2-2014

Attack-Resilient Sensor Fusion

Radoslav Ivanov
*University of Pennsylvania*, rivanov@seas.upenn.edu

Miroslav Pajic
*University of Pennsylvania*, pajic@seas.upenn.edu

Insup Lee
*University of Pennsylvania*, lee@cis.upenn.edu

Follow this and additional works at: [http://repository.upenn.edu/cis_papers](http://repository.upenn.edu/cis_papers)

Part of the [Computer Engineering Commons](http://repository.upenn.edu/cis_papers), and the [Computer Sciences Commons](http://repository.upenn.edu/cis_papers)

Recommended Citation


This paper is posted at ScholarlyCommons. [http://repository.upenn.edu/cis_papers/768](http://repository.upenn.edu/cis_papers/768)

For more information, please contact repository@pobox.upenn.edu.
Attack-Resilient Sensor Fusion

Abstract
This work considers the problem of attack-resilient sensor fusion in an autonomous system where multiple sensors measure the same physical variable. A malicious attacker may corrupt a subset of these sensors and send wrong measurements to the controller on their behalf, potentially compromising the safety of the system. We formalize the goals and constraints of such an attacker who also wants to avoid detection by the system. We argue that the attacker’s capabilities depend on the amount of information she has about the correct sensors’ measurements. In the presence of a shared bus where messages are broadcast to all components connected to the network, the attacker may consider all other measurements before sending her own in order to achieve maximal impact. Consequently, we investigate effects of communication schedules on sensor fusion performance. We provide worst- and average-case results in support of the Ascending schedule, where sensors send their measurements in a fixed succession based on their precision, starting from the most precise sensors. Finally, we provide a case study to illustrate the use of this approach.

Disciplines
Computer Engineering | Computer Sciences

Comments

This conference paper is available at ScholarlyCommons: http://repository.upenn.edu/cis_papers/768
Abstract—This work considers the problem of attack-resilient sensor fusion in an autonomous system where multiple sensors measure the same physical variable. A malicious attacker may corrupt a subset of these sensors and send wrong measurements to the controller on their behalf, potentially compromising the safety of the system. We formalize the goals and constraints of such an attacker who also wants to avoid detection by the system. We argue that the attacker’s capabilities depend on the amount of information she has about the correct sensors’ measurements. In the presence of a shared bus where messages are broadcast to all components connected to the network, the attacker may consider all other measurements before sending her own in order to achieve maximal impact. Consequently, we investigate effects of communication schedules on sensor fusion performance. We provide worst- and average-case results in support of the Ascending schedule, where sensors send their measurements in a fixed succession based on their precision, starting from the most precise sensors. Finally, we provide a case study to illustrate the use of this approach.

I. INTRODUCTION

With the proliferation of sensing technology available to modern Cyber Physical Systems (CPS), the problem of performing effective sensor fusion is more important than ever. For example, modern automotive systems have multiple ways of estimating speed; combining their sensor data to provide more accurate estimates to the controller can have a significant impact on the system’s performance and reliability. In addition, having diverse sensors with different accuracy and reliability decreases the system’s dependence on a particular sensor and increases its robustness to environmental disturbances (e.g., variations in terrain in automotive CPS).

As the system’s autonomy increases, so does the concern about its security. In modern vehicles, a malicious attacker may deceive the controller into performing a dangerous action by altering the measurements of some sensors [1], [2]. Depending on the attacker’s goal and capabilities, the consequences may range from minor disturbances in performance to crashes and loss of human lives. Consequently, performing attack-resilient sensor fusion is essential for the safety of such systems.

The problem of sensor fusion has drawn a lot of attention in recent years. Usually, sensor fusion methods assume a probabilistic model of each sensor and combine their measurements accordingly [3]. In particular, each sensor gives a numeric measurement that is corrupted by noise with a known distribution (e.g., Gaussian). In an alternative approach, an interval is constructed around each sensor measurement, containing all points that may be the true value (e.g., set membership methods [4]). One of the first works with this viewpoint [5] also considers the case where some of the sensors might be faulty (i.e., providing intervals that do not contain the true value); the author provides worst-case analysis when the number of faulty intervals can be bounded. An extension of [5] relaxes the worst-case guarantees in favor of obtaining more precise fused measurements [6]. In addition, intervals can be assumed to have a predefined distribution on the true value so that again statistical analysis can be performed [7]. Finally, one can use sensor information not only to aid control but also for fault detection [5], [8].

In this work, we consider the problem of attack-resilient sensor fusion for abstract sensors where each sensor’s measurement is converted to an interval by the controller. The width of the interval reflects the accuracy of the sensor – a larger interval means less confidence in provided measurements. This is a very general scenario as it does not make any assumptions about system dynamics nor about the distribution of sensor measurement noise. Instead, the intervals are constructed based on the manufacturer’s sensor specifications and implementation guarantees (e.g., sensor precision, implementation jitter). We propose the use of the sensor fusion algorithm outlined by Marzullo [5], which produces a fusion interval based on an assumed number of faulty sensors (see Section II).

Furthermore, we introduce the notion of a compromised (i.e., attacked) sensor, which may not be faulty but still acts in a detrimental way to the system. Assuming that the attacker knows the algorithm implemented by the system, her goal is to maximize the uncertainty in the system (i.e., the size of the fusion interval) while staying undetected. We show that given these two objectives and a fixed number of sensors that can be corrupted, the attacker’s capabilities depend on the number of uncompromised intervals that she is aware of before sending her ‘measurements’ to the system. As is true in many CPS, we assume sensors communicate over a shared bus (e.g., CAN bus) and messages can be seen from everyone. Thus, if the attacker sends her intervals last, she can maximize the width of the fusion interval based on placements of the correct intervals.

Based on these observations, we investigate how different communication schedules affect the attacker’s capabilities to compromise the system. We give theoretic results that show
Finally, Section V concludes the work.

The effects of the communication schedules on sensor fusion.

Regarding the size of the fusion interval. Section IV studies

constraints of the attacker, and presents worst-case results

addressed in this paper. Section III formalizes the goals and

Marzullo's sensor fusion algorithm and specifies the problem

of this approach on an autonomous vehicle case-study.

In addition, we provide experimental results and a case study,

and argue that systems applying the fusion algorithm discussed

in this paper should use the Ascending schedule, which orders

sensors according to the size of their intervals starting with the

most precise. The contributions of this paper are as follows: (1)

attack formalization through the goals and constraints of the

attacker, (2) analysis of the impact of different communication

schedules on the attacker's capabilities, and (3) an illustration

of this approach on an autonomous vehicle case-study.

This paper is organized as follows: Section II outlines

Marzullo's sensor fusion algorithm and specifies the problem

addressed in this paper. Section III formalizes the goals and

constraints of the attacker, and presents worst-case results

regarding the size of the fusion interval. Section IV studies

the effects of the communication schedules on sensor fusion.

Finally, Section V concludes the work.

II. PROBLEM STATEMENT AND PRELIMINARIES

In this section, we describe the sensor fusion algorithm [5]

before formalizing the problem considered in this paper.

A. Marzullo's Algorithm

The inputs to Marzullo's sensor fusion algorithm are \( n \) real intervals, and a number \( f \) that denotes the number of

faulty intervals the system might have. The fusion interval

is then computed as follows: its lower bound is the smallest

point contained in at least \( n - f \) intervals and the upper

bound is the largest such point. Intuitively, the algorithm works

conservatively: since at least \( n - f \) intervals are correct, any

point that is contained in \( n - f \) intervals may be the true value,

and hence it is included in the fusion interval.

The algorithm is illustrated in Fig. 1. When \( f = 0 \) and the

system is confident that every interval is correct, the fusion

interval is just the intersection of all intervals. When at most

one sensor can be faulty (\( f = 1 \)), the fusion interval is the

convex hull of all points contained in at least four intervals.

Similarly, when \( f = 2 \) the fusion interval contains the convex

hull of all points that lie in at least three intervals. As shown

in Fig. 1, as \( f \) increases so does the uncertainty represented as

the size of the fusion interval. In particular, for \( f = n - 1 \) the

fusion interval is the convex hull of the union of all intervals.

Three important results of this work are worth noting. If \( f < \lfloor n/3 \rfloor \) then the width of the fusion interval is bounded above

by the width of some correct interval. Additionally, if \( f < \lfloor n/2 \rfloor \) the width of the fusion interval is bounded above by

the width of some interval (not necessarily correct). Finally, if

\( f \geq \lfloor n/2 \rfloor \) then the fusion interval can be arbitrarily large and

may not contain the true value. Thus, in our work we assume

that the \( f \) used in the algorithm always satisfies \( f < \lfloor n/2 \rfloor \),

causing the fusion interval to be bounded.

B. System Description and Problem Statement

We consider a system with \( n \) sensors measuring the state

of the same physical variable and communicating with the

controller over a shared bus. Each sensor provides the con-

troller with a measurement; an interval containing all possible

values of the true state is then computed based on that sensor's

precision, implementation guarantees and sampling jitter. In

particular, if the manufacturer’s precision guarantee is \( \delta \), then

an interval of size \( 2\delta \) centered at the sensor's measurement

is constructed; the interval size is further increased if the

worst-case guarantees for sampling jitter (and implementation

limitations) are considered (e.g., the design of intervals in [5]).

Thus, we assume the intervals’ widths are known and fixed.

A sensor is said to be correct if the interval contains the true

value, and faulty otherwise. In addition, each sensor transmits

its measurement to the controller according to a predefined

communication schedule (i.e., in its predefined slot). Once

the controller has received all \( n \) intervals, it will perform the

fusion algorithm with a predefined \( f \) (e.g., conservative upper

bound) such that \( f < \lfloor n/2 \rfloor \), followed by the attack detection

procedure which discards all intervals that do not intersect the

fusion interval (for more details see Section III-A).

1) Attack Model: In this work, we assume sensors are not

faulty but can be under attack. In addition, when the attacker

takes control of a sensor, she is still able to obtain correct

measurements from the sensor before deciding what value

(correct or manipulated) to send to the controller. Note that the

measurements of all compromised sensors provide additional

information to the attacker regarding the true value. Thus, we

use \( \Delta \) to denote the intersection of the correct measurements

from all corrupted sensors. Assuming that the attacker knows

the sensor fusion algorithm, her goal is to maximize the width

of the fusion interval while staying undetected.

2) Problem Statement: The problem considered in this

work is two-fold. We first analyze what the best policy for the

attacker is, given her objective. Then we note that her actions

and capabilities depend on what intervals she has seen before

sending hers. Therefore, our goal is to find a communication

schedule that will minimize the attacker’s impact. We examine

specific schedules used to transmit sensor measurements and

deﬁne metrics that can be used to compare them.

C. Notation

We use \( N \), where \(|N| = n\), to denote the set of all sensors

and \( L \) to denote the set of the lengths of all intervals. In

addition, \( S_{N,f} \) denotes the fusion interval given all sensors in

\( N \), for a predefined (fixed) \( f \), while in general, \( S_{P,f} \) denotes

the fusion interval for a predefined \( f \) using the intervals in the

set \( P \). Since we distinguish between the number of attacked

sensors and the input to Marzullo’s algorithm \( f \), let \( f_a \) denote

Fig. 1. Marzullo’s fusion interval for three values of \( f \). Dashed horizontal

line separates sensor intervals from fusion intervals in all figures in this work.
the number of attacked sensors, and we assume that \( f_n \leq f \). Finally, \( C \) denotes the set of all correct sensors, where \(|C| = c\) (i.e., \( c + f_n = n \)). Note that \( S_{C,0} \) is the intersection of all intervals in \( C \). For each interval \( s \), let \( u_s \) and \( l_s \) denote its upper and lower bound, respectively (\( s = [l_s, u_s] \)), and let \( |s| = u_s - l_s \). Similarly, \( u_{S_{C,f}} \) and \( l_{S_{C,f}} \) denote the upper and lower bound of the fusion interval \( S_{C,f} \).

### III. Attack Policy and Worst-Case Analysis

This section focuses on the attacker’s capabilities and relates them with the number of intervals she has seen before sending hers. We start by formalizing the attacker’s goals and constraints before providing worst-case results when \( f_a < [n/2] \).

#### A. Attack Policy

1) **Staying Stealthy:** Due to the utilized sensor fusion algorithm, the detection mechanism the system uses is to check for overlap with the fusion interval; if an interval does not intersect the fusion interval, then it must be compromised [5]. To avoid detection, the attacker has two modes – passive and active. The attacker starts in passive mode where she has to include \( \Delta \) in her interval and use the rest of her interval (if any) to maximize the size of the fusion interval. The entire \( \Delta \) has to be included to ensure overlap with the fusion interval (otherwise, any excluded point may be the true value).\(^1\)

The attacker can switch to active mode when the number of transmitted sensor measurements is at least \( n - f - f_a \), where \( f_a \) is the number of unsent compromised intervals. In this mode there are no constraints on the placement of her intervals as long as overlap with at least \( n - f - 1 \) intervals is guaranteed, which ensures overlap with the fusion interval (note that she may have to protect her earlier intervals). The reason this attack is undetectable is that the attacker’s intervals will always intersect the fusion interval, and thus the system would not be able to determine which sensors are malicious.

2) **Maximizing the size of the fusion interval:** If the attacker has full knowledge of all correct intervals then her policy can be formulated as the optimization problem with variables \( a_1, \ldots, a_{f_a} \), representing the attacked intervals:

\[
\max_{a_1, \ldots, a_{f_a}} |S_{N,f}|
\]

\[
\text{s.t. } S_{N,f} \cap a_i \neq \emptyset, \quad i = 1, \ldots, f_a.
\]

(1)

This problem formulation leads to the following definition.

**Definition 1:** Given placements of the correct intervals, an attack policy is **optimal** if the fusion interval has the same size as in the solution of problem (1).

Thus, the policy described in (1) is optimal by definition.\(^2\) We argue, however, that in general there exists no optimal policy for the attacker if she is not aware of all correct intervals before sending hers.\(^3\) This is illustrated in Fig. 2. Suppose the attacker has only seen interval \( s_1 \) and obtained interval \( a_1 \).

\(^1\)A generalization of this work will include a fault model over time for each sensor (e.g., a sensor is compromised only if it is faulty more than \( f \) out of \( w \) measurements). Thus, a sensor may have a temporary fault without being discarded as compromised.

\(^2\)It is worth noting here that the optimal policy need not be unique.

\(^3\)Note that this problem can be mapped to the problem of optimization with limited information.

from her sensor before sending her interval. One option for her is to send \( a_1(1) \); this would not be optimal if \( s_2 \) appears as illustrated.\(^4\) Alternatively, if the attacker tries to attack on both sides (interval \( a_1(2) \) from Fig. 2), then \( s_2 \) could still appear as shown, in which case \( a_1(2) \) would not be optimal.

In cases such as Fig. 2, a reasonable goal for the attacker is to maximize the expected performance over all possible measurements obtained by correct and compromised unseen sensors. Formally, an instance of the following problem is solved for each compromised interval \( a_k \):

\[
\max_{a_k, \ldots, a_{f_a}} \mathbb{E}[|S_{N,f}|]
\]

\[
\text{s.t. } S_{N,f} \cap a_i \neq \emptyset, \quad i = k, \ldots, f_a,
\]

(2)

where \( C_k^R \) is the set of all possible placements of correct intervals that will be transmitted after interval \( a_k \), and \( \mathbb{E} \) denotes the expectation operator.

There do, however, exist cases in which there is an optimal policy for the attacker even if she is not aware of all correct intervals. In particular, there exist placements of the seen correct intervals that provide the attacker with enough information to place her intervals in an optimal way. To formalize the above statement, let \( C^S \) be the set of seen correct intervals and let \( C^R \) be defined as above. Let \( l_{n-f-f_a} \) be the \((n-f-f_a)\)th smallest seen lower bound and let \( u_{n-f-f_a} \) be the \((n-f-f_a)\)th largest seen upper bound. Finally, let \( m_{min} \) be the attacked sensor with smallest width.

**Theorem 1:** Suppose \( n - f - f_a \leq |C^S| < n - f_a \), and the attacker is scheduled to transmit in consecutive slots. There exists an optimal attack policy if one of the following is true:

1) \( \forall s_i, s_j \in C^S, l_{s_i} = l_{s_j}, u_{s_i} = u_{s_j} \) and \( \forall s \in C^R, |s| \leq (|m_{min}| - |S_{C^S \cup \Delta,0}|)/2 \)

2) \( |m_{min}| \geq u_{n-f-f_a} - l_{n-f-f_a} \) and \( \forall s \in C^R, |s| \leq \min\{l_{S_{C^S \cup \Delta,0}} - l_{n-f-f_a}, u_{n-f-f_a} - u_{S_{C^S \cup \Delta,0}}\} \)

**Proof:** First suppose the first statement is true. We argue that the optimal policy for the attacker is to attack on both sides of seen intervals. For each \( s \in C^R \), \( s \) must overlap with at least one point in \( S_{C^S \cup \Delta,0} \) (the overlap must contain the true value) and since \( |s| \leq (|m_{min}| - |S_{C^S \cup \Delta,0}|)/2 \) then \( s \) will necessarily overlap with all malicious sensors implementing the above policy. Note that since \( f < [n/2] \), the fusion interval cannot be larger than the union of all correct intervals. Therefore, this policy is optimal because the attacker can guarantee that all her intervals contain all correct intervals. Fig. 3(a) illustrates this case. All seen correct intervals coincide, and the attacker’s intervals are large enough to guarantee that attacking on both sides will make sure all unseen intervals are included.

\(^4\)The symmetric counterexample exists if the attacker tries to attack on both the right (this case is not shown in Fig. 2).
is bounded by the sum of the two largest correct intervals. Thus, the attacker to contain both correct while preserving the size of the fusion interval, hence the same worst case can be achieved with correct intervals.

Theorem 4: $|S_{\text{w}c}|$ is achievable if the $f_a$ smallest intervals are under attack.

Proof: Note that if $|S_{\text{w}c}| = |N_{\text{small}}|$, the theorem follows trivially. Consider the case $|S_{\text{w}c}| > |N_{\text{small}}|$. Suppose $|S_{\text{w}c}|$ is not achievable if the $f_a$ smallest intervals are attacked. Let $S$ be the configuration with $f_a$ corrupted intervals that achieves $|S_{\text{w}c}|$ and let $A$ be the rightmost point in $S_{\text{w}c}$. Since $|S_{\text{w}c}| > |N_{\text{small}}|$, there exist an interval $s \in S$ that does not contain the true value but contains $A$. Let $N'_{\text{small}}$ be the set of $f_a$ smallest intervals. If $s \notin N'_{\text{small}}$ for all such $s$ then $S_{\text{w}c}$ is achievable if $N'_{\text{small}}$ is under attack and the theorem follows.

Now suppose there exists an $s$ as above such that $s \notin N'_{\text{small}}$. Then there exists an interval $s_{\text{small}} \in N'_{\text{small}}$ that is not under attack. If we swap $s$ and $s_{\text{small}}$ such that $s_{\text{small}}$ now contains $A$ and $s$ contains the old interval $s_{\text{small}}$, $s$ is made correct and $s_{\text{small}}$ corrupted while preserving the size of the fusion interval. Since we can do the same for all such $s$, $|S_{\text{w}c}|$ can be achieved if $N'_{\text{small}}$ is under attack.

Fig. 4(b) gives an example the theorem. The worst-case for the setup can be achieved when $s$ or $s_{\text{small}}$ is attacked.

Theorems 3 and 4 suggest that the attacker can gain more power by corrupting the precise sensors as opposed to the imprecise. The intuitive explanation is that the uncertainty in the system increases – attacking imprecise sensors does not greatly affect the fusion interval since their intervals are large even when correct; attacking precise sensors, however, leaves the system with less information as to where the true value is.
IV. SENSOR SCHEDULING FOR ATTACK-RESILIENT
SENSOR FUSION

As shown in the previous section, the attacker’s capabilities vary greatly with the intervals she has compromised and correct intervals she has seen. In this section we investigate the impact of communication schedules on the attacker’s performance. Note that interval lengths are the only information available a priori to the system, hence any schedule will only be based on those. In this work we consider the two obvious choices of ordering sensors according to their precision. In the first, called the Descending schedule, the largest intervals (i.e., the most inaccurate sensors) are sent first. In contrast, the Ascending schedule makes the most accurate sensors send first. Other schedules are considered at the end of this section.

We first note that neither schedule is optimal in all scenarios. Theorem 1 suggests that it might be better for the attacker to see large intervals first. Fig. 5(a) illustrates an example where this is true. Fig. 5(b), however, shows that knowing the largest interval does not necessarily bring the attacker any useful information because she can only increase the fusion interval by overlapping with $s_1$ and $s_2$. Hence, if she is aware of $s_3$ when sending her interval she would send $s_D$ but that would be worse than sending $s_A$ which would be the case if the attacker had seen $s_1$ and $s_2$ instead.

Since neither schedule is absolutely better than the other, we consider the average case. In particular, we examine the expected length of the fusion interval for each schedule given fixed sensor precisions. The next subsection describes our simulation results in testing the usefulness of each schedule.

A. Schedule Comparison

We compare the two schedules by varying the number of sensors, their accuracies and the number of attacked sensors. In particular, the number of sensors vary from three to five; the lengths of the intervals are increased from 5 to 20 by increments of 3 for each interval. Finally, the number of attacked sensors is increased from one to $\lceil n/2 \rceil - 1$. For each setup, we generate all possible combinations of sensors and take the average length of the fusion interval, i.e., our best estimate of the true expected length of the fusion interval. Our analysis, the system always uses the sensor fusion algorithm configured for $f = \lceil n/2 \rceil - 1$.

Table I shows the simulation results. Due to the large number of combinations tried, we chose several setups that represent classes of combinations, two per $n$ and $f_a$ combination. It was noticed during the simulations that the expected lengths of the two schedules are similar when interval sizes were comparable, while they tend to get further apart when there are large differences in sizes. Regardless of the gap between the two lengths, the expected length under the Descending schedule was never smaller than that under Ascending.

B. Case Study

In addition to the simulations shown in the previous section, we also illustrate our results with a case-study. For this purpose, we use four sensors on the LandShark robot [9] (Fig. 6). The LandShark is commonly used in hostile environments to save injured people or carry other cargo. It has four sensors that estimate its speed – GPS, two encoders and a camera. The size of the GPS and camera intervals were determined empirically – the LandShark was driven in the open and largest deviations from actual speed were recorded for both the GPS and camera speed estimates. The encoders’ intervals were determined based on their manufacturer specification.

We simulated the scenario with three LandSharks in a platoon moving away from enemy territory. The leader sets a speed target $v$ mph for all three vehicles based on the environment. Each LandShark has a low-level controller that tries to keep the speed at $v$ mph. There are two additional restrictions on each vehicle – speed cannot exceed $v + \delta_1$ mph as that may make the LandShark go too fast and unable to stop fast enough (either of the last two LandSharks may collide with the one in front or the leader may crash into an obstacle). Furthermore, speed is not to drop below $v - \delta_2$ mph as that may cause a LandShark to collide with the one behind. These constraints are encoded into the fusion interval – if its upper bound exceeds $v + \delta_1$ mph or the lower bound is less than $v - \delta_2$ mph then a high-level algorithm will preempt the low-level controller to guarantee safety of the vehicles.

We assume that at most one sensor can be attacked at any given point of time. In addition, while it is true that some sensors are easier to spoof than others, we assume that any sensor can be attacked in this scenario; if it is known which sensor is being attacked then any schedule that places that sensor first would result in a smaller fusion interval.

Simulations were run with a desired speed of $v = 10$ mph, and $\delta_1 = \delta_2 = 0.5$ mph. As noted above, the GPS and camera interval sizes were determined empirically and are 1 mph and 2 mph, respectively. Each encoder’s interval size was computed based on an encoder with 192 cycles per revolution, a measuring error of 0.5% and sampling jitter error of 0.05%. The final interval length was computed to be 0.2 mph.

Table I shows the results. In addition to the Ascending and Descending schedules, we include a Random schedule that changes transmission order in every step. For each schedule we

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPARISON OF TWO SENSOR COMMUNICATION SCHEDULES.</td>
</tr>
<tr>
<td>$n = 3, f_a = 1, \mathcal{L} = {5, 11, 17}$</td>
</tr>
<tr>
<td>$n = 3, f_a = 1, \mathcal{L} = {5, 11, 11}$</td>
</tr>
<tr>
<td>$n = 4, f_a = 1, \mathcal{L} = {5, 8, 17, 20}$</td>
</tr>
<tr>
<td>$n = 5, f_a = 1, \mathcal{L} = {5, 5, 5, 5, 20}$</td>
</tr>
<tr>
<td>$n = 6, f_a = 2, \mathcal{L} = {5, 5, 5, 5, 5, 20}$</td>
</tr>
</tbody>
</table>

Note that we have discretized the real line with a sufficiently high precision in order to compute the expectation in the optimization problem.
compute the percentage of runs in which the fusion interval’s upper bound was above 10.5 mph and the percentage of runs in which its lower bound was less than 9.5 mph.

C. Discussion

The cases discussed in Theorem 1 are only sufficient and not necessary conditions for the existence of an optimal attack policy. Yet, simulations suggest that their intuition generalizes well – it is in general better for the attacker to know where the most imprecise intervals are so that she can attack on one side of them while the correct intervals would be on the other side, thus confusing the controller. This argument is also supported by Theorems 3 and 4, which suggest that the attacker will gain more power by compromising the most precise intervals.

Based on these results and observations, we argue that the controller will benefit from the Ascending schedule. This is especially true when there is a significant difference in size between small and large intervals, in which case the attacker can take advantage of seeing the large intervals first. This schedule makes sure that even if the attacker compromises the most precise sensors she will be forced to send first and will have to guess the position of the correct intervals.

At the same time, we note that often the fusion interval depends more on the position of the intervals than on their lengths (Fig. 5(b) is a good example of this). Hence any schedule that the controller picks will perform with various accuracy depending on the positions of the intervals.

One could also argue that imposing a scheduling order would make it easier for the attacker to know which sensors would be most useful to attack. This argument, however, could be made about any schedule that the attacker may know in advance. A way to prevent such a situation is to change the schedule constantly. This approach is addressed in Table II, which shows that constantly changing the schedule and using a Random schedule results in a larger number of critical speed violations than always using the Ascending schedule.

One last consideration when designing a communication schedule is the likelihood that certain sensors can be attacked. In particular, an IMU is in general much harder to spoof than a GPS or a camera. In cases like these, where the system is confident that some sensors are correct, our analysis shows that they should always be placed last in the schedule, thus preventing the attacker from knowing their measurements.

V. CONCLUSION

This work described a general sensor fusion algorithm for multiple sensors measuring the same physical value. We introduced security issues by formalizing an attack policy that tries to maximize the size of the fusion interval while staying undetected. Furthermore, we proposed a communication schedule, namely the Ascending schedule, that aims to minimize the attacker’s capabilities by either providing her with little information (sending at the beginning of the schedule) or little power (large intervals). We presented worst- and average-case results that support our choice of schedule and validated our results in simulation and a case study. Since we assumed uncompromised sensors always provide correct measurements, an extension of this work will introduce random faults in addition to attacks.

REFERENCES