Integration of Secure In-Network Aggregation and System Monitoring for Wireless Sensor Networks

Bo Sun*, Xing Jin
Dept. of Computer Science
Lamar University
Beaumont, TX USA 77710
Email: {bsun, xingjin}@cs.lamar.edu

Kui Wu
Dept. of Computer Science
University of Victoria
BC, Canada V8W 3P6
Email: wkui@cs.uvic.ca

Yang Xiao
Dept. of Computer Science
University of Alabama
Tuscaloosa, AL USA 35487
Email: yangxiao@ieee.org

Abstract—Secure in-network aggregation in Wireless Sensor Networks (WSNs) is a necessary and challenging task. In this paper, we address this research problem from an intrusion detection perspective. We propose that system monitoring modules, which provide one of the most important functionalities for WSNs, should be integrated with intrusion detection modules. Under this architecture, we first propose an Extended Kalman Filter (EKF) based mechanism to detect false injected data. Specifically, by monitoring behaviors of nodes’ neighbors and using EKF to predict the state (the real in-network aggregated value), we aim at setting up the normal range of the neighbor’s future transmitted aggregated values. We illustrate how we use EKF to create effective local detection mechanisms. Using different aggregation functions (average, sum, max, and min), we analyze how to obtain the threshold in theory. We then illustrate how our proposed local detection approach can work together with the system monitoring module to differentiate between malicious events and emergency events. We conduct simulations to evaluate performance of local detection mechanisms, including false positive rate and detection rate, under different aggregation functions.

I. INTRODUCTION

In-network aggregation has been proven to be an essential technique to reduce communication overhead and save energy for Wireless Sensor Networks (WSNs). However, sensor nodes could be easily compromised and can inject arbitrarily falsified values, leading the base station to accept significantly wrong information. Most of existing in-network aggregation protocols depend on prevention-based mechanisms, in which different keying and authentication mechanisms are used to provide the effective protection for aggregated data. So far, there is very little work that aims at addressing the secure in-network aggregation problem from an intrusion detection perspective.

In this paper, we propose that System Monitoring Modules (SMM) should be integrated with Intrusion Detection Modules (IDM) in the context of WSNs. In reality, WSNs are often deployed to monitor important emergency events, such as forest fire. This integration can facilitate the classification between malicious events and important emergency events. Under this system setting, we first propose an Extended Kalman Filter (EKF) [1] based mechanism to detect false injected data. Specifically, by monitoring the behavior of nodes’ neighbors and using EKF to predict the state (the real in-network aggregated value), we aim at setting up the normal range of the neighbor’s future transmitted aggregated value. In reality, this is challenging because of the potential high packet loss rate [2], harsh environments, sensing inaccuracy, time asynchrony between children and the parent node, etc. Utilizing a state-space model [1], EKF-based mechanism is suitable for WSN nodes because this mechanism can address those incurred uncertainties in WSNs and be implemented in a lightweight manner. We illustrate how we use EKF to address these challenges and compute the accurate estimate of aggregated values, based on which a normal range can be deduced. Promiscuously overheard values are then compared with a locally computed normal range to decide whether they are significantly different. We also analyze how to decide the threshold under different aggregation functions (average, sum, max, and min).

We conduct simulations to evaluate and analyze performance of local detection mechanisms, including false positive rate and detection rate, under different aggregation functions. Our implementation of the Extended Kalman Filter on MICA2 motes demonstrates that our proposed scheme is practical on resource stringent hardware.

II. NETWORK MODEL, ASSUMPTIONS AND MOTIVATIONS

Fig. 1(a) is one example of an aggregation tree in WSNs. In Fig. 1(a), A, B, C, and D perform sensing tasks, obtain values and transmit them to their parent node H. H aggregates (min, max, sum, average, etc) the received values from A, B, C, D and transmit the aggregated value further up to node K. The same is true for operation E, F, G → I → J and M, N → L → J. The base station collects all these data and, if necessary, can transmit them across the Internet.

WSNs are often deployed to monitor emergency events like forest fire. We assume that the majority of nodes around some unusual events are not compromised. We also assume that falsified data transmitted by the compromised node is significantly different from the state (the real value, for example, the real monitored average temperature) in that the falsified data can effectively disrupt the aggregated value. If the falsified value sent out by compromised nodes is only slightly different from the true value, the attacker cannot cause

---

*This research was supported in part by the Texas Advanced Research Program under grant 003581-0006-2006.
significant negative impacts on WSNs. Therefore, this makes the attack meaningless.

III. SECURE AGGREGATION

Our proposed system is equipped with two modules: Intrusion Detection Module (IDM) and System Monitoring Module (SMM). The functionality of the IDM is to detect whether the monitored nodes are malicious internal nodes, while the functionality of the SMM is to monitor important emergency events. Note that SMM is a necessary component for most of WSN applications. IDM and SMM need to be integrated with each other to work effectively. Relying on local detection alone in the context WSNs is not desirable because each node has only very limited information available. Furthermore, sensor nodes are prone to failure. These make it very difficult to differentiate between emergency events sent by good nodes and malicious events. Under our system setting, whenever IDM or SMM detects some abnormal events, they need to request the collaboration of more sensor nodes around the events in order to make a final decision.

A. Challenges

Many challenges exist when we try to predict the normal range of the in-network aggregated value in a lightweight manner. First, it is difficult to achieve the real aggregate values because of the many sources of potential uncertainty. WSNs suffer from a high packet loss rate. For example, based on [2], in an in-building environment, with 62 motes deployed with the granularity of one mote per office, at a low load of 0.5 packet per second, there is around 35% of the links whose packet loss is worse than 50% at the Medium Access Control (MAC) layer. Second, for the aggregation protocol, the lack of time synchronization among child and parent nodes makes the aggregation node use different sets of values for aggregation. Third, the complexity of existing aggregation protocols contributes to the challenges of modeling in-network aggregated values. Furthermore, individual sensor readings are subject to environmental noise. To demonstrate this, we set up a simple one-hop WSN testbed, in which node A periodically transmits the sensed value to a base station. Node A consists of a MICA2 mote and a MTS310 sensor board. In a lab setting, we measure the collected data, as shown in Fig. 2(a).

We conduct a further experiment to demonstrate the uncertainty of the aggregation function. We use four sensors to send their sensed temperature values to an aggregation node B. B periodically computes the average of the received values. The average values are illustrated in Fig. 2(b).

B. Extended Kalman Filter based Monitoring

We use an Extended Filter (EKF) based approach to predict and estimate the future values of nodes’ neighbors.

1) Extended Kalman Filter: Based on the state-space model, Kalman Filter (KF) [1] addresses the general problem of trying to estimate the state of a dynamic system perturbed by Gaussian white noise, using measurements that are linear functions of the system state, but corrupted by additive Gaussian white noise. Extended with the linear estimation theory, Extended Kalman Filter (EKF) can be applied to many nonlinear applications by linear approximation of the effects of small perturbations. By setting a proper process model and measurement model for specific WSN applications and utilizing time update and measurement update equations to recursively process the data, we can use EKF to obtain an accurate estimate of the state [1].

In our case, the state represents the real value to be measured, for example, the real temperature value monitored by WSNs. Because the real temperature value is perturbed by various uncertainties, it is impossible for aggregators to obtain the state. In another words, the aggregation node can only obtain the measured value and estimate the real value.

Aggregation nodes calculate aggregated values periodically. Therefore, we adopt a Discrete-Time Extended Kalman Filter (EKF) in which the system state is estimated at a discrete set of times $t_k$, $k = 0, 1, \ldots$. These discrete times correspond to time instants at which values are measured and states are estimated.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_k$</td>
<td>state - the real value at time $t_k$</td>
</tr>
<tr>
<td>$\hat{x}_{k-1</td>
<td>k}$</td>
</tr>
<tr>
<td>$z_k$</td>
<td>measurement (measured value) at time $t_k$</td>
</tr>
<tr>
<td>$h$</td>
<td>function relating $z_k$ to $x_k$</td>
</tr>
<tr>
<td>$Q_k$</td>
<td>variance of $h$ at time $t_k$</td>
</tr>
<tr>
<td>$P_k$</td>
<td>a priori estimate error at time $t_k$</td>
</tr>
<tr>
<td>$P_k^*$</td>
<td>a posterior estimate error at time $t_k$</td>
</tr>
<tr>
<td>$K_k$</td>
<td>Kalman gain at time $t_k$</td>
</tr>
</tbody>
</table>
2) System Dynamic Model: Table I lists the notations. If we treat the real aggregated values as a dynamic process, the process model governs the evolution of the process. The measurement model describes the measurement value.

Process Model : \( x_{k+1} = F(x_k) + w_k \)

Difference function \( F \) relates the state at the time step \( t_k \) to the state at the time step \( t_{k+1} \). Obviously, \( F(x_k) \) is application dependent and its discussion is beyond the scope of this paper. For example, if WSNs are deployed to monitor the average temperature that decreases or increases gradually in a time period, \( F(x_k) \) can be set to \( x_k + c \), where \( c \) is a constant depending on the environment. In practice, because of the complex nature of monitored phenomenon, \( F(x_k) \) may take complicated forms. Therefore, necessary simplifications of \( F(x_k) \) may be needed after careful analysis of the applications. It is also possible that for a given application, different forms of \( F(x_k) \) need to be used to characterize the state change. One possible solution is that the base station can broadcast a new form of \( F(x_k) \) over the network to adjust the process model.

As a dynamic system, the state of any applications has some variations, which are reflected in \( w_k \). \( w_k \) is the process noise at time \( t_k \), which is usually modeled as a normal random variable. We further assume that \( w_k \) follows normal distribution \( N(0, Q) \), where \( Q \) denotes the variance of \( w_k \) and is a constant parameter. Note that \( Q \) may also be broadcasted over the network by the base station to adapt to the changing environments.

Measurement Model: \( z_k = h(x_k) + v_k = x_k + v_k \). \( z_k \) is the measured value at time \( t_k \). Take Fig. 1(a) as the example, suppose that node \( I \) sends out an aggregated value \( z_k \) at time \( t_k \), node \( E, F, G \) can overhear this value. \( x_k \in \mathbb{R} \) is the real number set) is the state to be monitored at time \( t_k \) and represents the real aggregated value of the area that the current aggregation node covers. \( v_k \) is the measurement noise and can represent the various uncertainties in WSNs. Again, for a specific application, we assume \( v_k \) follows normal distribution \( N(0, R) \). \( R \) denotes the variance of \( v_k \). Note that \( R \) can also be adjusted by the base station.

3) System Equations: Time Update - State Estimate Equations: \( \hat{x}_{k+1} = F(\hat{x}_k) \)

Time Update - Error Project Equations: \( P_{k+1}^- = \frac{\partial F}{\partial x}|_{x=\hat{x}_k} P_k^+ \frac{\partial F}{\partial x}|_{x=\hat{x}_k}^T + Q_k \)

Measurement Update - Kalman Gain: \( K_{k+1} = P_{k+1}^- (P_{k+1}^- + R_k)^{-1} \)

Measurement Update - Error Covariance update: \( P_{k+1}^- = (I - K_{k+1} H) P_{k+1}^- = (1 - K_{k+1} H) P_{k+1}^- \)

Measurement Update - Estimate update with the measurement \( z_{k+1} \):

\[
\hat{x}_{k+1} = \hat{x}_{k+1}^- + K_{k+1} (z_{k+1} - \hat{x}_{k+1}^-)
\]

The time update equations are responsible to predict the state (\( \hat{x}_{k+1}^- \)) and estimate error (\( P_{k+1}^- \)) in order to obtain a prior estimate at the next time step (\( t_{k+1} \)). Applying a first order Taylor series approximation to \( F(x) \), \( \frac{\partial F}{\partial x}|_{x=\hat{x}_k} \) denotes the value of the first-order partial derivative of \( F \) with respect to \( x \) at \( x = \hat{x}_k \). Because in our case, the state is a scalar variable, therefore, \( \frac{\partial F}{\partial x}|_{x=\hat{x}_k} \) is also a scalar variable.

The measurement update equations are responsible for incorporating \( (z_{k+1}) \) into the a priori estimate to obtain a statistical optimal a posterior estimate (\( \hat{x}_{k+1} \)).

EKF can provide an accurate prediction of neighbor’s future aggregated values. To illustrate this point, using the network topology shown in Fig. 1(b) and the average as an example aggregation function, we plot the real, measurement, and estimate value using system equations described in Section III-B3. Specifically, we simulate an environment whose real temperature value is increasing smoothly and perturbed with a Gaussian white noise. The sensor nodes, \( N_i \) in Fig. 1(b), sense the temperature values (to simulate the sensing inaccuracy, the sensed temperature value is the real temperature value perturbed by some random noise) and periodically transmit them to its parent node, node \( N \). The computed average value at \( N \) is the measured value. Estimate value denotes the value estimated using EKF and can be computed by each \( N_i \).

The result is illustrated in Fig. 3(a). We can see that although many kinds of uncertainty exist, EKF-based approach can still provide a desirable prediction of the real value. The estimate error \( P_{k+1}^+ \) can stabilize quickly. Fig. 3(b) illustrates one example run of \( P_{k+1}^+ \) with an arbitrary initial value.

C. Local Detection

A child node monitors its neighbor’s behavior by predicting the normal range of the neighbor’s future aggregated value and compare this range with overheard values. The creation of the normal range should be centered on the estimated value. Our location detection algorithm is illustrated in Algorithm (1).

In Algorithm (1), node \( A \) can overhear \( B \)’s transmission \( z_{k+1} \) at time \( t_{k+1} \). \( A \)’s purpose is to decide whether \( z_{k+1} \) is abnormal or not. After estimating the state \( \hat{x}_{k+1}^- \) at time \( t_k \), node \( A \) can predict node \( B \)’s transmitted value \( \hat{x}_{k+1}^- \) at time \( t_{k+1} \). At time \( t_{k+1} \), \( A \) overhears \( B \)’s transmitted value, \( z_{k+1} \), and compares \( \hat{x}_{k+1}^- \) with \( z_{k+1} \) to decide whether \( B \) is acting normally or not. If the difference between \( \hat{x}_{k+1}^- \) and \( z_{k+1} \) (denoted as \( Diff \) in Algorithm (1)) is larger than \( \Delta \), a predefined threshold, \( A \) raises an alert on \( B \). Otherwise, \( A \) thinks that \( B \) functions normally. We will provide the analysis of \( \Delta \) in Section III-F.

D. Collaboration between IDM and SMM

Local detection alone suffers from a high false positive rate. Therefore, for WSNs, Intrusion Detection Modules (IDM) and
Local Detection Algorithm

Assumption Nodes $A$ and $B$, $A$ can overhear $B$’s transmission. $A$ thinks that $B$ is a normal node at and before $t_k$
Input $\hat{x}_{k+1}$ transmitted by $B$ and overheard by $A$
Output Whether $A$ raises an alert on $\hat{x}_{k+1}$

Procedure
1. At time $t_k$, $A$ computes $\hat{x}_k^+$ based on Eq.(1) (note that $\hat{x}_k^+$ is stored in node $A$);
2. $A$ computes $\hat{x}_{k+1}^+$ based on $\hat{x}_k^+$;
3. $A$ computes $Diff = |\hat{x}_{k+1}^+ - z_{k+1}|$;
4. if ($\Delta < Diff$) then
   5. $A$ raises an alert on $B$;
   6. else
   7. $A$ thinks that $B$ functions normally;
8. end if

![Collaboration between IDM and SMM to Differentiate Malicious Events from Emergency Events.](image)

System Monitoring Modules (SMM) need to integrate with each other to work effectively. Our proposed local detection module resides in IDM. When node $A$ raises an alert on node $B$ because of some event $E$ ($E$ could be a significantly different value sent out by $B$), $A$ starts the further investigation on $E$ by collaborating with the existing SMM. WSNs are usually densely deployed to collaboratively monitor some events. To save energy, some sensor nodes are periodically scheduled to sleep. Based on this, node $A$ can request SMM to wake up the sensor nodes (denoted as co-detectors in Fig. 4) around $B$ and request from these nodes their opinions on the behavior of $E$. Because the majority of sensor nodes around the investigated event $E$ are not compromised, if the majority of sensor nodes think that event $E$ may happen, $A$ makes a decision that $E$ is triggered by some emergency events.

On the other hand, if $A$ finds that the majority of sensor nodes think that event $E$ should not happen, $A$ then thinks that $E$ is triggered by either a malicious node or a faulty yet good node. $A$ can continue to request SMM to wake up those nodes around $E$ and their opinions about the behavior of $E$. If $A$ keeps finding that the majority of sensor nodes think that event $E$ should not happen, $A$ then decides that $E$ is malicious.

There may exist efficient approaches for SMM to collect information from those sensor nodes around event $E$. Wang et al. [8] proposed an efficient approach to construct a dominating tree to cover all the neighbors of the suspect (node $B$ in our example). Their approach tried to include the node which has more neighbor co-detectors (nodes that can provide useful information). An efficient dominating tree can then be constructed to collect information.

E. Cost Analysis

The computation of $F(x)$ and $\frac{\partial F}{\partial x}$ at $x = \hat{x}_k^+$ depends on the application. Given $F(x)$ taken the form of $\delta x + c$, where $\delta$ and $c$ are constants, the incurred computation costs include two multiplications, four additions, two subtractions, two divisions. Also, note that if $R$ and $Q$ are constant, the estimate error becomes constant soon. This will further reduce the computation overhead. As for the memory consumption, to predict the value at time $t_{k+1}$ for one neighbor, each sensor node only needs to keep the optimal estimate $\hat{x}_k^+$. Other memory consumption include the storage of $Q$, $R$, estimate error, and kalman gain.

F. Threshold Analysis

Let $U$ denote the variance of the uncertainty in the aggregated values. Based on the “three-sigma” control limits in Shewchart control charts, $\Delta$ can be set to $3\sqrt{U}$. Under conditions where $Q$ and $R$ are constant, the EKF estimation error $P_k^+$ stabilizes quickly and then remains constant. We can further adjust the normal range based on $P_k^+$.

In Fig. 1(b), suppose that $E[v_i] = \mu_i$ and $var(v_i) = \sigma_i^2$. Suppose that with a probability $0 < p < 1$ ($p$ denotes the probability that $N$ does not receive the packet from the children because of packet loss, packet collision, etc), the packet on each link is lost. Let random variable $X$ denote the aggregated value. We analyze the variance of $X$ considering different packet loss probability.

1) Average: If there are $m$ ($m > 0$ and $m < n$) packets lost, its probability times the corresponding aggregated value is $\frac{1}{m!} \sum_{n=1}^{m} v_i$. (We omit the deduction process because of the limited space.) Then $X$ is defined as:

$$X = \left\{ \begin{array}{ll}
\sum_{i=1}^{n} v_i & \text{with probability } (1-p)^n \\
\sum_{i=1}^{n} v_i & \text{with probability } (1-p)^{n-1}p \\
\vdots & \vdots
\end{array} \right.$$

For applications where $v_i$s are similar when aggregating (for example, the monitored temperature over an area does not have much difference), denote $E(v_i) = \mu_i$ and $var(v_i) = \sigma_i^2$. Therefore, considering the impact of packet loss, collision, etc, $E[X] = \sum_{m=0}^{n} (1-p)^m \sum_{i=1}^{n} v_i$. $E(X^2) = \sum_{m=0}^{n} (1-p)^m \sum_{i=1}^{n} v_i^2$. (We omit the deduction procedure because of the space limitation). We have $var(X) = E[X^2] - E^2[X]$. $U$ can be set to $\sqrt{var(X)}$.

2) Sum: For the sum, similarly, define $X$ as:

$$X = \left\{ \begin{array}{ll}
\sum_{i=1}^{n} v_i & (1-p)^n \\
\sum_{i=1}^{n} v_i & (1-p)^{n-1}p \\
\sum_{i=1}^{n} v_i & (1-p)^{n-2}p^2 \\
\vdots & \vdots
\end{array} \right.$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.
when there are \( m \) (\( m \geq 0 \) and \( m \leq n \)) packets lost, its probability times the corresponding aggregated value is 
\[
\frac{(n-1)!}{m!(n-m-1)!} (1-p)^m p^{n-m} \sum_{i=1}^{n} v_i. 
\]
For applications where the monitored values \( v_i \) are similar, \( E(X) = \sum_{m=0}^{\infty} (1-p)^m p^n \frac{m}{\prod_{i=n-m}} \mu. \) \( E(X^2) = \sum_{m=0}^{\infty} ((C^n_m - C^{n-1}_m)\mu^2 + \sigma^2) + 2(C^n_m - 2C^{n-1}_m + C^{n-2}_m \frac{n(n-1)}{2})], U \) can then be computed.

3) \( \text{Min/Max} \): The analysis of Max is similar to the analysis of \( \text{Min} \). We only provide the analysis of \( \text{Min} \). For the set of values \( v_i \) (\( 1 \leq i \leq n \)), without the loss of generality, by sorting these \( v_i \), we obtain \( v_1 < v_2 < \ldots < v_n \). Then \( X \) is:
\[
X = \{ v_1 (1-p) \quad v_2 p(1-p) \quad v_3 p^2(1-p) \quad \ldots \}
\]
For applications where \( v_i \) are similar, assume that the probability density function (pdf) for \( v_i \) is \( f(x) \) and the cumulative distribution function (cdf) for \( v_i \) is \( F(x) \). Assume that \( v_i \) follows normal distribution, \( f(x) \) and \( F(x) \) can be determined. Based on the order statistic, we can compute the pdf of \( X \) as \( f_{X} = n! [1-F(x)]^{n-1} f(x) \), where \( r \) denotes the \( r \)th order statistic. For min aggregation, \( r = 1 \). Based on this, we can further compute \( E(X) \) and \( E(X^2) \). \( U \) can be derived.

G. Discussion

In practice, we can further adjust \( \Delta \). For example, if an aggregation node is closer to the base station, its information is more important for the base station. Therefore, if the application can tolerate more errors, when a node is closer to the base station, we can adopt a relatively larger threshold.

Our proposed EKF-based mechanism is general in that it can be used not only to monitor malicious injected values, but also to detect emergency events. For example, the EKF conditions can be stored in a table and then distributed throughout the network. After the dissemination of the table, each node can then check its readings against the predictions and transmit those satisfied predictions back to the base station. This can save the bandwidth and energy significantly.

IV. PERFORMANCE EVALUATION

A. Simulation

1) Simulation Setup: We use Fig. 1(b) as an abstract network model to evaluate Algorithm (1). For each link, we use different packet loss ratio, 0.1, 0.25, and 0.5, respectively. For each packet loss ratio, we use two sets of \( v_i \). In the first set, we make all \( v_i \) randomly distributed between one predefined range [\( \text{min}, \text{max} \)]. This is to simulate WSN applications that are deployed to monitor an area which has same attributes. In the second set, we set different \( v_i \) randomly distributed between different [\( \text{min}, \text{max} \)] pairs. That is, \( \forall i, 1 \leq i \leq n, [\text{min}, \text{max}] \) pairs satisfy \( \text{min}_1 < v_i < \text{max}_i, \text{min}_{i+1} = \text{min}_i + T, \) and \( \text{max}_{i+1} = \text{max}_i + T. \) \( T \) is a constant parameter.

We set node \( N \) as a compromised node in order to simulate the attack data. Obviously, the more different the attack data is from the normal data, the easier the attack data can be detected. We introduce the concept degree of damage - \( D \). \( D \) is defined as the difference between the attack data and the normal data. Take Fig. 1(b) as the example, suppose that the correct aggregated value by node \( N \) is \( C \) and the malicious aggregated value sent out by \( N \) is \( M \). Then \( D = |C - M| \).

Under the same set of simulation parameters, we first obtain 5000 normal data items and 5000 malicious data items. We then plot the Receive Operating Characteristic (ROC) curves.

2) Simulation Results:

a) Average: Fig. 5(a) and Fig. 5(b) demonstrate the ROC curve for the first set of \( v_i \). We can see that the performance is not impacted much by the packet loss ratio. Because all the \( v_i \) fall in the same range, therefore, the loss of several \( v_i \) does not have big impacts on the overall performance. Comparing Fig. 5(b) with Fig. 5(a), we can see that with the increase of the degree of damage in the attack data, the performance improves (the curves moves upper-left).

Fig. 5(c) and Fig. 5(d) plot the ROC curve for the second set of \( v_i \). In this situation, when the packet loss ratio becomes larger, the detection rate becomes smaller, while the false positive rate keeps roughly the same value. With the increase of the packet loss ratio, the simultaneous loss of smaller \( v_i \) increases, this leads to an increase of the measurement value \( \hat{x}_k \). Based on Equation (1), \( \hat{x}_k \) becomes larger. \( \hat{x}_{k+1} \) then becomes larger given our simulated environment. Based on Algorithm (1), \( \text{Diff} \) becomes smaller. This leads to an decreased detection rate.

b) Sum: Fig. 6(a) and Fig. 6(b) plot the ROC curve for the first set of \( v_i \). Given a threshold, the larger the packet loss ratio, the larger the false positive rate and the
becomes smaller. When the packet loss ratio is larger, the detection rate decreases a little. With the increase of the packet loss ratio, the measurement value at time \( k \) becomes larger, while the false positive ratio keeps roughly the same value. When the packet loss ratio is larger, the measurement value at time \( k \) becomes smaller. Based on Equation (1), \( x_k^+ \) becomes smaller, \( x_{k+1}^- \) then becomes smaller given our simulated environment. Based on Algorithm (1), \( Diff \) becomes larger. This leads to an increased detection rate.

Fig. 6(c) and Fig. 6(d) plot the ROC curve for the second set of \( v_i \). We have the same observation as those of Fig. 6(a) and Fig. 6(b). The reasons are similar.

c) Min: Fig. 7(a) and Fig. 7(b) plot the ROC curve for the first set of \( v_i \). With the increase of the packet loss ratio, the detection rate decreases a little. With the increase of the packet loss ratio, the measurement value at time \( k \) becomes larger. Based on Equation (1), \( x_k^+ \) becomes larger, \( x_{k+1}^- \) then becomes larger given our simulated environment. Based on Algorithm (1), \( Diff \) becomes smaller. This leads to an increased detection rate. Similarly, we can see that with the increase of \( D \), the overall performance increases.

Fig. 7(c) and Fig. 7(d) plot the ROC curve for the second set of \( v_i \). The larger the packet loss ratio, the worse the performance becomes. The reason is similar to that when \( v_i \) randomly distributed between one predefined range \([min, max]\).

d) Max: Fig. 7(a) and Fig. 7(b) plot the ROC curve for the first set of \( v_i \). Given a threshold, when the packet loss ratio becomes larger, the detection rate becomes larger, while false positive ratio keeps roughly the same value. The explanations are very similar to those of the min aggregation. In Fig. 8(c) and Fig. 8(d), the performance of the larger packet loss ratio is roughly the same as that of the lower packet loss ratio. Compared to Fig. 7(a) and Fig. 7(b), this is because of the impact of different sets of \( v_i \).

B. Experiment

We implemented the EKF on MICA2 motes in TinyOS and evaluated its performance in terms of code size, RAM size, and execution time. For one MICA2 mote, EKF needs 1182 Bytes in code size and 35 Bytes in RAM size. It takes about 0.051S to execute one iteration of EKF on MICA2 mote. This illustrates that the incurred overhead can be handled by the current generation of network sensors efficiently.

V. RELATED WORK

Very little work has considered secure aggregation problems from the intrusion detection point of view. Wagner [3] used statistical estimation to design more resilient aggregation schemes against malicious data injection attacks. A mathematical framework was presented to formally evaluate the security of different aggregation algorithms. Hu et al. [4] tackled the problem of information aggregation in which one node was compromised. Their protocol might be vulnerable if both the parent node and the child node are compromised. Yang et al. [5] proposed SDAP, a secure hop-by-hop data aggregation protocol based on the principles of divide-and-conquer and commit-and-attest. Przydatek et al. [6] proposed the aggregate-commit-prove framework to design secure data aggregation protocols. Wu et al. [7] proposed a Secure Aggregation Tree (SAT) to detect and prevent cheating in WSNs, in which the detection of cheating is based on the topological constraints in the constructed aggregation tree.

VI. CONCLUSIONS

In this paper, we first propose that the Intrusion Detection Modules (IDM) and the System Monitoring Modules (SMM) should work together in order to provide intrusion detection capabilities for WSNs. We then propose an Extended Kalman Filter (EKF) based approach to detect the false injected data. We illustrate how we use EKF to address the various uncertainties in WSNs and create an effective local detection mechanism. Simulation results demonstrate that our proposed schemes can achieve desirable performance to detect false injected data.

REFERENCES