2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Bayesian Methods for Trust in Collaborative Multi-Agent Autonomy

R. Spencer Hallyburton and Miroslav Pajic

Abstract— **Multi-agent, collaborative sensor fusion is a vital component of a multi-national intelligence toolkit. In safety-critical and/or contested environments, adversaries may infiltrate and compromise a number of agents. We analyze state of the art multi-target tracking algorithms under this compromised agent threat model. We show that the track existence probability test ("track score") is significantly vulnerable to even small numbers of adversaries. To add security awareness, we design a trust estimation framework using hierarchical Bayesian updating. Our framework builds beliefs of trust on tracks and agents by mapping sensor measurements to trust pseudomeasurements (PSMs) and incorporating prior trust beliefs in a Bayesian context. In case studies, our trust estimation algorithm accurately estimates the trustworthiness of tracks/agents, subject to observability limitations.**

I. Introduction

Networks of low-cost autonomous sensing agents are proliferating in the surveillance and intelligence-gathering space. The use of multiple sensors to track dynamic targets has many benefits including the increased field of view by aggregation and resilience to occlusions, false positives (FPs), and false negatives (FNs). However, in contested environments, adversaries may infiltrate and compromise one or more agents. Yet, few security analyses have been performed on networks of autonomous agents, even though it is critical to analyze classical algorithms for multi-agent collaboration with security in mind; especially when the collaboration involves untrusted and potentially *distrusted* agents.

We consider a multi-agent *surveillance* problem in which a collection of agents jointly track dynamic objects. Optimal data fusion requires fusion of raw detections from each of the platforms in a centralized manner [3]. We consider the classical approach of multi-sensor, multi-target tracking (MTT) using centralized data fusion with a global nearest neighbors data association and Kalman filter estimator [3] (Fig. 1). We evaluate the MTT approach under a *compromised agent* threat model. We assume at least one *adversarially compromised* agent provides time-correlated FPs and/or FNs.

We test *the only built-in method of integrity in MTT*: the "track score", which is meant to filter FPs, by assigning low scores to nascent tracks, and to be robust to intermittent FNs; however, it was not designed with security in mind. We show that *even when benign agents outnumber adversaries*, attackers need only a small number of frames to establish high-confidence FP tracks mistakenly believed to be real objects.

Several works have proposed algorithms for "secure state estimation". Track score shortcomings were first noted in [7] and a minor modification to the score function was proposed. [11], [12] designed secure state estimation algorithms for Byzantine attacks on sensors; [5] considered "consensus" in the presence of malicious nodes in distributed estimation; [9] considered decentralized tracking a dynamical system.

Our approach is orthogonal (and can be performed in parallel) to secure state estimation and directly estimates whether tracks and agents are *trustworthy* via *trust estimation* (see Fig. 1). Related to trust estimation, [16] explored statistical models of trust assuming binary inputs; [4] considered vehicular ad hoc networks (VANETs) using a distributed, single-frame trust model to compute agent-based metrics; [2] applied a particle filter to track trust and confidence as an "opinion" in VANETs; [15] used graph theory to extend Dijkstra's shortest path algorithm to trust in ad hoc networks.

There are several shortcomings with existing approaches to trust estimation. First, several use either binary inputs or single-frame representations of trust ([2], [15], [16]). This does not allow for dynamically changing, real-valued, and uncertain outcomes. Further, few can incorporate prior information into the trust model ([4], [15]), which is essential with small numbers of agents and imperfect measurements.

The lack of security awareness in MTT and the inability of existing trust models to capture prior information and uncertainties motivates our approach to trust estimation. We formulate the trust estimation problem in a collaborative, multi-agent scenario within the context of Bayesian parameter estimation. In this context, a-priori information is incorporated if available via informative priors on agent and track trust parameters. At each timestep, sensor measurements are used by MTT to establish tracks and estimate their states. Our trust model also uses sensor measurements to update the belief on the trustworthiness of those tracks *and* the agents.

To estimate trust from sensor measurements, we design novel functions that map uncertain sensor data to real-valued "*pseudomeasurements*" (PSMs) of trust. Trust PSMs are reals on $[0, 1]$ that are ad-hoc estimates of the trust from a single frame of sensor data. We use PSMs to update the track and agent trust in an alternating procedure inspired by conditional Gibb's sampling (e.g., [13]). With parametric trust priors and a simple PSM likelihood function, the trust estimation framework performs Bayesian updating of the trust distribution parameters with closed-form, analytic equations.

We illustrate the effectiveness of our approach to trust estimation in two distinct case studies *both* with and without prior information. We consider the compromised agent threat model and task trust estimation with ascertaining distribu-

This work is sponsored in part by the ONR N00014-23-1-2206 and AFOSR FA9550-19-1-0169 awards, as well as the NSF NAIAD Award 2332744 and the NSF CNS-2112562 Award.

R. S. Hallyburton and M. Pajic are with Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA; {spencer.hallyburton, miroslav.pajic}@duke.edu.

979-8-3503-1632-2/24/$31.00 ©2024 IEEE

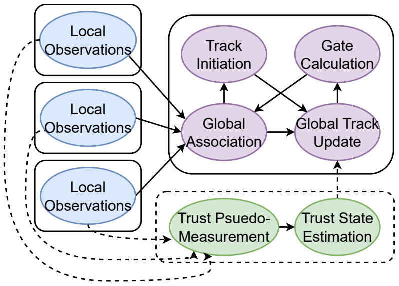


Fig. 1: Trust estimation (green) is complementary to existing sensor fusion architectures (purple) for performing inference on data from multiple platforms (blue).

tions over the trust of all tracks and agents. Trust estimation successfully verifies tracks on true objects as trusted and tracks on false objects as untrusted under favorable observability conditions. Moreover, we importantly find that prior information is highly useful in accelerating the determination of trust/distrust in multi-agent collaboration.

This paper is organized as follows: Sec. II describes the foundations of multiple target tracking, Sec. III shows the vulnerability of classical MTT algorithms, Sec. IV formalizes our approach to trusted sensor fusion and V presents experimental results on trust estimation case studies.

II. Multiple Target Tracking (MTT)

We consider multi-target surveillance where K agents are connected to a centralized data fusion engine and share detection-related data. The goal of multi-target surveillance is to determine the number of objects that exist in a dynamic scene and to estimate the states of those objects over time; such tasks are known collectively as MTT.

FPs and missed detections of true objects, i.e., FNs, create challenges in determining object existence. Fig. 2 illustrates a common case of such ambiguity: one agent believes to see an object not seen by the other agents. Naturally, questions arise as to whether this object is real or an FP. In the following, we present the de facto standard approach to MTT (following e.g., [3]): a two-step algorithm for solving existence and estimation tasks. We then formally present "track scoring" as the classical approach to the existence determination step.

A. MTT as a Two-Step Problem

We assume that agents $k = 1, ..., K$ provide $Q_{k,t} \geq 0$ detections at time t. We use $Z_t := \{z_{q,k,t}\}|_{q\in[1,\dots,Q_{k,t}],k=1,\dots,K}$ as the *set of detections* from all agents and $X_t := \{x_{i,t}\}$ as the set of all N_t true object states. Formally, the objective of MTT is to estimate the joint posterior:

$$\Pr(X_t|Z_{1:t}) = \frac{\Pr(Z_t|X_t)\Pr(X_t|Z_{1:t-1})}{\Pr(Z_t|Z_{1:t-1})}, \tag{1}$$

where Pr is a probability distribution. At each step, MTT retains a set of tracks, $\hat{X}_t := \{\hat{x}_{j,t}\}$ as estimates of object states. Subscripts j do not necessarily align with i since $\hat{X}_t$ estimates both existence and state, e.g., $\hat{X}_t$ can have natural FPs or FNs and both $\hat{X}_t$, X_t are permutation-invariant.

MTT usually takes a two-stage approach to reduce the multi-object posterior to multiple single-object problems. Instead of using all measurements to update all tracks, MTT often assigns measurements to specific tracks for single-track updating (see many examples in [1], [3]). Steps include:

1) **Data association:** perform bipartite matching to assign current detections, Z_t, to estimated track states, $\hat{X}_{t-1}$. Often, a measurement can only be used for a single track. Detections without a track start new tracks, tracks without detections are considered "missed".
2) **Existence & state estimation:** for each track, use assigned measurements from data association to update the track existence probability and state estimate.

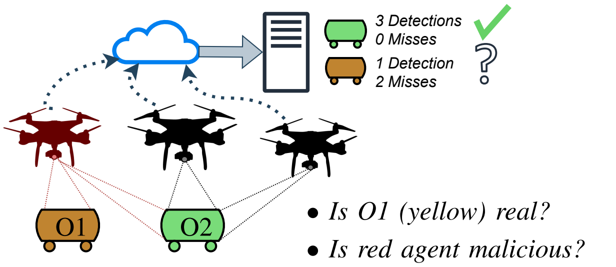


Fig. 2: A malicious agent (left, red) provides an FP (O1) that is not detected by the benign agents (black). Under what conditions can MTT identify that O1 is an FP? That the red agent is malicious? "Track scoring" is a natural tool for existence determination, yet, we show it is vulnerable to many adversarial cases. Instead, we propose to augment MTT by estimating the "trust" of tracks and agents.

The measurements help the existence task reason about whether the track represents a real object or is an FP. The state estimation task employs an estimator such as the Kalman filter to mix measurements and kinematic models. Important to MTT is both agent pose (i.e., position and orientation) and the field of view (FOV) model, $\Phi_k(\cdot)$, that takes as input a point in space and determines if agent k could reasonably observe an object at that point, if there existed one. The FOV model is important e.g., so as not to penalize agents and tracks for "misses' when the candidate track was not in the field of view of the agent to begin with. We group both under the term "agent characteristics", $A_t := \{a_{k,t}\}$, and assume A_t is known and uncompromised.

B. Track Existence Determination via Likelihood Scoring

A classic approach to determining whether a track represents a real object or is an FP is to "confirm" tracks when they have received a significant number of quality measurements [1], [3]. Confirmation is quantified in a process known as *track scoring* [14]. It uses hypothesis testing *for each track* as either real ($\mathcal{H}_1$) or fake ($\mathcal{H}_0$). We adopt the notation of [3] that represents the likelihood ratio between the hypotheses as

$$LR(\hat{x}_{j,t}) = \frac{\Pr(Z_t|\mathcal{H}_1)\Pr(\mathcal{H}_1)}{\Pr(Z_t|\mathcal{H}_0)\Pr(\mathcal{H}_0)} := \frac{P_T}{P_F}, \tag{2}$$

where Z_t is the measurement data and $\Pr(\mathcal{H}_i)$ is the prior probability of the hypotheses. The joint distribution of the data and $\mathcal{H}_i$ have probabilities P_T and P_F, respectively.

The likelihood ratio evaluated under the natural logarithm is known as the "track score". There is a direct transformation between score and the real-object ($\mathcal{H}_1$) probability

$$LLR := L = \log\frac{P_T}{P_F}, \quad P_T = \frac{e^L}{1+e^L}. \tag{3a,b}$$

The initial track score is set to

$$L_0 = \log\left[\frac{P_D \beta_{NT}}{\beta_{FP}}\right] \tag{4}$$

where β_{NT}, β_{FP} are the expected densities of new targets and FPs, respectively. As derived in [14], temporal updates to the track score can be made with the recursion

$$L_t = L_{t-1} + \Delta L_t \tag{5a}$$

$$\Delta L_t = \begin{cases} \Delta L_{m,t} & \text{if no assignment (miss)} \\ \Delta L_{h,t} & \text{if assignment (hit)} \end{cases} \tag{5b}$$

$$\Delta L_{m,t} = \log 1 - P_D \tag{5c}$$

$$\Delta L_{h,t} = \log\left[\frac{P_D}{(2\pi)^{\eta/2}\beta_{FP}\sqrt{|S|}}\right] - \frac{d^2}{2} \tag{5d}$$

where P_D is the probability of detecting a true object, η is the number of dimensions, $|S|$ is the determinant of the innovation covariance from the Kalman filter, and $d^2 = \tilde{y}^T S^{-1} \tilde{y}$ where $\tilde{y}$ is the innovation in the Kalman filter. A higher P_D yields a greater penalty for a "miss". A "hit" updates the score as a function of how closely the measurement matches the track's last estimated state.

Track scoring is fundamental to the existence task and the only statistical determination of whether a track represents a real object. Formally, following [3], track status is:

$$\text{status} = \begin{cases} \text{track confirmed;} & L \geq \mathcal{T}_2 \\ \text{continue test;} & \mathcal{T}_1 < L < \mathcal{T}_2 \\ \text{delete track;} & L \leq \mathcal{T}_1. \end{cases} \tag{6}$$

$$\mathcal{T}_2 = \log\left[\frac{1-\beta}{\alpha}\right], \quad \mathcal{T}_1 = \log\left[\frac{\beta}{1-\alpha}\right] \tag{7}$$

with (α, β) application-specific (see [1], [3]).

In what follows, we perform a security analysis of track scoring. We show that even under a threat model when benign agents outnumber adversaries, track scoring is vulnerable. This motivates the development of a novel technique in Sec. IV that quantifies the *trust* of tracks and agents.

III. Security Analysis of Track Scoring

The track score represents the belief that a track corresponds to a real object vs. an FP. To motivate a formal analysis, Fig. 3 dives into track scoring for the surveillance problem from Fig. 2. O2 (green) is detected by all three agents and will receive a gain contribution from each. On the other hand, O1 (yellow) receives a gain contribution from one agent and two losses from the misses, assuming O1 is within the FOVs of all agents. Despite having more misses than detections, it is not obvious whether O1 will be confirmed; the result depends on the size of the gain/loss contributions.

Despite its statistical motivation, in what follows we find that track scoring is vulnerable to adversarial manipulation. We consider a simple threat model and illustrate that even small numbers of adversaries relative to the number of benign agents can quickly lead to incorrect confirmation of an FP. The vulnerability arises because adversaries can create gains that outmatch the losses from benign agents.

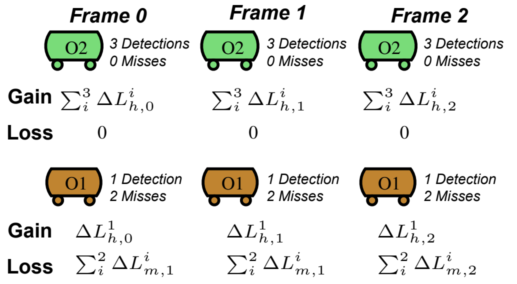

Frame 0
Frame 1
Frame 2
O2
3 Detections
0 Misses
Gain
Loss
$\sum_i^3 \Delta L_{h,0}^i$
$\sum_i^3 \Delta L_{h,1}^i$
$\sum_i^3 \Delta L_{h,2}^i$
0
0
0
O1
1 Detection
2 Misses
$\Delta L_{h,0}^1$
$\Delta L_{h,1}^1$
$\Delta L_{h,2}^1$
$\sum_i^2 \Delta L_{m,1}^i$
$\sum_i^2 \Delta L_{m,1}^i$
$\sum_i^2 \Delta L_{m,2}^i$

Fig. 3: Track scoring represents the probability that an object exists. Increments are calculated using gains from detections and losses from misses. With many detections, the true-object hypothesis score for O2 will increase. With a mix of detections and misses, the outcome for O1 is not obvious. Theorem 1 shows the conditions under which O2 is confirmed despite few detections and consistent misses.

A. Threat Model

We consider K_a adversaries and K_b benign agents. The adversaries can provide fictitious detections of their choosing (FPs) and/or can omit detections of objects within their FOVs (FNs). Adversaries have no way to manipulate the data from benign agents. We assume constant data rates – i.e., adversaries cannot send data faster or slower than benign agents.

B. Analysis of Track Score Updates

To perform analysis of MTT in potentially adversarial scenarios, we first bound the change in track score between frames. We then consider the threat model and a scenario in which an adversary wishes to "confirm" an FP. We approximate that all detections of the FP are from the adversary and all misses are from benign agents to obtain a more mathematically convenient (yet suboptimal) form.

1) Bounding Track Score Gain

The track score gain depends on the characteristics (noise, deviation from model) of the measurement. Since an adversary can completely control the measurement of an FP, he can achieve any gain up to a maximum fixed by the sensor characteristics and statistical models; these are set a-priori. Prop. 1 places a bound on the maximum possible gain to the track score for a single frame using these a-priori quantities.

Proposition 1: The contribution of any detection to the track score is bounded (from above) by:

$$\Delta L_h \leq \log\left[\frac{P_D}{(2\pi)^{\eta/2}\beta_{FP}\sqrt{|R|}}.\right]$$

Proof: It holds that $S := HPH^T + R$ for Kalman filtering where H is the linearization matrix, P the state covariance, and R the measurement covariance; $\{HPH^T, R\} \geq 0$ by construction, and

$$|R| \leq |HPH^T + R| = |S|$$

since $\det(A + B) \geq \det(A) + \det(B)$ for positive semi-definite matrices; see e.g., [10] for proof. The contribution of a detection hit to the log likelihood is given in (5a) and without the $-d^2/2$, the contribution is bounded by

$$\Delta L_h \leq \log\left[\frac{P_D}{(2\pi)^{\eta/2}\beta_{FP}\sqrt{|S|}}\right] \leq \log\left[\frac{P_D}{(2\pi)^{\eta/2}\beta_{FP}\sqrt{|R|}}\right]$$

since $|R| \leq |S|$ and log is a strictly increasing function. ■

2) *Bounding Change in Track Score*

The following bounds the total change in track score.

Proposition 2: For D_t detections and M_t misses, the change in track score in a single frame is bounded by

$$\Delta L_t \leq D_t \log \left[\frac{P_D}{(2\pi)^{\eta/2} \beta_{FP} \sqrt{|R|}} \right] + M_t \log \left[1 - P_D \right].$$

Proof: The aggregation of misses and hits results in

$$\Delta L_t = M_t \Delta L_{m,t} + D_t \Delta L_{h,t}$$
$$\Delta L_{m,t} = \log \left[1 - P_D \right]$$
$$\Delta L_{h,t} \leq \log \left[\frac{P_D}{(2\pi)^{\eta/2} \beta_{FP} \sqrt{|R|}} \right]$$

using Prop. 1, which concludes the proof. ■

3) *Natural False Positive Gate Probability is Small*

All sensors exhibit noise, so any volume in the environment can contain FPs. Each volume is statistically independent and the FP density is modeled as a constant, β_{FP}. The number of FPs in a bounded volume V_C is widely modeled as a homogeneous Poisson point process [1], [3]

$$f(N_{FP} = n; \Lambda) = \frac{\Lambda^n e^{-\Lambda}}{n!}, \tag{8}$$

with N_{FP} a number of FPs and $\Lambda = V_C \beta_{FP}$.

Measurements are assigned to tracks in Step 1 of MTT if they satisfy the *gating* criteria. Simply put, a measurement is allowed to update a track if they are statistically "close to" each other (i.e., if the measurement is within the "gating volume" of the track). It is possible for a natural FP to be close to an established track and satisfy the gating criteria; we consider the probability of this occurrence in Prop. 3.

Proposition 3: In an environment with constant FP density β_{FP}, at least one natural FP will be within the gating volume V_G of an existing track with probability $1 - e^{-V_G \beta_{FP}}$.

Proof: Suppose a confirmed track exists and on a round of measurements the volume of its gating region is V_G. Then,

$$\Pr(N_{FP} \geq 1 \in V_G) = 1 - \Pr(N_{FP} = 0 \in V_G) =$$
$$= 1 - F(N_{FP} = 0; \Lambda_G) = 1 - e^{-\Lambda_G} = 1 - e^{-V_G \beta_{FP}},$$

concluding the proof. ■

4) *Track Score Under Threat Model*

Finally, we consider that the adversary wishes for MTT to believe an FP is a real object. Specifically, we assume the adversary wishes to provide false detections to achieve FP confirmation via the track score as quickly as possible in the presence of benign agents that are providing negative results (i.e., no detections). Theorem 1 asserts the minimum number of frames, T_{min}, after which the score of an FP is above the confirmation threshold – i.e., an FP is confirmed.

Theorem 1: Given K_a adversaries and K_b benign agents observing a volume element, without aligned benign FPs, an adversary can establish a valid track in T_{min} steps, where

$$T_{min} = 1 + \frac{\mathcal{T}_2 - \log \left[\frac{P_D \beta_{NT}}{\beta_{FP}} \right]}{\left[K_a \log \left[\frac{P_D}{(2\pi)^{\eta/2} \beta_{FP} \sqrt{|R|}} \right] + K_b \log \left[1 - P_D \right] \right]}$$

frames, with $\mathcal{T}_2$ set according to (7).

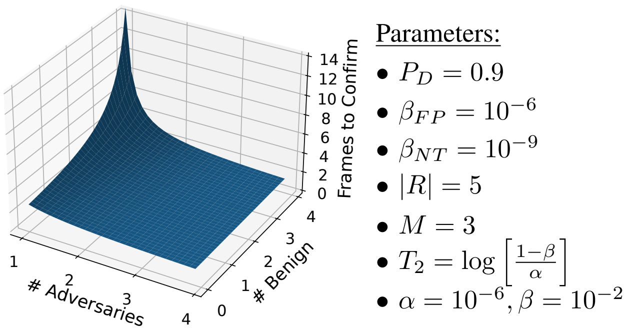


Fig. 4: Following Theorem 1, even with only few agents viewing an object, adversaries can quickly confirm fake tracks. More benign agents viewing a track forces adversaries to use more time to establish a confirmed track. E.g., evaluating $f(K_a = 1, K_b = 1) \approx 3$, $f(K_a = 1, K_b = 3) \approx 6$.

Proof: A track is confirmed if $L_t \geq \mathcal{T}_2$. Also, $L_t = L_{t-1} + \Delta L_t$ so $L_T = L_0 + \sum_{t=1}^{T} \Delta L_t$. Prop. 2 bounds the change in track score for any N_t detections and M_t misses. Since the track is an adversarial FP, N_t will be a combination of one adversarial FP from the malicious agent and some number of natural FPs from benign agents that coincidentally overlap. Similarly, M_t will be one miss from each benign agent not providing a natural FP, i.e., $K_a \leq N_t \leq K_a + K_b$ and $0 \leq M_t \leq K_b$. However, from Prop 3, the probability of benign agents having natural FPs near the adversary's FP will be small. Thus, we assume no aligned benign FPs such that $N_t = K_a$ and $M_t = K_b$. Since there are no frame-dependent terms in ΔL_t, after transforming the sum to get the threshold point, using (4) for initial score, we obtain

$$(T_{min} - 1) \Delta L_t = \mathcal{T}_2 - \log \left[\frac{P_D \beta_{NT}}{\beta_{FP}} \right]$$

$$T_{min} = 1 + \frac{\mathcal{T}_2 - \log \left[\frac{P_D \beta_{NT}}{\beta_{FP}} \right]}{\left[K_a \log \left[\frac{P_D}{(2\pi)^{\eta/2} \beta_{FP} \sqrt{|R|}} \right] + K_b \log \left[1 - P_D \right] \right]}.$$

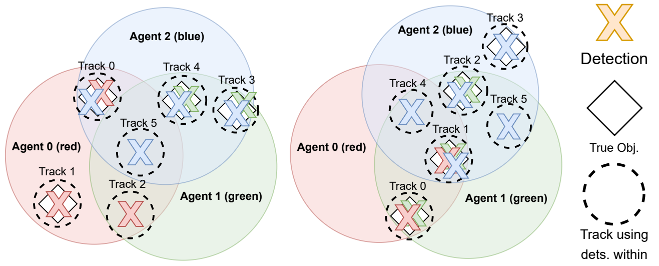
■

Theorem 1 establishes a minimum number of frames before MTT erroneously confirms an FP as valid if the adversary optimally places detections. Fig. 4 shows a surface plot of the number of frames as a function of (K_a, K_b) while fixing parameters to nominal values — *even when the number of benign agents outnumbers the adversary, the adversary easily achieves a confirmed track in single-digit number of frames;* e.g., for $K_a = 1, K_b = 3$, adversaries only require 6 frames to confirm an FP.

IV. MTT Estimation of Track and Agent Trust

To overcome the demonstrated MTT vulnerability, we consider that agents' detections can inform whether the agents and the tracks they help establish are *trusted*. Trust can then be used to inform MTT to ignore distrusted tracks.

We estimate track and agent trust within MTT. Trust is quantified as a belief over $[0, 1]$ of a track existing (track trust) or of agents providing measurements consistent with the true state (agent trust). To support the derivation of trust algorithms and for case study in Sec. V, we present two cases



(a) Case 1 (b) Case 2

Fig. 5: Three agents with partially overlapping FOVs. (Case 1) Agents 0 and 2 providing FP detections to establish malicious tracks–Tracks 2, 5. (Case 2) Agent 2 providing two FP detections for two malicious tracks, Tracks 4, 5. If a track is in only a single agent's FOV (e.g., Case 1, Track 1), not enough info is available to estimate track trust. With multiple overlapping observations (Case 2, Track 1), FOV models help form trust PSMs on tracks and agents.

of multi-agent MTT in Fig. 5. In both cases, three agents are providing detections from partially overlapping FOVs. Detections are fed to the central MTT that establishes global tracks. In Case 1, Agents 0 and 2 are both trying to establish malicious FPs (Tracks 2 and 5). In Case 2, agent 2 is trying to establish two malicious FPs (Tracks 4, 5). We introduce and evaluate algorithms that leverage these discrepancies to estimate the trustworthiness of each track and each agent.

A. MTT With Trust Estimation Posterior

Formally, the objective of MTT with trust estimation is to estimate the full joint posterior:

$$\begin{aligned}&\Pr(X_t, \mathrm{T}^c_t, \mathrm{T}^a_t | Z_{1:t}, A_{1:t}) = \\ &\quad = \Pr(\mathrm{T}^c_t, \mathrm{T}^a_t | Z_{1:t}, A_{1:t}) \Pr(X_t | \mathrm{T}^c_t, \mathrm{T}^a_t, Z_{1:t}, A_{1:t})\end{aligned} \tag{9}$$

where X_t are object states for all N objects, T^c_t are track trusts for each $j = 1...C$ tracks, T^a_t are agent trusts for each $k = 1...K$ agents, $Z_{1:t}$ are measurements from all agents, and $A_{1:t}$ are agent characteristics including the pose and FOV model, $\Phi_k(\cdot)$. In (9), we use conditional probability to decompose into subproblems: (9.1) trust estimation posterior, (9.2) state estimation posterior conditioned on trust. This decomposition allows us to run trust estimation as its own node independent of MTT, as illustrated in Fig. 1. The rest of this works focuses on (9.1); we leave a full treatment of (9.2), target tracking augmented with trust, to future works.

To estimate the trust posterior, we use a decomposition inspired by the popular Gibbs sampling (see e.g., [13]). This breaks the trust posterior of (9.1) into an alternating two-step process leveraging conditional probabilities, i.e.,

$$\begin{aligned}&(1)\text{ Update track trust: } \Pr(\mathrm{T}^c_t \mid \mathrm{T}^a_{t-1}, Z_{1:t}, A_{1:t}) \\ &(2)\text{ Update agent trust: } \Pr(\mathrm{T}^a_t \mid \mathrm{T}^c_t, Z_{1:t}, A_{1:t}).\end{aligned} \tag{10}$$

With this separation, we can update track/agent trusts sequentially. The drawback of a Gibbs-style approach is a loss of formal convergence guarantees for the general case; in our case of simple univariate trust distributions with two parameters, we observe rapid convergence in practice.

B. Trust Pseudomeasurements (PSMs)

As it is hard to obtain $\Pr(Z_{1:t} | \mathrm{T}^a_t, \mathrm{T}^c_t, A_{1:t})$, making an exact Bayesian approach to trust estimation intractable. Instead, we introduce *trust PSMs* and the approximations

$$\begin{aligned}&\Pr(\mathrm{T}^c_t | \mathrm{T}^a_{t-1}, Z_{1:t}, A_{1:t}) \\ &\qquad \approx \Pr(\mathrm{T}^c_t | g^c(\mathrm{T}^a_{t-1}, Z_{1:t}, A_{1:t})) = \Pr(\mathrm{T}^c_t | \mathrm{P}^c_{1:t}) \\ &\Pr(\mathrm{T}^a_t | \mathrm{T}^c_t, Z_{1:t}, A_{1:t}) \\ &\qquad \approx \Pr(\mathrm{T}^a_t | g^a(\mathrm{T}^c_t, Z_{1:t}, A_{1:t})) = \Pr(\mathrm{T}^a_t | \mathrm{P}^a_{1:t})\end{aligned}$$

where g^c, g^a denote track/agent-focused PSM functions that map the measurements to the trust domain of $[0, 1]$. What follows from this is an *ad-hoc measurement* of track and agent trust at every frame $\mathrm{P}^c \leftarrow \{\rho^c_j\}$ and $\mathrm{P}^a \leftarrow \{\rho^a_k\}$.

Each PSM is a set of datapoints, each containing a value and an uncertainty; this is akin to e.g., a position measurement that provides a measured value along with a standard deviation of the measurement's uncertainty. Each PSM datapoint uses information only from a single track-agent pair, (track j, agent k). The PSM is then $\rho_j = \{(v_{j,k}, c_{j,k})\}$ where each $(v_{j,k}, c_{j,k})$ is a PSM datapoint, $v_{j,k} \in [0, 1]$ is the datapoint's value, and $c_{j,k} \in [0, 1]$ is the confidence (uncertainty) in the datapoint's value. The subscripts (j, k) indicate the datapoint leveraging information from track j and agent k. For example, a track j' may receive PSM datapoints from each of the agents such that its PSM is $\rho_{j'} = \{(v_{j',1}, c_{j',1}), (v_{j',2}, c_{j',2}), ...\}$.

Not all agents will see all tracks; the expected set of observations on each frame is informed by the FOV model for each agent, $\Phi_k(\cdot)$, which is an indicator function. Alg. 1 presents the PSM routine for a track: each agent expected by the FOV model to see the track provides a PSM datapoint with value of whether or not the agent has a detection near the track and confidence of the agent's trust; the confidence manifests conditional Gibb's sampling. Alg. 2 presents the PSM routine for an agent: each track that the agent is expected to see by its FOV model provides a PSM datapoint with value as the expectation of the track trust ($\mathbb{E}$) if the agent saw the track or the negation of track trust if the agent did not see the track. Confidence is set by variance ($\mathbb{V}$) of the track trust; the value and confidence manifest Gibb's sampling.

C. Trust Estimation

After introducing trust PSMs and making independence assumptions, we have reduced the estimation problem to the posteriors $\Pr(\mathrm{T}^c_{j,t} | \mathrm{P}^c_{j,1:t})$, $\Pr(\mathrm{T}^a_{j,t} | \mathrm{P}^a_{j,1:t})$. We assume the PSMs are i.i.d.; this is sub-optimal as the construction of PSMs requires verification against other agents, generating inter-agent correlations. We also expect the PSMs to exhibit autocorrelation due to the temporal nature of tracking. These assumptions, while sub-optimal, enable the use of simple parameter estimation algorithms.

A Bayesian approach is warranted from the perspective of small sample sizes and prior knowledge. Tracking will generate relatively few PSMs since the PSM function *requires FOV overlap from multiple agents*; observation density is expected to be sparse. Significant prior knowledge may also be available in the form of correlations between platform types or prior trust/distrust of particular agents.

The estimation process is identical for track and agent trust posteriors. The unknown parameters θ are random variables.

The probability distribution of trust for tracks is:

$$\Pr(\tau_j^c|\mathrm{P}_j^c) = \int \Pr(\tau_j^c, \theta_j^c|\mathrm{P}_j^c)d\theta_j^c = \int \Pr(\tau_j^c|\theta_j^c)\Pr(\theta_j^c|\mathrm{P}_j^c)d\theta_j^c.$$

The parameter posterior is: $\Pr(\theta_j^c|\mathrm{P}_j^c) \propto \Pr(\mathrm{P}_j^c|\theta_j^c)\Pr(\theta_j^c)$. The same procedure applies for agent trust, $\Pr(\tau_k^a|\mathrm{P}_k^a)$.

In practice, tracks are in either the state of being true objects or FPs. Thus, τ_j^c is the belief of a track being in one state or the other. As such, $\Pr(\mathrm{P}_j^c|\theta_j^c)$ naturally takes on a Bernoulli distribution with a single parameter, θ_j^c. For a Bernoulli likelihood, the standard choice of prior $\Pr(\theta_j^c)$ is the Beta distribution [13]. The Beta has two parameters ("hyperparameters"), (α, β), and is conjugate to the Bernoulli likelihood meaning that the posterior, $\Pr(\theta_j^c|\mathrm{P}_j^c)$, is also a Beta distribution. It is well-known that the Bayesian parameter update for the Beta-Bernoulli pair has a closed form. As the trust estimation update, we use each of the PSM datapoints to update the posterior parameters. For a track with PSM $\rho_j = \{(v_{j,k}, c_{j,k})\}$, the posterior of the trust parameter distribution, $\Pr(\theta_j^c|\mathrm{P}_j^c)$, will be updated via:

$$\begin{aligned} \alpha_{j,t}^c &= \alpha_{j,t-1}^c + \sum\nolimits_k c_{j,k}v_{j,k} \\ \beta_{j,t}^c &= \beta_{j,t-1}^c + \sum\nolimits_k c_{j,k}(1 - v_{j,k}). \end{aligned} \tag{11}$$

The same process applies for agent parameters, $(\alpha_{k,t}^a, \beta_{k,t}^a)$.

D. Simplified Trust-Aware Sensor Fusion

The final step is to perform trusted state estimation. A general model would consider that besides FPs/FNs the adversary could furnish incorrect state estimates of true objects; e.g., a translation outcome [6], [8]. In this work, we limit the adversary to only FP/FN outcomes. Thus, trust estimation is sufficient to confirm or remove tracks from the database.

Algorithm 1 Trust pseudomeasurement for track $\hat{x}_j^c$

```
Input: $K > 0$ agents with trusts $\tau_k^a$, $Z \leftarrow \{z_k\}$ detections
    from agents, $\hat{x}_j^c$ track of interest, FOV functions $\Phi_k(\cdot)$
Output: $\rho_j^c$ if $N_{exp} > 1$ else [ ]
  $\rho_j^c \leftarrow [\,]$
  for $k = 1...K$ do                                    ▷ loop over agents
      if $\Phi_k(\hat{x}_j^c)$ then                         ▷ if expected to see
          $N_{exp} \leftarrow N_{exp} + 1$
          if $\exists z_{i,k} \in z_k$ s.t. dist$(z_{i,k}, \hat{s}_j^c)$ is small then
              $v_{j,k} \leftarrow 1.0$, $c_{j,k} \leftarrow \tau_k^a$
          else                                          ▷ if not observed
              $v_{j,k} \leftarrow 0.0$, $c_{j,k} \leftarrow \tau_k^a$
          end if
          $\rho_j^c$.append(($v_{j,k}$, $c_{j,k}$))
      end if
  end for
```

V. Multi-Agent Trust Experiments

We evaluate our trust estimation models on two case studies and two sets of prior information. We find the availability of prior information on agent trust influences the certainty with which the model identifies (dis)trusted agents & tracks.

We consider three static agents and partially overlapping circular FOVs (Fig. 5), making object observations. We neglect benign FPs by assuming MTT can filter transient detections given the lack of temporal persistence. We evaluate two cases with adversaries. First, two agents are partially adversarially compromised and are providing malicious data in the form of persistent FPs in the FOV; second, only a single agent is providing FPs (see Figs. 5a and‘5b, respectively).

Algorithm 2 Trust pseudomeasurement for agent k

```
Input: $\Phi_k(\cdot)$ FOV for agent k, $\hat{X}_c \leftarrow \{\hat{x}_j^c\}$ tracks from
    MTT with trusts $\tau_j^c$, $\hat{X}_k \leftarrow \{\hat{x}_{j'}^k\}$ tracks from agent k's.
Output: $\rho_k^a$
  $\rho_k^a \leftarrow [\,]$
  for $\hat{x}_j^c \in \hat{X}_c$ do                 ▷ loop over tracks from central
      if $\Phi_k(\hat{x}_j^c)$ then                          ▷ if expected to see
          if $\hat{x}_j^c \in \hat{X}_k$ then                   ▷ if agent has match
              $v_j \leftarrow \mathbb{E}[\tau_j^c]$,      $c_j \leftarrow 1 - \mathbb{V}[\tau_j^c]$
          else                           ▷ if agent does not have match
              $v_j \leftarrow 1 - \mathbb{E}[\tau_j^c]$, $c_j \leftarrow 1 - \mathbb{V}[\tau_j^c]$
          end if
          $\rho_k^a$.append(($v_j$, $c_j$))
      end if
  end for
```

We implement a Kalman-filter-based multi-sensor MTT with canonical track scoring [3]. We perform Bayesian estimation of the parameter posteriors, $\Pr(\theta_j^c|\mathrm{P}_j^c)$, $\Pr(\theta_k^a|\mathrm{P}_k^a)$, for both track trust and agent trust estimation. We reparameterize the Beta from its canonical (α, β) form to a $(\lambda\phi, \lambda(1-\phi))$ form where $\phi = \alpha/(\alpha+\beta)$ is the mean and $\lambda = \alpha + \beta$ is known as the "precision". For each case, we consider two prior conditions: (1) Without prior information available regarding track/agent trust; here, an uninformative prior on all parameters is appropriate (e.g., $\theta \sim$ Beta$(0.5, 1)$) with modes near the extrema, reflecting that tracks either exist or do not exist – more uninformative than a uniform prior [13]. (2) There is prior information that Agent 1 is trusted; then, a prior of $\theta_{k_1} \sim$ Beta$(0.8, 10)$ is heuristically chosen for Agent 1 and the uninformative prior for others.

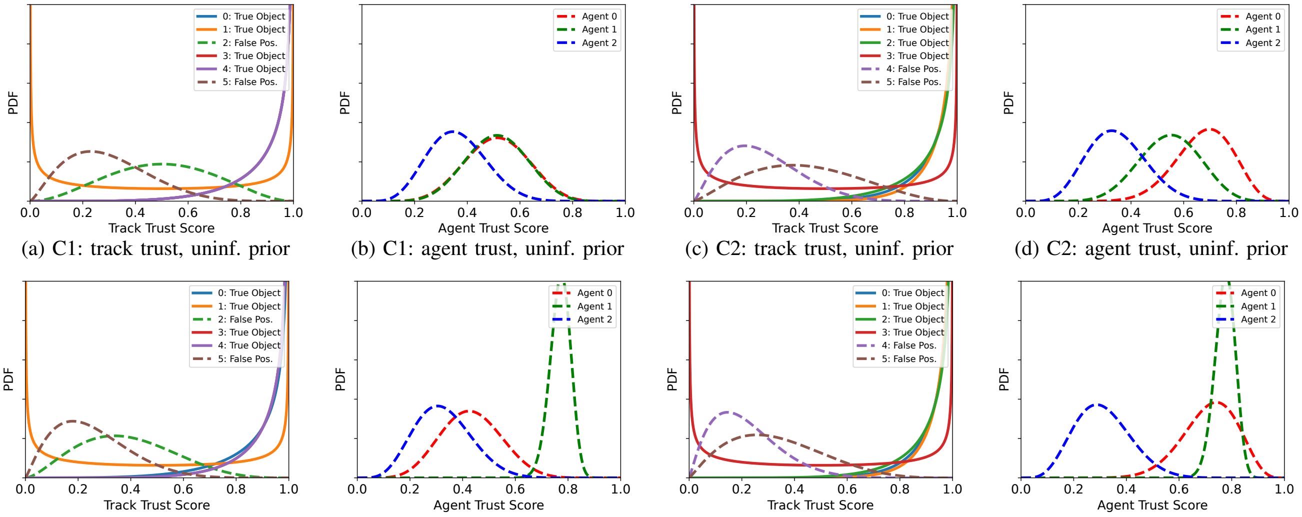
At each step, we add a small process noise to the trust posteriors by decreasing the precision parameter, ϕ, for all tracks and agents; since, in the absence of measurements, we should become less confident about trust estimates over time.

A. Results

Trust posteriors for tracks and agents in both cases with two sets of prior information are illustrated in Fig. 6.

a) Case 1, no prior; Figs 6a, 6b. Tracks corresponding to valid objects within multiple agents' FOVs are confirmed trusted (Tracks 0, 3, 4). Track 1 corresponds to a valid object, however, it is only visible in a single agent's FOV (see Fig. 5a). Thus, no trust update is performed and the track's trust remains as the prior. Track 5 (an FP) is viewable from all agents' FOVs; yet, since only Agent 2 is detecting it, the model distrusts Track 5. On the other hand, since Track 2 (an FP) is only viewable from Agents 0 and 1, the model finds it difficult to determine the trust on Agents 0 and 1.

b) Case 1, prior on Agent 1; Figs 6e, 6f. Adding prior information on the trust of Agent 1 significantly improves



(a) C1: track trust, uninf. prior (b) C1: agent trust, uninf. prior (c) C2: track trust, uninf. prior (d) C2: agent trust, uninf. prior

(e) C1: track trust, strong prior (f) C1: agent trust, strong prior (g) C2: track trust, strong prior (h) C2: agent trust, strong prior

Fig. 6: (a,b,e,f) Case 1 with Agents 0, 2 providing single FP each: (a, b) uninformative priors; (e, f) Strong Agent 1 prior; (c,d,g,h) Case 2 with Agent 2 providing multiple FPs: (c, d) easier to identify malicious agent due to multiple FPs. (g,h) Strong prior aids trust estimate.

the Case 1 outcomes. Tracks corresponding to true objects behave similarly as without the prior. Now, the model disambiguates the discrepancy between Agents 0 and 1 on Track 2; it now believes Agent 0 must be untrustworthy and providing FP detections to track 2 (an FP). The impact of prior is also reflected in the agent trust posterior (Agent 1 is now trusted).

c) Case 2, no prior; Figs 6c, 6d. Tracks corresponding to valid objects in shared regions of the FOVs are trusted (Tracks 0, 1, 2). Tracks isolated in a single agent's FOV maintain the prior (Track 3) As opposed to Case 1, the two FPs in Case 2 both originate from Agent 2. It is then easier for the model to identify that Agent 2 is distrusted since its information is inconsistent with both of the other agents. Consequentially, both FP tracks (Tracks 4, 5) and Agent 2 tend towards distrusted while Agents 0, 1 tend toward trusted.

d) Case 2, prior on Agent 1; Figs 6g, 6h. The addition of prior information in the form of a strong prior on Agent 1 accentuates the trust outcomes. The FP tracks become more distrusted while the trusted agents become more trusted.

Several general observations can be made. The model cannot verify the trust of tracks visible only from a single agent. In these cases, the distribution over trust remains as the prior, so, e.g., a sensor resource management may be used to dynamically direct sensing resources to mitigate uncertainty. Second, an even mix of positive (hit) and negative (miss) events for a single track from multiple agents is irresolvable in the absence of prior information. Third, accurate prior significantly improves the ability to estimate trust. Due to the alternating conditional Gibb's sampling step, agent trust is used to estimate track trust and vice verse. Thus, having a prior on either agent or track trust makes an immediate impact on the trust estimation process.

VI. Conclusion

Track scoring in MTT is provably vulnerable to adversarial attacks even when the number of benign agents significantly outnumbers the adversaries. To improve the security-awareness of MTT, we establish a Bayesian model that estimates the trust of agents and MTT's tracks by mapping the sensing inputs to a real-valued trust pseudomeasurement. Our trust estimation approach handles uncertain measurements and provides a probability distribution over the trust based on Bayesian updating. Trust estimation is capable of detecting and identifying adversarial false positive tracks while confirming true tracks as trusted entities in many cases.

References

[1] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: principles and techniques*. YBS publishing Storrs, CT, 1995, vol. 19.

[2] N. Bißmeyer, S. Mauthofer, K. M. Bayarou, and F. Kargl, "Assessment of node trustworthiness in vanets using data plausibility checks with particle filters," in *2012 IEEE Vehicular Net. Conf.*, 2012, pp. 78–85.

[3] S. Blackman, "Multiple-target tracking with radar," *Dedham*, 1986.

[4] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in vanets," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, 2004, pp. 29–37.

[5] C. N. Hadjicostis and A. D. Domínguez-García, "Trustworthy distributed average consensus," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 7403–7408.

[6] R. S. Hallyburton, Y. Liu, Y. Cao, Z. M. Mao, and M. Pajic, "Security analysis of camera-lidar fusion against black-box attacks on autonomous vehicles," in *31st USENIX SECURITY*, 2022, pp. 1–18.

[7] R. S. Hallyburton and M. Pajic, "Securing autonomous vehicles under partial-information cyber attacks on lidar data," *arXiv e-prints*, 2023.

[8] D. Hunt, K. Angell, Z. Qi, T. Chen, and M. Pajic, "MadRadar: A Black-Box Physical Layer Attack Framework on mmWave Automotive FMCW Radars," in *The 2024 NDSS*, 2024.

[9] Y. Mao and P. Tabuada, "Decentralized secure state-tracking in multi-agent systems," *IEEE Transactions on Automatic Control*, 2022.

[10] M. Marcus and H. Minc, *A survey of matrix theory and matrix inequalities*. Courier Corporation, 1992, vol. 14.

[11] Y. Mo and B. Sinopoli, "Secure estimation in the presence of integrity attacks," *IEEE Trans. Aut. Cont.*, vol. 60, no. 4, pp. 1145–1151, 2014.

[12] X. Ren, Y. Mo, J. Chen, and K. H. Johansson, "Secure state estimation with byzantine sensors: A probabilistic approach," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3742–3757, 2020.

[13] A. Shemyakin and A. Kniazev, *Introduction to Bayesian estimation and copula models of dependence*. John Wiley & Sons, 2017.

[14] R. W. Sittler, "An optimal data association problem in surveillance theory," *IEEE Trans. on Military Electronics*, pp. 125–139, 1964.

[15] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *3rd ACM Work. on Wireless Security*, 2004, pp. 1–10.

[16] W. Wang, G. Zeng, and T. Liu, "An autonomous trust construction system based on bayesian method," in *Conf. on Int. Agent Tech.*, 2006.