A Novel En-route Filtering Scheme against False Data Injection Attacks in Cyber-Physical Networked Systems

Xinyu Yang*, Jie Lin*, Paul Moulema†,Wei Yu†, Xinwen Fu‡ and Wei Zhao§
*Xi'an Jiaotong University, Emails: yxyphd@mail.xjtu.edu.cn, Dr.linjie@stu.xjtu.edu.cn
†Towson University, Email: pmoule1@students.towson.edu, wyu@towson.edu
‡University of Massachusetts Lowell, Email: xinwenfu@cs.uml.edu
‡University of Macau, Email: WeiZhao@umac.mo

***Abstract*—In Cyber-Physical Networked Systems (CPNS), attackers could inject false measurements to the controller through compromised sensor nodes, which not only threaten the security of the system, but also consumes network resources. To deal with this issue, a number of en-route filtering schemes have been designed for wireless sensor networks. However, these schemes either lack resilience to the number of compromised nodes or depend on the statically configured routes and node localization, which are not suitable for CPNS. In this paper, we propose a *Polynomial-based Compromised-Resilient En-route Filtering scheme (PCREF)*, which can filter false injected data effectively and achieve a high resilience to the number of compromised nodes without relying on static routes and node localization. Particularly, PCREF adopts polynomials instead of MACs (message authentication codes) for endorsing measurement reports to achieve the resilience to attacks. Each node stores two types of polynomials: authentication polynomial and check polynomial derived from the primitive polynomial, and used for endorsing and verifying the measurement reports. Via extensive theoretical analysis and simulation experiments, our data show that PCREF achieves better filtering capacity and resilience to the large number of compromised nodes in comparison to the existing schemes.**

***Keywords*—Cyber-physical networked system, false measurement report, sensor networks, polynomial-based en-route filtering.**

I. Introduction

Monitoring and controlling physical systems through geographically distributed sensors and actuators have become an important task in numerous environment and infrastructure applications. These applications have received a renewed attention because of the advances in sensor network technologies and new development in cyber-physical networked systems (CPNS) [3]. CPNS, consisting of sensor nodes, actuators, controller, and wireless networks, have been widely used to monitor and affect local and remote physical environments [11]. CPNS can make on how we interact with the physical world. In CPNS, sensor nodes obtain the measurement from the mobile physical components, process the measurements and send measured data to the controller through networks. According to these measurements, the controller estimates the states of physical systems and sends feedback commands to the actuators, which control physical environments and mobile systems.

CPNS may operate in hostile environments and sensor nodes in CPNS lacking tamper-resistance hardware increases the possibility to be compromised by attackers. The attacker can inject false measurement reports to the controller through the compromised nodes. This causes the controller to estimate wrong system states [8] and poses the dangerous threats to the system. The 2003 Eastern blackout was caused by the fact that programs for key areas were abnormal and failed to provide the system operators the correct state information [2]. The false reports consume lots of network and computation resources and shorten the lifetime of sensor networks and CPNS [6]. Hence, to ensure the normal operation of the system, it is critical to filter false data at forwarding nodes before arriving at the controller.

In the past, a number of schemes have been designed to filter the false injected data in sensor networks [15], [13], [16], [9], [20], [12], [17], [7]. However, those schemes have their limitations and cannot be used to effectively deal with attacks related to CPNS. For example, SEF [15] and IHA [20] have the T-threshold limitation, that is, if the attackers compromise T nodes from different groups, they could launch node impersonating attack on legitimate nodes. Thus, it lacks resilience to the increased number of compromised nodes. LBRS [13], LEDS [9] and CCEF [12] improve the resilience to the number of compromised nodes by introducing static routes for data dissemination and node localization. The static routes are not only vulnerable to node failure and denial-of-service (DoS) attacks (causing the controller not to receive measurement on time and loss of control over the system), but also not suitable for monitoring mobile physical components or systems. DEFS [17] and GRSEF [16] do not depend on static routes, but they achieve low resilience to the number of compromised nodes, and DEFS introduces lots of extra control messages, incurring energy consumption on nodes.

In this paper, we propose a Polynomial-based Compromised-Resilient En-route Filtering scheme (PCREF) for CPNS, which could filter false injected data effectively and achieve a high resilience to the number of compromised nodes without relying on the static data dissemination routes and node localization. PCREF is more suitable for CPNS to monitor and affect mobile physical components and systems.

PCREF adopts polynomials instead of MACs (message authentication codes) to verify reports, and can mitigate node impersonating attacks against legitimate nodes. In our scheme, the sharing the authentication information between nodes with a pre-defined probability avoids the node association to share authentication information between source nodes and forwarding nodes, and thus our scheme does not depend on static routes. The cluster-based polynomial assignment ensures that different clusters are assigned different primitive polynomials and suppresses the effect of compromised nodes into local area. Hence, PCREF achieves high resilience against the increased number of compromised nodes.

Via extensive theoretical analysis and simulation experiments, we evaluate the effectiveness of PCREF in comparison with SEF [15], LBRS [13], GRSEF [16] and LEDS [16] in terms of filtering efficiency, filtering capability, and resilience to the number of compromised nodes. Our data show that PCREF achieves better performance than existing schemes. For example, the filtering efficiency of PCREF increases as the forwarded hop increases, and it is always greater than that of existing schemes. When forwarded hops reach to 20, PCREF could filter out all false data, while the best of other schemes could only filter out 70%. In terms of filtering capability, PCREF could filter false data within 7 forwarded hops with a large number of compromised nodes, while other schemes could lose the en-route filtering capability completely. With the same number of compromised nodes, the compromised area ratio of PCREF is the lowest in comparison with the existing schemes.

The remainder of the paper is organized as follows: We introduce the network and threat models in the Section II. We present our proposed scheme in Section III. We analyze the security of our scheme in Section IV. In Section V, we show both analytical and experimental results to validate our findings. We review related work in Section VI and conclude this paper in Section VII.

II. Network and Threat Models

CPNS is used to receive measurements from sensor nodes, estimate system states, and send commands to the actuators to control the operation of physical systems. Each physical component or system is measured by multiple sensing nodes to increase resilience to faults and the nodes that measure the same component are organized as a cluster. A number of nodes in the cluster collect measurements and send data to the controller via multiple hops. To simplify our analysis, we assume only one controller in the system. Nodes may be mobile and nodes within the same cluster are relatively static to each other.

There are two types of nodes in the system: sensing nodes and forwarding nodes and these two types of nodes are denoted as sensor nodes in the paper, represented as green nodes and blue nodes in Fig. 1, respectively. The sensing node can not only sense and form the measurement reports of the monitored components, but also forward the measurement reports of other nodes. The forwarding node can only forward the measurement reports to the controller. We assume that each cluster has a unique cluster ID and each node has a unique node ID. Sensor nodes that measure or forward measurement reports have a limited computation and communication capability and limited energy resources. Sensor nodes lack tamper-resistance hardware and can be compromised by attackers. Fig. 1 shows the example of system model, where node v_1, v_2, v_3 and v_4 obtain the measurement reports of monitored component j and send them to the controller via v_4. Similarly, u_4 sends the measurement report of monitored component i to the controller through multiple forwarding nodes. We can see that v_1 can serve as a forwarding node to transmit the measurement reports of monitored component i.

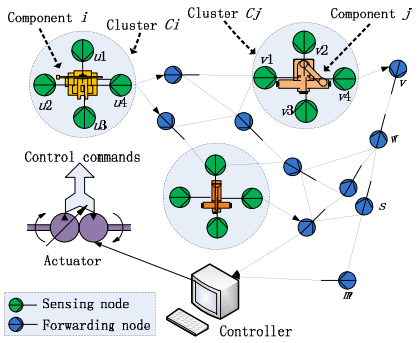

Fig. 1. System Model

We assume that the attacker can compromise sensor nodes, including both the sensing nodes and forwarding nodes. Once a node is compromised, the secret information stored in the node becomes visible to the attacker. The attacker can inject false measurement reports to the controller via the compromised nodes. This causes the controller to estimate wrong system states [8] and send wrong control commands to the actuators, posing the dangerous threats to the system. The false reports also consume lots of network and computation resources and shorten the lifetime of CPNS. We assume that the controller is well protected and the attacker could only obtain the authentication information through compromising sensor nodes. We also assume that there is a reliable node initialization after nodes being deployed, and the attacker cannot compromise or damage any node during the initialization phase.

III. Our Approach

A. Basic Idea

The basic idea of our scheme is described below. PCREF uses polynomials instead of MACs to verify reports, and can mitigate the node impersonating attack against legitimate nodes. By organizing a set of sensing nodes into a cluster, where nodes are responsible for the same monitored components, PCREF assigns the corresponding authentication polynomial and check polynomial to each sensor node. These polynomials stored in nodes are bundled with node ID and derived by the primitive polynomials assigned from a primitive polynomial pool. Different primitive polynomials will be used in different clusters through the cluster-based primitive polynomial assignment. This increases the resilience of our scheme to

TABLE I
NOTATION

N_s :	Total number of sensing nodes in the system.
N_f :	Total number of forwarding nodes in the system.
n :	The number of nodes used to measure a monitored component, i.e., the number of nodes in a cluster.
T :	The number of nodes used to endorse a measurement report.
C_i :	Cluster ID.
u, v, w :	Node ID.
CH_i :	Cluster-head of cluster C_i.
c_i^j :	Local ID of node in cluster C_i,$1 \leq j \leq n$.
MAP :	Message authentication polynomial.
l :	Number of primitive polynomials in the globe primitive polynomial pool.
$f_i(x, y, z)$:	Primitive polynomial of cluster C_i with parameter x, y and z.
$auth_i^u$:	Authentication polynomial of cluster C_i stored in node u with parameter y and z.
$verf_i^v$:	Check polynomial of cluster C_i stored in node v with parameter x and z.
R_t :	Communication radius of sensor nodes.
E :	Measurement from sensor nodes.
r :	Sensing report generated by sensing node.
R :	Integrated measurement report merged by a number of sensing report r and sent to the controller by the cluster-head.
$H(\cdot)$:	Hash function stored in each node.
$Time$:	Timestamp of measurement report.
K :	Master key used to establish the key shared between neighboring nodes.
K_C :	Master key used to establish the cluster key.
E :	Measurements of monitored component measured by sensing nodes.

the increasing number of compromised nodes without relying on the node localization and static data dissemination routes. The authentication polynomial stored in each node is used to endorse the report of local component measurement while the check polynomial is used to validate the received reports. Each sensing node stores the authentication polynomial of the local cluster and stores the check polynomial of other clusters with a pre-defined probability P. Each forwarding node stores the check polynomial of each cluster with the same probability P. Our scheme also uses T-authentication framework similar to [15], [13], [16], [9], [12], [17], [20], i.e., a legitimate report shall be authenticated by T nodes from the same cluster. Forwarding node could verify the report only if it shares the authentication information with the source node.

Our scheme consists of the following two key components: (i) *authentication information management* is used to assign the key, authentication polynomial, check polynomial, and local ID of sensing nodes, and (ii) *data security management* is used to detect and filter the false measurement reports. The detailed description of these two components will be described in the next two subsections. The notations used in this paper are shown in Table I.

B. Authentication Information Management

In our scheme, we assume that the node initialization phase is reliable and secure, and the attacker cannot compromise and launch attacks at any node during this phase. The node initialization of PCREF consists of four steps, including: *cluster organization*, *authentication information assignment*, *key generation*, and *local ID assignment*, which will be described below.

1) Step 1: Cluster organization: In our scheme, each monitored component is monitored by n sensing nodes organized as a cluster. We can deploy n sensing nodes close to the monitored component. Those nodes communicate with each other and each node stores the node IDs of its neighbors to organize the cluster. The node ID is stored in the node before being deployed. We assign the cluster ID to each cluster and each sensing node stores its cluster ID, e.g., each sensing node in cluster i stores the cluster ID C_i in its memory.

2) Step 2: Authentication information assignment: In this stage, the network designer initializes all nodes and the network with the following parameters:

$$\{K, K_C, f(x, y, z), T, H(\cdot)\},$$

where K_C are the master keys, $f(x, y, z)$ is the element from a set of primitive polynomials, T is the threshold, and $H(\cdot)$ is the hash function. For each sensing node u, the designer stores the master K_C and the hash function $H(\cdot)$ in u. He also reads the cluster ID C_i stored in the node, and computes the authentication polynomial of cluster C_i for u by

$$auth_i^u(y, z) = \alpha f_i(u, y, z), \tag{1}$$

where x, y, z are parameters, u is the sensing node in cluster C_i, $f_i(x, y, z)$ is the primitive polynomial of cluster C_i, $auth_i^u(y, z)$ is the authentication polynomial of cluster C_i for u, and $\alpha \in \{2, 2^2, 2^3, 2^4\}$. Note that the designer randomly chooses the value of α while computing the authentication polynomial. Thus, no other party knows the value of α except the designer. The usage of α is to increase the resilience of our scheme to the number of compromised nodes. After the computation, the $auth_i^u(y, z)$ is stored in node u. The designer then computes the check polynomials for node u. For each cluster $j (j \neq i)$, the designer computes the check polynomial $verf_j^u(x, z)$ with a probability P, and stores these check polynomials in node u by

$$verf_j^u(x, z) = \beta f_j(x, u, z), \tag{2}$$

where $verf_j^u(x, z)$ is the check polynomial of cluster C_j stored in node $u, \beta \in \{2^5, 2^6, 2^7, 2^8\}$ and it plays the same role as α. In fact, β can be any value of 2^t, where t is a positive integer. To make a reasonable memory storage for sensor node, we set its range as $\{2^5, 2^6, 2^7, 2^8\}$. For each forwarding node w, the designer computes the check polynomials of all the clusters with probability P and stores hash function $H(\cdot)$, and the check polynomials in node w.

Note that in this stage, we use the cluster-based primitive polynomial assignment mechanism to ensure that the primitive polynomial assigned for one cluster is different from others. The use of the ID-based polynomial generation ensures that the authentication polynomial and the check polynomial stored in one node are different from other nodes. Our scheme leads to a high resilience to node impersonation attacks because the authentication information of one cluster has no impact on another cluster. The formation of authentication information

in our scheme does not require node localization, which is required by [13], [16], [9], [12].

3) Step 3: Key generation: In this stage, by using the master key K_C, each sensing node generates the cluster key. Using Fig.1 as an example, the cluster key of cluster C_i stored in u_1 is

$$K_{C_i} = F_C(K_C|CH_i), \tag{3}$$

where $|$ is denoted as the connection operation, $F_C(\cdot)$ is the cluster key generation function. Notice that K_C is erased once generation is completed. With the assumption that attackers cannot compromise node during initialization-phase, no one knows K_C even if attackers compromise nodes in filtering-phase. Hence, K_C and K_{C_i} are not globally known because K_{C_i} is generated by K_C, and the nodes outside cluster cannot decrypt E from report.

4) Step 4: Local ID assignment: In this stage, each sensing node is assigned a local ID by its cluster-head. Cluster-head CH_i sends the local ID assignment message to every nodes u in its cluster,

$$CH_i \longrightarrow u : (CH_i|u|c_i^j), \tag{4}$$

where $|$ is denoted as the connection operation, c_i^j is the local ID of u assigned by CH_i. After receiving the message, node u stores the local ID and sends the following response message,

$$u \longrightarrow CH_i : (u|c_i^j). \tag{5}$$

The cluster-head collects all response messages and determines whether the n local IDs are assigned to the different cluster nodes. Note that n is the number of sensing nodes monitoring the physical component. If the cluster-head finds a local ID not being assigned, it repeats the above process and assigns it to a node. By using the local ID assignment, the cluster-head assigns the local ID to all nodes in the cluster and ensures that for any $j (j \in [1, n])$, there is a c_i^j stored in one and only one cluster node. With the use of the local ID, our scheme can detect the false measurement reports sent by the compromised cluster-head and increase the resilience to false data injection attacks.

C. Data Security Management

The data security management of PCREF consists of the *sensing report generation*, *measurement report generation and transmission*, *en-route filtering*, and *controller authentication*. We will describe those steps below.

1) Step 1: Sensing report generation: Each sensing node measures the data of the monitored component and generates the sensing report r, which consists of the encrypted measurement, node ID, local ID, and MAP. Sensing nodes generate different MAPs for the same measurement using its node ID and locally stored authentication polynomial. For example, node u first calculates

$$z = H((E)_{K_{C_i}}), \tag{6}$$

where E is the measurements of its monitored component, $H(\cdot)$ is the hash function stored in node u and K_{C_i} is the cluster key for the cluster, to which u belongs, and then node u generates MAP for the measurement by

$$MAP = auth_i^u(y, z) = \alpha f_i(u, y, H((E)_{K_{C_i}})), \tag{7}$$

where $auth_i^u(y, z)$ is the authentication polynomial stored in node u. As we can see, the MAP is a polynomial, which has only one parameter y and is bundled with node ID. After combining with the check polynomial stored in the intermediate nodes along the route, MAP can be used to detect the correctness of forwarded measurement reports. To reduce the communication overhead of forwarding the measurement reports, PCREF only adds the coefficients of each MAP into report.

After generating the sensing report r, every sensing node sends the report to the cluster-head CH_i. The report r is constructed by

$$r = \left((E)_{K_{C_i}}|u|c_i^j|MAP\right), \tag{8}$$

where $|$ is denoted as the connect operation, u is node ID, c_i^j is local ID of u and MAP is the authentication information of measurements E generated by node u and can be derived from Equation (7). The measurements E is encrypted through the cluster key K_{C_i}, and thus any node outside the cluster cannot decrypt E from the report.

2) Step 2: Measurement report generation and transmission: After receiving all sensing reports generated by the sensing nodes, cluster-head randomly chooses T reports from them and merges these T measurement reports to an integrated measurement report R and sends it to controller. The measurement report R is formed by,

$$\begin{aligned} R = \Big((E)_{K_{C_i}}|C_i|u_1|\cdots|u_T|c_i^{j_1}|\cdots|c_i^{j_T} \\ |auth_i^{u_1}(y, H((E)_{K_{C_i}}))|\cdots|auth_i^{u_T}(y, H((E)_{K_{C_i}}))|Time\Big), \end{aligned} \tag{9}$$

where C_i is the cluster ID, $auth_i^{u_m}(y, H((E)_{K_{C_i}}))$ is denoted as MAP generated by node u_m, $Time$ is the timestamp, and other notations are the same as ones in r.

After generating R, cluster-head sends it to the controller through the intermediate nodes along the route. Because of the broadcast nature of wireless communication, the sensing nodes in the same cluster also eavesdrop the measurement report sent by cluster-head and determine that

- T local IDs included in R satisfies $c_i^{j_1} \neq c_i^{j_2} \neq \cdots \neq c_i^{j_T}$ and $1 \leq c_i^{j_1}, c_i^{j_2}, \cdots, c_i^{j_T} \leq n$, where n is the number of sensing nodes for monitoring the component.
- The information attached in R is the same as ones stored in each sensing node with local ID $c_i^{j_m}$.

If the above two conditions are not satisfied, sensing nodes will send the warning massage to the first intermediate node and request it to drop report R. Otherwise, no warning message will be sent and this means that R is the true integrated measurement report of the monitored component. In this way, PCREF can drop the false measurement report forged by the compromised cluster-head effectively at the first

intermediate node along the forwarding route. The process of the measurement report delivery is shown in Appendix A of our technical report [14].

Note that the cluster-head and ordinary sensing node can also serve as the forwarding nodes. If the attacker compromises the ordinary sensing node or cluster-head, he can forge and send the false measurement reports of other components to the controller via the compromised nodes. Then the above approach cannot detect and filter this false report. To deal with this issue, PCREF adopts the en-route filtering mechanism described in the next step.

3) Step 3: En-route filtering: By leveraging the polynomial-based message authentication introduced in [18], PCREF conducts the en-route filtering on false measurement reported from the compromised nodes while the existing approaches [18] cannot do so. In PCREF, the measurement report is transmitted to the controller hop-by-hop. The intermediate node, which does not have the corresponding check polynomial of the cluster, where the measurement is originally generated (e.g., cluster ID is attached in the report), forwards the measurement report to the next node along the route. The intermediate node, which has corresponding check polynomial, determines whether the received measurement report R is false through validating the following conditions:

- *Condition 1*: The timestamp $Time$ attached in R is fresh.
- *Condition 2*: T MAPs attached in the report are different and are generated by the sensing nodes in the corresponding cluster, where cluster ID is claimed in the report.
- *Condition 3*: T MAPs can be verified by the corresponding check polynomial stored in the intermediate node.

If the above three conditions are not satisfied, the intermediate node will drop the measurements report. Otherwise, the measurement report will be forwarded. The *Condition 1* uses the timestamp $Time$ to detect the replayed false report. To verify the *Condition 2* and *Condition 3*, the intermediate node first calculates the values of $A_i^{u_m,v}$ and V_i^{v,u_m} with the value of z calculated by Equation (6),

$$\begin{aligned} A_i^{u_m,v} &= auth_i^{u_m}(v, H\left((E)_{K_{C_i}}\right)), \\ &= \alpha f_i(u_m, v, H\left((E)_{K_{C_i}}\right)), \qquad (10) \\ V_i^{v,u_m} &= ver f_i^v(u_m, z) = \beta f_i(u_m, v, z), \\ &= \beta f_i(u_m, v, H\left((E)_{K_{C_i}}\right)), \qquad (11) \end{aligned}$$

where v is the node ID of the intermediate node, u_m is the sensing node ID carried in the report, $auth_i^{u_m}(y, H\left((E)_{K_{C_i}}\right))$ is the MAP generated by u_m and in the report, $ver f_i^v(x, z)$ is the check polynomial of cluster C_i stored at node v. As we can see from Equation (10) and (11), $q = V_i^{v,u_m}/A_i^{u_m,v} = \beta/\alpha$, and only if q belongs to $\{2^1, 2^2, \cdots, 2^7\}$, the MAP generated by node u_m can be determined as valid one. The reason is that α belongs to $\{2, 2^2, 2^3, 2^4\}$, and β belongs to $\{2^5, 2^6, 2^7, 2^8\}$. Note that, this approach could lead to the negative rate with $1/(2^{l-3})^T$ for the forwarded false measurement report, where l is the number of median coefficients of each MAP. However, according to the security analysis of PCREF described in Appendix C shown in our technical report [14], the successful rate of forging MAPs is very small and can be generally ignored, when no knowledge of authentication information is revealed to the attacker.

The intermediate node verifies all MAPs in the report via the same check polynomial, which ensures that T MAPs are derived by the same primitive polynomial. Through the cluster-based primitive polynomial assignment, PCREF ensures that only the sensing nodes in the same cluster have the same primitive polynomial. By using the same check polynomial, the *Condition 2* can be satisfied; Meanwhile, the *Condition 3* can be satisfied only if all MAPs carried in the report are valid. If all three conditions are satisfied, the intermediate node forwards the measurement report. Otherwise, the measurement report will be filtered.

The detailed process of en-route filtering is shown in in Appendix A of our technical report [14]. The cluster-based primitive polynomial assignment provides the different primitive polynomials to each cluster, and the authentication polynomial stored in the node is bundled with node ID. Hence, the attacker could not obtain the authentication polynomial stored in a legitimate sensing node in one cluster by compromising the node in another cluster. Hence, PCREF has the resiliency condition: the attacker can successfully forge the false measurement report and send it to controller without being filtered by intermediate nodes only if he compromises more than T sensing nodes for the same component. Obviously, satisfying the resiliency condition makes PCREF solve the T-threshold problem appeared in [15], [20] and achieves a high resilience against the number of compromised nodes. In addition, PCREF does not require the node localization and node association, which are required by LBRS and LEDS to select intermediate nodes and establish the authentication key for them to verify the measurement reports. Hence, PCREF has unique benefits in comparison with the existing schemes.

4) Controller authentication: After receiving the measurement report, the controller validates it in the same way as the intermediate node does. Because the controller stores all primitive polynomials, it can validate all received measurement reports and filter the false measurement reports, which bypass the detection of intermediate nodes. If the report is confirmed as a legitimate one, the controller decrypts the measurements from the report, and estimates the state of monitored component and sends the commands to the actuators to control the operation of physical systems. Because of having the complete authentication information, the controller is the last defense in the system and can detect and filter all the false measurement reports forged by the attacker.

IV. Security Analysis

We now analyze the security of our proposed scheme to authenticate the measurement reports. The performance metrics include (i) *filtering efficiency* is defined as the probability of false data to be filtered out within a number of hops, (ii) *attack resilience* is defined as the ratio of compromised components (clusters) vs. the total components (clusters) in the system, and (iii) *filtering capability* is defined as the average forwarded

hops of false measurement reports, i.e., the average hops that the false measurement report will be forwarded before being detected and filtered. The energy cost analysis of PCREF can be found in Appendix A and the analytical data is shown in Fig. 5. Note that, with no knowledge of authentication information revealing to the attacker, the probability of the MAP of measurement to be successfully forged by the attacker can be largely ignored. The detailed proof can be found in Appendix C of our technical report [14].

A. Filtering Efficiency

In PCREF, each intermediate node stores the check polynomial for a cluster with the predefined probability P. After receiving the measurement report, the intermediate node verifies all T MAPs carried in the report to detect and filter out the false measurement reports. Hence, when $x < T$, the probability of a false measurement report filtered by the intermediate node is $P_f = P$, where the x is the number of compromised nodes in the cluster. Let the probability of a false measurement report filtered after being forwarded h hops be P_h and the probability of a false measurement report filtered within h be P'_h. We have

$$P_h = (1 - p_f)^{h-1} \cdot P_f, \tag{12}$$

$$P'_h = 1 - (1 - P_f)^h. \tag{13}$$

The filtering efficiency of PCREF can be represented by P'_h defined as the probability of false measurement report to be filtered within a number of hops. Obviously, the greater the probability, the better the filtering efficiency becomes. Numerical data in Fig.2 show the filtering efficiency vs. the forwarded hops when $P = 0.1, 0.2, 0.5$. As we can see, PCREF can filter the most of false measurement reports en-route, and thus it can detect and filter false measurement reports effectively. The higher the value of P, the smaller the forwarded hops is required to filter the false measurement reports. The reason is that the probability of the check polynomial stored at the intermediate node increases as P increases. However, each intermediate node stores $(Ns/n) \cdot P$ check polynomials, and the smaller P can reduce the storage overhead of the intermediate node.

B. Resilience to Attack

Because of the derivation from different primitive polynomials bundled with node ID, the authentication polynomial in compromised node could not be used to launch the node impersonating attack against the legitimate node. According to the filtering rules of PCREF, the measurement report is false if more than one MAP carried in the report is not derived from the primitive polynomial assigned to the cluster, where the report generates. Hence, to forge a "legitimate" false measurement report, the attacker shall compromise several sensing nodes and obtain T or more authentication polynomials of the attached cluster. In PCREF, to obtain T authentication polynomial of the target cluster, the attacker shall consider the cases listed below and our analysis will be based on these two cases:

- *Case 1*: Use the check polynomial and authentication polynomial stored in compromised sensing nodes and forwarding nodes to derive the primitive polynomial of the target cluster and derive enough valid authentication polynomials via the derived primitive polynomial.
- *Case 2*: Compromise more than T sensing nodes in the target cluster and obtain authentication polynomials stored in them.

The resilience of PCREF in Case 1. In PCREF, each cluster is assigned a primitive polynomial and sensing nodes in this cluster generate the authentication polynomial and intermediate nodes generate the check polynomial associated with this cluster by the primitive polynomial. Hence, the attacker can derive the desired authentication polynomials if he obtains the primitive polynomial assigned to the targeted cluster. Nevertheless, in PCREF, because no one knows the primitive polynomials except the controller, the attacker could not obtain the primitive polynomial of the target cluster directly. We now analyze the possibility that the attacker derives the primitive polynomial of the target cluster through the authentication polynomials and check polynomials stored at the compromised nodes.

We assume that the targeted cluster is cluster C_i, its primitive polynomial is $f_i(x, y, z)$, and the highest power of parameters x, y, z are n_x, n_y, n_z, respectively. Hence, $f_i(x, y, z)$ can be represent by,

$$f_i(x, y, z) = \sum_{l=0}^{n_x} \sum_{m=0}^{n_y} \sum_{s=0}^{n_z} C^i_{lms} x^l y^m z^s, \tag{14}$$

where C^i_{lms} is the coefficient and the number is $(n_x+1)\cdot(n_y+1)\cdot(n_z+1)$. Hence, the authentication polynomial of cluster C_i stored in the sensing node u and the check polynomial of cluster C_i stored in intermediate node v can be represented by

$$auth^u_i(y, z) = \alpha \sum_{l=0}^{n_x} \sum_{m=0}^{n_y} \sum_{s=0}^{n_z} C^i_{lms} u^l y^m z^s = \sum_{m=0}^{n_y} \sum_{s=0}^{n_z} D^i_{ms} y^m z^s, \tag{15}$$

$$verf^v_i(x, z) = \beta \sum_{l=0}^{n_x} \sum_{m=0}^{n_y} \sum_{s=0}^{n_z} C^i_{lms} x^l v^m z^s = \sum_{l=0}^{n_x} \sum_{s=0}^{n_z} G^i_{ls} u^l z^s, \tag{16}$$

where u is the sensing node in cluster C_i and v is the intermediate node outside cluster C_i, D^i_{ms} and G^i_{ls} are coefficients and can be fixed if the attacker compromises node u and v. For any value z_0 of z, the attacker derives the above equations by $(x, y, z) = (u, v, z_0)$.

$$\alpha \sum_{l=0}^{n_x} \sum_{m=0}^{n_y} \sum_{s=0}^{n_z} C^i_{lms} u^l v^m z_0^s = \sum_{m=0}^{n_y} \sum_{s=0}^{n_z} D^i_{ms} v^m z_0^s, \tag{17}$$

$$\beta \sum_{l=0}^{n_x} \sum_{m=0}^{n_y} \sum_{s=0}^{n_z} C^i_{lms} u^l v^m z_0^s = \sum_{l=0}^{n_x} \sum_{s=0}^{n_z} G^i_{ls} u^l z_0^s. \tag{18}$$

Equation (17) is derived by the authentication polynomial and with (n_x+2) unknown parameters in it (i.e., n_x+1 parameters regarding to the coefficients of x in $f_i(x, y, z)$ and one parameter α), while Equation (18) is derived by the check

polynomial and with (n_y+2) unknown parameters in it (i.e., n_y+1 parameters regarding to coefficients of y in $f_i(x, y, z)$ and one parameter β). Only if n_x+2 or n_y+2 unknown parameters in Equation (17) or (18) are obtained, the attacker can derive the primitive polynomial of the target cluster. As we can see, because the Equation (17) has different unknown parameters from Equation (18), they cannot be combined to derive the primitive polynomial. If the attacker compromises N_c nodes and obtains t_s authentication polynomials and t_f check polynomials of the cluster to be attacked, he can generate t_s Equation (17) and t_f Equation (18). However, from the attacker's viewpoint, t_s Equation (17) have $n_x + 1$ unknown coefficients of x in $f_i(x, y, z)$ and t_s unknown αs as different authentication polynomial has different α. Hence, these t_s equations have (t_s+n_x+1) unknown parameters, and the number of unknown parameters increases as the number of equations increases and it always has $ts <$(t_s+n_x+1). According to the calculus, we know that the attacker cannot derive all unknown parameters in Equation (17) and then obtain the desired primitive polynomial. Because of the similar reasons, the attacker cannot derive the desired primitive polynomial from t_f from Equation (18) as well.

According to the analysis above, it is difficult for the attacker to derive the desired primitive polynomial from the obtained authentication polynomials and check polynomials. Hence, only if the attacker compromises T sensing nodes in the attacked cluster, he can successfully forge the false measurement report of the targeted component.

The resilience of PCREF in Case 2. When N_{cs} sensing nodes are compromised by the attacker, the probability of a cluster with x compromised sensing nodes becomes

$$P_{\{x\}} = \frac{\begin{pmatrix} n \\ x \end{pmatrix}\begin{pmatrix} N_s - n \\ N_{cs} - x \end{pmatrix}}{\begin{pmatrix} N_s \\ N_{cs} \end{pmatrix}}, \tag{19}$$

where N_s is the total number of sensing nodes, and n is the number of sensing nodes monitoring a given component in the cluster. The probability of a cluster having T or more compromised sensing nodes, namely the compromised cluster, can be represented by

$$P_{ca} = \sum_{x=T}^{n} P_{\{x\}} = \sum_{x=T}^{n} \frac{\begin{pmatrix} n \\ x \end{pmatrix}\begin{pmatrix} N_s - n \\ N_{cs} - x \end{pmatrix}}{\begin{pmatrix} N_s \\ N_{cs} \end{pmatrix}}. \tag{20}$$

Note that P_{ca} is also defined as the compromised clusters (i.e., compromised components) ratio over the network, that is, the ratio of monitored components where the corresponding measurement reports authenticity will be manipulated by the attacker by compromising N_{cs} sensing nodes. Fig. 3 shows the ratio of compromised clusters vs. the number of compromised sensing nodes, where there are 10000 nodes in the system. As we can see, greater threshold T, the lower the rate of compromised cluster. When T is 5, the ratio of compromised clusters approaches to a low value even though a large number of sensing nodes are compromised. Hence, PCREF achieves a high resilience to the increased number of compromised nodes.

Notice that our approach achieves a higher resilience to the attacks against perturbation-polynomials-based schemes discussed in [4]. First, we divide the nodes into clusters, and different clusters are assigned with different primitive-polynomials. A node stores multiple bivariate-polynomials derived from different primitive-polynomials. To derive a target-cluster's primitive-polynomial, attackers have to compromise enough nodes, which store bivariate-polynomial derived from target-cluster's primitive-polynomial. Because node stores target-cluster's bivariate-polynomial in low probability, attackers need to compromise a large number of nodes with random-capture-attacks. Second, even if attackers compromised enough nodes, it is difficult to derive attacked cluster's primitive-polynomial. We introduce the two parameters α and β (unknown to attackers) to increase resilience. Even if attackers know α and β's ranges, computation overhead of ciphering them is high and increases as α and β's size and polynomial-degree (e.g., if α and β are 16-bits, and bivariate-polynomial is degree-3 polynomials, computation-overhead is $\Omega(2^{16*(3+1)})$). Third, even if attackers can derive target cluster's primitive-polynomials, the effect can be limited within the cluster-area without affecting other cluster areas.

C. Filtering Capacity

We now analyze the filtering capacity of PCREF vs. the number of compromised nodes (including both sensing nodes and forwarding nodes). Recall that to measure the filtering capability, we define the average forwarded hops before the false measurement report being filtered as $\overline{h}$ and we have

$$\overline{h} = E(h) = \sum_{i=1}^{h_{max}} h_i P_{h_i}^c, \tag{21}$$

where $P_{h_i}^c$ is the probability where given a number of compromised nodes in the system, the false measurement report is filtered after being forwarded at least h_i hops. Obviously, the smaller average forwarded hops leads to the greater filtering capacity. We assume that the average forwarded hops from sensors to the controller is h_{max}, and the forged measurement reports can be forwarded h_{max} hops and arrives at controller if the cluster, where the measurement report is generated, has more than $T-1$ compromised sensing nodes. When $x(x < T)$ sensing nodes are compromised in the attacked cluster, the attacker should forge $T-x$ MAPs in each forged measurement report.

Recall that these false reports could be filtered by the intermediate nodes, which have the corresponding check polynomial. However, if the attacker compromises the intermediate nodes, these forged reports can escape the filtering and be forwarded to controller. Hence, only if the desired cluster has less than T compromised sensing nodes and at least one intermediate node, which has the check polynomial of desired cluster, is in its routing path and is not compromised, the forged measurement report from the targeted cluster will be filtered. Based on the conditions stated above, we analyze

the filtering capacity of PCREF in terms of the number of compromised sensing nodes.

When the attacker compromises N_c sensor nodes, including N_{cs} sensing nodes and N_{cf} forwarding nodes, the probability of cluster with less than T compromised sensing nodes is $\sum_{x=0}^{T-1} P_{\{x\}}$, and the probability that an intermediate node filters the false measurement report is P_{fc}. We assume that the probabilities of a node being the sensing node or the forwarding node is 0.5 and we have

$$P_{fc} = P \cdot \left(1 - \frac{1}{2}\left(\frac{N_{cs}}{N_s} + \frac{N_{cf}}{N_f}\right)\right), \tag{22}$$

where P is the probability of intermediate node with corresponding check polynomial, N_s is the number of sensing nodes, and N_f is the number of forwarding nodes. Hence, the probability that the false measurement report is filtered after forwarding h_i hops is denoted as $P^c_{h_i}$ and we have

$$P^c_{h_i} = \sum_{x=0}^{T-1} P_{\{x\}} \cdot (1 - P_{fc})^{h_i - 1} \cdot P_{fc}, \tag{23}$$

where $h_i \in [1, h_{max} - 1]$. Recall that there are the following two ways for the attacker to successfully inject the false measurement reports to the controller: (i) compromising T or more sensing nodes within the cluster, and (ii) compromising less than T sensing nodes within the cluster and compromising all intermediate nodes storing the check polynomial of the target cluster. Hence, the probability that the false measurement reports are forwarded h_{max} hops becomes

$$P_{h_{max}} = \sum_{x=T}^{n} P_{\{x\}} + \sum_{x=0}^{T-1} P_{\{x\}} \cdot (1 - P_{fc})^{h_{max}-1}. \tag{24}$$

As we can see from the numerical data shown in Fig. 4, even though a large number of nodes is compromised, PCREF can still filter them within a few forwarded hops. For example, given a network with 10000 nodes and P is 0.1 and 0.2, PCREF can filter the false measurement reports within 10 hops and 7 hops, respectively, even if 20% sensor nodes are compromised. Hence, PCREF can filter false reports in small number of forwarded hops, and achieves greater filtering capacity, because the filtering capacity increases as the average forwarded hops decreases.

V. Performance Evaluation

In this section, we show both the analytical and simulation results of PCREF in comparison with SEF [15], LBRS [13], GRSEF [16] and LEDS [16] in terms of the filtering efficiency, filtering capability, and resilience to the number of compromised nodes (e.g., N_{cs} sensing nodes are compromised in the system).

A. Simulation Setup

In our evaluation setting, we consider the scenario of 100 components and 1000 sensing nodes (i.e., each component is monitored by 10 sensing nodes). To make the scenario suitable for LBRS and LEDS, we consider that components form a $10 * 10$ array and are deployed in a $[0, 500m] \times [0, 500m]$ area uniformly, i.e., each component is deployed in a square with side length of $50m$. The controller is located at $(0, 0)$. The cluster used in PCREF, responsible for monitoring the component is similar to the cell used in LBRS and LEDS. We also set $T = 5$, $n = 10$, and the node communication radius $R_t = 50m$, which are the typical values in [15], [13], [9], [16]. For LBRS, the beam width b is set to $150m$ [13]. For SEF, GRSEF, LBRS and PCREF, the key sharing probability or the check polynomial sharing probability q is 0.2. In each simulation, a number of sensing nodes are randomly selected as the compromised nodes.

The filtering efficiency is evaluated by the ratio of filtered false measurement reports within a forwarded hops. Filtering capability is evaluated by the average forwarded hops, where the false measurement report is forwarded until being filtered. The resilience can be evaluated by the ratio of total compromised components vs. the total number of components, that is, the probability of components those measurement reports can be successfully forged by the attacker. For PCREF, LEDS and LBRS, the ratio of compromised components can be obtained based on the definition. For GRSEF, we check whether the attacker can forge a valid report from each grid-point by dividing the area into virtual grids. The resiliency of SEF is evaluated by the times for obtaining T keys successfully from distinct partitions by the attacker vs. total number of experiments. Note that, For the MAP forging successful ratio mentioned in section IV, it just to prove that the attacker could not forge a legitimate MAP with no knowledge of authentication information revealing to him, could not need to be simulated. Hence, we don't simulate it in this section.

Numerical results are derived from the formulae in [15], [13], [16], [9] and theoretical analysis in Section IV. Each simulation is repeated 100 times and the simulation result shows the average value over 100 times. All simulations in this paper are completed by Matlab6.5.

B. Evaluation Results

1) Filtering Efficiency: Fig. 5 shows the analytical results of the ratio of filtered false measurement reports vs. the number of forwarded hops of SEF, PCREF, LEDS, GRSEF and LBRS. Fig. 6 shows the simulation results of those schemes, when 100 sensing nodes (i.e., 10% the total number of nodes) are compromised by the attacker. As we can see, both the analytical and simulation results constantly show that PCREF has the highest ratio of filtered false measurement reports and SEF achieves the worst performance. The filtering efficiencies of GRSEF, LBRS, and LEDS are always lower than that of PCREF.

2) Filtering Capability: Fig. 7 and Fig. 8 show the average hops that the measurement reports are forwarded vs. the number of compromised sensing nodes in term of analysis and simulation, respectively. As we can see, when the number of compromised sensing nodes increases, the average forwarded hops of PCREF increases slowly while others increase rapidly. When the number of compromised sensing nodes is less

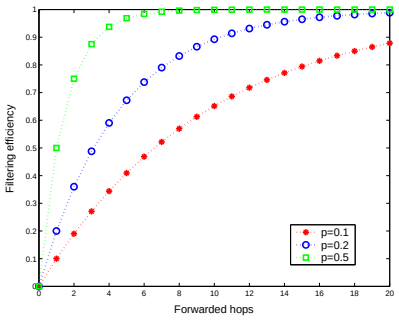

Fig. 2. Filtering efficiency vs. Forwarded hops

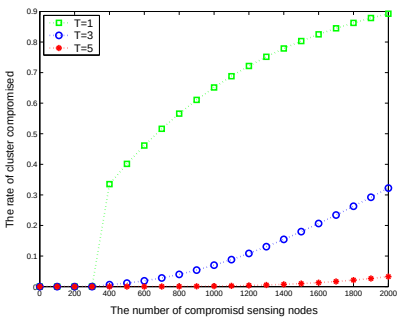

Fig. 3. Compromised component ratio vs. Number of compromised sensing nodes

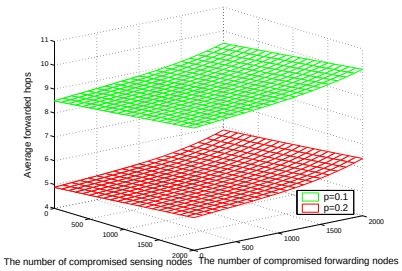

Fig. 4. Average forwarded hops vs. Number of compromised nodes

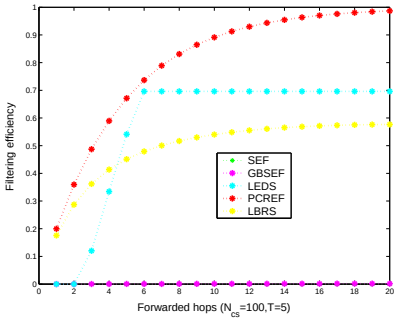

Fig. 5. Filtering efficiency vs. Forwarded hops (Analysis)

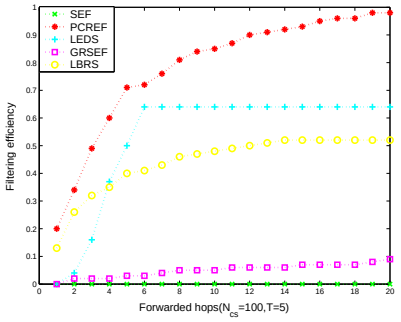

Fig. 6. Filtering efficiency vs. Forwarded hops (Simulation)

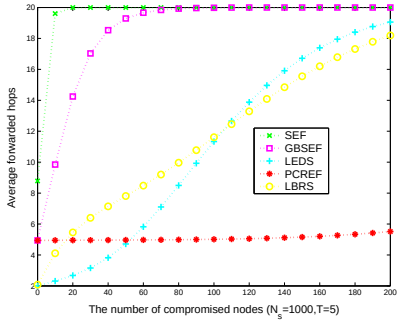

Fig. 7. Average forwarded hops (analysis)

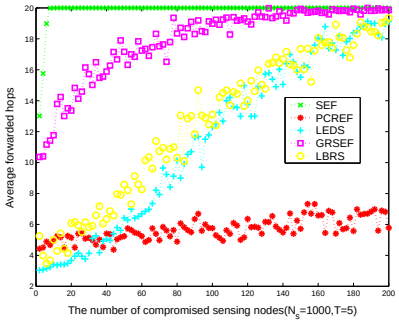

Fig. 8. Average forwarded hops (simulation)

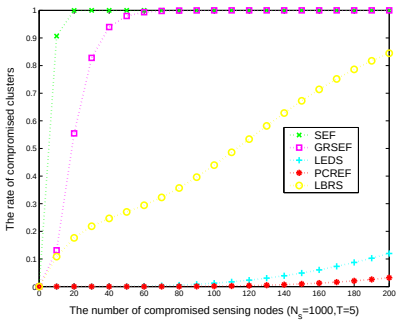

Fig. 9. Resilience vs. max number of compromised sensing nodes with 200 (Analysis)

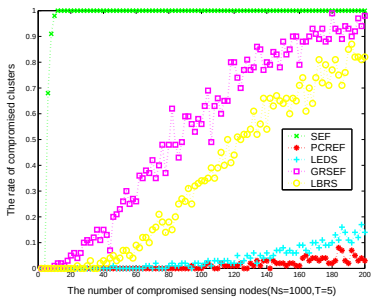

Fig. 10. Resilience vs. max number of compromised sensing nodes with 200 (Simulation)

than 30 (i.e., 3% of the total number of nodes), the average forwarded hops of PCREF is one hop larger than that of LBRS and LEDS. The reason is that LBRS and LEDS rely on the static routes and achieve higher filtering efficiency within first several forwarded hops. However, the specific routes make LEDS and LBRS vulnerable, because once the attacker damages the route (e.g., jamming), the measurement report could not be transmitted to the controller on time, posing the degradation of system performance.

3) Resilience: Fig. 9 show the analytical results of the ratio of successful times in SEF and the percentages of compromised components (cells or clusters) of GRSEF, LBRS, LEDS, and PCREF given the total number of compromised sensing nodes of 200 and 500, respectively. Fig. 10 show the simulation results. From Fig. 9, we can see that because of T-threshold limitation, the ratio of compromised components of SEF approaches to 100% when more than 10 nodes (i.e., 1% of the total number of nodes) are compromised, and the ratio of PCREF and LEDS are obvious less than these of LBRS and GRSEF. As we can see, PCREF achieves the highest resilience to the increased number of compromised sensing nodes without relying on the static routes and node localization.

VI. Related Works

To mitigate the false data injected by attackers in sensor networks, a number of en-route filtering schemes have been developed [15], [13], [16], [20], [12], [9], [17], [7]. SEF [15] and IHA [20] are the first two proposed schemes to carry out

en-route filtering of false reports. For example, SEF divides nodes into n groups via non-overlapping key partitions and nodes in the same partition share authentication key with a probability, only the intermediate node sharing authentication key with source nodes can validate the report. Both SEF and IHA have the T-threshold limitation. LBRS [13] and LEDS [9] avoid the T-threshold limitation through the cell-based report generation and location-ware key generation techniques. However, those schemes introduce the node localization and node association based on the statically configured routes or conforming bean model. This makes longer time to stabilize with a large amount of energy resource consumption. CCEF [12] introduced the commutative cipher instead of sharing symmetric key to en-route filter false data. EAB [7] introduces authentication bitmap, instead of using MAC as the proof to verify the reports. It also relies on the statical data dissemination routes, which is vulnerable to attacks. As we can see, the existing schemes either have T-threshold limitation, or rely on node localization, node association and statically configured routes, which limited their usage to CPNS for monitoring mobile physical components and systems. Because en-route filtering problem was originally studied in 2004 [20], [15], we only list the most related literatures after 2004.

The polynomial-based technique has been used for applications [10], [19], [18]. Work in citePercom and [19] proposed the perturbation number and perturbation polynomial-based techniques for compromise-resilient key management. [18] proposed a perturbation polynomial-based technique to authenticate messages. This scheme detects the report modified en-route effectively, but cannot deal with the forged reports injected by comprised nodes, because the attacker can obtain the authentication polynomial stored in the compromised node and successfully forge valid authentication information attached in the reports. Different from the existing research, we develop the polynomial-based technique to conduct the en-route filtering against false data injection attacks.

VII. Conclusion

In this paper, we proposed a *Polynomial-based Compromised-Resilient En-route Filtering scheme (PCREF)*, which can filter false data effectively and achieve high resilience to the number of compromised nodes without relying on static routes and node localization. PCREF adopts polynomials for endorsing measurement reports to improve resilience to the node impersonating attacks. Each node stores two types of polynomials: authentication polynomial and check polynomial derived by primitive polynomial, and used for endorsing and verifying the measurement reports, respectively. We develop techniques to effectively manage authentication information and filter out the false measurement reports. Via both theoretical analysis and simulation experiments, our data show that our schemes achieves better filtering capacity and resilience to the large number of compromised nodes in comparison with the existing schemes.

Acknowledgement

The work was supported in part by the following funding agencies in China: National 973 Basic Research Program of China under grant No. 2011CB302801, the Fundamental Research Funds for the Central Universities (xjj2011078), Xi'an industrial applied technology research project (CXY1017(4)). This work was also supported in part by US National Science Foundation (NSF) under grants CNS 1117175, 1116644, 0942113, 0958477 and 0943479. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] *Cyber Physical Networks(CPN) Research Lab*. http://cpn.berkeley.edu/.
[2] *Northeast Blackout of 2003*. http://en.wikipedia.org/wiki/Northeast_Blackout_of_2003.
[3] *CPS Week*. http://www.cpsweek2010.se/, 2010.
[4] M. Albrecht, C. Gentry, S. Halevi, and J. Katz. Attacking cryptographic schemes based on perturbation polynomials. In *Proc. of the ACM CCS*, 2009.
[5] A. Albur and A. G. Exposito. *Power System State Estimation: Theory and Implementation*. CRC Press.
[6] X. Chen, K. Makki, K. Yen, and N. Pissinou. Sensor network security: A survey. *IEEE Communications Surveys and Tutorials*, 11(2):52–73, 2009.
[7] Y.-S. Chen and C.-L. Lei. Filtering false messages en-route in wireless multi-hop networks. In *Proc. of IEEE WCNC*, 2010.
[8] Y. Liu, M. K. Reiter, and P. Ning. False data injection attacks against state estimation in electric power grids. In *Proc. of the 16th ACM conference on Computer and communications security*, 2009.
[9] K. Ren, W. Lou, and Y. Zhang. Leds: Providing location-aware end-to-end data security in wireless sensor networks. *IEEE Transactions on In Mobile Computing (TMC)*, 7(5):585–598, 2008.
[10] N. Subramanian, C. Yang, and W. Zhang. Securing distributed data storage and retrieval in sensor networks. In *Proc. of the 27th IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2007.
[11] V.Liberatore. Networked cyber-physical systems: An introduction. In *Proc. of Mobisensors'07*, January 2007.
[12] H. Yang and S. Lu. Commutative cipher based en-route filtering in wireless sensor networks. In *Proc. of 60th IEEE VTC*, 2004.
[13] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh. Toward resilient security in wireless sensor networks. In *Proc. of the 6th ACM MobiHoc*, 2005.
[14] X. Yang, J. Lin, P. Moulema, W. Yu, X. Fu, and W. Zhao. A novel en-route filtering scheme against false data injection attacks in cyber-physical systems. In *http://pages.towson.edu/wyu/ylmyfzReportMay2011.pdf*, 2011.
[15] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route filtering of injection false data in sensor networks. In *Proc. of the 23th IEEE INFOCOM*, 2004.
[16] L. Yu and J. Li. Grouping-based resilient statistical en-route filtering for sensor networks. In *Proc. of the 28th IEEE INFOCOM*, 2009.
[17] Z. Yu and Y. Guan. A dynamic en-route filtering scheme for data reporting in wireless sensor networks. *IEEE/ACM Transactions on Networking (ToN)*, 18:150–163, 2010.
[18] W. Zhang, N. Subramanian, and G. Wang. Lightweight and compromise-resilient message authentication in sensor networks. In *Proc. of the 27th IEEE INFOCOM*, 2008.
[19] W. Zhang, M. Tran, S. Zhu, and G. Cao. A random perturbation-based pairwise key establishment scheme for sensor networks. In *Proc. of the 8th ACM MobiHoc*, 2007.
[20] S. Zhu, S. Setia, S. Jajodia, and P. Ning. An interleaved hop-by-hop authentication scheme for filtering of injection false data in sensor networks. In *Proc. of the 25th IEEE Symposium on Security and Privacy (S&P)*, 2004.