Brief Announcement: On The Resilience of Decentralized Sensor Networks Under Byzantine Attacks

ANONYMOUS AUTHOR(S)

This paper explores issues that impact the resilience of a decentralized sensor network to malicious attack. We formulate a graph model for sensor data validation, where the data reported by a sensor is cross-checked with data from neighboring sensors. In this model, we show that perfect validation is impossible. Subsequent results aim to understand how the structure of the network influences its susceptibility to attacks that force validators to either incorrectly reject correct data from honest sensors, or incorrectly accept wrong data from attacker-controlled sensors. We show that natural formulations of these questions are NP-hard.

CCS Concepts: • **Computer systems organization → Reliability**.

Additional Key Words and Phrases: sensor networks, decentralized systems, resilience

> *The real point of the matter is that what we call a "wrong datum" is one which is inconsistent with all other known data. It is our only criterion of right and wrong. It is the Machine's as well.*
>
> *The Evitable Conflict*, Isaac Asimov, 1950.

1 INTRODUCTION

A sensor network is a composition of spatially scattered sensors that provide readings of physical quantities. We consider the *unstructured* and *decentralized* forms of these networks. By unstructured, we mean that the sensors are arbitrarily distributed. By decentralized, we mean that the sensors are controlled by parties with possibly conflicting interests. In this adversarial setting, malicious parties may collude to provide fake but mutually consistent sensor data that either advance their interests or harm that of others; we say that these parties conduct a Byzantine attack [5]. For instance, consider decentralized sensors that monitor a scarce resource, such as water. Sensor readings may be tampered with to create a false water scarcity that inflates the price of water. Compounding the problem, it is often possible to tamper with sensor data without tampering with the sensor itself, simply by manipulating its physical environment. An example would be making a sensor "think" that a water-well is dry by siphoning its water.

Given the ease of malicious reporting, any system that gathers and acts upon decentralized sensor data must include mechanisms that filter out bad data: ideally, accepting all correct values and rejecting all incorrect ones. However, validation faces several challenges. For instance, a validator that relies only upon the distribution of values may be fooled by a value that follows the distribution but is, in fact, incorrect. One may hope that cross-checking the data provided by one sensor against the data provided by other sensors leads to more accurate validation, as suggested by the quote.

In this paper, we consider the extent to which cross-checking makes the validation process resistant to malicious attacks. We model a sensor network as a graph, where a sensor is associated with each node, and the data at that node is cross-checked against data from its graph neighbors during validation. The identities of the sensors (nodes) under malicious control are not known.

We first argue that perfect validation is unachievable. One is thus forced to consider imperfect validators. However, we rule out *fragile* validators, by assuming that the validation of a sensor reading can be subverted only if at least half of the validator's inputs are provided by adversarial sensors.

The model leads naturally to the following network analysis question: Given a network graph and a budget k on the number of Byzantine sensors, could an attacker succeed in subverting enough validations to (i) reject correct data from a large fraction of the honest nodes or (ii) accept fake data

from a large number of the malicious nodes? The answers to these questions naturally depend on the structure of the network. We show that while the questions can be answered exactly for some regular networks, they are NP-hard to resolve in general.

The contributions of this work are in the modeling of Byzantine sensor attacks, the framing of the network resilience questions, and the hardness results. These results lead to new questions on the structure of resilient networks, which are the subject of ongoing work.

2 MODEL AND RESULTS

A *Decentralized Sensor Network* (DSN) is defined as a pair (G, F). Here, $G = (V, E)$ is an undirected graph with a set of nodes V and an irreflexive edge relation E. Each node represents a sensor. The edge relation E represents the "neighborhood" of a sensor (which need not be related to proximity in space). An edge between nodes u and v implies that the validation of data provided by sensor u depends on data from sensor v (and vice-versa). The set $F = \{f_v : \mathsf{D}^{deg(v)+1} \to \mathsf{Bool} \mid v \in V\}$ is a collection of validation procedures, one for each sensor. For simplicity, we suppose that all sensors provide data from a single domain D.

We assume a *synchronous* model where sensor readings are collected and validated in *rounds*. At the beginning of a round, every sensor v reports a reading $\rho(v)$ to its validator. The validation procedure f_v queries the neighbors of v for their readings and determines whether $\rho(v)$ should be accepted, based on $\rho(v)$ and the reported neighboring readings. The formulation allows f_v to utilize additional knowledge (e.g., rainfall data for water sensing). We focus here on the decision for a single round, leaving the analysis of multi-round attacks for future work.

Call a sensor node *Byzantine* if, during a round, it either (1) *may not* report any value, or (2) *may* report an incorrect value for the physical quantity the sensor is supposed to measure. Specifically, a Byzantine sensor is allowed to report different values to different neighbors when queried for its reading. An *honest* sensor is one that is not Byzantine. That is, it reports the correct value of the physical quantity that it measures.

A *perfect validator* is one that accepts every physically correct reading (Liveness) and rejects every incorrect reading (Soundness). A validator that accepts every reading is trivially live, while one that rejects every reading is trivially sound. The challenge is in meeting both:

Theorem 2.1 (Informal). *A perfect validator is unachievable.*

The argument is by contradiction. We show that given a perfect validator, one can define a perfect computational simulation of the physical world. Hence, if a perfect validator were to exist, there would be no need for sensors!

Although validation must be imperfect, we assume that it is not fragile. We assume that a validator can reach a wrong decision only if at least half of that validator's inputs are provided by Byzantine nodes. Precisely, for validator f_v: (1) If strictly more than half of the inputs to f_v are from honest nodes, then f_v makes the correct decision on $\rho(v)$ (i.e., it accepts a correct value for $\rho(v)$ and rejects an incorrect one); and (2) On the other hand, if at least half the inputs to f_v are from Byzantine nodes then, regardless of the inputs from honest nodes, it is possible for an attacker to choose the Byzantine inputs to force an incorrect decision.

In a real system, validation may also have inherent accuracy limitations. We ignore this issue to better focus on the role of the Byzantine attacker. We assume further that the computation of f_v cannot be tampered with, which may be ensured through replication and Byzantine fault-tolerant consensus. Thus, the only way to subvert a validation is by providing misleading inputs.

The model opens up two very different ways in which a Byzantine attacker may disrupt the network. Let $B(v)$ the set of Byzantine neighbors of node v and $H(v)$ the set of honest neighbors of v. The number of inputs to f_v is thus $1 + |H(v)| + |B(v)|$.

Consider a node v. There are two possibilities, depending on whether v is honest or malicious.

(1) **Blocking**: If node v is honest, its Byzantine neighbors may force its validator f_v to reject its correct datum $\rho(v)$ by maliciously reporting values that are inconsistent with $\rho(v)$. For the attack to be effective, at least half the inputs to f_v must be Byzantine; i.e., $|B(v)| \geq 1 + |H(v)|$, as v is itself honest. The honest node v is said to be *blocked*.

(2) **Poisoning**: If node v is Byzantine, its Byzantine neighbors may force its validator f_v to accept an incorrect datum $\rho(v)$ by reporting incorrect readings which support $\rho(v)$. For the attack to be effective, at least half the inputs to f_v must be Byzantine; i.e., $1 + |B(v)| \geq |H(v)|$, as v is itself Byzantine. The Byzantine node v is said to be *poisoned*.

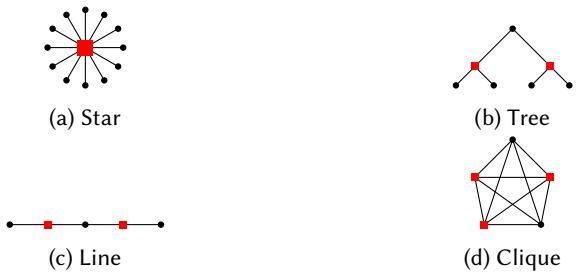

Fig. 1. Examples of blocking attacks where Byzantine sensors are represented as red squares.

Figure 1 illustrates attacks on regular networks. For the star network in part (a), a single Byzantine node at the center suffices to block *all* other nodes, each of which is honest. However, as all neighbors of the central node are honest, this node cannot be poisoned. For a complete binary tree shown in part (b), it suffices to choose the last-but-one row of nodes and every alternate row to block all remaining (honest) nodes. For a line shown in part (c), making every alternate node Byzantine suffices to block all honest nodes. But each Byzantine node has two honest neighbors, so it cannot be poisoned. Finally, for the complete graph shown in part (d), an attacker must control at least half the graph nodes to block the rest. Moreover, for each Byzantine node, two out of its four neighbors are also Byzantine, which is sufficient support for poisoning.

We see that in some networks (such as the star) many honest nodes can be blocked with a small number of well-chosen Byzantine nodes, so the network is susceptible to attack. In other cases, blocking a large number of honest nodes requires a large number of Byzantine nodes, so those networks have higher resilience. Thus, the resilience of a network is (not surprisingly) sensitive to its structure.

That gives rise to a framing of resilience in terms of network analysis: Can a large fraction of the nodes of a DSN be blocked or poisoned, provided the attacker has a budget k of Byzantine nodes? We show that both questions are NP-hard.

Theorem 2.2 (Blocking). *Given a* DSN $G = (V, E)$ *and* $k < |V|$*, it is NP-hard to determine if there exists a choice of* k *Byzantine nodes such that at least half of the remaining honest nodes are blocked.*

Theorem 2.3 (Poisoning). *Given a* DSN $G = (V, E)$ *and numbers* $t \leq k < |V|$*, it is NP-hard to determine if there exists a choice of* k *Byzantine nodes such that at least* t *of those are poisoned.*

The proof of Theorem 2.2 is through a reduction from the Positive Influence Dominating Set (PIDS) question, known to be NP-hard and even APX-hard [14, 15, 17]. The main observation is that Blocking is similar to VertexCover and PIDS, in the sense that Blocking essentially requires *some* proportion of vertices to be *half-covered*. Our proof of Theorem 2.3 is inspired by constructions in

Agrawal and Maity's study on the Small Set Vertex Expansion (SSVE) problem [1]. The key idea is that we can rephrase Poisoning in a way that is reminiscent to Vertex Expansion, and therefore mimic Agrawal and Maity's construction. Full proofs are available in the anonymized report at [8].

These results are, is in a way, encouraging. They suggest that it is difficult for an attacker to find an optimal attack, one that requires subverting the least number of sensors. On the other hand, they also suggest that it is difficult to determine whether a network is sufficiently resilient to attack. This conundrum prompts new questions for further research, such as: (1) Which classes of networks have inherently high resiliency? and (2) Could the resiliency of a given network be strengthened by augmenting its structure?

3 RELATED WORK

The resilience of sensor networks to Byzantine attacks has been studied from several perspectives including Byzantine node detection, game-theoretic analysis of Byzantine behavior, and establishment of a chain of trust. We discuss the most closely related work.

Marano, Matta and Tong [7] consider distributed detection in the presence of a cooperative Byzantine attack. Their model differs from ours in key respects: (1) A sensor can only fake its own data but cannot impact the other sensors' data, while we allow *blocking* and *poisoning*, and (2) The objective of an adversarial sensor is to maximize the difference between the true and the accepted fake value; we focus on bounding the number of accepted fake values.

The work of Sun, Zhang and Fan [12] identifies adversarial nodes based on a likelihood metric. In contrast to our model, this work assumes that the validators know the detection efficiency for each sensor, the total number of adversarial sensors, and the probability of receiving the information from the sensor.

Haeberlen, Kouznetsov and Druschel [3] designed a detection system called PeerReview, which assumes a certain degree of observability of Byzantine behaviour. Our model does not aim for detection, but rather supports the analysis of resilience of a network to Byzantine attacks.

Hespanha [4] models Byzantine attacks as a noncooperative game where the objective of the adversaries is to force validators to accept wrong readings. That work focuses on the construction of Nash equilibria for this game; while our results focus on how the resilience of the network is influenced by its structure.

Several so-called "oracle" data providers (e.g., Chainlink [2]) seek to supply trustworthy real-world data to smart contracts by relying on cryptographic signatures of intermediate data providers (*i.e.,* "end-to-end integrity"). However, the integrity of the data path does not suffice when the validity of the original raw data is at issue, which is a serious concern in distributed sensor networks [11]. Our work analyzes the degree to which the raw data can be validated.

REFERENCES

[1] Garima Agrawal and Soumen Maity. 2021. The Small Set Vertex Expansion Problem. *Theor. Comput. Sci.* 886 (2021), 84–93.

[2] Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, et al. 2021. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. *Chainlink Labs* 1 (2021), 1–136.

[3] Andreas Haeberlen, Petr Kouznetsov, and Peter Druschel. 2007. PeerReview: practical accountability for distributed systems. In *SOSP*. ACM, 175–188.

[4] João P Hespanha. 2021. Sensor manipulation games in cyber security. *Game Theory and Machine Learning for Cyber Security* (2021), 137–148.

[5] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* 4, 3 (jul 1982), 382–401. https://doi.org/10.1145/357172.357176

[6] Pangfeng Liu and Sandeep N. Bhatt. 2000. Experiences with Parallel N-Body Simulation. *IEEE Trans. Parallel Distributed Syst.* 11, 12 (2000), 1306–1323.

[7] Stefano Marano, Vincenzo Matta, and Lang Tong. 2008. Distributed detection in the presence of Byzantine attacks. *IEEE Transactions on Signal Processing* 57, 1 (2008), 16–29.

[8] Anonymized Full Paper. 2024. On The Resilience of Decentralized Sensor Networks Under Byzantine Attacks. https://github.com/anonjohn125/podc2024/blob/main/Towards_Resilient_Sensor_Network_in_Adversarial_Setting.pdf.

[9] John H. Reif. 2009. Mechanical Computing: The Computational Complexity of Physical Devices. In *Encyclopedia of Complexity and Systems Science*. Springer, 5466–5482.

[10] John H. Reif and Stephen R. Tate. 1993. The Complexity of N-body Simulation. In *ICALP (Lecture Notes in Computer Science, Vol. 700)*. Springer, 162–176.

[11] Amit Kumar Sikder, Giuseppe Petracca, Hidayet Aksu, Trent Jaeger, and A. Selcuk Uluagac. 2021. A Survey on Sensor-Based Threats and Attacks to Smart Devices and Applications. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 1125–1159. https://doi.org/10.1109/COMST.2021.3064507

[12] Ziteng Sun, Chuang Zhang, and Pingyi Fan. 2016. Optimal Byzantine attack and Byzantine identification in distributed sensor networks. In *2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 1–6.

[13] N. N. Vasiliev and D. A. Pavlov. 2017. Computational complexity of the initial value problem for the three body problem. *CoRR* abs/1704.08762 (2017).

[14] Feng Wang, Erika Camacho, and Kuai Xu. 2009. Positive Influence Dominating Set in Online Social Networks. In *COCOA (Lecture Notes in Computer Science, Vol. 5573)*. Springer, 313–321.

[15] Feng Wang, David Hongwei Du, Erika Camacho, Kuai Xu, Wonjun Lee, Yan Shi, and Shan Shan. 2011. On positive influence dominating sets in social networks. *Theor. Comput. Sci.* 412, 3 (2011), 265–269.

[16] Xu Zhu, Jieun Yu, Wonjun Lee, Donghyun Kim, Shan Shan, and Ding-Zhu Du. 2010. New dominating sets in social networks. *J. Glob. Optim.* 48, 4 (2010), 633–642.

[17] Feng Zou, Zhao Zhang, and Weili Wu. 2009. Latency-Bounded Minimum Influential Node Selection in Social Networks. In *WASA (Lecture Notes in Computer Science, Vol. 5682)*. Springer, 519–526.

A PROOF OF MAIN RESULTS

In this section, we prove our main results.

A.1 Infeasibility Result

We postulate that perfect physical modelling is infeasible in practice. Combined with Theorem A.1 (below), we conclude that it is infeasible to build a perfect validator in practice, establishing Theorem 2.1.

For a sensor v, we denote by $W(v)$ the true but hidden value for the sensor in the real world. A simulation M is a program that, provides a value $M(v)$ for sensor v. Simulation M is said to be *ϵ-close* if $W(v) - \epsilon \leq M(v) \leq W(v) + \epsilon$ for all v's. We postulate that, when ϵ is sufficiently small, a ϵ-close simulation is not feasible. We support our postulate by an abundance of references. A famous result in physical simulation by Reif and Tate [10] states that simulating the n-body problem is PSPACE-hard to approximate within poly(n) bits of accuracy. Here, the n-body simulation problem is: "Given initial positions and velocities of n particles that have pair-wise force interactions, simulate the movement of these particles so as to determine the positions of the particles at a future time." For infeasibility of simulating the n-body problem in practice, see [6] which attempts parallel classical computing. For other infeasibility results on simulating physical systems, we refer readers to Reif's survey [9]. In particular, Vasiliev and Pavlov showed that complexity lower bound for the three-body problem is not bounded by any polynomial, and the authors reckoned that the choice of the three-body problem is not crucial and the results can be extended to "other systems where trajectories with a complex dynamical behavior can be constructed and analyzed by methods of symbolic dynamics" [13].

Theorem A.1. *An ϵ-perfect validator induces an ϵ-perfect simulation.*

Proof. We say a validator for a sensor v is *ϵ-perfect* if, given a reading $\rho(v)$ from its sensor v, and readings from neighboring sensors, it accepts $\rho(v)$ if that value is ϵ-close to the true value $W(v)$. Given an ϵ-perfect validator, we define an ϵ-close simulation M as follows. First, fix a feasible range for a sensor reading; e.g. 0 to 100 degrees Centigrade for water temperature. The simulator M generates values in steps of ϵ within this range, e.g., 0, ϵ, 2ϵ, ..., through 100, and submits each value to the ϵ-perfect validator. When the validator accepts a value, the simulator outputs that value as its result. It is clear that this is an ϵ-close simulation. Since that is postulated to be impossible in practice, an ϵ-perfect validator is impossible in practice. □

A.2 Proof of Theorem 2.2

Recall the statement of Theorem 2.2: Given a DSN $G = (V, E)$, it is NP-hard to determine if there exists an arrangement of k Byzantine sensors such that at least half of the remaining honest sensors are blocked.

Recall that a Positive Influence Dominating Set (PIDS) for a graph $G = (V, E)$ is a subset $D \subseteq V$ such that any node v in $V \setminus D$ has at least $\left\lceil \frac{d(v)}{2} \right\rceil$ neighbors in D, where $d(v)$ is the degree of node v [17]. A closely related variant is called Total Positive Influence Dominating Set (TPIDS), where the domination property holds for all $v \in V$, i.e. including those in D [14] (In fact, [15] calls this variant PIDS). Consider the decision problem for PIDS (*resp.* TPIDS): given a graph $G = (V, E)$ and a positive integer k, does there exist a PIDS (*resp.* TPIDS) D of size at most k? Both decision problems are shown to be NP-complete and even APX-hard [15–17].

To prove Theorem 2.2, we need a helper lemma. We define *a m-constrained PIDS language* (for integer $m > 0$) as a set of pairs (G, k) with $k > 0$ and $n > m * k$ (where n is the number of nodes of G) that belong to PIDS.

Lemma A.2. *Deciding m-constrained PIDS is NP-hard (NP-complete) for every integer $m > 0$.*

Proof. We reduce from the original PIDS formulation, which is NP-hard, to the constrained ones.

Given a graph G with n nodes and k such that $0 < k \leq n$, the original PIDS formulation asks whether there is a "dominating" subset D of the nodes of G such that every node of G not in D is dominated. A node x is said to be *dominated* if at least $\lceil \frac{\text{degree}(x)}{2} \rceil$ of its neighbors are in D. Equivalently, a node is dominated if $2 \cdot |\text{neighbors of } x \text{ in } D| \geq \text{degree}(x)$.

Given an instance (G, k) of PIDS, we form the instance (G', k') as follows, where $k' = k + 1$. Define G' as a copy of G, together with a disjoint star graph that has $wk + 1$ nodes for $w = 2m + 1$. That is, wk nodes in the periphery of the star, and one node in the center.

We first show that G' is an m-constrained PIDS instance. For this, we require that n', the number of nodes of G', is strictly greater than mk'. A direct calculation shows that

$$\begin{aligned} n' &= n + wk + 1 \\ &\geq k + wk + 1 \quad \text{... as } n \geq k \text{ by definition of PIDS instances.} \\ &> (1 + w)k \end{aligned}$$

It remains to show that $(1 + w)k \geq mk' = m(k + 1)$. As $k > 0$, dividing both sides by k, we get $(1+w) \geq m(1+\frac{1}{k})$. As $1+\frac{1}{k} \leq 2$ for all integer $k > 0$, it suffices to choose w such that $(1+w) \geq 2m$, which is true for $w = 2m + 1$.[1]

We now show that any solution to (G, k) induces a solution to (G', k'). Let D be a dominating set for (G, k). Let $D' = D + \{\text{center of the star graph}\}$. Any node x in G' that is not in D' is either in G and not in D, or is a node in the periphery of the star graph. In the first case, it is dominated as D is a dominating set for G, and the star graph and G are disconnected. In the second case, it is dominated as node x has degree 1 and the center of the star is in D'. Note that D' has size $k + 1$.

We now show that any solution to (G', k') induces a solution to (G, k). Let D' be a dominating set for (G', k'). The number of peripheral nodes in the star graph is $(2m + 1)k$ which is strictly greater than $(k + 1)$ for any integers $k > 0$ and $m > 0$. Thus, at least one peripheral node of the star is not in D'. This node is required to be dominated, which forces the center of the star to be in D'. Let D be the restriction of D' to G; as the center of the star is in D', the set D has size at most k. Every node x of G not in D' is dominated, hence x must be dominated by D. Thus, D is a dominating set for G of size at most k. This completes the proof. □

We are now ready to prove Theorem 2.2.

Proof of Theorem 2.2. We show this by reducing the 4-constrained PIDS problem to Blocking. Given an instance (G, k) of the 4-constrained PIDS problem, i.e., where n (number of nodes in G) $> 4k$, we construct an instance (G', k') of Blocking as follows.

G' is G together with a clique W of $n - k + 1$ nodes. In W, there is a distinguished node w_0 that is connected to every node of G. We set $k' := k + 1$.

G' has $2n-k+1$ nodes. In any assignment, $k+1$ nodes are Byzantine, so there are $2n-k+1-(k+1) = 2(n - k)$ honest nodes. For the assignment to be a Blocking set, at least half of those honest nodes must be blocked, i.e., there should be at least $(n - k)$ blocked nodes.

Let D be a dominating set for G. A *blocking set* for G' is a set of Byzantine nodes of G' that suffices to block at least half of the honest nodes of G'.

Claim: Any dominating set for G induces a blocking set for G'.

[1] That is also true for $w = 2m - 1$, but we need the higher value for the other claim.

Proof. Let $D+\{w_0\}$ be the Byzantine nodes creates a blocking set for G'. Consider a honest node x in G. This node is not in D. As D is a dominating set for G, we know that 2|neighbors of x from G in $D| \geq$ degree(x, G).

Now we calculate:

$$\begin{aligned} 2|\text{byzantine neighbors of } x \text{ in } G'| &= 2(|\text{neighbors of } x \text{ from } G \text{ in } D| + 1) \quad (\text{ the } +1 \text{ is node } w_0) \\ &= 2|\text{neighbors of } x \text{ from } G \text{ in } D| + 2 \\ &\geq \text{degree}(x, G) + 2 \quad (\text{ by PIDS property for } G) \\ &= \text{degree}(x, G') + 1 \quad (\text{ as } x \text{ is connected to } w_0 \text{ in } G') \end{aligned}$$

Therefore, x is blocked. The number of nodes in G that are not in D is $(n-k)$. As all of these are blocked, $D + \{w_0\}$ is a blocking set for G'. □

Conversely, we note that:
Claim: Any blocking set for G' induces a dominating set for G.

Proof. Let B' be a blocking set for G'. From B', we show how (below) to construct a (possibly different) solution B'' which consists of w_0 and k nodes of G, and where every node in G that is not in B'' is blocked.

In such a solution, we have that for any node x of G not in B'':

$$2|\text{neighbors of } x \text{ in } B''| \geq 1 + \text{degree}(x, G'), \quad \text{i.e.,}$$
$$2 * (|\text{neighbors of } x \text{ in } B'' \text{ and in } G| + 1) \geq 1 + 1 + \text{degree}(x, G)$$

Simplifying, we get:

$$2|\text{neighbors of } x \text{ in } B'' \text{ and in } G| \geq \text{degree}(x, G)$$

Hence, the nodes in B'' and G form a dominating set for G. □

We now provide the transformation from a blocking set B' to a blocking set B'' that has the properties stated above. We first show that no node in W can be blocked in B', essentially as there are not enough Byzantine nodes. Consider a honest node y in W. There are two possibilities for y.

If y is w_0, then to block w_0, we must have

$$2|\text{Byzantine neighbors of } w_0 \text{ in } G'| \geq 1 + \text{degree}(w_0, G') \quad \text{i.e.,}$$
$$\begin{aligned} 2|\text{Byzantine neighbors of } w_0 \text{ in } G'| &\geq 1 + n + (n-k) \quad (\text{as } w_0 \text{ is connected to all nodes of } G') \\ &\geq 1 + 2n - k \\ &\geq 1 + 8k + 2 - k \quad (\text{as } (n > 4k), \text{i.e., } (n \geq 4k+1) \\ &> 2(k+1) \end{aligned}$$

Consider any other node y in W. To block y, we must have

$$\begin{aligned} 2|\text{byzantine neighbors of } y \text{ in } G'| &\geq 1 + \text{degree}(y, G') \quad \text{i.e.,} \\ 2|\text{byzantine neighbors of } y \text{ in } G'| &\geq 1 + (n-k) \\ &\geq 3k + 2 \quad (\text{as } n > 4k, \text{i.e., } n \geq 4k+1) \\ &> 2(k+1) \quad (\text{as } k > 0) \end{aligned}$$

In both situations, the number of Byzantine neighbors required to block the node is strictly greater than $k+1$, which breaks the budget.

It follows that all the blocked nodes are in G. As B' is a blocking set, there must be at least $(n-k)$ blocked nodes in G. Let k_0 be the number of Byzantine nodes in G. This number cannot be $k+1$, as

G would have at least $(n-k)+(k+1) > n$ nodes, a contradiction. Hence, at least one Byzantine node is in W. Let W have k_1+1 Byzantine nodes, for some $k_1 \geq 0$.

Now we transform from B' to B'' as follows. First, if w_0 is not Byzantine, we transfer the Byzantine label from a different node of W to w_0. This cannot unblock any node in G as all nodes in G gain a Byzantine neighbor. Next, we choose a subset X of G with exactly $(n-k)$ blocked nodes. We then move the remaining k_1 Byzantine labels from W to the k_1 honest nodes outside X in G. (There are precisely k_1 such, as G has $(n-k_0)$ honest nodes in B' and $(n-k_0) = (n-k)+k_1$.) This relabeling also does not unblock any node in X, as every node in X may only gain a Byzantine neighbor. The resulting assignment B'' meets the requirements of the proof. □

Remark. Note that the proof actually shows the following slightly stronger result, since the constructed G' is connected and the reduction is clearly an L-reduction.

Corollary A.3. *Given a connected* DSN $G = (V, E)$*, it is APX-hard to determine if there exists an arrangement of k Byzantine sensors such that at least half of the remaining honest sensors are blocked.*

A.3 Proof of Theorem 2.3

Recall the statment of Theorem 2.3: Given a DSN $G = (V, E)$ and numbers $t \leq k < n$, it is NP-hard to determine if there exists an arrangement of k Byzantine sensors such that at least t of them are supported (i.e., poisoned).

Our proof of Theorem 2.3 is inspired by Agrawal and Maity's study on the Small Set Vertex Expansion (SSVE) problem [1].

Proof of Theorem 2.3. We denote an input to our problem to be (G, k, t). A moment of reflection should convince the reader that it suffices to prove the NP-hardness of the following problem: given a DSN $G = (V, E)$, does there exist a set of t sensors which can be simultaneously supported by $k-t$ Byzantine sensors? We shall present a polynomial time reduction from CLIQUE which will finish the proof.

Given an input (G, k) to CLIQUE, we construct the input $(G', \frac{k^2+k}{2}, \frac{k^2-k}{2})$ to our problem: $G' = (V', E')$ where $V' = \{e \in E\} \cup \{v^i \mid v \in V, i \in \{1,2\}\} \cup \{p_j^i \mid i \in \{1,2\}, j \in \{1,2,\ldots,k^2+k+1\}\}$ and $E' = \{\{p_j^i, v^i\} \mid i \in \{1,2\}, j \in \{1,2\ldots,k^2+k+1\}, v \in V\} \cup \{\{e, v^i\} \mid i \in \{1,2\}, v \in e\} \cup \{\{p_j^i, p_{j'}^i\} \mid i \in \{1,2\}, j, j' \in \{1,2,\ldots,k^2+k+1\}\}$. See Figure 2 for a visualization of the construction. The middle column of vertices is a copy of the edges from the original graph; the leftmost column is the vertices p_j^1, which form a clique; the second from left column is the v^1 copy of the original vertex set. The right-hand side of the figure similarly represents the v^2 copies and a clique over p_j^2.

It remains to show this is a valid reduction.

In one direction, suppose that (G, k) is an Yes instance to CLIQUE and let C_k be the k-clique subgraph in G, then by choosing the Byzantine sensors to be $\{e \in C_k\} \cup \{v^1 \mid v^1 \in e\}$, we find a Byzantine set of size $\frac{k^2-k}{2} + k = \frac{k^2+k}{2}$ with supported set $\{e \in C_k\}$ of size $\frac{k^2-k}{2}$.

In the other direction, suppose the input $(G', \frac{k^2+k}{2}, \frac{k^2-k}{2})$ is an Yes instance to our problem, we claim that the supported set must be all "edge" vertices and therefore the remaining Byzantine sensors form a $\frac{k^2+k}{2} - \frac{k^2-k}{2} = k$-clique in G. Indeed, suppose this is not the case, then there must exist some v^i or some p_j^i in the Byzantine sensors that is supported. Say that p_j^i for some i, j is supported. This vertex has at least k^2+k+1 neighbors (k^2+k many $p_{j'}^i$'s and at least one v^i's), half of which must be Byzantine. But $\frac{k^2+k+1}{2} > \frac{k^2+k}{2}$, exceeding the budget. Similarly, suppose that v^i is supported. This vertex has at least k^2+k+1 neighbors among the p_j^i's, which exceeds the Byzantine budget. This finishes the proof. □

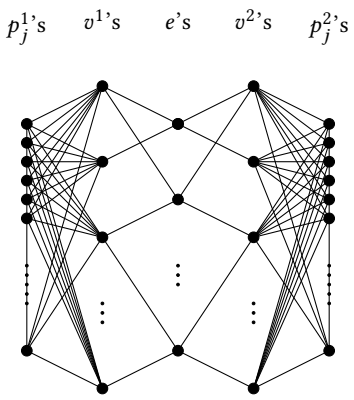

Fig. 2. Construction in the proof of Theorem 2.3

Remark. The proof of Theorem 2.3 can be easily adapted so that a Byzantine sensor is said to be $\frac{1}{r}$-*supported*, where $r \in \mathbb{N}$, if more than $\frac{1}{r}$ neighbors are Byzantine. The number $\frac{1}{r}$ is called the *validator robustness*: A higher $\frac{1}{r}$ means implies that more Byzantine neighbors are needed to support a single Byzantine sensor, which yields a higher security of the network. We simply define $G' = (V', E')$ where $V' = \{e \in E\} \cup \{v^i \mid v \in V, i \in [r]\} \cup \{p_j^i \mid i \in [r], j \in [k^2 + k]\}$ and $E' = \{\{p_j^i, v^i\} \mid i \in [r], j \in [k^2 + k], v \in V\} \cup \{\{e, v^i\} \mid i \in [r], v \in e\}$ while the input is $(G', \frac{k^2+k}{2}, \frac{k^2-k}{2})$. The proof of the following theorem will follow the same reasoning in the proof of Theorem 2.3.

Theorem A.4. *Given a* DSN $G = (V, E)$*, it is NP-hard to determine if there exists an arrangement of k Byzantine sensors such that at least t of them are $\frac{1}{r}$-supported.*