attached to it, it immediately drops it. SEF exploits network scale and density to drop false data through the collective detection power of several intermediate relay nodes. However, while making a decision to drop the report, SEF does not allow the forwarding nodes to consider the results of the previous verifications. Every intermediate node that finds an invalid MAC makes an independent decision to drop the report. This inflexibility of SEF allows room for the compromised nodes to impact the performance of the network. Compromised nodes launch a false negative attack by attaching false MACs to the legitimate reports that are dropped en route by the verification nodes. The false negative attack stalls the passage of true reports to the BS [1, 2, 6]. By appending a few extra bits in the header of the report being forwarded, we can make SEF restrict false negative attacks. Once a threshold for the verification of true reports is reached, they are marked safe, and forwarded without further verification.

The FRI attack aims to drain the energy resource of the sensor network, and render it useless in the presence of compromised nodes. The detection probability in SEF increases with distance. However, relying on the filtering capability of filtering nodes farther from the report generating cluster and closer to the BS leads to an uneven load share. An energy-hole syndrome appears in which the filtering nodes around the BS soon die out on account of their rapid depletion of energy and unceasing verification activity. The energy-hole phenomenon causes information lose and shortened network lifetime.

In SEF, each forwarded report is verified against T MACs created by keys from T distinct non-overlapping sub-pools of authentication keys. Firstly, each intermediate node checks if a report carries T MACs, as well as T key indices from T different partitions. Secondly, the intermediate node tries to check if a key's index in the report matches that of one of its own keys. If so, the intermediate node tries to authenticate the report by calculating a new MAC with the same key. If the new calculated MAC matches the MAC contained in the report, the report is authenticated, and forwarded. If the MAC is found to be false, the report is immediately dropped. If none of the key indices in the report matches a key index of the keys possessed by the node itself, an intermediate node simply forwards the report. Thus if it possesses the matching key, every intermediate node is virtually required to authenticate the report. None of the intermediate nodes considers the outcome of the previous verifications performed by the earlier nodes in the decision making. If a single MAC is found to be false, any intermediate node immediately drops the report. This is why the SEF schemes do not handle the false negative attack, as well as it incurs more energy by requiring every intermediate node to verify the report.

2. Statistical En route Filtering (SEF)

SEF is the first scheme that was proposed to filter false data injected by adversaries exploiting compromised nodes. In SEF, a pre-generated global key pool of size N, maintained at the BS, is divided into multiple non-overlapping n partitions, each of size m, i.e.

$$N = m \times n$$

Figure 1 shows the partitions of the global key pool and allocation of k keys to each sensor node in the network. Every key is mapped against a unique key index for identification purpose during the process of en route filtering. Prior to sensor deployment,

each node is preloaded with k (k<m) keys, along with their key indices from a single partition.

When an event occurs, neighbouring nodes prepare the reports and broadcast them. The broadcasted report is of the form: {LE,TS, E}, in which LE indicates the event occurrence location, TS is the event time-stamp, and E indicates the type of event. If a node finds that the difference between the broadcast values and its observed values are within the predefined error boundary, then the broadcast values are accepted. The node whose broadcast values are accepted by more nodes is elected as the Center of Stimulus (CoS) node. CoS is responsible for preparing a final report endorsed by T MACs attached to it. After the selection of the CoS, every detecting node A selects one key Ki from the pool of keys it possesses and generates a MAC.

$$MAC_i = f(K_i, Report)$$

Where, Report is of the form {LE||Ts||E}, and f(n,m) computes a MAC of the message m using key n. The node forwards {i, MACi}, the key index, and the MAC to the CoS. All the {i, MACi}'s tuples are collected by the CoS from the detecting nodes, and classified according to the key partitions. MACs created by the keys from the same partition belong to the same category. CoS randomly selects a single tuple {i, MACi} from each of T(T≤n) categories, and attaches them to the report. The final report forwarded towards the BS looks like:

$$\{L_E, T_s, E, i_1, MAC_{i1}, i_2, MAC_{i2}, \dots, i_T, MAC_{iT}\}.$$

each forwarded report carries exactly T key indices and T MACs. Reports carrying more or less than T key indices and T MACs are dropped en route. Reports that contain more than one key index and MACs from the same partition are also dropped en route.

Since each node is preloaded with a set of randomly chosen keys from a randomly selected partition of the global key pool, it is predicted to possess a key with certain probability that is used to generate one of the T MACs attached to the report. That key is used to verify the authenticity of the report.

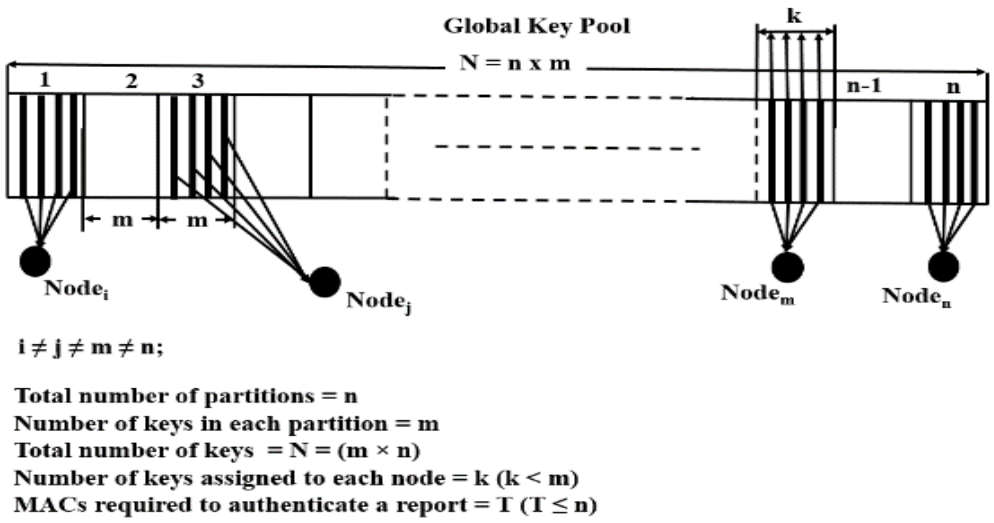

Figure. 1: Global Key pool, Its Partition and Allocation of Keys to Individual Nodes.

3. False Negative Attack Detection in (SEF)

The robustness of SEF against FRI attack is solely based on the fact that a compromised node can possess keys from only one category. In order to produce a counterfeit report, the compromised node is still required to forge the remaining T-1 MACs. This is why SEF provides a strong protection against the FRIA attack and becomes an ideal choice among the filtering schemes.

However, SEF suffers from a serious weakness when it comes to protection against a false negative attack viz. a False MAC injection (FMI) attack. SEF doesn't consider an FMI attack: neither does it provide a means to safeguard against FMI attack. Compromised nodes are exploited to launch an FMI attack which causes the dropping of legitimate reports.

We propose to include a few more bits in the report header, at the expense of a little energy-per-bit, to achieve greater security against FMI attack. Our proposed method also helps to save a significant amount of energy, by relieving nodes around BS from the verification of legitimate reports. Relieving nodes around the BS from the task of verification avoids energy-hole syndrome and increases network lifetime.

A. Proposed Methodology: When CoS finalizes the report, it also appends two extra fields Verf and Vert, and a flag bit Accepted. Verf records the number of verified false MACs, while Vert records the number of verified true MACs. Once we include these two fields in the header, the intermediate verification nodes will no longer drop the report when finding a single false MAC attached to it. If Vert reaches its threshold, the verification node marks the report safe, sets the corresponding flag bit Accepted to 1, and forwards the report. If the verification node finds that Verf has reached its threshold, it immediately drops the report and informs the BS about its decision. The length of the two fields depends on the length of T. Notice that even though the remaining MACs may be false after Vert has reached its threshold, there is still no need to verify the report further, as the majority of the MACs are true. Every node, Ni, shares a symmetric key KNi, BS with the BS. Using its symmetric key, the intermediate verification node creates a signature of its verification and sends it along with a report to the BS. Now the report that is forwarded to the BS looks like:

$$\left\{L_E, T_s, E, \{(i_s, MACs)\}, Ver_f, Ver_t, Accepted, \{Sig \leq T\}\right\}.$$

Where

$$\{(i_s, MACs)\} = \{i_1, MAC_{i1}, i_2, MAC_{i2}, \dots, i_T, MAC_{iT}\}$$

The inclusion of a few extra bits provides higher security against the false negative attack, and consumes very little energy. Algorithm 1 shows the verification process of the report at an intermediate node Nk.

Algorithm 1.

```
1:  Nk: Intermediate node
2:  Report = Nk.receive({L_E, T_s, E, {(i_s, MACs)}, Ver_f, Ver_t, Accepted, {Sig ≤ T}})
3:  if Report.Accepted = 1 then
4:      Forward Report, EXIT;
5:  end if
6:  if {(i,MAC_i) and (j,MAC_j)} ∈ Report.({(i_s, MACs)}) and i = j then
7:      Drop Report, EXIT
8:  end if
9:  For each i in T (i, MAC_i) tuples in Report.({(i_s, MACs)})
10:     if Nk stores (j,key_j) such that i = j then
11:         if MAC_i = f(Key_j, Report) then
12:             Report.Ver_t := Report.Ver_t +1;
13:             if Report.Ver_t = 3 then
14:                 set Accepted := 1;
15:             end if
16:         else if MAC_i ≠ f(Key_j, Report) then
17:             Report.Ver_f := Report.Ver_f +1;
18:                 if Report.verf = 2 then
19:                     Drop report, EXIT;
20:                 end if
21:         end if
22:     end if
23: end for
24: Forward report, EXIT;
```

As soon as the value of Verf reaches 2, the report is immediately dropped whereas when

the value of Vert reaches 3, the report is marked safe and forwarded to the BS, without further en route verifications.

4. Simulation Results

We have performed simulations to verify the efficiency of our proposed method against the FMI attack. We assume a network comprising 400 sensor nodes uniformly distributed across a field of size 200×40 m2. Simulations are carried out in a custom simulator developed in Microsoft Visual C++ 2012. The hops between the source node and the BS are varied. Reports are generated by the source every 2 seconds. A global key pool of 1000 keys is divided into 10 partitions with 100 keys in each partition. Each node is equipped with 70 keys. Each report and MAC is 36 and 4 bytes in size, respectively. It takes 15 μJ to generate a MAC, 75 μJ to verify a report, and 16.25 μJ and 12.5μJ to transmit and receive a byte, respectively. The threshold values of Verf and Vert are set to 2 and 3, respectively, for T = 5. As soon as the value of Vert reaches 3, the Flag bit Accepted is set to 1, and no more verifications are required.

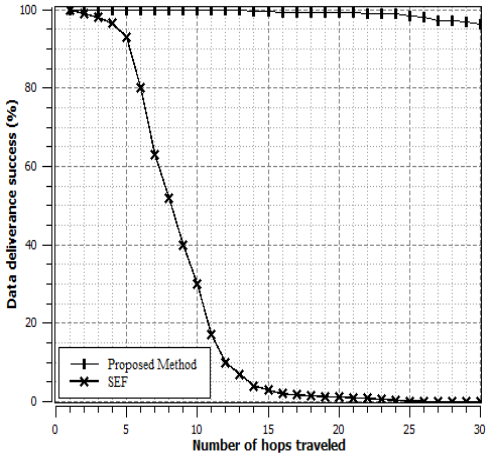

Figure 2: Comparative Analysis of Report Delivering SEF and the Proposed Method, in the Presence of False Negative Attacks.

Figure 2 shows that legitimate reports are delivered to the BS with higher success rate in our proposed method than in SEF. The delivery of legitimate reports is low, solely because after being detected with a false MAC attached to them they are dropped immediately.

5. Conclusion

FRI and FMI attacks are two major attacks that can happen to sensor networks. While SEF provides an excellent safeguard against the former, it leaves room for the latter to badly impact the network's information delivery capability. By appending a few extra bits in the report, we can make SEF to reject false negative attacks. The inclusion of a few extra bits provides higher security against the false negative attack, while consuming very little energy in transmitting them along with the report.

Acknowledgments

This research was supported by the 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

[1] F. Ye, H. Luo, and S. Lu, "Statistical en-route filtering of injected false data in sensor networks", IEEE J. Selected Areas Commun., vol. 23, issue 4, pp. 839-850, April 2005.

[2] F. Li, A. Srinivasan, and J. Wu, "PVFS: a probabilistic voting-based filtering scheme in wireless sensor networks", Int. J. Security Netw., vol. 3, issue 3, pp. 173-182, January 2008.

[3] Z. Yu, and Y. Guan, "A dynamic en-route scheme for filtering false data injection in wireless sensor networks", SenSys, vol. 5, pp. 294-295, November 2005.

[4] A. S. Uluagac, R. A. Beyah, and J. A. Copeland, "TIme-Based dynamic keying and en-route filtering (TICK) for wireless sensor networks", IEEE Global Telecommunications Conference (GLOBECOM 2010), pp. 1-6, December 2010.

[5] C. Kraub, et al. "STEF: A secure ticket-based en-route filtering scheme for wireless sensor networks", The second international conference on Availability, reliability and security, IEEE ARES, pp. 310-317, April 2007.

[6] M. Akram, and T.H. Cho, "Energy efficient fuzzy adaptive selection of verification nodes", Ad Hoc Networks, vol. 47, issue 1, pp. 16-25, September 2016.

[7] H. Yang, and L. Songwu, "Commutative cipher based en-route filtering in wireless sensor networks", Vehicular Technology Conference, VTC2004, Vol. 2.pp. 1223-1227, September 2004.

[8] 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 IEEE symposium on Security and privacy, pp. 259-271, May 2004.