2018 ICSEE International Conference on the Science of Electrical Engineering

Detection of Data Injection Attacks on Decentralized Statistical Estimation

Or Shalom
Faculty of Engineering
Bar-Ilan University
Ramat Gan, Israel

Amir Leshem
Faculty of Engineering
Bar-Ilan University
Ramat Gan, Israel

Anna Scaglione
School of ECEE
Arizona State University
Tempe, AZ, USA

Angelia Nedić
School of ECEE
Arizona State University
Tempe, AZ, USA

***Abstract*—This paper describes a distributed statistical estimation problem, corresponding to a network of agents. The network may be vulnerable to data injection attacks, in which the attackers' main goal is to steer the network's final state to a state of their choice. We show that the detection metric of the straightforward attack scheme proposed by *Wu et. al* in [1], is vulnerable to a more sophisticated attack. To overcome this attack we propose a novel metric that can be computed locally by each agent to detect the presence of an attacker in the network, as well as a metric that localizes the attackers in the network. We conclude the paper with simulations supporting our findings.**

***Index Terms*—Distributed projected gradient, Decentralized optimization, Data injection attacks, Convex optimization, Maximum likelihood**

I. Introduction

DECENTRALIZED multi-agent optimization is an important problem in distributed computation. These algorithms rely on local computations as well as in-neighborhood communication to reach their common goal of minimizing a common cost function or converging to a stable point. As these networks gain in popularity [2–10], it has become apparent that they are sensitive to false data injection which can steer the network's final state, see [1], [11–21] for examples. The structure of an independently - self updating network, which has been the main advantage of these methods, can turn into a vulnerability by allowing an attacker which controls a single node to have a global impact. This type of attack cannot be detected using cryptographic techniques, since they use a legitimate node in the network.

This paper focuses on the problem of detecting attacks on distributed statistical estimation, using maximum likelihood estimators when implemented using the distributed projected gradient (DPG) algorithm [2]. We begin with a novel data injection attack scheme, and its effects on decentralized optimization algorithms, and primarly the DPG. We propose a new, more sophisticated attack scheme that is invisible to all previous detection methods. This attack scheme is shown to be always successful on synchronous-communication networks, even when the network is dynamically changing over time. We then propose two new metrics, computed locally by each agent, to detect the presence of an attacker in the network and localize it. The proposed detector is shown analytically and numerically to be successful.

Notations: We use boldfaced letters to denote vectors and boldfaced uppercase letters to denote matrices. For a vector $\boldsymbol{\theta}$, $[\boldsymbol{\theta}]_i$ denote its i-th element, similarly, for a matrix $\boldsymbol{A}$, $\boldsymbol{A}_{ij}$ denotes its (i,j)-th element. We mark the vector of ones as $\mathbf{1}$, the $(m \times m)$ identity matrix as $\boldsymbol{I}_m$ and the j-th unity vector as $\boldsymbol{e}_j$ to be all zeros except for the j-th element.

II. Problem Formulation

We consider a grid of sensors measuring independent random processes that depend on a joint parameter. In order to extract this parameter, the sensors solve a maximum likelihood problem

$$\arg\max_{\boldsymbol{\theta}} \sum_i \log f_{\boldsymbol{x}_i}(\boldsymbol{x}_i|\boldsymbol{\theta}) \tag{1}$$

where $f_{\boldsymbol{x}_i}$ is the i-th agent's density function and $\boldsymbol{\theta}$ is an unknown parameter vector. We assume that the measured noise is i.i.d. between sensors. We consider a distributed setup where agents do not share their private information $\boldsymbol{x}_i$.

A. Preliminaries

We consider an undirected time varying graph $G(t) = (V, E(t))$ defining a network of N agents, where $V = \{1, ..., N\}$ is a set of N nodes (agents) and $E(t) \subseteq V \times V$ denotes the connections between the nodes for some time $t \in \mathbb{N}$. For each node i, we define $\mathcal{N}_i \subset V$ as the neighborhood set of agent i, as $\mathcal{N}_i := \{j : (j,i) \in E\}$, note that $E = \cup_{t=1}^{\infty} E(t)$. We mark the agents' states for some time $t \geq 0$ in the k-th instance as $\boldsymbol{\Theta}^k(t) = [\boldsymbol{\theta}_1^k(t), ..., \boldsymbol{\theta}_N^k(t)]^T \in \mathbb{R}^{N \times P}$. We assume that the agents perform a randomized optimization algorithm for K instances which are used as a detection metric.

B. Distributed stochastic maximum likelihood estimation

The network graph consists of N agents sharing the common goal of minimizing a joint likelihood function in a distributed manner; i.e., we need to solve the following optimization problem:

$$\min_{\boldsymbol{\theta}} h(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} h_i(\boldsymbol{\theta}), \quad \text{s.t. } \boldsymbol{\theta} \in \mathcal{C} \tag{2}$$

This work is supported by the NSF CCF–BSF 1714672.

where $\mathcal{C} \subseteq \mathbb{R}^P$ is a closed, convex, compact set and $h_i : \mathbb{R}^P \to \mathbb{R}$, $h_i(\boldsymbol{\theta}) = -\log f_{\boldsymbol{x}_i}(\boldsymbol{x}_i|\boldsymbol{\theta})$ is a private differentiable[1] function over $\mathcal{C}$, known to the i-th agent alone. In our problem, we assume that $\boldsymbol{x}_i$ are i.i.d. given any value of $\boldsymbol{\theta}$; i.e., the parametric distribution $f_{\boldsymbol{x}}(\boldsymbol{x}|\boldsymbol{\theta})$ is the same function, and is known to all nodes. However the specific realization $f_{\boldsymbol{x}_i}(\boldsymbol{x}_i|\boldsymbol{\theta})$ is private since each node has its own data. We mark the optimal solution of the optimization problem as $h^* = h(\boldsymbol{\theta}^*)$, where $\boldsymbol{\theta}^* \in interior(\mathcal{C})$ is the optimal parameter. In this paper we assume that all $h_i(\boldsymbol{\theta})$ are convex. The maximum likelihood estimation can be solved distributedly using a stochastic distributed projected gradient algorithm. Let $G(t)$ be the graph associated with a weighted adjacency matrix $\boldsymbol{W}(t) \in \mathbb{R}^{N \times N}$, where $\boldsymbol{W}(t)$ satisfies:

Assumption 1: $\boldsymbol{W}(t)$ fullfils the next terms for $t \geq 0$:

- $\boldsymbol{W}(t)$ is a symmetric, nonnegative matrix.
- If $(i,j) \in E(t)$ then $\boldsymbol{W}_{i,j}(t) \geq \xi$ for some $\xi \in (0,1)$.
- If $(i,j) \notin E(t)$ then $\boldsymbol{W}_{i,j}(t) = 0$.
- $\boldsymbol{W}(t)$ is doubly stochastic.

Assumption 2: There exists $B < \infty$ such that the graph $(V, \cup_{l=1}^{B} E(t+l))$ is connected.

The distributed projected gradient (DPG) method [2] solves the optimization problem shown in (2) by performing the recursion:

$$\begin{aligned} \boldsymbol{\theta}_i^k\left(t+1\right) &= \bar{\boldsymbol{\theta}}_i^k\left(t\right) - \eta\left(t\right)\nabla h_i\left(\bar{\boldsymbol{\theta}}_i^k\left(t\right)\right) \\ \bar{\boldsymbol{\theta}}_i^k\left(t\right) &= \sum_{j=1}^{n} \boldsymbol{W}_{ij}\left(t\right)\boldsymbol{\theta}_j^k\left(t\right), \quad \forall i \in V, t \geq 0, \end{aligned} \tag{3}$$

where $\eta(t)$ satisifies:

Assumption 3: $\eta\left(t\right)$ is a time-varying step size satisfying $\sum_{t=1}^{\infty} \eta\left(t\right) = \infty$ and $\sum_{t=1}^{\infty} \eta^2\left(t\right) < \infty$.

Proposition 1: Under assumptions 1-3, for a compact space, the joint objective function asymptotically reaches a minimum, as seen in [3], [11].

$$\lim_{t\to\infty} h(\boldsymbol{\theta}(t)) = h^* \tag{4}$$

Our goal is to detect malicious nodes in the network that attempt to destroy the distributed computation by injecting false data.

III. Data Injection Attacks

Consider a distributed maximum likelihood estimation, where some nodes are malicious and inject false data into the network. We divide the set of nodes, V, into two subsets: $R \subset V$ is the set of reliable agents and $A := V \backslash R$, $A \neq \emptyset$ is the set of attackers. Let $n_a = |A|$ be the number of attacking nodes. The attackers' goal is to steer the network's final state $\lim_{t\to\infty} \boldsymbol{\theta}(t)$ to a target state of their choice, while remaining transparent to the network. To do so the attackers follow a deceiving update rule of their choice while the trustworthy agents follow the DPG update rule as shown in (3). A previous work suggested a straightforward attack scheme, as can be seen in [1]. Unfortunately, the attack scheme can be changed to be even smarter and thus evade this detection method. In this section we present a novel improved attack method, and then propose a detection scheme that exploits the statistical nature of the problem to detect this more advanced attack.

A. Existing attacks

As stated above, the attacker's goal is to steer the network's final state $\lim_{t\to\infty} \boldsymbol{\theta}(t)$ to a target state of its choice. To do so the attacker's update rule is:

$$\boldsymbol{\theta}_j(t) = \boldsymbol{\alpha_0} + \boldsymbol{z_j}(t), \quad \forall j \in A, \tag{5}$$

while the trustworthy agents follow the DPG update rule as shown in (3). $\boldsymbol{\alpha_0}$ is the attacker's desired final state and $\boldsymbol{z_j}\left(t\right)$ is a zero mean and $\sigma_z^2(t)\boldsymbol{I}_P$ variance random noise, vanishing a.s. over time and satisfying the expected convergence rate of the graph for all $j \in A$. The detection method proposed in [1] relies on the difference between the final and the initial state of the agents. For an attacker, the mean value of the difference between the final state and the initial state equals zero, whereas for a trustworthy agent, the mean value of the difference between the final state and the initial state is not zero.

B. New Attack Scheme

The new attack scheme proposed here is a mixture of two update rules:

- The trustworthy agents' DPG update rule.
- The straightforward attacker's update rule [1].

To combine both update rules we generate a new time-varying proportion coefficient marked as $g(t)$.

Assumption 4: The new proportion coefficient $g(t)$ fulfils the following conditions:

- For all $t \geq 0$, $0 \leq g(t) \leq 1$.
- $g(t)$ decreases over time, i.e. $g(t+1) < g(t)$.
- $g(0) = 1$, $\lim_{t\to\infty} g(t) = 0$.

The new proposed attack scheme is

$$\begin{aligned} \boldsymbol{\theta}_j(t+1) = \ & g(t) \times DPG(\boldsymbol{\theta}_j(t)) \\ & + (1-g(t)) \times (\boldsymbol{\alpha_0} + \boldsymbol{z_j}(t+1)), \quad \forall j \in A \end{aligned} \tag{6}$$

where $DPG(\boldsymbol{\theta}_j(t))$ refers to (3), $\boldsymbol{\alpha_0}$ is the attacker's desired final state and $\boldsymbol{z_j}\left(t\right)$ is a zero mean and $\sigma_z^2(t)\boldsymbol{I}_P$ variance random noise, vanishing a.s. over time and satisfying the expected convergence rate of the graph for all $j \in A$. Table 1 depicts the new attack scheme.

The result of implementing the new attack scheme on the network forces the initial state of the attackers' nodes to be similar to that of the trustworthy agents. Therefore, the detection scheme in [1] fails. The major change is that the network reaches convergence to some unstable point we mark as $\bar{\boldsymbol{\alpha}}$. Over time, while the network maintains convergence to $\bar{\boldsymbol{\alpha}}$, $\bar{\boldsymbol{\alpha}}$ drifts towards $\boldsymbol{\alpha_0}$, the chosen state of the attacker as can be seen in Figure 1.

[1] In the case where the objective function is non-differentiable, each gradient reference should be considered a sub-gradient.

Table 1: New proposed attack scheme

```
Input: no. of time iterations $T$, no. of instances $K$.
for $k = 1 : K$ do
    Initialize $\boldsymbol{\theta}^k(0)$ with sensors data.
    for $t = 1 : T$ do
        for $i = 1 : N$ do
            if $i \in A$ then
                Node $i$ updates according to the new
                  attack scheme, as seen in (6).
            else
                Node $i$ updates according to the DPG
                  method, as seen in (3).
            end
        end
    end
end
```

Proposition 2. Under the proposed attack scheme in (6), the network converges to the attacker's desired state $\boldsymbol{\alpha_0}$.

$$\lim_{t \to \infty} \|\boldsymbol{\theta}_i(t) - \boldsymbol{\alpha_0}\| = 0, \quad \forall i \in V, \tag{7}$$

Proof in appendix VII-A.

IV. Detecting the mixed attack

In this section we propose a new detection scheme that can detect the presence of an attacker. Once we find the attackers we can ignore their data and have a trustworthy network solving (2), reaching an optimal point.

In order to detect and localize the attackers we run the recursive DPG algorithm as seen in (3) for K instances, where the attackers are following the proposed attack scheme shown in (6). The algorithm runs for some time index marked as T_∞ that is sufficient for the algorithm to converge.

Detection task. Denote the two hypotheses:

$\mathcal{H}^0$ – There are no attackers in the network; i.e., $A \cap V = \emptyset$.

$\mathcal{H}^1$ – There exists an attacker in the network; i.e., $A \cap V \neq \emptyset$.

We propose a new metric, computed upon convergence of the algorithm (of each instance k), to detect an attacker in the network, marked as d_i, $\forall i \in V$.

$$d_i = \left\| \frac{1}{K} \sum_{k=1}^{K} \nabla h_i \left(\boldsymbol{\theta}_i^k \left(T_\infty \right) \right) \right\| \underset{\mathcal{H}^1}{\overset{\mathcal{H}^0}{\lessgtr}} \delta_d \tag{8}$$

where δ_d is a predefined threshold[2].

Note: The proposed detection metric is purely local, it does not require agents to keep track of neighbors activities, as opposed to previous works, including [1].

Theorem 1. For the detection metric, d_i, as presented above, $E[d_i] \neq 0$ only if an attacker is present in the network. Proof in appendix VII-B.

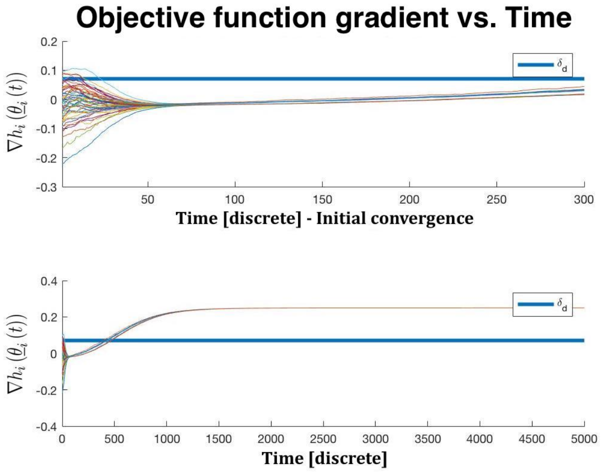

Fig. 1. An example of the proposed detection metric for the new attack scheme, implemented in a network as described in section II-A. The top sub-plot shows the initial convergence of the innocent DPG algorithm to some arbitrary point $\nabla h_i(\bar{\boldsymbol{\alpha}}), \forall i \in V$, whereas the bottom sub-plot shows how the arbitrary point $\nabla h_i(\bar{\boldsymbol{\alpha}})$ drifts to $\nabla h_i(\boldsymbol{\alpha_0}) \neq 0$ over time, exceeding the given threshold.

V. Simulations

This section presents the simulations proving the results shown in the figures below. We begin by generating the edge matrix $\boldsymbol{E}(t)$ representing the network's connections for some time t. The network generated is an "Erdos–Renyi" network consisting of N agents ($N = 50, 100, 500$) with some random probability $0 \leq p \leq 1$. Using the edge matrix we generate the adjacency matrix $\boldsymbol{W}(t)$ as $\boldsymbol{W}(t) = \boldsymbol{I} - \frac{1}{2N}\boldsymbol{S} + \frac{1}{2N}(\boldsymbol{P} + \boldsymbol{P}^T)$, where $\boldsymbol{P}$ is a random $N \times N$ matrix and $\boldsymbol{S}$ is a diagonal matrix consisting of the column sum of $(\boldsymbol{P} + \boldsymbol{P}^T)$. We run the algorithm many times, each time for K instances and up to final time index $T_\infty = 10^5$ (or less if convergence is achieved earlier), representing $t \to \infty$. In each trial, the existence of an attacker is random and unknown.

A. *Example 1:* Estimating Logistic Distribution mean:

We assume that each agent holds a single measurement, (in this example $P = 1$) consisting of the desired signal with zero mean noise. Our goal is to extract the desired noise by eliminating the noise from the given measurement in a distributed manner. To simulate the problem we initialize the agents' state with values generated from a logistic distribution with parameters (μ, σ). The agents solve a ML problem with the following private objective function:

$$h_i(\theta) = 2 \log \left(2 \cosh \left(\frac{x_i - \theta}{2\sigma} \right) \right) + C \tag{9}$$

for all $i \in V$, where C is a constant number known to all agents, x_i is the measured signal for some agent i and $\theta = \mu$, the desired variable. We repeated the simulations for 5×10^5 iterations. The results are depicted in Figure 2. It shows that the success of the attacker's detection depends on the strength of the attack (the distance from the real result), as well as on

[2] Note that h_i is the same objective function for each instance k, for some agent $i \in V$.

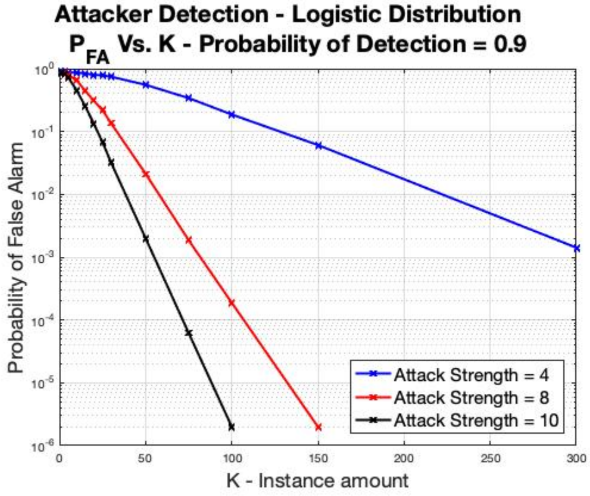

Fig. 2. Attacker Detection performance analysis - Probability of false alarm vs. K for $P_D = 0.9$. Attacker's desired state pulls the network 4, 8 and 10 units away from the real final state.

the number of instances for which the algorithm is run. The stronger the attack, the greater the detection margin as the mean value of the objective function gradient at T_∞ diverges from zero.

B. Example 2: **Linear model**

Consider a simple linear model where each agent receives the following data:

$$\boldsymbol{x}_i = \boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{w}_i \tag{10}$$

where $\boldsymbol{w}_i$ is a Gaussian distributed additive noise received by each agent, $\boldsymbol{w}_i \sim N(\boldsymbol{0}, \boldsymbol{C})$, where $\boldsymbol{C}$ is a known covariance matrix and $\boldsymbol{H}$ is a known matrix. In this example $P = 3$. We simulated two scenarios: without attackers and then with attackers. In each simulation the network was the same as in the previous example and the matrix $\boldsymbol{H}$ was selected randomly, as well as the additive noise $\boldsymbol{w}_i$ for each agent. We repeated the simulation for 5×10^5 iterations. The results are depicted in Figure 3. Figure 4 depicts the attacker detection performance analysis. The values of the objective function gradient at the convergence time index reveals major difference for a strong attack between both cases, giving us a large margin for our detector threshold. Clearly, full convergence is not needed to detect the presence of the attacker, allowing us to run for many fewer iterations. For a modest attack it is shown that the success of the detection scheme relies on the instance amount the algorithm is run.

VI. Conclusions

In this paper we presented a novel attack on distributed multi-agent optimization. We then described a detection method for distributed maximum likelihood estimation with i.i.d. agent data. In an extension of this work we present exponential bounds on the detection probability, as well as the localization algorithm which enables each agent to detect malicious neighbors.

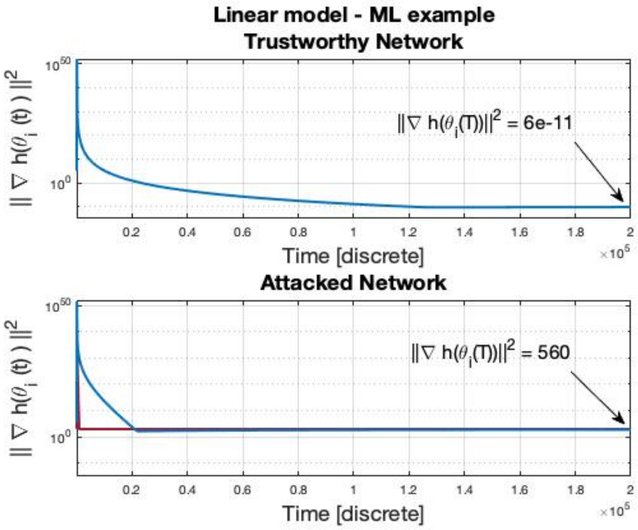

Fig. 3. Example of a linear model using the DPG algorithm with $K = 1$ and the attack scheme in (6). In both subplots the objective function gradient is presented; the top depicts the gradient for a trustworthy agent and the bottom depicts an attacked network. Note that in order to reach such margin a strong attack was applied.

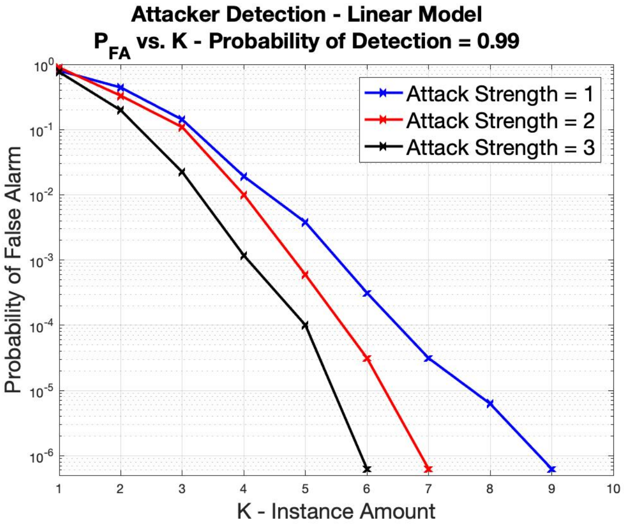

Fig. 4. Attacker Detection performance analysis for the linear model - Probability of false alarm vs. K for $P_D = 0.99$. Attacker's desired state pulls the network 1, 2 and 3 units away from the real final state.

VII. Appendix

A. Proof of Proposition 2.

Using the definition of the new attack scheme presented in (6) and the definition of the classic DPG method presented in

(3), we observe the following inequality for the attacker:

$$
\begin{aligned}
&\|\boldsymbol{\theta}_j(t+1) - \boldsymbol{\alpha}_{\mathbf{0}}\| \leq \\
&\leq g(t)\|\boldsymbol{W}_j(t)(\boldsymbol{\Theta}(t) - \boldsymbol{\alpha}_{\mathbf{0}}\mathbf{1}^T)\| + g(t)\eta(t)\|\nabla h(\boldsymbol{W}_j(t)\boldsymbol{\Theta}(t))\| \\
&+ [1 - g(t)]|\boldsymbol{z_j}(t+1)| \\
&\leq g(t)\|\boldsymbol{W}_j(t)(\boldsymbol{\Theta}(t) - \boldsymbol{\alpha}_{\mathbf{0}}\mathbf{1}^T)\| + g(t)\eta(t)Q \\
&+ [1 - g(t)]|\boldsymbol{z_j}(t+1)|
\end{aligned}
$$

where Q is the objective function gradient bound. From the definition of $g(t)$ in assumption 4, we know that $\lim_{t\to\infty} g(t) = 0$. Recall that $\boldsymbol{z_j}(t)$ vanishes a.s. over time. Therefor, we get the next asymptotic behavior:

$$\lim_{t\to\infty} \|\boldsymbol{\theta}_j(t) - \boldsymbol{\alpha}_{\mathbf{0}}\| = 0$$

Recall that trustworthy agents inject false data of second order to the network. The remaining part of the proof for the trustworthy agents is given in sub-section II.A in [11], which shows how a stubborn agent steers the network to a final state of its choice. Therefore

$$\lim_{t\to\infty} \|\boldsymbol{\theta}_i(t) - \boldsymbol{\alpha}_{\mathbf{0}}\| = 0, \quad \forall i \in V.$$

B. Proof of Theorem 1.

From **Proposition 1**, we know that the joint objective function reaches a minimum for a trustworthy network as shown in (4). For the objective function, the gradient at the optimal solution satisfies:

$$\nabla h(\boldsymbol{\theta}^*) = \nabla\left(\sum_i h_i(\boldsymbol{\theta}^*)\right) = \nabla\left(-\sum_i \log f_{\boldsymbol{x}_i}(\boldsymbol{x}_i|\boldsymbol{\theta}^*)\right) = 0$$

for every set $\boldsymbol{x_1}, ..., \boldsymbol{x_N}$. Therefore, $E_{\boldsymbol{x_1},...,\boldsymbol{x_N}}[\nabla h(\boldsymbol{\theta}^*)] = 0$. As stated before, $\boldsymbol{x}_i, i \in V$ are i.i.d. random variables; hence:

$$E\left[\nabla h(\boldsymbol{\theta}^*)\right] = E\left[\nabla\left(\sum_i h_i(\boldsymbol{\theta}^*)\right)\right] = N \times E\left[\nabla\left(h_i(\boldsymbol{\theta}^*)\right)\right]$$

for some i.
Thus $E\left[\nabla\left(h_i(\boldsymbol{\theta}^*)\right)\right] = 0$, for all $i \in V$. Therefore $E[d_i] = 0$ for a trustworthy network.
If an attacker is present in the network, the network converges to the attacker's desired final state rather to the optimal parameter; therefore $\nabla h(\boldsymbol{\theta}(T_\infty)) \neq 0$, resulting in $E\left[\nabla\left(h_i(\boldsymbol{\theta}^*)\right)\right] \neq 0$ and $E[d_i] \neq 0$.

References

[1] S. X. Wu, H.-T. Wai, A. Scaglione, A. Nedić, and A. Leshem, "Data injection attack on decentralized optimization," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 3644–3648.

[2] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[3] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[4] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." Massachusetts Inst of Tech Cambridge lab for information and decision systems, Tech. Rep., 1984.

[5] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2012.

[6] A. H. Sayed *et al.*, "Adaptation, learning, and optimization over networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[7] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

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

[9] D. Jakovetić, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014.

[10] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, 2013.

[11] R. Gentz, S. X. Wu, H.-T. Wai, A. Scaglione, and A. Leshem, "Data injection attacks in randomized gossiping," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 523–538, 2016.

[12] Y. Guan and X. 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*, vol. 4, no. 1, pp. 48–59, 2018.

[13] E. Nurellari, D. McLernon, M. Ghogho *et al.*, "Distributed detection and estimation in wireless sensor networks: resource allocation, fusion rules, and network security," Ph.D. dissertation, University of Leeds, 2017.

[14] C. Zhao, J. He, and J. Chen, "Resilient consensus with mobile detectors against malicious attacks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 60–69, 2018.

[15] Y. Guan and X. Ge, "Distributed secure estimation over wireless sensor networks against random multichannel jamming attacks," *IEEE Access*, vol. 5, pp. 10 858–10 870, 2017.

[16] M. Toulouse and P. K. Nguyen, "Protecting consensus seeking nids modules against multiple attackers," in *Proceedings of the Eighth International Symposium on Information and Communication Technology*. ACM, 2017, pp. 226–233.

[17] R. Atat, L. Liu, H. Chen, J. Wu, H. Li, and Y. Yi, "Enabling cyber-physical communication in 5g cellular networks: challenges, spatial spectrum sensing, and cyber-security," *IET Cyber-Physical Systems: Theory & Applications*, vol. 2, no. 1, pp. 49–54, 2017.

[18] R. Atat, L. Liu, J. Ashdown, M. J. Medley, J. D. Matyjas, and Y. Yi, "A physical layer security scheme for mobile health cyber-physical systems," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 295–309, 2018.

[19] E. Nurellari, D. McLernon, and M. Ghogho, "A secure optimum distributed detection scheme in under-attack wireless sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 325–337, 2018.

[20] R. Gentz, "Wireless sensor data transport, aggregation and security," Ph.D. dissertation, Arizona State University, 2017.

[21] W. Xu, Z. Li, and Q. Ling, "Robust decentralized dynamic optimization at presence of malfunctioning agents."