Detection of data injection attacks in decentralized learning

Reinhard Gentz†, Hoi-To Wai†, Anna Scaglione† and Amir Leshem‡

†School of ECEE, Arizona State Univ., Tempe, AZ, USA. ‡Faculty of Engr., Bar-Ilan Univ., Ramat-Gan, Israel.
Email: {rgentz,htwai,Anna.Scaglione}@asu.edu, leshema@eng.biu.ac.il

***Abstract*—Gossip based optimization and learning are appealing methods that solve big data learning problems sharing computation and network resources when data are distributed. The main advantage these methods offer is that they are fault tolerant. Their flat architecture, however, expands the attack surface in the case of a data injection attack. We analyze the effects of data injection on the asymptotic behavior of the network and draw a parallel with the case of opinion dynamics in a network where zealots inject opinions to mislead a community. We further propose a possible decentralized detection of such attacks and analyze its performance.**

***Index terms*—** data injection attack, attack detection, decentralized learning, randomized gossip protocol

I. Introduction

In wireless sensor networks (WSN) as well as big data learning problems, gossip-based algorithms are an attractive way to distribute computational tasks and share network resources. One main advantage of gossip-based algorithms is that they are fault tolerant as the nodes can be reorganized automatically if one of them fails. Moreover, as illustrated in [1], [2], attacks launched by unauthorized nodes can be easily prevented by augmenting the network with authentication and encryption protocols. However, in the case of an *insider attack*, the flat architecture and the self-organizing nature of the gossip-based algorithms have made the algorithm highly vulnerable even if only one of the nodes is compromised.

This paper focuses on the insider attack scenario. Our aim is to develop tools for detecting and locating the attackers. We avoid the challenge of encryption and authentication and perform our detection strategy solely by using statistical analysis of the transmitted values during the gossip-based algorithm. Specifically, we model the attackers for randomized consensus algorithm as a group of coordinated agents that are trying to steer the consensus state to a value of their choosing, while hiding their nature by appearing to follow the expected exponential convergence rate [3]. Ideally, one should detect a compromised neighboring node, locally at each node, without the help of a central organization. In this paper we present a strategy for a fully decentralized detection and identification schemes for a randomized consensus algorithm, only using already locally present information and therefore do not require any any additional communication overhead.

This material is based upon work supported by NSF CCF-1011811 and CREDC DOE grant DE-OE0000780.

Protection scheme for consensus algorithms have only been considered in a few works, e.g., [4], [5]. For instance, [5] has recently analyzed data injection attacks in a synchronous consensus protocol using a likelihood test. However one should notice that synchronous consensus requires all values to be transmitted in unison, making the practical implementation for WSNs complicated compared to a randomized consensus algorithm.

Reference [4] has proposed two protection schemes for randomized consensus algorithms. Their first scheme is motivated by the fact that the convergence speed is usually slower under the presence of an attacker; thus a data injection attack can be spotted by detecting possible anomalies in the convergence speed. Notice that the 'normal' convergence speed can be estimated only when prior knowledge on the underlying physical model is available; e.g., see [3], [6], [7]. The second scheme in [4] requires all transmissions to be cryptographically signed and confirmed by each neighbor in radio range. This requires the presence of a majority of honest neighbors around the attackers and may be impractical. As mentioned before, we consider a data injection attack model where the attackers can deceive their neighbors by following the 'normal' convergence rate. It is likely that the attack will remain undetected by the scheme proposed in [4]. In contrast, our proposed scheme relies on detecting the statistical anomalies in the presence of an attack. As we shall show later, the proposed scheme is robust to this type of attack.

This paper is organized as follows. Section II describes the basic randomized consensus algorithm and introduces the data injection attack model. Section III describes the proposed protection scheme and analyzes its performance. We conclude with preliminary simulation results in Section IV.

II. Consensus Network Model

Consider a network of sensors that can be described by a simple connected & undirected graph $G = (V, E)$, where $V = [n] = \{1, ..., n\}$ denotes the set of sensor nodes and $E \subseteq V \times V$ denotes the connection between the nodes. We assume that the sensors perform a randomized consensus algorithm many times, over which they can build our detection metric. The n

dimensional random vector $\boldsymbol{x}^k[t]$ vector represents the states at the tth consensus iteration in the kth instance of the algorithm. For each node $i \in [n]$, we assume that its initial state $x_i^k[0]$ over every instance of the algorithm forms an ergodic random sequence:

$$x_i^k[0] = \gamma_i[k] \tag{1}$$

where $\gamma_i[k]$ is a random variable (r.v.) with mean $\bar{\gamma}$ and sub-Gaussian[1] parameter σ_γ^2 and that the consensus algorithm is to compute the network initial states' average:

$$x^{av}[k] := \frac{1}{n}\mathbf{1}^\top \boldsymbol{x}^k[0]. \tag{2}$$

The consensus algorithm employed is as follows.

Randomized consensus protocol:

1) At time $t \in \mathbb{Z}$, a sensor $i \in V$ is selected randomly (uniformly) to wake up.
2) Sensor i selects sensor j from its neighborhood with the probability
$$P_{ij}, \text{ i.e., } j \in \mathcal{N}_i \text{ and } \mathcal{N}_i := \{j : (i,j) \in E\}. \tag{3}$$
3) Sensor i and j update their variables as:
$$x_i^k[t+1] = x_j^k[t+1] = \frac{x_i^k[t] + x_j^k[t]}{2}; \tag{4}$$
the other sensors keep their original variables, i.e., $x_v^k[t+1] = x_v^k[t+1]$ for all $v \neq i, j$.
4) Update $t = t+1$ and repeat from step 1.

Notice that the above protocol can be implemented asynchronously.

Under mild assumptions, the above protocol finds the true average $\overline{\boldsymbol{x}}$ at a geometric rate. Define:

$$\overline{\boldsymbol{W}} = \boldsymbol{I} - \frac{1}{2n}\boldsymbol{\Sigma} + \frac{\boldsymbol{P} + \boldsymbol{P}^\top}{2n}, \tag{5}$$

where $\boldsymbol{\Sigma}$ is diagonal with $\Sigma_{ii} = \sum_{j=1}^n (P_{ij} + P_{ji})$. Notice that $\overline{\boldsymbol{W}}$ is symmetric and doubly stochastic. The matrix $\overline{\boldsymbol{W}}$ governs the average dynamics:

$$\overline{\boldsymbol{x}}^k[t] = \mathbb{E}_{\Omega_t}\{\boldsymbol{x}^k[t]|\boldsymbol{x}^k[0]\} = \overline{\boldsymbol{W}}\overline{\boldsymbol{x}}^k[t-1] = \overline{\boldsymbol{W}}^t\boldsymbol{x}^k[0], \tag{6}$$

where $\mathbb{E}_{\Omega_t}\{\cdot\}$ is the expectation taken with respect to the random nodes chosen at each randomized consensus instance up to time t.

Fact 1 *Given k, for $t \geq 3\log\Delta^{-1}/\log\lambda_2(\overline{\boldsymbol{W}})^{-1}$, the consensus error of the variable at every sensor $i \in V$ converges to a Δ-neighborhood of $x^{av}[k]$ with high probability, i.e.,*

$$\mathbb{P}\Big(|x_i^k[t] - x^{av}[k]| < \Delta \max_{j\in V}|x_j^0[k]|\Big) \geq 1 - \Delta, \tag{7}$$

[1]That is, $\mathbb{E}_k\{\exp(\lambda(\gamma_i[k] - \bar{\gamma}))\} \leq \exp(\sigma_\gamma^2\lambda^2/2)$ for all $\lambda \in \mathbb{R}$.

for all $\Delta \geq 0$, where $\lambda_2(\cdot)$ denotes the second largest eigenvalue.

Fact 1 is proven by taking an infinity norm in [7, Theorem 3]. Notice that the lower bound on t is finite only when $\lambda_2(\overline{\boldsymbol{W}}) < 1$. The latter requirement depends on the design of $\boldsymbol{P}$ and can be satisfied typically if G is a connected graph; see [3] for further discussions on the second largest eigenvalue of $\overline{\boldsymbol{W}}$.

As a corollary of Fact 1, the consensus error decays (with high probability) at a geometric rate:

$$\big\|\boldsymbol{x}^k[t] - x^{av}[k]\mathbf{1}\big\| = \mathcal{O}(\lambda_2^t(\overline{\boldsymbol{W}})), \tag{8}$$

where $\|\cdot\|$ is a norm in the Euclidean space.

A. Data Injection Attack

The data injection attack model is analogous to the *stubborn agent* model studied under the framework of DeGroot opinion dynamics in social learning [8] whose convergence properties were studied in [9]. We assume that the sensor network is compromised by a set of attackers, denoted by $V_s \subseteq V$. For simplicity, we set $V_s = [n_s] = \{1, ..., n_s\}$. The aim of the attackers is to steer the consensus value of the network to a certain target value of their choice $\alpha[k]$, so that asymptotically, the kth instance of the gossip algorithm converges to

$$\lim_{t\to\infty} \boldsymbol{x}^k[t] = \alpha[k]\mathbf{1}, \tag{9}$$

such that $\alpha[k] \neq x^{av}[k]$. The malicious nodes follow a modified update rule that eventually steer the consensus value of the network to $\alpha[k]$.

In particular, in the *randomized consensus protocol*, if a malicious node $j \in V_s$ is selected at time t, the node's state will be generated as

$$x_j^k[t] = \alpha[k] + m_j^k[t],\ x_j^k[t+1] = \alpha[k] + m_j^k[t+1], \tag{10}$$

in lieu of (4), where $m_j^k[t]$ is an independent r.v. with zero mean and sub-Gaussian parameter $(\hat{\lambda}^{t_j}\sigma_M)^2$, where $\hat{\lambda} < 1$ is an estimate of the convergence rate *without* the malicious agents and t_j is the total number of interactions the malicious agent j had prior to time t. The noise process $m_j^k[t]$ is designed to confuse the network peers about the malicious agent intentions. This model is based on the following considerations:

Remark 1 *To deceive their peers and appear as making progress towards the computation of $x^{av}[k]$, the malicious agents keep changing their states randomly by mixing the target value $\alpha[k]$ with the noise process $m_j^k[t]$. Moreover, to ensure that the trend of the value they exchange to appear as* normal*, the variance of $m_j^k[t]$ decays exponentially with t.*

Remark 2 *We assume that the attack is coordinated such that all the malicious agents bias their state with the same value*

$\alpha[k]$. *If the malicious nodes do not coordinate the attack and choose an arbitrary target value* $\alpha_j[k]$, *the network will not reach consensus almost surely [9], [10]. This coordinated kind of attack is clearly harder to detect.*

Let us study the first order statistics of the nodes' states under data injection attack. We define

$$\boldsymbol{x}^t[k] = \begin{pmatrix} \boldsymbol{s}^k[t] \\ \boldsymbol{r}^k[t] \end{pmatrix}, \tag{11}$$

where $\boldsymbol{s}^k[t]$ and $\boldsymbol{r}^k[t]$ corresponds to an n_s-dimensional and $(n-n_s)$-dimensional random vector, respectively.

What changes in the *randomized consensus protocol* is Eq. (4) whenever (at least) one of the sensors i and j that are performing the update is a malicious node. As a consequence of the modified update rule (10), we have

$$\bar{\boldsymbol{s}}^k[t] = \mathbb{E}_{\Omega_t}\{\boldsymbol{s}^k[t]|\alpha[k]\} = \bar{\boldsymbol{s}}^k[t-1] = \alpha[k]\mathbf{1}. \tag{12}$$

Consequently, the average matrix $\overline{\boldsymbol{W}}$ that governs the average dynamics admits the following partition:

$$\overline{\boldsymbol{W}} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \overline{\boldsymbol{B}} & \overline{\boldsymbol{D}} \end{pmatrix}, \tag{13}$$

where we assume that $\|\overline{\boldsymbol{D}}\|_2 < 1$. Taking the average matrix to the power t yields

$$\overline{\boldsymbol{W}}^t = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \sum_{s=0}^{t} \overline{\boldsymbol{D}}^{t-s}\overline{\boldsymbol{B}} & \overline{\boldsymbol{D}}^t \end{pmatrix} \tag{14}$$

As such, we have

$$\begin{aligned} \bar{\boldsymbol{r}}^k[t] &= \mathbb{E}_{\Omega_t}\{\boldsymbol{r}^k[t]|\boldsymbol{r}^k[0], \alpha[k]\} \\ &= \alpha[k]\sum_{s=0}^{t} \overline{\boldsymbol{D}}^{t-s}\overline{\boldsymbol{B}}\mathbf{1} + \boldsymbol{D}^t\boldsymbol{r}^k[0]. \end{aligned} \tag{15}$$

Notice that $\boldsymbol{r}^k[0] = \boldsymbol{\gamma}[k]$, where $\boldsymbol{\gamma}[k] := (\gamma_{n_s+1}[k], ..., \gamma_n[k])$. Before concluding this section, we notice that $\overline{\boldsymbol{W}}$ is not doubly stochastic as the network-wide average is evolving with t. Moreover, as $\|\overline{\boldsymbol{D}}\|_2 < 1$ and $\sum_{s=0}^{\infty} \overline{\boldsymbol{D}}^{t-s}\overline{\boldsymbol{B}}\mathbf{1} = \mathbf{1}$, the latter component in (15) vanishes exponentially fast while the former component becomes dominant as $t \to \infty$. As such, it holds true that $\lim_{t\to\infty} \overline{\boldsymbol{x}}^k[t] = \alpha[k]\mathbf{1}$. We remark that the previous mean convergence can be strengthened to almost surely convergence; see [7] for a proof.

III. Detection of Data Injection Attack

In this section, we propose a strategy for detecting and locating the data injection attack. For obvious reasons, we analyze the detection rule at a non-malicious agent i, i.e., $i \notin V_s$. The detection is done in two stages, first the agent i decides:

$$\begin{aligned} \mathcal{H}_0^i &: \text{ there is no malicious agent, i.e., } V_s \cap \mathcal{N}_i = \emptyset. \\ \mathcal{H}_1^i &: \text{ there is a malicious agent, i.e., } V_s \cap \mathcal{N}_i \neq \emptyset. \end{aligned} \tag{16}$$

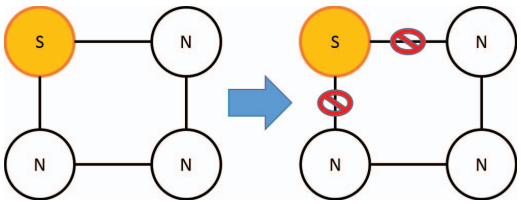

Fig. 1: Each normal node n will independently, upon detection and localization, cut all communication with the malicious agent s and therefore isolate the malicious agent from the network.

If $\mathcal{H}_0^i$ is true, then no action will be taken. Otherwise, in the second stage the agent i decides:

$$\begin{aligned} \mathcal{H}_0^{ij} &: \text{ agent } j \text{ is not malicious, i.e., } j \notin V_s. \\ \mathcal{H}_1^{ij} &: \text{ agent } j \text{ is malicious, i.e., } j \in V_s. \end{aligned} \tag{17}$$

for all $j \in \mathcal{N}_i$. If $\mathcal{H}_1^{ij}$ is true, then agent i will sever edge (i,j) for all future communication. As the rule is checked on every non-malicious agent, this will effectively isolate the attacker and prevent any future impact from him/her; see Figure 1.

In the sequel, we shall assume that $\alpha[k]$, $\gamma_i[k]$ are independent, zero mean r.v.s with sub-Gaussian parameters σ_α^2, σ_γ^2, respectively. The first test (16) will be referred to as the *detection task*, and the second test (17) as the *localization task*.

Our strategy focuses on detecting the anomalies caused by the malicious agent through evaluating the (average) *temporal difference* of the values held by an agent. Our intuition stems from that the (expected) initial value of a malicious agent $s \in V_s$ is different from a normal agent $j \in V_r$, i.e.,

$$\mathbb{E}_k\{x_s^k[0]\} = \bar{\alpha} \neq \bar{\gamma} = \mathbb{E}_k\{x_j^k[0]\}. \tag{18}$$

Meanwhile, the network will be mislead by the malicious agent as $t \to \infty$:

$$\mathbb{E}_k\{x_s^k[\infty]\} = \bar{\alpha} = \mathbb{E}_k\{x_j^k[\infty]\}. \tag{19}$$

This implies that the quantity $|x_i^k[\infty] - x_i^k[0]|$ will be close to zero if $i \in V_s$ or large if otherwise, indicating a possible anomaly. Let us consider the following statistics to be evaluated at the non-malicious agent i:

$$\xi_{ij} := \frac{1}{K}\sum_{k=0}^{K}(x_j^k[T_{ij}] - x_j^k[0]), \tag{20}$$

for all $j \in \mathcal{N}_i$ and some sufficiently large T_{ij}. [2]

Based on ξ_{ij}, we propose the following detector for the detection task (16):

$$\sum_{m\in\mathcal{N}_i} |\xi_{im} - \bar{\xi}_i| \underset{\mathcal{H}_1^i}{\overset{\mathcal{H}_0^i}{\lessgtr}} \delta \tag{21}$$

where $\bar{\xi}_i := (1/|\mathcal{N}_i|)\sum_{m\in\mathcal{N}_i}\xi_{im}$. Intuitively, the detector (21) finds if there is an outlier in $\mathcal{N}_i$ for the set of statistics

[2]Note that $x_j^k[0]$ is the first value seen by the evaluating node i and $x_j^k[T_{ij}]$ the last value seen by node i

$\{\xi_{im}\}_{m\in\mathcal{N}_i}$. The following results provide insights to the performance of the first detector.

Proposition 1 *Let $T_{ij}\to\infty$, we have*

$$P(\hat{\mathcal{H}}^i=\mathcal{H}_1^i|\mathcal{H}_0^i)\le 2|\mathcal{N}_i|\cdot\exp\Big(-\frac{K}{|\mathcal{N}_i|}\frac{\delta^2}{2\sigma_\gamma^2(|\mathcal{N}_i|-1)}\Big), \quad (22)$$

$$P(\hat{\mathcal{H}}^i=\mathcal{H}_0^i|\mathcal{H}_1^i)\le \exp\Big(-K\frac{(\max\{0,-\delta+|\mu_i|\})^2}{2\sigma_i^2}\Big), \quad (23)$$

with $\mu_i=\frac{|V_r\cap\mathcal{N}_i|}{|\mathcal{N}_i|}(\bar{\alpha}-\bar{\gamma})$ *and* $\sigma_i^2=\Big(\frac{|\mathcal{N}_i|^2-2|\mathcal{N}_i|+|V_s\cap\mathcal{N}_i|}{|\mathcal{N}_i|^2}\Big)\sigma_m^2+\frac{|V_r\cap\mathcal{N}_i|^2}{|\mathcal{N}_i|^2}\sigma_\alpha^2+\frac{|V_r\cap\mathcal{N}_i|}{|\mathcal{N}_i|^2}\sigma_\gamma^2$.

Proof of Proposition 1. Observe the following chain for the false alarm rate:

$$\begin{aligned}&P\left(\hat{\mathcal{H}}=\mathcal{H}_1^i\mid\mathcal{H}_0^i\right)=P\left(\sum_{m\in\mathcal{N}_i}|\xi_{im}-\bar{\xi}_i|\ge\delta\mid\mathcal{H}_0^i\right)\\&\le|\mathcal{N}_i|P\left(|\xi_{im}-\bar{\xi}_i|\ge\frac{\delta}{|\mathcal{N}_i|}\mid\mathcal{H}_0^i\right),\ \text{for some } m\in\mathcal{N}_i,\end{aligned} \quad (24)$$

where we have applied the union bound in the last inequality. We have

$$\xi_{im}-\bar{\xi}_i=\frac{1}{K}\sum_{k=1}^K\Big((-1+\frac{1}{|\mathcal{N}_i|})\gamma_m[k]+\sum_{j\in\mathcal{N}_i\setminus\{m\}}\frac{1}{|\mathcal{N}_i|}\gamma_j[k]\Big).$$

The above quantity is a zero mean r.v. with sub-Gaussian parameter $\sigma_\gamma^2(|\mathcal{N}_i|-1)/(K|\mathcal{N}_i|)$. Applying the Hoefding's inequality [11] to the last term of (24) yields the desired inequality.

For the missed detection rate, we have

$$\begin{aligned}&P\left(\hat{\mathcal{H}}=\mathcal{H}_0^i\mid\mathcal{H}_1^i\right)=P\left(\sum_{m\in\mathcal{N}_i}|\xi_{im}-\bar{\xi}_i|\le\delta\mid\mathcal{H}_1^i\right)\\&\le P\left(|\xi_{im}-\bar{\xi}_i|\le\delta\mid\mathcal{H}_1^i\right)\ \forall\ m\in\mathcal{N}_i.\end{aligned} \quad (25)$$

Observe that

$$\xi_{ij}=\begin{cases}(1/K)\sum_{k=1}^K(m_j^k[0]), & j\in V_s,\\(1/K)\sum_{k=1}^K(\alpha[k]-\gamma_j[k]), & j\notin V_s.\end{cases}$$

and

$$\bar{\xi}_i=\frac{1}{K|\mathcal{N}_i|}\sum_{k=1}^K\left(\sum_{j\in V_s\cap\mathcal{N}_i}m_j^k[0]+\sum_{j\in V_r\cap\mathcal{N}_i}(\alpha[k]-\gamma_j[k])\right)$$

We observe that $\xi_{im}-\bar{\xi}_i$ is an r.v. with mean μ_i and sub-Gaussian parameter σ_i^2/K for $m\in V_s$. Let us write $\xi_{im}-\bar{\xi}_i=\tilde{\xi}_{im}+\mu_i$ and upper bound the last term in (25) as:

$$P\left(|\xi_{im}-\bar{\xi}_i|\le\delta\mid\mathcal{H}_1^i\right)\le P\left(\tilde{\xi}_{im}\ge-\delta+|\mu_i|\mid\mathcal{H}_1^i\right) \quad (26)$$

Consequently, the desired inequality can be obtained by applying Hoefding's inequality. □

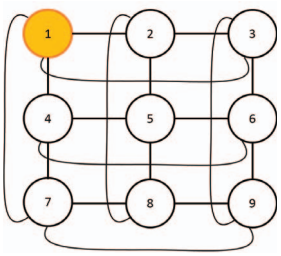

Fig. 2: Simulated topology. Node 1 is a malicious agent, while the rest of the nodes are normal nodes.

The analysis above shows the impact of the variance on the detection (21) performance. Specifically, the false alarm rate (22) depends solely on σ_γ^2, while the missed detection rate (23) depends on the other parameters as well.

For the localization task (17), we have:

$$|\xi_{ij}|\underset{\mathcal{H}_0^{ij}}{\overset{\mathcal{H}_1^{ij}}{\lessgtr}}\epsilon \quad (27)$$

for all $j\in\mathcal{N}_i$. The performance can be summarized as:

Proposition 2 *Let $T_{ij}\to\infty$, we have*

$$P(\hat{\mathcal{H}}^{ij}=\mathcal{H}_1^{ij}|\mathcal{H}_0^{ij})\le\exp\Big(-K\frac{(\max\{0,-\epsilon+|\bar{\alpha}-\bar{\gamma}|\})^2}{2(\sigma_\alpha^2+\sigma_\gamma^2)}\Big), \quad (28)$$

$$P(\hat{\mathcal{H}}^{ij}=\mathcal{H}_0^{ij}|\mathcal{H}_1^{ij})\le 2\cdot\exp\Big(-K\frac{\epsilon^2}{2\sigma_M^2}\Big). \quad (29)$$

Proof of Proposition 2. Under $\mathcal{H}_1^{ij}$, we have $\xi_{ij}=(1/K)\sum_{k=1}^K m_j^k[0]$, where $m_j^k[0]$ are zero mean, independent r.v.s with sub-Gaussian parameter σ_M^2. Under $\mathcal{H}_0^{ij}$, we have

$$\xi_{ij}=\frac{1}{K}\sum_{k=1}^K\big(\alpha[k]-\gamma_j[k]\big), \quad (30)$$

note that the terms inside the summation have mean $\bar{\alpha}-\bar{\gamma}$ and are independent with sub-Gaussian parameter $\sigma_\alpha^2+\sigma_\gamma^2$. Similar to Proposition 1. the desired inequalities can be obtained by applying Hoeffding's inequality. □

Remark 3 *The error rates derived in Proposition 1 & Proposition 2 are loose upper bounds in general. When K is sufficiently large, one can apply an Gaussian approximation to obtain tighter approximations to the error rates.*

Remark 4 *The detectors and the analytical results in this section can be applied to detect the similarly modeled attackers for a general randomized consensus setting, e.g., in [12].*

IV. Numerical Results

We consider a simple example with $n=9$ agents. The graph G follows the Manhattan topology as seen in Figure 2. There is only one malicious agent in the network such that $V_s=\{1\}$. The randomized gossip algorithm is terminated

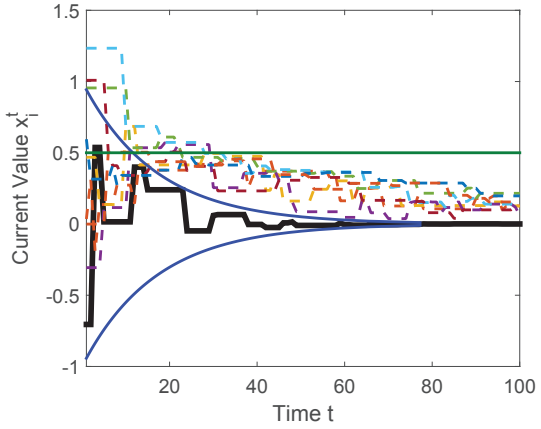

Fig. 3: Example of a single random consensus run. The malicious agent (black) is forcing all normal nodes (dashed) to its target value $\alpha = 0$, while the true $x^{av} = 0.5$ (green). Furthermore the noise of the malicious agent is given by the true λ_2 of the non-perturbed network (8) (blue).

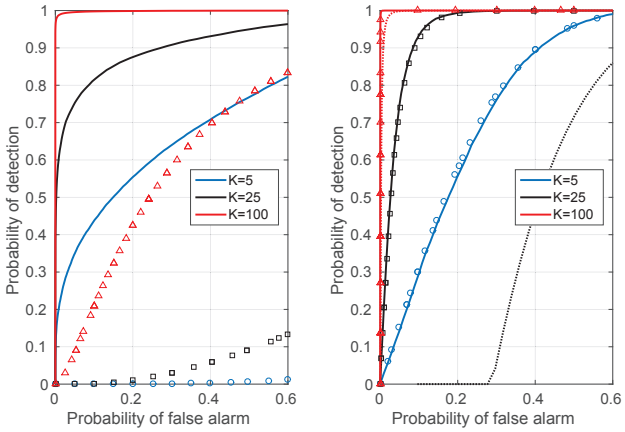

Fig. 4: Detector performance: (Left) ROCs for detection of malicious agent(s). For the considered Ks, the theoretical bounds (22) & (23) are trivial and therefore omitted. (Right) ROCs for localization of malicious agent(s). Dotted lines show the theoretical bounds in (28) & (29). Markers show the bounds obtained by applying Gaussian approximation (cf. Remark 3).

with $T = 500$. We have $\alpha[k] \sim \mathcal{N}(0,1)$, $\gamma_i[k] \sim \mathcal{U}[-0.5, 1.5]$, $m_j^k[t] \sim \mathcal{U}[-\hat{\lambda}^t, \hat{\lambda}^t]$ and the r.v.s are independent. Furthermore each link, P_{ij} has the same probability of being selected in the random consensus algorithm. We average our results with a Monte Carlo simulation of 10^5 trials. An example of a single consensus run can be seen in Figure 3.

Each normal node V_r is trying to detect data injection attack by monitoring the temporal difference in $x_j^k[t]$. We plot receiver operating characteristic (ROC) curves averaged over all nodes in Figure 4. From the figure, we see that the performance for both detection and localization improves with K. Moreover, under the same K, the performance of the detection task is worse than the localization task. This corroborates with our observations in Proposition 1 and Proposition 2. We can further see that the theoretical performance (shown dotted in Figure 4) for detection (left) (22) & (23) does not show close results as the found bounds are loose for a small K. On the other hand, the theoretical performance (with the Gaussian approximation) for localization (right) (28) & (29) matches our simulated curves.

We numerically found that, for the same case, increasing the number of nodes attacking from 1 up to 4 slightly improved the performance and the reverse trend is true up until there are 7 attackers and 2 neighboring nodes that are not malicious. Further analysis will be done in future work.

To conclude, we have proposed a novel strategy to detect and identify malicious agents in a randomized consensus algorithm. The algorithm can be performed at each individual node in a completely decentralized manner. The performance bounds of the algorithm are analyzed, and the simulation results confirmed our findings. As part of the ongoing research, we shall explore and analyze different metrics for the detection and localization of malicious agents. We also plan to apply the malicious agent detection strategy to the social system identification problem presented in our previous work [13].

REFERENCES

[1] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: Security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[2] S. Zhu, S. Setia, and S. Jajodia, "Leap: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc CCS '03*, 2003, pp. 62–72.

[3] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov 2010.

[4] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. Hou, "Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks," in *Proc INFOCOM 2012*, March 2012, pp. 900–908.

[5] B. Kailkhura, S. Brahma, and P. K. Varshney, "Consensus based detection in the presence of data falsification attacks," *arXiv preprint arXiv:1504.03413*, 2015.

[6] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. on Signal Process.*, vol. 57, no. 7, pp. 2748–2761, 2009.

[7] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[8] M. DeGroot, "Reaching a consensus," in *Journal of American Statistcal Association*, vol. 69, 1974, pp. 118–121.

[9] M. E. Yildiz and A. Scaglione, "Computing along routes via gossiping," *IEEE Trans. on Signal Process.*, vol. 58, no. 6, pp. 3313–3327, 2010.

[10] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, "Robust Average Consensus using Total Variation Gossip Algorithm," in *VALUETOOLS*, 2012, pp. 99–106.

[11] P. Massart, *Concentration Inequalities and Model Selection*. Springer, 2003.

[12] J. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., Boston, MA, 1984.

[13] H.-T. Wai, A. Scaglione, and A. Leshem, "The Social System Identification Problem," accepted by *IEEE CDC 2015*. [Online]. Available: http://arxiv.org/abs/1503.07288