When Seeing Isn't Believing: On Feasibility and Detectability of Scapegoating in Network Tomography

Shangqing Zhao and Zhuo Lu
University of South Florida
Tampa, FL 33620
Emails: {shangqing@mail., zhuolu@}usf.edu

Cliff Wang
North Carolina State University
Raleigh, NC 27695
Email: cliffwang@ncsu.edu

***Abstract*—Network tomography is a vital tool to estimate link qualities 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 *scapegoating*, 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 scapegoating approaches and show under what conditions attacks can be successful. We also reveal conditions to detect such attacks. Our theoretical and experimental results show that simply trusting measurements leads to scapegoating vulnerabilities. Thus, existing methods should be revisited accordingly for security in various applications.

***Index Terms*—Network tomography, trusted measurements, scapegoating, security, attack feasibility, attack detection.**

I. 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, such as the simple network management protocol (SNMP) [1], 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., [2]–[6]) 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., [6]–[9]) and wireless networks (e.g., [10]–[12]) without special cooperation from internal nodes.

By nature, network tomography does not directly observe network link metrics, but infers them via measurements over end-to-end paths, each of which consists of a few or more links. Existing work mainly focused on algorithm design and applications (e.g., [6]–[12]); and some recent papers also considered the problems of placement of monitors and identifiability of link metrics (e.g., [13]–[16]). In essence, network tomography can be considered as an algorithmic process to transfer end-to-end measurements into link metric estimates. Interestingly, all 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 [17], [18], backdoor-infected routers [19], and node-capture attacks [20], [21] as these adversaries actively affect packet forwarding and have become increasingly possible in today's complicated environments. Rather, the assumption renders a potential security vulnerability that may jeopardize the major objective of network monitoring and diagnosis.

In this paper, we develop an attack strategy, called *scapegoating*, taking advantage of this *seeing-is-believing* vulnerability in network tomography. Unlike conventional data integrity problems that are usually protected by standard methods (e.g., encryption and authentication), a key challenge associated with scapegoating attacks is that the facts (e.g., packet transmission/delivery timings) during network measurement can not be protected by such standard methods, but can be easily manipulated by malicious attackers. The basic idea of scapegoating 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 scapegoating strategies as follows.

1) Chosen-victim scapegoating, in which attackers target one or more given victims in the network.
2) Maximum-damage scapegoating, in which attackers find a number 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 amount 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 scapegoating. We also use network datasets to perform simulation experiments to show the success possibility, damage, and detectability 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 scapegoating is able to damage the network while substantially misleading network tomography.
- We systemically construct three scapegoating attack strategies, and investigate the feasibility of such attacks, then propose a detection method against scapegoating.
- We use real-world datasets to evaluate the threats of scapegoating in network systems with various settings. Experimental results confirm that, even for a single attacker, network tomography is vulnerable to scapegoating attacks.

Our work demonstrates that when scapegoating 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.

The remainder of this paper is organized as follows. In Section II, we introduce the models and state the research problem. In Section III, we design and discuss the scapegoating strategies. In Section IV, we analyze the feasibility of scapegoating and describe how to detect scapegoating. In Section V, we present experimental results. We discuss observations from analysis and experiments in Section VI, describe related work in Section VII and finally conclude in Section VIII.

II. Models and Problem Statement

In this section, we first review network tomography and introduce the basic idea of scapegoating. Then, we state our research problems. All notations are defined in Table I.

TABLE I
Notations.

$\mathbf{A}^T$	The transpose of matrix $\mathbf{A}$.
$\mathbf{A}^{-1}$	The inverse of matrix $\mathbf{A}$.
$\lVert\mathbf{a}\rVert_1$	The $\mathcal{L}$-1 norm of vector $\mathbf{a} = [a_1, a_2, \cdots, a_n]^T$, i.e., $\lVert\mathbf{a}\rVert_1 = \sum_{i=1}^{n} \lvert a_i \rvert$.
$\mathbf{x} \succeq \mathbf{y}$	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.
$\lvert\mathcal{A}\rvert$	The cardinality of set $\mathcal{A}$.

A. 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 link metrics over the path. For example, delay metrics are additive; and packet delivery or loss ratios are also additive in the logarithmic form [5], [16], [22].

Throughout this paper, we adopt similar assumptions in the literature for network tomography (e.g., [14]–[16]): (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 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., [14]–[16]). The literature (e.g., [23], [24]) have shown 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 scapegoating attacks.

B. Network Tomography and Formulation

Network tomography [6]–[16] is an algorithm to estimate link metrics from end-to-end measurements. To efficiently estimate link metrics, denoted by a column vector $\mathbf{x} = [x_1, x_2, \cdots, x_{|\mathcal{L}|}]^T$, monitors first select a set of measurement paths between each other, denoted by $\mathcal{P} = \{P_i\}_{i \in [1, |\mathcal{P}|]}$, and then send probe packets over these paths to obtain the path measurement metrics denoted by a column vector

$\mathbf{y} = [y_1, y_2, \cdots, y_{|\mathcal{P}|}]^T$. It has been shown [6]–[16] that the following linear relation between $\mathbf{x}$ and $\mathbf{y}$ holds

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

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 work on selecting or placing monitors (e.g., [14], [15]) ensures 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.

C. Motivation and Basic Idea of Scapegoating

By nature, 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 [17], [18], intentional bandwidth throttling systems [25], backdoor-infected routers [19] or attack-captured nodes [20], [21] 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, and M_3 are monitors that perform network tomography to estimate link metrics, and the number on each edge denotes the link index. These monitors choose 23 paths[1] listed in Fig. 1 for end-end measurement. For example, path 3 is formed by links 1, 4, 7, 10 (meaning that probe packets over the path in turn go through nodes M_1, A, C, D, and finally reach M_2). Assume that nodes B and C are malicious, which means that they can adversely affect the performance of all links connecting to them (i.e., links 2-8 shown in Fig. 1).

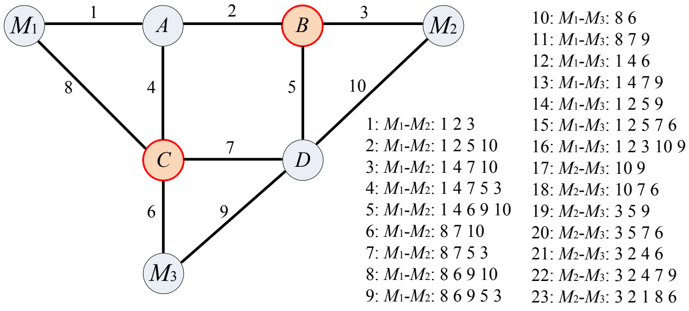


Fig. 1. A simple network example with 23 measurement paths, 10 links, and 7 nodes in which M_1, M_2, and M_3 are monitors, and B and C are malicious.

Apparently, nodes B and C cannot simply delay or drop all packets to do the damage, which may be easily identified as the root cause. Instead, our proposed attack strategy is they can try to delay or drop packets along certain directions to mislead tomography. Specifically, in Fig. 1, B and C exist on all measurement paths containing link 1 (i.e., paths 1-5, 12-16, 21-23). If B and C only do damage on these paths and be cooperative (i.e., delay or drop no packet) on other paths (e.g., path 5 consisting of links 8, 7, 5, and 3), the induced measurements under the network tomography algorithm (2) will show that path measurements containing link 1 always suffer long delay or high loss, while the others appear to be normal. This unavoidably misleads the network operator to believe that link 1 or its end-node A might have some issues. Therefore, we call such an attack strategy *scapegoating* and call link 1 or node A a *scapegoat* in the case.

D. Problem Statement

From the example in Fig. 1, we can consider scapegoating 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 scapegoating in the above example: How cooperative B and C need to be in order to launch a feasible attack? Can B and C make some other node like D the scapegoat? Is it possible to detect scapegoating?

To address these issues, we state two major research problems: In the network $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ with a small set of malicious nodes (called attackers) $\mathcal{V}_m \subset \mathcal{V}$ controlling a set of links $\mathcal{L}_m \subset \mathcal{L}$,

1) how to launch scapegoating attacks to delay or drop packets 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) how to detect scapegoating given the observed end-to-end measurements?

We assume that monitors can also be malicious nodes 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 amount of nodes are usually required to be chosen as monitors to ensure identifiability in network tomography [15], [16].

[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., [15], [16]). Fig. 1 shows such an example with 23 paths chosen.

III. Scapegoating Strategies

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

A. 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 scapegoating, a normal link may be misleadingly identified as abnormal. To facilitate formulating scapegoating strategies, we first define the normal and abnormal states of a network link.

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 few seconds, and considered normal if the delay is tens of milliseconds (ms). However, when the link delay is few hundred 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 scapegoating 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.

B. Attack Manipulation Vector and Inflicted Damage

Apparently, except for scapegoating, a major goal of attackers is to cause damage to the network. Therefore, we also need to measure the damage due to scapegoating. 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 B can obviously affect any data flow going through links 2, 3, and 5 (e.g., delaying or dropping packets).

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

$$\mathbf{y}' = \mathbf{y} + \mathbf{m}, \tag{3}$$

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 the attacks over all paths. For example, when an end-to-end path has a delay metric 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 I. For example, attackers can intentionally postpone forwarding packets, thus incurring more delay. 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, attackers B and C are not on path 17 (formed by links 9 and 10), and thus cannot manipulate the measurement of path 17. 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, the 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 Scapegoating): The damage of scapegoating 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 scapegoating 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 scapegoating 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.

C. Formulation of Scapegoating

Scapegoating aims to do the damage to the 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. Attackers can choose different strategies to hide themselves and inflict the damage. 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 amount of links.

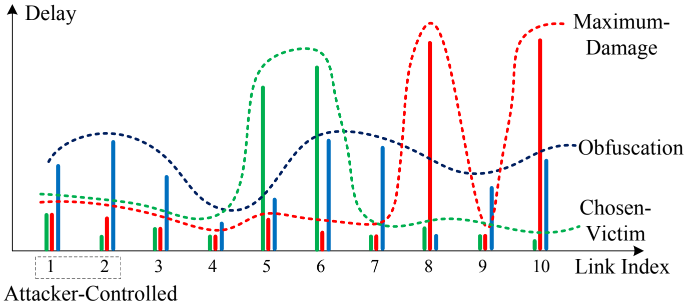


Fig. 2. Examples of chosen-victim scapegoating, maximum-damage scapegoating, and obfuscation. Under different strategies, the link metrics under tomography exhibit different patterns. Solid lines represent the values of end-to-end delay metrics and each dotted line denotes the envelope of solid lines under the same scapegoating strategy.

Fig. 2 shows an illustrative example of how different scapegoating strategies affect the link delay metrics obtained by network tomography. 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 find that links 8 and 10 can be the scapegoats with highest delays. Under obfuscation, the attackers can make most links exhibit similarly high delays, which can confuse the network operator to find which links are truly problematic.

In the following, we mathematically formulate these scapegoating strategies.

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 Definitions 1 and 2, we can formulate this basic scapegoating strategy as

$$\begin{aligned} &\underset{\mathbf{m}}{\text{maximize}} && \|\mathbf{m}\|_1 , && (4)\\ &\text{subject to} && \mathbf{m} \text{ satisfies Constraint } 1, && \\ &&& S(l) = \mathsf{normal},\ \forall\, l \in \mathcal{L}_m, && (5)\\ &&& S(l) = \mathsf{abnormal},\ \forall\, l \in \mathcal{L}_s, && (6)\\ &&& \mathcal{L}_m \cap \mathcal{L}_s = \varnothing, && (7)\end{aligned}$$

where constraints (5) and (6) 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 (7), achieve the goal of scapegoating under network tomography.

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

$$\begin{aligned} &\underset{\mathbf{m},\mathcal{L}_s\subset\mathcal{L}}{\text{maximize}} && \|\mathbf{m}\|_1 , && (8)\\ &\text{subject to} && \mathbf{m} \text{ satisfies Constraint } 1, && \\ &&& \text{Constraints in (5), (6), and (7).} && \end{aligned}$$

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 amount 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.

$$\begin{aligned} &\underset{\mathbf{m},\mathcal{L}_s\subset\mathcal{L}}{\text{maximize}} && \|\mathbf{m}\|_1 , && (9)\\ &\text{subject to} && \mathbf{m} \text{ satisfies Constraint } 1, && \\ &&& S(l) = \mathsf{uncertain},\ \forall\, l\in\mathcal{L}_o=\mathcal{L}_s\cup\mathcal{L}_m, && (10)\\ &&& \mathcal{L}_s \neq \varnothing, && (11)\end{aligned}$$

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 (10). 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 in the uncertain state result in obfuscation.

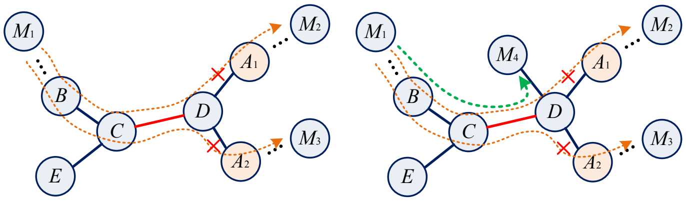
Given these formally defined basic strategies, attackers are able to launch scapegoating attacks against network tomography to maximize the damage, make scapegoats, or obfuscate the network operator. In addition, attackers may also develop more sophisticated strategies based upon these three ones.

IV. Feasibility and Detectability of Scapegoating

After we formulated scapegoating 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 scapegoating if it is successfully launched? In this section, we answer these two questions by first analyzing the feasibility of scapegoating, then describing how to detect scapegoating.

A. Feasibility Analysis

Whether scapegoating 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 $M_1 \rightarrow B \rightarrow C \rightarrow D \rightarrow M_4$ 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.

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): Scapegoating 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 have developed three strategies, but do not need to prove the feasibility individually. It is easy to know that if the chosen-victim scapegoating (4) is feasible, the maximum-damage one (8) is also feasible. Then, we write the chosen-victim scapegoating (4) and obfuscation (9) into a generic form, and show that a feasible solution exists in the generic form when the set of malicious nodes $\mathcal{V}_m$ can perfectly cut the set of victim links $\mathcal{L}_s$ from all measurements paths.

According to Definition 1, the constraints in (6), (7) and (10) represent 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 these constraints as

$$\mathbf{s}_u \succeq \hat{\mathbf{x}} \succeq \mathbf{s}_l, \tag{12}$$

where $\hat{\mathbf{x}}$ is link metric vector estimated by network tomography, $\mathbf{s}_u$ and $\mathbf{s}_l$ are called the upper and lower bound vectors. By controlling $\mathbf{s}_u$ and $\mathbf{s}_l$, we can accommodate either chosen-victim scapegoating or obfuscation.

Therefore, we only need to show that for a given manipulated metric vector $\hat{\mathbf{x}}^*$ satisfying (12), there exists a resultant vector $\mathbf{m}^*$ that meets Constraint 1 for successful scapegoating. To this end, we can first write the measurement model (1) as

$$\mathbf{y}' = \mathbf{y}^* + \mathbf{m}^* = \mathbf{R}\hat{\mathbf{x}}^*, \tag{13}$$

where $\mathbf{y}'$ is the observed measurement vector under scapegoating, and $\mathbf{y}^*$ is the true measurement vector if there is no scapegoating and satisfies

$$\mathbf{y}^* = \mathbf{R}\mathbf{x}^*, \tag{14}$$

with $\mathbf{x}^*$ being the true link metric vector.

It follows from (13) and (14) that

$$\mathbf{m}^* = \mathbf{R}\Delta\hat{\mathbf{x}}^*, \tag{15}$$

where $\Delta\hat{\mathbf{x}}^* = \hat{\mathbf{x}}^* - \mathbf{x}^*$. For the i-th entry m_i^* in $\mathbf{m}^*$, we have

$$m_i^* = \sum_j R_{i,j} \Delta\hat{x}_j^*, \tag{16}$$

where $R_{i,j}$ is the (i,j)-th entry in routing matrix $\mathbf{R}$ and $\Delta\hat{x}_j^*$ is the j-th entry of $\Delta\hat{\mathbf{x}}^*$.

Because $\mathcal{V}_m$ is a perfect cut, if there is no attacker on path $P_i \in \mathcal{P}$, there will be no victim link on path P_i as well. This indicates that $R_{i,j} = 0$ if link $l_j \in \mathcal{L}_m \cup \mathcal{L}_s$. In addition, if link $l_j \notin \mathcal{L}_m \cup \mathcal{L}_s$, $\Delta\hat{x}_j^* = 0$ as the attackers do not manipulate the metric of link l_j. Combining the two observations, we obtain $m_i^* = 0$ if there is no attack on path P_i, which satisfies Constraint 1 thus completes the proof. □

2) Imperfect Cut: If attackers only form an imperfect cut of the victim links, the formulation of a scapegoating strategy may not always yield a feasible solution, which depends on specific network settings. We are interested in understanding the scapegoating 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 (Scapegoating Success Probability under Imperfect Cut): The scapegoating success probability is defined as the probability that a scapegoating strategy yields a feasible solution. Under generic random assumptions, the scapegoating 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: Let $L = |\mathcal{P}|$ be the total number of measurement paths in network tomography. Assume that attackers $\mathcal{V}_m$ are present on $k < L$ measurement paths. Define a vector space $\mathcal{M}_k^L = \{\mathbf{v} \,|\, \mathbf{v} \in \mathbb{R}^{L\times 1}$, and $L-k$ entries in $\mathbf{v}$ are always zeros$\}$, where $\mathbb{R}^{L\times 1}$ denotes the L-dimensional vector space. It is clear that for any $s \geq k$, it always holds that

$$\mathcal{M}_k^L \subset \mathcal{M}_s^L. \tag{17}$$

It is clear that the attack manipulation vector $\mathbf{m}_k \in \mathcal{M}_k^L$. The scapegoating success probability p can be denoted as

$$p(\mathbf{m}_k) \quad = \quad \mathbb{P}\left(\mathcal{E}(\mathbf{m}_k)\right), \tag{18}$$

where $\mathcal{E}(\mathbf{m}_k)$ denotes a set satisfying

$$\mathcal{E}(\mathbf{m}_k) = \left\{\mathbf{m} | \mathbf{m} \in \mathcal{M}_k^L, \text{and } \mathbf{m} \text{ is a feasible solution}\right\}. \tag{19}$$

Now, consider the scenario that the total number of paths used in tomography is still fixed to L, but increase the total number of infected end-to-end measurement paths. Specifically, there are $s > k$ paths that include the victim links and at least one attack link, i.e., less entries in the new manipulation vector $\mathbf{m}_s$ are always zeros. The success probability becomes

$$p(\mathbf{m}_s) \quad = \quad \mathbb{P}\left(\mathcal{E}(\mathbf{m}_s)\right), \tag{20}$$

where

$$\mathcal{E}(\mathbf{m}_s) = \left\{\mathbf{m} | \mathbf{m} \in \mathcal{M}_s^L, \text{and } \mathbf{m} \text{ is a feasible solution}\right\}. \tag{21}$$

Therefore, it suffices to prove $p(\mathbf{m}_k) < p(\mathbf{m}_s)$, or equivalently to prove

$$\mathbf{a} \in \mathcal{E}(\mathbf{m}_s), \forall\, \mathbf{a} \in \mathcal{E}(\mathbf{m}_k), \tag{22}$$

which immediately follows from (17), (19), and (21). □

B. Detecting Scapegoating Attacks

We have analyzed the feasibility of scapegoating attacks. If a scapegoating attack is successfully launched, we should never trust the result obtained by network tomography. It is necessary to know how to detect scapegoating 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 scapegoating. In other words, verifying $\hat{\mathbf{x}}$ and $\mathbf{y}'$ according to (1) results in our detection method

$$\text{scapegoating} \begin{cases} \text{exists,} & \text{if } \mathbf{R}\hat{\mathbf{x}} \neq \mathbf{y}', \\ \text{does not exist,} & \text{if } \mathbf{R}\hat{\mathbf{x}} = \mathbf{y}'. \end{cases} \tag{23}$$

with the following detectability.

Theorem 3 (Detectability): Under the detection mechanism (23), scapegoating 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 is partly based on the proof for Theorem 1.

First, if $\mathbf{R}$ is a square matrix, it is easy to verify that $\mathbf{R}\hat{\mathbf{x}} = \mathbf{y}'$ always holds. Therefore, it is not possible to detect scapegoating under the linear model (1).

Then, we consider that $\mathbf{R}$ is not a square matrix. If attackers $\mathcal{V}_m$ can perfectly cut victims $\mathcal{L}_s$, the attackers can always choose an attack manipulation vector $\mathbf{m}^*$ such that $\mathbf{R}\hat{\mathbf{x}} = \mathbf{y}'$ as shown in (13). Therefore, no inconsistency can be found in the detection method (23).

If attackers $\mathcal{V}_m$ do not perfectly cut victims $\mathcal{L}_s$, there always exists at least one path on which there is no attacker but at least a victim link with metric manipulated. This means that the observed measurement on this path in the sum of all true link metrics because there is no attacker on the path. However, because the metric of the victim link is manipulated by attackers, the observed measurement will be inconsistent with the sum of the manipulated link metrics. Consequently, $\mathbf{R}\hat{\mathbf{x}} \neq \mathbf{y}'$, meaning the existence of scapegoating. □

Remark 3: 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 4: In practice, even when there is no attack, $\mathbf{R}\hat{\mathbf{x}}$ may not exactly equal to $\mathbf{y}'$ in (23) due to randomness in packet delivery and measurement error. Therefore, the scapegoating 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.

V. Experimental Evaluation

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

A. Experimental Setups

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 as many as possible in the network, 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.

B. Simple Network Scenario

We first consider the simple wireline network scenario in Fig. 1 to illustrate the feasibility and impact of scapegoating attacks. There are 10 links and two attackers B and C in the network. As we can see from the network topology in Fig. 1, B and C are on many measurement paths, it is expected that they can easily launch scapegoating attacks.

In experiments, we find that scapegoating is always successfully if B and C can perfectly cut a link from all measurement paths, which verifies Theorem 1. A more interesting case is whether they can make a link a scapegoat which cannot be perfectly cut by them. Fig. 4 shows the delays of all 10 links under network tomography when B and C launch chosen-victim scapegoating to target link 10, which they do not perfectly cut. The attack leads to an average delay of 820.87ms. Fig. 4 also demonstrates a successful attack, indicating that scapegoating is still feasible even when there is no perfect cut by attackers.

Fig. 5 depicts the impacts of maximum-damage scapegoating that leads to an average end-to-end delay of 1239.4ms in the network, highest in all chosen-victim attacks. We can see from Fig. 5 that links 1 and 9 are misleadingly identified as abnormal.

Fig. 6 illustrates the impacts of obfuscation due to attackers B and C. It is observed from Fig. 6 that all delays range from 200ms to 1000ms that represents the uncertain state. Under

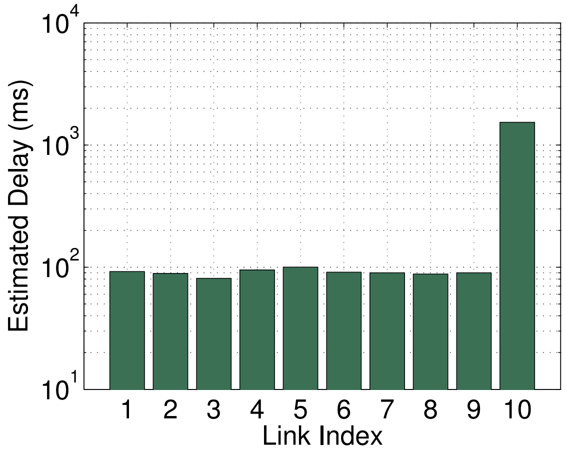


Fig. 4. Chosen-victim scapegoating: the estimated delay under misled network tomography for link 10 is greater than the abnormal threshold 800ms. The attackers do not perfectly cut link 10.

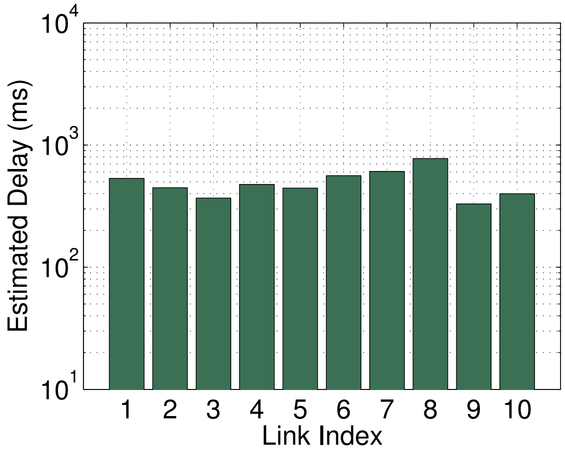


Fig. 6. Obfuscation: the estimated delays under misled network tomography are all in the intermediate state.

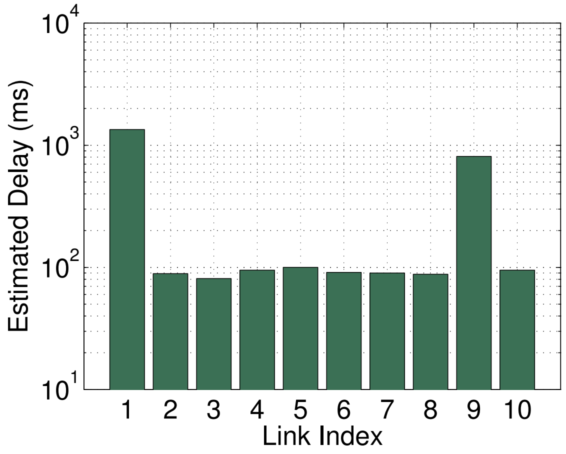


Fig. 5. Maximum-damage scapegoating: the estimated delays under misled network tomography for links 1 and 9 are greater than the abnormal threshold 800ms.

obfuscation in Fig. 6, it is difficult to tell which link is exactly problematic.

Figs. 4, 5, and 6 have shown the feasibility and impacts of the proposed scapegoating strategies in a simple network. Next, we move on to larger network scenarios.

C. Scapegoating Success Probabilities

We consider two types of network scenarios.

- Wireline networks. We use the Rocketfuel datasets [26] 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 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, [27], [28]). 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.

We choose monitors and measurement paths according to a random selection algorithm based on the minimum monitor placement rule in [16]. In our experiments, we define the scapegoating success probability as the ratio between the number of successful attacks and the total number of runs for a network topology, and use it to measure the feasibility of scapegoating.

1) Chosen-Victim Scapegoating: A straightforward way to show the feasibility is to measure the success probability as a function of the number of attackers in the network. However, as shown in Theorems 1 and 2, an essential condition for scapegoating is not the absolute number of attackers in the network, but the number of measurement paths between monitors where chosen-victim scapegoating attackers are present. Therefore, we aim to illustrate the attack success probability as a function of 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. It is obvious that the attack presence ratio is 100% if the attackers can perfectly cut all victims.

Fig. 7 depicts the success probabilities of chosen-victim scapegoating in both wireline and wireless topologies. We see that from both types of networks, the success probability increases as the attack presence ratio increases. For example, when the attack presence ratio goes from 60% to 70%, the success probability increases accordingly from 19.5% to 51.2% for the wireline topology as shown in Fig. 7. We also find that scapegoating is less successful in the wireless topology with node density $\lambda = 5$, because the 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.

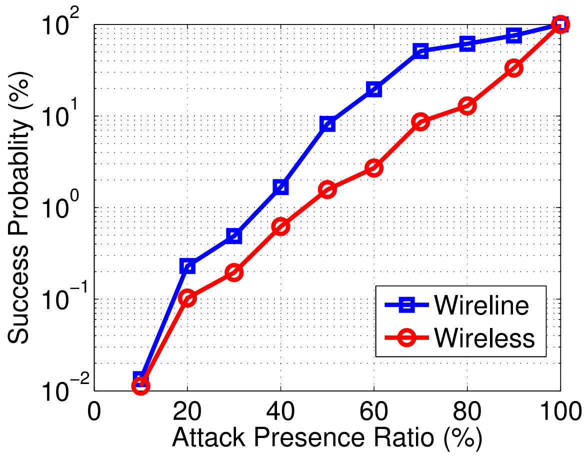


Fig. 7. The success probabilities of chosen-victim attacks versus attack presence ratios in wireline and wireless networks.

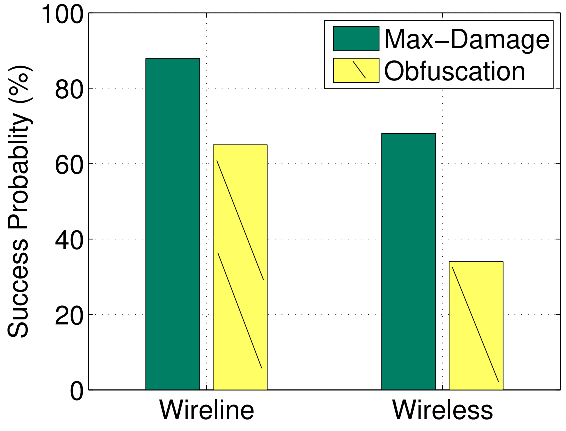


Fig. 8. The success probabilities of maximum-damage scapegoating and obfuscation by one single attacker in wireline and wireless networks.

2) Maximum-Damage Scapegoating and Obfuscation: In the maximum-damage scapegoating and obfuscation strategies, attackers do not target a particular victim, but aim to find the best victims to fulfill their goals. Because the number of malicious or compromised nodes is usually limited in practice, we focus on the scenarios, where there is only one single attacker to launch maximum-damage scapegoating and obfuscation attack. We also impose another condition on obfuscation, in which the attacker must make at least 5 victim links show the uncertain status to be considered successful.

Fig. 8 shows the success probabilities of maximum-damage and obfuscation attacks. It is noted from Fig. 8 that even one single attacker is likely to succeed. In fact, 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. As we observe from Fig. 8, obfuscation is generally less possible than maximum-damage scapegoating as it has to manipulate a number of victim links.

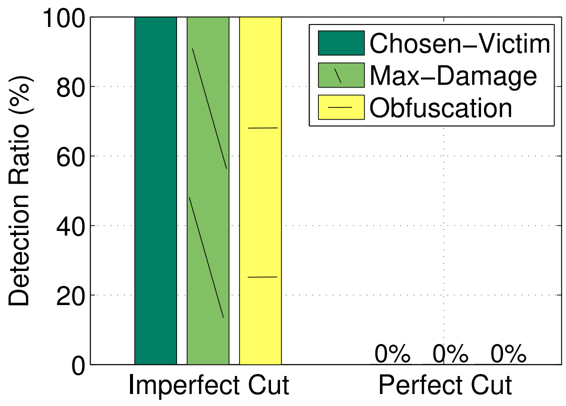


Fig. 9. The detection ratios of chosen-victim, maximum-damage and obfuscation attackers with perfect and imperfect cuts.

D. Detection

We then use the detection method proposed in Section IV-B to detect scapegoating. According to Theorem 3, there is no way for the method to detect scapegoating 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 scapegoating attacks 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.

VI. Discussions and Further Applications

In this section, we discuss our results associated with scapegoating feasibility and detection, as well as the potential impact on other related work.

- To launch scapegoating attacks, the attackers must have the information of 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, scapegoating does 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.
- The scope of Theorem 3 is bounded by the general formulation in (1). Under this model, we show that it is

impossible to detect scapegoating when attackers can perfectly cut victims. This does not eliminate the possibility of detection methods in other domains, such as intrusion detection deployed at the application layer in each node's computer system to detect potential compromise, and some other self-diagnosis systems, which are orthogonal to the scope of this paper.

- The results presented in this paper may further lead to new monitor placement algorithms developed for security. 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 Fig. 7 reveal that scapegoating becomes more likely as the attack presence ratio increases. Hence, a potential perspective for monitor placement is to first ensure identifiability under network tomography, then make sure that each node's presence ratio on measurement paths is minimized, assuming that the node becomes compromised.

VII. Related Work

In this section, we discuss existing work related to the research in this paper.

1) 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., [6]–[12]); and some recent papers also considered the problem of placement of monitors and identifiability of link metrics (e.g., [13]–[16]). Network tomography has been proposed for measurement, fault diagnosis and localization in both wireline networks (e.g., [6]–[9]) and wireless networks (e.g., [10]–[12]).

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.

2) Packet Dropping 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., [29]–[33]). Scapegoating 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 [34].

However, such traditional attacks can be discovered by finding out the links which always suffer long delay or high loss under network tomography [33]. In contrast, our scapegoating 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, scapegoating is a new attack strategy that is able to deteriorate the network performance, while misleading network tomography based diagnostics.

3) Attack Detection and Defense: Exiting network attack 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., [35], [36]). 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., [37], [38]).

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., [39], [40]). The work in [41] designed a strategy to detect misbehaving routers that absorb, discard or misroute packets. Such mechanism usually requires explicit communication among routers. The work in [42] 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 [43].

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).

VIII. Conclusions

In this paper, we provided theoretical and experimental results to analyze the feasibility of scapegoating against network tomography. We considered three basic strategies: chosen-victim, maximum-damage and obfuscation attacks, and showed 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 presented the conditions to detect scapegoating. The results in this paper indicate that the current *see-is-believing* assumption in network tomography renders a security vulnerability. Instead of simply trusting measurements, we should be always aware of scapegoating and carefully revisit existing designs for security in various applications.

Acknowledgement

The work by Shangqing Zhao and Zhuo Lu was supported in part by NSF CNS-1464114, CNS-1701394.

References

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

[2] 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.

[3] 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.

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

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

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

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

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

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

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

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

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

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

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

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

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

[22] 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.

[23] 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.

[24] 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.

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

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

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

[28] 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.

[29] 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.

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

[31] T. Chothia, Y. Kawamoto, C. Novakovic, and D. Parker, “Probabilistic point-to-point information leakage,” in *Prof. of IEEE CSF*, 2013.

[32] L. Yu, J. Deng, R. R. Brooks, and S. B. Yun, “Automobile ECU design to avoid data tampering,” in *Prof. of ACM CISR*, 2015.

[33] B. Sharma, “A distributed cooperative approach to detect gray hole attack in MANETs,” in *Prof. of ACM WCI*, 2015.

[34] 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.

[35] 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.

[36] 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.

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

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

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

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

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

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

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