Attack detection and secure estimation under false data injection attack in cyber-physical systems

Arpan Chattopadhyay & Urbashi Mitra

***Abstract*—In this paper, secure, remote estimation of a linear time-varying Gaussian process via observations at multiple sensors is considered. Such a framework is relevant to many cyberphysical systems and internet-of-things applications. Sensors make sequential measurements that are shared with a fusion center; the fusion center applies a form of optimal filtering to make its estimates. The challenge is the presence of malicious sensors which can inject anomalous observations to skew the estimates at the fusion center. The set of malicious sensors may be time-varying. The problems of malicious sensor detection and secure estimation are considered. First, a novel detector to detect injection attack on an unknown sensor subset is developed. Next, an algorithm for secure estimation is proposed. The proposed estimation scheme uses a novel filtering and learning algorithm, where an optimal filter is learnt over time by using the sensor observations in order to filter out malicious sensor observations while retaining other sensor measurements. Numerical results demonstrate the efficacy of the proposed algorithms.**

***Index Terms*—Secure remote estimation, CPS security, false data injection attack, Kalman filter, stochastic approximation.**

I. Introduction

Cyber-physical systems (CPS) combine the cyber world and the physical world via seamless integration of sensing, control, communication and computation. CPS has widespread applications such as networked monitoring and control of industrial processes, intelligent transportation systems, smart grid, and environmental monitoring. Most of these applications critically depend on reliable remote estimation of a physical process via multiple sensors over a wireless network. Hence, any malicious attack on sensors can have a catastrophic impact. Such attacks could be of two types: (i) deception or integrity attacks, where the attacker attempts to modify the information sent via the data packets; e.g., replay attack [1], [2], and (ii) denial-of-service attack where the attacker attempts to block resources for the system, e.g., wireless jamming attack to block bandwidth usage [3]. False data injection (FDI) attacks belong to the first category. FDI attacks modify the data sent by some sensors to the fusion center, either by breaking the cryptography of the data packets or by physical manipulation of the sensors (e.g., putting a heater near a temperature sensor). Therefore, it is necessary to develop efficient security algorithms to combat FDI attacks.

The problem of FDI attack and its detection has received recent attention. In [4], conditions for undetectable attack are developed, and the minimum number of sensors to be attacked to ensure undetectability is computed. In [5], a linear deception attack scheme that can fool the popular χ^2 detector is provided. Later, new detection algorithm against such linear deception attacks is designed in [6], where observations are available from a few known *safe* sensor nodes. The optimal attack strategy to steer the control of CPS to a target value is provided in [7], while ensuring a constraint on the attack detection probability. Centralized and decentralized attack detection schemes for *noiseless* systems have been developed in [8]. Coding of sensor output for efficient attack detection using χ^2 detector is proposed in [9]; but this scheme will be vulnerable if the attacker breaks the encryption of the encoder. Attack-resilient state estimation of a dynamical system with only *bounded* noise has been discussed in [10]. Efficient attack detection and secure estimation schemes for linear Gaussian systems under cyber attack on a static, unknown sensor subset have been developed in [11], but this detector is not designed to tackle the linear deception attack of [5], since it uses Kalman innovation sequence based detection algorithm. Sparsity models to characterize the switching location attack in a *noiseless* linear system and state recovery constraints for various attack modes have been described in [12]. Attack detection, secure estimation and control in the presence of FDI attack for power systems are addressed in [13], [14].

In this paper, we make the following contributions:

- In Section III, we propose an algorithm for FDI attack detection, that does not require observations from any *safe* sensor. Instead, the algorithm detects an attack via anomaly detection between sensor subsets.
- In Section IV, we develop a learning algorithm that learns a linear filter over time; the goal is to minimize a combination of the estimation error in the absence of attack and the anomaly in estimates returned by various sensor subsets, in order to obtain a filter that achieves small error and at the same time suppresses observations from possibly malicious nodes.
- Numerical results show that the proposed detection algorithm offers a much higher attack detection probability and a smaller false alarm probability than a competetive algorithm [6]. Also, the proposed adaptive filtering strategy offers a mean squared error (MSE) close to two competetive algorithms that use more side information, and performs much better than simple Kalman filtering.

The rest of the paper is organized as follows. Preliminaries are discussed in Section II. The attack detection scheme is described in Section III. The secure estimation algorithm to combat the FDI attack is described in Section IV. Numerical results are provided in Section V, followed by the conclusions in Section VI.

The authors are with the Department of Electrical Engineering, University of Southern California. Email: {achattop,ubli}@usc.edu

This work was supported by the following funding sources: ONR N00014-15-1-2550, NSF CNS-1213128, NSF CCF-1718560, NSF CCF-1410009 , NSF CPS-1446901, and AFOSR FA9550-12-1-0215

978-1-5386-0579-0/18/$31.00 ©2018 IEEE

II. Preliminaries

A. Sensing and remote estimation model

We consider a set of smart sensors $\mathcal{N} := \{1, 2, \cdots, N\}$, which are sensing a discrete-time stochastic process $\{x(t)\}_{t\geq 0}$. The sensors send their observation directly to a fusion center via *error-free* wireless links so that the fusion center can estimate $\hat{x}(t)$ at each time t. The physical process $\{x(t)\}_{t\geq 0}$ (where $x(t) \in \mathbb{R}^q$) is a linear Gaussian process that evolves according to the following equation:

$$x(t+1) = Ax(t) + w(t), \tag{1}$$

where $w(t)$ is a zero-mean Gaussian noise vector with covariance matrix Q, and is i.i.d. across t. The scalar or vector observation made by sensor i is given by the following observation equation if sensor i is used in sensing:

$$y_i(t) = C_i x(t) + v_i(t), \tag{2}$$

where C_i is a matrix of appropriate dimension and $v_i(t)$ is a Gaussian observation noise with covariance matrix R_i. Observation noise $v_i(t)$ is assumed to be independent across sensors and i.i.d. across time. The pair $(A, Q^{\frac{1}{2}})$ is assumed to be stabilizable and the pair (A, C_i) is assumed to be detectable for all $i \in \mathcal{N}$.

The goal of the fusion center is to minimize the time-average expected mean squared error (MSE) in estimation:

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}||x(t) - \hat{x}(t)||^2. \tag{3}$$

If all sensors send their observations to the fusion center in real time, then the system is equivalent to a sensor and a remote estimator with real-time communication. The sensing and observation models can be rewritten as:

$$\begin{aligned} x(t+1) &= Ax(t) + w(t) \\ y(t) &= Cx(t) + v(t), \end{aligned} \tag{4}$$

where $v(t) \sim \mathcal{N}(0, R)$ is the observation noise and $y(t) \in \mathbb{R}^{m\times 1}$ is the complete observation vector. The minimum mean-squared error (MMSE) estimator in this case is a linear filter called Kalman filter (see [15]):

$$\begin{aligned} \hat{x}_{t+1|t} &= A\hat{x}_t \\ P_{t+1|t} &= AP_tA' + Q \\ K_{t+1} &= P_{t+1|t}C'(CP_{t+1|t}C' + R)^{-1} \\ \hat{x}_{t+1} &= \hat{x}_{t+1|t} + K_{t+1}(y(t+1) - C\hat{x}_{t+1|t}) \\ P_{t+1} &= (I - K_{t+1}C)P_{t+1|t}, \end{aligned} \tag{5}$$

where $\hat{x}_{t+1}$ is the MMSE estimate and P_{t+1} is the the error covariance matrix for the estimate $\hat{x}_{t+1}$, provided that the iteration starts from $\hat{x}_0 \sim \mathcal{N}(0, P_0)$. It has been shown in [15] that $\lim_{t\to\infty} P_{t+1|t} = \bar{P}$ exists and is the unique fixed point to the $P_{t+1|t}$ iteration called the *Riccati equation*. Another quantity of interest is the innovation sequence $z_t := y(t) - C\hat{x}_{t|t-1}$; it was proved in [15] that $\{z_t\}_{t\geq 1}$ is a zero-mean Gaussian sequence which is pairwise independent across time and whose covariance matrix in the steady state is $\Sigma_z := (C\bar{P}C' + R)$.

B. False data injection (FDI) attack

Any *unknown* subset $\mathcal{A} \subset \mathcal{N}$ of sensors can be under attack by an external attacker. Any sensor $i \in \mathcal{A}$ sends an observation that follows the following attack equation:

$$y_i(t+1) = C_i x(t) + e_i(t) + v_i(t) \tag{6}$$

where $e_i(t)$ is an error term injected by the attacker. The goal of the attacker is to insert the false data sequance $\{e_i(t)\}_{t\geq 0}$ for all $i \in \mathcal{A}$ so as to maximize the MSE given by (3).

The χ^2 detector: Since the innovation sequence at steady state behaves like a zero mean Gaussian noise sequence with covariance matrix Σ_z when there is no attack, a natural technique (see [5], [6]) to detect any FDI attack is to detect any anomaly in the innovation sequence. This is done by observing the innovation sequence over a pre-specified window of J time-slots, and declaring that an attack has occured if the condition $\sum_{t=\tau-J+1}^{\tau} z_t'\Sigma_z^{-1} z_t \geq \eta$ is satisfied at time τ, where η is a pre-specified threshould used to tune the false alarm probability.

In [5], a linear injection attack to fool the χ^2 detector is constructed; at time t, the malicious sensor(s) modifies the innovation vector as $\tilde{z}_t = Tz_t + b_t$, where T is a square matrix and $b_t \sim \mathcal{N}(0, \Sigma_b)$ is i.i.d. Gaussian noise. It was shown in [5] that $\tilde{z}_t \sim \mathcal{N}(0, \Sigma_{\tilde{z}})$ where $\Sigma_{\tilde{z}} = T\Sigma_z T' + \Sigma_b$. Hence, if T and Σ_b are chosen such that Σ_b is a positive semidefinite covariance matrix and $\Sigma_{\tilde{z}} = \Sigma_z$, then the modified innovation sequence $\{\tilde{z}_t\}_{t\geq 1}$ will have the same distribution $\mathcal{N}(0, \Sigma_z)$ as $z(t)$, and hence the detection probability of the χ^2 detector will remain unaffected even under the attack. The estimation error is maximized when the attacker just inverts the innovation sequence (i.e., when $T = -I$ and $b_t = 0$). However, the authors of [6] proposed another efficient scheme to detect such attack. The detection algorithm in [6] assumed the presence of a few *safe* sensors which can never be attacked, and hence one can detect an attack by exploiting any anomaly between the observations made by the safe sensors and the sensors potentially under attack. The assumption of the existence of a set of safe sensors is restrictive, and the design of efficient attack detection and secure estimation schemes in the absence of such safe sensors in the topic of our current paper.

III. Attack detection

In this section, we develop an efficient detector for the FDI attack on a *static* sensor subset, though the proposed detector can be heuristically used under switching location attack. We assume that, at most n_0 sensors can be under attack at any given time instant.

The detection problem is mathematically represented as a hypothesis testing problem on the two hypotheses:

- $\mathcal{H}_0$: there is no attack; $e_i(t) = 0$ for all $i \in \mathcal{N}$, $t = 1, 2, \cdots$
- $\mathcal{H}_1$: there is an attack; $e_i(t) \neq 0$ for some $i \in \mathcal{N}$

with observation sequance $\{y(t)\}_{t\geq 1}$. Note that, due to the complicated dynamics involved in Kalman filtering, it is difficult to carry out standard hypothesis testing schemes. Also, due to the unavailability of any known safe sensor, we cannot compare the innovation sequence against any reliable quantity.

However, if a subset of sensors is under attack, then the process estimate obtained only from these sensor observations is likely to have high error, and hence is supposed to be significantly different from the estimates made by other sensor observations. We exploit this fact to develop an attack detector.

Let us denote by $\hat{x}_{\mathcal{B}}(t)$ the process estimate returned by an optimal Kalman filter that uses observations made by the sensor subset $\mathcal{B}$ only (see (5) for Kalman filtering equation on a sensor set). Let us denote the covatiance matrix of the anomaly $(\hat{x}_{\mathcal{B}}(t)-\hat{x}_{\mathcal{B}^c}(t))$ by $\bar{P}_{\mathcal{B},\mathcal{B}^c}$ under steady state. Clearly, if there is no attack, then, under steady state, $(\hat{x}_{\mathcal{B}}(t)-\hat{x}_{\mathcal{B}^c}(t))$ has a distribution $\mathcal{N}(0,\bar{P}_{\mathcal{B},\mathcal{B}^c})$; since the error $(\hat{x}_{\mathcal{B}}(t)-x(t))$ and $(\hat{x}_{\mathcal{B}^c}(t)-x(t))$ are zero-mean Gaussian, $(\hat{x}_{\mathcal{B}}(t)-\hat{x}_{\mathcal{B}^c}(t))$ also follows the Gaussian distribution. Hence, one can detect an attack by checking whether $(\hat{x}_{\mathcal{B}}(t)-\hat{x}_{\mathcal{B}^c}(t))$ is coming from the distribution $\mathcal{N}(0,\bar{P}_{\mathcal{B},\mathcal{B}^c})$ for each subset $\mathcal{B}$ of size n_0. The covariance matrix $\bar{P}_{\mathcal{B},\mathcal{B}^c}$ can be pre-computed by simulating the process beforehand.

The algorithm to detect and localize an attack is given below.

Algorithm 1. Off-line pre-computation: *Simulate the dynamic process $x(t)$ of (1) off-line, and, for each sensor subset $\mathcal{B}$ of size n_0 and its complement set $\mathcal{B}^c$, use a separate Kalman filter to estimate $\hat{x}_{\mathcal{B}}(t)$ and $\hat{x}_{\mathcal{B}^c}(t)$ virtually using observations coming only from $\mathcal{B}$ and $\mathcal{B}^c$ respectively. For each subset $\mathcal{B}$ of size n_0, compute the anomaly covariance matrix $\bar{P}_{\mathcal{B},\mathcal{B}^c} := \lim_{\tau\to\infty}\frac{1}{\tau}\sum_{t=1}^{\tau}(\hat{x}_{\mathcal{B}}(t)-\hat{x}_{\mathcal{B}^c}(t))(\hat{x}_{\mathcal{B}}(t)-\hat{x}_{\mathcal{B}^c}(t))'$; this is the limiting covariance matrix of $(\hat{x}_{\mathcal{B}}(t)-\hat{x}_{\mathcal{B}^c}(t))$. Fix any observation window size $J>0$.*

Attack detection in the physical system: *Any attack on the sensors observing the real $x(t)$ process is detected using the following test, which uses a separate Kalman filter for each subset $\mathcal{B}$ of size n_0.*

At any real time instant t, check if

$$\max_{\{\mathcal{B}\in 2^{\mathcal{N}}:|\mathcal{B}|=n_0\}}\sum_{\tau=t-J+1}^{t}(\hat{x}_{\mathcal{B}}(\tau)-\hat{x}_{\mathcal{B}^c}(\tau))'\bar{P}^{-1}_{\mathcal{B},\mathcal{B}^c}(\hat{x}_{\mathcal{B}}(\tau)-\hat{x}_{\mathcal{B}^c}(\tau))>\eta$$

where η is a pre-specified threshold. If the condition is satisfied, declare that an attack has happended on the maximizing set

$$\arg\max_{\mathcal{B}\in 2^{\mathcal{N}}:|\mathcal{B}|=n_0}\sum_{\tau=t-J+1}^{t}(\hat{x}_{\mathcal{B}}(\tau)-\hat{x}_{\mathcal{B}^c}(\tau))'\bar{P}^{-1}_{\mathcal{B},\mathcal{B}^c}(\hat{x}_{\mathcal{B}}(\tau)-\hat{x}_{\mathcal{B}^c}(\tau))$$

.

So long as there is no attack detection, a Kalman filter is used to estimate $x(t)$.

Discussion of Algorithm 1:

- The detection step is exactly the same as the standard χ^2 test used to test whether a sequence of random vectors are coming from a desired Gaussian distribution, except that this test is conducted on all possible subsets of size n_0, and hence the max operation is needed.
- The false alarm probability can be controlled by controlling the threshold η.
- This algorithm is only meant for attack detection, with the assumption that necessary measures will be taken if an attack is detected. Secure estimation of $x(t)$ even in the presence of FDI attack is described in Section IV.

1) A learning scheme to find the optimal η for a given target on the false alarm probability: The false alarm probability PFA under Algorithm 1 is defined as:

$$\begin{aligned} & PFA \\ = & \lim_{t\to\infty}\mathbb{P}\Big(\max_{\mathcal{B}\in 2^{\mathcal{N}}:|\mathcal{B}|=n_0}\sum_{\tau=t-t_0}^{t}(\hat{x}_{\mathcal{B}}(\tau)-\hat{x}_{\mathcal{B}^c}(\tau))' \\ & \bar{P}^{-1}_{\mathcal{B},\mathcal{B}^c}(\hat{x}_{\mathcal{B}}(\tau)-\hat{x}_{\mathcal{B}^c}(\tau))>\eta\Big|\mathcal{H}_0\Big). \end{aligned} \tag{7}$$

In order to satisfy the constraint $PFA\le\alpha$, we need to choose an optimal threshold η^*_α in Algorithm 1. The optimal η^*_α can be computed by using the following stochastic approximation step in the pre-computation phase of Algorithm 1:

Learning η^*_α : Consider a positive sequence $\{a(t)\}_{t\ge 0}$ such that $\sum_{t=0}^{\infty}a(t)=\infty$ and $\sum_{t=0}^{\infty}a^2(t)<\infty$. After computing $\bar{P}_{\mathcal{B},\mathcal{B}^c}$ for all subset $\mathcal{B}$ of size n_0, simulate the $x(t)$ process again off-line. Also, maintain a detector as in Algorithm 1 with an initial threshold $\eta(0)$. Let us denote the number of false alarm triggers made by this detector up to time $(t-1)$ in the simulated process by n_{t-1}.

At time t (in the simulated process), check if $\max_{\mathcal{B}\in 2^{\mathcal{N}}:|\mathcal{B}|=n_0}\sum_{\tau=t-t_0}^{t}(\hat{x}_{\mathcal{B}}(\tau)-\hat{x}_{\mathcal{B}^c}(\tau))'\bar{P}^{-1}_{\mathcal{B},\mathcal{B}^c}(\hat{x}_{\mathcal{B}}(\tau)-\hat{x}_{\mathcal{B}^c}(\tau))>\eta(t)$; if this condition is satisfied, update $n_t=n_{t-1}+1$, else $n_t=n_{t-1}$. Then, update the threshold $\eta(t+1)=[\eta(t)+a(t)(\mathbb{I}(n_t>n_{t-1})-\alpha)]^l_0$.

Discussion:

- The $\eta(t)$ update scheme is a stochastic approximation algorithm (see [17]).
- The goal of the $\eta(t)$ update scheme is to meet the false alarm probability constraint with equality. If a false alarm is triggered at time t in the simulation, $\eta(t)$ is increased; else, $\eta(t)$ is decreased. By the theory of [17], it is straightforward to show that $\lim_{t\to\infty}\eta(t)=\eta^*_\alpha$ almost surely, and $\lim_{t\to\infty}\mathbb{P}(n_{t+1}>n_t)=\alpha$.
- l is a large positive number such that $\eta^*_\alpha\in(0,l)$. The projection operation is used to ensure boundedness of the $\eta(t)$ iterates.

2) Switching location attack: In this case, at any time t, a sensor subset $\mathcal{A}(t)$ is subject to the FDI attack, where $|\mathcal{A}(t)|\le n_0$. The subset $\mathcal{A}(t)$ could be chosen i.i.d. across time, or according to a positive recurrent Markov chain. In this case, all possible sensor subsets will potentially be subject to attack over time, but still Algorithm 1 can be used for attack detection, though it is hard to provide a performance guarantee for Algorithm 1. Note that, if $\mathcal{A}(t)$ constitutes a stationary sequence, one can still use the learning scheme for $\eta(t)$ in order to obtain the optimal threshold η^*_α.

IV. Secure estimation

In this section, we will provide an algorithm to obtain a reliable estimate $\hat{x}(t)$ in the presence of FDI attacks, without explicitly detecting the malicious sensor subset. This algorithm

is useful when it is not possible for the system administrator to take necessary measure even upon the detection of an attack (e.g., if a heater is deliberately kept by an attacker near a temperature sensor, it may not be always be possible to physically remove the heater).

A. A learning algorithm for secure estimation

Note that, any sensor observation is ignored if the corresponding entries in the Kalman gain matrix K_{t+1} in (5) are set to 0. Ideally, one should ignore the readings from the malicious sensors, and if done so, the anomalies in estimates from various sensor subsets will be small. However, since the estimation error depends collectively on the process noise, the observation noise and the noise injected by the attacker, a reasonable technique for reliable estimation would be to dynamically learn an optimal gain matrix that minimizes a combination of the MMSE in the absence of attack and the anomalies in estimates returned by various sensor subsets.

Now, let us assume that the attack is a switching location attack, where the attacked sensor subset sequence $\{\mathcal{A}(t)\}_{t\geq 0}$ comes either from an i.i.d. sequence or a Markov chain (static attack is a special case). We restrict the discussion to the class of linear estimators. The estimator we consider is similar to the Kalman filter in (5), except that the Kalman gain matrix K_{t+1} is learnt via a stochastic gradient descent scheme so as to minimize the following time-average cost function over the gain matrix sequence $\{K_t\}_{t\geq 0}$:

$$\limsup_{\tau\uparrow\infty} \frac{1}{\tau}\sum_{t=0}^{\tau} \underbrace{\mathbb{E}(trace(P_t) + \lambda \max_{\mathcal{B}\in 2^{\mathcal{N}}:|\mathcal{B}|=n_0} ||\hat{x}_{\mathcal{B}}(t) - \hat{x}_{\mathcal{B}^c}(t)||^2)}_{:=c(t)} \quad (8)$$

The first term $trace(P_t)$ in the single-stage cost $c(t)$ represents the MSE in the estimate if a gain matrix sequence $K_0, K_1, K_2, \cdots$ is used for estimation. Note that, P_t can be calculated iteratively. The second term penalizes any anomaly between the estimates $\hat{x}_{\mathcal{B}}(t)$ and $\hat{x}_{\mathcal{B}^c}(t)$ *when the restriction of K_t to the subsets $\mathcal{B}$ and $\mathcal{B}^c$ are used as gain matrices applied to the observations coming from these sensor subsets.* The second penalty term with a sufficiently large multiplier $\lambda \geq 0$ will ensure that, under any possible attack, the linear filter assigns less weight in the gain matrix for the sensors that are under attack.

1) The proposed algorithm: Due to unavailability of any closed-form expression of the cost function in (8), direct computation of a gradient estimate w.r.t. K_t is not possible. However, we seek to minimize cost function (8) by iteratively learning an optimal gain matrix K^* over time, via a stochastic gradient descent algorithm. Hence, we employ simultaneous perturbation stochastic approximation (SPSA, see [16]) for this optimization problem. In SPSA, all elements of K_t are perturbed simultaneously by a random vector in two opposite directions, the single stage cost function is evaluated for these two perturbed gain matrices, and a noisy estimate of the gradient of the single stage cost is obtained from this. This noisy estimate of the gradient is then used in a stochastic gradient descent algorithm for asymptotically minimizing the single stage cost function $c(t)$.

The algorithm uses two positive sequences $\{a(t)\}_{t\geq 0}$ and $\{b(t)\}_{t\geq 0}$ that satisfy the following conditions: (i) $\sum_{t=0}^{\infty} a(t) = \infty$, (ii) $\sum_{t=0}^{\infty} a^2(t) < \infty$, (iii) $\lim_{t\to\infty} b(t) = 0$, and (iv) $\lim_{t\to\infty} \frac{a^2(t)}{b^2(t)} < \infty$. The first two conditions are standard for stochastic approximation (see [17]). The third condition ensures that the gradient estimate is asymptotically unbiased. The fourth condition is a technical condition required for the convergence of SPSA (see [16]).

Pick a small $\delta > 0$. Let us define $\mathcal{K} := \{K \in \mathbb{R}^{q\times m} : ||\lambda_{max}(I - KC)|| \leq 1 - \delta\}$.

The proposed algorithm is described below.

Algorithm 2. *Start with an initial K_0, P_0 and $\hat{x}(0)$. Choose a large number $l > 0$ and a small number $\delta > 0$.*

At time $t = 1, 2, 3, \cdots$:

1) *Collect the observation vector y_t from all sensors.*
2) *Declare $\hat{x}(t) = A\hat{x}(t-1) + K_t(y_t - CA\hat{x}(t-1))$.*
3) *Pick a random matrix Δ_t such that each entry in Δ_t is chosen from $\{-1, 1\}$ independently with probability $\frac{1}{2}$. Let us define $K_t^+ := K_t + b(t)\Delta_t$ and $K_t^- := K_t - b(t)\Delta_t$.*
4) *Compute the estimates $\hat{x}^+(t) = A\hat{x}(t-1) + K_t^+(y_t - CA\hat{x}(t-1))$ and $\hat{x}^-(t) = A\hat{x}(t-1) + K_t^-(y_t - CA\hat{x}(t-1))$ as if the Kalman gain matrices K_t^+ and K_t^- were used for estimation at time t.*
5) *Compute $P_t^+ := (I - K_t^+C)(AP_{t-1}A' + Q)(I - K_t^+C)' + K_t^+R(K_t^+)'$ and similarly compute P_t^-.*
6) *Compute $c^+(t) := trace(P_t^+) + \lambda \max_{\mathcal{B}\in 2^{\mathcal{N}}:|\mathcal{B}|=n_0} ||\hat{x}_{\mathcal{B}}^+(t) - \hat{x}_{\mathcal{B}^c}^+(t)||^2$, and similarly compute $c^-(t)$.*
7) *For each entry $K_t(i,j)$, do the following SPSA update:*

$$K'_{t+1}(i,j) = \left[K_t(i,j) - a(t)\frac{c^+(t) - c^-(t)}{2b(t)\Delta_t(i,j)}\right]_{-l}^{l} \quad (9)$$

and project K'_{t+1} onto $\mathcal{K}$ in order to obtain K_{t+1}.

8) *Return to step* 1.

Discussion of Algorithm 2:

- $K'_{t+1}(i,j)$ is projected onto a compact interval $[-l, l]$ to ensure stability of the iteration (9). Also, the spectral radius of $(I - K_{t+1}C)$ is maintained at less than $(1-\delta)$ to ensure that the error covariance matrix P_t remains bounded. A standard result says that, the covariance matrix P_t of the estimation error $\hat{x}(t) - x(t)$ varies according to the following recursive equation:

$$P_t := (I - K_tC)(AP_{t-1}A' + Q)(I - K_tC)' + K_tR(K_t)', \quad (10)$$

  when the gain matrix K_t is chosen arbitrarily (not optimally as in (5)). Step 5 is motivated by the above expression.
- Equation (9) is a stochastic gradient descent algorithm where a noisy estimate of the gradient of $\mathbb{E}c(t)$ w.r.t. K_t is used instead of the true gradient. The noisy gradient estimate is $\frac{c^+(t)-c^-(t)}{2b(t)\Delta_t(i,j)}$.

Let us consider the problem (8). Let us define a function $C(K)$ which is the time-average cost (and also the limiting expected

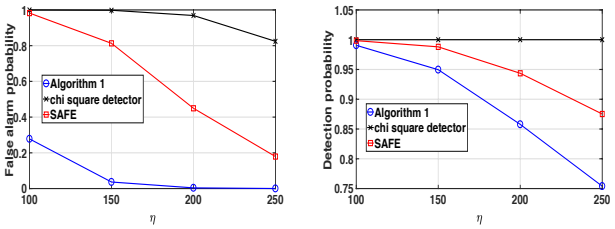

Figure 1. Performance comparison of Algorithm 1 against χ^2 and SAFE detectors, under static attack. $N = 8$, $n_0 = 3$, $k = 4$, $q = 3$. Three safe sensors are known to the SAFE algorithm. $\eta = 150$ is chosen for Algorithm 1 and $\eta = 250$ is chosen for χ^2 and SAFE detectors.

single-stage cost) achieved if a linear estimator as in Kalman filter is used with a constant gain matrix K for all t.

Assumption 1. $C(K)$ is Lipschitz continuous in $K \in [-l, l]^{q\times m} \cap \mathcal{K}$.

Remark 1. Let us recall that,

$$P_t := (I - K_tC)(AP_{t-1}A' + Q)(I - K_tC)' + K_tR(K_t)'$$

. Since the spectral radius of $(I - K_tC)$ is less than or equal to $(1 - \delta) < 1$, the P_t iteration converges almost surely, and hence the MSE under Algorithm 2 is uniformly bounded across sample paths. If a constant gain matrix K is used, it is still easy to prove that $\lim_{t\to\infty} P_t$ is Lipschitz continuous in $K \in [-l, l]^{q\times m} \cap \mathcal{K}$. Thus, Assumption 1 is specifically required for the second term in the expression of $c(t)$ in (8).

Conjecture 1. *Under Algorithm 2 and Assumption 1, the iterates* $\{K_t\}_{t\geq 1}$ *converges almost surely to the set* $\{K : \nabla_K(C(K)) = 0\}$*, provided that each such stationary point belongs to the interior of* $[-l, l]^{q\times m} \cap \mathcal{K}$.

Remark 2. Conjecture 1 says that $\{K_t\}$ converges to the set of local minima of $C(K)$ in case there is no saddle point that is not a local minimum.

V. Numerical results

In this section, we numerically demonstrate the efficacy of Algorithm 1 for attack detection and Algorithm 2 for secure estimation. For attack detection, we compare the performance of Algorithm 1 with the traditional χ^2 detector, and also with the detector of [6]. The detector of [6] assumes the availability of a set of *safe sensors*. At each time t, observations are collected from all sensors, but the safe sensor observations are used by the Kalman filter to generate an initial estimate. Then, the observations from potentially unsafe sensors are passed through a χ^2 detector, and those observations are used in a Kalman filter to obtain the final estimate $\hat{x}(t)$ if and only if the χ^2 detector is not triggered. We call this algorithm SAFE.

For secure estimation, we also compare the performance of Algorithm 2 with a blind Kalman filter oblivious to cyber-attack (KALMAN), and a Kalman filter which perfectly knows the malicious sensors and ignores their onservations (we call this estimator PERFECT).

	Algorithm 1	χ^2 detector	SAFE
P_d	0.9900	1	0.8988
PFA	0.0285	0.7509	0.2337

Table I
Comparison among attack detection algorithms; $N = 10$, $n_0 = 2$, $k = 4$, $J = 10$, $\eta = 150$ for Algorithm 1 and $\eta = 250$ for χ^2 and SAFE detectors. Three safe sensors are known to the SAFE detector.

In each case, we consider an independent realization of a system with the following parameters. The state transition matrix A is taken as a randomly generated $q \times q$ stochastic matrix multiplied by 0.5. State noise covariance matrix Q is chosen to be a positive semidefinite (PSD) matrix whose square root is 0.1 multiplied by a $q\times q$ matrix whose entires are independently generated uniformly from the interval $[-1, 1]$. The observarion noise covariance matrix R is also chosen similarly. Observation matrix $C \in \mathbb{R}^{kN\times q}$ is chosen randomly, with each entry drawn uniformly and independently from the interval $[0, 1]$; the assumption here is that the observation made by each sensor is a k-dimensional column vector. We also assume that at most n_0 sensors can be under attack at a time. The attacker inverts the innovation vectors coming from the malicious sensors; this is the worst possible linear attack [5].

A. Detection under static attack

Here we consider that sensors $\{1, 2, \cdots, n_0\}$ are under attack (which is not known to the fusion center). We run Algorithm 1 for a large number of time slots; the fraction of time slots where the detector is triggered is defined as the detection probability P_d. We also compute the fraction of slots when the detector is triggered in absence of any attack, and use this fraction as a measure of the false alarm probability (PFA). We perform the same operation for both the χ^2 detector and the SAFE detector. In this particular numerical example, we have fixed $J = 10$ as the observation window for all detectors.

From Figure 1, we observe that (i) false alarm probability and detection probability decrease with η, and (ii) false alarm and detection probabilities are smallest for Algorithm 1 for a given η. However, in all simulated problem instances, it turns out that Algorithm 1 has the best detection performance. This becomes evident from Table I; for specific choices of η for the three detectors, Algorithm 1 has higher detection probability and smaller false alarm probability than SAFE. False alarm probability of χ^2 detector is extremely high. Hence, Algorithm 1 provides a better attack detector than the state-of-the-art detector of [6].

Note that, Algorithm 1 requires us to pre-compute $\bar{P}_{\mathcal{B},\mathcal{B}^c}$ for $\binom{N}{n_0}$ possible subset pairs; hence, we gain the detection performance improve w.r.t. SAFE (which uses more side information) at the expense of more computation.

B. Secure estimation under static attack

Here we compare the time-average MSE of Algorithm 2 (for $\lambda = 0.3$) with BLIND, PERFECT and SAFE. We ran the simulation for a number of independent problem instances, and computed time-average MSE of various algorithms along

Algorithm 2 (with $\lambda = 0.3$)	KALMAN	PERFECT	SAFE
0.0365	0.2177	0.0183	0.0453
0.0405	0.3126	0.0151	0.0261
0.0234	0.2014	0.0126	0.0250
0.0409	0.3021	0.0169	0.0330

Table II
COMPARISON AMONG MSE OF VARIOUS SECURE ESTIMATION ALGORITHMS UNDER STATIC ATTACK. $N = 6$ SENSORS, AT MOST $n_0 = 2$ SENSORS ARE UNDER ATTACK, $\lambda = 0.3$ FOR ALGORITHM 2, $k = 3$, TWO SAFE SENSORS KNOWN TO THE SAFE ALGORITHM, $\eta = 200$ IN THE SAFE ALGORITHM, $J = 10$ IN SAFE.

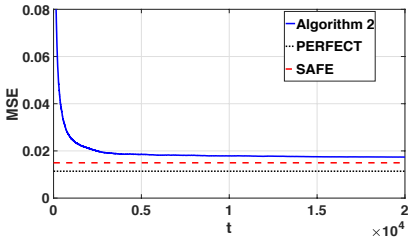

Figure 2. Performance comparison of Algorithm 2 against PERFECT and SAFE, under static attack. $N = 10$, $n_0 = 3$, $k = 4$, $q = 3$, $\lambda = 0.5$. Three safe sensors are known to the SAFE algorithm.

a single sample path. We observe from Figure 2 that the time-average MSE of Algorithm 2 converges close to that of SAFE and PERFECT. However, from Table II, we observe that Algorithm 2 can sometimes yield smaller MSE than SAFE, without using the additional side information. Also, Algorithm 2 has an MSE much smaller that KALMAN.

C. Secure estimation under switching location attack

Here we consider an attack model where, at time instances $t = 1, T+1, 2T+1, \cdots$ (with $T = 20$), a random sensor subset of size n_0 is chosen in an i.i.d. fashion, and this subset is attacked over the next T slots by inverting its innovation sequence. We assume that the probability of attacking a sensor i is proportional to $\frac{1}{i^2}$. Since each sensor is susceptible to FDI attack, SAFE is not applicable here due to the lack of availability of any *safe* sensor. Hence, we compare Algorithm 2 (with $\lambda = 0.5$) only with PERFECT. We observe from Figure 3 that Algorithm 2 performs close to PERFECT.

VI. CONCLUSIONS

In this paper, we first proposed a detection scheme for FDI attack on unknown sensor subset. Next, we developed a secure estimation scheme to reduce estimation error under FDI attack on a static or time-varying unknown sensor subset. The algorithms are validated numerically. In future, we would endeavour to extend our work to unknown system dynamics as well as the proof of Conjecture 1.

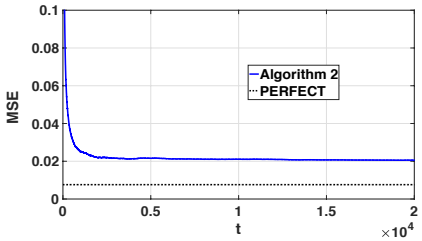

Figure 3. Performance comparison of Algorithm 2 against PERFECT under switching location attack. $N = 10$, $n_0 = 3$, $k = 4$, $q = 3$, $\lambda = 0.5$, $T = 20$.

REFERENCES

[1] Yilin Mo and Bruno Sinopoli. Secure control against replay attacks. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 911–918. IEEE, 2009.

[2] Yilin Mo, Rohan Chabukswar, and Bruno Sinopoli. Detecting integrity attacks on scada systems. *IEEE Transactions on Control Systems Technology*, 22(4):1396–1407, 2014.

[3] Yanpeng Guan and Xiaohua Ge. Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks. *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

[4] Yuan Chen, Soummya Kar, and José MF Moura. Optimal attack strategies subject to detection constraints against cyber-physical systems. *IEEE Transactions on Control of Network Systems*, 2017.

[5] Ziyang Guo, Dawei Shi, Karl Henrik Johansson, and Ling Shi. Optimal linear cyber-attack on remote state estimation. *IEEE Transactions on Control of Network Systems*, 4(1):4–13, 2017.

[6] Yuzhe Li, Ling Shi, and Tongwen Chen. Detection against linear deception attacks on multi-sensor remote state estimation. *IEEE Transactions on Control of Network Systems*, 2017.

[7] Yuan Chen, Soummya Kar, and José MF Moura. Cyber physical attacks with control objectives and detection constraints. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 1125–1130. IEEE, 2016.

[8] Fabio Pasqualetti, Florian Dörfler, and Francesco Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.

[9] Fei Miao, Quanyan Zhu, Miroslav Pajic, and George J Pappas. Coding schemes for securing cyber-physical systems against stealthy data injection attacks. *IEEE Transactions on Control of Network Systems*, 4(1):106–117, 2017.

[10] Miroslav Pajic, Insup Lee, and George J Pappas. Attack-resilient state estimation for noisy dynamical systems. *IEEE Transactions on Control of Network Systems*, 4(1):82–92, 2017.

[11] Shaunak Mishra, Yasser Shoukry, Nikhil Karamchandani, Suhas N Diggavi, and Paulo Tabuada. Secure state estimation against sensor attacks in the presence of noise. *IEEE Transactions on Control of Network Systems*, 4(1):49–59, 2017.

[12] Chensheng Liu, Jing Wu, Chengnian Long, and Yebin Wang. Dynamic state recovery for cyber-physical systems under switching location attacks. *IEEE Transactions on Control of Network Systems*, 4(1):14–22, 2017.

[13] Kebina Manandhar, Xiaojun Cao, Fei Hu, and Yao Liu. Detection of faults and attacks including false data injection attack in smart grid using kalman filter. *IEEE transactions on control of network systems*, 1(4):370–379, 2014.

[14] Gaoqi Liang, Junhua Zhao, Fengji Luo, Steven Weller, and Zhao Yang Dong. A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid*, 2017.

[15] Brian DO Anderson and John B Moore. Optimal filtering. *Englewood Cliffs*, 21:22–95, 1979.

[16] J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[17] Vivek S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Cambridge University Press, 2008.