Measurement Integrity Attacks against Network Tomography: Feasibility and Defense

Shangqing Zhao, *Student Member, IEEE,* Zhuo Lu, *Member, IEEE,* and Cliff Wang, *Fellow, IEEE*

Abstract—Network tomography is an important tool to estimate link metrics from end-to-end network measurements. An implicit assumption in network tomography is that observed measurements indeed reflect the aggregate of link performance (i.e., *seeing is believing*). However, it is not guaranteed today that there exists no anomaly (e.g., malicious autonomous systems and insider threats) in large-scale networks. Malicious nodes can intentionally manipulate link metrics via delaying or dropping packets to affect measurements. Will such an assumption render a vulnerability when facing attackers? The problem is of essential importance in that network tomography is developed towards effective network diagnostics and failure recovery. In this paper, we demonstrate that the vulnerability is real and propose a new attack strategy, called *measurement integrity attack*, in which malicious nodes can substantially damage a network (e.g., delaying packets) and at the same time maliciously manipulate end-to-end measurement results such that a legitimate node is misleadingly identified as the root cause of the damage (thereby becoming a scapegoat) under network tomography. We formulate three basic attack approaches and show under what conditions attacks can be successful. We also reveal conditions to detect and locate such attacks in a network. Our theoretical and experimental results show that simply trusting measurements leads to measurement integrity vulnerabilities. Thus, existing methods should be revisited accordingly for security in various applications.

Index Terms—Network tomography; measurement integrity; scapegoating; security; attack feasibility; attack detection and localization.

1 Introduction

Accurate and timely monitoring of network performance is vital to ensure a reliable and efficient network environment. To this end, network operators may use network management protocols to monitor the network. For example, tools based on the simple network management protocol (SNMP) [2] can be used to periodically query individual network components to find potential anomalies or malfunctions. However, such a way of directly measuring the performance of internal components is not always feasible due to the lack of support functionality at network components, measurement traffic overhead, or prohibition in autonomous systems.

Network tomography has emerged as an alternative measurement algorithm primarily used for network monitoring, diagnosis and failure localization (e.g., [3], [4], [5], [6], [7]) inside a network, where directly measuring the performance of individual components is not always possible. In network tomography, monitoring nodes (also known as monitors) send packets between each other. A network link's quality metric, such as delay or packet loss, is inferred from the end-to-end measurements based on the knowledge of how packets are routed over end-to-end paths between these monitors. Therefore, it avoids directly measuring the performance of individual network links and has enabled wide applications in both wireline networks (e.g., [7], [8], [9], [10]) and wireless networks (e.g., [11], [12], [13]) without special cooperation from internal nodes.

By nature, network tomography does not directly observe network link metrics, but "tele-measure" them via measurements over end-to-end paths. Each path consists of a few or more links. Existing work mainly focused on algorithm design and applications (e.g., [7], [8], [9], [10], [11], [12], [13]); and some recent papers also considered the problems of placement of monitors and identifiability of link metrics (e.g., [14], [15], [16], [17]). In essence, network tomography can be considered as an algorithmic process to transfer end-to-end measurements into link metric estimates. Interestingly, most existing studies on network tomography emphasize extracting as much information about link metrics as possible from available measurements, and always make a *seeing-is-believing* assumption that measurements over end-to-end paths between monitors indeed reflect the real performance aggregates over individual links. However, such an assumption does not always hold in the presence of malicious autonomous systems [18], [19], backdoor-infected routers [20], and node-capture attacks [21], [22] as these adversaries actively affect packet forwarding and have become increasingly possible in today's complicated environments. Rather, the assumption renders a serious security problem of measurement integrity during the network tomography process.

In this paper, we develop a new class of attack strategies, called *measurement integrity attacks*, which take advantage of this *seeing-is-believing* vulnerability in network

Shangqing Zhao and Zhuo Lu are with Department of Electrical Engineering, University of South Florida, Tampa FL, 33620.
Emails: {zhuolu@, shangqing@mail.}usf.edu.
Cliff Wang is with Department of Electrical and Computer Engineering with North Carolina State University, Raleigh NC 27695.
Email: cliffwang@ncsu.edu.
An earlier version of this work [1] was presented in IEEE ICDCS 2017.

tomography. Unlike conventional data integrity problems that are usually protected by standard methods (e.g., encryption and authentication), a key challenge associated with measurement integrity attacks is that the facts (e.g., packet transmission/delivery timings) during network measurement cannot be protected by such standard methods, but can be easily manipulated by malicious attackers. The basic idea of measurement integrity attacks is to intentionally delay or drop packets at malicious nodes to manipulate end-to-end measurements between monitors in a way such that a legitimate node is incorrectly identified by network tomography as the root cause of the problem, thereby becoming a scapegoat. We propose three basic attack strategies with different objectives.

1) Chosen-victim scapegoating, in which attackers target one or more given victims such that these victims are misleadingly identified by network tomography as the root cause of a problem.
2) Maximum-damage scapegoating, in which attackers find the optimal set of victims among all nodes to inflict the maximum damage to the network.
3) Obfuscation, by which network tomography is tricked to produce a substantial number of link estimates beyond the normal status to confuse a network operator.

We analyze the feasibility of these strategies, and present the conditions for detecting and locating such attacks. We also use network datasets to perform simulation experiments to show the success possibility, damage, and the detectability and locatability of such attacks. Our main contributions can be summarized as follows.

- We are the first to investigate the vulnerability in network tomography mechanisms from a security perspective, and reveal that measurement integrity attacks are able to damage the network while substantially misleading network tomography without knowing the global routing knowledge of the network.
- We systematically construct three basic attack strategies, and investigate the feasibility of such attacks, then propose methods to detect and locate measurement integrity attacks.
- We use real-world datasets to evaluate the threats of measurement integrity attacks in network systems with various settings. Experimental results demonstrate that the current practice of network tomography is even vulnerable to a single attacker.

Our work demonstrates that when a measurement integrity attack is successfully launched, network tomography generates misleading and erroneous outputs, based on which failure recovery or mitigation procedures may further exacerbate the damage caused by the attack. As security plays a critically important role in network design and measurement, network tomography should be developed not only for conventional goals such as efficiency and identifiability, but also for security. Hence, existing network tomography methods in various applications need to be revisited to increase attack resilience and adopt necessary detection mechanisms.

Note that our early work in [1] has identified that such vulnerability of network tomography can be leveraged by attackers to mislead the network tomography. In this work, we have the following improvements:

1) In [1], the global routing information is assumed to be known by attackers while launching the attacks. However, in most networks, such information is generally unavailable since it is encrypted in the network or higher layers. To this end, different from [1], this paper focuses on investigating a more practical problem of how to launch the attack with partial information, i.e., attackers have no global routing information. Then, we systematically model and analyze this problem in Section 3-C, and introduce a new constraint for our improved optimization framework in Section 3-D. We also adjust the three attack strategies by reconsidering this new constraint in Section 3-E.
2) From the detection perspective, apart from a detection method in [1], this paper proposes a new localization strategy which can not only detect if attacks exist, but, most importantly, explicitly pinpoint which links or nodes are true attackers (as shown in Section 5).
3) We re-conduct our simulations to accommodate the partial information environments, and provide new results of locating attacks in Section 6.

The remainder of this paper is organized as follows. In Section 2, we introduce the models and state the research problem. In Section 3, we design and discuss the attack strategies. In Section 4, we analyze the feasibility of measurement integrity attacks and describe how to detect it. In Section 5, we demonstrate how to locate such attacks. In Section 6, we present experimental results. We discuss observations and introduce the future work from analyses and experiments in Section 7, describe related work in Section 8 and finally conclude in Section 9.

2 Models and Problem Statement

In this section, we first review network tomography and introduce the basic idea behind measurement integrity attacks. Then, we state our research problems. All notations are defined in Table 1.

2.1 Network Models and Assumptions

We consider a connected network with a known topology denoted by graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V} = \{v_i\}_{i\in[1,|\mathcal{V}|]}$ and $\mathcal{L} = \{l_i\}_{i\in[1,|\mathcal{L}|]}$ represent the sets of nodes and links, respectively. There is at most one link between nodes v_i and v_j for $i \neq j$ and no link for $i = j$ (i.e., no self-loop). Link l_i is associated with a link metric x_i. We assume that link metrics are additive, i.e., the overall measurement metric of an end-to-end path is the sum of individual

TABLE 1
Notations used throughout the paper.

$\mathbf{A}^T$	The transpose of matrix $\mathbf{A}$.
$\mathbf{A}^{-1}$	The inverse of matrix $\mathbf{A}$.
$\|\mathbf{a}\|_1$	The $\mathcal{L}$-1 norm of vector $\mathbf{a} = [a_1, a_2, \cdots, a_n]^T$, i.e., $\|\mathbf{a}\|_1 = \sum_{i=1}^{n} \|a_i\|$.
$\mathbf{x} \succeq \mathbf{y}$	Componentwise larger than or equal to, i.e., $x_i \geq y_i$ for every index i and pair of $x_i \in \mathbf{x}$ and $y_i \in \mathbf{y}$.
$\mathbf{0}$	All-zero vector.
$\|\mathcal{A}\|$	The cardinality of set $\mathcal{A}$.

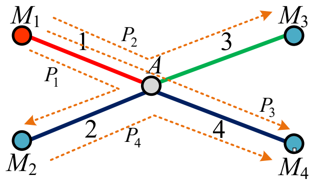
link metrics over the path. For example, delay metrics are additive; and packet delivery or loss ratios are also additive in the logarithmic form [6], [17], [23].

Throughout this paper, we adopt similar assumptions in the literature for network tomography (e.g., [15], [16], [17]): (i) a network operator chooses a number of nodes in the network as monitors, which send probe packets between each other to monitor the additive metric of each individual link; (ii) the network operator will collect all measurement results from monitors, and then perform network tomography for monitoring and diagnosis purposes.

In addition, we adopt the assumption that the monitors can control the routing of probe packets over a path as long as the path starts and ends at different monitors. Although end nodes usually have no control of the routing path of a common IP packet, network tomography relies on such a controllable routing assumption (e.g., [15], [16], [17]). It is known from existing studies (e.g., [24], [25]) that controllable routing served for network measurement can be generally supported in (i) networks under common administration, (ii) networks with strict (or loose) source routing, such as wireless networks with ad-hoc on demand distance vector (AODV) routing, or (iii) certain software-defined network (SDN) scenarios where monitors, with the help of the SDN controller, can decide paths of measurement packets. How exactly controllable routing is designed for network tomography is complementary to the work in this paper that focuses on exploiting the network tomography process and launching measurement integrity attacks.

2.2 Network Tomography and Formulation

Network tomography [8] is an algorithm to estimate link metrics from end-to-end measurements. Theoretically, we form the link metrics as a column vector $\mathbf{x} = [x_1, x_2, \cdots, x_{|\mathcal{L}|}]^T$. To efficiently estimate $\mathbf{x}$, monitors first select a set of measurement paths between each other, denoted by $\mathcal{P} = \{P_i\}_{i \in [1,|\mathcal{P}|]}$. Then, they send probe packets over the paths in $\mathcal{P}$ to obtain the path measurement metrics, which are denoted as a column vector $\mathbf{y} = [y_1, y_2, \cdots, y_{|\mathcal{P}|}]^T$. As a path measurement metric in $\mathbf{y}$ is usually the sum of multiple link metrics in $\mathbf{x}$ (e.g., a path delay is the sum of multiple link delays), it has been shown [14] that the linear relation between $\mathbf{x}$ and $\mathbf{y}$ can be represented as



Path 1 $(M_1 \to M_2)$: links 1, 2
Path 2 $(M_1 \to M_3)$: links 1, 3
Path 3 $(M_1 \to M_4)$: links 1, 4
Path 4 $(M_2 \to M_4)$: links 2, 4

Fig. 1. A simple network example, where $M_1 - M_4$ are monitors, and M_1 is malicious.

$$\mathbf{y} = \mathbf{R}\mathbf{x}, \tag{1}$$

where $\mathbf{R} = (R_{i,j})$ is called the routing or measurement matrix whose entry $R_{i,j}$ has value 1 if link $l_j \in \mathcal{L}$ is present on path $P_i \in \mathcal{P}$, and value 0 otherwise. Network tomography in essence inverts the linear system in (1) to solve for $\mathbf{x}$ given $\mathbf{R}$ and $\mathbf{y}$. Existing studies on selecting or placing monitors (e.g., [15], [16]) ensure that $\mathbf{R}$ is revertible (or full column rank) and the solution to (1) can be obtained as

$$\hat{\mathbf{x}} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\mathbf{y}. \tag{2}$$

The estimate $\hat{\mathbf{x}}$ is expected to have values close to the real link metric vector $\mathbf{x}$, and will be used as decisive information for link status monitoring, network diagnostics or further failure recovery.

2.3 Motivation and Basic Idea of Measurement Integrity Attacks

Network tomography does not directly measure network link performance, but deduces such performance from the aggregate measurements observed by monitors. Therefore, the reliability of network tomography relies on an implicit assumption that measurements over end-to-end paths indeed reflect the real performance aggregates over individual links. However, such probe packets may go through malicious autonomous systems [18], [19], intentional bandwidth throttling systems [26], backdoor-infected routers [20] or attack-captured nodes [21], [22] that can intentionally or maliciously cause negative impacts on end-to-end measurements. Thus, such an assumption may not always hold in today's complicated network environments.

Suppose that some nodes in the network are malicious and intend to cause damage. A straightforward attack is that they delay or drop all packets routed to them. However, it is easy for the network operator to detect that the links connecting to these nodes suffer long delay or high loss under network tomography. Therefore, a much more important question is whether it is possible for these malicious nodes to launch attacks and at the same time mislead network tomography.

To demonstrate the idea of such an attack, we consider a naive scenario shown in Fig. 1, where nodes M_1,

M_2, M_3 and M_4 are monitors that perform network tomography to estimate link metrics, and the number on each edge denotes the link index. These monitors choose 4 paths[1] listed in Fig. 1 for end-to-end measurement. Assume that node M_1 is malicious, which means that it can adversely affect the performance of link 1 to damage the network, such as delaying packets passing through link 1.

From Fig. 1, the attacker M_1 associated with link 1 presents on paths 1-3. If the attacker M_1 simply delays or drops all packets going through link 1, it is very easy to be identified by network tomography as the root cause. Instead, our proposed attack strategy is that the attacker can try to delay or drop packets along certain directions to mislead tomography. Specifically, in Fig. 1, the attacker M_1 is on all measurement paths containing link 3 (i.e., path 2). If the attacker M_1 only does the damage on path 2, and do nothing on other paths (e.g., paths 1, 3), the induced measurements under the network tomography algorithm (2) will show that path measurements containing link 3 always suffer long delay or high loss, while the others appear to be normal.

For example in Fig. 1, we have the system (1) as

$$\begin{aligned} &\text{Path } 1: y_1 = x_1 + x_2 \\ &\text{Path } 2: y_2 = x_1 + x_3 \\ &\text{Path } 3: y_3 = x_1 + x_4 \\ &\text{Path } 4: y_4 = x_2 + x_4. \end{aligned} \tag{3}$$

Clearly, the routing matrix $\mathbf{R}$ in (3) satisfies the full column rank requirement. Now suppose an ideal case that the network is congestion free (i.e., almost 0ms delay on every link), and the attacker only damages path 2 by inflicting extra 1000ms delay on link 1. Then, we have the observed path measurement vector $\mathbf{y} = [0, 1000, 0, 0]^T$. According to (1), the estimated link metrics become $\hat{\mathbf{x}} = [0, 0, 1000, 0]^T$. Such result indicates that

1) The estimated link vector $\hat{\mathbf{x}}$ shows that no anomaly happens on link 1; therefore, the anomaly of link 1 due to attacker M_1 can be successfully concealed by such an attack strategy against the network tomography.
2) This unavoidably misleads the network operator to believe that link 3 or its end-node M_3 must have some issues.

Therefore, we call such an attack strategy *measurement integrity attack* and call link 3 or nodes M_3 a *scapegoat* in the case.

2.4 Problem Statement

From the example in Fig. 1, we can consider the measurement integrity attack as a potential attack to hide the real identities of attackers and make some legitimate nodes or links the scapegoats. Many questions can be raised concerning the feasibility of such an attack strategy in the above example: How can M_1 damage the network to launch a feasible attack? Can M_1 make other links, such as link 2, the scapegoat? Is it possible to detect or locate such an attack?

Before theoretically addressing these issues, we first introduce several necessary definitions. Considering a network $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, we define $\mathcal{V}_m \subseteq \mathcal{V}$ as the malicious nodes set (called attackers), which control a set of links $\mathcal{L}_m \subseteq \mathcal{L}$. Then the attackers can launch the attack by inconsistently inflicting damage on particular paths $\mathcal{P}_m \subseteq \mathcal{P}$, where every entry $P_i \in \mathcal{P}_m$ travels at least one link $l_j \in \mathcal{L}_m$.

According to previous definitions, we unfold our major problems into two aspects as follows.

1. **Attack Strategy**: Network tomography infers the link metrics based on (2). Therefore, in order to mislead network tomography, the routing matrix $\mathbf{R}$ is necessary to be known by attackers. However, for most networks, the path information is generally encrypted in network or higher layers, so that the attacker cannot obtain it. Therefore the first challenge is how to launch measurement integrity attacks to delay or drop packets on paths from $\mathcal{P}_m$ in a way such that another set of links $\mathcal{L}_s$ is identified as the root cause and $\mathcal{L}_s \cap \mathcal{L}_m = \varnothing$ (where $\varnothing$ denotes the empty set).

2. **Detection Strategy**: From the defenders' perspective, they know the global routing matrix $\mathbf{R}$ and the manipulated path measurement vector, but they do not know the true malicious links $\mathcal{L}_m$. Therefore, another challenge is how to detect if measurement integrity attacks exist and how to localize which links have real problems.

We assume all nodes, including norm nodes and monitors, can be malicious in $\mathcal{V}_m$, because they are not dedicated nodes with special protection, but normal nodes representing sources and destinations on measurement paths in the network. A large number of nodes are usually required to be chosen as monitors to ensure identifiability in network tomography [16], [17].

3 Attack Strategies

In this section, we formally address the measurement integrity problem. In particular, we categorize measurement integrity attacks into three basic strategies, and then formulate them and discuss their impacts.

3.1 Network Link States

The network operator uses network tomography to identify an abnormal link by checking its link metric exhibiting long delay or high loss. Under measurement integrity attacks, a normal link may be misleadingly identified as abnormal. To facilitate formulating such attacks, we first define the normal and abnormal states of a network link.

[1] Monitors do not need to enumerate all possible paths between them. They only need to choose a sufficient number of paths to ensure identifiability in network tomography (e.g., [16], [17]). Fig. 1 shows such an example with 4 paths chosen.

Definition 1 (Link States): Define the state space of a link as $\mathcal{S} = \{\textsf{normal}, \textsf{abnormal}, \textsf{uncertain}\}$. Let the state of link $l_i \in \mathcal{L}$ be a function $S : \mathcal{L} \to \mathcal{S}$ such that $S(l_i) = \textsf{abnormal}$ if l_i's link metric x_i is larger than an upper bound b_u (i.e., $x_i > b_u$), and $S(l_i) = \textsf{normal}$ if x_i is less than a lower bound b_l (i.e., $x_i < b_l$), and $S(l_i) = \textsf{uncertain}$ otherwise (i.e., when $x_i \in [b_l, b_u]$). In particular, the state $S(l_i)$ satisfies

$$S(l_i) = \begin{cases} \textsf{normal} & x_i < b_l, \\ \textsf{uncertain} & b_l \le x_i \le b_u, \\ \textsf{abnormal} & x_i > b_u. \end{cases}$$

Remark 1: The state of uncertain indicates that some links may be in an intermediate state that cannot be clearly classified to abnormal or normal. There is no standardized definition to clarify all problematic conditions in practical network diagnostics. For example, in an enterprise network, a link can be considered abnormal if the link delay is larger than a few seconds, and considered normal if the delay is tens of milliseconds (ms). However, when the link delay is hundreds of milliseconds (e.g., 150ms), it really depends on the network operation rules in the organization to decide the state of the link. As a result, we introduce the state of uncertain to accommodate this intermediate state. We also note that our three-state scenario can be easily transitioned into the two-state scenario by setting a single threshold $b = b_u = b_l$ in Definition 1.

With Definition 1, we can say that one of the goals for measurement integrity attacks is to make sure that the links associated with attackers are identified as normal; at the same time, some innocent links are, however, identified as abnormal.

3.2 Attack Manipulation Vector and Inflicted Damage

Apparently, except for generating misleading results, a major goal of attackers is to cause damage to the network. Therefore, we also need to measure the damage due to the attacks. The first thing towards measuring the attack damage is to determine what attackers can manipulate. By nature, attackers can affect any end-to-end path that goes through them, accordingly manipulating the end-to-end measurement vector observed at monitors. For example, in Fig. 1, node M_1 can obviously affect any data flow going through links 1(e.g., delaying or dropping packets).

Denote by $\mathbf{y}'$ and $\mathbf{y}$ the end-to-end measurement vectors with and without the attack, respectively. Without loss of generality, we can always write

$$\mathbf{y}' = \mathbf{y} + \mathbf{m}, \tag{4}$$

where $\mathbf{y}$ reflects the real end-to-end performance, and $\mathbf{m}$ is called the attack manipulation vector that denotes the damage (e.g., intentional delay or packet dropping ratio) inflicted by attacks over all paths. For example, when an end-to-end path has a delay of 10ms, an attacker on the path can incur an extra delay of 1000ms for every packet, making the observed end-to-end measurement 1010ms; and the extra delay of 1000ms can be controlled by the attacker and will be an entry in $\mathbf{m}$ to represent the damage to the network. Accordingly, each entry in $\mathbf{m}$ reflects the performance degradation induced by the attacker on each path in the network.

All entries in $\mathbf{m}$ should be non-negative in that attackers should not boost, but degrade the network performance, i.e., $\mathbf{m} \succeq \mathbf{0}$, where $\succeq$ means "componentwise greater than or equal to" defined in Table 1. For example, attackers can intentionally postpone forwarding packets on paths going through them, thus incurring more delay on those paths. But they are never expected to reduce the delay, because it is in contrast to the attacker's goal to damage the network and it may be technically infeasible for them to further reduce the delay at will. In addition, for the measurement paths that contain no attacker, the corresponding entries in $\mathbf{m}$ must be zero, indicating that attackers cannot manipulate the measurements on these paths. For example, in Fig. 1, attacker M_1 is not on path 4, and thus cannot manipulate the measurement of path 4. We formally define these constraints of $\mathbf{m}$ as follows.

Constraint 1 (Constraints of Attack Manipulation): The attack manipulation vector $\mathbf{m} = \{m_i\}_{i \in [1, |\mathcal{P}|]}$ satisfies (i) $\mathbf{m} \succeq \mathbf{0}$; and (ii) $m_i = 0$ when there exists no such node $v \in \mathcal{V}_m$ that is on path $P_i \in \mathcal{P}$, where $\mathcal{V}_m$ and $\mathcal{P}$ denote the sets of malicious nodes and measurement paths, respectively.

Under the Constraint 1, attackers will attempt to maximize the damage to the network. In the following, we define the damage as total performance degradation over all paths.

Definition 2 (Damage of Measurement Integrity Attack): The damage of the measurement integrity attack is measured by $\|\mathbf{m}\|_1$, i.e., the $\mathcal{L}_1$ norm of attack manipulation vector $\mathbf{m}$.

Remark 2: Definition 2 defines the damage metric of the attack as the total sum of all entries, representing the total performance degradation over all paths. The larger the value of $\|\mathbf{m}\|_1$, the more damage the attack brings. We can also change the damage metric to the average performance degradation or to any other form. For the sake of simplicity, we always use the damage metric in Definition 2 for formulation and analysis, and note that the change of the damage metric (e.g., to accommodate the metrics of packet loss or delivery ratios) is a straightforward extension in mathematical manipulations.

3.3 Partial Information

According to (2), in order to manipulate the inferred link metrics $\hat{\mathbf{x}}$, the global routing information lying in the routing matrix $\mathbf{R}$ and the original overall path measurement vector $\mathbf{y}$ should be known by attackers. For example, in Fig. 1, if M_1 knows the linear system

(3) and $\mathbf{y} = [0, 0, 0, 0]^T$, then M_1 can scapegoat link 3 by introducing $\mathbf{m} = [0, 1000, 0, 0]$. However, in practice, from the security perspective, both $\mathbf{R}$ and $\mathbf{y}$ are less likely open to every node in most networks. Therefore, before introducing the attack strategy, we need to clarify what information that attackers know while launching attacks.

Definition 3 (Partiality Factor): Attackers know the routing information and path measurement vector on paths that go through them (i.e., paths in $\mathcal{P}_m$). Specifically, for each path $P_i \in \mathcal{P}_m$, the routing information $\mathbf{r}_i$ (i.e., ith row of the routing matrix $\mathbf{R}$), and the true path measurement metric $y_i \in \mathbf{y}$ are known by attackers. The partiality factor β is defined as the ratio between the number of paths in $\mathcal{P}_m$ and the total number of paths, i.e., $\beta = |\mathcal{P}_m|/|\mathcal{P}|$.

Remark 3: The partiality factor β is a ratio indicating how much information that attackers know. $\beta = 1$ denotes attackers completely know $\mathbf{R}$ and $\mathbf{y}$, and $0 < \beta < 1$ means attackers only know several rows of $\mathbf{R}$ and associated several entries of $\mathbf{y}$. In fact, β is also positively related to the number of attackers in a network. For example, $\beta = 1$ denotes attackers are present on all paths in a network, and $\beta = 0$ means there is no attacker in the network.

Remark 4: In many networks, attackers may acquire the path measurement vector and the routing information of paths that go through them. For example, in AODV, the routing information is available during the routing discover process, and the path measurement vector can be obtained if the attacker is the source or destination node of a path.

According to Definition 3, to clearly model the partiality, we split $\mathbf{R}$ and $\mathbf{y}$ into two parts, i.e., $\mathbf{R} = [\mathbf{R}_d^T, \mathbf{R}_r^T]^T$ and $\mathbf{y} = [\mathbf{y}_d^T, \mathbf{y}_r^T]^T$, where $\mathbf{R}_d$ and $\mathbf{y}_d$ denote the parts that attackers know, and $\mathbf{R}_r$ and $\mathbf{y}_r$ are unknown to attackers. Similarly, according to Constraint 1, we also divide the attack manipulation vector as $\mathbf{m} = [\mathbf{m}_d^T, \mathbf{0}^T]^T$, where $\mathbf{m}_d^T$ denotes the inflicted damage on paths in $\mathcal{P}_m$. Then, the linear system (4) can be written as

$$\mathbf{y}' = \begin{bmatrix}\mathbf{y}_d \\ \mathbf{y}_r\end{bmatrix} + \mathbf{m} = \begin{bmatrix}\mathbf{R}_d \\ \mathbf{R}_r\end{bmatrix}\mathbf{x} + \begin{bmatrix}\mathbf{m}_d \\ \mathbf{0}\end{bmatrix}. \tag{5}$$

Attackers have no knowledge on entries in $\mathbf{R}_r$ and $\mathbf{y}_r$, therefore we consider $\mathbf{R}_r$ and $\mathbf{y}_r$ as a random matrix and a random vector, respectively. If the partiality factor $\beta = 1$, then $\mathbf{y}_r$ and $\mathbf{R}_r$ will vanish and $\mathbf{m}_d = \mathbf{m}$, indicating attackers know and can control all paths. If $\beta = 0$, $\mathbf{m}_d$ will vanish, meaning that this network cannot be damaged by attackers, therefore in order to investigate the malicious behaviors, in this paper, we only consider the scenario $0 < \beta \leq 1$.

3.4 Attack Strategy

Given the partial knowledge $\mathbf{R}_d$ and $\mathbf{y}_d$, one major goal of attackers is to find an attack manipulation vector $\mathbf{m}$, such that inferred link metrics $\hat{\mathbf{x}}$ can scapegoat victim links as the root cause. However, the unknown information in $\mathbf{R}_r$ and $\mathbf{y}_r$ renders uncertainties for attackers to find $\mathbf{m}$. Therefore, we need a deterministic rule to guide attackers to address such uncertainties.

According to Definition 1, the scapegoating purpose represents that the estimated metric $\hat{x}_i$ for link $l_i \in \mathcal{L}$ must meet certain conditions to be in normal, abnormal, or uncertain state. This means that we can write the goal as

$$\mathbf{s}_l \preceq \hat{\mathbf{x}} \preceq \mathbf{s}_u, \tag{6}$$

where $\mathbf{s}_u$ and $\mathbf{s}_l$ are called the upper and lower bound vectors. By controlling $\mathbf{s}_u$ and $\mathbf{s}_l$, we can accommodate various scapegoating purposes. Inserting (2) and (4) into (6), we have

$$\mathbf{s}_l \preceq (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T(\mathbf{y} + \mathbf{m}) \preceq \mathbf{s}_u. \tag{7}$$

The normal link delay in a network is usually small (e.g., several or tens of milliseconds). By contrast, an attacker should significantly delay packets (e.g., by more than thousands of milliseconds) to cause damage to the network. Therefore, the true path measurement vector $\mathbf{y}$ should be of lesser magnitude than the attack manipulation vector $\mathbf{m}$ (i.e., $\mathbf{y} + \mathbf{m} \approx \mathbf{m}$). Then (7) can be approximated by

$$\begin{aligned}&(\mathbf{R}_d{}^T\mathbf{R}_d + \mathbf{R}_r{}^T\mathbf{R}_r)\mathbf{s}_l \\ &\preceq \begin{bmatrix}\mathbf{R}_d^T & \mathbf{R}_r^T\end{bmatrix}\begin{bmatrix}\mathbf{m}_d \\ \mathbf{0}\end{bmatrix} \preceq (\mathbf{R}_d{}^T\mathbf{R}_d + \mathbf{R}_r{}^T\mathbf{R}_r)\mathbf{s}_u.\end{aligned} \tag{8}$$

Obviously, if attackers know the routing matrix $\mathbf{R}$ completely, $\mathbf{m}_d$ can be solved from (8) through standard linear programming [27]. However, with the random matrix $\mathbf{R}_r$, it is infeasible to solve $\mathbf{m}_d$ directly. Therefore, we need a constraint to convert such random matrix $\mathbf{R}_r$ to a deterministic one. By considering the worst case, we have the following constraint.

Constraint 2 (Constraints of Partial Information): Given the upper bound $\mathbf{s}_u$ and lower bound $\mathbf{s}_l$ of $\hat{\mathbf{x}}$, the partial routing matrix $\mathbf{R}_d$ and vector $\mathbf{m}_d$ in the attack manipulation vector $\mathbf{m}$ satisfies

$$\begin{aligned}&(\mathbf{R}_d{}^T\mathbf{R}_d + \mathbf{R}_r^{+}{}^T\mathbf{R}_r^{+})\mathbf{s}_l \\ &\preceq \mathbf{R}_d^T\mathbf{m}_d \preceq (\mathbf{R}_d{}^T\mathbf{R}_d + \mathbf{R}_r^{-}{}^T\mathbf{R}_r^{-})\mathbf{s}_u,\end{aligned} \tag{9}$$

where

1) $\mathbf{R}_r^+ = \underset{\mathbf{R}_r}{\arg\sup}(\mathbf{R}_r{}^T\mathbf{R}_r\mathbf{s}_l)$, and $\mathbf{R}_r^+$ does not contain two identical rows and all 0 row.
2) $\mathbf{R}_r^- = \underset{\mathbf{R}_r}{\arg\inf}(\mathbf{R}_r{}^T\mathbf{R}_r\mathbf{s}_u)$, and $\mathbf{R}_r^-$ does not contain two identical rows and all 0 row.

Remark 5: Constraint 2 converts the random matrix $\mathbf{R}_r$ into deterministic matrices $\mathbf{R}_r^+$ and $\mathbf{R}_r^-$, which maximize and minimize $\mathbf{R}_r{}^T\mathbf{R}_r\mathbf{s}_l$, respectively. Constraint 2 can be considered as the worst case of (8) because it shrinks the solution space of $\mathbf{m}_d$, i.e., it is easy to know that for any solution $\mathbf{m}_d^*$ satisfying (9), it must also satisfy (8).

3.5 Formulation of Measurement Integrity Attacks

Measurement integrity attacks aim to bring damage to a network, and at the same time hide the attacker-controlled link set $\mathcal{L}_m$ but expose another set of victim links $\mathcal{L}_s$ as scapegoats to network tomography. The basic idea to implement measurement integrity attacks is that attackers cooperatively inflict damage on paths which contain victims, and do nothing on other paths. Based on this idea, attackers can choose different strategies to launch the attacks. Specifically, we consider three basic strategies: (i) chosen-victim attacks, where the victim link set $\mathcal{L}_s$ is already chosen and targeted, (ii) maximum-damage attacks, where attackers aim at finding the best victim link set $\mathcal{L}_s$ to maximize their damage, (iii) obfuscation, where attackers attempt to make network tomography show no evident performance outliers but uniform degradation over a substantial number of links.

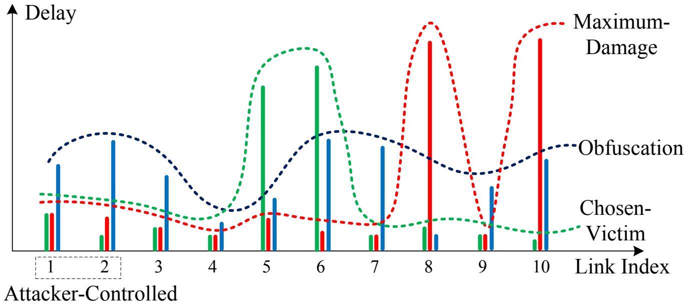


Fig. 2. Examples of link metrics under tomography for chosen-victim scapegoating, maximum-damage scapegoating, and obfuscation.

Fig. 2 shows an illustrative example of how different attack strategies affect the link delay metrics obtained by network tomography. In Fig. 2, solid lines represent the values of end-to-end delay metrics and each dotted line denotes the envelope of the solid line under the same scapegoating strategy. We see from Fig. 2 that there are 10 links, and links 1 and 2 are controlled by attackers. Under chosen-victim scapegoating, the attackers choose links 5 and 6 to be scapegoats that exhibit much higher delays than other links. Under maximum-damage scapegoating, the attackers found that links 8 and 10 can be the scapegoats with highest delays. Under obfuscation, the attackers can make most links exhibit similarly delays, which can confuse the network operator to find which links are truly problematic.

In the following, we mathematically formulate these attack strategies.

3.5.1 Chosen-Victim Scapegoating

When the victim set $\mathcal{L}_s$ is already given, this strategy can be formulated as choosing the best attack manipulation vector $\mathbf{m}$ to maximize the attack damage, at the same time satisfying the constraints for $\mathbf{m}$, $\mathcal{L}_m$, and $\mathcal{L}_s$. According to Constraint 1 and 2 and Definitions 1 and 2, we can formulate this basic scapegoating strategy as

$$\begin{aligned} \underset{\mathbf{m}}{\text{maximize}} \quad & \|\mathbf{m}\|_1 , && (10)\\ \text{subject to} \quad & \mathbf{m} \text{ satisfies Constraint 1 and 2,} && \\ & S(l) = \text{normal},\ \forall\, l \in \mathcal{L}_m, && (11)\\ & S(l) = \text{abnormal},\ \forall\, l \in \mathcal{L}_s, && (12)\\ & \mathcal{L}_m \cap \mathcal{L}_s = \varnothing, && (13) \end{aligned}$$

where constraints (11) and (12) mean that all links associated with the attackers should appear normal, and all links in the victim set should be abnormal, respectively. These two together, combined with constraint (13), achieve the goal of scapegoating under network tomography.

3.5.2 Maximum-Damage Scapegoating

If the attackers aim to bring maximum damage to the network, they may do so by searching the best victim set in the set of all links. Therefore, maximum-damage scapegoating can be written as

$$\begin{aligned} \underset{\mathbf{m}, \mathcal{L}_s \subset \mathcal{L}}{\text{maximize}} \quad & \|\mathbf{m}\|_1 , && (14)\\ \text{subject to} \quad & \mathbf{m} \text{ satisfies Constraint 1 and 2,} && \\ & \text{Constraints in (11), (12), and (13).} && \end{aligned}$$

3.5.3 Obfuscation

Different from the chosen-victim and maximum-damage attacks, the idea behind obfuscation is to make every link look mostly similar without evident outliers. Obfuscation does not necessarily lead to a unique strategy. As long as a strategy makes a substantial number of link metrics look approximately similar, and at the same time incurs damage to the network, it should be considered as a successful obfuscation one. We leverage the state of uncertain in Definition 1 to define obfuscation as follows.

$$\begin{aligned} \underset{\mathbf{m}, \mathcal{L}_s \subset \mathcal{L}}{\text{maximize}} \quad & \|\mathbf{m}\|_1 , && (15)\\ \text{subject to} \quad & \mathbf{m} \text{ satisfies Constraint 1 and 2,} && \\ & S(l) = \text{uncertain},\ \forall\, l \in \mathcal{L}_o = \mathcal{L}_s \cup \mathcal{L}_m, && (16)\\ & \mathcal{L}_s \neq \varnothing, |\mathcal{L}_o| > \gamma |\mathcal{L}| && (17) \end{aligned}$$

where $\mathcal{L}_s$ is the set of victim links that attackers want to find such that any link $l \in \mathcal{L}_o$ is manipulated under network tomography to be in the uncertain state defined in (16). γ is a predefined threshold ratio indicating the lower bound of the number of infected links. As we have mentioned, the uncertain state represents an intermediate state, in which a link cannot be clearly classified into either normal or abnormal. Hence, a substantial number of links (i.e., more than $\gamma|\mathcal{L}|$) in the uncertain state result in obfuscation.

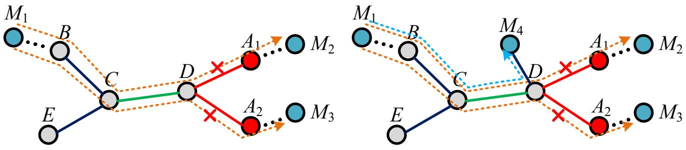
Given these formally defined basic strategies, attackers are able to launch scapegoating attacks against network tomography to maximize the damage, make scapegoats, or obfuscate the network operator. In addition, attackers may also develop more sophisticated strategies based upon these three ones.

4 Feasibility and Detectability

After we formulate measurement integrity attack strategies, two questions naturally follow: (i) Whether these attacks are indeed feasible (i.e., whether feasible solutions exist in the optimization-based strategies)? (ii) Can we detect or locate such an attack if it is successfully launched? In this section, we answer these two questions by first analyzing the feasibility of the attack, then describing how to detect it. The method to locate attacks is discussed in Section 5.

4.1 Feasibility Analysis

Whether an attack is feasible depends on the network connectivity, selections of measurement paths, and where attackers are. Consider a simple example in Fig. 3(a): Attackers A_1 and A_2 aim to manipulate the end-to-end measurements to scapegoat the link between nodes C and D. They should be able to succeed if they are on all the measurement paths that go through the link between C and D. We say it is a *perfect cut* case, in which for any measurement path $P \in \mathcal{P}$ containing a victim link, there always exists a malicious node $v \in \mathcal{V}_m$ present on that path P. Fig. 3(b) illustrates an *imperfect cut* case, in which the path $M_1 \to B \to\ C \to\ D \to M_4$ contains neither A_1 nor A_2.



(a) Perfect Cut (b) Imperfect Cut

Fig. 3. Perfect and imperfect cuts by attackers A_1 and A_2 to scapegoat the link between nodes C and D on the measurement paths between monitors.

4.1.1 Perfect Cut

We show in the following that a perfect cut always leads to a successful attack in any strategy.

Theorem 1 (Feasibility under Perfect Cut): A measurement integrity attack is always feasible if the set of malicious nodes $\mathcal{V}_m$ can perfectly cut the set of victim links $\mathcal{L}_s$ from all measurements paths.

Proof: We do not need to prove the feasibility of all three strategies because the maximum-damage scapegoating (14) must be feasible if chosen-victim one (10) is feasible. Then, we write (10) and (14) into a generic form, which is (6), i.e., $\mathbf{s}_l \preceq \hat{\mathbf{x}} \preceq \mathbf{s}_u$,. By adjusting s_u and s_l, we can accommodate either chosen-victim scapegoating or obfuscation because constraints (11), (12) and (16) indicate the estimated $\hat{\mathbf{x}}$ must meet certain conditions.

Then we need to show that for a given manipulated metric vector $\hat{\mathbf{x}}^*$ satisfying (6), there exists a resultant vector $\mathbf{m}^*$ that meets Constraints 1 and 2. Because Constraint 2 is derived from (6) in Section 3.4. According to (7) and (8), it is clear that $\mathbf{m}^*$ satisfies Constraint 2. Thus we only need to show the proof under Constraint 1. The proof to show $\mathbf{m}^*$ satisfies Constraint 1 can be found in our conference version [1]. □

4.1.2 Imperfect Cut

If attackers only form an imperfect cut of the victim links, the formulation of an attack strategy may not always yield a feasible solution, which depends on specific network settings. We are interested in understanding the attack success probability under generic random assumptions (i.e., we do not use specific distribution models such as power-law network connectivity, but only assume that network connectivity, placement of monitors, and selection of measurement paths are random in the network). We show that it increases with the increasing of the number of measurement paths that include at least one victim link and at least one attacker.

Theorem 2 (Attack Success Probability under Imperfect Cut): The success probability of a measurement integrity attack is defined as the probability that an attack strategy yields a feasible solution. Under generic random assumptions, the success probability is an increasing function of the number of measurement paths that include at least one victim link and at least one attacker.

Proof: The proof of this theorem can be found in our conference version [1]. □

4.2 Detecting Measurement Integrity Attacks

We have analyzed the feasibility of measurement integrity attacks. If an attack is successfully launched, we should never trust the result obtained by network tomography. It is necessary to know how to detect such an attack in a network. Our insight is that attackers have to manipulate packet delivery in certain directions to make scapegoating possible in the network. This means that if we verify the estimated link metric vector $\hat{\mathbf{x}}$, which can be obtained by (2), with observed measurement vector $\mathbf{y}'$ in all entries, it is likely to observe the inconsistency under the measurement model (1) in the presence of a measurement integrity attack. In other words, verifying $\hat{\mathbf{x}}$ and $\mathbf{y}'$ according to (1) results in our detection method

$$\text{scapegoating} \begin{cases} \text{exists,} & \text{if } \mathbf{R}\hat{\mathbf{x}} \neq \mathbf{y}', \\ \text{does not exist,} & \text{if } \mathbf{R}\hat{\mathbf{x}} = \mathbf{y}'. \end{cases} \tag{18}$$

with the following detectability.

Theorem 3 (Detectability): Under the detection mechanism (18), a measurement integrity attack is undetectable if attackers $\mathcal{V}_m$ can perfectly cut victim links $\mathcal{L}_s$ from measurement paths or $\mathbf{R}$ is a square matrix; and is detectable otherwise.

Proof: The proof of this theorem can be found in our conference version [1]. □

Remark 6: Theorem 3 shows that if attackers $\mathcal{V}_m$ can perfectly cut victim links from measurement paths, there is no way to detect them based on the inconsistency check. This is intuitively true. For example, in Fig. 3(a), attackers A_1 and A_2 cut the victim link between nodes C and D completely from the measurement paths $M_1 \rightarrow M_2$ and $M_1 \rightarrow M_3$. Any information about the victim link is from these two paths whose measurements can be surely manipulated by the attackers to evade the detection.

Remark 7: In practice, even when there is no attack, $\mathbf{R}\hat{\mathbf{x}}$ may not exactly equal to $\mathbf{y}'$ in (18) due to randomness in packet delivery and measurement error. Therefore, the attack detection can be slightly modified to test $\|\mathbf{R}\hat{\mathbf{x}} - \mathbf{y}'\|_1 > \alpha$, where α is a given threshold that can be empirically determined.

5 Locatability Analysis

According to the detection mechanism (18), if an attack is successfully launched and it is also detected, the inferred link performance $\hat{\mathbf{x}}$ obtained by network tomography becomes untrusted. Therefore, it is necessary to know which nodes or links are the attackers. In this section, we discuss how to locate the real attackers in the network. We first present the design motivation, then present the method of locating attacks.

5.1 Motivation and Example

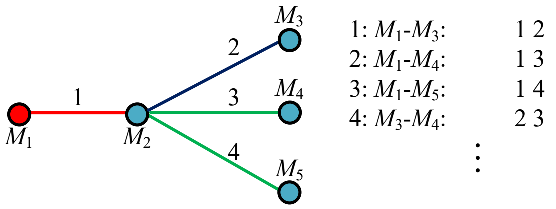


Fig. 4. A simple network consisting of 5 nodes and 4 links, in which all nodes are monitors and M_1 is malicious.

The key idea of the measurement integrity attack is that attackers only damage the paths that contain victim links and do nothing to other paths. Therefore, a malicious link used by attackers to cause damages should be present on multiple paths, i.e., some of them contain victim links and others do not. However, if the link controlled by a attacker is the only shared link in a network, then the only explanation for the inconsistency in (18) is that this shared link is malicious, because it is the only link that can really inflict the traffic differentiation among different paths.

For example, in a simple network consisting of 5 nodes and 4 links (shown in Fig. 4), we consider three paths (i.e., path 1: $M_1 \rightarrow M_3$, path 2: $M_1 \rightarrow M_4$, and path 3: $M_1 \rightarrow M_5$), the only shared link among them is link 1. Assume link 1 is malicious and controlled by attacker M_1, aiming to scapegoat links 3 and 4. Suppose if the presence of an attacker is detected among these three paths, we can pinpoint that link 1 is malicious. However, according to Theorem 3, only using paths 1-3 is insufficient to detect the attacker link 1 since link 1 can perfectly cut both links 3 and 4.

In order to detect the attacker, knowledge of extra paths is needed. Our design is based on the concept of path pairs in network neutrality inference [28]. A path pair $\{P_i, P_j\}$ is a single path which includes all links traveled by at least one of path i or path j. For example, path pair $\{P_2, P_3\}$ is formed by links 1, 3 and 4. In the real world, this path pair is measurable if M_2 is a monitor such that we can measure link 4 and path 2 in a same time period and then combine them together. Note that we are able to detect the attack by using other normal paths, such as path 4: $M_3 \rightarrow M_2 \rightarrow M_4$, however, if so, we cannot put the blames only on the link 1.

Now we use the path pair $\{P_2, P_3\}$, combined with paths 1-3 to locate link 1. Considering the existence of the attacker, the measurement model (1) can be expressed as

$$\begin{aligned} \text{Path } 1 : y_1' &= x_1 + x_2 \\ \text{Path } 2 : y_2' &= x_1 + x_3 \\ \text{Path } 3 : y_3' &= x_1 + x_4 \\ \text{Path Pair } \{P_2, P_3\} : y_4' &= x_1 + x_3 + x_4. \end{aligned} \tag{19}$$

Without loss of generality, and to clearly show the inconsistency, we assume the network, shown in Fig. 4, is congestion free, thus delays only come from the attacker. To scapegoat links 3 and 4, link 1 introduces extra delays (e.g., 1000ms) on paths 2 and 3. Then the measurements of these four paths are $\mathbf{y}' = [0, 1000, 1000, 1000]^T$. It is easy to observe that a contradiction happens among these measurements: y_1' indicates that $x_1 = 0$, whereas y_2', y_3' and y_4' indicate $x_1 = 1000$ and $x_3 = x_4 = 0$. If we apply the detection mechanism (18) into the system (19), this contradiction will definitely lead to the inconsistency. Then link 1 can be located because it is the only reason for the inconsistency under network neutrality inference.

From this example, we have three observations to locate measurement integrity attacks leveraging network neutrality inference:

1) To locate the malicious link, we should be able to find a path set, in which the malicious link is the only shared link. For example, link 1 is the only shared link among paths 1-3.
2) Path pairs formed by the paths in the path set should be measurable. For example, the path pair $\{P_2, P_3\}$ is measurable and it is necessary because it guarantees the unique solution to the last three equations.
3) If the shared link is malicious, to observe the contradiction, at least two paths should contain different victim links, and at lease one path should not contain a victim link. For example, path 1 does not contain victim links, and paths 2 and 3 contain links 3 and 4 respectively. Then, we can use path 2, path 3 and their path pair $\{P_2, P_3\}$ to conclude link 1 is malicious, and use path 1 to contradict that link 1 is congestion free.

The basic idea in the previous example is to use inconsistency to locate a malicious link. However, inconsistency may occur regardless of the presence of an attacker. There are two types of inconsistency: (i) inconsistency resulted by measurement noise; and (ii) inconsistency incurred by attackers. For the first type, even when there is no attacker in a network, inconsistency is still possible to occur since random measurement noise is inevitable while probing the network. However, this type of inconsistency is slight, so it can be solved by setting a threshold of measurement according to [28].

For the second type, if the inconsistency is larger than the threshold, we consider that the inconsistency is resulted by attackers. According to Theorem 3, once inconsistency occurs, we can detect attacks happened in the network. But not all inconsistencies can be used to locate malicious links. Therefore, in the following, we elaborate how to locate malicious links when attackers exist. Note that the locating process is triggered only when an attack is detected by (18).

5.2 Locating Measurement Integrity Attacks

From the example in Fig. 4, we iteratively inspect the maliciousness of each link throughout the entire network to locate attackers. For any link $l_\alpha \in \mathcal{L}$ in a network $\mathcal{G}$, to judge whether link l_α is malicious, we first create a path set $\mathcal{P}_\alpha \subseteq \mathcal{P}$ based on the link l_α. Then, we use $\mathcal{P}_\alpha$ to form a new system, such as the example shown in (19), and apply the detection mechanism (18) to this new system, to check if link l_α is malicious. Therefore, our attack-locating mechanism consists of two parts: (i) path set generation, and (ii) maliciousness discrimination. In the following, we elaborate each part in detail.

5.2.1 Path Set Generation

Directly applying the detection mechanism (18) to the overall path set $\mathcal{P}$ will only yield a binary result to show whether attacks exist or not, and cannot locate which links or nodes are really malicious. The purpose of creating a new path set is that we can attribute the inconsistency in (18) to a certain link. Specifically, to locate link l_α, the path set $\mathcal{P}_\alpha$ can be formed as follows:

1) Find all path pairs $\{P_i, P_j\}$ where $P_i, P_j \in \mathcal{P}$, such that the shared link between P_i and P_j is l_α.
2) Add P_i and P_j and their corresponding path pair $\{P_i, P_j\}$ to the path set $\mathcal{P}_\alpha$, i.e.,

$$\mathcal{P}_\alpha = \mathcal{P}_\alpha \cup \{P_i, P_j, \{P_i, P_j\}\}.$$

5.2.2 Maliciousness Discrimination

After obtaining the path set $\mathcal{P}_\alpha$, we can only focus on inspecting the measurements of paths in $\mathcal{P}_\alpha$. Then we write the measurement model as

$$\mathbf{y}' = \mathbf{R}_\alpha \mathbf{x}, \tag{20}$$

where $\mathbf{R}_\alpha$ is the routing matrix formed with respect to the path set $\mathcal{P}_\alpha$.

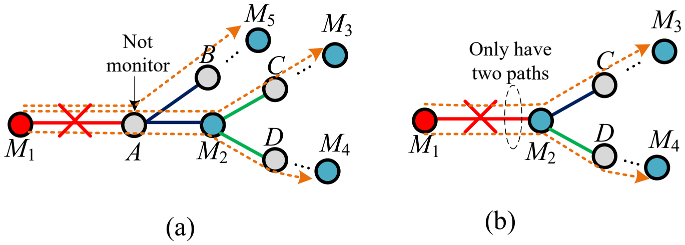


Fig. 5. Examples of unlocalizable scenarios where (a) violates the condition 1) in Theorem 5 and (b) includes only two paths.

Theorem 4 (Maliciousness): Given a link l_α and its measurement model (20), the link l_α is malicious if scapegoating exists in (20) based on the detection mechanism (18).

Proof: If scapegoating is detected by the detection mechanism (18) in (20), it is clear that $\mathbf{R}_\alpha \hat{\mathbf{x}} \neq \mathbf{y}'$. This indicates that directly solving (20) as a linear equation system will yield no solution. Then, according to Lemma 2 in [28], link l_α must have different link rates for different paths. Therefore, link l_α is malicious. □

5.3 Locatability

In the following, we present the network condition to indicate when an attack link can be located in the network.

Theorem 5 (Locatability): For an attack link l_α, and its path set $\mathcal{P}_\alpha$, if the following conditions hold:

1) for any path $P_i \in \mathcal{P}_\alpha$, nodes between link l_α and other links must be monitors;
2) the path set $\mathcal{P}_\alpha$ should contain at least three individual paths and two path pairs, in which both paths in one path pair contain victim links and the other path pair must contain at least one path which does not travel any victim links;

then l_α is localizable.

Proof: First, if condition 1) holds, it is easy to verify that all path pairs in the path set $\mathcal{P}_\alpha$ are measurable. Then in the following, we prove this theorem by showing that if condition 2) holds, inconsistency exists in the measurement model (20). Considering an arbitrary path pair $\{P_i, P_j\} \in \mathcal{P}_\alpha$, the measurable model can be written as

$$\begin{aligned} \text{Path } P_i &: y'_i = x_\alpha + x^*_{i\alpha} \\ \text{Path } P_j &: y'_j = x_\alpha + x^*_{j\alpha} \\ \text{Path Pair } \{P_i, P_j\} &: y'_{ij} = x_\alpha + x^*_{i\alpha} + x^*_{j\alpha}, \end{aligned} \tag{21}$$

where $x^*_{n\alpha}$ is the adjacent link of x_α on the path P_n, which is resulted from deleting the link l_α from path P_n, and $n \in \{i, j\}$. It is easy to know that (21) has the unique solution

$$x_\alpha = y'_i + y'_j - y'_{ij}. \tag{22}$$

Then we consider three scenarios: (i) both P_i and P_j contain victim links; (ii) only one path (e.g., P_i) contains victim links; and (iii) neither P_i nor P_j includes victim links.

The ground truth metric $\tilde{x}_\alpha$ of link l_α can always be the solution to scenarios (ii) and (iii). It is easy to verify for scenario (iii) since there is no extra damage on all paths. For scenario (ii), $\tilde{x}_\alpha$ still can be the solution because the extra damage is inflicted to the adjacent link $x^*_{i\alpha}$. The proof of Lemma 3 in [28] shows that the solution to scneario (i) is exactly distinct from the ground truth metric $\tilde{x}_\alpha$. Therefore, if condition 2) holds, we get different solutions from different paths, thus the inconsistency exists and l_α can be located. □

Remark 8: Theorem 5 shows two sufficient conditions to locate a malicious link. The violation of any one or both of them will lead to the failure of locating. Fig. 5 shows two typical examples in which the malicious link is not locatable. In Fig. 5, nodes $M_1 - M_5$ are monitors, where M_1 is the attacker aiming to scapegoat links between M_2 and D, M_2 and C (only exists in Fig. 5(a)). In Fig. 5(a), the malicious link does not satisfy condition 1 in Theorem 5 because node A is not a monitor and we cannot add the path pair between path $M_1 \to M_3$ and path $M_1 \to M_5$ to the path set. One question may raise if node A is a monitor: Can we directly locate the malicious link by using M_1 and A without following the previous attack-locating mechanism? The answer is no because the path between M_1 and A does not contain any intended victim by the attacker. If we directly send probe packets on this path, the obtained result will not be affected as the attacker does not manipulate the packets traversing. In Fig. 5(b), there are only two paths, thus the path set formed by this network does not satisfy condition 2), which needs at least two different path pairs.

6 Experimental Evaluation

In this section, we use simulation experiments to evaluate the feasibility of measurement integrity attacks and effectiveness of attack detection and locating methods based on real-world and simulated network topologies.

6.1 Experimental Setups

6.1.1 Network Topology

We consider two types of network scenarios.

- Wireline networks. We use the Rocketfuel datasets [29] as the topologies for wireline networks. Rocketfuel models the topologies of autonomous systems of Internet Service Providers (ISPs), such as AT&T and Ebone. In the following, we only show the results from the AS1221 system, consisting of 108 nodes and 152 links, due to similar experimental results.
- Wireless networks. We use the random geometric graph to generate wireless network topologies because it has been widely used to model multi-hop wireless networks (e.g, [30], [31]). We adopt the extended network generation mode, and randomly distribute 100 nodes on region $[0, \sqrt{100/\lambda}]^2$ according to node density $\lambda = 5$ such that each node has 5 neighbors on average.

6.1.2 Parameter Setting

In experiments, we use delay as the performance metric. There is a routine traffic on each link with random delay performance from 1ms to 20ms. We consider a link normal if its delay is less than 100ms, and abnormal if the delay is greater than 800ms.

The objective of malicious nodes is to delay packets going through them as much as possible, and at the same time make network tomography yield a misleading result. For practical considerations, we also impose a limit on attackers that they should not delay the delivery of a packet on a measurement path for more than 2000ms.

We choose monitors and measurement paths according to a random selection algorithm based on the minimum monitor placement rule in [17]. We also randomly select nodes to be malicious in a network.

For the obfuscation attack, we set the default threshold $\gamma = 10\%$, i.e., the obfuscation attack is successful if more than 10% of total number of links are in the uncertain state.

6.2 Feasibility

In our experiments, in order to show the impact of measurement integrity attacks, we define the attack success probability as the ratio between the number of successful attacks and the total number of runs for a network topology. Then, in the following, we measure the feasibility of chosen-victim scapegoating, maximum-damage scapegoating and obfuscation by changing the partiality factor β and the obfuscation threshold γ in both wireline and wireless networks.

6.2.1 Varying the Partiality Factor

A straightforward way to show the feasibility is to measure the success probability as a function of the partiality factor β in the network. This is because, as shown in Theorems 1 and 2, an essential condition for scapegoating is the number of paths in $|\mathcal{P}_m|$, which is closely related to the partiality factor β. It is obvious that the $\beta = 100\%$ if attackers present on all measurement paths, which can perfect cut any victim link.

Fig. 6 depicts the success probabilities of three attack strategies in wireline network. It is easy to find that the success probability increases as the partiality factor β increases. For example, when β goes from 70% to 80%, the success probability increases accordingly from 28% to 67% for the chosen-victim scapegoating as shown in Fig. 6. When $\beta = 1$, the success probability becomes 100% for all attack strategies since attackers can perfectly cut all victim links. We also notice that obfuscation is less likely to succeed, compared with chosen-victim and maximum-damage scapegoating as it has to manipulate

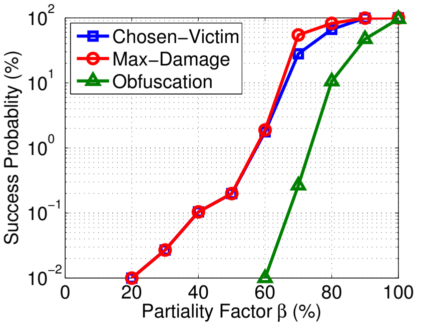


Fig. 6. The success probabilities versus partiality factor β for three types attackers in wireline network.

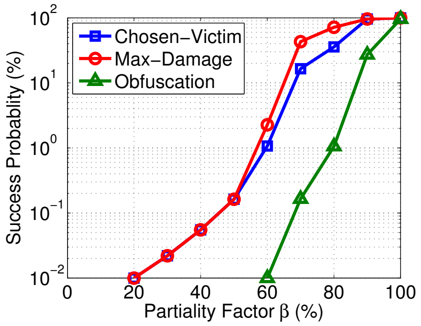


Fig. 7. The success probabilities versus partiality factor β for three types attackers in wireless network.

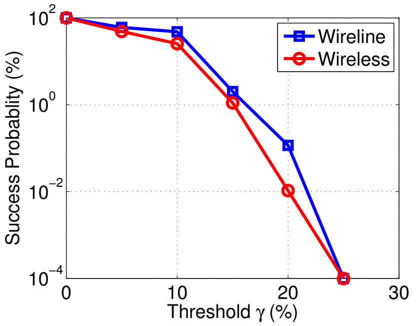


Fig. 8. The success probabilities versus the obfuscation threshold γ in both wireline and wireless networks.

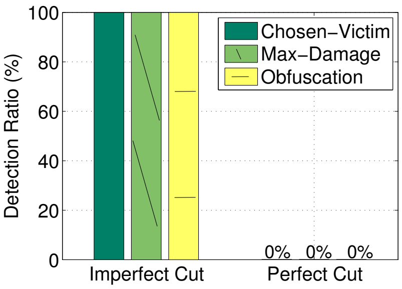


Fig. 9. The detection ratios of chosen-victim, maximum-damage and obfuscation attackers with perfect and imperfect cuts.

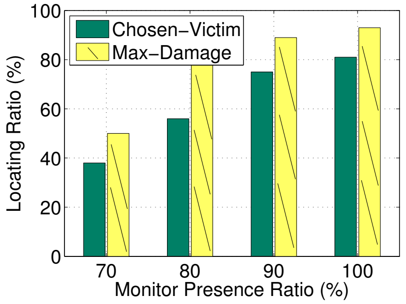


Fig. 10. The localization ratios versus the monitor presence ratio for chosen-victim attackers, maximum-damage attackers.

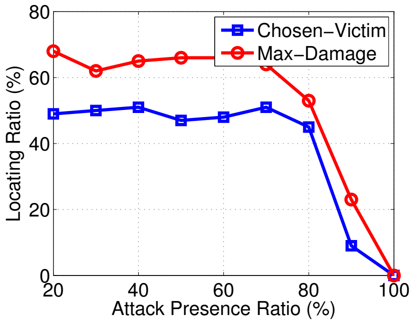


Fig. 11. The locating ratios versus the attack presence ratio for chosen-victim attackers, maximum-damage attackers.

a number of victim links. In addition, maximum-damage attacks are always more likely than chosen-victim attacks. This is because the attacker does not specifically target a given victim; as long as it can find such a victim among all the nodes, it will be successful.

Fig. 7 shows the success probabilities under the wireless network topology. We can see from Fig. 7 that when β is less than a threshold, scapegoating is unlikely to succeed, because attackers know too little information about the network to launch the attacks. For example, the obfuscation always has 0 success probability when $\beta \leq 60\%$.

6.2.2 *Varying the Threshold of Obfuscation*

For obfuscation, the success probability is also related to the threshold γ since attackers must make at least $\gamma|\mathcal{L}|$ victim links exhibit the uncertain status to be considered successful. Therefore, we also evaluate how γ can affect the success probability.

Fig. 8 depicts the success probabilities of obfuscation in both wireline and wireless topologies when $\beta = 90$. We can see from both types of networks, the success probability decreases as γ increases. This is because a larger γ means that attackers need to affect more links at the same time, which is less likely to succeed. In addition, we notice that obfuscation is less successful in the wireless topology. For example, when $\gamma = 10\%$, the success probabilities are 48% and 60% in wireless and wireline networks, respectively. This is because the wireless topology is sparser and our monitor placement algorithm results in shorter measurement paths, which are more difficult to be affected by attackers from our observations in experiments.

6.3 Detection

We then use the detection method proposed in Section 4.2 to detect measurement integrity attacks. According to Theorem 3, there is no way for the method to detect an attack if attackers perfectly cut a victim. We separate experiments into the perfect cut and imperfect cut cases. We set the threshold $\alpha = 200$ms in all experiments.

Fig. 9 shows the detection ratios over all three attack strategies in the perfect cut and imperfect cut cases, respectively. From Fig. 9, the detection ratio in the presence of all three attacks is 100% when attackers can perfectly cut victim links, and 0% otherwise, which verifies the theoretical predictions in Theorem 3. We also find that the detection method yields no false alarm in all attack detection experiments.

6.4 Locating Attacks

We locate malicious links by leveraging the method in Section 5.2. According to the condition 1 in Theorem 5, a malicious link is not locatable if both end nodes of the link are not monitors. Therefore, locatability of a link in a network is directly related to the monitor presence ratio. We conduct our experiments by showing the relationship between the monitor presence ratio and the locating ratio, which is defined as the probability that a malicious link can be located.

Fig. 10 shows the locating ratio under different attack strategies, where 5% nodes are malicious. In Fig. 10, we see that the locating ratio increases when we place more monitors, since more paths can meet the condition 1 of Theorem 5. However, even when all nodes are monitors, we still cannot guarantee to locate every malicious link since locating attack links is also related to the network topology. For example, the case shown in Fig 5(b) is not locatable even if all nodes are monitors. In addition, the locating ratio of the maximum-damage scapegoating is larger than the chosen-victim attacks because the size of the victim link set $\mathcal{L}_s$ of the maximum-damage scapegoating is usually larger than the size of $\mathcal{L}_s$ for the chosen-victim attacks, thus providing more chances to satisfy the condition 2. Note that we do not evaluate the locating ratio for the obfuscation strategy because the success probability of the obfuscation attack is very low with 5% attackers. In other words, it is very unlikely to launch a feasible obfuscation attack with such a limited number of attackers, thus we do not need to locate it.

Fig. 11 shows the relationship between the locating ratio and the attack presence ratio (defined as the ratio of the number of measurement paths including at least one victim and at least one attacker over the number of total measurement paths including any victim) where 75% nodes are monitors. We can see a slight fluctuation of locating ratios when the attack presence ratio increases from 20% to 80%. But when the attack presence ratio reaches 90%, the locating ratio decreases sharply, since almost all links are controlled by attackers, and the numbers of normal links and victim links decrease dramatically, which are necessary to locate attack links.

7 Discussions and Future Work

In this section, we discuss our results associated with the feasibility and defense of measurement integrity attacks, as well as the potential impacts on other related work. Then we provide our future work.

To launch measurement integrity attacks, the attackers must have the information about the measurement paths, which the network operator can definitely attempt to hide. For example, the operator can avoid publishing such information or avoid using some protocols containing path information, such as AODV routing for wireless networks, to prevent attackers from inferring such information from probe packets in the network. This can constitute the first line of defense. Nevertheless, from a security point of view, it should not be assumed that attackers can never get such information. Moreover, measurement integrity attacks do pose a threat to affect the trustiness of the measurement results. Follow-up actions, such as fault recovery, do rely on such results. Our results indicate that instead of simply assuming *seeing-is-believing*, we should always be cautious of malicious manipulation in network measurement.

Our future work includes both monitor placement and hiding routing information to combat measurement integrity attacks against network tomography. **Monitor Placement:** Existing monitor placement methods mainly focus on minimizing the number of monitors or enhancing the robustness. The theoretical results in Theorem 3 and experimental results in Figs. 6 and 7 reveal that a measurement integrity attack becomes more likely as the number of attackers increases. Hence, we aim to design a new monitor placement algorithm to first ensure identifiability under network tomography, then minimize each node's presence ratio on measurement paths such that when it is compromised, its impact to measurement manipulation is minimized. **Hiding Routing Information:** Routing information, existing in the routing matrix, is necessary for launching the attacks. Therefore, defenders can leverage anonymous routing protocols to hide the routing information, to prevent attackers from inferring the link metrics. We aim to understand how to hide routing information to minimize the impact of measurement integrity attacks against network tomography.

8 Related Work

Network Tomography: Network tomography is a generic way to compute network component (usually network link) metrics from measurements on end-to-end paths in a network. In essence, network tomography can be considered as an algorithmic process to transfer end-to-end measurements into link metric estimates. Existing work mainly focused on algorithm design and applications (e.g., [7], [8], [9], [10], [11], [12], [13]); and some recent papers also considered the problem of placement of monitors and identifiability of link metrics (e.g., [14], [15], [16], [17]). Network tomography has been proposed for measurement, fault diagnosis and localization in both wireline networks (e.g., [7], [8], [9], [10]) and wireless networks (e.g., [11], [12], [13]).

In general, these papers implicitly assume that individual link metrics can be inversely derived from the path measurements that indeed reflect the real link performance aggregate. In fact, it is not guaranteed that there exists no anomaly or malicious behavior in today's large-scale networks. However, potential security vulnerabilities in network tomography have not yet been investigated in the literature.

Packet Dropping and Delaying Attacks: There are various malicious attacks against a network, such as passive eavesdropping, active interfering, leakage of secret information, data tampering, impersonation, message distortion and denial-of-service attacks (e.g., [32], [33], [34], [35], [36]). Measurement integrity attacks drop or delay packets to damage a network, which is related to packet dropping attacks, such as black hole attacks that attract and drop all packets routed to malicious nodes and grey hole attacks (also called selective forwarding attacks) that only drop certain selected packets [37].

However, such traditional attacks can be discovered by finding out the links which always suffer long delay or high loss under network tomography [36]. In contrast, our measurement integrity attack strategy can not only hide the real identities of attackers in network tomography, but also make some legitimate nodes or links the scapegoats. Therefore, the proposed attack strategy is a new one that is able to deteriorate the network performance, while misleading network tomography based diagnostics.

Attack Detection and Defense: Existing network defense approaches are usually deployed in individual host systems (e.g., end nodes or edge routers). These mechanisms can directly detect anomalies on some particular victims. For example, the process of tracing back the forged IP packets to their true sources rather than the spoofed IP addresses that was used in the attack is called traceback. There are various IP traceback mechanisms that have been proposed to date (e.g., [38], [39]). Packet marking and filtering mechanism aims to mark legitimate packets at each router along their path to the destination so that victims' edge routers can filter the attack traffic (e.g., [40], [41]).

There are a few studies to detect network neutrality violations [28], [42], [43], [44], [45]. For example, the strategy in [42] relies on detecting whether traffic on specific ports is blocked. Authors in [43], [44] proposed a system to detect neutrality violations by inferring whether an ISP discriminates traffic based on performance data obtained passively.

There are also studies related to monitoring and analyzing network traffic to protect a system from network-based threats. For instance, route-based packet filtering system uses routing information to distinguish if a traffic flow at a router is valid and ensure that resources are made available only for legitimate use (e.g., [46], [47]). The work in [48] designed a strategy to detect misbehaving routers that absorb, discard or misroute packets. Such mechanism usually requires explicit communication among routers. The work in [49] presented a heuristic data structure to monitor traffic characteristics of network devices like routers to detect and eliminate attacks. In addition, traffic monitoring can also be leveraged for detecting anomalous packet forwarding [50].

Network tomography is performed by the network operator to obtain the global picture of the healthiness of a network. Therefore, the detection proposed in this paper is a network-wide approach that should follow immediately the network tomography process to detect whether such a process is manipulated or exploited by malicious behavior. Our network-wide attack detection approach to protect network tomography can be regarded as complementary to defense strategies deployed in individual host systems (e.g., end nodes and routers).

9 Conclusions

In this paper, we have provided theoretical and experimental results to analyze the feasibility of measurement integrity attacks against network tomography. We consider three basic strategies: chosen-victim, maximum-damage and obfuscation attacks, and show that malicious nodes can substantially damage a network and at the same time manipulate end-to-end measurements to make legitimate nodes scapegoats. We also present the conditions to detect and locate these attacks. The results in this paper indicate that the current *seeing-is-believing* assumption in network tomography renders a security vulnerability. Instead of simply trusting measurements, we should be always aware of measurement integrity attacks and carefully revisit existing designs for security in various applications.

Acknowledgement: This work was supported in part by NSF CNS-1717969.

References

[1] S. Zhao, Z. Lu, and C. Wang, "When seeing isn't believing: On feasibility and detectability of scapegoating in network tomography," in *Proc. of IEEE ICDCS*, 2017.

[2] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Netw.*, vol. 17, pp. 27–35, 2003.

[3] L. Ma, T. He, A. Swami, D. Towsley, K. K. Leung, and J. Lowe, "Node failure localization via network tomography," in *Proc. of ACM IMC*, 2014.

[4] T. He, C. Liu, A. Swami, D. Towsley, T. Salonidis, A. I. Bejan, and P. Yu, "Fisher information-based experiment design for network tomography," in *Proc. of IEEE SIGMETRICS*, 2015.

[5] H. Yao, S. Jaggi, and M. Chen, "Network coding tomography for network failures," in *Proc. of IEEE INFOCOM*, 2010.

[6] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statistical Science*, vol. 19, pp. 499–517, 2004.

[7] J. D. Horton and A. Lopez-Ortiz, "On the number of distributed measurement points for network tomography," in *Proc. of ACM IMC*, 2003.

[8] T. Bu, N. Duffield, F. L. Presti, and D. Towsley, "Network tomography on general topologies," in *Proc. of ACM SIGMETRICS*, 2002.

[9] M. H. Firooz and S. Roy, "Link delay estimation via expander graphs," *IEEE Trans. Commun.*, vol. 62, pp. 170–180, 2014.

[10] M. Rabbat, R. Nowak, and M. Coates, "Multiple source, multiple destination network tomography," in *Proc. of IEEE INFOCOM*, 2004.

[11] C.-K. Yu, K.-C. Chen, and S.-M. Cheng, "Cognitive radio network tomography," *IEEE Trans. Veh. Technol.*, vol. 59, 2010.

[12] J. Zhao, R. Govindan, and D. Estrin, "Sensor network tomography: Monitoring wireless sensor networks," *ACM SIGCOMM Computer Communication Review*, vol. 32, 2002.

[13] Y. Li, W. Cai, G. Tian, and W. Wang, "Loss tomography in wireless sensor network using Gibbs sampling," in *Proc. of EWSN*, 2007.

[14] A. Chen, J. Cao, and T. Bu, "Network tomography: Identifiability and fourier domain estimation," *IEEE Trans. Signal Process.*, vol. 58, pp. 6029–6039, 2010.

[15] T. He, L. Ma, A. Gkelias, K. K. Leung, A. Swami, and D. Towsley, "Robust monitor placement for network tomography in dynamic networks," in *Proc. of IEEE INFOCOM*, 2016.

[16] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Monitor placement for maximal identifiability in network tomography," in *Proc. of IEEE INFOCOM*, 2014.

[17] ——, "Identifiability of link metrics based on end-to-end path measurements," in *Proc. of ACM IMC*, 2013.

[18] C. A. Shue, A. J. Kalafut, and M. Gupta, "Abnormally malicious autonomous systems and their Internet connectivity," *IEEE/ACM Trans. Netw.*, vol. 20, pp. 220–230, 2012.

[19] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal, "RAPTOR: routing attacks on privacy in Tor," in *Proc. of USENIX Security*, 2015.

[20] L. Constantin, "Attackers slip rogue, backdoored firmware onto Cisco routers," *PC World - Security*, 2015.

[21] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. of IEEE S&P*, 2003.

[22] P. Tague and R. Poovendran, "Modeling node capture attacks in wireless sensor networks," in *Proc. of Allerton Conference on Communication, Control, and Computing*, 2008.

[23] Q. Zhao, Z. Ge, J. Wang, and J. Xu, "Robust traffic matrix estimation with imperfect information: Making use of multiple data sources," in *Proc. of ACM SIGMETRICS*, 2006.

[24] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Trans. Netw.*, vol. 20, pp. 906–916, 2012.

[25] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Inferring link metrics from end-to-end path measurements: Identifiability and monitor placement," *IEEE/ACM Trans. Netw.*, vol. 22, pp. 1351–1368, 2014.

[26] L. Yang and F. Li, "mTor: a multipath Tor routing beyond bandwidth throttling," in *Proc. of IEEE CNS*, 2015.

[27] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on optimization*, vol. 2, pp. 575–601, 1992.

[28] Z. Zhang, O. Mara, and K. Argyraki, "Network neutrality inference," in *Proc. of ACM SIGCOMM*, 2014.

[29] "Rocketfuel: An ISP topology mapping engine," *University of Washington*, 2002, [Online]. http://www.cs.washington.edu/research/networking/rocketfuel/.

[30] M. Penrose, *Random Geometric Graphs*. Oxford Univ. Press, 2003.

[31] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, pp. 1029–1046, 2009.

[32] D. Kapetanovic, G. Zheng, and F. Rusek, "Physical layer security for massive MIMO: An overview on passive eavesdropping and active attacks," *IEEE Commun. Mag.*, vol. 53, pp. 21–27, 2015.

[33] J. Sen, S. Koilakonda, and A. Ukil, "A mechanism for detection of cooperative black hole attack in mobile Ad Hoc networks," in *Proc. of IEEE ISMS*, 2011.

[34] T. Chothia, Y. Kawamoto, C. Novakovic, and D. Parker, "Probabilistic point-to-point information leakage," in *Proc. of IEEE CSF*, 2013.

[35] L. Yu, J. Deng, R. R. Brooks, and S. B. Yun, "Automobile ECU design to avoid data tampering," in *Proc. of ACM CISR*, 2015.

[36] B. Sharma, "A distributed cooperative approach to detect gray hole attack in MANETs," in *Proc. of ACM WCI*, 2015.

[37] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures," *J. Netw. and Comput. Appl.*, vol. 35, pp. 867–880, 2012.

[38] L. Cheng, D. M. Divakaran, A. W. K. Ang, W. Y. Lim, and V. L. Thing, "FACT: A framework for authentication in cloud-based IP traceback," *IEEE Trans. Inf. Forensics Security*, 2016.

[39] G. Yao, J. Bi, and A. V. Vasilakos, "Passive IP traceback: Disclosing the locations of IP spoofers from path backscatter," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 471–484, 2015.

[40] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against DDoS attacks," in *Proc. of IEEE S&P*, 2003.

[41] T. Peng, C. Leckie, and K. Ramamohanarao, "Protection from distributed denial of service attacks using history-based IP filtering," in *Proc. of IEEE ICC*, 2003.

[42] R. Beverly, S. Bauer, and A. Berger, "The internet is not a big truck: toward quantifying network neutrality," in *Proc. of Springer PAM*, 2007.

[43] M. B. Tariq, M. Motiwala, and N. Feamster, "Nano: Network access neutrality observatory," Tech. Rep., 2008.

[44] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar, "Detecting network neutrality violations with causal inference," in *Proc. of ACM CoNEXT*, 2009.

[45] U. Weinsberg, A. Soule, and L. Massoulie, "Inferring traffic shaping and policy parameters using end host measurements," in *Proc. of IEEE INFOCOM*, 2011.

[46] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets," in *Proc. of ACM SIGCOMM*, 2001.

[47] R. Thomas, B. Mark, T. Johnson, and J. Croall, "Netbouncer: client-legitimacy-based high-performance DDoS filtering," in *Proc. of IEEE DISCEX*, 2003.

[48] J. R. Hughes, T. Aura, and M. Bishop, "Using conservation of flow as a security mechanism in network protocols," in *Proc. of IEEE S&P*, 2000.

[49] T. M. Gil and M. Poletto, "MULTOPS: A data-structure for bandwidth attack detection," in *Proc. of USENIX Security*, 2001.

[50] A. T. Mizrak, S. Savage, and K. Marzullo, "Detecting compromised routers via packet forwarding behavior," *IEEE Netw.*, vol. 22, pp. 34–39, 2008.

Shangqing Zhao received his B.S. degree from Fujian Agriculture and Forestry University, Fuzhou, China, in 2010; his M.S. degree from Henan Polytechnic University, Jiaozuo, China, in 2015. He is working toward the Ph. D. degree in the Department of Electrical Engineering, University of South Florida. His research interests include network and mobile system design and security. He is a student member of IEEE and ACM.

Zhuo Lu is an Assistant Professor in the Department of Electrical Engineering, University of South Florida. He received his B.S. and M.S. degrees from Xidian University, Xi'an China, in 2002 and 2005, respectively, and his Ph.D. degree in computer engineering from North Carolina State University, Raleigh NC, in 2013. His research interests include network science, cyber security, data analytics, cyber-physical systems, mobile computing and wireless networking. He is a member of IEEE, ACM and USENIX.

Cliff Wang is the division chief of ARO Computing Sciences division. He manages resources to execute the Army's basic research investment in Computing Sciences. He leads the extramural program to help establish scientific foundation of information sciences and to create new knowledge in the field. Dr. Cliff Wang graduated from North Carolina State University with a PhD in computer engineering in 1996. He has been carrying out research in the area of computer vision, medical imaging, high speed networks, and most recently information security. He has authored over 50 technical papers and 3 Internet standards RFCs. Dr. Wang also holds adjunct professor appointment at both Department of Computer Science and Department of Electrical and Computer Engineering at North Carolina State University. Dr. Wang is a Fellow of IEEE.