Traffic Inference in Anonymous MANETs

Yunzhong Liu, Rui Zhang, Jing Shi, and Yanchao Zhang
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Email: {yl92, rz23, js39, yczhang}@njit.edu

***Abstract*—The open wireless medium in a mobile ad-hoc network (MANET) enables malicious traffic analysis to dynamically infer the network traffic pattern in hostile environments. The disclosure of the traffic pattern and its changes is often devastating in a mission-critical MANET. A number of anonymous routing protocols have been recently proposed as an effective countermeasure against traffic analysis in MANETs. In this paper, we propose a novel traffic inference algorithm, called TIA, which enables a passive global adversary to accurately infer the traffic pattern in an anonymous MANET without compromising any node. As the first work of its kind, TIA works on existing on-demand anonymous MANET routing protocols. Detailed simulations show that TIA can infer the traffic pattern with an accuracy as high as 95%. Our results in this paper highlight the necessity for cross-layer designs to defend a MANET against traffic analysis.**

I. Introduction

Mobile ad-hoc networks (MANETs) have great potential in hostile battlefield-like environments without a wired communication infrastructure. The shared wireless medium in a MANET unfortunately enables passive, adversarial eavesdropping on arbitrary radio transmissions. The adversary can then run traffic analysis on overheard transmissions to infer the *network traffic pattern*, which consists of a set of end-to-end flows with each described by a 6-tuple, ⟨source, destination, start-time, end-time, rate, path⟩.

The disclosure of the traffic pattern and its changes is often devastating for a mission-critical MANET. For example, a node as the source or destination of many end-to-end flows may be a VIP node which often issues tactical commands or collects tactical information for making critical decisions. In addition, high-rate flows may imply the relationships of the two end nodes in terms of rank (a node may be allowed to communicate with others with rank just above or below itself). Also, an unexpected change of the traffic pattern in a tactical MANET may indicate a forthcoming action, a chain of commands, or a state change of network alertness [1]. The adversary can then exploit the obtained information to launch various targeted attacks such as compromising the VIP nodes.

Anonymous routing protocols (e.g., [2]–[13]) have been proposed as a countermeasure against malicious traffic analysis in MANETs. They all aim to prevent inferring the traffic pattern by hiding the real sources, real destinations, and source-destination pairs of overheard packets. These schemes can withstand a local adversary who is incapable of overhearing every radio transmission to various degrees. It remains unclear whether they can defeat a global adversary who is able to eavesdrop on every radio transmission.

To the best of our knowledge, Huang *et al.* [14]–[16] made the only effort to study the resilience of anonymous MANET routing protocols to a global adversary. Under the assumption of a perfect anonymous routing protocol but unreliable transmissions in the MAC-layer, the packet-counting technique is used for tracing flows. They treat all packets as data packets and the output of SPTA [14], [15] is an estimation of the maximum possible traffic volume between any two nodes instead of the real traffic pattern defined before. Based on SPTA, STARS [16] estimates the probability of any node being a source or destination and the probability of any two nodes communicating with each other, but the actual traffic pattern still remains secret.

In this paper, we present a novel *T*raffic *I*nference *A*lgorithm, called TIA, which allows a global adversary to accurately infer the MANET traffic pattern despite the use of existing anonymous on-demand routing protocols [2]–[5], [7], [9], [10], [12]. TIA first exploits the overheard routing frames for flow recognition and then traces each flow in rounds based on the data-frame interarrival times. Given $\lambda > 2$ consecutive frames of a flow, the vector of $\lambda - 1$ corresponding frame interarrival times on any link is highly correlated with that of the next link along the flow path. This enables the adversary to iteratively derive the hidden flows and thus the traffic pattern from overheard MAC frames without prior knowledge of the interarrival time distribution of any flow. Although interarrival-based traffic analysis has been extensively conducted on low-latency mix networks (e.g., [17], [18]), its feasibility in anonymous MANETs remains untouched. Our major contributions are summarized as follows.

- We validate that routing frames and data-frame interarrival times are effective metrics to recognize and trace anonymous MANET flows.
- We develop an interarrival-based algorithm, called TIA, whereby a passive global adversary can infer the traffic pattern despite the use of some well-known anonymous on-demand MANET routing protocols.
- We evaluate TIA by extensive simulations involving CBR and VBR flows following various rate distributions. Our simulation results show that TIA can infer the traffic pattern with an accuracy as high as 95%.

Our results in this paper highlight the necessity for cross-layer solutions to truly anonymous MANET communications.

II. Related Work

Most anonymous MANET routing protocols like [2]–[5], [7], [9], [10] and [12] are adaptations of regular on-demand routing protocols such as AODV [19] or DSR [20] where routes are discovered as needed. Regardless of their implementation details, these schemes all aim to hide real packet sources and destinations in both route discovery and packet forwarding from a limited number of local and internal attackers. In this paper, we show that it is feasible for a passive global adversary to accurately discover the traffic pattern despite the use of these elegant schemes.

In [21], a small set of special nodes known as *Mix*es are purposely deployed to relay data packets from different end-to-end connections by reordering and re-encrypting the packets such that incoming and outgoing data packets cannot be related. However, this scheme is still vulnerable to the adversary with the global and dynamic flow context information. AO2P [6] is another on-demand anonymous routing protocol for MANETs, where route discovery is performed via geographic routing based on the destination's position. Once a route is set up, packets are forwarded from the source to the destination based on the pseudo IDs established during route discovery. In [8], Aad *et al.* propose a packet-coding technique as a combination of source routing, multicast, and onion routing [22] to achieve anonymous communications. This technique assumes proactive routing protocols in which routes are always maintained. Our TIA technique also works on [6], [8], [21] after small adaptations. Due to space limitations, however, we will focus on anonymous on-demand routing protocols [2]–[5], [7], [9], [10] and [12] in this paper.

ALARM [11] and PRISM [13] are anonymous location-aided routing protocols for MANETs, where routing is based on node locations instead of their IDs. The resilience of these schemes to traffic analysis is beyond the scope of this paper.

Timing-analysis attacks and defenses (e.g., [17], [18], [23]) on low-latency mix systems have been extensively studied. The objective of these attacks is often to identify whether two nodes are communicating via a low-latency mix system using various flow correlation techniques. In contrast, we are the first to apply timing analysis on anonymous MANET communications and also have a finer requirement to infer the traffic pattern consisting of 6-tuples as defined before.

III. Network and Adversary Models

A. Network Model

We consider a single-authority MANET deployed in hostile scenarios such as military and homeland security operations. There are N nodes in the network, where N may change with node join, leave, or failure over time. Depending on different applications, N may range from several tens to several thousands or even more. Each node has a unique network address and a unique MAC address. To simplify the notation, we use i to represent both the network and MAC addresses of node i ($i = 1, \ldots, N$).

We assume an anonymous on-demand MANET routing protocol such as [2]–[5], [7], [9], [10], [12]. In such protocols, data packets are of a constant size, and per-hop re-encryption is also used to ensure that the same packet looks different across each hop enroute to the destination. These measures are to prevent packet tracing based on unique packet sizes and unchanging packet contents.

We also assume reliable MAC-layer communications, which means that there should be a suitable MAC protocol like 802.11 DCF involving an RTS-CTS-DATA-ACK four-way handshake. Although most anonymous routing protocols require each node to not use its true MAC address in MAC-layer communications, reliable communications can still be realized with unique temporary link identifiers established during route discovery, see [2], [7], [9] for example.

B. Adversary Model

We assume a global adversary controlling some eavesdroppers which can collaboratively overhear every MAC frame. The eavesdroppers communicate among themselves and with the adversary via a separate channel invisible to the MANET nodes. We have justified the feasibility of global eavesdropping using extensive simulations. According to our simulation results, less than 40 eavesdroppers with a reception range of 250 m are needed for global eavesdropping in a 2000×2000 m^2 MANET region; in contrast, more than 400 nodes with the same transmission range are needed to ensure sufficiently high network connectivity. The adversary and his eavesdroppers are assumed to remain passive and external to the MANET during eavesdropping and traffic analysis, though compromising some nodes may certainly help traffic analysis.

We assume that the adversary can associate a unique ID with every MANET node. This can be achieved in many ways, e.g., based on the nodes' unique exterior features captured by video sensors on eavesdroppers. This assumption means that the adversary is able to associate the packets initiated from or destined to the same node no matter whether it uses its real MAC address or not. Without this assumption, any passive traffic analysis on MAC frames is meaningless if nodes use changing MAC addresses. Note that the adversary only cares about whether and when any two nodes communicate with which rates instead of their real addresses. In the rest of this paper, we will refer to each node by its ID assigned by the adversary. Each flow in the flow pattern is also described by the source and destination IDs assigned by the adversary.

Another important assumption is that the adversary can locate the transmitter of any MAC frame using methods like signal triangulation. Note that any reliable transmission of a data frame involves at least the data frame from the sender and an acknowledgement frame from the receiver [2], [7], [9], which are sent in sequence. With this assumption, the adversary can ascertain the senders of the data and acknowledgement frames, respectively. More specifically, this means that the adversary can associate (sender ID, receiver ID) with each frame even if the real sender and receiver MAC addresses are not used as in [2], [7], [9].

IV. TIA: A Traffic Inference Algorithm for Anonymous MANETs

In this section, we detail the TIA design. TIA works on existing anonymous on-demand MANET routing protocols such as [2]–[5], [7], [9], [10], [12]. Common in these protocols is a similar anonymous route discovery process preceding anonymous data forwarding. For ease of presentation, hereafter we take ANODR [2], [12] as an example which is the most classical and has been fully implemented in the popular network simulator QualNet 4.5.1.

The adversary partitions time into periods and uses TIA to infer the traffic pattern in each period. The traffic patterns of consecutive periods can be aggregated to obtain the long-term pattern or compared to infer the pattern changes. Our discussion will focus on one period. TIA consists of three key components as follows.

- *Evidence generation*: The adversary divides all the MAC frames overheard in the target period into data frames, routing frames, and MAC control frames.
- *Flow recognition*: The adversary recognizes each flow except its traffic volume and end time by analyzing routing frames.
- *Traffic inference*: The adversary decides the traffic volume and end time of each flow by analyzing data frames and thus infers the traffic pattern.

In what follows, we first detail each component and then give the complete process in Algorithm 5.

A. Evidence Generation

The adversary periodically collects all the MAC frames overheard by its eavesdroppers. The MAC frames comprise data frames, MAC control frames such as RTS/CTS/ACK, and routing frames. There are three kinds of routing frames in ANODR [2], [12], which include route requests (RREQs), routing replies (RREPs), and routing errors (RERRs). Among them, RREQs are broadcast frames, while RREPs and RERRs are unicast frames. Since frame headers contain frame types and are not encrypted in ANODR, the adversary can easily extract all the data frames (denoted $\mathcal{D}$) and all the routing frames (denoted by $\mathcal{R}$). Each frame record in $\mathcal{D}$ and $\mathcal{R}$ contains at least its type, sender, receiver, and sent time. Recall our assumption in Section III-B that the adversary assigns one unique ID to each node for its own use. The frame senders and receivers are actually the corresponding IDs assigned by the adversary instead of the real MAC addresses. Note that even if frame headers are encrypted, the adversary can still construct $\mathcal{D}$ and $\mathcal{R}$ based on the size differences and unique transmission characteristics of each type of frames. It is possible that a MAC frame is retransmitted multiple times, in which case only the last frame successfully transmitted is included in $\mathcal{D}$ or $\mathcal{R}$.

B. Flow Recognition

After obtaining the routing-frame set $\mathcal{R}$, the adversary proceeds to identify a set $\mathcal{F}$ of flows occurring in the target period. Each flow $f \in \mathcal{F}$ consists of the following fields.

- $f.src$: the source of flow f.
- $f.dest$: the destination of flow f.
- $f.path$: an ordered set of nodes from $f.src$ to $f.dest$ along the routing path.
- $f.start$: the time when the first identified data frame was output from $f.src$.
- $f.end$: the time when the last identified data frame was output from $f.src$.
- $f.volume$: the number of identified data frames from $f.src$ along $f.path$ to $f.dest$.

The adversary constructs $\mathcal{F}$ based on the anonymous route discovery process in ANODR. In particular, if a source intends to initiate a flow for a certain destination, it broadcasts a RREQ frame which will be forwarded once by any other node in the network. Each RREQ frame has a unique identifer in plain text. Only the intended destination can open the RREQ frame and answer it by a RREP frame which is unicasted back to the source via the reverse path established during the RREQ propagation. Given this knowledge, the adversary identifies from $\mathcal{R}$ all the RREP initiators. From each RREP initiator, say D, the adversary traces the RREP frame enroute to the corresponding RREQ initiator, say S. Unlike a RREQ frame, there is no unique identifier in a RREP frame to permit easy tracing; the RREP frame content also changes due to hop-by-hop re-encryption. The adversary, however, can exploit the frame type in plain text to trace the RREP frame with the following simple rule. Assume that D initiates a RREP frame sent to node A_2 at time t_1. If A_2 immediately sends a RREP frame to node A_3 at time $t_2 \in (t_1, t_1 + \tau_1]$, where τ_1 denotes the maximum per-hop forwarding delay for a routing frame, the adversary can relate the two RREP frames to obtain a partial path $\{A_3, A_2, D\}$; otherwise, the adversary considers A_2 the corresponding source S and obtains the complete path $\{S, D\}$. Repeating this process, the adversary can obtain the whole path $\{A_{n+2}, \ldots, A_1\}$, where n denotes the number of intermediate nodes between D and S, $A_1 = D$, and $A_{n+2} = S$. Assuming symmetric links, S will use this path for transmitting data frames to D. A flow record f can then be created with $f.src$ equal to S, $f.dest$ equal to D, $f.start$ and $f.end$ equal to the sending time of the RREP from A_{n+1} to S, $f.path$ equal to $\{A_{n+2}, \ldots, A_1\}$, and $f.volume$ equal to zero. $f.start$, $f.end$, and $f.volume$ will be updated in the subsequent traffic inference. Following this process, the adversary can obtain the flow set $\mathcal{F}$.

The detailed process is shown in Algorithm 1. It starts from the first RREQ frame in $\mathcal{R}$. If there is one node initiating a RREP frame in response to the RREQ frame under consideration within half of a predetermined maximum network round-trip delay, the flow path can be identified iteratively with the procedure in the previous paragraph. Line 19 tests whether the RREP can be traced to the original RREQ source, which may fail if some intermediate nodes along the path have moved after forwarding the RREQ. It is also possible that no RREP frame can be found for the RREQ, which may happen if the destination is temporarily isolated. When multiple flows

Algorithm 1: FlowRecog($\mathcal{R}$)

Require: The routing-frame set $\mathcal{R}$
Ensure: The flow set $\mathcal{F}$ with incomplete fields
1: $rreq \leftarrow$ the first RREQ frame in $\mathcal{R}$
2: **while** $rreq$ exists **do**
3: $src \leftarrow rreq.sender$
4: $path \leftarrow \emptyset$, $bFindPath = false$
5: $rrep^* \leftarrow$ the first RREP frame in $\mathcal{R}$ in response to $rreq$ within half of a maximum round-trip delay
6: **while** ($bFindPath = false$) *and* ($rrep^*$ exists) **do**
7: $rrep \leftarrow rrep^*$
8: **while** $rrep$ exists **do**
9: $t \leftarrow rrep.time$
10: **if** $path = \emptyset$ **then**
11: $path \leftarrow \langle rrep.receiver, rrep.sender \rangle$
12: **else**
13: Append $rrep.receiver$ to the head of $path$
14: **end if**
15: $rrep \leftarrow$ the first RREP frame sent from $rrep.receiver$ in $(t, t + \tau_1]$
16: **end while**
17: **if** $path \neq \emptyset$ **then**
18: $src' \leftarrow$ the first node in $path$
19: **if** $src' = src$ **then**
20: $bFindPath$ = true
21: $dest \leftarrow$ the last node in $path$
22: $(f.src, f.dest, f.path) \leftarrow (src, dest, path)$
23: $(f.start, f.end, f.volume) \leftarrow (t, t, 0)$
24: Insert f into $\mathcal{F}$
25: **end if**
26: **end if**
27: **if** $bFindPath$ = false **then**
28: $rrep^* \leftarrow$ the next RREP frame of the current $rrep^*$ frame in $\mathcal{R}$ in response to $rreq$ within half of a maximum round-trip delay
29: **end if**
30: **end while**
31: Delete from $\mathcal{R}$ the RREQs with the same identifer as $rreq$ and the corresponding RREPs (if found) from $\mathcal{R}$
32: $rreq \leftarrow$ the first RREQ frame in $\mathcal{R}$
33: **end while**
34: **return** $\mathcal{F}$

are initiated at almost the same time, the first RREP frame following one RREQ frame may not be the reply to it. Lines 27-29 consider multiple candidates as the initial RREP.

C. Traffic Inference

After flow recognition, the adversary proceeds to infer the traffic pattern hidden in the data-frame set $\mathcal{D}$. In particular, it needs to update the start time $f.start$, the end time $f.end$, and the volume $f.volume$ of each flow $f \in \mathcal{F}$ to their exact values. To do so, the adversary first labels all flows in $\mathcal{F}$ as untraced. The flows are individually traced in an ascending order of their current start times. When one flow is finished, the adversary relabels it as traced and starts to trace a new untraced flow with the earliest start time.

The tracing of a flow consists of rounds. In each round, the adversary first selects from $\mathcal{D}$ a sequence of earliest non-processed data frames originating from the flow source, called an initial *source sequence*, among which some frames belong to the traced flow and others belong to other untraced flows

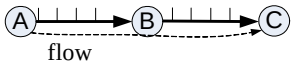


Fig. 1. Source-sequence correlation.

initiated by the flow source. The adversary then follows the flow path to trace the source sequence in a hop-by-hop fashion by correlating frame interarrival times. During the tracing, the source sequence will be updated across each hop and may shrink if containing the frames of other flows. If a subsequence of the original source sequence can be traced to end at the flow destination, the flow fields are updated accordingly. The adversary then deletes from $\mathcal{D}$ this subsequence and all the frames across each hop corresponding to each frame in the subsequence and starts a new round. The tracing of a flow finishes if the adversary cannot identify any more non-processed data frame for this flow.

We detail the traffic-inference process in what follows. Section IV-C1 explains the basic idea of source-sequence correlation at one intermediate node with one input frame sequence and one output frame sequence, which is the foundation of TIA. Section IV-C2 details the general source-sequence correlation process with multiple input and output frame sequences at one intermediate node. Section IV-C3 describes how to identify one initial source sequence with the correlation technique discussed before. Section IV-C4 finally gives the complete process of tracing one flow.

1) Source-sequence correlation: We use a simple example in Fig. 1 to illustrate the basic idea of source-sequence correlation, where $\{A, B, C\}$ belongs to the path of the flow under tracing. Assume that the adversary traces a source sequence to node A which is forwarded in λ consecutive frames to node B. Let $\mathcal{I} = \{I_1, \ldots, I_\lambda\}$ denote these input frames to B and $\{t_{I_1}, \ldots, t_{I_\lambda}\}$ be the corresponding frame transmission times. Assume that B subsequently outputs λ frames $\mathcal{O} = \{O_1, \ldots, O_\lambda\}$ to C at times $\{t_{O_1}, \ldots, t_{O_\lambda}\}$, where $0 < t_{O_i} - t_{I_i} \leq \tau_2$ for all $1 \leq i \leq \lambda$ and τ_2 is the maximum per-hop forwarding delay for a data frame. We can measure the correlation between $\mathcal{I}$ and $\mathcal{O}$, denoted by $\mathsf{corr}(\mathcal{I}, \mathcal{O})$, based on their frame interarrival times. The best known correlation metric is the Pearson product-moment correlation coefficient [24]. In particular, we have

$$\begin{aligned} \mathsf{corr}(\mathcal{I}, \mathcal{O}) &= \mathsf{corr}(\{x_1, \ldots, x_{\lambda-1}\}, \{y_1, \ldots, y_{\lambda-1}\}) \\ &= \frac{\sum_{i=1}^{\lambda-1}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{\lambda-1}(x_i - \mu_x)^2 * \sum_{i=1}^{\lambda-1}(y_i - \mu_y)^2}}, \end{aligned} \quad (1)$$

where $\lambda > 2, x_i = t_{I_{i+1}} - t_{I_1}, y_i = t_{O_{i+1}} - t_{O_1}, \mu_x = \frac{\sum_{i=1}^{\lambda-1} x_i}{\lambda-1}, \mu_y = \frac{\sum_{i=1}^{\lambda-1} y_i}{\lambda-1}$. We also define $\mathsf{corr}(\mathcal{I}, \mathcal{O}) = 1$ for $\lambda = 1$ and $\mathsf{corr}(\mathcal{I}, \mathcal{O}) = \frac{\min(x_1, y_1)}{\max(x_1, y_1)}$ for $\lambda = 2$. Note that $\mathsf{corr}(\mathcal{I}, \mathcal{O})$ ranges from -1 to 1 and indicates the degree of linear dependence between their frame interarrival times. Since FIFO queueing is used in ANODR and other anonymous MANET routing protocols, $\mathsf{corr}(\mathcal{I}, \mathcal{O})$ will be sufficiently close to 1 if there is a one-to-one mapping from $\mathcal{I}$ to $\mathcal{O}$. This conjecture is validated by our simulations in Section V.

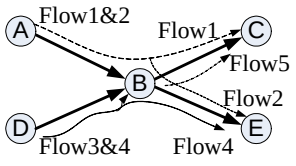


Fig. 2. An example for multiple intersecting flows.

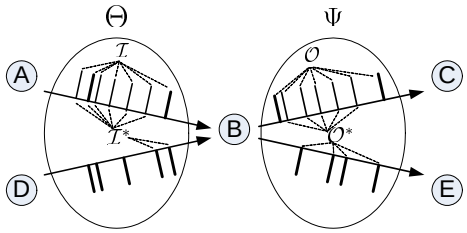


Fig. 3. An example for Algorithm 2.

2) Source-sequence extraction: Fig. 1 only gives the simplest case with one frame sequence entering B and one frame sequence leaving B, which contain the same number of frames. More generally, for the input sequence $\mathcal{I}$ on link_{AB}, there might be multiple candidate output frame sequences on link_{BC}, which may consist of frames from other flows arriving at B from links other than link_{AB}. Even worse, $\mathcal{I}$ may comprise frames from multiple flows starting from the same flow source and sharing the path until B. This means that the adversary needs to identify both the input subsequence from $\mathcal{I}$ and the corresponding output subsequence from multiple candidates for the traced flow. After this, the adversary treats the identified output subsequence as the new source sequence and traces it forward in the similar fashion. The shrinking of the source sequence may occur at each intermediate node along the traced flow path.

An example is given in Fig. 2, where flows 1 and 2 share link_{AB}, flows 3 and 4 share link_{DB}, flows 2 and 4 share link_{BE}, and node B initiates flow 5 which shares link_{BC} with flow 1. Assume that the adversary traces flow 1 to node B. The input sequence $\mathcal{I}$ on link_{AB} comprises frames from flows 1 and 2 which have the same source and will split after B. The frames of flow 1 on link_{BC} are mixed with those from flow 5, which should be identified as a new source sequence and traced forward from C.

We propose a source-sequence extraction algorithm which jointly considers all the frame sequences entering or leaving B on different links within a certain time window. Assume that the first and last frames of the traced source sequence $\mathcal{I}$ are sent on link_{AB} at times t_s and t_e, respectively. The frames of the traced flow will be forwarded during $(t_s, t_e + \tau_2]$ from B to C which is the next hop along the flow path. Let $\mathcal{O}$ denote the frame sequence from B to C during $(t_s, t_e + \tau_2]$. The adversary first identifies $\Theta \subseteq \mathcal{D}$ as the set of data frames sent to B during $[t_s, t_e]$ and also $\Psi \subseteq \mathcal{D}$ as the set of data frames sent by B during $(t_s, t_e + \tau_2]$. Apparently, we have $\mathcal{I} \subseteq \Theta$ and $\mathcal{O} \subseteq \Psi$. The frames in both Θ and Ψ are ordered in an ascending order of the frame transmission time. The adversary aims to find a subsequence of $\mathcal{O}$ as a new source sequence of the traced flow that should be traced forward from C.

Algorithm 2: $\text{SrcExtract}(\Theta, \Psi)$

Require: Θ, Ψ
Ensure: $\mathcal{I}^*$, $\mathcal{O}^*$
1: $\lambda = \min\{|\Theta|, |\Psi|\}$
2: **while** $\lambda > 0$ **do**
3: $c \leftarrow \xi$
4: **for** each $X = \{\alpha_1, \alpha_2, ..., \alpha_\lambda\} \subseteq \Theta$ **do**
5: **for** each $Y = \{\beta_1, \beta_2, ..., \beta_\lambda\} \subseteq \Psi$ such that $0 < t_{\beta_i} - t_{\alpha_i} \leq \tau_2$ for all $1 \leq i \leq \lambda$ **do**
6: **if** $\mathsf{corr}(X, Y) > c$ **then**
7: $\mathcal{I}^* \leftarrow X$, $\mathcal{O}^* \leftarrow Y$
8: $c \leftarrow \mathsf{corr}(X, Y)$
9: **end if**
10: **end for**
11: **end for**
12: **if** $c > \xi$ **then**
13: Create one-to-one mappings between $\mathcal{I}^*$ and $\mathcal{O}^*$
14: **return** $\mathcal{I}^*$, $\mathcal{O}^*$
15: **end if**
16: $\lambda = \lambda - 1$
17: **end while**
18: **if** $\lambda = 0$ **then**
19: $\mathcal{I}^* \leftarrow \emptyset$, $\mathcal{O}^* \leftarrow \emptyset$
20: **return** $\mathcal{I}^*$, $\mathcal{O}^*$
21: **end if**

Our approach is to first find $\mathcal{I}^* \subseteq \Theta$ and $\mathcal{O}^* \subseteq \Psi$ which satisfy two conditions. First, I^* and $\mathcal{O}^*$ are the longest input/output subsequence pairs with the correlation value $\mathsf{corr}(I^*, \mathcal{O}^*)$ larger than a predetermined threshold ξ. Second, I^* and $\mathcal{O}^*$ achieve the highest correlation value among all the input/output subsequence pairs of the same cardinality. We thus have a dual-criteria optimization problem which can be solved by exhaustive search when Θ and Ψ are of limited cardinalities.

Once I^* and $\mathcal{O}^*$ are identified, the adversary can get the one-to-one mappings from frames in I^* to $\mathcal{O}^*$. We call the frames in both I^* and $\mathcal{O}^*$ as *mapped* frames which need not be considered in subsequent tracing rounds of the same flow. Then the adversary locates the longest input/output subsequence pairs from $\mathcal{I} \cap \mathcal{I}^*$ and $\mathcal{O} \cap \mathcal{O}^*$ such that each frame in the input subsequence can be mapped to a unique frame in the output subsequence. This output subsequence is the new source sequence on link_{BC} that should be traced forward, and the input subsequence is the corresponding source sequence on link_{AB}.

The detailed process for finding $\mathcal{I}^*$ and $\mathcal{O}^*$ is shown in Algorithm 2, where $\xi \in (0, 1)$ denotes a correlation threshold. Lines 2 to 17 are for satisfying the first condition of the optimization, while Lines 4 to 11 are for satisfying the second condition. An example corresponding to Fig. 2 is given in Fig. 3, where non-bold lines represent the discovered input/output frame sequences of the traced flow. The following theorem is about its computation complexity.

***Theorem** 1: The computation complexity of Algorithm 2 is $O((1 + \tau_2 R)^{TR})$, where R is the maximum link rate in frames/second and T is the time duration of one tracing round.*

Proof: Given λ, we first generate all the subsets $X \subseteq \Theta$

Algorithm 3: SrcSelect($\mathcal{D}, f$)

Require: $\mathcal{D}, f \in \mathcal{F}$
Ensure: link$_{\mathcal{I}}$, $\mathcal{I}$
1: link$_{\mathcal{I}} \leftarrow$ the first-hop directional link in $f.path$
2: node $\leftarrow$ link$_{\mathcal{I}}.sender$
3: $\mathcal{I} \leftarrow$ the ordered set of consecutive unmapped frames on link$_{\mathcal{I}}$ during $(f.t_e, f.t_e + l\omega]$
4: $t_s \leftarrow$ the transmission time of the first frame in $\mathcal{I}$
5: $t_e \leftarrow$ the transmission time of the last frame in $\mathcal{I}$
6: $\Theta \leftarrow$ the set of unmapped frames in $\mathcal{D}$ sent to node during $[t_s - \tau_2, t_e]$
7: $\Psi \leftarrow$ the set of unmapped frames in $\mathcal{D}$ sent from node during $(t_s - \tau_2, t_e + \tau_2]$
8: $(\mathcal{I}^*, \mathcal{O}^*) \leftarrow$ SrcExtract(Θ, Ψ)
9: $\mathcal{I} \leftarrow \mathcal{I} \cap (\Psi - \mathcal{O}^*)$
10: **return** link$_{\mathcal{I}}$, $\mathcal{I}$

with each comprising λ frames. The number of operations to generate $\binom{|\Theta|}{\lambda}$ of combinations is in the order of $O(\binom{|\Theta|}{\lambda})$. Given X, for each $\alpha_i \in X$, there are at most $\tau_2 R$ frames in Ψ which are sent during $(t_{\alpha_i}, t_{\alpha_i} + \tau_2]$. For each subset X=$\{\alpha_1, \alpha_2, ..., \alpha_\lambda\}$, there are at most $(\tau_2 R)^\lambda$ ways to choose the subset Y=$\{\beta_1, \beta_2, ..., \beta_\lambda\}$ in Line 5. So the total number of tries for a given λ is $\binom{|\Theta|}{\lambda}(\tau_2 R)^\lambda$. For λ from 1 to $|\Theta|$, the total number of combinations is upper-bounded by the following formulas.

$$\sum_{\lambda=1}^{|\Theta|} \binom{|\Theta|}{\lambda} (\tau_2 R)^\lambda < \sum_{\lambda=0}^{|\Theta|} \binom{|\Theta|}{\lambda} (\tau_2 R)^\lambda = (1 + \tau_2 R)^{|\Theta|} \leq (1 + \tau_2 R)^{TR}. \quad (2)$$

Therefore, the computation complexity is $O((1+\tau_2 R)^{TR})$. ■

Note that traffic inference is an off-line process conducted by the powerful adversary. We are also seeking ways to reduce the computation complexity of Algorithm 2.

3) Source-sequence selection: We now discuss how the adversary decides an initial source sequence. In ANODR and other anonymous or non-anonymous on-demand MANET routing protocols, a routing entry will be deleted if not used for ω seconds, where ω is a universal parameter. This implies that the minimum flow rate between any source-destination pair should be at least $1/\omega$ packets/second to avoid initiating another route discovery and thus a new flow in our case. In addition, it is better to make the source sequence consist of at least three packets to enable the correlation as in Eq. (1). These observations motivate the following source-sequence selection method.

Assume that the adversary is tracing flow f from source S with node A as the next hop. The adversary first selects from $\mathcal{D}$ a candidate source sequence $\mathcal{I}' \subseteq \mathcal{D}$, which consists of all the unmapped frames sent from S to A during $(f.t_e, f.t_e + l\omega]$. Here $f.t_e$ is the sending time of the last identified data frame belonging to flow f, and l is a design parameter larger than two to ensure that there are at least three frames of flow f in $\mathcal{I}'$ except for the last round of tracing. However, $\mathcal{I}'$ may contain some frames forwarded by S for other flows, which can fortunately be eliminated using Algorithm 2.

The detailed process of initial source-sequence selection is shown in Algorithm 3, where link$_{\mathcal{I}}$ refers to the directional first-hop link along the flow path. Line 9 removes from the candidate source sequence the set of frames that have been mapped to other flows entering $f.src$. Note that $\mathcal{I}$ will shrink in the following flow tracing process if containing some frames of other flows initiating from $f.src$.

The choice of l determines the tradeoff between tracing accuracy and algorithm complexity. On the one hand, the larger l, the more frames the adversary traces in each round, the less likely Algorithm 2 will falsely map the frames of the traced flow to those of another on consecutive links of the traced flow path. The underlying intuition is that the unique flow signature (i.e., the vector of frame interarrival times) can be better represented and identified by a large number of frames. On the other hand, the larger l, the longer tracing interval $l\omega$, which is denoted as T in *Theorem 1*, the more complex the combinatorial optimization problem Algorithm 2 need solve across each hop to the destination. In particular, the computation complexity of finding an optimal input/output subsequence pair in Algorithm 2 grows exponentially with l.

4) Tracing one flow: Now we are ready to illustrate the tracing of any flow $f \in \mathcal{F}$, which consists of rounds. At the beginning of each round, the adversary uses Algorithm 3 to select an initial source sequence $\mathcal{I}$ and then repeats applying Algorithm 2 to trace $\mathcal{I}$ along the flow path $f.path$ to the destination $f.dest$. The adversary then updates the fields of f accordingly. The tracing of flow f finishes in the first round where no data frames can be traced to $f.dest$.

The detailed process of one-round tracing of flow f is shown in Algorithm 4, which takes the data-frame set $\mathcal{D}$, the flow set $\mathcal{F}$, the traced flow f, the directional first-hop link link$_{\mathcal{I}}$, and the initial source sequence $\mathcal{I}$ as inputs. link$_{\mathcal{I}}$ and $\mathcal{I}$ are the outputs of Algorithm 3. Each time the source sequence $\mathcal{I}$ is traced forward, Algorithm 2 is invoked at Line 7 to find the optimal input/output mapping frame sequences at the current tracing node. Lines 9 to 16 find the new initial source sequence that should be traced forward from node which is not the flow destination, while Line 18 decides the set of data frames arriving at the flow destination node. For the latter case, Lines 24 to 30 update the flow volume and end time and also delete all the identified data frames of flow f from $\mathcal{D}$. Note that the start and end times of flow f initially are the same; Lines 24 to 26 update $f.start$ to the transmission time of the first frame output from $f.src$ in the first round. In addition, Line 33 labels f as "traced" because no data frames can be traced to $f.dest$ in this round, in which case the flow path has broken.

D. A Complete Traffic Inference Algorithm

The complete TIA is given in Algorithm 5, where Ω denotes all the MAC frames overheard in the target period. The adversary initially labels all the flows in $\mathcal{F}$ as "untraced" and traces the flows in the ascending order of their start times. Lines 6 to 14 describe the tracing of one flow f, which follows the process given in Section IV-C4. Note that the status of flow

Algorithm 4: SrcTrace($\mathcal{D}, \mathcal{F}, f, \text{link}_{\mathcal{I}}, \mathcal{I}$)

Require: $\mathcal{D}$, f, $\mathcal{F}$, $\text{link}_{\mathcal{I}}$, $\mathcal{I}$
Ensure: $\mathcal{F}$
1: node $\leftarrow$ $\text{link}_{\mathcal{I}}.receiver$
2: $t_s \leftarrow$ the transmission time of the first frame in $\mathcal{I}$
3: $t_e \leftarrow$ the transmission time of the last frame in $\mathcal{I}$
4: **while** $\mathcal{I} \neq \emptyset$ **do**
5: $\Theta \leftarrow$ the set of unmapped frames in $\mathcal{D}$ sent to node during $[t_s, t_e]$
6: $\Psi \leftarrow$ the set of unmapped frames in $\mathcal{D}$ sent from node during $(t_s, t_e + \tau_2]$
7: $(\mathcal{I}^*, \mathcal{O}^*) \leftarrow \text{SrcExtract}(\Theta, \Psi)$
8: **if** node $\neq f.dest$ **then**
9: $\mathcal{I} \leftarrow \mathcal{I} \cap \mathcal{I}^*$
10: $\text{link}_{\mathcal{O}} \leftarrow$ the directional next-hop link in $f.path$ starting from node
11: $\mathcal{O} \leftarrow$ the set of data frames in $\mathcal{D}$ sent on $\text{link}_{\mathcal{O}}$ during $(t_s, t_e + \tau_2]$
12: $\mathcal{O} \leftarrow \mathcal{O} \cap \mathcal{O}^*$
13: $\mathcal{I} \leftarrow$ the subset of frames in $\mathcal{O}$ with each having a mapping in $\mathcal{I}$
14: **if** $\mathcal{I} \neq \emptyset$ **then**
15: $\text{link}_{\mathcal{I}} \leftarrow \text{link}_{\mathcal{O}}$, node $\leftarrow \text{link}_{\mathcal{I}}.receiver$
16: **end if**
17: **else**
18: $\mathcal{I} \leftarrow (\mathcal{I} - \mathcal{I} \cap \mathcal{I}^*)$
19: **end if**
20: **if** $(\mathcal{I} \neq \emptyset)$ **then**
21: $t_s \leftarrow$ the transmission time of the first frame in $\mathcal{I}$
22: $t_e \leftarrow$ the transmission time of the last frame in $\mathcal{I}$
23: **if** node $= f.dest$ **then**
24: **if** $f.t_s = f.t_e$ **then**
25: $f.t_s \leftarrow$ the transmission time of the frame sent by $f.src$ which corresponds to the first frame in $\mathcal{I}$
26: **end if**
27: $f.t_e \leftarrow$ the transmission time of the frame sent by $f.src$ which corresponds to the last frame in $\mathcal{I}$
28: $f.volume \leftarrow f.volume + |\mathcal{I}|$
29: Delete from $\mathcal{D}$ all the mapped frames of f along $f.path$ which correspond to the frames in $\mathcal{I}$
30: $\mathcal{I} \leftarrow \emptyset$
31: **end if**
32: **else**
33: Label flow f as "traced"
34: **end if**
35: **end while**

Algorithm 5: TIA(Ω)

Require: All the overheard MAC frames Ω
Ensure: $\mathcal{F}$
1: Extract the routing-frame set $\mathcal{R}$ and the data-frame set $\mathcal{D}$ from the overheard MAC frames
2: $\mathcal{F} \leftarrow \text{FlowRecog}(\mathcal{R})$
3: Label each flow $f \in \mathcal{F}$ as "untraced"
4: **while** $\mathcal{F}$ contains untraced flows **do**
5: $f \leftarrow$ the untraced flow with the earliest start time
6: $(\text{link}_{\mathcal{I}}, \mathcal{I}) \leftarrow \text{SrcSelect}(\mathcal{D}, f)$
7: **while** $(\mathcal{I} \neq \emptyset)$ **do**
8: $\mathcal{F} \leftarrow \text{SrcTrace}(\mathcal{D}, \mathcal{F}, f, \text{link}_{\mathcal{I}}, \mathcal{I})$
9: **if** f is "untraced" **then**
10: $(\text{link}_{\mathcal{I}}, \mathcal{I}) \leftarrow \text{SrcSelect}(\mathcal{D}, f)$
11: **else**
12: $\mathcal{I} \leftarrow \emptyset$
13: **end if**
14: **end while**
15: Unmap all the mapped frames in $\mathcal{D}$
16: **end while**
17: **return** $\mathcal{F}$

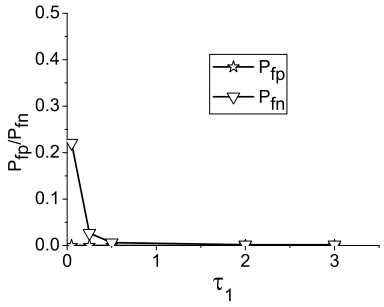


Fig. 4. False-positive and false-negative probabilities of flow recognition.

f will change to "traced" in the SrcTrace algorithm if no data frames are traced to $f.dest$ in one round. This condition is checked in Line 9.

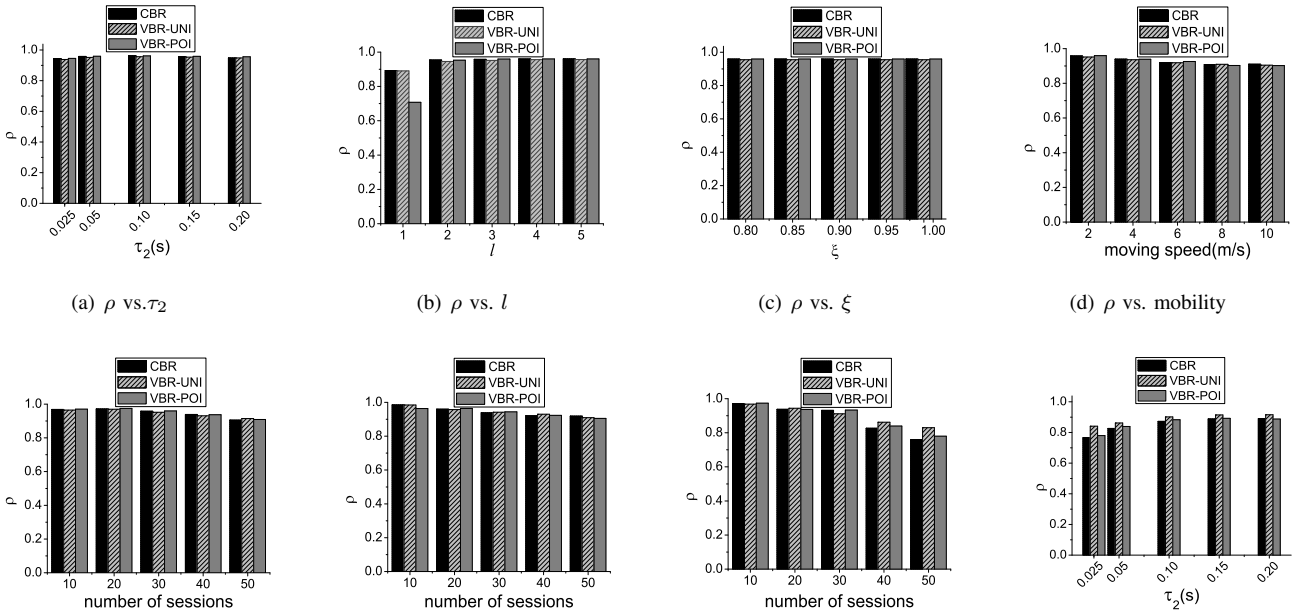
V. Performance Evaluation

In this section, we use simulations done in QualNet 4.5.1 to evaluate the performance of TIA.

A. Default Network Setting

We simulate N = 100 MANET nodes with a transmission range R = 250 m and initial locations uniformly distributed in a 1000×1000 m^2 region. The nodes move according to a random waypoint mobility model with a fixed speed 2 m/s and pause time fixed to 30 seconds. QualNet 4.5.1 includes an implementation of ANODR [12] which we use as the anonymous routing protocol. The live time of each routing entry is $\omega = 5$ s. The DCF of IEEE 802.11 is used as the MAC layer, the physical-layer path loss model is the two-ray model, and the channel capacity is 2 Mbps. Each simulation is executed for 15 simulated minutes. Each data point in our result represents an average of ten runs with identical traffic models in different randomly generated mobility scenarios.

We test TIA for CBR and VBR traffics. Two types of VBR traffics are simulated. The first type is denoted by VBR-UNI whose rate follows a uniform distribution, and the second type is denoted by VBR-POI whose rate follows a Poisson distribution. Unless otherwise stated, we simulate 30 sessions for each simulated traffic type with each corresponding to a random source-destination pair. Each session is randomly initiated during the first 5 simulated minutes. For each CBR session, the source sends packets at a constant rate uniformly distributed between $[0.5, 4]$ packets/second; for each VBR-UNI session, the sending rate dynamically changes between $[0.5, 4]$ packets/second; for each VBR-POI session, the average sending rate is uniformly distributed between $[0.5, 4]$ packets/second. Each data packet is of 512 bytes. In ANODR, the maximum per-hop forwarding delay (i.e., τ_1) of a routing frame is larger than that (i.e., τ_2) of a data frame due to



(a) ρ vs.τ_2 (b) ρ vs. l (c) ρ vs. ξ (d) ρ vs. mobility

(e) ρ vs. number of sessions (f) ρ vs. number of sessions with higher flow rates (g) ρ vs. number of sessions in a 2400×600 m^2 network (h) ρ vs. τ_2 with 40 sessions in a 2400×600 m^2 network

Fig. 5. The TIA accuracy.

cryptographic operations. Both τ_1 and τ_2 need be estimated by the adversary. By default, we set $\tau_1 = 3$ s, $\tau_2 = 0.05$ s, the tracing-interval parameter $l = 3$, and the correlation threshold $\xi = 0.95$.

B. Evaluation Metrics

We use the following three metrics to evaluate TIA.

- P_{fp} **(false-positive probability)**: Defined as the ratio of the number of fake flows falsely identified by TIA over the number of real flows.
- P_{fn} **(false-negative probability)**: Defined as the ratio of the number of real flows not identified by TIA over the number of real flows.
- ρ **(TIA accuracy)**: Defined as the average ratio of the identified aggregated traffic volume over the real aggregated traffic volume between any source-destination pair, where the aggregated traffic volume of a source-destination pair equals the sum of the traffic volumes of all the associated flows.

C. Simulation Results

1) Impact of τ_1: The adversary has to estimate τ_1 to recognize the flows using Algorithm 1. Fig. 4 shows the performance of Algorithm 1 with τ_1 varying from 0.025 s to 3 s. As we can see, P_{fp} is always zero, while P_{fn} quickly approaches zero as τ_1 increases. This means that Algorithm 1 can correctly recognize almost all the flows with a sufficiently large τ_1. Therefore, we set $\tau_1 = 3$ s in the rest of our simulations.

2) Impact of τ_2: The adversary also need estimate τ_2 to execute Algorithms 2, 3, and 4. Fig. 5(a) shows the TIA accuracy ρ varying with τ_2. We can see that ρ increases as τ_2 increases from 0.025 s to 0.05 s and does not improve much as τ_2 increases from 0.05 s to 0.2 s. According to Theorem 1, a large τ_2 will greatly increase the algorithm complexity, so we set a conservatively small value of $\tau_2 = 0.05$ s by default. However, when the heavy traffic load with large random delay is detected by the adversary, they can adaptively select a larger τ_2 to improve ρ, as shown in Fig. 5(h).

3) Impact of l: Fig. 5(b) shows ρ varying with the tracing-interval parameter l. We can see that ρ is relatively low for $l = 1$, especially for VBR-POI traffic. The reason is that a short sequence with $l = 1$ cannot sufficiently represent and identify the unique flow traffic signature, which is more serious for Poisson traffic. In general, the larger l, the higher ρ, which is expected. In addition, ρ improves very slowly as l increases from 2. Since the complexity of Algorithm 2 increases exponentially with the increase of l, we need to strike a good balance between TIA accuracy and computation complexity. We set $l = 3$ by default.

4) Impact of ξ: Fig. 5(c) shows the impact of the correlation threshold ξ on ρ. As we can see, ρ remains almost unchanged and sufficiently high as ξ increases from 0.8 to 0.99, which shows that TIA is not sensitive to ξ. We set $\xi = 0.95$ by default.

5) Impact of node mobility: Fig. 5(d) shows the impact of node mobility on ρ. As we expect, ρ decreases slowly with the increase of the moving speed. The reason is that higher mobility leads to more frequent routing-path breaks, resulting in potentially more packet drops at intermediate nodes. The undelivered packets add noise to the TIA execution and are not immediately distinguishable from other packets. Higher mobility also increases the packet forwarding delay, which is against the packet tracing. Therefore, it is more difficult to

infer the aggregated traffic volume with higher mobility, thus leading to the decrease of ρ. However, TIA can still achieve an accuracy above 90% even with a high moving speed of 10 m/s.

6) Impact of sessions and flow rates: Fig. 5(e) shows the impact of sessions on ρ. We can see that TIA can achieve an accuracy above 95% with 10/20/30 sessions and above 90% with 40/50 sessions for all three types of traffics. In Fig. 5(f), the range of flow rates is increased from $[0.5, 4]$ packets/second to $[0.5, 8]$ packets/second, and ρ does not decrease notably as compared with Fig. 5(e). This shows that TIA also works very well under higher flow rates.

7) Impact of path lengths: Fig. 5(g) shows the impact of longer routing paths on ρ. The network consists of 150 nodes uniformly deployed in a rectangle region of 2400×600 m^2, where the flow paths are averagely longer than those in a 1000×1000 m^2 region. We can see that TIA still achieves an accuracy above 90% for 10/20/30 sessions. For 40/50 sessions, ρ is not very high. The reason is that the packet delivery ratio is decreased in the rectangle region and is only 60% in some scenarios. The packet forwarding delay also increases significantly. These two factors affect the TIA performance similar to high mobility. Fig. 5(h) shows that a larger τ_2 is necessary for improving the TIA accuracy.

We also simulate a 5×40 static grid network, in which 200 nodes are deployed in 5 rows in a rectangle region, and each row comprises 40 nodes. The distances between adjacent rows and columns are 100 m and 150 m, respectively. Other network settings remain unchanged. For 30 sessions, the average and longest path lengths are 8.93 and 19 hops, respectively. For all three types of traffics, ρ is above 95%, which shows TIA works very well with long routing paths.

VI. Conclusion and Future Work

TIA works well on on-demand anonymous MANET routing protocols. It is also interesting to investigate anonymous proactive MANET routing protocols and compare them with on-demand ones with regard to the resilience against traffic analysis. We also envision a few possible defenses against TIA, such as cross-layer design, hiding routing frames, frame mixing, dummy packets [25], and information-theoretic approaches [26]–[28]. It is important and challenging to compare their effectiveness against TIA and the related overhead in consideration of the resource constrains and unique features of MANETs.

Acknowledge

This work was supported in part by the US National Science Foundation under grants CNS-0716302 and CNS-0844972 (CAREER). We would also like to thank anonymous reviewers for their constructive comments and helpful advice.

References

[1] DARPA, "Research challenges in high confidence networking," White paper, Arlington, VA, July 1998.

[2] J. Kong and X. Hong, "ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *ACM MobiHoc'03*, Annapolis, MD, June 2003, pp. 291 – 302.

[3] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng, "Anonymous secure routing in mobile ad-hoc networks," in *LCN'04*, Dublin, Ireland, Nov. 2004, pp. 102–108.

[4] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "SDAR: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks," in *IEEE LCN'04*, Dublin, Ireland, Nov. 2004, pp. 618–624.

[5] R. Song, L. Korba, and G. Yee, "AnonDSR: efficient anonymous dynamic source routing for mobile ad-hoc networks," in *SASN'05*, Alexandria, VA, Nov. 2005, pp. 33–42.

[6] X. Wu and B. Bhargava, "AO2P: Ad hoc on-demand position-based private routing protocol," *IEEE Trans. Mobile Comput.*, vol. 4, no. 4, pp. 335–348, July/Aug. 2005.

[7] Y. Zhang, W. Liu, and W. Lou, "Anonymous communications in mobile ad hoc networks," in *IEEE INFOCOM'05*, Miami, FL, Mar. 2005, pp. 1940–1951.

[8] I. Aad, C. Castelluccia, and J.-P. Hubaux, "Packet coding for strong anonymity in ad hoc networks," in *SecureComm'06*, Baltimore, MD, Aug. 2006.

[9] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "MASK: anonymous on-demand routing in mobile ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 5, no. 9, pp. 2376–2385, Sep. 2006.

[10] H. Choi, P. McDaniel, and T. F. La Porta, "Privacy preserving communication in MANETs," in *IEEE SECON'07*, San Diego, CA, June 2007, pp. 233–242.

[11] K. Defrawy and G. Tsudik, "ALARM: Anonymous location-aided routing in suspicious MANETs," in *IEEE ICNP'07*, Beijing, China, Oct. 2007, pp. 304–313.

[12] J. Kong, X. Hong, and M. Gerla, "An identity-free and on-demand routing scheme against anonymity threats in mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 8, pp. 888–902, Aug. 2007.

[13] K. Defrawy and G. Tsudik, "PRISM: Privacy-friendly routing in suspicious MANETs (and VANETs)," in *IEEE ICNP'08*, Orlando, Florida, Oct. 2008, pp. 258–267.

[14] D. Huang, "Unlinkability measure for ieee 802.11 based manets," *IEEE Trans. Wireless Commun.*, vol. 7, no. 3, pp. 1025–1034, Mar. 2008.

[15] ——, "On an information theoretical approach to model anonymous manet communications," in *IEEE ISIT'09*, Seoul, Korea, June/July 2009, pp. 1029–1033.

[16] Y. Qin and D. Huang, "A statistical traffic pattern discovery system for manets," in *IEEE Milcom'09*, Bonston, MA, Oct. 2009.

[17] Y. Zhu, X. Fu, B. Graham, R. Bettati, and W. Zhao, "On flow correlation attacks and countermeasures in mix networks," in *PET'04*, Toronto, Canada, May 2004, pp. 207–225.

[18] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: attacks and defenses," in *ESORICS'06*, Hamburg, Germany, Sept. 2006, pp. 18–33.

[19] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, July 2003.

[20] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Ad Hoc Wireless Networks*, edited by T. Imielinski and H. Korth, Kluwer Academic Publishers, New York, NY, 1996.

[21] S. Jiang, N. H. Vaidya, and W. Zhao, "A mix route algorithm for mix-net in wireless mobile ad hoc networks," in *MASS'04*, Fort Lauderdale, FL, Oct. 2004, pp. 406–415.

[22] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 482–494, May 1998.

[23] B. Levine, M. Reiter, C. Wang, and M. Wright, "Timing attacks in low-latency mix systems (extended abstract)," in *FC'04*, Key West, FL, Feb. 2004, pp. 251–265.

[24] W. Navidi, *Statistics for Engineers and Scientists*, 2nd ed. New York, NY: McGraw-Hill Higher Education, 2008.

[25] W. Wang, M. Motani, and V. Srinivasan, "Dependent link padding algorithms for low latency anonymity systems," in *ACM CCS'08*, Oct. 2008, pp. 323–332.

[26] P. Venkitasubramniam, T. He, and L. Tong, "Anonymous networking amidst eavesdroppers," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2770–2784, June 2008.

[27] P. Venkitasubramaniam and L. Tong, "Anonymous networking with minimum latency in multihop networks," in *IEEE Symposium on Security and Privacy*, Oakland, CA, May 2008, pp. 18–32.

[28] ——, "Throughput anonymity trade-off in wireless networks under latency constraints," in *IEEE INFOCOM'08*, Phoenix, AZ, Apr. 2008, pp. 241–245.