ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 11, November 2016

Traffic Pattern Analysis System for MANET

[1]Mr.M. Suresh Chinnathampy,[2] K.V.S.S.S.S.SAIRAM,[3]P.Sorimuthu Iyan.
Assistant Professor, Francis Xavier Engineering College, Tirunelveli [1]
Professor, Francis Xavier Engineering College, Tirunelveli [2]
PG Scholar, Francis Xavier Engineering College, Tirunelveli[3]

Abstract: Communication anonymity is a critical issue in MANETs which generally consists of the following aspects: 1) Source/destination anonymity- it is difficult to identify the sources or the destination of the network follows.2) End-to-End relationship anonymity-it is difficult to identify the end-to-end communication relations. The system proposes statistical traffic analysis system approach that consider the Salian characteristics of MANETs: The broadcasting, ad-hoc, and mobile nature. The most of the previous approaches are particular attacks in the sense that they either only try to identify the source nodes or to find out the corresponding destination nodes .The proposed method is a complete attacking system that first identifies all source and destination node and then determine their relationship.

I. Introduction

Traffic analysis attacks against the static wired networks (e.g., Internet) have been well investigated. The brute force attack proposed in [11] tries to track a message by enumerating all possible links a message could traverse. In node flushing attacks (a.k.a blending attacks, n 1 attacks) [10], the attacker sends a large quantity of messages to the targeted anonymous system (which is called a mix-net). Since most of the messages modified and reordered by the system are generated by the attacker, the attacker can track the rest a few (normal) messages. The timing attacks as proposed in [9] focus on the delay on each communication path. If the attacker can monitor the latency of each path, he can correlate the messages coming in and out of the system by analysing their transmission latencies. The message tagging attacks (e.g., [12]) require attackers to occupy at least one node that works as a router in the communication path so that they can tag some of the forwarded messages for traffic analysis. By recognizing the tags in latter transmission hops, attackers can track the traffic flow. The watermarking attacks are actually variants of the message tagging attacks. They reveal the end-to-end communication relations by purposely introducing latency to selected packets.

Different from the attacks is mentioned above statistical traffic analysis intends to discover sensitive information from the statistical characteristics of the network traffic, for example, the traffic volume. The adversaries usually do not change the network behavior (such as injecting or modifying packets). The only thing they do is to quietly collect traffic information and perform statistical calculations. The pre- decessor attacks are first pointed out by Reiter and Rubin [14]. Later works such as [5] and [6] extend them to all kinds of anonymous communication systems including onion-rout- ing [9], mix-net [10], and DC-net [22]. In a typical predecessor attack, the attackers act exactly as legitimate nodes in the network communications. They collectively maintain a single predecessor counter for each legitimate node in the system. When an attacker finds himself to be on an anonymous path to the targeted destination, he increments the shared counter for its predecessor node in this path. The counters are then used for the attackers to infer the possible source nodes of the given destination. Obviously, to launch such an attack, a large number of legitimate nodes must first be compromised and controlled by the attackers. This is usually not achievable in MANETs. Moreover, in a MANET protected by anonymity enhancing techniques, it is a difficult task itself to identify an actual destination node as the target due to the ad hoc nature. That is, destinations are indistinguishable from other nodes (e.g., relays) in a MANET. In fact, they usually act as relay nodes as well, forwarding traffic for others. The adversaries are not able to determine whether a particular node is a destination depending on whether the node sends out traffic. This is totally different from the situation in traditional infrastructural networks where the role of every node is determined. The statistical disclosure attacks as mentioned in [17], [18], [19], and [20] are similar. A statistical disclosure attack often targets a particular given source node and intends to expose its corresponding destinations. It is assumed that the packets initiated by the source are sent to several destinations with certain prob- ability distribution. The background (covering) traffic also has certain probability distribution (usually assumed to be uniformly distributed). After a large number of observations, the attackers are able to figure out the possible destinations of the given source. Nonetheless, the statistical disclosure attacks cannot be applied to MANETs either, because the attackers cannot easily identify the actual source nodes in MANETs. Even if a source node is identified, the attacks can only be performed when the attackers know for sure when the targeted source is originating traffic and can observe the network behavior in the absence of the source. However, the attackers are prevented from being able to do so by the ad hoc nature of MANETs, i.e., they cannot tell if the source is originating traffic or just forwarding traffic as a relay.

Due to the unique characteristics of MANETs, very limited investigation has been conducted on traffic analysis in the context of MANETs. He et al. proposed a timing-based approach in [23] to trace down the potential destinations given a known source. In this approach, assuming the transmission delays are bounded at each relay node, they estimate the flow rates of communication paths using packet matching. Then based on the estimated flow rates, a set of nodes that partition the network into two parts, one part to which the source can communicate in sufficient rate and the other to which it cannot, are identified to estimate the potential destinations. In [24], Liu et al. designed a traffic inference algorithm (TIA) for MANETs based on the assumption that the difference between data frames, routing frames, and MAC control frames is visible to the passive adversaries, so that they can recognize the point-to-point traffic using the MAC control frames, identify the end-to- end flows by tracing the routing frames, and then infer the actual traffic pattern using the data frames. The TIA achieves good accuracy in traffic inference, while the mechanism is tightly tied to particular anonymous routing protocols but not a general approach. Both [23] and [24] are analytical strategies which heavily rely on the deterministic network behaviors.

II. System Models

In this section, we present the fundamental system models adopted (assumed) by STARS.

a) Communication Model

We assume the anonymity enhancing techniques (such as [1], [2], [3]) are used to protect the MANETs. However, these techniques are designed to different levels of anonymity. To focus on the statistical traffic analysis, we assume, based on [21], that a combination of these techniques is applied and the targeted MANET commu- nication system is subject to the following model:

1. The PHY/MAC layer is controlled by the commonly used 802.11(a/b/g) protocol. But all MAC frames (packets) are encrypted so that the adversaries cannot decrypt them to look into the contents.
2. Padding is applied so that all MAC frames (packets) have the same size. Nobody can trace a packet according to its unique size.
3. The “virtual carrier sensing” option is disabled. The source/destination addresses in MAC and IP head- ers are set to a broadcasting address (i.e., all “1”) or to use identifier changing techniques. In this case, adversaries are prevented from identifying point-to- point communication relations.
4. No information about the traffic patterns is disclosed from the routing layer and above.
5. Dummy traffic and dummy delay are not used due to the highly restricted resources In MANETs

b)Attack Model

The attackers’ goal is to discover the traffic patterns among mobile nodes. Particularly, we have the following four assumptions for attackers:

1. The adversaries are passive signal detectors, i.e., they are not actively involved in the communications. They can monitor every single packet trans- mitted through the network.

2. The adversary nodes are connected through an additional channel which is different from the one used by the target MANET. Therefore, the communication between adversaries will not influence the MANET communication.

3. The adversaries can locate the signal source accord- ing to certain properties (e.g., transmission power and direction) of the detected signal, by using wireless location tracking techniques [25] such as triangulation, nearest sensor, or RF fingerprinting. Note that none of these techniques can identify the source of a signal from several nodes very close to each other. Hence, this assumption actually indicates that the targeted networks are sparse in terms of the node density. In other words, any two nodes in such a network are distant from each other so that the location tracking techniques in use are able to uniquely identify the source of a wireless signal. In the following of this paper, unless specifically denoted as "signal source" or "source of signal," the word "source" indicates the source of a network flow

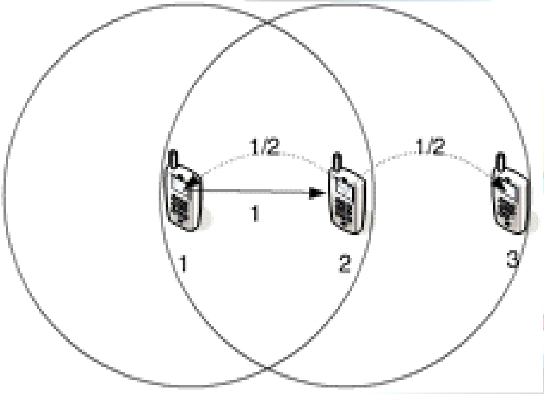


Fig. 1. A simple wireless ad hoc network.

4. The adversaries can trace the movement of each mobile node, by using cameras or other types of sensors. In this case, the signals (packets) trans- mitted by a node can always be associated with it even when the node moves from one spot to another.

III. Statistical Traffic Pattern Discovery System

To disclose the hidden traffic patterns in a MANET communication system, STARS includes two major steps. First, it uses the captured traffic to construct a sequence of point-to-point traffic matrices and then derives the end-to- end traffic matrix. Second, further analyzing the end-to- end traffic matrix, it calculates the probability for each node to be a source/destination (the source/destination probability distribution) and that for each pair of node to be an end-to-end communication link (the end-to-end link probability distribution)

To illustrate the basic idea of STARS, we use a simple scenario shown in Fig. 1 as an example. In this network, there are three wireless nodes (1, 2, and 3). Node 2 is located in the transmission range of node 1, and node 3 is located in the transmission range of node 2 (but not the transmission range of node 1). Two consecutive packets are detected: node 1 broadcasts a packet and then node 2 broadcasts a packet.

Traffic Matrices Construction

1.Point-to-Point Traffic Matrix

With the captured point-to-point (one-hop) traffic in a certain period T, we first need to build point-to-point traffic matrices such that each traffic matrix only contains "independent" one-hop packets. Note that two packets captured at different time could be the same packet appearing at different locations, such as the two packets sent by node 1 and node 2 consecutively in Fig. 1, so they are "dependent" on each other. To avoid a single point-to- point traffic matrix from containing two dependent packets, we apply a "time slicing" technique as.shown in Fig. 2. That is, we take snapshots of the network, and each snapshot is triggered by a captured packet. A sequence of snapshots during a time interval t_e constructs a slice represented by a traffic matrix W_e, which is an N ×N one-hop traffic relation matrix. The length of each time interval te is determined by two criteria:

1) A node can be either a sender or a receiver within this time interval. But it cannot be both.

2) Each traffic matrix must correctly represent the one-hop transmissions during the corresponding time interval In this way ,the construction of matrices

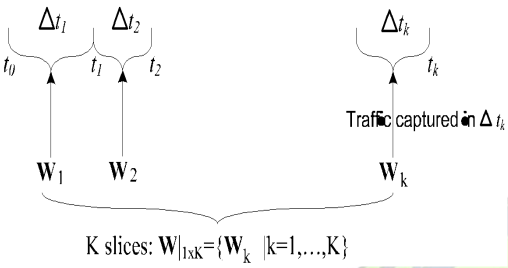

Δt_1
Δt_2
Δt_k
t_0
t_1
t_2
t_k
Traffic captured in Δt_k
W_1
W_2
W_k
K slices: $\mathbf{W}|_{1xK}=\{\mathbf{W}_k \ |k=1,\ldots,K\}$

Fig. 2. Slicing the time domain.

W|1 × e = (W 1 , W 2 , W k)the traffic matrices constructions. For example, traffic matrix We=(we(i,j))N×N is created for direct transmis- sions between nodes during time interval te . Since each snapshot of the network is triggered by capturing a packet, as long as potential receiver j is located within sender i's communication range (i.e., di;j ≤ r), a small change of distance di;j due to mobility will not alter the value assigned to we (i, j). If j moves out of the communication range of i due to mobility, the value of we (i; j) = 0. In this way, we slice the period T into a sequence of time intervals Δt1 ,Δ t2 , . . .,Δ tK , and record the captured packets into their corresponding traffic matrices W1 K = (W1 , W2 ,. . . ,WK). In each traffic matrix We=(we (i, j))N×N (N is the size of the network, e = 1, 2, . . . , K), the entry we (i, j) is the point-to-point traffic volume (number of packets) captured from node i to node j during the time interval Δte (we define we (i, i) to be 0). In addition, we use we (i; j):pkt to denote the set of all packets contributing to we (i, j). The "time slicing" has to make sure that all packets captured in any of the time intervals are independent with each other. In other words, two packets residing in different entries of the same matrix must not be the same packet transmitted through multiple hops. Note that, using the "time slicing" techniques, we also effectively handle the nodal mobility by taking snapshots of a sequence of relatively fixed network topologies. In addition to the "time slicing," we need to follow the three rules listed below: 1) The number of captured packets rather than the actual size of payloads is considered as the "traffic volume," since the size of payloads does not affect the traffic pasttern (and we assumed all MAC frames are of the same length due to the application of padding). 2) All nodes within the transmitting range of a packet have the same probability to sbe the actual receiver. For example, if a node i broadcasts a packet in the time interval Δ te , and nodes j1 , j2 , . . . jn are all within i's transmitting range, then the entries we (i; j1), we (i; j1),We(i,j2) . . . we (i, jn) should be all equally increased by 1=n. This is equivalent to dividing a packet into n subpackets and each sent to one neighboring node. For simplicity, in the remainder of the paper, we denote the original packet as "virtual size" 1 and each of the subpackets as "virtual size" 1=n. 3) Each packet p in we (i, j).pkt, has three associated features: p.vsize, p.time, and p.hop, denoting the "virtual size," transmitting time,and hop count of this packet, respectively. A packet's hop count is set to 1 when added to the point-to-point traffic matrix.

The said,for the example given in Fig. 1, we could derive:

$$W1=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}.\ W2=.\begin{bmatrix}0 & 0 & 0\\ 0.5 & 0 & 0.5\\ 0 & 0 & 0\end{bmatrix}$$

Note that in W2 , a real packet sent by node 2 is divided into two subpackets of virtual size 0.5, which means nodes 1 and 3 are equally likely to be the actual receiver.

2. End-to-End Traffic Matrix

Given a sequence of point-to-point traffic matrices Wj1 K , our goal is to derive the end-to-end traffic matrix R=(r(i,j))N×N,where r(i,j) is the accumulative traffic deduced from the point-to-point traffic captured directly and multihop traffic deduced from the point-to-point traffic. In this paper, we use the term accumulative traffic matrix and end-to-end traffic matrix interchangeably. The following Algorithm 1 (function f) takes Wj1 K as the inputs to derive the accumulative traffic matrix R

Algorithm1. —f(Wj1 K)

```
1: R=W1
2: for e = 1 to K    1 do
3:  R =g(R; $W_{e+1}$)(We+1)
4: end for
5: return R
```

In this algorithm, each update to R (line 3) includes the multihop traffic derivation function g shown as in Algorithm 2, and the addition of the point-to-point traffic matrix which is the evidence of possible direct (single- hop) communication

Algorithm; - 2 g(R, We+1).

```
1. R'=R
2. For I =1 to N do
3. For k=1 to N and k=/=0
4.  for j = 1 to N do
5. for each x E $w_{e+1}$ (j; .k): pkt do6.if E    y E r (I, j).pkt s.t. x.time --y: time < T
And y.hop < H then
7: create z with z .time =x. time
    Z. hop = y, hop +1
    Z.vsize = min {x.vsize,y.vsizeg}
 8:  r' (I, k): pkt = r' (I, k): pkt U(z)
 9: r' (I, k) = r' (I, k) + z.vsize
10: end if
11: end for
12: end for
13: end for
14: end for
15: return
```

Function g takes two inputs:

1) R is an end-to-end traffic matrix derived from point-to-point matrices W1 to We, and

2) Weþ is the next point-to-point traffic matrix. The output is the end-to-end traffic matrix derived from W1 to Weþ1

For each packet x recorded in Weþ1 , the function tries to find a packet y in R that is potentially the same packet transmitted at x's previous hop If such a packet y exists, then multi hop flow from the source of y to the destination of x should be derived. For instance, in our example scenario, we first let R = W1 . Then g(R; W2) should derive all possible end-to-end flows. W2 contains two packets, sent from node 2 to nodes 1 and 3, respectively. Let p2;1 and p2;3 denote these two packets. The current R contains only one packet p1;2 sent from node 1 to node 2. Thus, it is possible that p1;2 and p2;3 are the same packet appearing at different hops. In this case, a new packet p1;3 is derived to represent a multihop flow from node 1 to node 3. Since the volume of a multihop flow consisting of a sequence of one-hop transmissions cannot exceed the volume of any of the transmissions, we have p1;3 :vsize = min{p1;2 :vsize; p2;3 :vsize} = 0:5. Two constraints are considered for reasonable traffic inference: The differ- ence between the transmitting time of a packet at two consecutive hops cannot be too large and the hop-count of a packet cannot exceed a maximum value . We use T and H to represent the timing threshold and maximal hop-count threshold, respectively. If the network diameter is d, the average transmission distance of a mobile node is r, we can derive the approximated maximal hop-count threshold as: H = [d/r]. The timing threshold T must be at least the value of the maximum retransmission time. It depends on the specification of the MAC protocol. For instance, if the802.11 protocol is being used, T is determined by the maximum number of retransmissions, the contention window size, and the exponential back-off algorithm.

After executing function f(w/1×k), wecan derive the accumulative traffic matrix R for the time period ,in which ith row is the vector of the outgoing from node i and the jth column is the vector of the traffic destining to node j.

Applying Algorithm 1, we derive the following matrix R for our presented example. For simplicity, we assume the timing and hop-count thresholds do not filter any packet out.

$$R=\begin{bmatrix} 0 & 1 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}$$

It can be seen that, R contains not only all the one-hop packets captured by W_1 and W2 , but also a derived two- hop flow of size 0.5 from node 1 to node 3.

Traffic Pattern Discovery

The traffic matrix R tells us the deduced end-to-end traffic volume between each pair of nodes. However, we still need to perform further investigation to discover the actual Source/destination probability distribution and end-to-end link probability, that is, to figure out who are the actual source and destination and who are communicating with whom.

Source/Destination Probability Distribution

We denote the actual source and destination probability distribution, respectively, as two vectors S=(s(1),s(2),…,s(N))

And D=(d(1),d(2),…,d(N)), where s(i) and d(i) (i=1 to N) represent the probability for node i to be an actual source and destination respectively. Note that if the total number of source nodes is m, then we should have i=1 N s(i)=m for S. However, since we only care about the relative order among all possibilities(to know which nodes are more possible to be the actual sources) but not the total number m, we can always assume m = 1. It is the same case for D and all the probability vectors we will calculate later in this paper.That is, all probability distribution vectors in this paper are normalized[1] and only the relative orders among the elements of each vector actually make sense.

To derive S and D, we compute two series of vectors which converge to S and D, respectively: the source probability distribution vector series S = (S_0 . S1 . . . .Sn . . . .), and the destination probability distribution vector series D =(D0 . D1 . . . . . Dn . . . .).

First, both S_0 and D_0 should be uniform probabilitydistribution vectors: S_0 =D_0 = (1/N, 1/N, . . . , 1/N), sincewithout any traffic information, all nodes are equally likely to be sources and destinations.

Second, we note that the ith row (r(i; 1) . . . r(i; N)) in the matrix R is a vector of the traffic from node i to every node in the MANET. If we multiply this vector by D_0 (inner product), we get

$$\acute{S}(i)= \sum_{j=0}^{N} r(i,j) \times d0(j)$$

which is the probability for node i to be a source based on the destination probability distribution D_0 . This is intuitive, since if a node sends a lot of packets to another node with high probability of being a destination, the node itself has a high probability of being a source. According to this, the normalized inner product of R and D0 is a vector of probabilities for nodes to be a sources nodes. Similarly, using Ś to denote the vector (ś(1),ś(2),…,ś(N)) resulted from(1) and multiplying the ith row in the transpose of R(i.e.,) by Ś, we will get

$$d1(i)= \sum_{j=0}^{N} r(i,j) \times \acute{s}(j),$$

Which is the probability for node i to be a destination derived From Ś and in turn based on D0.This claim is based on the fact if a node receives a lot of packets from a node with high probability of being a destination. Consequently, the normalized inner product of R^T and S0 generates D_1 as a new probability vector for nodes to be destinations. Through this procedure, D1 is closer to the actual destination probability distribution than D0.

For the example scenario given in Fig. 1, we initialize D0 to be (1/3, 1/3, 1/3)T, , without any prior knowledge About the actual destinations.Then we computes Ś=R.D0=(1/2,1/3,0)T,which can be normalized to Ś=(3/5,2/5,0)T.Ś indicates that node 1is most likely to be an actual source, while node 3 is definitely not a source. Next, we multiply R^T by S0 and get the normalized D1 as:(0.15,0.46,0.39)T,which shoud be closer to the actual destination than D0.

According to the analysis above,we derive the following interative algorithm for D:

$$D_{n+1} = (RT.\ R)\ .\ D_n,$$ and similarly that for S:

$$S_{n+1} =(R\ \ RT)\ .\ S_n\ .$$

We also notice that geographically adjacent nodes may have negative impacts on the accuracy of the algorithms above. For example, if node j is one of the neighbors of node i, j may frequently forward the packets originated from node i to other nodes in the network and also frequently forward the packets from other nodes to node i.

In this case, the high probability for node j to be a source does not indicate the high probability for node i to be a destination, though the traffic volume from j to i is large. On the other hand, the high probability for node j to be a destination and the large traffic volume from i to j do not indicate the high probability for node i to be a source. We call this kind of negative impacts as the "neighborhood noise." Especially, when the mobility is low, the negative impacts will be substantial since the neighborhood of a node rarely changes.

To reduce the neighborhood noise, we utilize the vector space similarity assessment. The vector space similarity (or cosine similarity) of two vectors V and U is defined as follows:

Sim(V, U) =V . U/(|V||U|),

where V U denotes the dot product of V, and U, | V|, and |U| denote the ssnorm of V and U. We realize that, if two nodes have similar outgoing and incoming traffic vectors (in the end-to-end traffic matrix R), they are likely to be neighboring nodes (relays of each other), and so they should have less impact on the source/destination probability distribution of each other. Thus, we rewrite(1)and(2)by the following two formulas:

$$\acute{S}(i)= \sum_{j=1}^{N} r(i,j) \times d0(j) \times c(i,j),$$

$$d1(i)= \sum_{j=1}^{N} r(i,j) \times \acute{s}(j) \times c(i,j),$$

Where

(i,j)=c(j,i)= $(\mathrm{Sim}(O(I),O(j) + \mathrm{Sim}(I(I),I(j))/2,$

where O(i) and O(j) denote the ith row and jth row in R (i.e., the outgoing traffic from i and j), while I(i) and I(j) denote the ith and jth column in R (i.e., the incoming traffic to i and j).

Define a function ϕ such that $\phi(R) =(\phi(i,j))_{N\times N}$, where (i, j) = r(i,j) c(i; j). Obviously, we have $\phi(R^T) = \phi^T(R)$, in which $\phi^T(R)$ denotes the transpose of (R). According to (5) and (6), we improve (3) and (4) with the following two iterations, respectively:

$$D_{n+1} = (\phi^T(R). \phi(R))\ D_n,$$

$$S_{n+1} = (\phi^T(R). \phi(R))\ S_n,$$

By introducing the vector space similarity assessment, we ensure that, two nodes with higher probability to be neighbors (relays of each other) have less impact on each others source/destination probability distribution,which reasonably reduces the neighborhood noise. Finally, we propose the following Algorithms 3 and 4 to compute S and D

Algorithm 3. —Src(R).

```
$S_0$ = (1/N,1/N. . . . . ,1/N)
n = 0
do
$S_{n+1}$=($(R).$T(R)).Sn
Normalize $S_{n+1}$
n = n+1
while $S_n$=/= $S_{n-1}$
S = $S_n$
return S
```

Algorithm 4. —Dest(R).

```
$D_0$ = (1=N; 1=N; . . . ; 1=N)
n = 0
do
Dn+1=($t(R).$(R)).Dn
normalize Dn+1
n=n+1
While Dn=/=Dn-1
D=Dn
return D
```

The iterations will converge to S and D, which are the actual source and destination probability distribution vectors.

Based on the proposed algorithms, we can derive the twofinal vectors for the example scenario as given below. Here, we ignore the neighborhood noise reduction for simplicity:

$$S = (0.77, 0.23, 0)\ T\ ,$$

$$D\ = (0.08, 0.55, 0.37)^{T}\ .$$

End-to-End Link Probability Distribution

Our goal in this section is to derive a probability distribution matrix $P = (p(i, j))_{N\times N}$, in which each entry p(i, j) represents the probability of the i →j linkability (i.e.,node i and node j are a pair of actual source and destination). Again, note that only the relative order among these entries is of interest, since we aim at discovering the most possible communication links.

As described above, the probability for node i to be a destination depends on two factors: the traffic from each node j to node i and node j's probability to be a source. Suppose j i is an actual source-destination pair. If we set the total traffic coming out from j to zero, the probability for i to be a destination will decrease. Similarly, if we set the incoming traffic to node i to zero, the probability for node j to be a source will also decrease. Thus, we can identify a source-destination (S-D) pair by evaluating the significance of the probability reduction due to the elimination of the traffic sent by the source or received by the destination. For instance, in the example scenario shown in Fig. 1, to identify the most possible destination of node 1, we can erase all traffic sent by node 1 from the point-to-point traffic matrices, base on which we compute the destination probability distribution D . By comparing D with D (obtained using the original point-to-pointmatrices),we can find out the node whose destinationprobability drops most significantly due to elimination ofthe traffic sent by node 1. This node is most possible to be the destination of node 1. That said, we propose Algorithms 5 and 6 to discover the S-D linkability. The two algorithms are quite similar, so we only explain Algorithm 5 here. First, we apply Algorithm 1 (function f) to the original point-to-point traffic matrices and derive the original end-to-end traffic matrix R (line 1). Then we apply Algorithm 4 (function Dest) to R and obtain the original destination probability distribution vector D(line 2). Then, the point-to-point matrices are modified by eliminating the traffic sent by node i (line 3), and the destination probability distribution vector D.

The work flow of STARS is shown in Fig. 1.

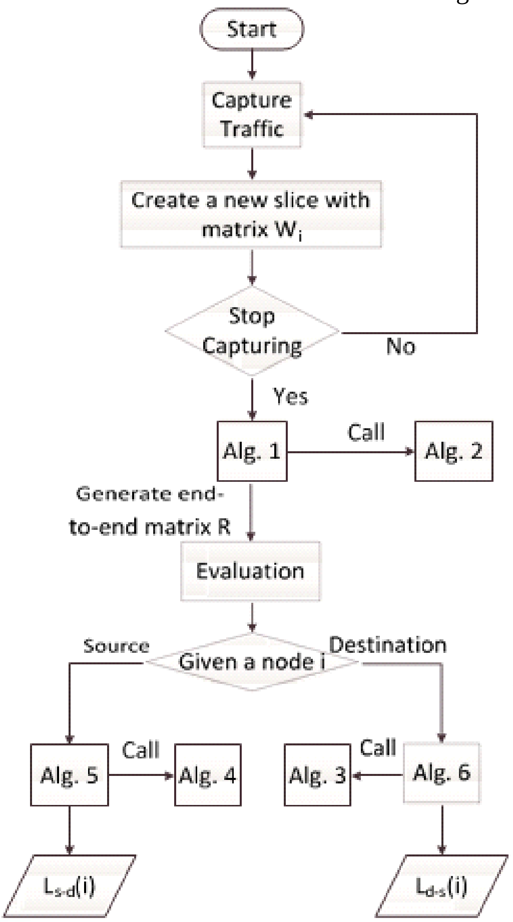


Fig. 1. Work flow of STARS

III. EXPERIMENTS

In this section, we present the empirical study, consisting of two components: demonstrationand evaluation. First,weuse three simple scenarios to demonstrate (partially) the direct outputs of STARS,i.e.,theprobability distributions. Then, we use the probabilitydistributions produced by STARS to identify the actual sources, destinations and end- to-end links for a large set of simulations, and evaluate the performance in terms of average false-positive rate (fpr) and false-negative rate (fnr). The network environment is simulated using Qualnet [26].The network protocol stack is modified so that the communication model presented in Section 3.1 is simulated.

Demonstrations

The MANET for demonstration is comprised of 30 mobile nodes randomly deployed in an 800×800 m^2 area. There are three different scenarios: (S1) Only one source (node 2) generates constant bit-rate (CBR) traffic to four destination (nodes 3, 6, 18, and 29). (S2) Four source nodes (nodes 2, 5, 11, and 20) generate CBR traffic to the only destination (node 3). (S3) CBR traffic is generated between 15 end-to-end communication pairs: 2-3, 5-6, 5-7, 7-8, 8-9, 8-10, 9-10, 10-11,10-12, 12-13, 12-14, 14-15, 15-16, 16-17, and 16-18. The simulation lasts for 200 seconds for each scenario. Other system parameters are shown inTable1.

TABLE1

System Parameters Configuration

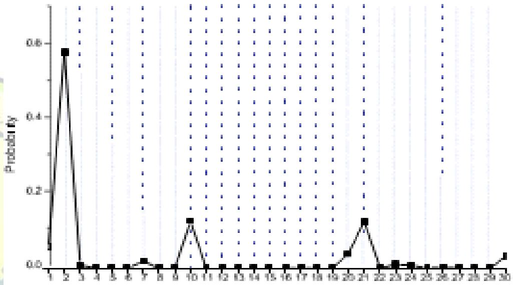
Node speed	Transmission Rate	Mobility Model	T(s)	H(flops)
5~10ms	11Mbs	Random Waypoint	1.0	5

Source/Destination Probability Distribution

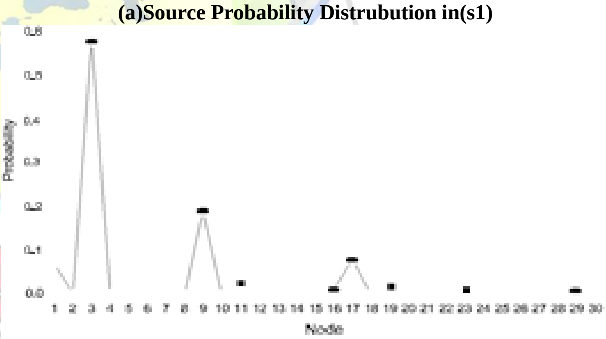
The first two scenarios demonstrate the ability of STARS to identify the source and destination by calculating the source/destination probability distribution. Figs. 2a and 2b are the source probability distribution of (S1) and the destination probability distribution of (S2), derived by Algorithms 3 and 4, respectively. In Fig. 2a, node 2 has much higher probability than other nodes to be the source, and in Fig. 2b, node 3 also has the highest probability to be the destination, which match the simulation setup

End-to-End Link Probability Distribution

Fig. 3 shows the results of applying Algorithm 5 to (S3). The results of applying Algorithm 6 are symmetric to those shown here, so they are not illustrated. In Fig. 5, the probability distribution for every node to be the intended destination is depicted for each source node. Most of these curves tell the truth of the actual traffic pattern. For example, in Fig. 5a, the highest peak is at node 3 (which means node 3 is most likely to be the intended destination of source node 2); in Fig. 5b, the highest peak is at node 6; in Fig. 5e, the highest peak is at node 10. All these results match the actual CBR traffic pattern perfectly. However some of the derived probability distributions have incorrect indications, such as in Fig. 5d, node 28 has the highest probability to be the destination of node 8. This is because some of the forwarders cannot be distinguished from the actual destination of a source or the actual source of a destination by using STARS, which means



(a)Source Probability Distrubution in(s1)



(b)Destination Probability Distribution in (s2) the MANET still has a certain level of communication relation anonymity under STARS

Evaluation

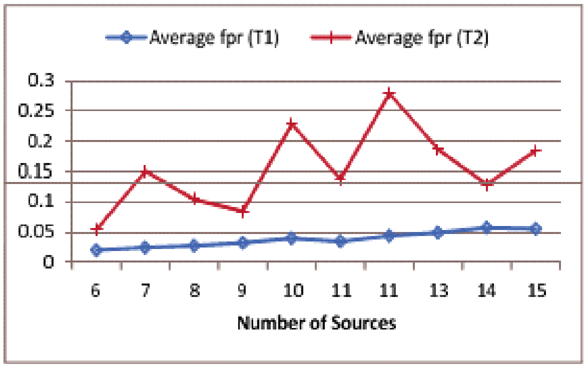
From the previous section, we see that the probability distributions produced by STARS are good indicators of the actual traffic patterns, i.e., actual sources, destinations, and end-to-end links. Different strategies can be used to speculate the actual traffic patterns from the probability distributions. In this section, we evaluate the performance of STARS based on the following two basic strategies, T1 and T2 .[T1] Suppose the number of actual sources, destinations,or end-to-end links is known to be k. We simply select the top k items (nodes or links)with the highestprobabilities.[T] Suppose the number k is unknown. We keep selecting the top items with the highest probabilities untilboth of the two criteria are satisfied: 1) the sum of the probabilities of the selected items has reached u; and 2) the probability of the last selected item is v times largerthan the current one. u and v are two adjustable thresholds, which are set to 0.8 and 4 in our experiments, respectively.The simulated MANET is comprised of 80 mobile nodes deployed in a 1,000 1,000 m^2 area.

The average values (over the 10 rounds for each case) of the false-positive rate and false-negative rate are shown in Fig. 6. From Figs. 6a and 6b, we can see that both T1 and T2 achieve reasonably good accuracy for source identification. Using T1 , the false-positive rate is almost always less than 0.05, although it increases slightly as the number of sources

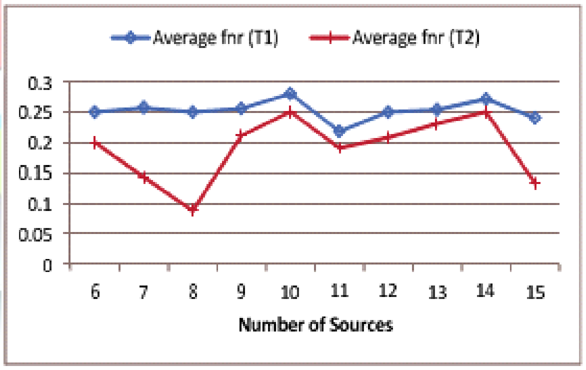
IV. Discussion And Future Work

The adversarial model presented in Section 3.2 assumes that the adversaries can globally monitor the traffic across the entire network region. This assumption is conservative from the network users' point of view. Usually, it is difficult for the attackers to perform such a global traffic detection. However, even though the adversaries are not able to monitor the entire network, they can monitor several parts of the network simultaneously. For example, an attacker can deploy sensors (signal detectors) around some particular mobile nodes to track their movements and eavesdrop all of their traffic. These sensors may even move accordingly. With the restricted capabilities, the attacker can take advantage of STARS to perform traffic analysis as follows:

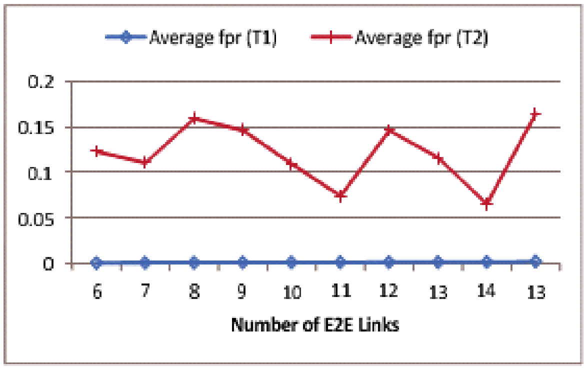
1. divide the entire network into multiple regions geographically;
2. deploy sensors along the boundaries of each region to monitor the cross-component traffic;
3. analyze the traffic even when nodes are close to each other by



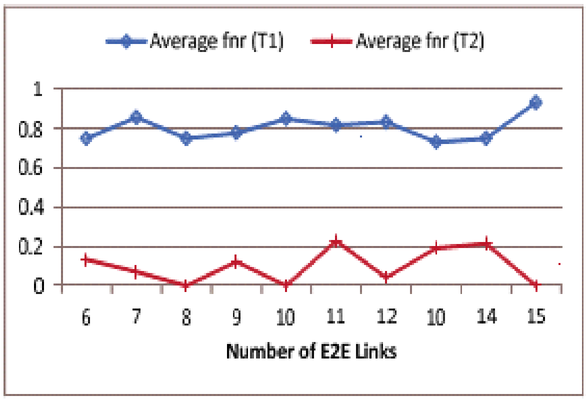
(a)Average False Positive Rate of Source Identification



(b)Average False Negative Rate of Source Identification



(c)Average False Positive Rate of E2E Link Identification



(d)Average False Negative Rate of E2E Link Identification

V. Conclusion

In this paper, we propose a novel STARS for MANETs. STARS is basically an attacking only needs to capture the raw traffic from the PHY/MAC layer without looking into the contents of the intercepted packets. From the captured packets, STARS constructs a sequence of point-to-point traffic matrices to derive the end-to-end traffic matrix, and then uses a heuristic data processing model to reveal the hidden traffic patterns from the end-to- end matrix. Our empirical study demonstrates that the existing MANET systems can achieve very restricted communication anonymity under the attack of STARS.

References

[1]. 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 Computing, vol. 6, no. 8, pp. 888-902, Aug. 2007.

[2]. Y. Zhang, W. Liu, W. Lou, and Y. Fang, “MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks,” IEEE Trans. Wireless Comm., vol. 5, no. 9, pp. 2376-2385, Sept. 2006.

[3]. Y. Qin and D. Huang, “OLAR: On-Demand LightweightAnonymous Routing in MANETs,” Proc. Fourth Int’l Conf. MobileComputing and Ubiquitous Networking (ICMU ’08), pp. 72-79, 2008

[4]. M. Blaze, J. Ioannidis, A. Keromytis, T. Malkin, and A. Rubin, “WAR: Wireless Anonymous Routing,” Proc. Int’l Conf. Security Protocols, pp. 218-232, 2005.

[5]. Aswiny.S, M.Suresh Chinnathampy ,“ NSDE Based Power Allocation and Relay Selection in Secure Cooperative Networks”, International Journal &amp; Magazine of Engineering,Technology, Management and Research, Volume No: 3 (2016), Issue No: 5 (May), Page 79-82

[6]. [6] S.Esakki Rajavel, C.Jenita Blesslin, “Energetic Spectrum Sensing For Cognitive Radio Enabled Remote State Estimation Over Wireless Channels”, International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Vol. 3, Special Issue 19, April 2016 (12 – 15).

[7]. R. Shokri, M. Yabandeh, and N. Yazdani, “Anonymous Routing in MANET Using Random Identifiers,” Proc. Sixth Int’l Conf. Networking (ICN ’07), p. 2, 2007.

[8]. R. Song, L. Korba, and G. Yee, “AnonDSR: Efficient Anonymous Dynamic Source Routing for Mobile Ad-Hoc Networks,” Proc. Third ACM Workshop Security of Ad Hoc and Sensor Networks (SASN ’05), pp. 33-42, 2005.

[9]. M. Reed, P. Syverson, and D. Goldschlag, “Anonymous Connections and Onion Routing,” IEEE J. Selected Areas in Comm., vol. 16, no. 4, pp. 482-494, May 2002.

[10]. D. Chaum, “Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms,” Comm. ACM, vol. 24, no. 2, pp. 84-88, 1981.

[11]. J. Raymond, “Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems,” Proc. Int’l Workshop Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobserva- bility, pp. 10-29, 2001.

[12]. W. Dai, “Two Attacks against a PipeNet-Like Protocol Once Used by the Freedom Service,” http://weidai.com/freedom-attacks.txt, 2013.

[13]. X. Wang, S. Chen, and S. Jajodia, “Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems,” Proc. IEEE Symp. Security and Privacy, pp. 116-130, 2007.

[14]. M. Reiter and A. Rubin, “Crowds: Anonymity for Web Transactions,” ACM Trans. Information and System Security, vol. 1, no. 1, pp. 66-92, 1998.