Journal of System Simulation

Volume 31 | Issue 1 Article 4

4-16-2019

Modeling and Simulation of False Report Filtering Scheme Based on Position in Wireless Sensor Networks

Zhixiong Liu
School of Computer engineering and applied mathematics, Changsha University, Changsha 410003, China;

Limiao Li
School of Computer engineering and applied mathematics, Changsha University, Changsha 410003, China;

Modeling and Simulation of False Report Filtering Scheme Based on Position in Wireless Sensor Networks

Abstract

Abstract: In wireless sensor networks, the adversary can inject false reports from compromised nodes. Previous security designs cannot detect faked reports that are forged coordinately by a group of compromised nodes. Furthermore, in sparse areas, some events failed to be reported to sink. *This paper proposes a position based filtering scheme (PFS). It derives the optimal coverage degree ω, and the nodes are deployed accordingly. After deployment, each node distributes its position to some other nodes. When a report is generated for an observed event, it must carry t distinct MACs (Message Authentication Codes) along with positions of all detecting nodes. Each forwarding node verifies the correctness of MACs and positions, and checks the legitimacy of all locations attached in the report.* Analysis and simulation results show that PFS outperforms existing schemes in terms of covering efficiency, compromise tolerance and filtering capacity.

Keywords

wireless sensor network, false report injection, optimal coverage, position, compromise tolerance

Recommended Citation

Liu Zhixiong, Li Limiao. Modeling and Simulation of False Report Filtering Scheme Based on Position in Wireless Sensor Networks[J]. Journal of System Simulation, 2019, 31(1): 27-35.

第 31 卷第 1 期
2019 年 1 月

系统仿真学报©
Journal of System Simulation

Vol. 31 No. 1
Jan., 2019

Modeling and Simulation of False Report Filtering Scheme Based on Position in Wireless Sensor Networks

Liu Zhixiong, Li Limiao

(School of Computer engineering and applied mathematics, Changsha University, Changsha 410003, China)

Abstract: In wireless sensor networks, the adversary can inject false reports from compromised nodes. Previous security designs cannot detect faked reports that are forged coordinately by a group of compromised nodes. Furthermore, in sparse areas, some events failed to be reported to sink. *This paper proposes a position based filtering scheme (PFS). It derives the optimal coverage degree ω, and the nodes are deployed accordingly. After deployment, each node distributes its position to some other nodes. When a report is generated for an observed event, it must carry t distinct MACs (Message Authentication Codes) along with positions of all detecting nodes. Each forwarding node verifies the correctness of MACs and positions, and checks the legitimacy of all locations attached in the report.* Analysis and simulation results show that PFS outperforms existing schemes in terms of covering efficiency, compromise tolerance and filtering capacity.

Keywords: wireless sensor network; false report injection; optimal coverage; position; compromise tolerance

传感器网络中基于位置的虚假数据过滤的建模与仿真

刘志雄，黎梨苗
(长沙学院 计算机工程与应用数学学院，长沙 410003)

摘要：攻击者可以通过妥协节点往传感器网络中注入虚假数据。已有安全机制无法检测由一组妥协节点协同伪造的虚假数据，且部分稀疏区域的事件无法顺利上报到 sink。*提出一种基于位置的过滤方案 PFS。推导出最优覆盖度 ω，并基于该结论部署节点。节点在部署后将位置分发给部分其它节点。每个数据报告必须附带 t 个不同的消息验证码 MAC (Message Authentication Codes) 以及各检测节点的位置。各转发节点同时对 MAC 和位置的正确性，以及位置的合法性进行验证。*分析及仿真表明，PFS 在覆盖效率，妥协容忍以及过滤性能方面均优于已有方案。

关键词：无线传感器网络；虚假数据注入；最优覆盖；位置；妥协容忍
中图分类号：TP393　　文献标识码：A　　文章编号：1004-731X (2019) 01-0027-09
DOI: 10.16182/j.issn1004731x.joss.17-0043E

Introduction

Wireless sensor networks (WSNs) collect and process critical information for both military and industrial applications [1-2]. Therefore, security issues are indispensable for the designing of the networks. The adversary can use the secrets stored in compromised nodes to launch false data injections[3], i.e., to inject bogus reports into networks. This type of attacks not only causes false alarms, but also drains out the constrained resources of sensor nodes[4].

Received: 2017-01-09　　Revised: 2017-05-18;
Foundation Item: National Natural Science Foundation of China (61502057);
Biographies: Liu Zhixiong (1982-), male, Loudi, Hunan, Doctor, associate professor, Research on wireless sensor networks; Li Limiao (1979-)，female, Yueyang, Hunan, Doctor, lecturer, Research on wireless sensor networks.

To resist false report injections, a general en-route filtering framework called SEF has been proposed[5]. It divides a global key pool G into n partitions, each partition has m keys. Every node is pre-loaded with a randomly selected key partition from G. All nodes are then randomly deployed in the monitoring region. Detecting nodes handle the stimulus coordinately and generate a report that carries t (t >1) distinct MACs. A MAC is generated by one of the symmetric keys. Each forwarding node has some probability to verify the attached MACs. A report that carries less than t MACs or wrong MACs is treated as an intrusive.

The state-of-the-art false data filtering schemes[6-10] are all based on the framework. Yu et al.[6] presented a grouping-based resilient filtering scheme called GRSEF. Nodes form t groups in GRSEF rather than n ($n > t$) groups in SEF, which ensures that any position being covered by t groups simultaneously. Every node fetches keys through multiple axes. Yang et al.[7] proposed a dynamic en-route filtering scheme DEFS. All nodes are grouped into clusters, and distribute keys along multiple paths using Hill *Climbing* method. The method ensures that nodes near a cluster storing more keys than those far away do. Bashir et al.[8] presented ERFS to detect in- network RFID copies in sensor networks. The scheme uses a clustering mechanism where cluster heads eliminate duplicate data and forward filtered data towards the base station. Dobrev et al.[9] put forward OARB. The scheme assumes that adversary can move arbitrarily fast, and take full control of sensors. The aimed network is a combination of rotating sensors of beam sensors and angle zero. It filters out false reports along the path from the cluster head to sink. Wei et al.[11] proposed a dynamic covering algorithm of WSN called DCA. It is based on the Centralized Voronoi Tessellation theory, and combined with the Lloyd's algorithm to achieve high covering efficiency. Gil et al.[12] addressed a Maximum Set Covers for DSNs (MSCD) problem, which is known to be NP-complete, and then proposed a target coverage scheduling scheme based on a genetic algorithm. Bara'a et al.[13] modeled the Maximum Set Covers for DSNs (MSCD) problem to design an energy-efficient wireless sensor network that can reliably cover a target area.

As can see above, keys are not bound to positions in these schemes, and thus they cannot detect fake reports that are forged coordinately by multiple compromised nodes from different regions. Moreover, for lacking of enough detecting nodes, some events occurred in sparse areas fail to be reported to sink. An example is shown in Fig. 1. Here t is set to five, and we assume the adversary has compromised S_1, …, S_5 with distinct key partitions. Obviously, the adversary can claim fabricated events arbitrarily, and breaks through existing security designs. Moreover, with only three key partitions covering B, the detecting nodes fail to generate legitimate reports.

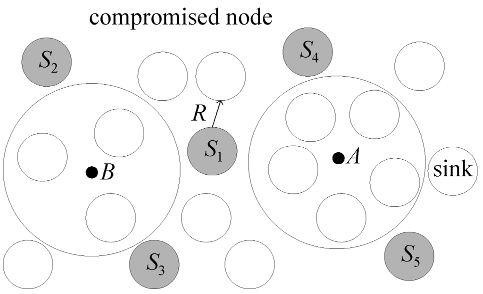

Fig. 1 Coordinated false report injection and sparse covering

This paper proposes a position based false report filtering scheme (PFS). In PFS, we employed covering algorithm to deploy nodes and derived the optimal coverage degree ω by considering both monitoring quality and network overhead.

Consequently, all events can be reported to sink successfully. After deployment, each node distributes its position to some other nodes. The report should carry t MACs with distinct key partitions and t positions of these nodes, respectively. Each forwarding node checks not only the correctness of MACs and positions attached in the report, but also the legitimacy of locations. As a result, coordinated false report injections can be resisted efficiently. For example, if 10 nodes are compromised, the probabilities for the adversary to break down SEF and PFS are 93.4%, 3.1%, respectively.

1 Modeling of PFS Scheme

In this section, we first derive the optimal coverage degree for false report filtering, which is the basis of nodes deployment in PFS, then discuss the system model of the scheme, and finally illustrate the four aspects of PFS, namely the initialization and bootstrapping phase, the reports generating phase, the en-route filtering phase and the sink verifying phase. As sink verifying is almost the same in SEF[5], we are going to omit it here.

1.1 Derivation of optimal coverage degree

For simplicity, the covering of t key partitions (or more) is noted as t-k cover; For some covering algorithm with covering degree ω, the probability of t-k cover is noted as $p_\omega(t,...,n)$, here n is the number of key partitions in a global key pool G' with total N keys ($N = n\times m$). Obviously, each partition includes m keys.

Definition 1*:* Given $0<\theta<1$, $0<\varepsilon<1$. Assume that all nodes are deployed according to some covering algorithm. With ω increases, When $p_\omega(t,...,n) \geq\theta$, if ($p_\omega(t,...,n)$ - $p_{\omega\text{-}1}(t,...,n)\geq\varepsilon$) and ($p_{\omega+1}(t,...,n)$-$p_\omega(t,...,n)<\varepsilon$) is true with ω increases, ω is *optimal.*

Parameters θ and ε are going to be discussed in section 4.6.

Theorem 1: In a global key pool G' divides into n partitions ($N = n\times m$), if each node selects one from n partitions randomly. For totally ω nodes, the probability that obtain at least t distinct partitions is

$$p_\omega(t,\cdots,n) = 1 - \frac{\sum_{m=1}^{t} C_n^m \times C_{\omega-1}^{\omega-m}}{C_\omega^{\omega+n-1}} \tag{1}$$

Proof: First, we consider the situation of obtaining exactly t distinct partitions. Denote ω nodes by set Q_1, and n key partitions by set Q_2. Each node in Q_1 takes an element from Q_2 randomly, the number of combinations is $C(n+\omega\text{-}1,\ \omega)$. The number of combinations that exact t nodes hold distinct partitions is $C(n,\ t)C(\omega\text{-}1,\ \omega\text{-}t)$. Thus, the probability that exact t nodes hold distinct partitions is

$$p_\omega(t) = \frac{C_n^t \times C_{\omega-1}^{\omega-t}}{C_\omega^{n+\omega-1}} \tag{2}$$

Similarly, the probabilities to get exact 1, …, t -1 partitions are computed. Then, the probability to get at most t partitions is

$$p_\omega(1,\cdots,t) = \frac{\sum_{m=1}^{t} C_n^m \times C_{\omega\text{-}1}^{\omega\text{-}m}}{C_\omega^{n+\omega-1}} \tag{3}$$

So the probability to get at least t partitions is p=1- ($p_\omega(1)$+$p_\omega(2)$+…+$p_\omega(t)$).

The covering degree should be large enough to ensure the effectiveness of t-k cover; it could not be too large to avoid incurring huge energy consumption.

Theorem 2: Assume that nodes are deployed according to some covering algorithm, and each report carries t MACs. Let ε be 0.05, and θ be 0.8, then ω= 2t is the optimal covering degree for false report filtering.

Proof: From Theorem 1,

$$P_{\omega+1}(t)=\frac{C_n^t\times C_\omega^{\omega-t+1}}{C_{n+\omega}^{\omega+1}}=\frac{\omega\cdot(\omega+1)}{(\omega+1-t)\cdot(n+\omega)}P_\omega \quad (4)$$

As n is larger than 1, when t=1, there is,

$$P_2(t)=\frac{C_n^1\times C_1^1}{C_{n+1}^2}=\frac{2}{n+1}\le 0.8 \quad (5)$$

Assume Equation 4 is right under $t = k$, then in case t=k+1,

$$P_2(k+1)=\frac{C(n,k+1)C(2k+1,k+1)}{C(n+2k+1,2k+2)}=$$

$$\frac{4(n-t)(2t+1)^2}{(t+1)(n+2t+1)(n+2t)}P_{2t} \quad (6)$$

Obviously,

$$\frac{4(n-t)(2t+1)^2}{(t+1)(n+2t+1)(n+2t)}\le 1 \quad (7)$$

Consequently, we get $P_2(k+1) \le P_2(t) \le 0.8$. Using derivation methods, in case $\omega \ge 2t$, there is $P_{\omega+1}(t)$-$P_\omega(t) \le 0.05= \varepsilon$. In the similar way, $P_\omega(t) - P_{\omega-1}(t) \ge 0.05= \varepsilon$ can be computed. From the above we know that, for any covering algorithm, only when covering degree ω equals to $2t$, it is optimal for false report filtering in wireless sensor networks.

The change of p according to ω is investigated by treating both of the number of key partitions n and the system security threshold t as constants in Fig. 2, here t = 5, n = 10, θ=0.8, and ε =0.05. We can see, when $\omega < 2t$, p increases quickly with the increasing of covering degree. E.g., when covering degree increases from seven to eight, the increment of p is 0.21 (larger than ε). While when ω >2t, p only get a small increment according to the increase of covering degree, e.g, the increment of p is 0.04 < ε when covering degree turns from 11 to 12. Therefore, both theoretical and simulation results show that ω =2t is optimal.

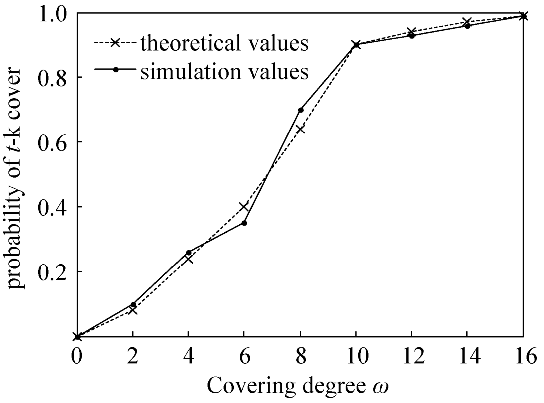

Fig. 2 Theoretical and simulation results of covering probability

1.2 System and Threat Model

Assume that all detecting sensors (sensing radius r_s and communication radius r_c) process the stimulus coordinately and elect one of them as the Center-of-Stimulus (CoS) [5]. The CoS collects the results from all detecting nodes and produces a report, then forwards it to sink.

Sink has full knowledge of the keys and positions of all nodes. It is strong enough to detect all false reports that escaped from en-route filtering.

Assume the network has a short safe bootstrapping period, during which nodes deployment, localization and distribution are finished without any node being compromised.

1.3 Initialization and Bootstrapping

Each node randomly selects one key partition from G' to store, and then is deployed according to DCA algorithm[11]. Based on the former investigations, covering degree is set to 2t.

After deployment, the localization algorithm ALSW [14] is adopted to obtain positions for nodes. L_i:(X_i, Y_i) denotes the position of S_i , here X_i, Y_i represents x coordinate and y coordinate of S_i, respectively.

Then, each node distributes g copies of (S_i, L_i, U_i)

to the other nodes using Bubble-geocast algorithm[2], here U_i denotes the key partition index of S_i. *Bubble-geocast* can disseminate a large number of seeds uniformly. Finally, each node has probability g / N_a to store positions for other nodes, where N_a denotes total number of nodes in the network. The choice of g is discussed in section 2.5.

1.4 Report Generation

Upon detecting an event, the CoS broadcasts its sensing value E to all neighbors. Each detecting node S first checks whether its sensing result is consistent with E within a certain error range. If they match, S selects one key randomly to generate a MAC M_i: $K_i(E)$. It then sends a short data package including its ID, MAC and position to the CoS. The CoS gathers all information and selects t nodes to generate a report by attaching some extra information. The final report sent out by the CoS looks like $\{E, L_E; i_1, i_2, \ldots, i_t; M_{i1}, M_{i2}, \ldots, M_{it}; j_1, j_2, \ldots, j_t; L_{j1}, L_{j2}, \ldots, L_{jt}\}$, where L_E, i_t and j_t denote position of E, key index and node ID, respectively.

1.5 En-route Filtering

Upon receiving report R, the node first checks whether R includes t distinct key partitions, t IDs, t MACs and t positions, respectively. If any of these does not meet, R is dropped. Then it verifies position's legitimacy, i.e., for detecting node S_i and event E, the distance between positions of S_i and E should be within sensing radius r_s, which is denoted as $| L_i , L_E |\leq r_s$. Next, if holding exactly one key in the report, the node uses its own key to re-produce a MAC and compares the two MACs. The report is also dropped when the attached one differs from the new one. Finally, the report is forwarded. The pseudo-code of en-route filtering is illustrated in Algorithm 1.

Algorithm 1 En-route Filtering in PFS

```
/* upon receiving $R$ */
1. Check if $R$ include $t$ $\{i_v, M_{iv}\}$ tuples; drop it otherwise.
2. Check if the $t$ key indexes $\{i_v, 1 \leq v \leq t\}$ from distinct partitions; drop it otherwise.
3. Check if $t$ $\{j_v, L_{jv}\}$ tuples exist in $R$; drop it otherwise.
4. Check ($|L_{jv}, L_e| \leq r_s$, $1 \leq v \leq t$); drop it otherwise.
5. If holding a position $L\in\{L_{jv}, 1 \leq v \leq t\}$ and partition $U$, check if the key index $i_v$ from $U$; then check the position the same as $L$; drop it otherwise.
6. If holding a key $K\in\{K_{iv}, 1 \leq v \leq t\}$, computes $M = K(e)$ and check $M_{iv}$ the same as $M$; drop it otherwise.
7. Forward $R$.
```

2 Performance Evaluation and Simulations

In this section we first analyze the coordinated false report injection attacks by compromised nodes from different regions, and then quantify the effectiveness of compromise tolerance, en-route filtering, energy savings and storage overhead. Base on the above results, we discuss how to choose appropriate parameters to improve the performance of PFS. Finally, we provide simulation evaluations.

2.1 Coordinated False Report Injections

PFS checks not only correctness of MACs, but also legitimacy of positions in each report, thus can filter out false reports injected coordinately by compromised nodes from arbitrary areas. As in Fig. 3, the attacker has captured five key partitions stored in $S_1, \ldots, S_5$ and t is set at five. If the attacker abuses secrets stored in these nodes to fake a report R: $\{E, L_E; i_1, i_2, \ldots, i_5; M_{i1}, M_{i2}, \ldots, M_{i5}; j_1, j_2, \ldots, j_5; L_{j1}, L_{j2}, \ldots, L_{j5}\}$, then R will be treated as illegitimate and dropped by intermediate nodes. The reason is, due to the distance between S_1 and S_4 is larger than $2r_s$, the attacker can not forge L_E to make the distance between L_1 and L_E, and that between L_4 and L_E are both no larger than r_s.

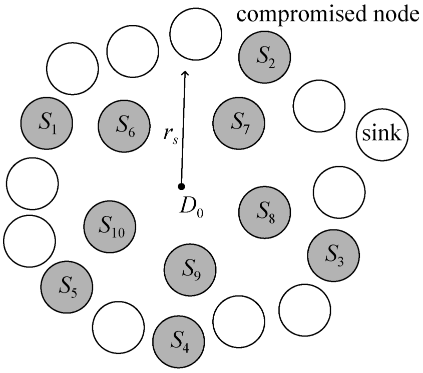

Fig. 3 Coordinated false report injection

In the worst situation, when captured t distinct key partitions within a π × r_s^2 circular region, the attacker can forge t legitimate positions and t correct MACs in a report to sneak through en-route filtering.

Theorem 3: In PFS, when N_c nodes located in arbitrary areas are compromised ($N_c \geq t$), and total area of the network is D, the probability that the attacker get at least t distinct key partitions within a π × r_s^2 region is

$$p_p = \sum_{i=t}^{N_c} A(i,n) \frac{C_i^{N_c} (\pi r_s^2 / D)^i (1 - \pi r_s^2 / D)^{N_c - i}}{n^i} \quad (8)$$

Proof: Assume D_0 in Fig. 3 is a π × r_s^2 region. Consider the attacker getting exactly t distinct key partitions in D_0. First, each node has probability π × r_s^2 / D to locate in D_0. So the probability that exact t nodes locating in D_0 is $p_u = C(t, N_c)\ (\pi \times r_s^2 / D)^{\ t}\ (1 - \pi \times r_s^2 / D)^{\ Nc\text{-}t}$. As each node has probability $p_r = A(i,\ n)\ /\ n^t$ to hold a distinct partition. So the probability that exact t key partitions locating in D_0 is $p_u \times p_r$.

In the similar way, the probabilities that attacker getting exactly t +1,…, n partitions in D_0 can be computed. Using accumulation, the probability that the attacker obtains at least t distinct key partitions within a πr_s^2 area is p_p.

Fig. 4 illustrates the theoretical and simulation results of p_S and p_p as t, D_0 / D and n are set to 5, 1/4 and 20. Here p_S denotes the probability to break through SEF. The simulation results are averaged over 5 000 random tests. We can see SEF is easily broken down by little compromised nodes, but PFS has much powerful defense. For example, with ten compromised nodes, the attacker has probabilities of 0.934 and 0.031 to break down SEF and PFS, respectively.

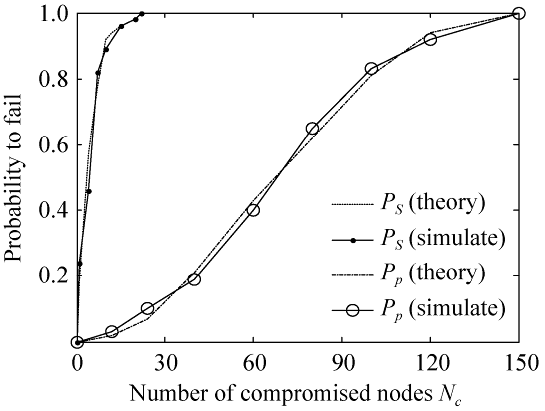

Fig. 4 Results of p_S and p_p from theory and simulate

2.2 Filtering Effectiveness

Assume N_c ($N_c < t$) nodes in a π × r_s^2 region are compromised. To produce a *seemingly* true report, the attacker must forge t–N_c MACs and t–N_c positions. The probability that a forwarding node S_i holds one of the faked keys is

$$p_a = \frac{k}{m} \cdot \frac{t - N_c}{n} = \frac{k(t - N_c)}{N} \quad (9)$$

The probability that S_i holds one of the faked positions is

$$p_b = 1 - (1 - c / N_c)^{t - N_c} \quad (10)$$

Let P_1 denote $P_a + P_b$ - $P_a P_b$, the fraction of false reports being dropped within h hops is

$$p_h = 1 - (1 - p_1)^h \quad (11)$$

The averaged hops a forged report traverses is

$$H=\sum_{i=1}^{\infty} p_1 \cdot i(1-p_1)^{i-1} \qquad (12)$$

2.3 Energy Consumption

The energy consumption of PFS comes from four aspects. The first is communication cost of nodes deployment and position distribution during bootstrapping. The second is communication overhead among all detecting nodes on report generation phase. The third is computation cost of en-route authentication. The fourth is communication overhead of reports forwarding.

As pointed out in References [2], the cost of MAC computation is much smaller than that of data transmission. And the iterative transmission of short packages among detecting nodes consumes little energy too. Moreover, the distribution of positions also sends short packets and thus is energy efficient. So we overlook these types of costs on the analysis.

The energy consumption is quantified as follows. Denote length of a clear report without any attachments, length of node ID, length of position and length of MAC as I_r, I_n, I_k and I_u, respectively. The lengths of a PFS report and a SEF report become $I_{r0}=I_r+I_k+t(I_n+I_k+I_u)$ and $I_{r1}=I_r+t(I_n+I_u)$.

The extra information in PFS incurs some more cost in reports transmission, reception and computation, which is reasonable for providing PFS with the strong capability to resist coordinated false report injections. Moreover, PFS even saves energy than existing schemes through its earlier dropping of false reports. Further simulation results verified this.

2.4 Storage Overhead

In PFS each node needs some extra storage for g positions than SEF. Let the size of a key, a position be 64 bits and 10 bits, respectively. The required storage of 50 keys and 50 positions is about 0.5 KB, slightly larger than 0.4 KB in SEF. Considering the performance promotion in PFS, such extra storage overhead is affordable.

The mainstream sensors, e.g., the MICA2 nodes, are equipped with 4KB SRAM and 128KB ROM, and thus easily to satisfy the requirements. Tab. 1 lists the storage overheads of the main filtering schemes.

Tab. 1 Storage cost comparison

Scheme	Storage overhead/ KB
SEF	0.4
GRSEF	2.4
DEFS	3.5
ERFS	3.8
OARB	4.2
PFS	0.5

2.5 Parameters Analysis

The choice of t and N_c can be referred to SEF[5]. The parameter θ is set to guarantee a large enough covering degree, thus to ensure the smooth reporting of data. The selection of θ is a tradeoff between energy costs and filtering efficiency. A larger θ brings a bigger t-k cover probability, but also incurs a heavier energy consumption resulted by complicated covering procedure. Only if the covering degree meets the limitation of θ, we then start to seek the optimal one.

The selection of ε is also a tradeoff between overhead and filtering efficiency. Given a small enough ε, with covering degree increases, if the increment of t-k cover probability is not larger than ε, then ω is treated as optimal. That is because with the increment of ω, the t-k cover probability only has a small increment while the corresponding energy costs increases considerably.

The parameter g affects filtering efficiency and the capacity to resist coordinated false report injections. At worst (i.e., g =0), the security capacity of PFS is equal to SEF.

2.6 Simulation Evaluation

To further verify the performance, we establish a simulation platform using C++ language. Due to space constraint, we only present results for en-route filtering and energy consumption when g =20, 40, θ=0.8, and ε =0.05. We use a 40 m×40 m area where 400 nodes are deployed according to: (1) uniform distribution; (2) DCA covering, respectively. One stationary sink and source sit in opposite sides of the area. For SEF and PFS, we use a global key pool of 150 keys, which is divided into 15 partitions. Other parameters are shown in Tab. 2. The results are averaged over 20 topologies.

Tab. 2 Simulation parameters

Parameters	Value
Interval of generating a report	2s
Total number of reports	100
Communication radius	2.5 m
Sensing radius	10 m
Transmission energy	6×10^{-3} Joules
Reception energy	1.2×10^{-3} Joules
Security threshold (t)	5

Fig. 5 shows the t-k cover comparison. With little amount of deployed nodes, the performance of both deploy methods is weak. As deployed nodes increase, t-k cover effectiveness of uniform distribution increases slowly, e.g., there is only an increment of 3% from 200 deployed nodes to 400. While under the same situation, DCA gets a big increment of 87%.

Fig. 6 illustrates the fraction of dropped false reports grows as hops. According to the increase of hops, filtering probability of both algorithms increases gradually. When travelling the same hops, filtering ability of PFS is much better than SEF. For example within five hops, the percentage of dropped reports is 22% in SEF, 68% in PFS (g=20), respectively.

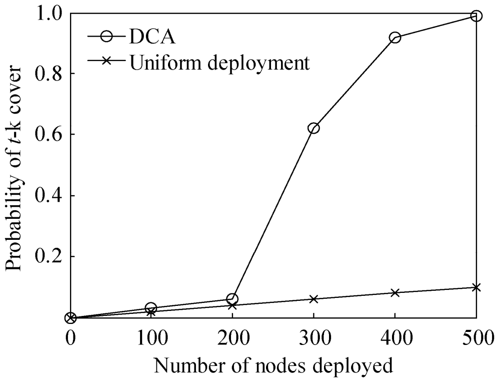

Fig. 5 t-k cover probability comparison

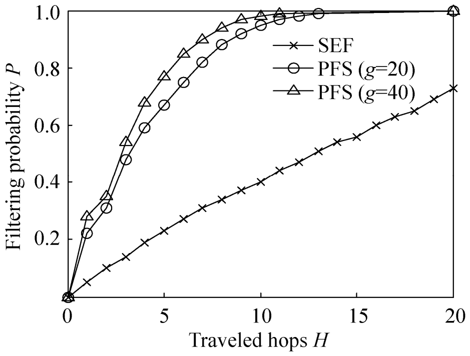

Fig. 6 Fraction of dropped false reports grows as hops

Fig. 7 plots how energy cost changes as a function of traveling hops. It is easy to see that when traveling more than two hops in the network, PFS consumes less energy than SEF due to earlier dropping of the detected false reports.

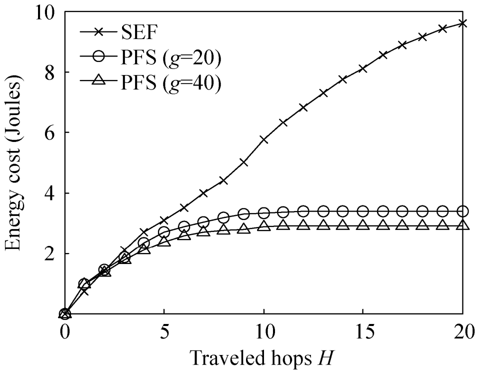

Fig. 7 Energy cost changes as H grows

3 Conclusion

(1) A new scheme PFS is proposed to resist coordinated false report injections in sensor networks.

(2) PFS derives the optimal coverage degree suitable for false report filtering, and utilizes positions of sensor nodes to detect false reports that are forged by compromised nodes from different regions. PFS increases the difficulty for the adversary to break down the security design.

(3) Simulation results indicate that PFS outperforms SEF in terms of compromise tolerance, filtering efficiency and energy consumption.

References:

[1] Jaggi S, Langberg M, Katti S, et al. Resilient Network Coding in the Presence of Byzantine Adversaries [J]. IEEE Transactions on Information Theory (S0018-9448), 2008, 54(6): 2596-2603.

[2] Peng S L, Li S S, Liao X K, et al. Estimation of a Population Size in Large-Scale Wireless Sensor Networks [J]. Journal of Computer science and technology (S1000-9000), 2009, 24(5): 987-996.

[3] Naresh K, Pradeep K P, Sathish K S. An Active En-route Filtering Scheme for Information Reporting in Wireless Sensor Networks [J]. International Journal of Computer Science and Information Technologies (S1947-5500), 2011, 2(4): 1812-1819.

[4] Bose P, Morin B, Stojmenovic I, et al. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks [J]. Wireless Networks (S1022-0038), 2011, 7(6): 609-616.

[5] Ye F, Luo H, Zhang L. Statistical En-route Filtering of Injected False Data in Sensor Networks [C]. Proceedings of 23th Annual Joint Conference of the IEEE Computer and Communications Societies (S1930-1650). Hong Kong, China: IEEE, 2004: 2446-2457.

[6] Yu L, Li J Z. Grouping-based Resilient Statistical En-route Filtering for Sensor Networks [C]. Proceedings of 28th Annual Joint Conference of the IEEE Computer and Communications Societies (S1930-1650). Rio de Janeiro, Brazil: IEEE, 2009: 1782-1790.

[7] Yang F, Zhou X H, Zhang Q Y. Multi-Dimensional Resilient Statistical En-Route Filtering in Wireless Sensor Networks [J]. Journal of Internet Technology (S1607-9264), 2010, 12(1): 130-139.

[8] Bashir A K, Lim S J, Hussain C S, et al. Energy Efficient In-Network RFID Data Filtering Scheme in Wireless Sensor Networks [J]. IEEE Sensors Journal (S1530-437X), 2011, 11(7): 7004-7021.

[9] Dobrev S, Narayanan L, Opatrny J. Optimal Sensor Networks for Area Monitoring Using Rotating and Beam Sensors [J]. Theory of Computing Systems (S1433-0490), 2014, 54(4): 622-639.

[10] Cao Z, Deng H, Guan Z, et al. Information-Theoretic Modeling of False Data Filtering Schemes in Wireless Sensor Networks [J]. ACM Transactions on Sensor Networks (S1550-4859), 2012, 8(2): 72-83.

[11] Wei H X, Mao Q. A Dynamic Covering Algorithm of Wireless Sensor Network Based on CVT [C]. 20th International Conference on Embedded and Real-Time Computing Systems and Applications (S1533-2306). Chongqing, China: IEEE, 2014: 1-6.

[12] Gil J M, Han Y H. A Target Coverage Scheduling Scheme Based on Genetic Algorithms in Directional Sensor Networks [J]. Sensors (S1424-8220), 2011, 11(2): 1888-1906.

[13] Attea B A, Hameed S M. A Genetic Algorithm for Minimum Set Covering Problem in Reliable and Efficient Wireless Sensor Networks [J]. Iraqi Journal of Science (S0067-2904), 2015, 55(1): 224-240.

[14] Bernas M, Plazek B. Energy Aware Object Localization in Wireless Sensor Network Based on Wi-Fi Fingerprinting [J]. Computer Networks (S1389-1286), 2015, 522(1): 33-42.